LABORATORY NOTEBOOK

CNWRA / SWRI

Goodluck Ofoegbu
(210) 522-6641

NOTEBOOK NO. 263
ISSUED TO G I Ofoegbu
ON April 01 19 98
DEPARTMENT CNWRA (Div 20)
RETURNED 19

—SCIENTIFIC NOTEBOOK CO.—
2831 LAWRENCE AVE.
P.O. BOX 238
STEVENSVILLE, MI 49127
616-429-8285

INSTRUCTIONS

1. **The primary purpose of this notebook is to protect your and the Company's Patent-Rights by keeping records of all original work in a form acceptable as evidence if any legal conflict arises.**

2. • When starting a page, enter the title, project number, and book number.
   • Use <u>ink</u> for permanence -- avoid pencil.
   • Record your work as you progress, including any spur-of-the-moment ideas which may be developed later.
   • Avoid making notes on loose paper to be recopied.
   • Record your work in such a manner that a co-worker can continue from where you stop. You might be ill and to protect your priority it could be urgent that the work continue while you are absent.

3. • Give a complete account of your experiments and the results, both positive and negative, including your observations.
   • Record all diagrams, layouts, plans, procedures, new ideas, or anything pertinent to your work including the details of any discussions with suppliers, or other people outside the Company.
   • Do not try to erase any incorrect entries; draw lines deleting them, note the corrections, sign and date the changes. This extra care is worthwhile because of the necessity of original data to prove priority of new discoveries.

4. • After entering your data, sign and date the entries.
   • Explain your work to at least two witnesses who are not co-inventors, and have them sign and date the pages in the place provided.
   • Record the names of operators and witnesses present during any demonstration and have at least two witnesses sign the page. If no witnesses are present during an experiment of importance, repeat it in the presence of two witnesses.

5. Since computer programs can be patented these instructions apply to the development of computer software. In this case a description of the structure and operation of the program should be recorded in the notebook, together with a basic flow diagram which illustrates the essential features of the program. In the course of developing the code, the number of lines of code written each day should be recorded in the notebook, together with a statement of the portion of the flow diagram to which the section of code is directed.

6. This notebook and its contents are the exclusive property of the Company. It is confidential and the contents are not to be disclosed to anyone unless authorized by the Company. You must return it when completed, upon request, or upon termination of employment. It should be kept in a protected place.
   If loss occurs, notify your supervisor immediately, and make a written report describing the circumstances of the loss.

Form 590

TITLE Repository Scale Thermal-Mechanical (TM) Model (2D) Project No. 20-1402-671
Book No. March 2 1998 1

From Page No.____

GLO

Two ~~Dimensional~~ Dimensional Repository Scale Thermal-Mechanical Model

Objective: To examine spatial and temporal variations of tunnel stability at the repository horizon considering spatial variations of rock-mass quality (derived from ESF mapping) and possible temporal variations caused by degradation of *intact* rock under sustained long-term loading and wall-rock weathering of rock joints.

1. Analysis will be based on a north-south vertical 2D section along the exhaust main (drift).

2. Section is about 3,500 m long north to south. Each model will be executed in two stages because of the large vertical extent required to simulate the appropriate thermal and mechanical boundary conditions: First, a model extending over a vertical height of about 1,000 m (elevation 500 to 1,500 m above sea level), referred to as the *full model*, will be analyzed using a coarse mesh to develop the histories of temperature and vertical displacement at the top and base of the second, smaller model. The vertical height of the second model, referred to as the *submodel*, will be limited; its exact top and base elevations will be determined from preliminary analyses of the full model.

3. Only the emplacement drifts will be represented in the model. Each emplacement drift will be modeled as a square opening separated from adjacent drifts by pillars.

4. Each analysis will consist of a heat-conduction stage followed by a thermal-mechanical stage based on linear elasticity. Mechanical analyses of the submodel will include processing of the calculated stresses using a failure criterion to identify zones of overstress.

5. Rock-mass strength will be modeled using the Hoek-Brown failure criterion, with parameters calculated using NGI Q values developed by the DOE based on ESF data. Rock-mass degradation due to joint wall-rock weathering will be represented through a degradation of Q values.

6. The hypothesized rock-mass degradation process is likely to depend on time and the availability of moisture, warm temperature, and oxygen. However, the dependence of rock-mass quality on these factors is currently not known quantitatively. We will be able to estimate a long-term rock-mass quality from the current Q values and our hypothesis of rock-mass degradation from wall-rock weathering.

7. Our strategy for evaluation of the analysis results is to examine tunnel stability (through over-stress analyses) at selected times using both the current and estimated long-term rock-mass quality (Q) values. Any differences between the tunnel stability calculated using the current and long-term Q values may help determine possible impacts of rock-mass degradation on repository design and performance.

GLO

Investigator: Goodluck I Ofoegbu
Initial: GIO Signature: [signature]

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.

Pages 2-11 entries by GWO 3/2/98

Problem Geometry

Problem geometry was determined using information from two DOE documents:

(1) Repository Thermal Loading Management Analysis.
CRWMS/M&O Document Number
B00000000-01717-0200-00135 Rev 00
Document referred to hereafter as
RTLM-97. GWO 3/2/98

(2) Repository Subsurface Layout Configuration Analysis.
CRWMS/M&O Document Number
BCA000000-01717-0200-00008 Rev 00
Document referred to hereafter as
RSLC-97.

Information Used to Set Problem Geometry

(1) Emplacement drift diameter is 5.5 m
Concrete lining thickness = 0.2 m
Therefore internal diameter of emplacement drift is 5.1 m.
[p. 21 of RSLC-97]

(2) For the selected thermal loading of 85 MTU/acre [RTLM-97, p. 25] the drift (center-to-center) spacing is 28 m [RTLM-97, p. 38] GWO 3/2/98

(3) The minimum waste-package spacing is 13.26 m to accomodate ~~for~~ GWO 3/2/98 HLW packages between the commercial SNF packages.

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No. 2

GWO 3/2/98

(4) Length of emplacement drifts required on a year-by-year basis is given on Table 7.7-1, p. 54, of RTLM-97. The total length of emplacement drift required is 107,150 m.

(5) Average usable length for individual emplacement drifts is 1080 m [RTLM-97, p. 53].

(6) The average usable length of 1080 m per drift implies that 100 drifts will be required to provide the total drift length of 107,150 m.

(7) The ~~proposed repository~~ layout gives number of emplacement drifts as 105 with ~~an~~ GWO 3/2/98 additional 15 drifts for contingency [RSLC-97, p. 33 GWO 3/2/98 Fig. 7-1]. 100 drifts will be modeled.

Summary of Problem Geometry

(1) Problem domain is 2800 m long (south to north). Domain will be extended 200 m to the south and 200 m to the north to reduce truncation of the temperature and displacement fields. Zero-perturbation conditions (no normal displacement and no temperature change) will be applied at the north and south extended boundaries.

GWO 3/2/98

To Page No.

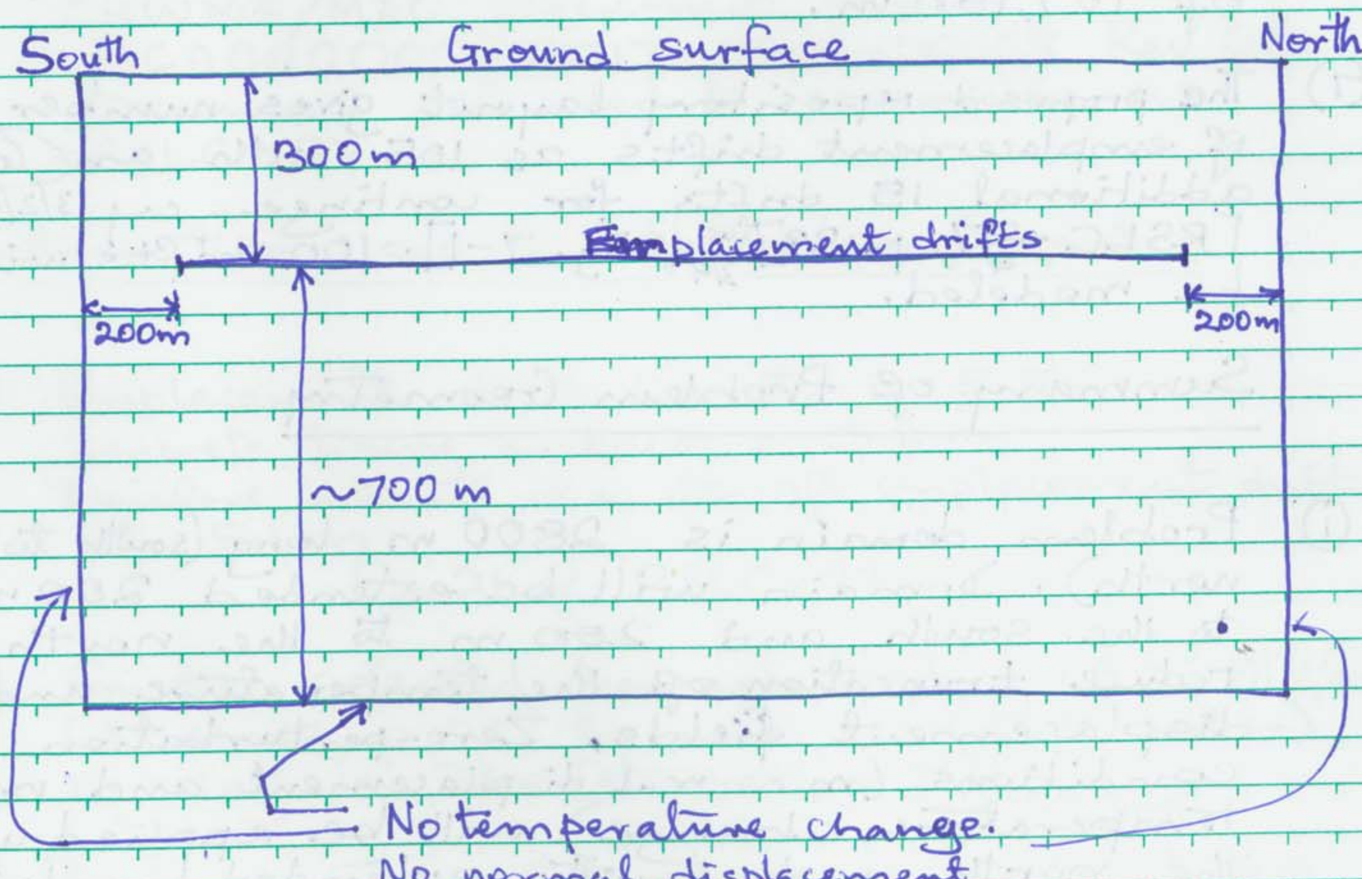
Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No. 3

(2) The top of the model domain, i.e., the ground surface, will be assumed to be horizontal.

(3) Emplacement drifts will be placed at a depth of about 300 m below the ground surface.

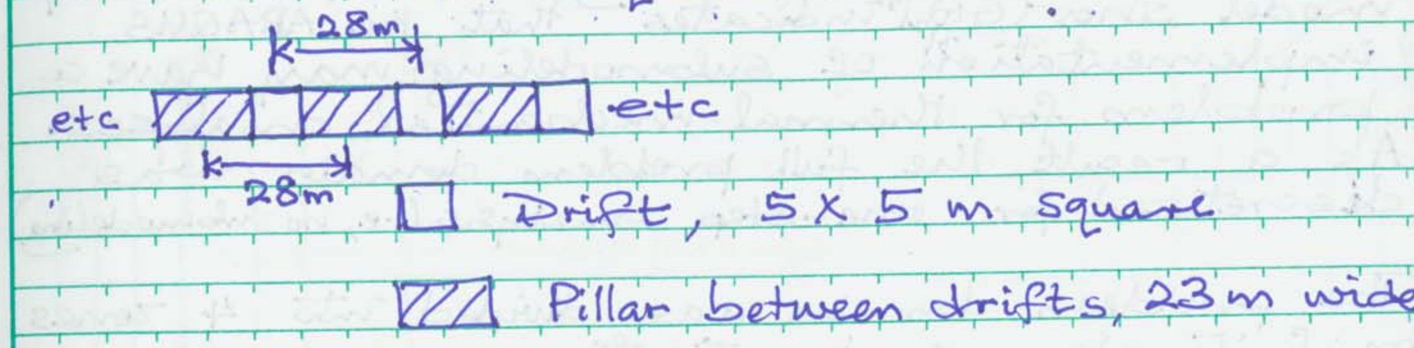
(4) The base of the model will be placed at about 700 m below the repository (1000 m below the ground surface) to reduce truncation of temperature and displacement fields. Zero perturbation conditions will be applied at the model base.



To Page No.

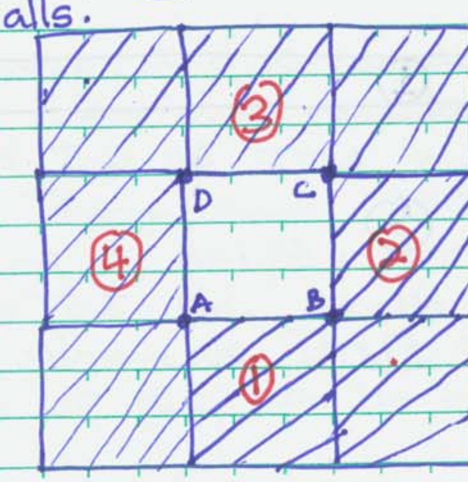
From Page No. 4

(5) Each emplacement drift will be represented as a 5 x 5 m square.



Tunnel Lining

To simplify model, tunnel lining will be simulated using beam elements placed on the tunnel walls.



AB, BC, AD, and DC are beam elements as well as the near-tunnel edges of elements (1), (2), (4), and (3), respectively.

Each of the fictitious beams will be assigned relatively high Young's modulus and low Poisson's ratio to simulate stiff tunnel lining.

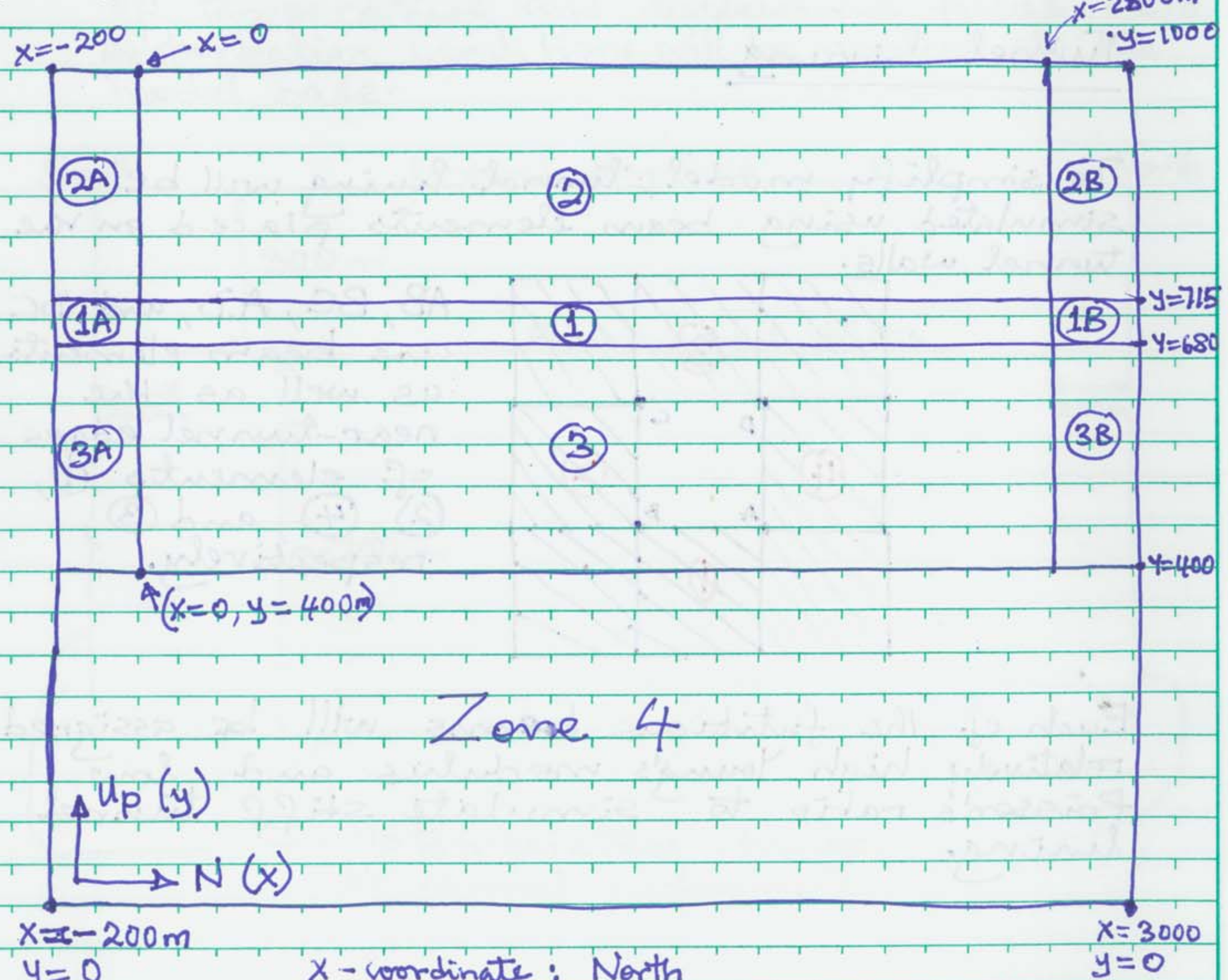
To Page No.

From Page No.___

Finite Element Discretization

Submodeling approach described on p. 1 was abandoned because testing on a preliminary model ~~show~~ (GLP 9/21/98) indicates that the ABAQUS implementation of submodeling may have a problem for thermal-mechanical analyses. As a result, the full problem domain was discretized for one-step analysis (i.e., no submodeling)

The problem domain was divided into 4 zones for finite element discretization:



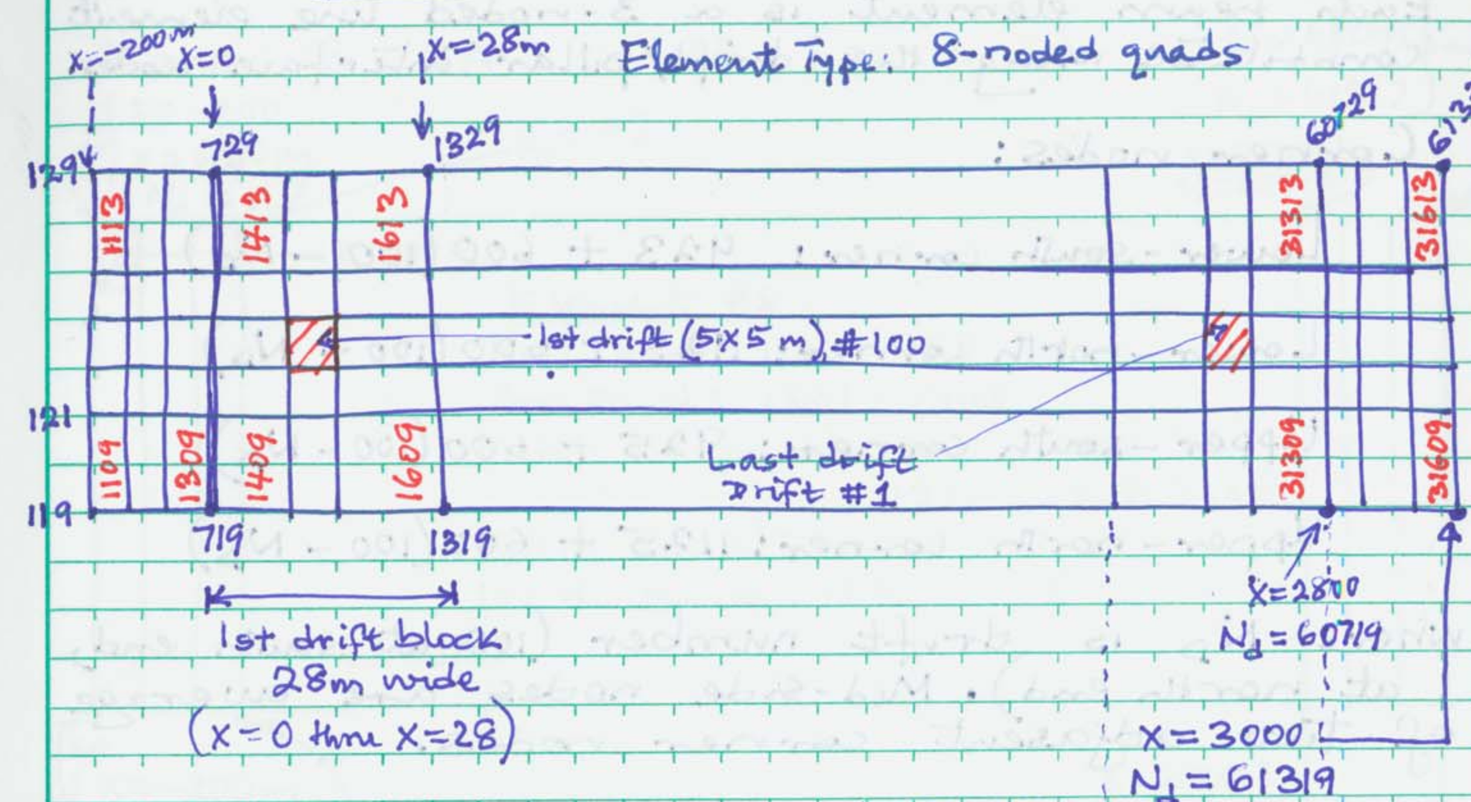
X - coordinate: North
y - coordinate: Up

To Page No.___

From Page No.___

Node and element numbering for Zones (1), 1A, and 1B (Middle Zone)

Element Type: 8-noded quads

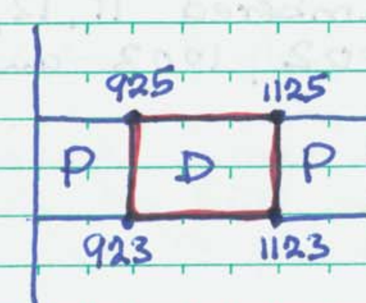


Node numbers and x-coordinates (m) in blue. Element numbers in red. y=715 m at top of zone and 680 m at base. There are 100 drift blocks each containing 1 drift. Drift #100, first drift in the model is at the south end of the drift array whereas Drift #1, last drift in the model, is at the north end.

Last drift block (ending at x=2800)

Beam Elements

Example: Drift #100 at South end of drift array.



Corner nodes: 923 1123
925 1125

D = drift (opening)
P = pillar (rock)

To Page No.___

From Page No.

Each beam element is a 3-noded line element constituted using the drift/pillar interface nodes.

Corner nodes:

Lower-south corner: $923 + 600(100 - N_D)$

Lower-north corner: $1123 + 600(100 - N_D)$

Upper-south corner: $925 + 600(100 - N_D)$

Upper-north corner: $1125 + 600(100 - N_D)$

where N_D is drift number (100 at south end, 1 at north end). Mid-side nodes are average of two adjasent corner nodes.

Beam element numbers are:

Floor beam: $11 + 10(100 - N_D)$

North side-wall beam: $12 + 10(100 - N_D)$

Roof beam: $13 + 10(100 - N_D)$

South side-wall beam: $14 + 10(100 - N_D)$

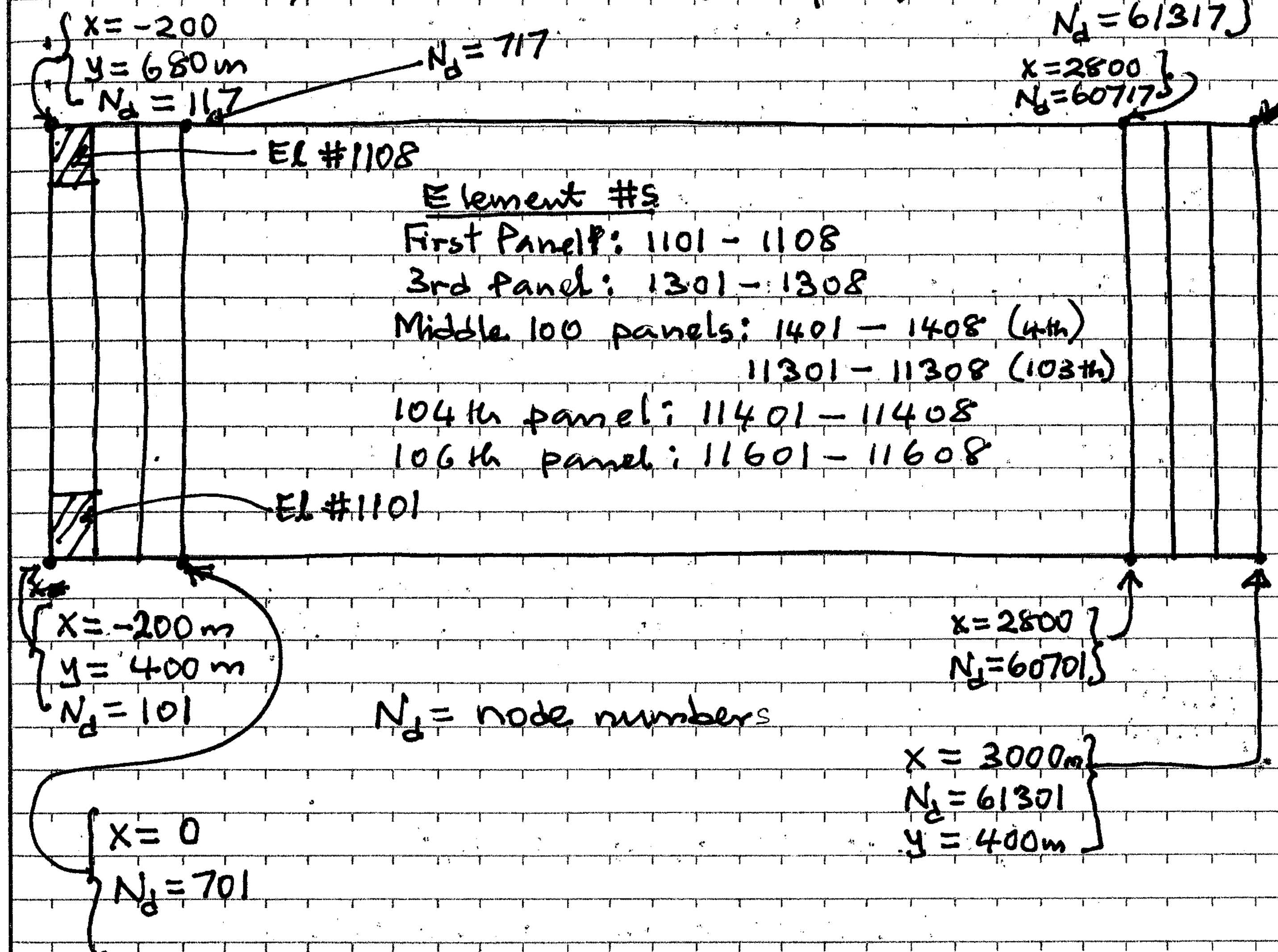
Hence beam elements are numbered 11, 12, 13 and 14 for drift #100; and 1001, 1002, 1003 and 1004 for drift #1 (north end).

To Page No.

From Page No.

Node and Element Numbering for Zone 3 (Base Zone)

Zones 3, 3A and 3B (see p. 6)



106 vertical panels (3 panels from x = -200 to x = 0
100 panels from x = 0 to x = 2800
3 panels from x = 2800 to x = 3000

Each panel has 8 elements (stacked vertically)

Node numbers increment horizontally by 200 within first 3 panels, by 600 within panels 4 - 103, and by 200 within last 3 panels.

To Page No.

From Page No.

$N_d = 101 + 200(P_n - 1)$ $\quad P_n = 1 - 3$

$701 + 600(P_n - 1)$ $\quad P_n = 4 - 103$

$60701 + 200(P_n - 1)$ $\quad P_n = 104 - 106$

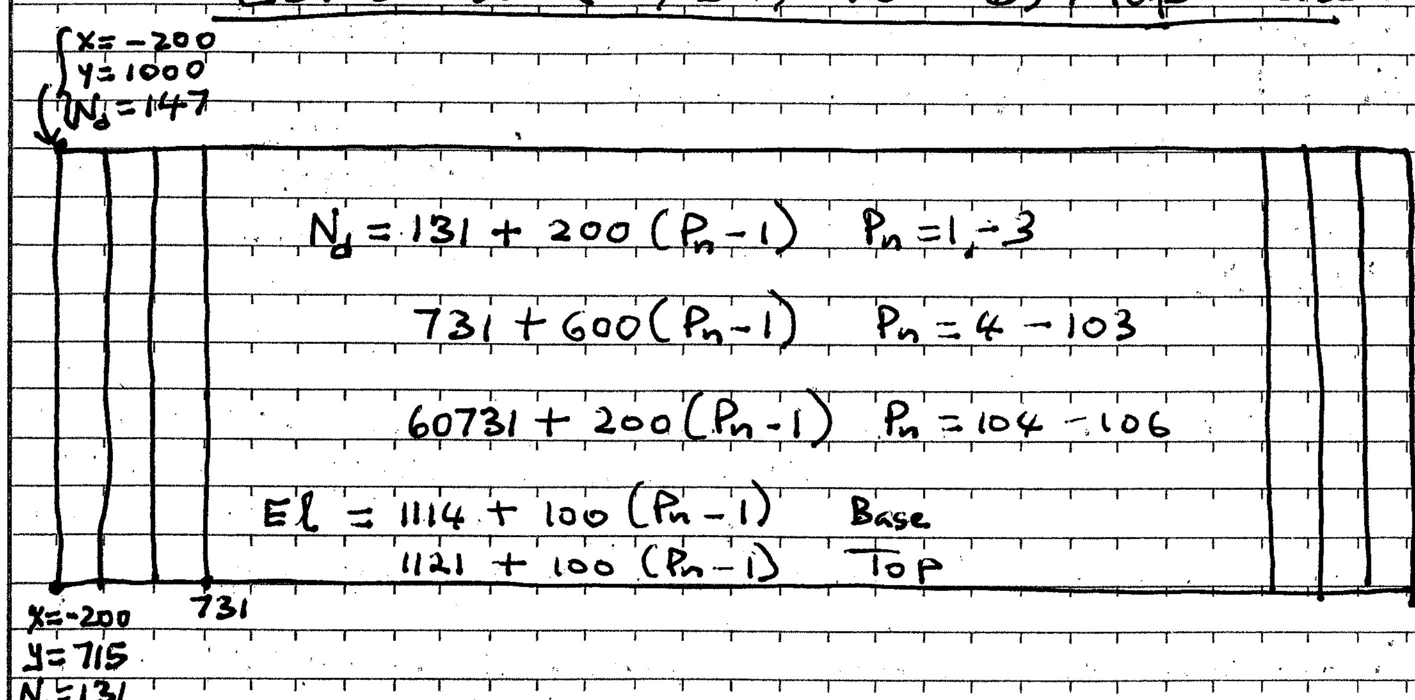
P_n = panel number.

$El = 1101 + 100(P_n - 1)$ at base

$1108 + 100(P_n - 1)$ at top.

El = element number.

Element and Node Numbers in Zone 2 (2, 2A, and 2B); Top Zone



$N_d = 131 + 200(P_n - 1)$ $\quad P_n = 1 - 3$

$731 + 600(P_n - 1)$ $\quad P_n = 4 - 103$

$60731 + 200(P_n - 1)$ $\quad P_n = 104 - 106$

$El = 1114 + 100(P_n - 1)$ Base

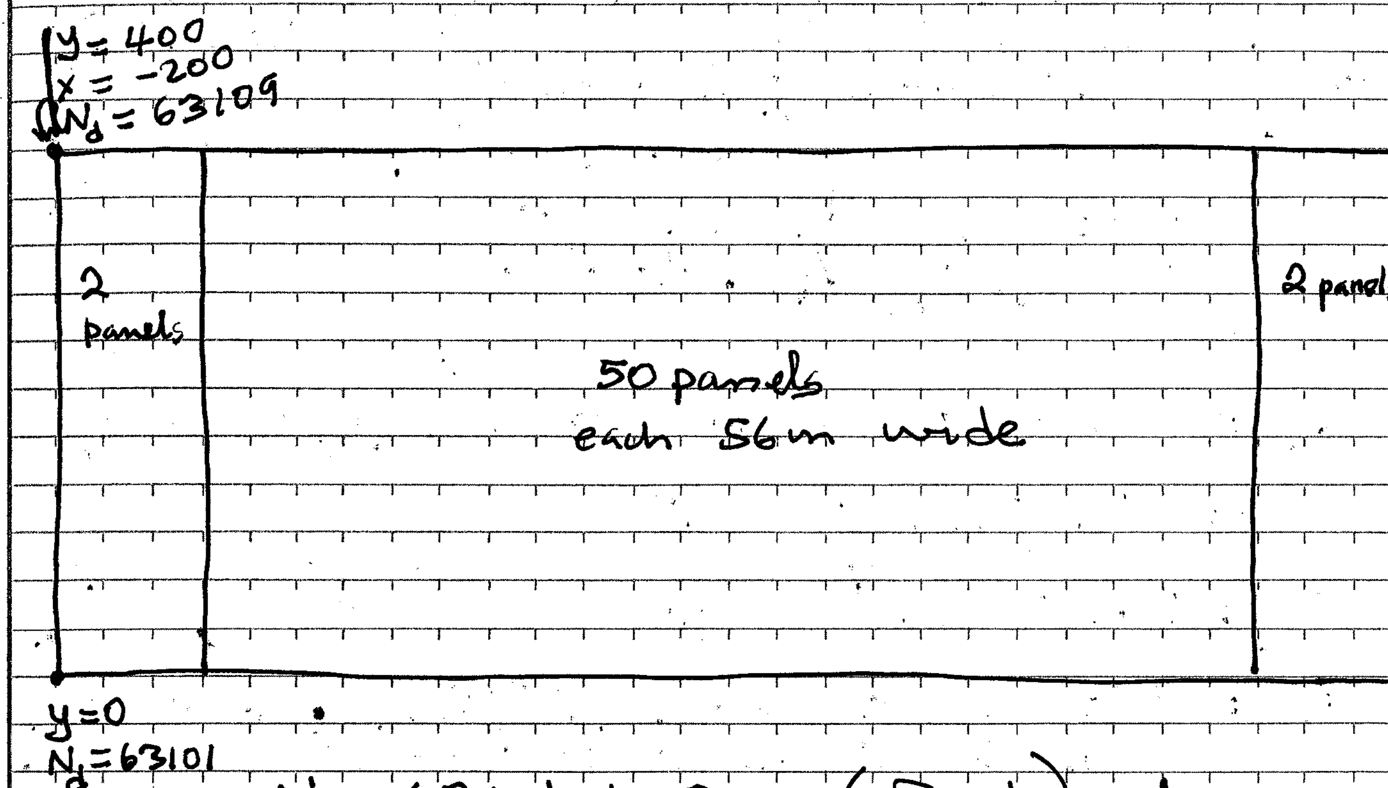
$1121 + 100(P_n - 1)$ Top

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.

Element and Node Numbering for Zone 4 (Far Base)



$N_d = 63101 + 200(P_n - 1)$ base

$63109 + 200(P_n - 1)$ top

$P_n = 1 - 54$.

$El = 33101 + 100(P_n - 1)$ base

$33104 + 100(P_n - 1)$ top.

Drift number D_n and panel number P_n are related as follows:

$$\left.\begin{aligned} D_n &= 104 - P_n \\ P_n &= 104 - D_n \end{aligned}\right\} \text{ for } \begin{aligned} 1 \le D_n \le 100 \\ 4 \le P_n \le 103 \end{aligned}$$

GW ~~Jan~~ 15 1999

To Page No.

Witnessed & Understood by me,	Date 2-11	Invented by	Date
Pages entries by		Recorded by GW 3/2/98	

12–17 entries

From Page No.

Thermal Load

Pages 12–17 by GW 3/27/98

Thermal load will be modeled as a volumetric heat source distributed uniformly within and among the emplacement drifts. The volumetric source q is given by

$$q = \left(\frac{Q}{100}\right)\left(\frac{1}{V_d}\right)\left(S_y\right)\left(10^6\right)$$

V_d = emplacement drift volume (of 1 drift)

$= 5 \times 5 \times 1080 \ m^3$

$S_y = 3.156 \times 10^7 \ s/yr$

Q = total heat source (10^6 J/s) per unit time due to all 10,938 waste packages.

Values of Q were obtained from Table V-1 (Attachment V) of Reference 1 (see p. 2). The table of Q versus yr (following emplacement) is reproduced on p. 13.

Heat load ~~was~~ will be (GW 3/27/98) applied at the same time over all 100 drifts.

Thermal Properties

All elements are assigned TSw2 thermal properties (from Reference 1 cited on p. 2) as follows:

Density $\rho = 2274 \ kg/m^3$

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.

Years	HeatSource (MW) (10^6 J/s)
0	71.499486
5	63.518801
10	57.65222
20	48.598077
30	41.428829
40	36.000998
50	31.544301
60	27.912077
70	24.90998
80	22.386533
90	20.530418
100	18.942351
150	14.040354
200	11.499602
300	9.02496
400	7.585398
500	6.574892
600	5.77481
700	5.149266
800	4.610581
900	4.179814
1000	3.817243
2000	2.04884
3000	1.59641
4000	1.425174
5000	1.328105
6000	1.229386
7000	1.151405
8000	1.078041
9000	1.017151
10000	0.965009

GW 3/27/98

Thermal conductivity

$K = 2.1 \ J/s.m.K$

$= 6.6271 \times 10^7 \ J/yr.m.K$

Specific heat capacity: Function of temperature as follows:

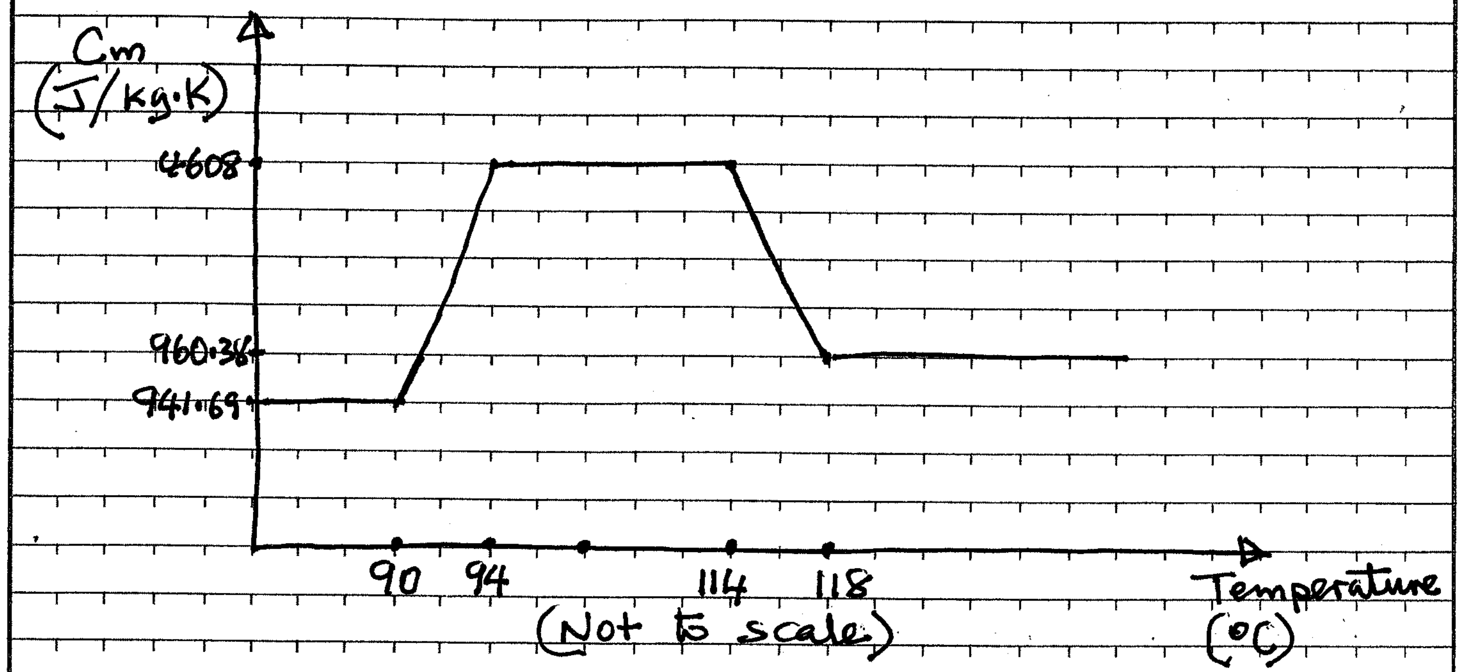
To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.

Temperature (°C)	C_v (J/m³·K)	C_m (J/kg·K)
25 — 94	2.1414×10^6	941.69
94 — 114	10.4786×10^6	4608.0
>114	2.1839×10^6	960.38

The heat capacity vs temperature relationship ~~will~~ be implemented as follows to avoid infinite gradient



Mechanical Properties

Thermal expansivity
Young's modulus
Poisson's ratio
Friction angle
Cohesion
Unconfined compressive strength

To Page No.

From Page No.

Thermal expansivity is implemented as a function of temperature as follows:

Temperature (°C)	Thermal expansivity /(°C)
0.0	5.07×10^{-6}
29.0	5.07×10^{-6}
51.0	7.30×10^{-6}
98.0	7.30×10^{-6}
102.0	8.19×10^{-6}
148.0	8.19×10^{-6}
152.0	8.97×10^{-6}
200.0	8.97×10^{-6}

Poisson's ratio is ~~was~~ GM 3/27/98 assigned constant value of 0.21 (from Reference 3 on this page).

Unconfined compressive strength of intact rock (σ_{ci}) is assigned constant value of 180 MPa (from Reference 4 on this page).

Values of Young's modulus (E), friction angle (Ø) and cohesion (c) were determined from rock-mass quality Q, which varies spatially in the north-south direction as was determined from DOE data (attached as Attachment 1 on p. 16-17).

References.
(3) Repository Ground Support Analysis for Viability Assessment. CRWMS/M&O Document Number BCAA00000-01717-0200-00004 Rev 00.

(4) Brechtel, C.E., M. Lin, E. Martin, and D.S. Kessel. Geotechnical Characterization of the North Ramp of the Exploratory Studies Facility, Volumes I of II. Sandia Report SAN95-0488/1, (1995).

To Page No.

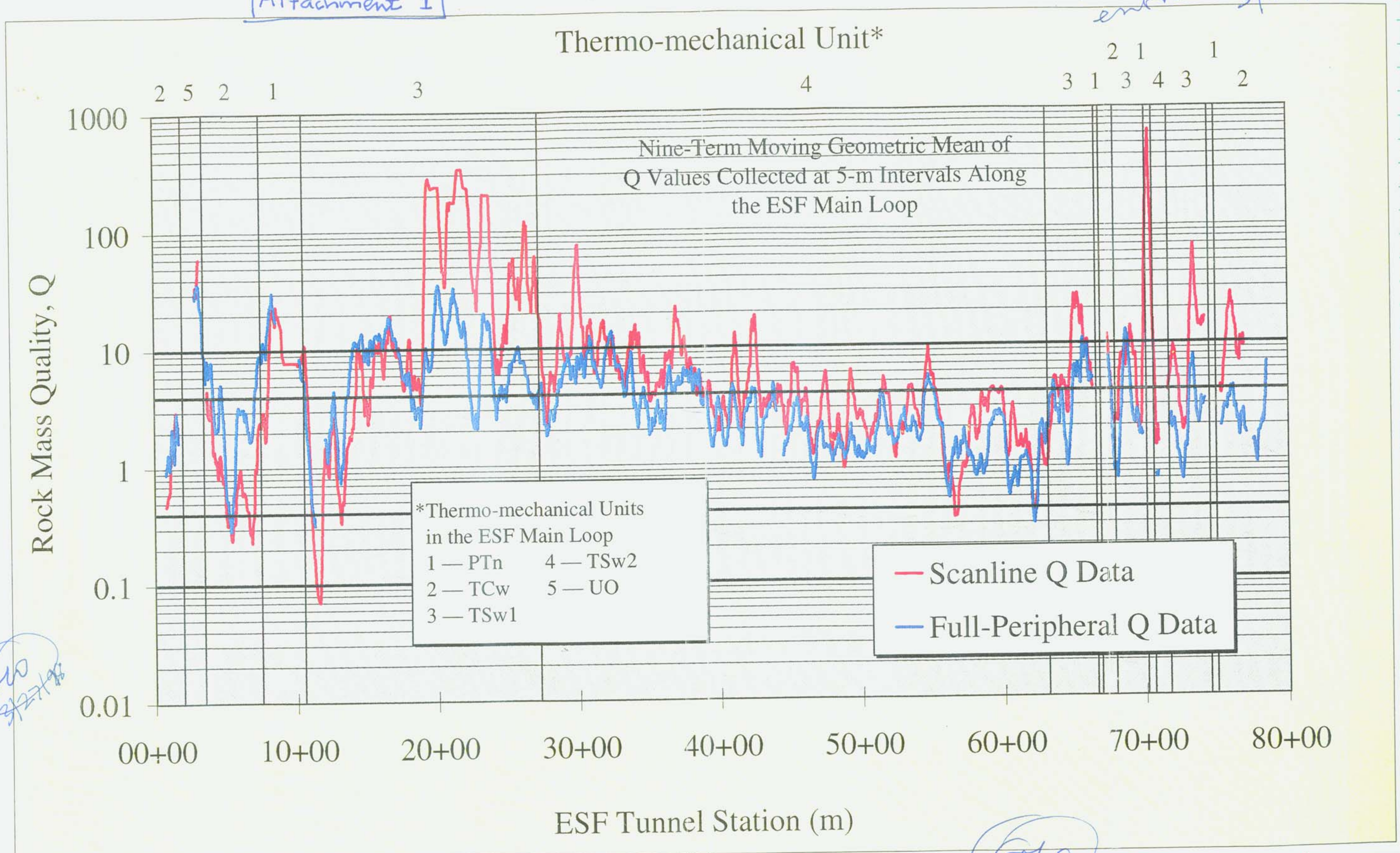
GLO 3/27/98

Attachment 1

GLO 3/27/98

Pages 12-17 entries by GLO 3/27/98

ESF ROCK MASS QUALITY



GLO 3/27/98

TITLE April 17 1998

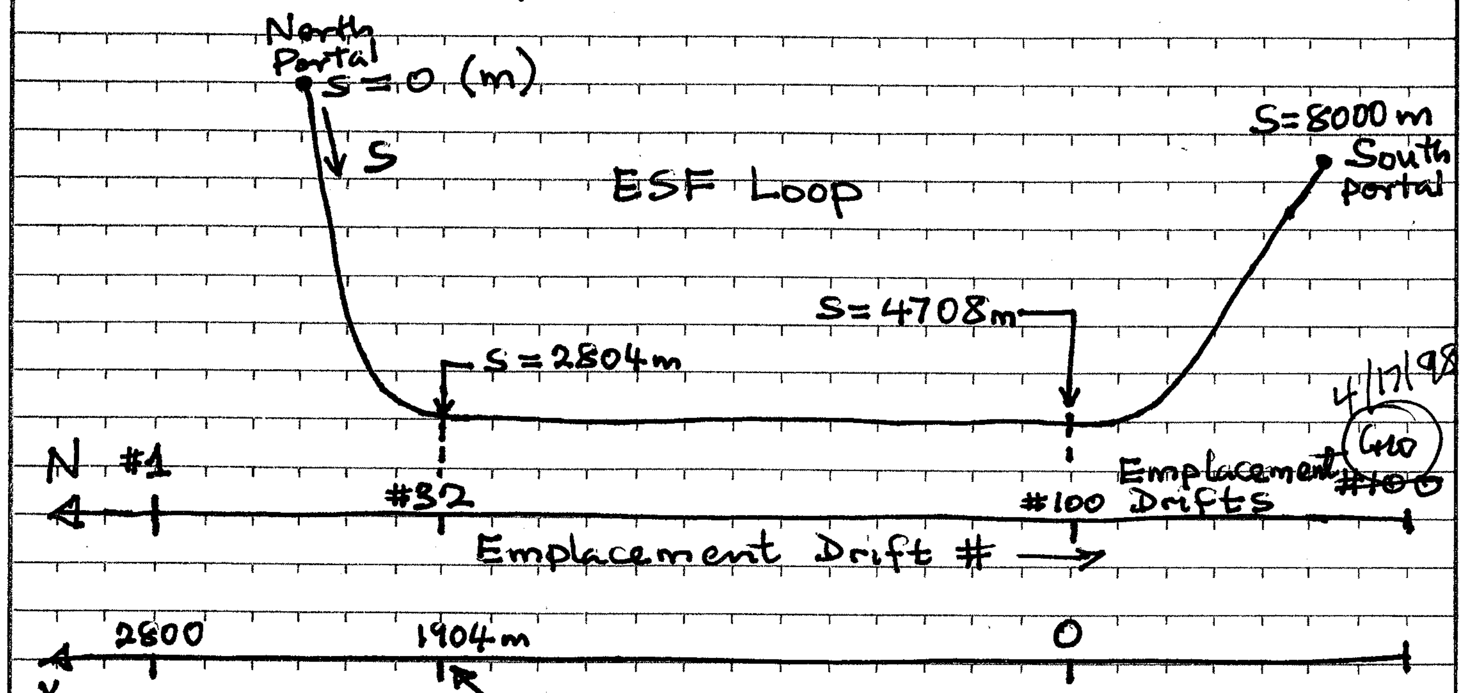
From Page No.____

Rock Mass Quality (Q)

Pages 18-24 entries by GLO 4/17/98

$Q = Q(x,t)$; i.e., Q is a function of north-south coordinate x and time t.

Function $Q(x, t=0)$ is evaluated from ESF data (p. 16-17). ESF data gives Q as a function of North-to-South distance, S, along the ESF main drift.



$x = 2800 - (32 \times 28)$
$= 1904$ m

$$X = 1904 - (S - 2804) \qquad 2804 \le S \le 5000$$

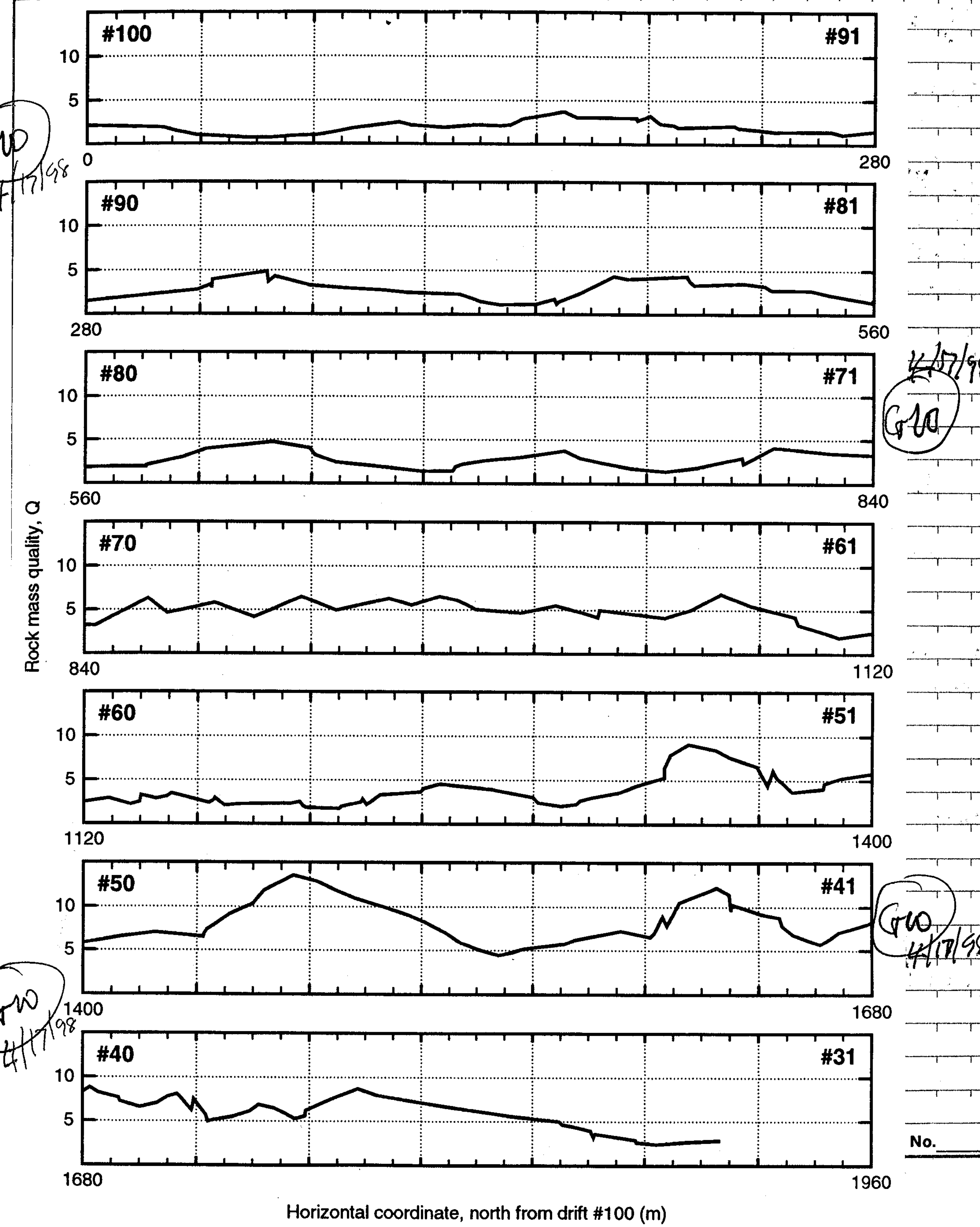
$$S = 2804 + (1904 - X) \qquad -200 \le X \le 1904$$

Q values from the full-peripheral line (p. 16-17) between S = 2804 m and S = 5000 m were used to obtain Q values for $-200 \le X \le 1904$ m.

To Page No.________

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.____ Values of Q at $X > 1904$ m were set to the value at $X = 1904$ m.



No.________

From Page No.

Mohr-Coulomb Parameters for Rock Mass

$\tau_f = c + \sigma_n \tan\phi$

where τ_f = shear strength of rock mass on an internal plane experiencing normal stress σ_n. The strength parameters, cohesion (c) and friction angle (ϕ) are estimated from Q using relationships from Hoek and Brown (1997) [Hoek, E. and E.T. Brown. 1997. Practical estimates of rock mass strength. Int. J. Rock Mech. Min. Sci., 34(8): 1165-1186].

These relationships require:

(1) Hoek-Brown intact rock strength parameter, m_i

(2) Hoek-Brown geological strength index, GSI; and

(3) unconfined compressive strength of intact rock, σ_{ci} (see p. 15).

From Hoek (1994) [Hoek, E. 1994. Strength of rock and rock masses. ISRM News Journal, 2(2): 4-16]

$$GSI = 9 \ln Q + 44$$

Also, from Ref (4) on p. 15,

$m_i = 10$ for TSw2 rock unit.

Charts on Figures 7 and 8 of Hoek and Brown (1997) [cited on this page] are used to evaluate the functions

$$\phi = \phi(GSI, m_i = 10)$$

$$\frac{c}{\sigma_{ci}} = \frac{c}{\sigma_{ci}}(GSI, m_i = 10)$$

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.

GSI	ϕ ($m_i = 10$)	c/σ_{ci} ($m_i = 10$)
1.0	10 degrees	0.006
10	18.3 ✓	0.013
20	22.5 ✓	0.018
30	26.2 ✓	0.03
40	28.0 ✓	0.035
50	31.2 ✓	0.041
60	33.2 ✓	0.054
70	36.2 ✓	0.07
80	37.5 ✓	0.10
90	38.5 ✓	0.15

Information potentially subject to copyright protection was redacted from this location. The redacted material is from the following reference:

Hoek, E. and E.T. Brown. "Practical Estimates of Rock Mass Strength." International Journal of Rock Mechanics-Mineral Science. Vol. 34, No. 8. pp. 1,165-1,186. 1997.

Information potentially subject to copyright protection was redacted from this location. The redacted material (Figure 7) is from the following reference:

Hoek, E. and E.T. Brown. "Practical Estimates of Rock Mass Strength." International Journal of Rock Mechanics, Mineral and Science. Vol. 34, No. 8. pp. 1,165-1,186. 1997.

Rock Mass Young's Modulus

Young modulus, E, for rock mass is evaluated using relationships from Hoek (1994) [cited on p. 20]

$$E = 10^{[(RMR-10)/40]} \quad \text{GPa}$$

$$RMR = 9\ln(Q) + 49$$

From Page No.

Drucker-Prager Strength Parameters from Mohr-Coulomb Parameters

Rock mass strength is modeled using the Drucker-Prager strength criterion, which defines rock strength in terms of a friction angle β and a shear-strength intercept d.

For plane strain conditions, β and d are calculated from c and ϕ as follows:

$$\left.\begin{aligned} \tan\beta &= \frac{\sqrt{3}\sin\phi}{\sqrt{1+\frac{1}{3}\sin^2\phi}} \\ d/c &= \frac{\sqrt{3}\cos\phi}{\sqrt{1+\frac{1}{3}\sin^2\phi}} \end{aligned}\right\}$$

For associative flow, i.e., $\psi/\phi = 1$

where ψ is dilation angle.

$$\left.\begin{aligned} \tan\beta &= \sqrt{3}\sin\phi \\ d/c &= \sqrt{3}\cos\phi \end{aligned}\right\}$$

For non-dilatant flow, i.e., $\psi/\phi = 0$

Analyses were conducted with dilation angle equal to half of friction angle. Therefore, the average values of β and d/c between the above two sets of equation (i.e., $\psi/\phi = 1$ and $\psi/\phi = 0$ cases) are used.

North-South Variation of Properties

Spatial variation of E, c, and ϕ is implemented by specifying these parameters as functions of a field variable that is assigned the value of x-coordinate at every node.

From Page No.

Initial Temperature

Initial nodal temperatures are specified using the following information from Reference (1) [cited on p. 2] and the value of y-coordinate at every node.

Y-coordinate at ground surface (top of model): 1000m

Temperature at ground surface: 18.7°C

Depth (m)	Temperature gradient (°/m)
0 – 150	0.02
150 – 400	0.018
400 – 700	0.03
> 700	0.008

Thermal Boundary Conditions

Temperature fixed at initial value at all exterior-boundary nodes, with the initial value calculated as described above.

Mechanical Boundary Conditions

Free-surface condition (zero traction) at the ground surface (top of model) and zero boundary-normal displacement at all other exterior boundaries.

Pages 18-24 entries by GLO 4/17/98

Witnessed & Understood by me, | Date | Invented by | Date | Recorded by

Pages 25-31 entries by GLO 4/23/98

From Page No.

Analysis Procedure

GLO 4/23/98

Analyses consist of sequentially coupled thermal (heat conduction) and ~~mechan~~ thermal (GLO) and mechanical procedures. Temperature calculated in the thermal analysis are used as input for the mechanical analysis.

Thermal Analysis Steps

Step Number	Time at End of Step (yr)	Remarks
1	2×10^{-6}	Dummy step to synchronize thermal and mechanical analyses.
2	150.0	Time-varying volumetric heat source (p. 12) applied to simulate thermal load of emplaced waste.

Mechanical Analysis Steps

Step Number	Time at End of Step (yr)	Remarks
1	1×10^{-6}	Initial static equilibrium under gravitational loading and boundary restraint. No drifts
2	1×10^{-6}	Drifts excavated
3	150	Temperatures from Step 2 of thermal analysis applied in this step
4	151	Drift-support removed.

Witnessed & Understood by me, | Date | Invented by | Date | Recorded by

From Page No. There is no change in nodal temperature between Steps 3 and 4 of the mechanical analyses. So any differences between calculated responses at the end of these steps arises from the removal of rock support.

~~Ta?~~ GWD
4/23/98

Degradation of Rock-Mass Strength:

Two sets of mechanical analyses are conducted as follows (using temperature distributions from one thermal analysis):

Case 1:

Young's modulus and all strength parameters assigned values corresponding to the values of Q on p. 19 with $\sigma_{ci} = 180$ MPa as stated on p. 15.

Case 2:

(i) Young's modulus assigned same value as in Case 1.

(ii) All strength parameters assigned values corresponding to 0.1Q (where Q is given on p. 19) and $\sigma_{ci} = 90$ MPa.

Both Poisson's ratio and thermal expansivity are not modified.

Keeping Young's modulus, Poisson's ratio, and thermal expansivity at their current values (while strength parameters are reduced) is intended to simulate the occurrence of stress buildup (during the first 75 yr or so) prior to onset of rock-mass degradation.

From Page No. Stress buildup is controlled by these elastic parameters.

Degradation of intact rock strength from 180 MPa to 90 MPa (a factor of 0.5) is intended to simulate the difference between the strength of intact rock under rapid loading (laboratory test condition) and sustained loading (in situ condition), as documented in Lajtai and Schmidtke (1986) [Lajtai, E.Z. and R.H. Schmidtke. 1986. Delayed failure in rock loaded in uniaxial compression. Rock Mechanics and Rock Engineering, 19:11-25].

Degradation of Q to 0.1Q arises from an expectation that the ratio Jr/Ja (Jr is joint roughness number and Ja is joint alteration number) in the formula for Q can decrease by one order of magnitude due to ~~???~~ fracture wallrock alteration under exposure to heat and moisture. The formula for Q can be found in rock engineering texts or Barton et al. (1974) [Engineering classification of rock masses for the design of tunnel support. Rock Mechanics, 6:189-236].

Input Files: Thermal Analysis

ht0150.inp	Main ABAQUS input file. Commands in this file cause ABAQUS to read input information from the following auxiliary input files:
allNodes.def	Node definitions
htElems.def	Element definitions
drftHeatSrc.def	Heat Source history definition
iniTemp.def	Initial nodal temperature
bndTemp.def	Fixed boundary temperature

From Page No.

Input Files: Mechanical Analysis: Case 1 Property Set

mCBase.inp   Main ABAQUS input file for Analysis Step 1 and Step 2 (p. 25)

mC150.inp   Main ABAQUS input file for Analysis Step 3 and Step 4 (p. 25).

Auxiliary Input Files (Called from mCBase.inp and mC150.inp)

allNodes.def   Node definitions
mElemCur.def   Element definitions
iniTemp.def   Initial nodal temperature
mdfld1.def   Defines x-coordinate as a field variable used to assign material property values
ElPropCurrent.def   Elastic properties (current values)
FricAngCurrent.def   Friction properties (current values)
CohesionCurrent.def   Cohesion (current values)
ht0150.fil   Results file from thermal analysis

Input Files: Mechanical Analysis Case 2 Property Set

mLBase.inp   Main ABAQUS input file for Analysis Step 1 and Step 2 (p. 25)

mL150.inp   Main ABAQUS input file for Analysis Steps 3 and 4 (p. 25).

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date | Recorded by

From Page No.

Auxiliary Input Files: Called from mLBase.inp and mL150.inp

allNodes.def   Node definitions
mElemLt.def   Element definitions
iniTemp.def   Initial nodal temperature
mdfld1.def   Field variable (x-coordinate) for mech property definitions.
ElPropCurrent.def   Current elastic properties
FricAngLongTerm.def   Degraded frictional properties
CohesionLongTerm.def   Degraded cohesion properties
ht0150.fil   Thermal analysis results file

All of the input files listed on p. 27 - current page are ASCII files, except ht0150.fil, which is a binary file. (generated from the thermal analysis run).

Results Processing

Analysis results are presented in terms of contour plots of the magnitude of inelastic strain. The contours are produced using ABAQUS/POST by invoking the following file

cPlotDefs.jnl   Invoked from within ABAQUS/POST This file causes ABAQUS/POST to read the following files:

4/23/98 GLO

plotElems.def   Lists elements within ± ~~15m~~ 17.5 of repository elevation (drift axis)
cSettings.def   Defines settings for contour plotting
peContours.def   Plots inelastic strain contours.

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date | Recorded by

GVO 4/23/98

4/23/98 GVO

Pages 25-31 by entries GVO 4/23/98

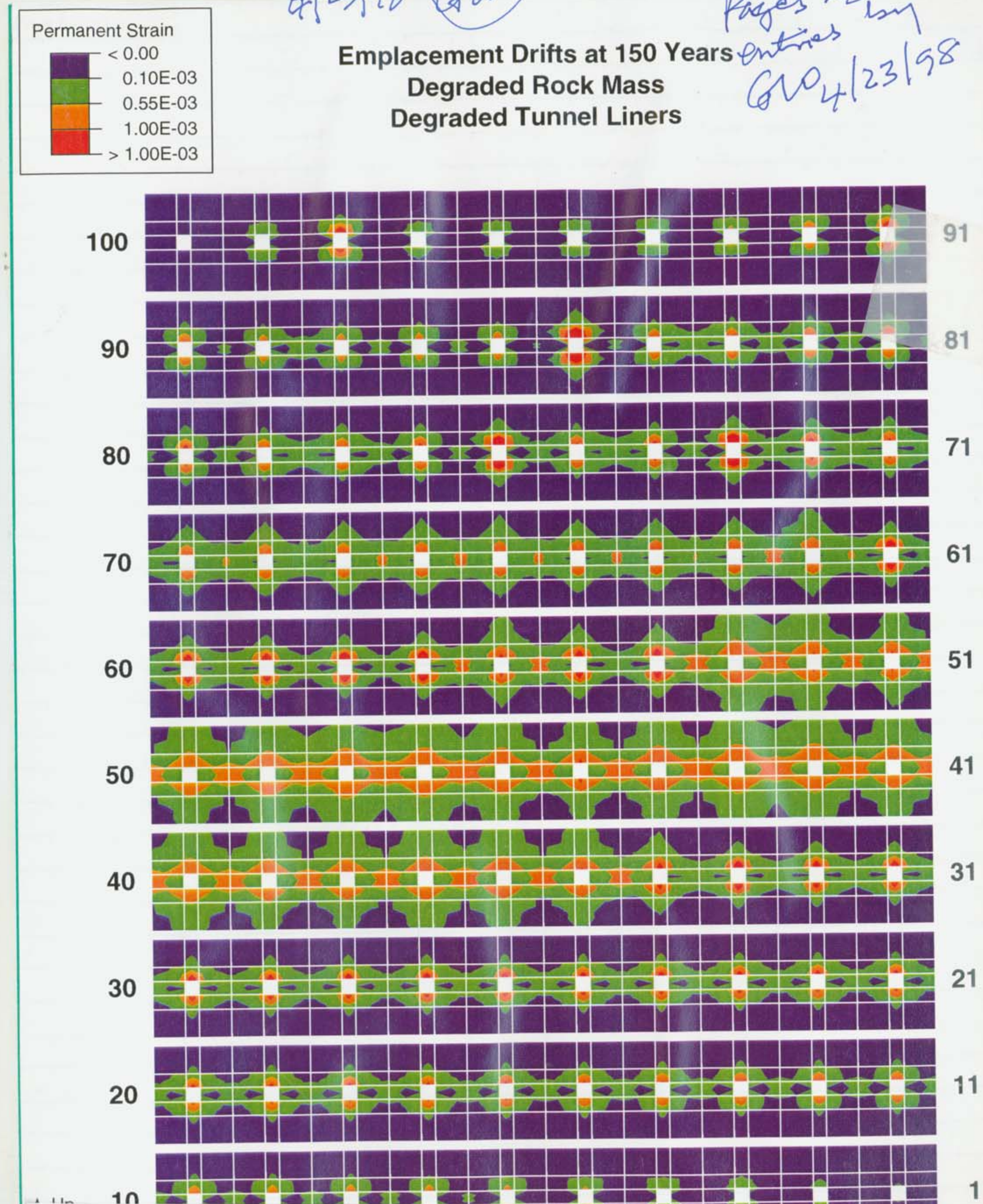


Figure Distribution of permanent (i.e., inelastic) strain intensity for the analysis case with current values of Young's modulus, degraded values of friction angle, cohesion, and intact-rock strength, and nondegraded concrete lining. Result is presented in ten sections, each 280 m long (in north-south direction) by 35 m high (vertical dimension). Ten white squares within each section represent emplacement drifts, and the numbers for the end drifts of each section are shown. Drift #100 is at the south end of the figure

Pages 32–33 entries by GIO 10/12/98

Project No. ____ Book No. ____ TITLE October 12 1998

From Page No. ____

Additional Analysis To Develop Rock-Mass Quality Contours (Similar to the inelastic strain contours on p. 30–31).

Approach: Submit distribution of rock-mass quality Q to ABAQUS as a user-defined field variable. Then contour the variable in ABAQUS/POST.

Input Files

qField.inp — This file reads model definitions from the following auxiliary files:

allNodes.def — Node definitions (same as mentioned on pp. 27 and 29).

elements.def — Element definitions in this file are the same as in mElemLt.def (p. 29), except that the material property definitions in mElemLt.def were replaced with a dummy elastic property set. Material properties play no important role in this analysis (but are required by ABAQUS).

qAsField.def — This file defines Q at nodal points following ABAQUS format for user-defined field variables. It was prepared using C++ code RmqAsField (documented on p. 33).

GIO 10/12/98

To Page No. ____

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

TITLE ____ Project No. ____ Book No. ____

From Page No. ____

```
RmqAsField.cpp                                                          Page 1

/*****************************************************************************
RmqAsField

This code defines values of rock-mass quality Q at nodal points of a finite
element mesh using nodal-point x and y coordinates from ABAQUS and Q-vs-x data
prepared using another code. The output follows ABAQUS input format for
user-defined field variables.

Information is read from two files.

1. xyFile, which gives nodal-point x and y coordinates.

2. matPFile, which gives values of Q, Young's modulus, friction angle,
   and cohesion as functions of x coordinate.

For each input line read from xyFile, the x coordinate is interpolated
in the Q-vs-x table to obtain the value of Q.

Author:                  G. I. Ofoegbu
Date:                    October 12 1998
System:                  C++ Compiler
                         (Code developed using Borland C++Builder 3)
*****************************************************************************/

#include <stdio.h>
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>
#include <strstrea.h>
#include <conio.h>
#include <stdlib.h>

#pragma hdrstop
#include <condefs.h>

char* outFile = "qAsField.def";
char* xyFile = "d:\\ThermMech\\RepoScale\\allNd.xy";
char* matPFile = "d:\\ThermMech\\RepoScale\\mechProp2.txt";

char* messageBuffer;
void DumpAndQuit();
void DeleteVectors(float** x, float** q);

int ERROREND = 1;
int NORMALEND = 0;
int ReadMatProps(float** x, float** rmQ, int* num);
float LinearInterp(const float* t, const float* f, float atTemp, int n);

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

// 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);
  }

// Setup vectors for mechanical property vs x-coordinate tables and read the
// table data from file

  float *xCoord = 0;
  float *rmQ = 0;
  int numData;
  if (ReadMatProps(&xCoord,&rmQ,&numData) == ERROREND){
```

GIO 10/12/98

```
RmqAsField.cpp                                                          Page 2

    DeleteVectors(&xCoord,&rmQ);
    DumpAndQuit();
  }
  cout << "Found " << numData << " lines in material property file\n";

  char buf[100];
  size_t nBuf = sizeof(buf);
  ifstream Fxy(xyFile);
  if (!Fxy){
    sprintf(messageBuffer,"Unable to open file %s",xyFile);
    DeleteVectors(&xCoord,&rmQ);
    DumpAndQuit();
  }
  ofstream Fout(outFile);
  if (!Fout){
    sprintf(messageBuffer,"Unable to open file %s",outFile);
    DeleteVectors(&xCoord,&rmQ);
    DumpAndQuit();
  }
  Fout << "***\n"
       << "*** Rock mass quality Q as field variable 1\n"
       << "***\n"
       << "*INITIAL CONDITION, TYPE=FIELD, VAR=1"
       << endl;

  int node;
  float x,y;
  float q;

  while (Fxy){
    Fxy.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> node >> x >> y;
    if (inpLine.good()){
      q = LinearInterp(xCoord,rmQ,x,numData);
      Fout << setiosflags(ios::fixed | ios::showpoint)
           << setw(10) << node << ","
           << setprecision(3)
           << setw(10) << q
           << endl;
    }
  }

  DeleteVectors(&xCoord,&rmQ);
  delete[] messageBuffer;
  cout << "\nDone ... Press any key to end: ";
  getch();
  return 0;
}

int ReadMatProps(float** xCoord, float** rmQ, int* numData)
{
  ifstream Fin(matPFile);
  if (!Fin){
    sprintf(messageBuffer,"Unable to open file %s",matPFile);
    return(ERROREND);
  }

  char buf[100];
  size_t nBuf = sizeof(buf);
  float x1,q,yml,phil,cpl;
  int nn = 0;

  while (Fin){
    Fin.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> x1 >> q >> yml >> phil >> cpl;
    if (inpLine.good())
      ++nn;
  }
```

GIO 10/12/98

GIO 10/12/98

To Page No. ____

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

GAO 10/12/98

```
RmqAsField.cpp                                                   Page 3

   if (!nn){
     sprintf(messageBuffer,"No valid input in file %s",matPFile);
     return(ERROREND);
   }

   *xCoord = new float[nn];
   *rmQ = new float[nn];
   if (!(*xCoord) || !(*rmQ)){
     sprintf(messageBuffer,"Memory allocation failure on matP vectors");
     return(ERROREND);
   }

   Fin.seekg(0,ios::beg);    // Rewind input file
   int i = 0;
   while (Fin){
     Fin.getline(buf,100);
     istrstream inpLine(buf,nBuf);
     inpLine >> x1 >> q >> ym1 >> phi1 >> cp1;
     if (inpLine.good()){
       (*xCoord)[i] = x1;
       (*rmQ)[i] = q;
       if ((++i) >= nn)
         break;
     }
   }

   *numData = nn;
   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]);

  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 DeleteVectors(float** x, float** q)
{
  delete[] *x;
  delete[] *q;
}

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


RmqAsField.cpp                                                   Page 4

  exit(1);
}
```

Results of the dummy analysis (to display Q as a user-defined field) are processed following the procedure described on p. 29. The input file invoked from within ABAQUS/POST is

qPlotDefs.jnl

which causes POST to read the following files:

qPlotElem.def
qPlotSettings.def
qContours.def

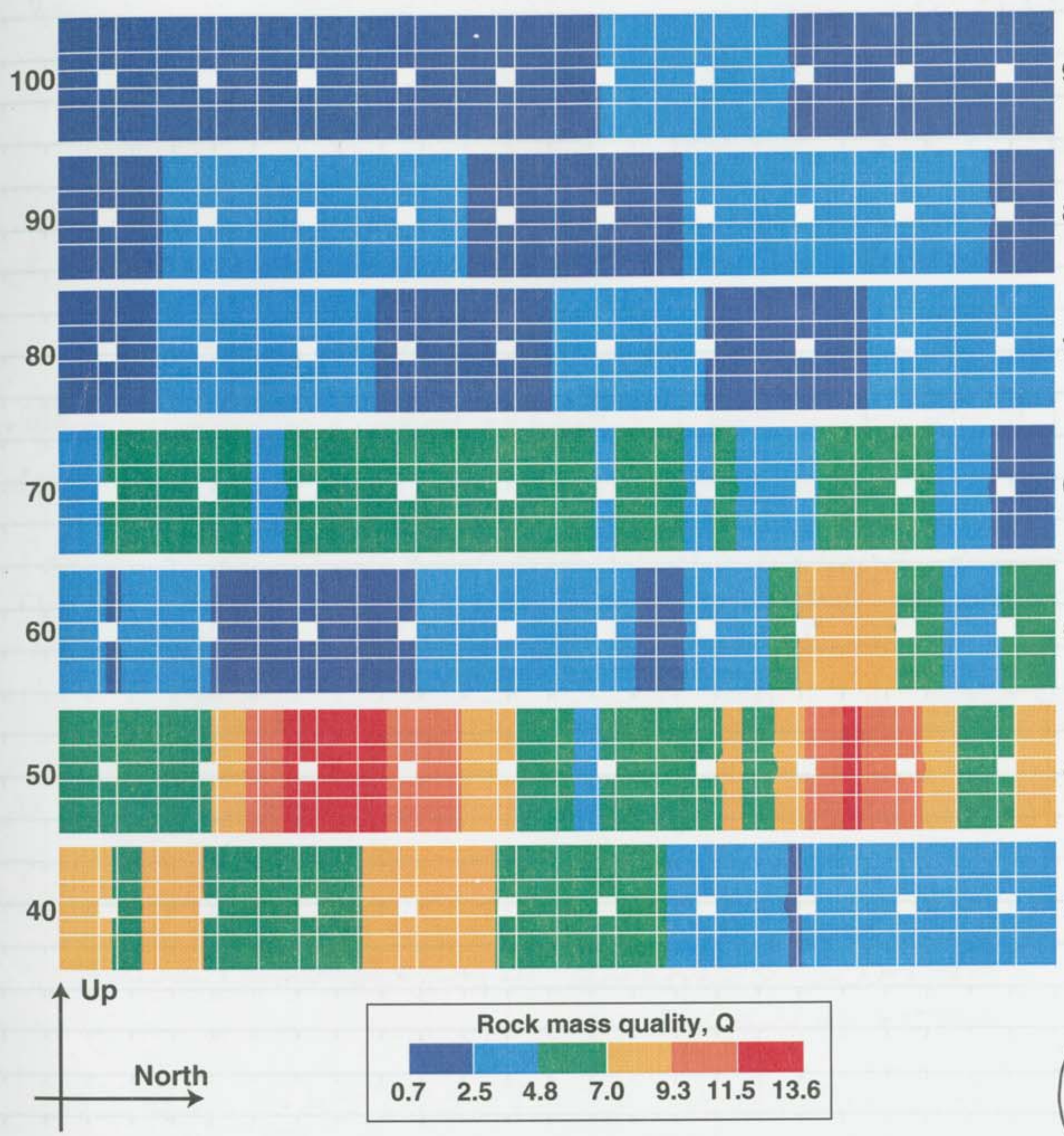
GAO 10/12/98

The ABAQUS/POST result is a file that is processed using ABAQUS/PLOT to obtain 7 postscript files. These postscript files were later assembled using Adobe Illustrator to obtain the file (figure)

qFields.ai

that is shown on p. 35.

GAO 10/12/98



GAO 10/12/98

Distribution of rock-mass quality, Q, within 17.5 m above and below the drift axis, presented following the format described on pp. 30 and 31. Values of Q from Drifts 1–30 (not shown) are the same as for Drifts 31–32.

From Page No.

Analysis to Develop Scatter Plots of Γ^N (Magnitude of Inelastic Strain) Vs Mechanical Properties

The following plots are desired:

Γ^N vs Q

Γ^N vs E (E = Youngs modulus)

Γ^N vs ϕ (ϕ = friction angle)

GIO 10/12/98 Γ^N vs c (c = cohesion parameter)

A table of x and y coordinates and Γ^N values at element integration points was obtained from ABAQUS for ~~all~~ GIO solid elements in the following zones: 10/12/98

Zone (1), (2) and (3) [p. 6]

This table was prepared in three steps as follows:

1. Run ABAQUS POST-OUTPUT job to obtain element integration-point coordinates.

2. Run ABAQUS POST-OUTPUT job to obtain integration-point Γ^N values.

3. Combine tables from both analyses to obtain one table, using C++ code documented on pp. 37-38.

GIO 10/12/98

~~The results were plotted using Kaleidograph, and the postscript files produced by Kaleidograph were assembled using Adobe Illustrator~~

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

From Page No.

GIO 10/12/98

```
PstrnMag.cpp                                                    Page 1

/*****************************************************************************
PstrnMag

This code combines two ABAQUS output tables in two separate files into one.
The first file (xyFile) contains element integration-point coordinates
with the element and integration point numbers.
The second file (strnFile) gives values of inelastic strain intensity
at the integration points along with the element and integration point
numbers. Because strnFile does not include integration points that have
zero inelastic strain intensity, not every point in xyFile is included in
strnFile; conversely, every point in strnFile is also in xyFile. The combined
table will include every point, assigning zero values to points not found
in strnFile.

Author:                 G. I. Ofoegbu
Date:                   October 8 1998
System:                 C++ Compiler
                        (Code developed using Borland C++Builder 3)
*****************************************************************************/

#include <stdio.h>
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>
#include <strstrea.h>
#include <conio.h>
#include <stdlib.h>

#pragma hdrstop
#include <condefs.h>


struct DPoint{
  int element;
  int ipt;
  float x;
  float y;
  float pemag;
  struct DPoint* next;
  DPoint(int elNum, int ipoint, float xValue, float yValue){
    element = elNum;
    ipt = ipoint;
    x = xValue;
    y = yValue;
    pemag = 0.0;
    next = 0;
  }
};
char* outFile = "s0150XyStrn.dat";
char* strnFile = "s0150Pemag.dat";
char* xyFile = "intpCoord.dat";

char* messageBuffer;
int ERROREND = 1;
int NORMALEND = 0;

void StorePoint(DPoint* current, DPoint** first, DPoint** last);
bool IsGreater(DPoint* current, DPoint* old);
int InsertStrnValue(DPoint* first, int element, int ipt, float strn);
int PrintPoints(DPoint* first);
void DeletePoints(DPoint** first);
void DumpAndQuit();

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

// Allocate memory for possible error messages
```

GIO 10/12/98

GIO 10/12/98

```
PstrnMag.cpp                                                    Page 2

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

  char buf[100];
  size_t nBuf = sizeof(buf);
  int element,ipt;
  float x,y,strn;
  DPoint *first = 0;
  DPoint *last = 0;

  ifstream Fxy(xyFile);
  if (!Fxy){
    sprintf(messageBuffer,"Unable to open file %s",xyFile);
    DumpAndQuit();
  }

  while (Fxy){
    Fxy.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> element >> ipt >> x >> y;
    if (inpLine.good()){
      DPoint* current = new DPoint(element,ipt,x,y);
      if (!current){
        sprintf(messageBuffer,"Memory error on el %d ipt %d",element,ipt);
        DeletePoints(&first);
        DumpAndQuit();
      }
      StorePoint(current,&first,&last);
    }
  }
  Fxy.close();

  ifstream Fstrn(strnFile);
  if (!Fstrn){
    sprintf(messageBuffer,"Unable to open file %s",strnFile);
    DeletePoints(&first);
    DumpAndQuit();
  }
  while (Fstrn){
    Fstrn.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> element >> ipt >> strn;
    if (inpLine.good())
      if (InsertStrnValue(first,element,ipt,strn) == ERROREND){
        DeletePoints(&first);
        DumpAndQuit();
      }
  }
  Fstrn.close();

  if (PrintPoints(first) == ERROREND){
    DeletePoints(&first);
    DumpAndQuit();
  }
  DeletePoints(&first);
  delete[] messageBuffer;
  cout << "\nDone ... Press any key to end: ";
  getch();
  return 0;
}

void StorePoint(DPoint *current, DPoint **first, DPoint **last)
{
  DPoint *old,*p;
```

GIO 10/12/98

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

From Page No.

```
PstrnMag.cpp                                                    Page 3

  p = *first;

  if (!*last){           /* current is the very first item in the list */
    current->next = 0;
    *last = current;
    *first = current;
    return;
  }

  old = 0;
  while (p){
    if (IsGreater(current,p)){
      old = p;
      p = p->next;
    }
    else{
      if (old) {       /* insert current at this point in the list */
        old->next = current;
        current->next = p;
        return;
      }
      current->next = p;     /* current becomes new first element */
      *first = current;
      return;
    }
  }

  (*last)->next = current;  /* current becomes new last element */
  current->next = NULL;
  *last = current;
}

bool IsGreater(DPoint* current, DPoint* old)
{
  if ((current->element) < (old->element))
    return (false);

  if ((current->element) > (old->element))
    return (true);

// current element same as old element; check integration points

  if ((current->ipt) > (old->ipt))
    return (true);

  return (false);
}

int InsertStrnValue(DPoint* first, int element, int ipt, float strn)
{
  if (!first){
    sprintf(messageBuffer,"Empty list passed to InsertStrnValue");
    return (ERROREND);
  }
  while (first){
    if (((first->element) == element) && ((first->ipt) == ipt)){
      first->pemag = strn;
      return (NORMALEND);
    }
    first = first->next;
  }

  sprintf(messageBuffer,"Elem %d ipt %d not found in array",element,ipt);
  return (ERROREND);
}

int PrintPoints(DPoint* first)
{
  if (!first){
    sprintf(messageBuffer,"Empty list passed to PrintPoints");
```

```
PstrnMag.cpp                                                    Page 4
    return (ERROREND);
  }
  ofstream Fout(outFile);
  if (!Fout){
    sprintf(messageBuffer,"Unable to open file %s",outFile);
    return (ERROREND);
  }
  Fout << setw(10) << "Element"
       << setw(5) << "IPT"
       << setw(15) << "xCoord (m)"
       << setw(15) << "yCoord (m)"
       << setw(15) << "Microstrain"
       << endl;
  while (first){
    Fout << setw(10) << first->element
         << setw(5) << first->ipt
         << setiosflags(ios::fixed | ios::showpoint)
         << setprecision(1)
         << setw(15) << first->x
         << setw(15) << first->y
         << resetiosflags(ios::fixed)
         << setiosflags(ios::scientific)
         << setprecision(4)
         << setw(15) << (1000.*(first->pemag))
         << resetiosflags(ios::scientific)
         << endl;
    first = first->next;
  }
  return (NORMALEND);
}

void DeletePoints(DPoint** first)
{
  DPoint *old;
  DPoint *p = *first;

  while (p){
    old = p;
    p = p->next;
    delete old;
  }
}

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

10/12/98

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No. Having obtained the X-Y-Γ^N table the X values were interpolated into tables of Q-vs-X, E-vs-X, Ø-vs-X, and C-vs-X (the material property tables) to obtain values of Q, E, Ø, and C at the element integration points, using code StrnVsProps (pp. 40–41).

The material property tables were developed from the ESF rock-mass quality data using procedure described on pp. 18–22 and implemented through code mechProps (pp. 42–44).

The digitized ESF Q data (file esfRmQ.txt) used to develop the material property tables (with code mechProps) are reproduced on p. 45.

The resulting scatter plots of Γ^N-vs-Q, Γ^N-vs-E, Γ^N-vs-Ø, and Γ^N-vs-C (based on results at element integration points) are presented on pp. 46–49. Each mechanical parameter is plotted as a scaled value, defined as

$$\text{ScaledValue} = \frac{(\text{ActualValue} - \text{MinimumValue})}{(\text{MaximumValue} - \text{MinimumValue})}$$

Values of minimum and maximum values are as follows (obtained as by-product of code StrnVsProps on pp. 40–41):

mechPMaxMin.txt

	PropertyName	MinimumValue	MaximumValue
Current →	NGI Q	0.729	13.573
Based on degraded Q and σ_{ci} values.	Ymod GPa	2.432	11.060
	FricAngle	22.66	30.16
	Cohesion MPa	1.667	3.514

10/12/98

10/12/98

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

Project No.
Book No.
TITLE
From Page No.

```
StrnVsProps.cpp                                                    Page 1

/****************************************************************************
StrnVsProps

This code prepares a table of x and y coordinates, mechanical parameters
(NGI Q, Young's modulus, friction angle, and cohesion), and inelastic strain
intensity using results from a finite element model. The mechanical-property
values for the output table are scaled as follows:

   ScaledValue = (ActualValue - MinimumValue)/(MaximumValue - MinimumValue)

As a result, every mechanical-property value in the output table lies between
0.0 and 1.0. Information is read from two files.

1. strnFile, which gives x and y coordinates and inelastic strain intensity
   at element integration points.

2. matPFile, which gives values of NGI Q, Young's modulus, friction angle,
   and cohesion as functions of x coordinate.

For each input line read from strnFile, the x coordinate is
interpolated in the material property tables to obtain values of NGI Q,
Young's modulus, friction angle, and cohesion for the integration point.

Author:                    G. I. Ofoegbu
Date:                      October 8 1998
System:                    C++ Compiler
                           (Code developed using Borland C++Builder 3)
****************************************************************************/

#include <stdio.h>
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>
#include <strstrea.h>
#include <conio.h>
#include <stdlib.h>

#pragma hdrstop
#include <condefs.h>

char* outFile = "s0150ScldProp.dat";
char* strnFile = "s0150XyStrn.dat";
char* matPFile = "d:\\ThermMech\\RepoScale\\mechProp2.txt";
char* maxminFile = "d:\\ThermMech\\RepoScale\\mechPMaxMin.txt";

char* messageBuffer;
void DumpAndQuit();
void DeleteVectors(float** x, float** q, float** ym, float** phi, float** cp);

int ERROREND = 1;
int NORMALEND = 0;
int ReadMatProps(float** x, float** rmQ, float** ym, float** phi,
  float** cp, int* num);
float LinearInterp(const float* t, const float* f, float atTemp, int n);

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

// 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);
  }
```

GIO 10/12/98

```
StrnVsProps.cpp                                                    Page 2

// Setup vectors for mechanical property vs x-coordinate tables and read the
// table data from file

  float *xCoord = 0;
  float *rmQ = 0;
  float *yMod = 0;
  float *fric = 0;
  float *cpar = 0;
  int numData;
  if (ReadMatProps(&xCoord,&rmQ,&yMod,&fric,&cpar,&numData) == ERROREND){
    DeleteVectors(&xCoord,&rmQ,&yMod,&fric,&cpar);
    DumpAndQuit();
  }
  cout << "Found " << numData << " lines in material property file\n";

  char buf[100];
  size_t nBuf = sizeof(buf);
  ifstream Fstrn(strnFile);
  if (!Fstrn){
    sprintf(messageBuffer,"Unable to open file %s",strnFile);
    DeleteVectors(&xCoord,&rmQ,&yMod,&fric,&cpar);
    DumpAndQuit();
  }
  ofstream Fout(outFile);
  if (!Fout){
    sprintf(messageBuffer,"Unable to open file %s",outFile);
    DeleteVectors(&xCoord,&rmQ,&yMod,&fric,&cpar);
    DumpAndQuit();
  }
  Fout << setw(10) << "X-crd m"
       << setw(10) << "Y-crd m"
       << setw(10) << "NGI Q"
       << setw(10) << "YMod"
       << setw(10) << "fricAng"
       << setw(15) << "Cohesion"
       << setw(15) << "Microstrain"
       << endl;

  int element,iPoint;
  float x,y;
  float strn;
  float q,ym,phi,cp;

  while (Fstrn){
    Fstrn.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> element >> iPoint >> x >> y >> strn;
//    if (inpLine.good() && (strn > 0.0)){
    if (inpLine.good()){
      q = LinearInterp(xCoord,rmQ,x,numData);
      ym = LinearInterp(xCoord,yMod,x,numData);
      phi = LinearInterp(xCoord,fric,x,numData);
      cp = LinearInterp(xCoord,cpar,x,numData);
      Fout << setiosflags(ios::fixed | ios::showpoint)
           << setprecision(3)
           << setw(10) << x
           << setw(10) << y
//           << setprecision(3)
           << setw(10) << q
           << setw(10) << ym   // << setw(10) << (ym/1000.)
           << setw(10) << phi
           << setw(15) << cp
           << resetiosflags(ios::fixed)
           << setiosflags(ios::scientific)
           << setprecision(3)
           << setw(15) << strn
           << resetiosflags(ios::scientific)
           << endl;
```

GIO 10/12/98

GIO 10/12/98

GIO 10/12/98

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

TITLE
From Page No.
Project No.
Book No.

```
StrnVsProps.cpp                                                    Page 3

  DeleteVectors(&xCoord,&rmQ,&yMod,&fric,&cpar);
  delete[] messageBuffer;
  cout << "\nDone ... Press any key to end: ";
  getch();
  return 0;
}

int ReadMatProps(float** xCoord, float** rmQ, float** yMod, float** fric,
  float** cpar, int* numData)
{
  ifstream Fin(matPFile);
  if (!Fin){
    sprintf(messageBuffer,"Unable to open file %s",matPFile);
    return(ERROREND);
  }
  ofstream Fout(maxminFile);
  if (!Fout){
    sprintf(messageBuffer,"Unable to open file %s",maxminFile);
    return(ERROREND);
  }

  char buf[100];
  size_t nBuf = sizeof(buf);
  float x1,q,ym1,phi1,cp1;
  float qMax = 0.01;
  float ymMax = 0.01;
  float phiMax = 0.01;
  float cpMax = 0.01;
  float qMin = 1.0E10;
  float ymMin = 1.0E10;
  float phiMin = 1.0E10;
  float cpMin = 1.0E10;
  int nn = 0;

  while (Fin){
    Fin.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> x1 >> q >> ym1 >> phi1 >> cp1;
    if (inpLine.good()){
      if (q > qMax) qMax = q;
      if (q < qMin) qMin = q;
      if (ym1 > ymMax) ymMax = ym1;
      if (ym1 < ymMin) ymMin = ym1;
      if (phi1 > phiMax) phiMax = phi1;
      if (phi1 < phiMin) phiMin = phi1;
      if (cp1 > cpMax) cpMax = cp1;
      if (cp1 < cpMin) cpMin = cp1;
      ++nn;
    }
  }

  if (!nn){
    sprintf(messageBuffer,"No valid input in file %s",matPFile);
    return(ERROREND);
  }

  *xCoord = new float[nn];
  *rmQ = new float[nn];
  *yMod = new float[nn];
  *fric = new float[nn];
  *cpar = new float[nn];
  if (!(*xCoord) || !(*rmQ) || !(*yMod) || !(*fric) || !(*cpar)){
    sprintf(messageBuffer,"Memory allocation failure on matP vectors");
    return(ERROREND);
  }

  Fin.seekg(0,ios::beg);    // Rewind input file
  int i = 0;
  while (Fin){
```

GIO 10/12/98

GIO

```
StrnVsProps.cpp                                                    Page 4

    Fin.getline(buf,100);
    istrstream inpLine(buf,nBuf);
    inpLine >> x1 >> q >> ym1 >> phi1 >> cp1;
    if (inpLine.good()){
      (*xCoord)[i] = x1;
      (*rmQ)[i] = (q - qMin)/(qMax - qMin);
      (*yMod)[i] = (ym1 - ymMin)/(ymMax - ymMin);
      (*fric)[i] = (phi1 - phiMin)/(phiMax - phiMin);
      (*cpar)[i] = (cp1 - cpMin)/(cpMax - cpMin);
      if ((++i) >= nn)
        break;
    }
  }

  *numData = nn;

  Fout << setw(15) << "PropertyName"
       << setw(15) << "MinimumValue"
       << setw(15) << "MaximumValue"
       << endl;
  Fout << setw(15) << "NGI Q"
       << setiosflags(ios::fixed | ios::showpoint)
       << setprecision(3)
       << setw(15) << qMin
       << setw(15) << qMax
       << endl;
  Fout << setw(15) << "Ymod GPa"
       << setiosflags(ios::fixed | ios::showpoint)
       << setprecision(3)
       << setw(15) << (ymMin/1000.)
       << setw(15) << (ymMax/1000.)
       << endl;
  Fout << setw(15) << "FricAngle"
       << setiosflags(ios::fixed | ios::showpoint)
       << setprecision(2)
       << setw(15) << phiMin
       << setw(15) << phiMax
       << endl;
  Fout << setw(15) << "Cohesion MPa"
       << setiosflags(ios::fixed | ios::showpoint)
       << setprecision(3)
       << setw(15) << cpMin
       << setw(15) << cpMax
       << endl;

  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]);

  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])
      );

StrnVsProps.cpp                                                    Page 5

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

void DeleteVectors(float** x, float** q, float** ym, float** phi, float** cp)
{
  delete[] *x;
  delete[] *q;
  delete[] *ym;
  delete[] *phi;
  delete[] *cp;
}

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

GIO 10/12/98

GIO 10/12/98

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

Project No.______ Book No.______ TITLE ______

From Page No.____

D:\ThermMech\RepoScale\mechProps.cpp Page 1

```
/*****************************************************************************
mechProps

This code implements procedure described on pp. 18-23 of Scientific Notebook
Number 263 to develop tables of mechanical properties as functions of
horizontal coordinate x (for finite element model described in the same
notebook). The mechanical parameters evaluated are:

    1. Young's modulus
    2. Mohr-Coulomb friction angle
    3. Mohr-Coulomb cohesion parameter
    4. Drucker-Prager friction angle
    5. Drucker-Prager cohesion parameter

Parameters 1, 4, and 5 are output following ABAQUS 5.6 input formats. Also,
the rock-mass quality Q and parameters 1, 2, and 3 are output in a format
suitable for plotting. The input data (ESF Q values) is read from the file

    inFileName

which contains digitized table of values of ESF Q vs ESF north-to-south
station distance s. The latter (s) is converted to x following procedure
on p. 18 of Notebook 263.

Author:                          G. I. Ofoegbu
Date:                            April 1998
System:                          C++ compiler
                                 (Code developed using Borland C++ 5.0)
*****************************************************************************/

#include <stdio.h>
#include <iomanip.h>
#include <fstream.h>
#include <strstrea.h>
#include <string.h>
#include <math.h>

struct DPoint{
  float x;
  float rmQ;
  float yMod;
  float cohesion;
  float fricAngle;
  struct DPoint* next;
  DPoint(float xValue, float qValue){
    x = xValue;
    rmQ = qValue;
    next = 0;
  }
};

float RockMassModulus(float rmQ);
void StorePoint(DPoint* current, DPoint** first, DPoint** last);
void PrintPoints(DPoint* first, char* elasticFile, char* dcBetaFile,
  char* dcDParFile);
void MakePlotFile(DPoint* first, char* fileName);
void DeletePoints(DPoint** first);
float LinearInterp(const float* t, const float* f, float atTemp, int n);
void MohrCoulombFromQ(float rmQ, float ucsIntact, float& fricAngle,
  float& cohesion);
void DruckerPragerFromMohrCoulomb(float fricAngle, float cohesion,
  float& dcSlopeAngle, float& dcIntercept);

main()
{
  char buf[81];
  char* inFileName = "esfRmQ.txt";
  char* elasticFile = "ElPropLongTerm.def";
  char* dcBetaFile = "FricAngLongTerm.def";
  char* dcDParFile = "CohesionLongTerm.def";
```

D:\ThermMech\RepoScale\mechProps.cpp Page 2

```
  char* plotFileName = "mechProp2.txt";

  float r,y;
  float qDegradeFactor = 0.1;
  float ucsDegradeFactor = 0.5;
  float ucsIntact = 180.0;

  DPoint *first = 0;
  DPoint *last = 0;

  ifstream Fin(inFileName);
  if (!Fin){
    cerr << "Unable to open file " << inFileName << endl;
    cout << "Press return to end ";
    int nn = getchar();
    return (nn);
  }

  while (Fin){
    Fin.getline(buf,80);
    istrstream inpLine(buf,sizeof(buf));
    inpLine >> r >> y;
    if (inpLine.good()){
      float s = 1000. * r;
      if ((s > 2800.) && (s < 5001.)){
        float rmQ = pow(10.0, y);
        float x = 4708. - s;
        DPoint* current = new DPoint(x,rmQ);
        if (!current){
          cerr << "Memory allocation error " << endl;
          cout << "Press return to end ";
          int nn = getchar();
          return (nn);
        }
        current->yMod = RockMassModulus(rmQ*qDegradeFactor);
        MohrCoulombFromQ(rmQ*qDegradeFactor, ucsIntact*ucsDegradeFactor,
          current->fricAngle, current->cohesion);
        StorePoint(current,&first,&last);
      }
    }
  }
  PrintPoints(first, elasticFile, dcBetaFile, dcDParFile);
  MakePlotFile(first, plotFileName);
  DeletePoints(&first);
}

void StorePoint(DPoint *current, DPoint **first, DPoint **last)
{
  DPoint *old,*p;

  p = *first;

  if (!*last){           /* current is the very first item in the list */
    current->next = NULL;
    *last = current;
    *first = current;
    return;
  }

  old = NULL;
  while (p){
    if ((current->x) > (p->x)){
      old = p;
      p = p->next;
    }
    else{
      if (old) {       /* insert current at this point in the list */
        old->next = current;
        current->next = p;
        return;
```

GW GW GW GW

To Page No.____

Witnessed & Understood by me, | Date | Invented by | Date | Recorded by

TITLE ______ Project No.______ Book No.______

From Page No.____

D:\ThermMech\RepoScale\mechProps.cpp Page 3

```
      }
      current->next = p;    /* current becomes new first element */
      *first = current;
      return;
    }
  }

  (*last)->next = current;  /* current becomes new last element */
  current->next = NULL;
  *last = current;
}

void PrintPoints(DPoint* first, char* elasticFile, char* dcBetaFile,
  char* dcDParFile)
{

  if (!first){
    cerr << "Empty list: Nothing to print " << endl;
    cout << "Press return to end ";
    int nn = getchar();
    cout << nn << endl;
    return;
  }

  ofstream Fe(elasticFile);
  if (!Fe){
    cerr << "Unable to open file " << elasticFile << endl;
    cout << "Press return to end ";
    int nn = getchar();
    cout << nn << endl;
    return;
  }

  ofstream Fbeta(dcBetaFile);
  if (!Fbeta){
    cerr << "Unable to open file " << dcBetaFile << endl;
    cout << "Press return to end ";
    int nn = getchar();
    cout << nn << endl;
    return;
  }

  ofstream Fd(dcDParFile);
  if (!Fd){
    cerr << "Unable to open file " << dcDParFile << endl;
    cout << "Press return to end ";
    int nn = getchar();
    cout << nn << endl;
    return;
  }

  Fe << "*ELASTIC, TYPE=ISO, DEPENDENCIES=1\n";
  Fbeta << "*DRUCKER PRAGER, DEPENDENCIES=1\n";
  Fd << "*DRUCKER PRAGER HARDENING, TYPE=SHEAR, DEPEND=1\n";
  float temperature = 0.0;
  float pRatio = 0.21;
  float kParam = 0.78;
  float dcSlopeAngle;
  float dcIntercept;
  while (first){
    Fe << setiosflags(ios::scientific) << setprecision(2)
       << setw(12) << (first->yMod) << ","
       << setiosflags(ios::fixed) << setprecision(2)
       << setw(8) << pRatio << ","
       << setw(8) << temperature << ","
       << setw(10) << (first->x)
       << endl;

    DruckerPragerFromMohrCoulomb(first->fricAngle,first->cohesion,
      dcSlopeAngle,dcIntercept);
```

D:\ThermMech\RepoScale\mechProps.cpp Page 4

```
    float dilationAngle = 0.5 * dcSlopeAngle;
    float shearStrain = 0.0;
    Fbeta << setiosflags(ios::fixed) << setprecision(2)
       << setw(10) << dcSlopeAngle << ","
       << setw(10) << kParam << ","
       << setw(10) << dilationAngle << ","
       << setw(5) << setprecision(1) << temperature << ","
       << setw(10) << setprecision(2) << (first->x)
       << endl;

    Fd << setiosflags(ios::fixed) << setprecision(2)
       << setw(10) << dcIntercept << ","
       << setw(5) << setprecision(1) << shearStrain << ","
       << setw(5) << temperature << ","
       << setw(10) << setprecision(2) << (first->x)
       << endl;

    first = first->next;
  }
}

void MakePlotFile(DPoint* first, char* fileName)
{
  if (!first){
    cerr << "Empty list: Nothing to print " << endl;
    cout << "Press return to end ";
    int nn = getchar();
    cout << nn << endl;
    return;
  }

  ofstream Fout(fileName);
  if (!Fout){
    cerr << "Unable to open file " << fileName << endl;
    cout << "Press return to end ";
    int nn = getchar();
    cout << nn << endl;
    return;
  }
  Fout << setw(15) << "X-coord (m)"
       << setw(15) << "NGI Q"
       << setw(15) << "YModulus (MPa)"
       << setw(15) << "fricAng (MPa)"
       << setw(15) << "Cohesion (MPa)"
       << endl;
  while (first){
    Fout << setiosflags(ios::fixed) << setprecision(2)
         << setw(15) << (first->x)
         << setprecision(4)
         << setw(15) << (first->rmQ)
         << setiosflags(ios::scientific)
         << setw(15) << (first->yMod)
         << setiosflags(ios::fixed) << setprecision(2)
         << setw(15) << (first->fricAngle)
         << setw(15) << setprecision(3) << (first->cohesion)
         << endl;
    first = first->next;
  }
}

float RockMassModulus(float rmQ)
{
  float gPa2MPa = 1000.;
  float rMR = 9.0*log(rmQ) + 49.0;
  float yModGPa = pow(10., ((rMR - 10.)/40.));
  return (gPa2MPa * yModGPa);
}

void DeletePoints(DPoint** first)
{
```

GW GW GW GW GW

To Page No.____

Witnessed & Understood by me, | Date | Invented by | Date | Recorded by

From Page No.

D:\ThermMech\RepoScale\mechProps.cpp

Page 5

```
  DPoint *old;
  DPoint *p = *first;

  while (p){
    old = p;
    p = p->next;
    delete old;
  }
}

// Function MohrCoulombFromQ estimates values of rock-mass Mohr-Coulomb
// parameters (fricAngle and cohesion) from NGI Q index using empirical
// relationships

void MohrCoulombFromQ(float rmQ, float ucsIntact, float& fricAngle,
  float& cohesion)
{
  float gsiArray[] = {1.00,10.0,20.0,30.0,40.0,50.0,60.0,70.0,80.0,90.0};
  float phiArray[] = {10.0,18.3,22.5,26.2,28.0,31.2,33.2,36.2,37.5,38.5};
  float cArray[] = {0.006,0.013,0.018,0.03,0.035,0.041,0.054,0.07,0.1,0.15};

  float gsi = 9.0 * log(rmQ) + 44.0;
  fricAngle = LinearInterp(&gsiArray[0],&phiArray[0],gsi,10);
  cohesion = ucsIntact * LinearInterp(&gsiArray[0],&cArray[0],gsi,10);
}

//
// Function DruckerPragerFromMohrCoulomb calculates Drucker-Prager parameters
// (slopeAngle and intercept) from Mohr-Coulomb parameters (fricAngle and
// cohesion) using table interpolation (for plane strain) from ABAQUS manual,
// for case of dilation angle equal to half of friction angle
//

void DruckerPragerFromMohrCoulomb(float fricAngle, float cohesion,
  float& dcSlopeAngle, float& dcIntercept)
{
  float phi[]   =   {1.00,5.00,10.0,20.0,30.0,40.0,50.0};
  float r1Beta[] = {0.03,0.15,16.7,30.2,39.8,46.2,50.5}; // Associative flow
  float r1dc[] =    {1.73,1.72,1.70,1.60,1.44,1.24,1.02}; // Associative flow
  float r0Beta[] = {0.03,0.15,16.7,30.6,40.9,48.1,53.0}; // Nondilatant flow
  float r0dc[]  =  {1.73,1.73,1.70,1.63,1.50,1.33,1.11}; // Nondilatant flow

//
// Use average of associative and nondilatant flow values for case of
// dilation equal to half of friction angle
//
  float beta = LinearInterp(&phi[0],&r1Beta[0],fricAngle,7);
  beta += LinearInterp(&phi[0],&r0Beta[0],fricAngle,7);
  dcSlopeAngle = 0.5*beta;

  float dRatio = LinearInterp(&phi[0],&r1dc[0],fricAngle,7);
  dRatio += LinearInterp(&phi[0],&r0dc[0],fricAngle,7);
  dcIntercept = cohesion * (0.5*dRatio);
}

//
// 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]);

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

D:\ThermMech\RepoScale\mechProps.cpp

Page 6

```
    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
}
```

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From

esfRmQ.txt

Digitized ESF Q profile (full peripheral map data).
Multiply Column 1 by 1000 to get ESF North-South distance in m.
Q = 10^y where y is data in Column 2.

Above Notes by G I Ofoegbu (1998 March 20)

....+....1....+....2....+....+....+....+

2.720025	0.641789
2.736191	0.692841
2.743634	0.600129
2.747722	0.511408
2.755186	0.416274
2.757841	0.339640
2.767405	0.309096
2.773864	0.253495
2.787401	0.301285
2.788248	0.356982
2.791892	0.396574
2.792900	0.433711
2.801340	0.456417
2.812211	0.428312
2.823844	0.388918
2.830851	0.423711
2.831208	0.459225
2.834306	0.485091
2.846081	0.506226
2.845670	0.553836
2.847286	0.597437
2.851643	0.631389
2.857321	0.666165
2.858371	0.698461
2.865510	0.717922
2.875894	0.746303
2.885550	0.781938
2.897192	0.818405
2.912715	0.867029
2.923721	0.900260
2.930686	0.939895
2.939117	0.886740
2.945561	0.832752
2.949210	0.794868
2.949607	0.748873
2.953144	0.723900
2.957327	0.778026
2.960348	0.812768
2.966103	0.838667
2.969192	0.788671
2.975602	0.738718
2.982467	0.712982
2.983949	0.695246
2.984755	0.755784
2.989713	0.797008
2.989357	0.838162
2.989030	0.876088
2.994729	0.908444
2.998197	0.891541
3.001887	0.848816
3.008130	0.818229
3.015074	0.860285
3.015536	0.883695
3.022563	0.916068
3.025640	0.944354
3.029275	0.908085
3.039719	0.852534
3.042759	0.808187

esfRmQ.txt

3.046505	0.759006
3.055814	0.834988
3.059947	0.894763
3.060844	0.944810
3.067305	0.965877
3.077786	0.982961
3.077528	1.012818
3.078452	1.059638
3.082865	1.087134
3.096049	1.021935
3.101976	0.950993
3.100451	0.896902
3.104833	0.850957
3.106447	0.817890
3.116684	0.863217
3.133224	0.794026
3.136831	0.760984
3.151841	0.715177
3.156133	0.679723
3.160369	0.650725
3.167852	0.707315
3.173978	0.767115
3.179889	0.851930
3.186659	0.914160
3.192924	0.957821
3.202504	1.002332
3.212133	1.041195
3.219140	1.075988
3.225489	1.109966
3.233929	1.132673
3.244394	1.074702
3.248216	1.016645
3.256654	0.962682
3.261106	0.908669
3.264755	0.870785
3.265677	0.840937
3.265886	0.816728
3.282883	0.848423
3.295097	0.818721
3.307478	0.769652
3.319160	0.724610
3.324982	0.665771
3.325428	0.614127
3.336432	0.570689
3.345097	0.644242
3.341736	0.725709
3.343213	0.785449
3.348891	0.820226
3.358339	0.880069
3.363262	0.925327,
3.372933	0.959348
3.379454	0.896483
3.381570	0.805315
3.381575	0.727840
3.391647	0.638389
3.397665	0.556957
3.407668	0.475576
3.411372	0.431236
3.412684	0.356199
3.418367	0.313500
3.426382	0.385430
3.428168	0.486333
3.437538	0.555052
3.443168	0.595477
3.457370	0.643277

esfRmQ.txt

3.461887	0.658669
3.467653	0.606286
3.468694	0.562720
3.483046	0.516097
3.489547	0.455654
3.490021	0.400782
3.487900	0.338612
3.496891	0.297570
3.497994	0.246741
3.509713	0.274332
3.510720	0.311469
3.514854	0.371243
3.511905	0.405100
3.533640	0.349697
3.538534	0.321514
3.544024	0.378078
3.542187	0.436967
3.541880	0.472473
3.558962	0.494485
3.557243	0.539656
3.568128	0.509937
3.562638	0.453373
3.568481	0.392114
3.571570	0.342118
3.579195	0.459237
3.599941	0.287607
3.607985	0.432978
3.614491	0.525872
3.615633	0.624345
3.631228	0.741568
3.641704	0.836127
3.652153	0.703101
3.661540	0.616063
3.671840	0.654127
3.685431	0.618794
3.684713	0.701909
3.691550	0.679401
3.700292	0.744077
3.712910	0.667572
3.729181	0.706521
3.732160	0.746104
3.735804	0.785696
3.742181	0.816446
3.752080	0.747169
3.760268	0.798926
3.779108	0.694255
3.791361	0.813856
3.808317	0.619580
3.822340	0.765028
3.839145	0.665173
3.845959	0.799230
3.864408	0.508936
3.882721	0.542261
3.903341	0.615969
3.914559	0.470848
3.914289	0.348175
3.930270	0.266871
3.941320	0.141117
3.953755	0.239736
3.963931	0.368994
3.971792	0.458677
3.977452	0.572543
3.992174	0.483153
4.006030	0.417155
4.014012	0.338976

esfRmQ.txt

4.016520	0.279288
4.017085	0.213925
4.017559	0.159053
4.029031	0.138218
4.039949	0.258608
4.059487	0.380724
4.066979	0.513176
4.068828	0.606816
4.082177	0.676394
4.105450	0.596799
4.113788	0.477466
4.121175	0.391208
4.126494	0.313802
4.126766	0.282331
4.148188	0.263240
4.149107	0.156723
4.163949	0.361096
4.170538	0.444306
4.184445	0.449328
4.186544	0.513918
4.194160	0.555176
4.211673	0.527157
4.213220	0.578827
4.214032	0.638559
4.235517	0.612205
4.239944	0.638087
4.252011	0.394519
4.261243	0.248565
4.259884	0.175107
4.260581	0.094412
4.268048	0.075947
4.280912	0.047867
4.287872	0.164978
4.291518	0.281238
4.295464	0.362800
4.315076	0.399371
4.322705	0.439015
4.336935	0.483587
4.349206	0.524097
4.364079	0.571098
4.361543	0.634014
4.364418	0.685702
4.383776	0.597986
4.384347	0.531816
4.384129	0.480163
4.389043	0.449559
4.439561	0.061222
4.443843	0.180719
4.463902	0.165646
4.469441	0.216561
4.476280	0.270721
4.477862	0.318357
4.497514	0.273417
4.499040	0.327509
4.503970	0.371960
4.512083	0.432593
4.512371	0.476176
4.507407	0.512429
4.533590	0.480487
4.538124	0.570933
4.553006	0.462983
4.555570	0.396840
4.557323	0.347633
4.560895	0.318626
4.569999	0.341341

esfRmQ.txt

4.581205	0.274502
4.593281	0.337608
4.597415	0.397382
4.613912	0.256364
4.621025	0.124909
4.626567	0.021681
4.634969	-0.028247
4.643161	-0.130634

END
12349

4.653180	-0.136961
4.669761	0.019819
4.677002	0.181321
4.681445	0.282258
4.707547	0.336668
4.720525	0.141533
4.753652	0.151647
4.784726	0.014853
4.817425	0.228335
4.866970	0.029639
4.912256	0.323988
4.955647	0.068719
5.011900	0.016184
5.040277	0.191679
5.090523	0.065626
5.136376	0.602093
5.194579	0.246944
5.232500	0.163504
5.283701	0.465192
5.314040	0.259791
5.349554	0.378079
5.397169	0.172094
5.456208	0.720031
5.543355	0.320870
5.613897	-0.309318
5.652709	0.196390
5.681489	0.017601
5.719894	0.262631
5.794198	-0.110871
5.819331	0.055704
5.845294	-0.104559
5.885682	0.295447
5.946276	0.432622
6.003926	0.295366
6.026807	-0.277334
6.067609	-0.079081
6.091405	-0.219197
6.108730	0.005384
6.149955	0.077745
6.198272	-0.517222
6.258711	0.330144
6.302341	0.124108
6.309108	0.340481

END
12350
END
END

Page No.

Witnes

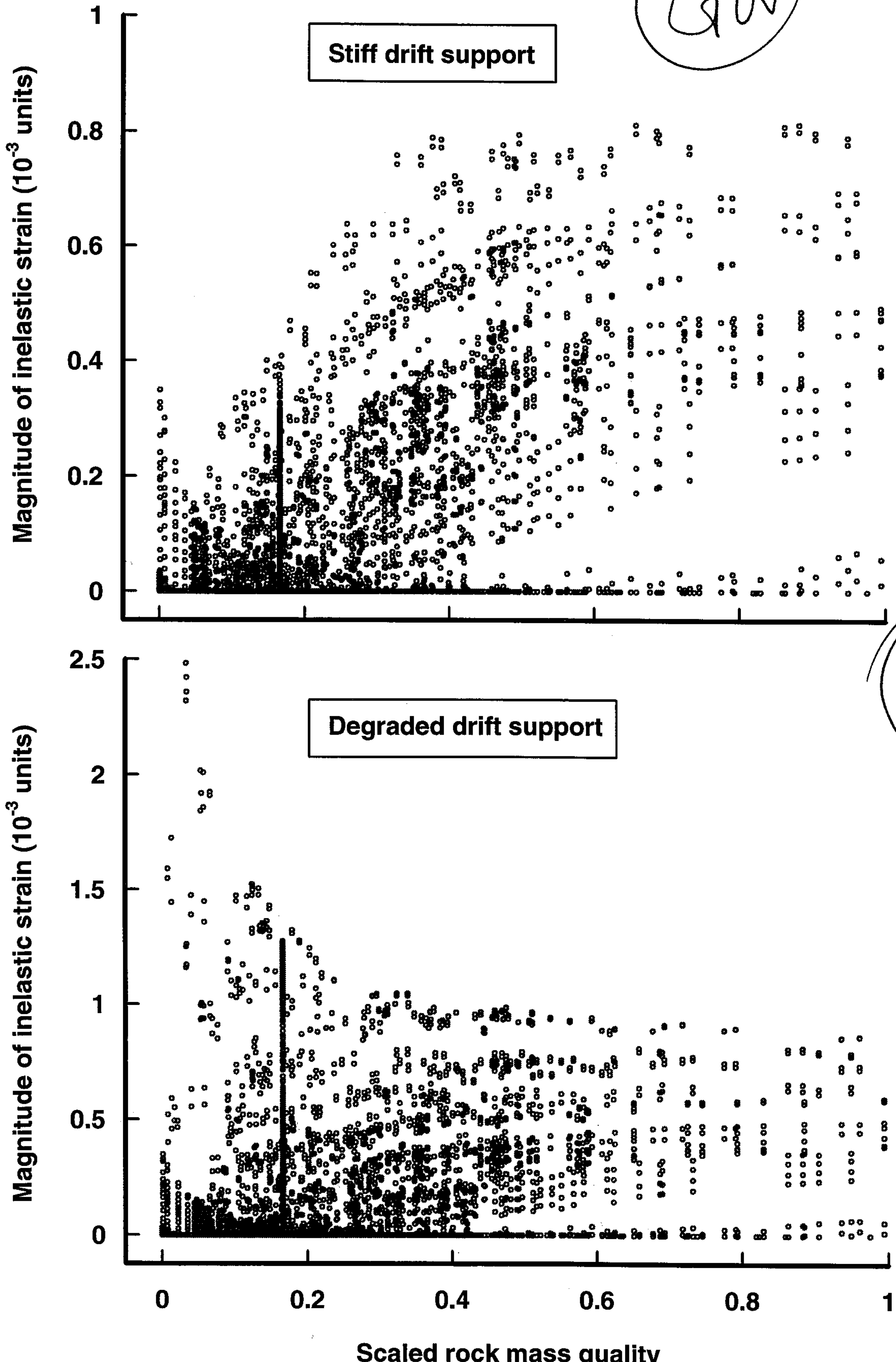
Inv

Recorded by

Project No.____

Book No.____ TITLE____ GW

Fro



GW

e No.____

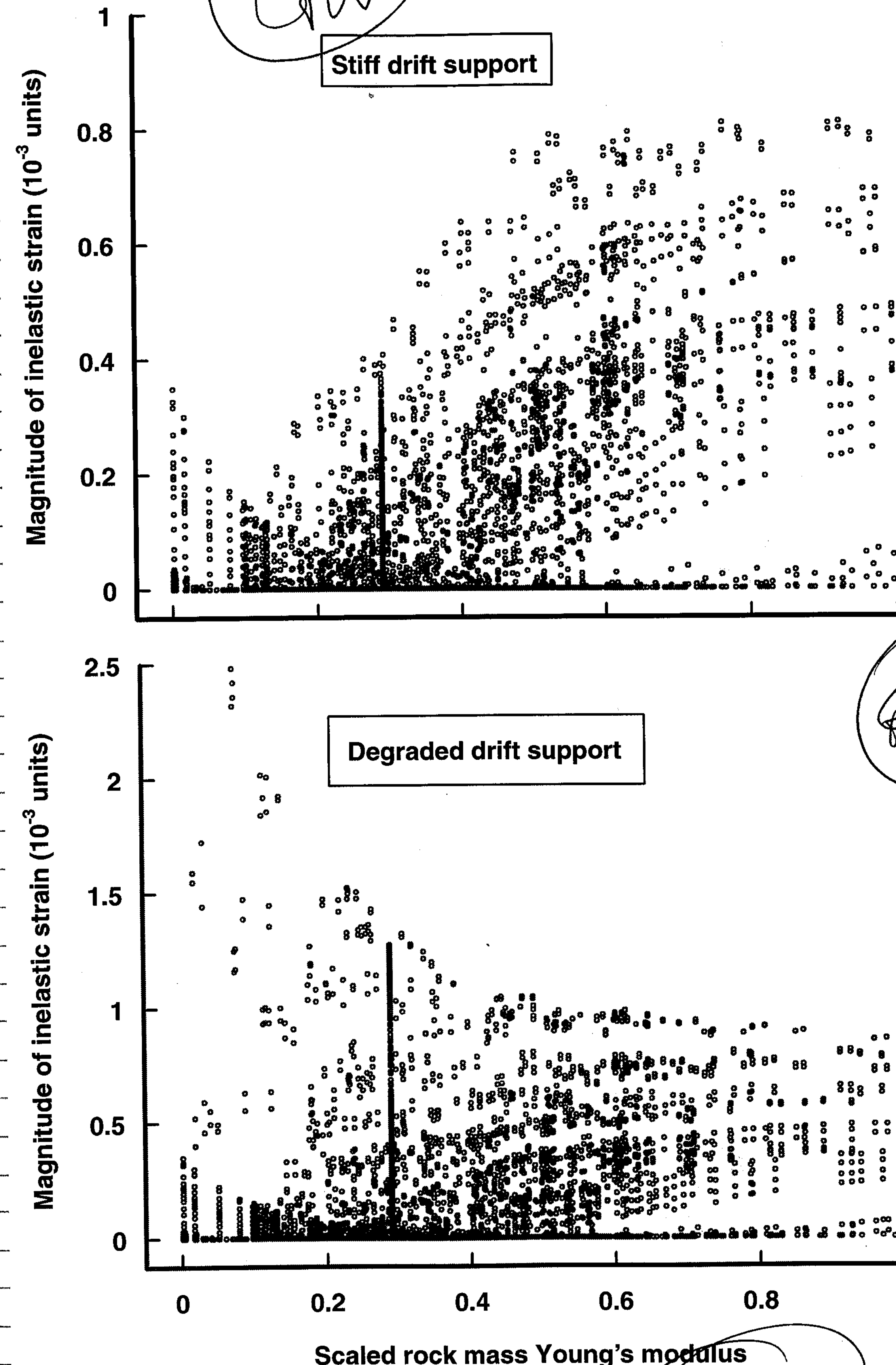
Witnessed & Understood by me,	Date	Invented by GW	Date
		Recorded by	

TITLE____ GW

Project No.____

Book No.____

From



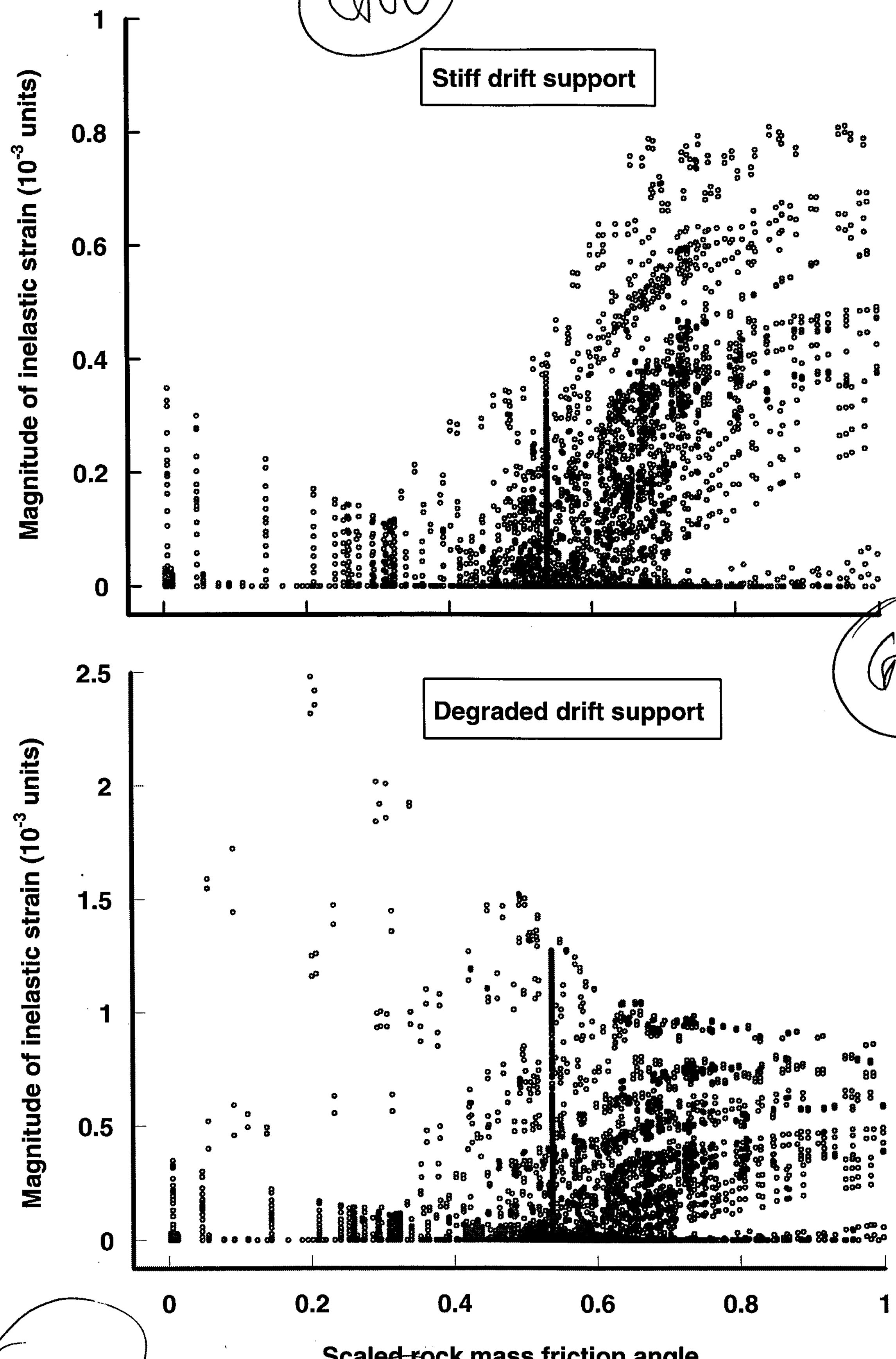
GW

GW

age No.____

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

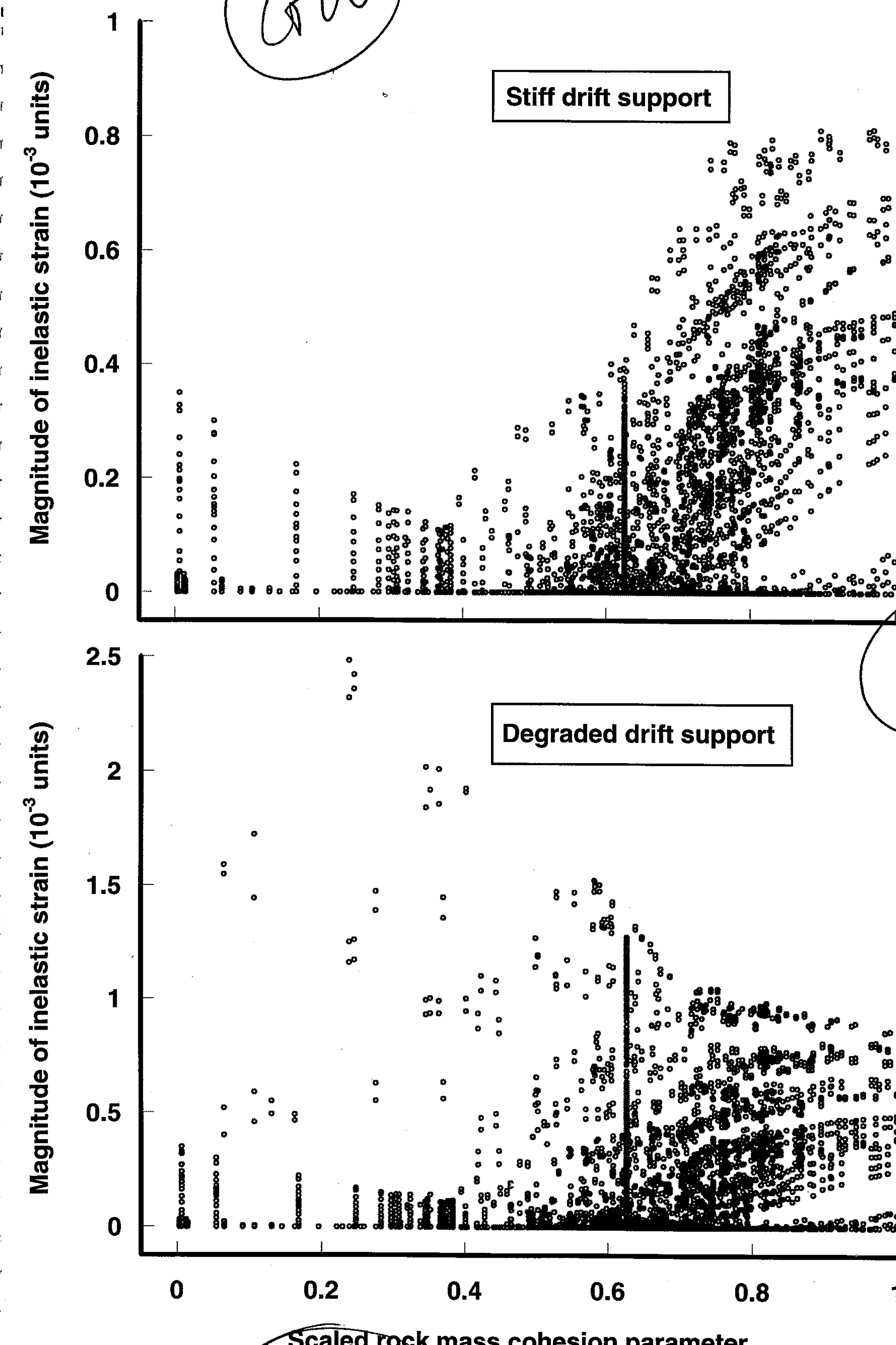
From P



Witness

Recorded by

From



Witnessed & Understood by me, | Date | Invented by | Date

Recorded by

From Page No. ____

Analysis to Develop Cummulative Percentile Curves of (Distributions) of Rock Mass Quality.

GVO 10/12/98

This analysis uses the output file from code StrnVsProps (referred to as outFile on p. 40). The file gives values of x and y coordinates, rock mass quality (Q), Young's modulus, friction angle, and cohesion at a large number of finite-element integration points. Two such output files are analysed:

s0150ScldProp.dat:	Case of stiff drift support
u0150ScldProp.dat:	Case of degraded drift support.

The files are processed using SPLUS (statistical analysis code) through the following functions:

Analysis for total integration-point population, ~~irrespective~~ of yield status.

```
function(fig = F, tbl = F)
{
        subdir <- "RepoScale/PlasticModel/"
        infile <- paste(subdir, "u0150ScldProp.dat", sep = "")
        figfil <- paste(subdir, "rmQ.ps", sep = "")
        tblfil <- paste(subdir, "rmQ.dat")
        mat <- list(x = 0, y = 0, q = 0, eMod = 0, fric = 0, cpar = 0, strn = 0
                )
        mat <- scan(infile, what = mat, skip = 1)
        rmq <- mat$q
        pLevels <- seq(0, 1, 0.05)
        rmqLevels <- quantile(rmq, probs = pLevels)
        plot(rmqLevels, 100 * pLevels, type = "l", xlab =
                "Scaled rock mass quality", ylab = "Occurrence percentile")
        if(fig) {
                ps.portrait()
                dev.print(device = postscript, file = figfil)
        }
        if(tbl) {
                qMin <- 0.729
                qMax <- 13.573
                xq <- qMin + (qMax - qMin) * rmqLevels
                cat(paste("ScaledRMQ", "RMQ", "Percentile", "\n", sep = "    "),
                        file = tblfil)
                cat(paste(format(rmqLevels), format(xq), format(100 * pLevels),
                        "\n", sep = "      "), file = tblfil, append = T)
        }
}
```

Reads table into memory using format defined on previous line, which corresponds ~~with input file~~ format.

Calculates percentile values of rock mass quality at the percentage levels in pLevels.

Prints output to table.

GVO GVO

10/12/98

To Page No. ______

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

Analysis for integration points under plastic state for case of stiff drift support.

```
function(fig = F, tbl = F)
{
        subdir <- "RepoScale/PlasticModel/"
        infile <- paste(subdir, "s0150ScldProp.dat", sep = "")
        figfil <- paste(subdir, "s0150g00rmq.ps", sep = "")
        tblfil <- paste(subdir, "s0150g00rmq.dat")
        mat <- list(x = 0, y = 0, q = 0, eMod = 0, fric = 0, cpar = 0, strn = 0
                )
        mat <- scan(infile, what = mat, skip = 1)
        strn <- mat$strn
        rmq <- mat$q[strn > 0]
        pLevels <- seq(0, 1, 0.05)
        rmqLevels <- quantile(rmq, probs = pLevels)
        plot(rmqLevels, 100 * pLevels, type = "l", xlab =
                "Scaled rock mass quality", ylab = "Yield percentile")
        if(fig) {
                ps.portrait()
                dev.print(device = postscript, file = figfil)
        }
        if(tbl) {
                qMin <- 0.729
                qMax <- 13.573
                xq <- qMin + (qMax - qMin) * rmqLevels
                cat(paste("ScaledRMQ", "RMQ", "CumYield%", "\n", sep = "    "),
                        file = tblfil)
                cat(paste(format(rmqLevels), format(xq), format(100 * pLevels),
                        "\n", sep = "      "), file = tblfil, append = T)
        }
}
```

Extracts Q values for all points with $\Gamma^N > 0$.

Calculates Q percentiles for this population.

Prints result.

GVO

Results are written to output files as follows:

rmQ.dat

File name

See code on p. 50, where this file is identified as variable tblfil.

ScaledRMQ	RMQ	Percentile
0.000	0.729000	0
0.058	1.473952	5
0.089	1.872116	10
0.108	2.116152	15
0.131	2.411564	20
0.161	2.796884	25
0.166	2.861104	30
0.166	2.861104	35
0.166	2.861104	40
0.166	2.861104	45
0.166	2.861104	50
0.166	2.861104	55
0.177	3.002388	60
0.208	3.400552	65
0.262	4.094128	70
0.302	4.607888	75
0.356	5.301464	80
0.406	5.943664	85
0.472	6.791368	90
0.581	8.191364	95
0.994	13.495936	100

GVO GVO

Rock-mass quality distribution in terms of cummulative percentiles for total population of integration points.

To Page No. ______

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

GVO

s0150g00rmq.dat ← File name. See variable tblfil in code on p. ~~52~~ 51. 10/12/98

ScaledRMQ	RMQ	CumYield%
0.0000	0.729000	0
0.0540	1.422576	5
0.1076	2.111014	10
0.1400	2.527160	15
0.1660	2.861104	20
0.1660	2.861104	25
0.1660	2.861104	30
0.1660	2.861104	35
0.1660	2.861104	40
0.1940	3.220736	45
0.2330	3.721652	50
0.2740	4.248256	55
0.3080	4.684952	60
0.3480	5.198712	65
0.3710	5.494124	70
0.4100	5.995040	75
0.4590	6.624396	80
0.4900	7.022560	85
0.5680	8.024392	90
0.6930	9.629892	95
0.9940	13.495936	100

Rock-mass quality distribution in terms of cummulative percentiles for integration points under yield state for case of stiff ~~were~~ drift support.

u0150g00rmQ.dat ← File Name

ScaledRMQ	RMQ	CumYield%
0.000	0.729000	0
0.053	1.409732	5
0.091	1.897804	10
0.124	2.321656	15
0.155	2.719820	20
0.166	2.861104	25
0.166	2.861104	30
0.166	2.861104	35
0.166	2.861104	40
0.166	2.861104	45
0.196	3.246424	50
0.237	3.773028	55
0.279	4.312476	60
0.318	4.813392	65
0.356	5.301464	70
0.392	5.763848	75
0.439	6.367516	80
0.475	6.829900	85
0.551	7.806044	90
0.687	9.552828	95
0.994	13.495936	100

Rock-mass quality distribution in terms of cummulative percentiles for integration points under yield state for case of degraded drift support.

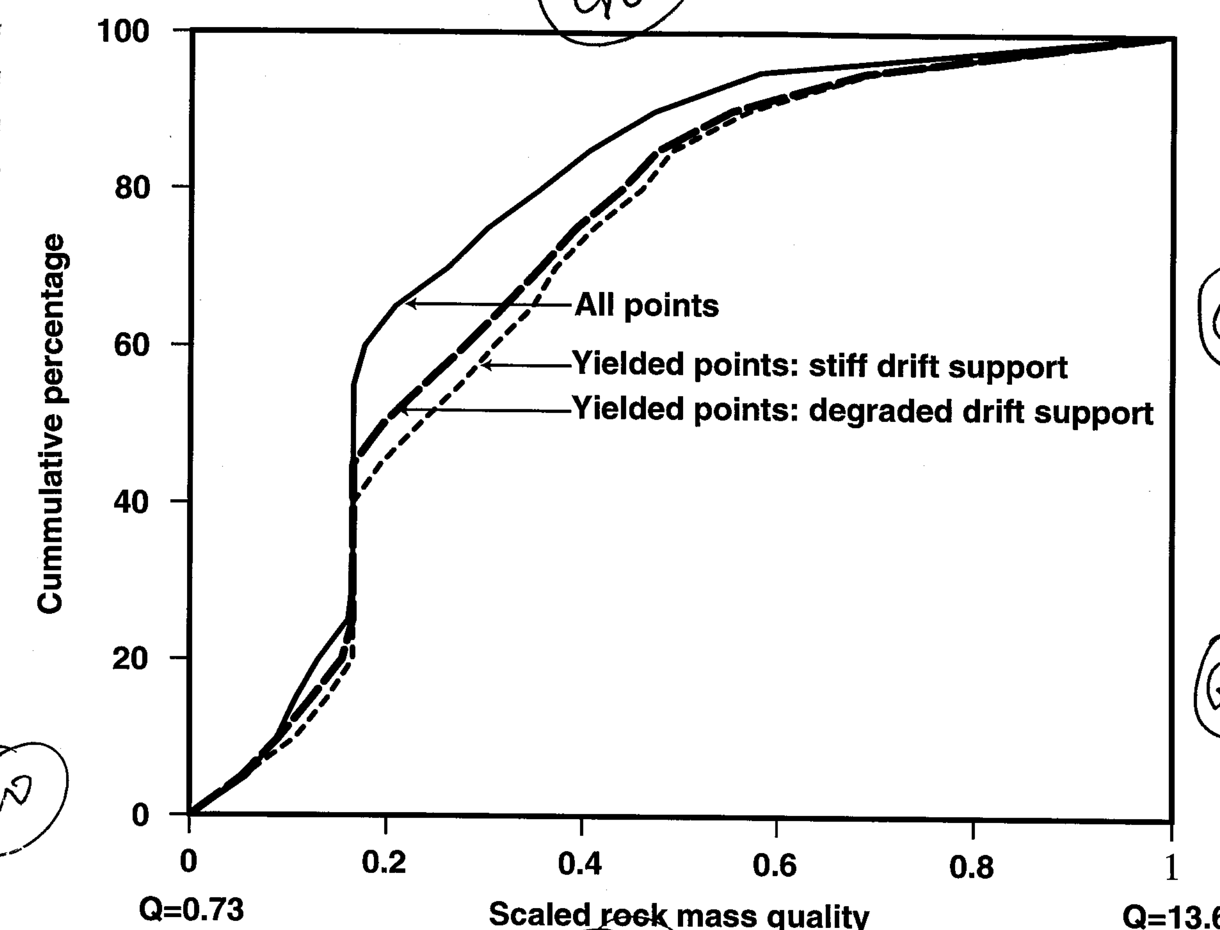
10/12/98

The three distributions (files on pp. 51 and 52) were plotted together using ~~Kaleidograph~~ XPLOT 4.1. The resulting postscript file was ~~red~~ read into

To Page No.______

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

From Page No.______ Adobe Illustrator where the annotations were enhanced.



$$\text{ScaledRockMassQuality} = \frac{Q - Q_{min}}{Q_{max} - Q_{min}}$$

$Q_{min} = 0.73$
$Q_{max} = 13.6$ } see p. 39.

Pages 32-53 entries by GVO 10/12/98

To Page No.______

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

Pages 54-58 entries GW 10/23/98

From Page No.

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.

<u>Relationship Between Fracture Porosity Change and Inelastic Volumetric Strain</u>

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

$$\varepsilon^{\mathrm{N}} = \frac{v_{t1} - v_{to}}{v_{to}} = \frac{v_{f1} - v_{fo}}{v_{to}} = \frac{\Delta v_f}{v_{to}} \tag{1}$$

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

$$\phi_{fo} = \frac{v_{fo}}{v_{to}} \tag{2}$$

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

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

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

$$\phi_{f1} = \frac{\phi_{fo} + \varepsilon^{\mathrm{N}}}{1 + \varepsilon^{\mathrm{N}}} \tag{4}$$

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

$$\phi_{f1} = \phi_{fo} + \varepsilon^{\mathrm{N}} \tag{5}$$

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.

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

$$\Delta\phi_f = \varepsilon^{\mathrm{N}} \tag{6}$$

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

<u>Fracture Porosity and Fracture Aperture</u>

It is assumed that all the fractures in the hypothetical rock body can be consolidated into a single fracture of aperture b. The fracture volume $v_f = bA_f$, where A_f is the surface area of the consolidated fracture (i.e., area of an imaginary surface parallel to and midway between the fracture walls). The total rock volume $v_t = A_f\lambda_c$, where λ_c is a characteristic length of the rock body normal to the fracture surface. For example, if the rock body is rectangular with the consolidated fracture parallel to one of the faces, then λ_c is the fracture-normal dimension of the rectangle.

It can be shown using the definition of fracture porosity that the consolidated fracture apertures at the beginning and end of the load-unload cycle, b_o and b_1, respectively, are related to fracture porosity as follows:

$$b_o = \lambda_c\phi_{fo} \qquad \text{and} \qquad b_1 = \lambda_c(\phi_{fo} + \Delta\phi_f) \tag{7}$$

<u>Fracture Permeability, Fracture Porosity, and Inelastic Volumetric Strain</u>

Fracture permeability κ_f is related to fracture aperture through the equation

$$\kappa_f = \frac{b^2}{12} \tag{8}$$

Therefore, using the consolidated-fracture concept, the fracture permeabilities κ_{fo} and κ_{f1}, at the beginning and end, respectively, of the load-unload cycle are:

$$\kappa_{fo} = \frac{(\lambda_c\phi_{fo})^2}{12} \qquad \text{and} \qquad \kappa_{f1} = \frac{[\lambda_c(\phi_{fo} + \Delta\phi_f)]^2}{12} \tag{9}$$

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

$$\kappa_{f1} = R_\kappa\kappa_{fo} \qquad \text{where} \qquad R_\kappa = \left(1 + \frac{\Delta\phi_f}{\phi_{fo}}\right)^2 = \left(1 + \frac{\varepsilon^{\mathrm{N}}}{\phi_{fo}}\right)^2 \tag{10}$$

The only input data required to calculate R_κ using this procedure (equation 10) is the

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.____ initial fracture porosity ϕ_{fo} (where the word "initial" refers to the strain-free state). Fracture porosity data for the stratigraphic units at YM is given in the following document (cited hereafter as TSPA-VA2)

Chapter 2, Total System Performance Assessment — Viability Assessment Analyses Technical Basis Document: Unsaturated Zone Hydrology Model.
CRWMS/M&O Document Number
B00000000-01717-4301-00002 REV 00 (Aug 14 1998)

Table 2-19 of the document ~~gives~~ (reproduced on p.57) gives ϕ_{fo} for the TSw unit & that ranges from 8.92×10^{-5} to 4.92×10^{-4}. A value of ϕ_{fo} of 1×10^{-4} will be applied in the model to obtain order-of-magnitude estimates of R_k.

10/23/98

The procedure described on p. 54-55 for calculation of permeability change associated with inelastic deformation was implemented in ABAQUS as a user-defined code module (see file KChange.f reproduced on p.58).

To Page No.________

Table 2-19. Matrix (a) and Fracture (b) Parameter Sets for the DKM Base Case (Minimum ∀f Present-day Infiltration Divided by Three). (DTN: LB971212001254.002)

Model Layer / Block Name	matrix permeability k_m (m^2)	matrix porosity ϕ_m (-)	matrix van Genuchten alpha α_m (1/Pa)	matrix van Genuchten m (λ) m_m (-)	matrix residual saturation S_{rm} (-)	matrix satiated saturation S_{sm} (-)
tcw11	5.37E-18	0.066	1.18E-06	0.232	0.13	1.00
tcw12	5.37E-18	0.066	1.32E-06	0.236	0.13	1.00
tcw13	4.90E-17	0.140	6.46E-07	0.427	0.33	1.00
ptn21	3.09E-14	0.369	3.80E-05	0.231	0.10	1.00
ptn22	3.02E-15	0.234	8.71E-06	0.488	0.14	1.00
ptn23	8.32E-14	0.353	4.57E-05	0.287	0.17	1.00
ptn24	1.15E-13	0.469	4.27E-05	0.349	0.10	1.00
ptn25	2.46E-13	0.464	1.95E-04	0.279	0.10	1.00
tsw31	4.90E-17	0.042	1.00E-05	0.237	0.11	1.00
tsw32	2.75E-16	0.146	2.29E-05	0.273	0.04	1.00
tsw33	1.15E-17	0.135	6.76E-06	0.248	0.06	1.00
tsw34	4.07E-18	0.089	1.02E-06	0.322	0.18	1.00
tsw35	1.55E-17	0.115	3.31E-06	0.229	0.08	1.00
tsw36	8.91E-17	0.092	7.41E-07	0.414	0.18	1.00
tsw37	1.29E-17	0.020	1.55E-06	0.387	0.50	1.00
ch1zc	1.38E-17	0.193	8.32E-07	0.366	0.36	1.00
ch2zc	9.12E-18	0.240	1.95E-06	0.220	0.20	1.00
ch3zc	9.12E-18	0.240	1.95E-06	0.220	0.20	1.00
ch4zc	1.55E-17	0.169	7.76E-07	0.477	0.33	1.00
ch1vc	1.32E-12	0.265	6.61E-05	0.190	0.04	1.00
ch2vc	2.57E-13	0.321	7.41E-05	0.224	0.06	1.00
ch3vc	2.57E-13	0.321	7.41E-05	0.224	0.06	1.00
ch4vc	2.57E-13	0.321	7.41E-05	0.224	0.06	1.00
pp3vp	2.82E-15	0.274	1.74E-05	0.311	0.07	1.00
bf3vb	2.82E-15	0.274	1.74E-05	0.311	0.07	1.00
tm3vt	2.82E-15	0.274	1.74E-05	0.311	0.07	1.00
pp2zp	5.75E-17	0.197	1.66E-06	0.316	0.18	1.00
bf2zb	5.75E-17	0.197	1.66E-06	0.316	0.18	1.00
tsw37/pcM37	6.08E-19	0.036	3.37E-07	0.372	0.20	1.00
ch1zc/pcM1z	5.40E-19	0.288	1.90E-07	0.359	0.36	1.00
ch2zc/pcM2z	4.50E-20	0.332	4.21E-06	0.228	0.20	1.00
ch2zc/pcM62	4.50E-19	0.332	4.21E-06	0.228	0.20	1.00
ch3zc/pcM3z	4.50E-19	0.332	4.21E-06	0.228	0.20	1.00
ch4zc/pcM4z	8.40E-19	0.266	1.50E-07	0.476	0.33	1.00

(a)

TSw unit

Model Layer / Block Name	vertical fracture permeability k_f (m^2)	horizontal fracture permeability k_f (m^2)	fracture porosity ϕ_f (-)	fracture van Genuchten alpha α_f (1/Pa)	fracture van Genuchten m (λ) m_f (-)	fracture residual saturation S_{irf} (-)	fracture satiated saturation S_{sf} (-)	fracture frequency f (1/m)	fracture-matrix connection area modification factor X_{fm} (-)
tcw11	2.29E-11	6.21E-12	2.33E-04	5.73E-04	0.492	0.01	1.00	1.020	1.90E-01
tcw12	1.38E-11	6.03E-12	2.99E-04	4.88E-04	0.492	0.01	1.00	1.830	1.90E-01
tcw13	2.82E-12	2.40E-13	7.05E-05	5.67E-04	0.492	0.01	1.00	1.270	1.90E-01
ptn21	5.25E-13	5.25E-13	4.84E-05	5.85E-04	0.492	0.01	1.00	0.870	1.00E+00
ptn22	1.95E-13	1.95E-13	4.83E-05	2.74E-04	0.492	0.01	1.00	0.290	1.00E+00
ptn23	2.57E-13	2.57E-13	1.30E-04	6.18E-04	0.492	0.01	1.00	0.290	1.00E+00
ptn24	6.17E-14	6.17E-14	6.94E-05	2.35E-04	0.492	0.01	1.00	0.630	1.00E+00
ptn25	7.76E-14	7.76E-14	3.86E-05	4.14E-04	0.279	0.01	1.00	0.650	1.00E+00
tsw31	1.07E-11	1.00E-12	8.92E-05	3.98E-05	0.481	0.01	1.00	1.100	5.00E-01
tsw32	1.51E-11	7.08E-13	1.29E-04	2.53E-04	0.488	0.01	1.00	1.010	1.90E-01
tsw33	2.63E-11	8.91E-13	1.05E-04	2.76E-04	0.492	0.01	1.00	0.690	1.90E-01
tsw34	6.76E-12	4.27E-13	1.24E-04	1.98E-04	0.492	0.01	1.00	1.880	1.90E-01
tsw35	3.80E-12	9.12E-13	3.29E-04	3.40E-04	0.492	0.01	1.00	1.810	1.90E-01
tsw36	1.20E-12	1.20E-12	3.99E-04	3.92E-04	0.492	0.01	1.00	2.100	1.90E-01
tsw37	1.20E-12	1.20E-12	4.92E-04	5.77E-04	0.492	0.01	1.00	2.880	1.90E-01
ch1zc	2.40E-14	2.40E-14	1.10E-05	2.85E-04	0.492	0.01	1.00	0.067	1.00E+00
ch2zc	1.18E-14	1.18E-14	1.10E-05	2.85E-04	0.492	0.01	1.00	0.067	1.00E+00
ch3zc	1.18E-14	1.18E-14	1.10E-05	2.85E-04	0.492	0.01	1.00	0.067	1.00E+00
ch4zc	1.55E-14	1.55E-14	1.10E-05	2.85E-04	0.492	0.01	1.00	0.067	1.00E+00
ch1vc	1.74E-13	1.74E-13	7.14E-05	5.42E-04	0.492	0.01	1.00	0.420	1.00E+00
ch2vc	2.88E-13	2.88E-13	7.14E-05	5.42E-04	0.492	0.01	1.00	0.420	1.00E+00
ch3vc	2.88E-13	2.88E-13	7.14E-05	5.42E-04	0.492	0.01	1.00	0.420	1.00E+00
ch4vc	2.88E-13	2.88E-13	7.14E-05	5.42E-04	0.492	0.01	1.00	0.420	1.00E+00
pp3vp	6.92E-13	6.92E-13	7.14E-05	2.53E-04	0.492	0.01	1.00	0.420	1.90E-01
bf3vb	6.92E-13	6.92E-13	7.14E-05	2.53E-04	0.492	0.01	1.00	0.420	1.90E-01
tm3vt	6.92E-13	6.92E-13	7.14E-05	2.53E-04	0.492	0.01	1.00	0.420	1.90E-01
pp2zp	6.46E-14	6.46E-14	1.10E-05	2.85E-04	0.492	0.01	1.00	0.067	1.00E+00
bf2zb	6.46E-14	6.46E-14	1.10E-05	2.85E-04	0.492	0.01	1.00	0.067	1.00E+00
tsw37/pcM37	3.04E-19	3.04E-19	4.92E-04	3.37E-07	0.372	0.20	1.00	2.880	1.00E+00
ch1zc/pcM1z	1.20E-18	1.20E-18	1.10E-05	1.90E-07	0.359	0.36	1.00	0.067	1.00E+00
ch2zc/pcM2z	3.50E-19	3.50E-19	1.10E-05	4.21E-06	0.228	0.20	1.00	0.067	1.00E+00
ch2zc/pcM62	4.50E-19	4.50E-19	1.10E-05	4.21E-06	0.228	0.20	1.00	0.067	1.00E+00
ch3zc/pcM3z	4.50E-19	4.50E-19	1.10E-05	4.21E-06	0.228	0.20	1.00	0.067	1.00E+00
ch4zc/pcM4z	8.40E-19	8.40E-19	1.10E-05	1.50E-07	0.476	0.33	1.00	0.067	1.00E+00
tsw37/pwF37	3.04E-19	1.50E-13	4.92E-04	3.37E-07	0.372	0.20	1.00	2.880	1.00E+00

(b)

From Page No.____

KChange.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     This code computes fracture-permeability change ratio (Kr)
C     following procedure documented in CNWRA Scientific Notebook
C     #263, pp. -- --
C
C     Values of inelastic strain required for the calculation are
C     obtained through ABAQUS user-interface subroutine GETVRM
C     The externally supplied input parameter is:
C
C     PHIOF        Average consolidated fracture porosity
C                  (i.e., total fracture volume per unit bulk volume)
C                  at strain-free state.
C
C     The calculated value of Kr is stored in UVAR(1)
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)
      RATIO = 1.0 + DPHIF/PHIOF
      UVAR(1) = RATIO*RATIO
      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
```

} Disabled because ABAQUS reports JRCD ≠ 0 if every PE component is zero.

GW

GW 10/23/98

Pages 54-58 entries by GW 10/23/98

To Page No.____

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

Pages 59-64 entries by GW 10/29/98

From Page No.____

Elastic-plastic behavior was hitherto applied over the entire model, but previous results indicate that plastic behavior is restricted to a thin zone above and below the repository horizon. As a result, the model is now simplified (by restricting elastic-plastic material behavior to a selected zone) to reduce both analysis time (computer time) and size of of results files.

Elastic Zone

All elements more than 80 m above or more than 80 m below repository horizon. The element-group names are

ELTZ	All Zone 2 (p. 6) elements above $y \approx 795$ m
ELBZ	All Zone 3 (p. 6) elements below $y \approx 600$ m
Z4ELEM	All Zone 4 Elements (p. 6)

Poisson's ratio and expansivity are as defined previously (p. 15). Young's modulus varies with x coordinate as defined previously (p. 22).

Elastic-Plastic Zone

All elements within 80 m above and 80 m below the emplacement drifts, i.e., the elastic-plastic zone is about 165 m thick.

To Page No.____

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

From Page No.___ The element-group names for the elastic-plastic zone are:

PLTZ All zone 2 elements (p. 6) below y ≈ 795 m

PLBZ All zone 3 elements (p. 6) above y ≈ 600 m

MZELEM All zone 1 elements (p. 6).

Nov 10 1998

Five analysis cases will be performed as follows

Case 1: Q varies with X as defined on p. 18

Case 2: Q is uniform and equal to minimum Q (i.e., Q at x = 55 m) from Case 1.

Case 3: Q is uniform and equal to maximum value (i.e., Q at x = 1474 m) from Case 1

Case 4: Same as Case 1, but analysis performed using current (instead of degraded) values of strength parameters

Case 5: Same as Case 1, but maximum

To Page No.____

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.___

friction-angle curve from the Hoek-Brown chart (p. 22) is used.

Case 1 Details

Input files:

~~m01Base.imp~~ GW 10/29/98

m01Base.inp Mechanical analysis to end of excavation (no heat load).

m01.inp Mechanical analysis from end of excavation (and instant waste emplacement) to 150 yr.

Auxiliary input files

allNodes.def Node definitions
mElem01.def Element definitions
iniTemp.def Initial nodal temperature
mdfld1.def X-as-field definition.

mElem01.def incorporates the following files

ElPropCurrent.def Elastic properties. Current values
FricAngLongTerm.def Drucker-Prager frictional behavior. Long-term values

CohesionLongTerm.def Drucker-Prager cohesion parameters. Long-term values.

GW 10/29/98

To Page No.____

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

From Page No.___

Case 2 Details

$Q = 0.7295$ (from $x = 55$ m in Case 1)

$E = 8.02 \times 10^3$ MPa (Young's modulus)

$\beta = 33.05^\circ$ Drucker-Prager friction angle (equivalent to Mohr-Coulomb friction angle of 23°)

Dilation angle $= 16.52^\circ$ GW 10/29/98

Drucker-Prager Cohesion $= 2.63$ MPa (equivalent to Mohr-Coulomb cohesion of 1.67 MPa).

Input files (See Case 1 for brief description of each file's content)

m02Base.inp
m02.inp

allNodes.def
mElem02.def
iniTemp.def

November 16 1998

Case 3 Details

$Q = 13.57$ (from $x = 1474$ m in Case 1)
$E = 3.65 \times 10^4$ MPa
$\beta = 40.46^\circ$ (equivalent $\phi = 30.2^\circ$)
Dilation angle $= 20.23^\circ$
Drucker-Prager Cohesion $= 5.16$ MPa (equivalent $c = 3.5$ MPa)

To Page No.___

From Page No.___

~~Inp~~ Input files

m03Base.inp
m03.inp

allNodes.def
mElem03.def
iniTemp.def.

Case 4 Details

Input Files (Compare with Case 1)

m04Base.inp
m04.inp

Auxiliary Input Files

allNodes.def
mElem04.def
iniTemp.def
mdfld1.def

mElem04.def incorporates the following files

- ElPropCurrent.def
  FricAngCurrent.def
  CohesionCurrent.def

GW
11/16/98

To Page No.___

From Page No.

Case 5 Details

Input Files

m05Base.inp
m05.inp

allNodes.def
mElem05.def
iniTemp.def
mdfld1.def

ElPropCurrent.def
CohesionLongTerm.def
Lt2FricDP.def ← Drucker-Prager friction ~~case~~ properties for case of friction angle from maximum curve on Hoek-Brown chart.

Pages 59-64 entries by GVO 10/29/98

GVO 11/16/98

Results Processing

Pages 64-69 entries by GVO 11/23/98

November 23 1998

Each analysis result is processed using ABAQUS/POST and the appropriate .res file from ABAQUS. Contour plots of plastic strain, permeability ratio R_k (p. 55), or rock-mass quality Q (p. 32-35). The contours are plotted by executing the following command files while inside ABAQUS/POST

peContPlots.jnl for plastic strain

```
input,file=peContPars.def
input,file=grpsOf10.def
input,file=g10PeContours.def
```
} File peContPlots.jnl

GVO

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

From Page No.

```
input,file=kContPars.def
input,file=grpsOf10.def
input,file=g10KContours.def
```
GVO } File kContPlots.jnl for R_k

```
input,file=qContPars.def
input,file=grpsOf10.def
input,file=g10QContours.def
```
GVO } File qContPlots.jnl for Q

The auxiliary command files pulled in by these files are reproduced here:

GVO

```
set,fill
set,outline=per
set,outlin=0
set,clev=5
set,clegendsize=.2
set,clegend=(.012,.99)
set,cmin=0.2e-3
set,cmax=2.0e-3
```
File peContPars.def

GVO

GVO

```
set,fill
set,outline=per
set,outlin=0
set,clev=5
set,clegendsize=.2
set,clegend=(.012,.99)
set,cmin=10
set,cmax=50
```
File kContPars.def

GVO

GVO

```
set,fill
set,outline=per
set,outlin=0
set,clev=5
set,clegendsize=.2
set,clegend=(.012,.99)
set,cmin=2.5
set,cmax=11.5
```
File qContPars.def

GVO

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

Project No.
Book No.
TITLE

From Page No.

File grpsOf10.def

```
elset,elset=pgrp01,gen
1409,4309,100
1410,4310,100
1411,4311,100
1412,4312,100
1413,4313,100

elset,elset=pgrp02,gen
4409,7309,100
4410,7310,100
4411,7311,100
4412,7312,100
4413,7313,100

elset,elset=pgrp03,gen
7409,10309,100
7410,10310,100
7411,10311,100
7412,10312,100
7413,10313,100

elset,elset=pgrp04,gen
10409,13309,100
10410,13310,100
10411,13311,100
10412,13312,100
10413,13313,100

elset,elset=pgrp05,gen
13409,16309,100
13410,16310,100
13411,16311,100
13412,16312,100
13413,16313,100

elset,elset=pgrp06,gen
16409,19309,100
16410,19310,100
16411,19311,100
16412,19312,100
16413,19313,100

elset,elset=pgrp07,gen
19409,22309,100
19410,22310,100
19411,22311,100
19412,22312,100
19413,22313,100

elset,elset=pgrp08,gen
22409,25309,100
22410,25310,100
22411,25311,100
22412,25312,100
22413,25313,100

elset,elset=pgrp09,gen
25409,28309,100
25410,28310,100
25411,28311,100
25412,28312,100
25413,28313,100

elset,elset=pgrp10,gen
28409,31309,100
28410,31310,100
28411,31311,100
28412,31312,100
28413,31313,100
```

GW GW GW GW GW

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

Project No.
TITLE

From Page No.

g10PeContours.def

GW 11/23/98

```
***
*** Plot Drifts #100 thru 91 (counting down, i.e., South to North)
***
detail,elset=pgrp01
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #90 thru 81 (counting down, i.e., South to North)
***
detail,elset=pgrp02
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #80 thru 71 (counting down, i.e., South to North)
***
detail,elset=pgrp03
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #70 thru 61 (counting down, i.e., South to North)
***
detail,elset=pgrp04
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #60 thru 51 (counting down, i.e., South to North)
***
detail,elset=pgrp05
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #50 thru 41 (counting down, i.e., South to North)
***
detail,elset=pgrp06
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #40 thru 31 (counting down, i.e., South to North)
***
detail,elset=pgrp07
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #30 thru 21 (counting down, i.e., South to North)
***
detail,elset=pgrp08
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #20 thru 11 (counting down, i.e., South to North)
***
detail,elset=pgrp09
set,clegend=on
cont,v=peeq
pause
***
*** Plot Drifts #10 thru 1 (counting down, i.e., South to North)
***
detail,elset=pgrp10
set,clegend=on
cont,v=peeq
```

GW GW GW

To Page No.

Witnessed & Und | Invented by | Date
Recorded by

Project No.

Book No. TITLE

g10KContours.def

From Page No.

GVO

```
***
*** Plot Drifts #100 thru 91 (counting down, i.e., South to North)
***
detail,elset=pgrp01
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #90 thru 81 (counting down, i.e., South to North)
***
detail,elset=pgrp02
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #80 thru 71 (counting down, i.e., South to North)
***
detail,elset=pgrp03
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #70 thru 61 (counting down, i.e., South to North)
***
detail,elset=pgrp04
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #60 thru 51 (counting down, i.e., South to North)
***
detail,elset=pgrp05
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #50 thru 41 (counting down, i.e., South to North)
***
detail,elset=pgrp06
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #40 thru 31 (counting down, i.e., South to North)
***
detail,elset=pgrp07
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #30 thru 21 (counting down, i.e., South to North)
***
detail,elset=pgrp08
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #20 thru 11 (counting down, i.e., South to North)
***
detail,elset=pgrp09
set,clegend=on
cont,v=uvarm1
pause
***
*** Plot Drifts #10 thru 1 (counting down, i.e., South to North)
***
detail,elset=pgrp10
set,clegend=on
cont,v=uvarm1
```

GVO GVO GVO

To Page No.

Witnessed & Un | Invented by | Date

Recorded by

Project No.

Book No.

TITLE

g10QContours.def

From Page No.

GVO

```
***
*** Plot Drifts #100 thru 91 (counting down, i.e., South to North)
***
detail,elset=pgrp01
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #90 thru 81 (counting down, i.e., South to North)
***
detail,elset=pgrp02
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #80 thru 71 (counting down, i.e., South to North)
***
detail,elset=pgrp03
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #70 thru 61 (counting down, i.e., South to North)
***
detail,elset=pgrp04
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #60 thru 51 (counting down, i.e., South to North)
***
detail,elset=pgrp05
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #50 thru 41 (counting down, i.e., South to North)
***
detail,elset=pgrp06
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #40 thru 31 (counting down, i.e., South to North)
***
detail,elset=pgrp07
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #30 thru 21 (counting down, i.e., South to North)
***
detail,elset=pgrp08
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #20 thru 11 (counting down, i.e., South to North)
***
detail,elset=pgrp09
set,clegend=on
cont,v=fv1
pause
***
*** Plot Drifts #10 thru 1 (counting down, i.e., South to North)
***
detail,elset=pgrp10
set,clegend=on
cont,v=fv1
```

GVO GVO GVO

Pages 64–69 entries by GVO 11/22/98

To Page No.

Witnessed & Unde | Invented by | Date

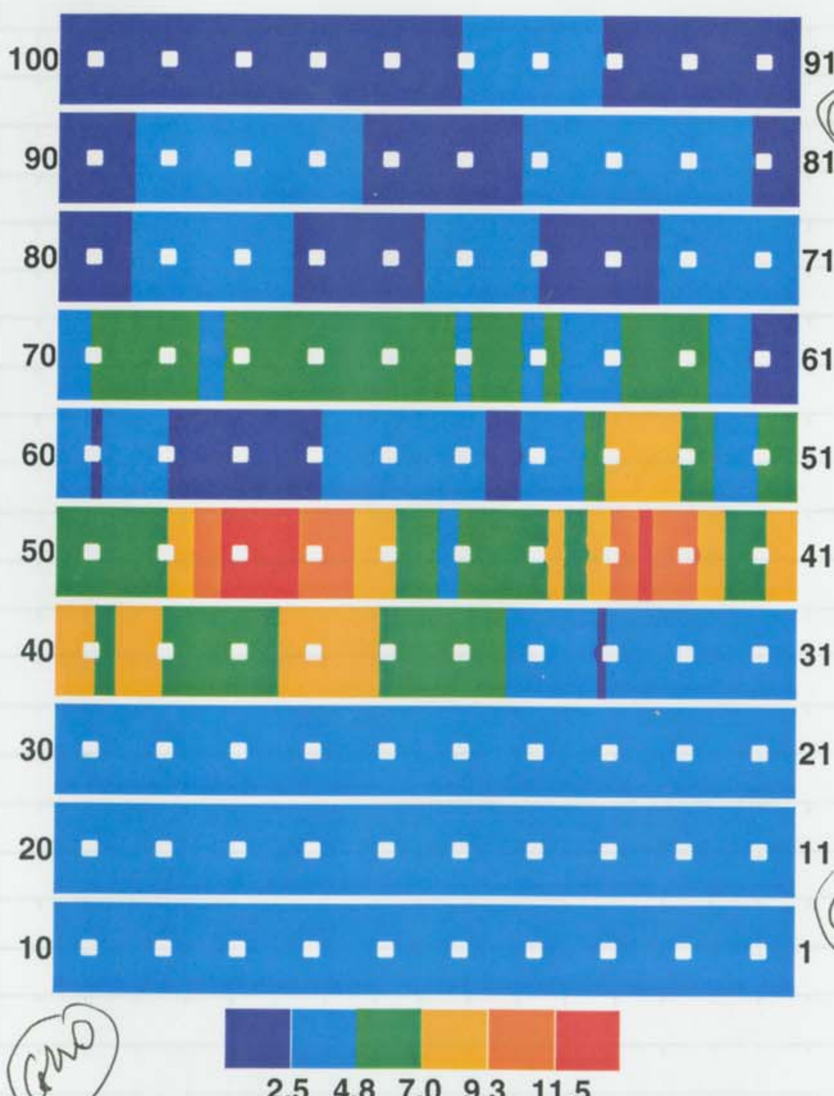
Recorded by

 Pages 70 – 76 entries by GW 12/14/98
Project No.
Book No.
TITLE Dec 14 1998

From Page No.

The plotting routines described on pp. 64-69 give contours of inelastic strain magnitude (Γ^N), permeability ratio R_k, and rock mass quality Q, over 35-m high strips of the model. Each strip is centered on the repository axis (15 m above roof, and 15 m below floor, of emplacement drifts) and covers 10 emplacement drifts (white squares). The strips were ~~combined~~ GW 12/14/98 assembled using ADOBE ILLUSTRATOR to obtain the composite plots presented here. Numbers at the ends of each ~~drif~~ strip indicate the range of emplacement drifts covered.

DISTRIBUTION OF ROCK-MASS QUALITY (Q)
ALONG EAST BOUNDARY OF PROPOSED DRIFTS



GW

(1) Recall that Q value at drift #32 was applied over drifts #31 thru #1 because of lack of Q data in that area. GW
(2) This figure shows Q decreasing southward generally.
(3) Maximum Q is 13.57 (≃ GSI of ~67.5).
(4) Minimum Q is ~~about~~ 0.7295 (≃ GSI of ~41.2)
(5) Minimum Q occurs ~~near~~ b/w drift #99 and #98. GW 12/14/98
(6) Maximum Q occurs b/w drift #48 and #47.
(7) This figure is same as ~~was~~ shown on p. 35.

GW

To Page No.

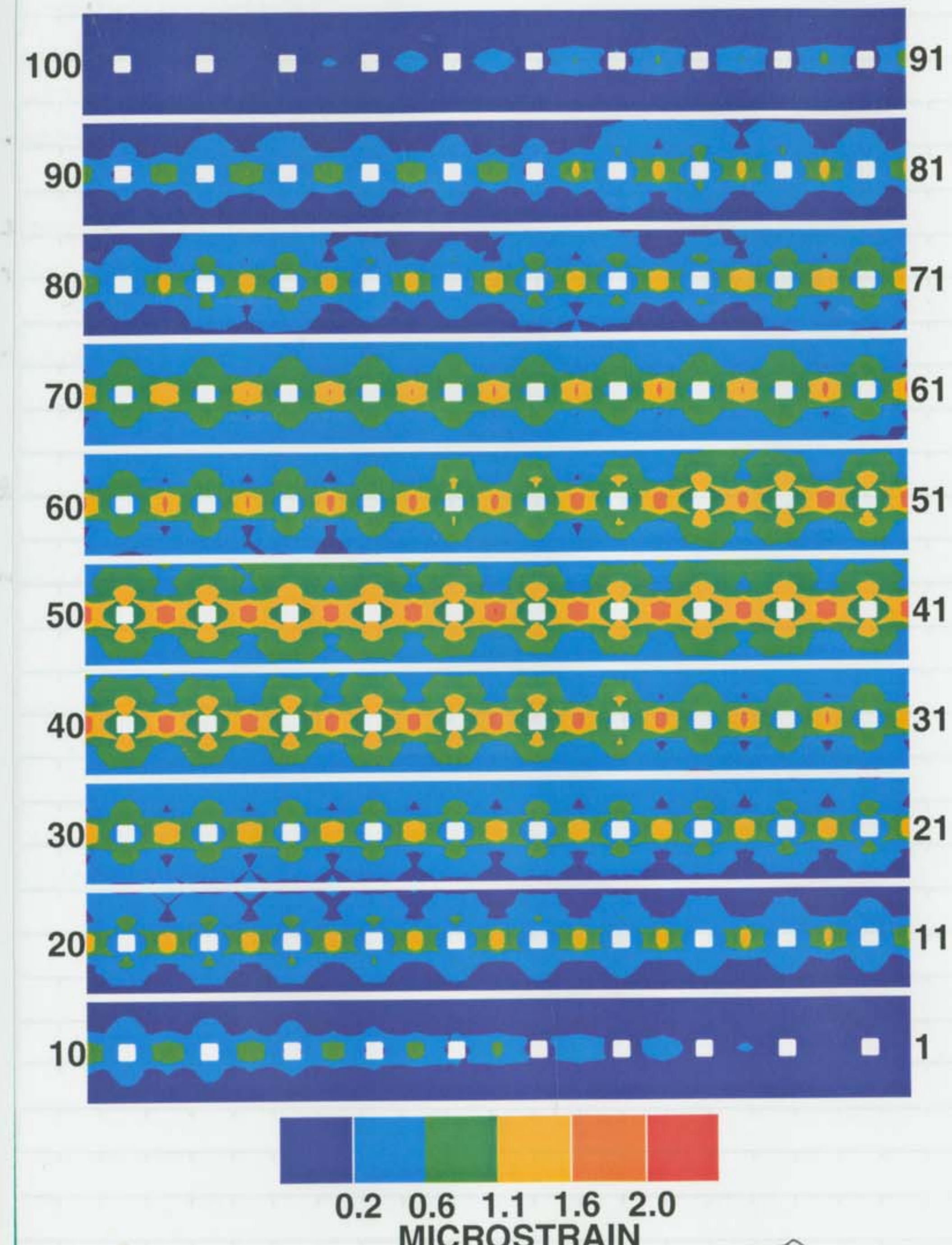
Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

Project No.
Book No. GW

TITLE

INELASTIC STRAIN DISTRIBUTION AT 150 YR
STIFF DRIFT SUPPORT

GW



$$\Gamma^N = \sum \sqrt{\frac{2}{3}\left[(\Delta\epsilon_{11})^2 + (\Delta\epsilon_{22})^2 + 2(\Delta\epsilon_{12})^2\right]}$$

and $\Delta\epsilon_{11}$, $\Delta\epsilon_{22}$, and $\Delta\epsilon_{12}$ are the components of inelastic strain increment and the summation indicates accumulation of the square-root term through the simulation time. Therefore, the distribution of Γ^N at the end of the 150-yr simulation time, for example, represents all inelastic processes that occurred in the model from zero time to end of 150 yr.

GW 12/14/98

GW

This figure shows distribution of Γ^N at 150 yr in the presence of stiff drift support.
(1) Γ^N attains max values at pillar centers and secondary maxima at roof and floor areas.
(2) Highest Γ^N values occur in areas of higher Q.
(3) Γ^N distribution strongly influenced by Q distribution.

See bottom of figure for definition of Γ^N.

This result from Case 1 model.

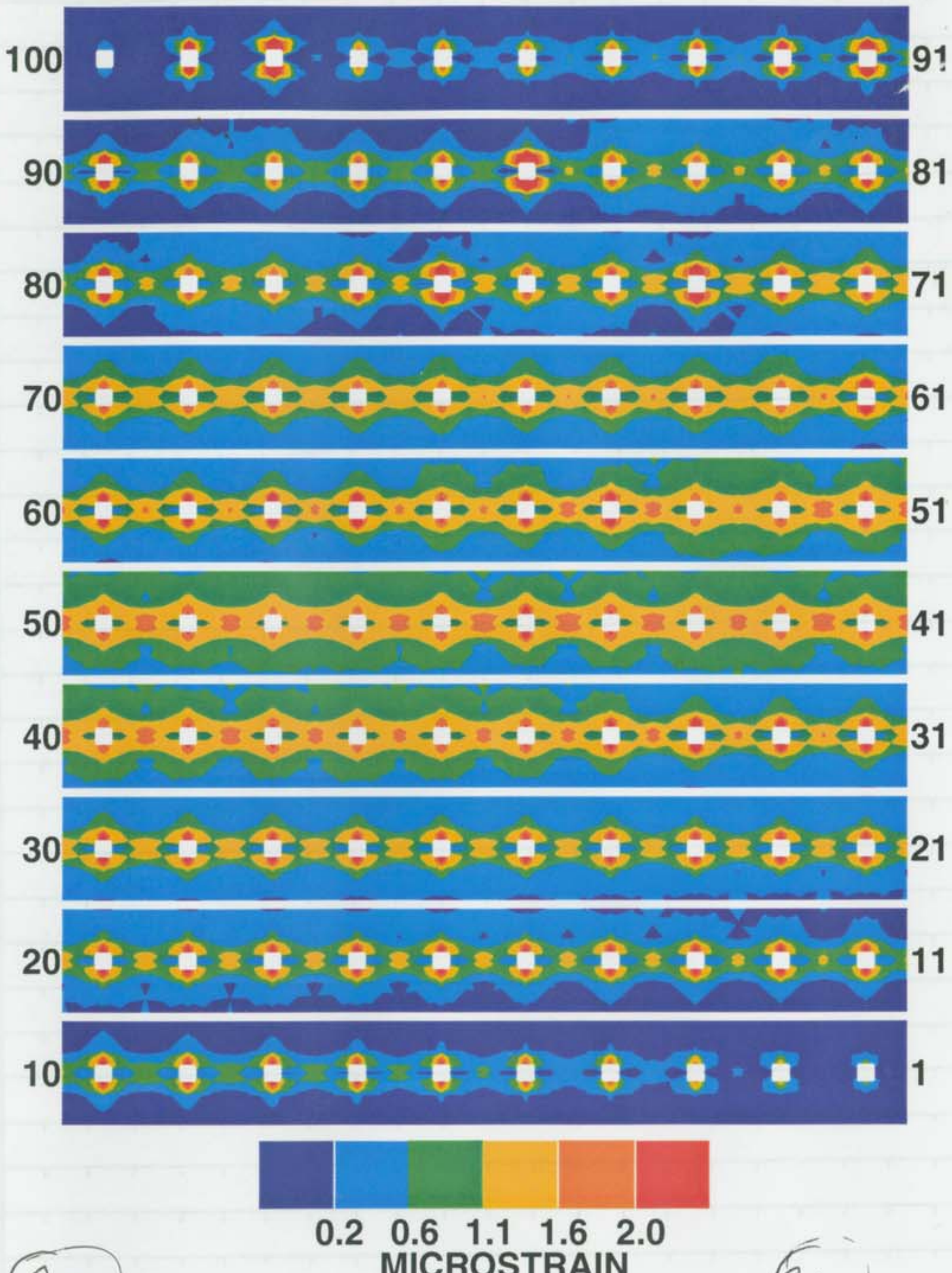
To Page No.

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

Project No.

Book No. GW TITLE

INELASTIC STRAIN DISTRIBUTION AT 150 YR DEGRADED DRIFT SUPPORT



(1) Support degradation causes increased Γ^N everywhere, especially in roof and floor areas.

(2) Γ^N increases more in area of lower Q (compare distributions near drift #98 for example)

(3). Loss of confinement near openings owing to degradation of support has more severe effects in areas of lower Q.

This result from Case 1 model.

GW GW

To Page No.

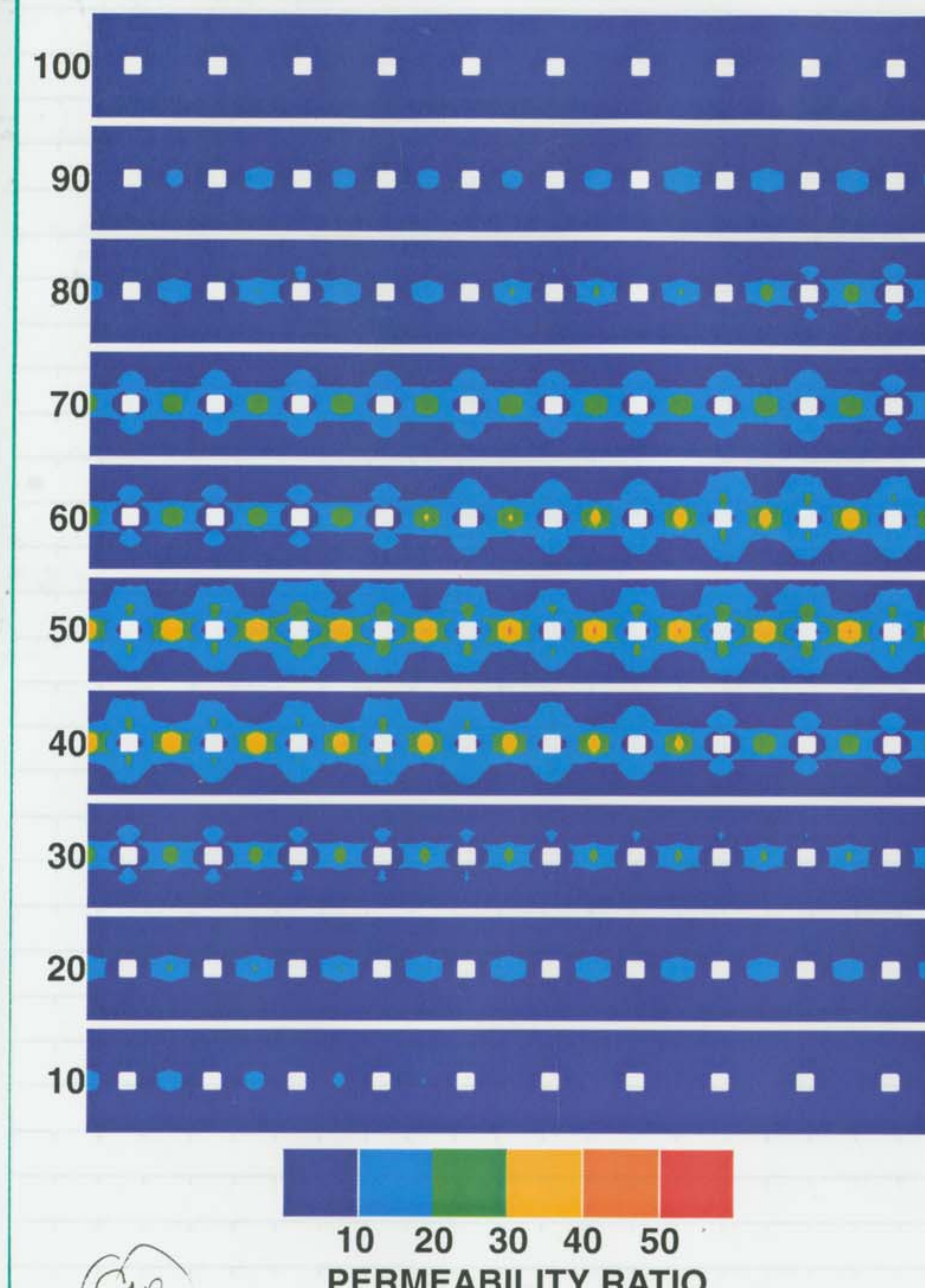
Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

TITLE GW

Project No.

Book No. GW

SHEAR-INDUCED PERMEABILITY CHANGE STIFF DRIFT SUPPORT



Distribution of R_k from Case 1 in the presence of stiff drift support

(1) R_k attains maximum at pillar centers with secondary maxima at roof and floor areas.

(2) R_k is higher in areas of higher Q.

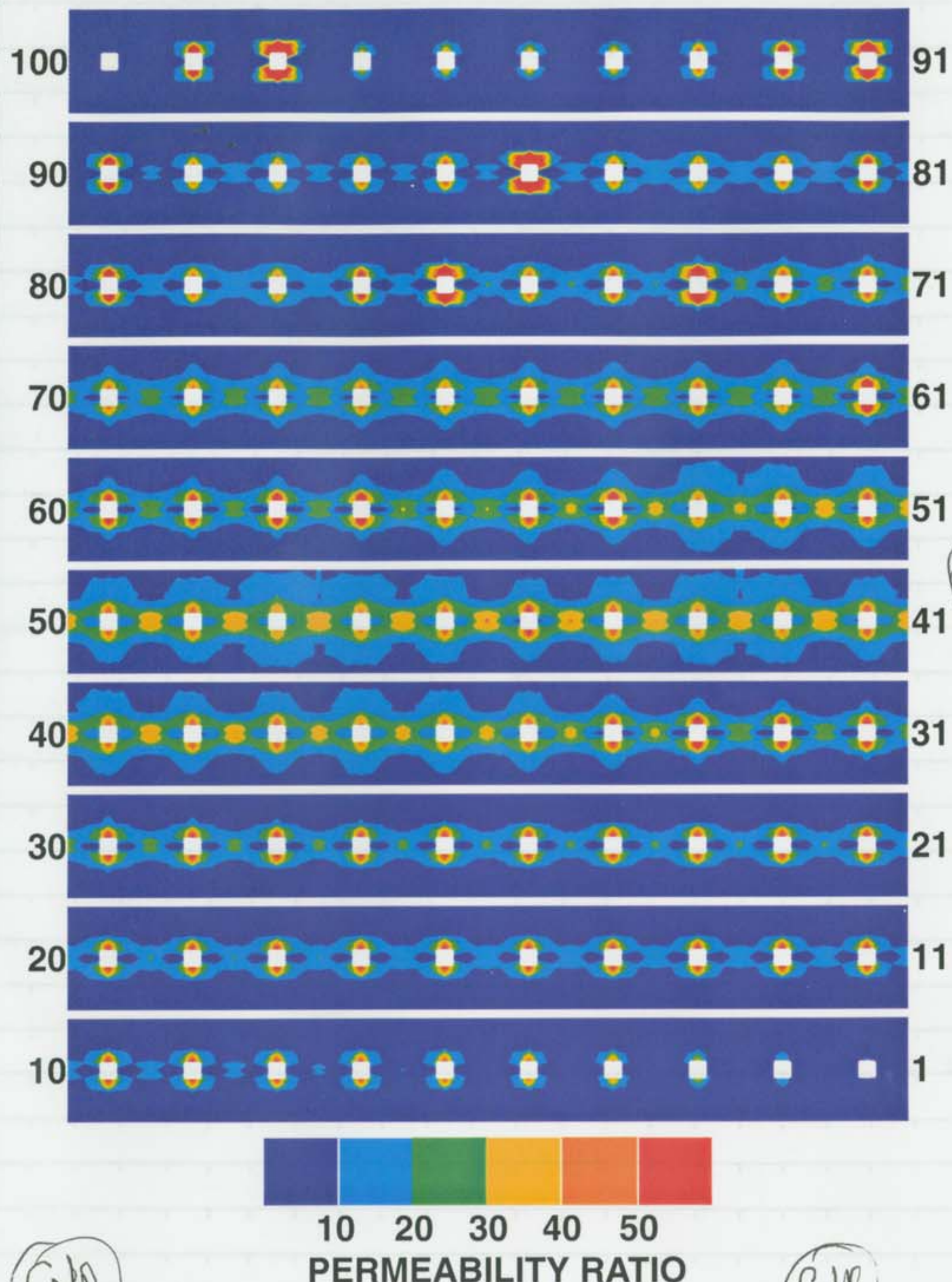
See p. 55 for definition of R_k and p. 70 for explanation of plot organization.

GW GW

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

SHEAR-INDUCED PERMEABILITY CHANGE DEGRADED DRIFT SUPPORT



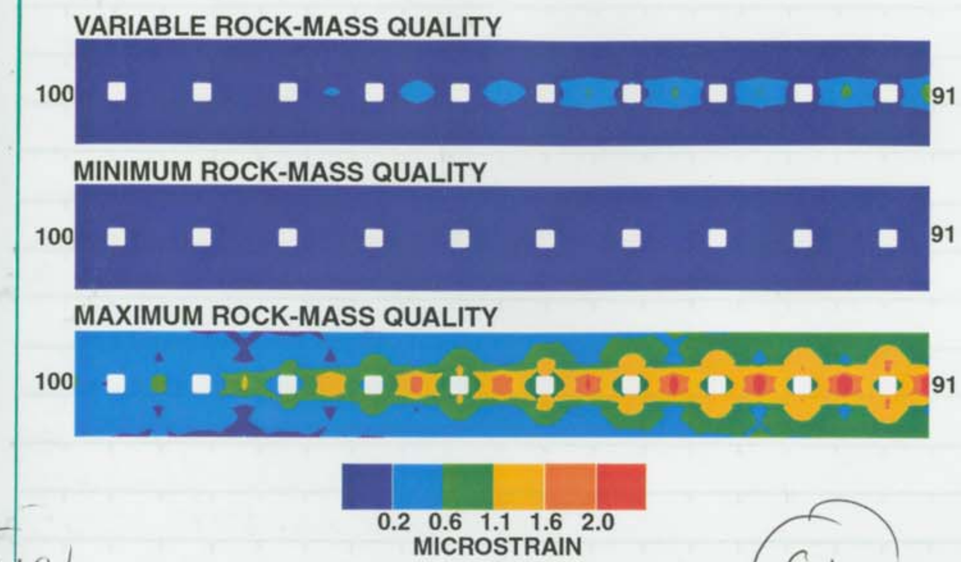
Distribution of R_k from Case 1 for degraded drift support.

(1) Support degradation causes increased R_k everywhere, especially in roof and floor areas.

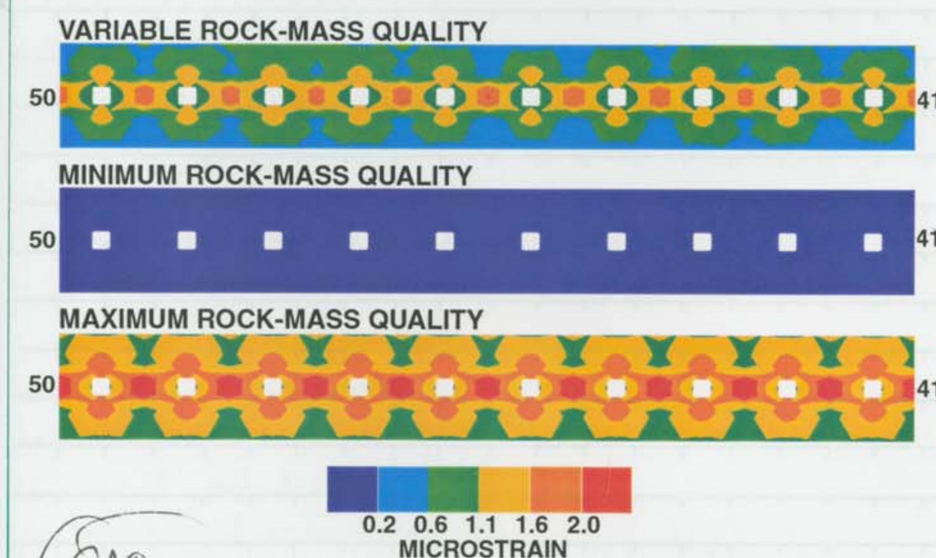
(2) Support degradation has more severe effects in areas of ~~higher~~ lower Q

12/14/98

INELASTIC-STRAIN DISTRIBUTIONS WITHIN 15 m ABOVE AND BELOW DRIFTS #91-100 (STIFF DRIFT SUPPORT)



INELASTIC-STRAIN DISTRIBUTIONS WITHIN 15 m ABOVE AND BELOW DRIFTS #41-50 (STIFF DRIFT SUPPORT)



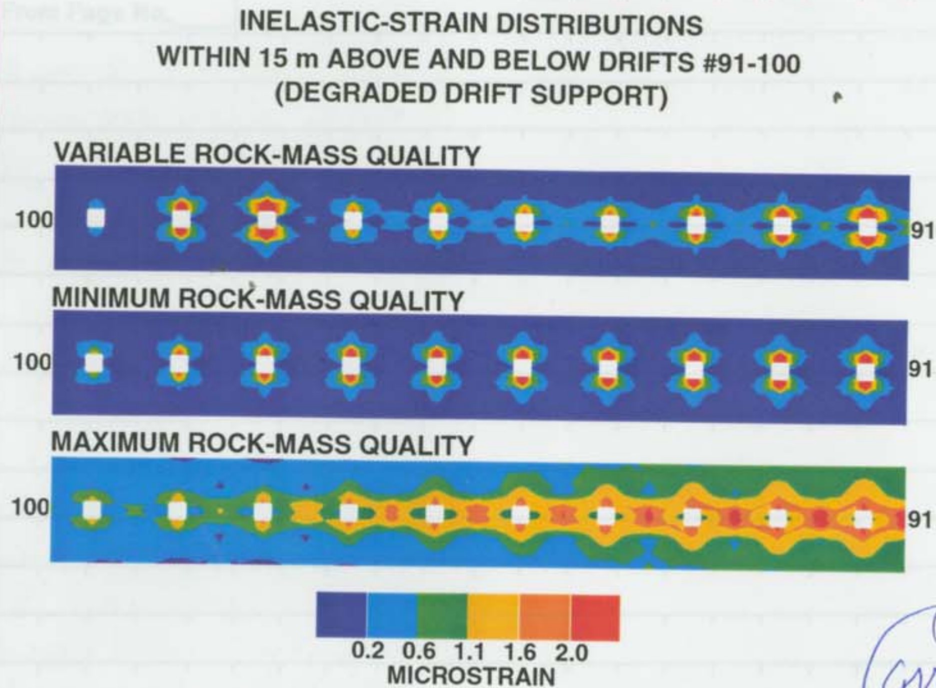
Comparison of results from Cases 1, 2, and 3 with stiff ground support. Top plot is for low Q area of Case 1 and bottom plot for high Q area.

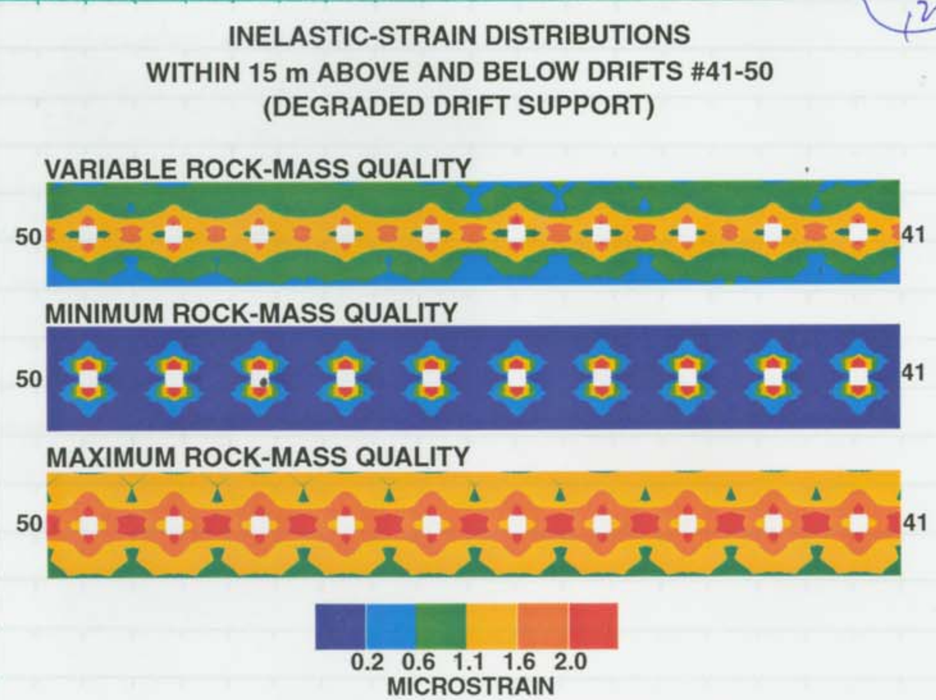
In the presence of stiff supports:

(1) Use of minimum Q in analysis would result in Γ^N values that are too small everywhere.

(2) Use of maximum Q in analysis would result in Γ^N values that are too large everywhere.

(3) If support load increases with Γ^N (to be investigated) then use of minimum Q would result in underestimate of support load whereas use of ~~the~~ maximum Q would result in overestimates.





Comparison of results from Cases 1, 2, and 3 with degraded drift support.

(1) Use of minimum Q everywhere would cause under-estimation of the extent of potential damage zones ~~around~~ in the roof and floor areas. The intensity of potential fracturing in inter-drift pillars is also likely to be underestimated in analyses based on ~~a~~ uniform minimum Q.

GW 12/14/98

On the other hand, use of uniform maximum Q will likely lead to underestimation of the ~~extent~~ of damage in the roof and floor areas, and over-estimation of the intensity of fracturing in the inter-drift pillars.

Pages 70-76 entries by GW 12/14/98

Pages 77-84 entries by GW 2/15/99

From Page No.____ Two drift-scale models are set up to examine the following:

(1) Shape and size of damaged zone (zone of high plastic strain in the roof area of opening)

(2) Distribution of contact pressure on drift lining

(3) Distribution of the permeability-change ratio R_k

The drift-scale models ~~will be~~ are synchronized with the repository-scale model in terms of (i) coordinates, (ii) spatial variation of mechanical properties, and (iii) Drift number. Results from the repository-scale model are used to set the boundary conditions for the drift-scale models.

GW 2/15/99

Two drift-scale models are analyzed:

(1) Drift #48 of repository-scale model. Boundary x coordinates (middle of the bounding pillars) are:

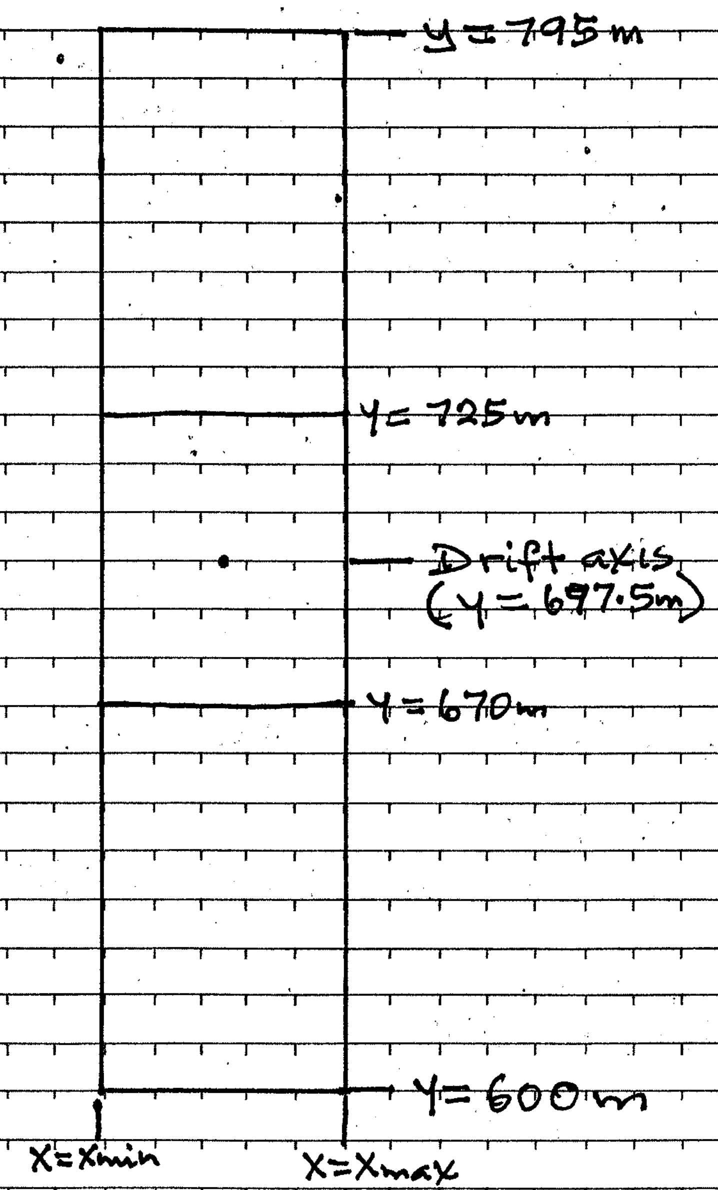
$$X_{min} = 28(100 - D_n) = 28 \times 52 \quad (D_n = 48)$$
$$= 1456 \text{ m}$$
$$X_{max} = 1456 + 28 = 1484 \text{ m}$$
$$X_0 = X \text{ at drift axis} = \tfrac{1}{2}(1456 + 1484) = 1470 \text{ m}$$

(2) Drift #85 of repository-scale model

$$X_{min} = 28(100 - 85) = 420 \text{ m}$$
$$X_{max} = 420 + 28 = 448 \text{ m}$$
$$X_0 = \tfrac{1}{2}(420 + 448) = 434 \text{ m}$$

From Page No.

Drift #48 is in the maximum Q area, and #85 in the minimum Q area, of the repository-scale model. Minimum Q actually occurs between drifts 98 and 99 (p.70) but these drifts lie in the zone of significant lateral temperature gradient and relatively low temperature. Drift #85 ~~was~~ (cf. p.70) was chosen to represent low-Q areas to avoid the effects of lateral temperature gradient and relatively low temperature on the calculated response.

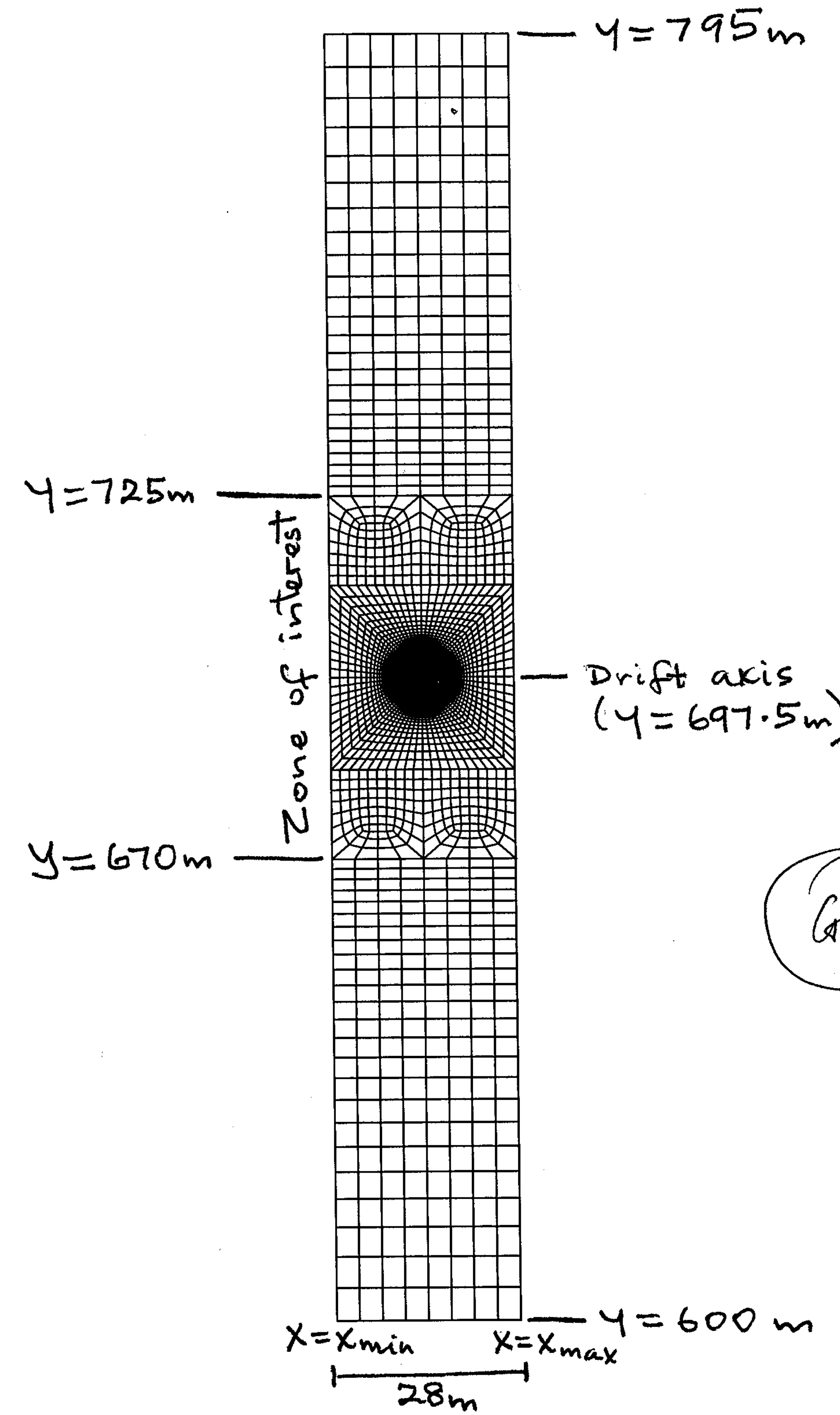


This figure illustrates schematically the domain of each drift-scale model. The domain extends about 200m vertically, i.e. ~100 m above and below the drift axis. Boundary conditions at the exterior boundaries are set using temperature and x and y displacement histories calculated in the repository-scale model. Such boundary conditions are applied using the ABAQUS SUBMODEL facilities.

Each drift is modeled ~~using~~ as a circular section with external diameter of 5.4m, including a 0.2m thick concrete lining.

To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

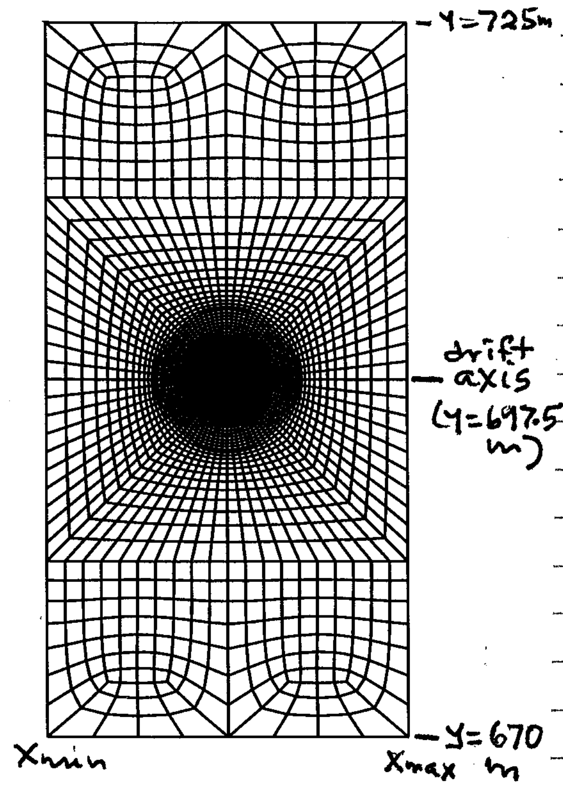


Finite element mesh used for drift-scale models. See p.80 for more details.

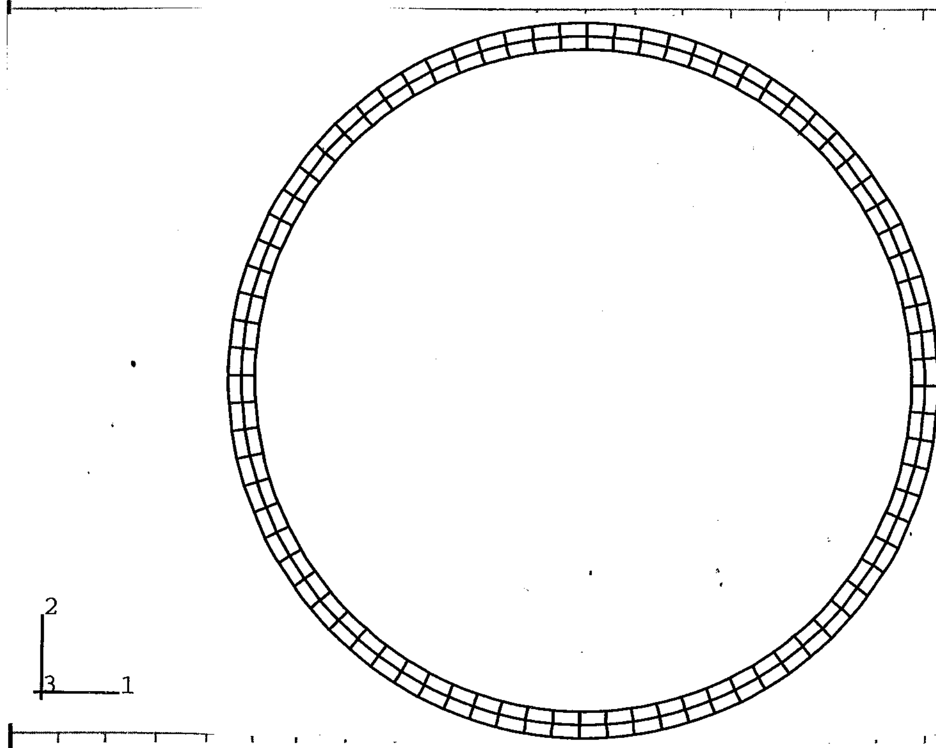
GW 2/15/99

To Page No.

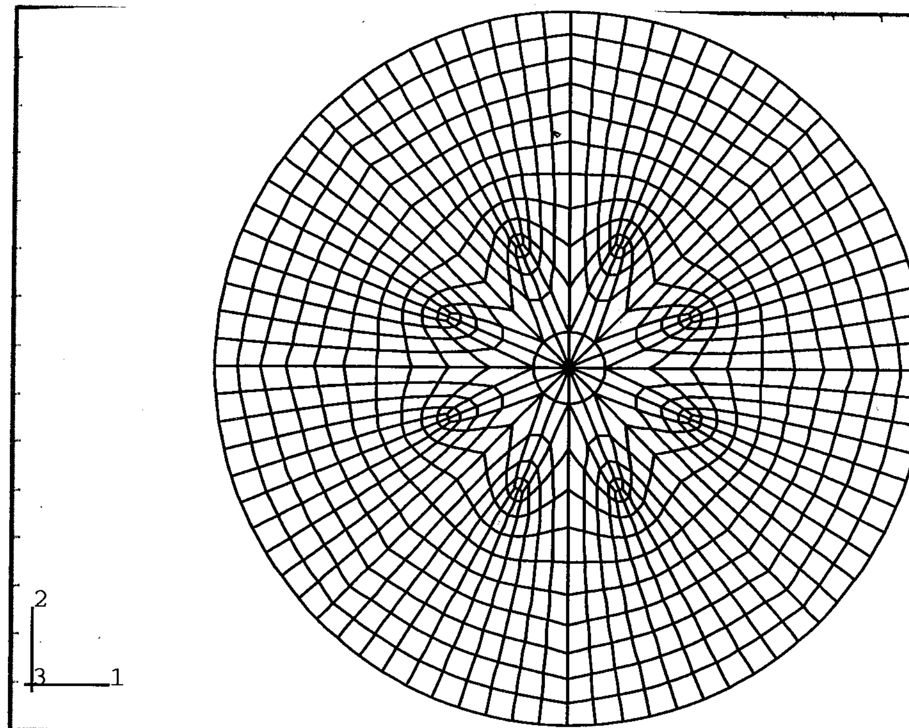
Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	



Finite element mesh within the zone of interest, which extends vertically 27.5 m above and below the drift axis. Materials within this zone, except the liner and tunnel section, are modeled as ~~lin~~. elastic-plastic using the Drucker-Prager strength criterion.



Finite element mesh used to model liner. The liner has internal and external diameters of 5.0 and 5.4 m, respectively. Pressure distribution at the exterior surface of the liner is monitored as part of the analysis.



Finite element mesh of the tunnel section. Material within this zone is treated as a uniform volumetric heat source in the thermal analysis. In the mechanical analysis, the tunnel-section elements are removed to simulate excavation. The interior surface of the liner coincides with the exterior surface of the tunnel section.

<u>Material Properties</u>

Zones: (1) Elastic-plastic zone (see p. 80).

GWO 2/15/99

(2) Elastic zone: includes the tunnel section, areas above and the upper elastic-plastic zone, and areas below the lower elastic-plastic zone (i.e., areas above y=725m and areas below y=670m on p. 78).

(3) Liner: modeled as linear elastic with concrete properties.

Rock mass thermal and mechanical properties are assigned the same values (some as functions of X) described on p. 18–23. The properties of concrete liner are assigned from literature as follows:

From Page No.____

Properties of Concrete

Density: 2100 kg/m³ [p.15 of reference (1) cited on p.2].

Thermal conductivity: 1.37 J/(s.m.K)

$= 4.32 \times 10^7$ J/(yr.m.K)

[p.15 of same reference]

Specific heat capacity = 880 J/(kg.K)

[p.15 of same reference]

Young's modulus: 2.76×10^4 MPa

Poisson's ratio: 0.25 ← Changed to 0.21

[p.8 of reference (3) cited on p.15]

Thermal expansivity: 9.9×10^{-6}/K

[p. 8 of same reference].

GN 2/15/99

Input Files

(1) Repository-scale model ~~to~~ m06 (same as m01 described on p.61) is used to generate histories of x and y displacements required to set drift-scale model boundary conditions.

To Page No.____

From Page No.____

Input files for model m06: (Global model)

m06.inp
allNodes.def
mElem01.def and auxiliary files listed on p.61
d1GmNodes.def defines driver node set for model d1
d2GmNodes.def defines driver node set for d2

(2) Drift-Scale Models (submodels).

Model d1 — Drift #48

2.1	d1t.inp	Thermal analysis input files
2.2	d1Nodes.def	Node definitions.
2.3	nodeSets.def	Node set definitions.
2.4	drivenNodes.def	Node sets that get temperature and displacement histories from model m06.
2.5	tElems.def	Element connectivity for thermal model definitions
2.6	elemSets.def	Element set definitions
2.7	contacts.def	Defines liner-to-rock contact sets.
2.8	tProps.def	Thermal property definitions
2.9	iniTemp.def	Initial temperature definitions
2.10	cDrftSrc.def	Heat source definition
2.11	mElems.def	Element connectivity for mechanical model
2.12	mProps.def	Mechanical property definitions
2.13	ndxFld.def	Defines a field variable assigned value of x-coordinate at each node.
2.14	d1m.inp	Mechanical analysis input files
2.15	d1EMat.def	Young's modulus and Poisson's ratio
2.16	d1Friction.def	Drucker-Prager friction properties
2.17	d1Cohesion.def	Drucker-Prager cohesion properties

To Page No.____

Project No.______
Book No.______ TITLE______

From Page No.____

Model ~~Ø~~ d2 Drift #85

2.18	d2t.inp	Thermal analysis input file
2.19	d2Nodes.def	Node definitions.
2.20	d2ndxFld.def	Defines a field variable assigned value of x-coord at every node.
2.21	d2m.inp	Mechanical analysis input
2.22	d2EMat.def	Young's modulus & Poisson's ratio
2.23	d2Friction.def	Drucker-Prager friction properties
2.24	d2Cohesion.def	Drucker-Prager cohesion properties
2.25	d2mProps.def	Mechanical properties definitions

The following input files ~~are~~ listed for model d1 are also used by model d2:

2.3 thru 2.11

GW 2/15/99

Pages 77-84 entries by GW 2/15/99

To Page No.______

Witnessed & Understood by me, | Date | Invented by | Date
Recorded by

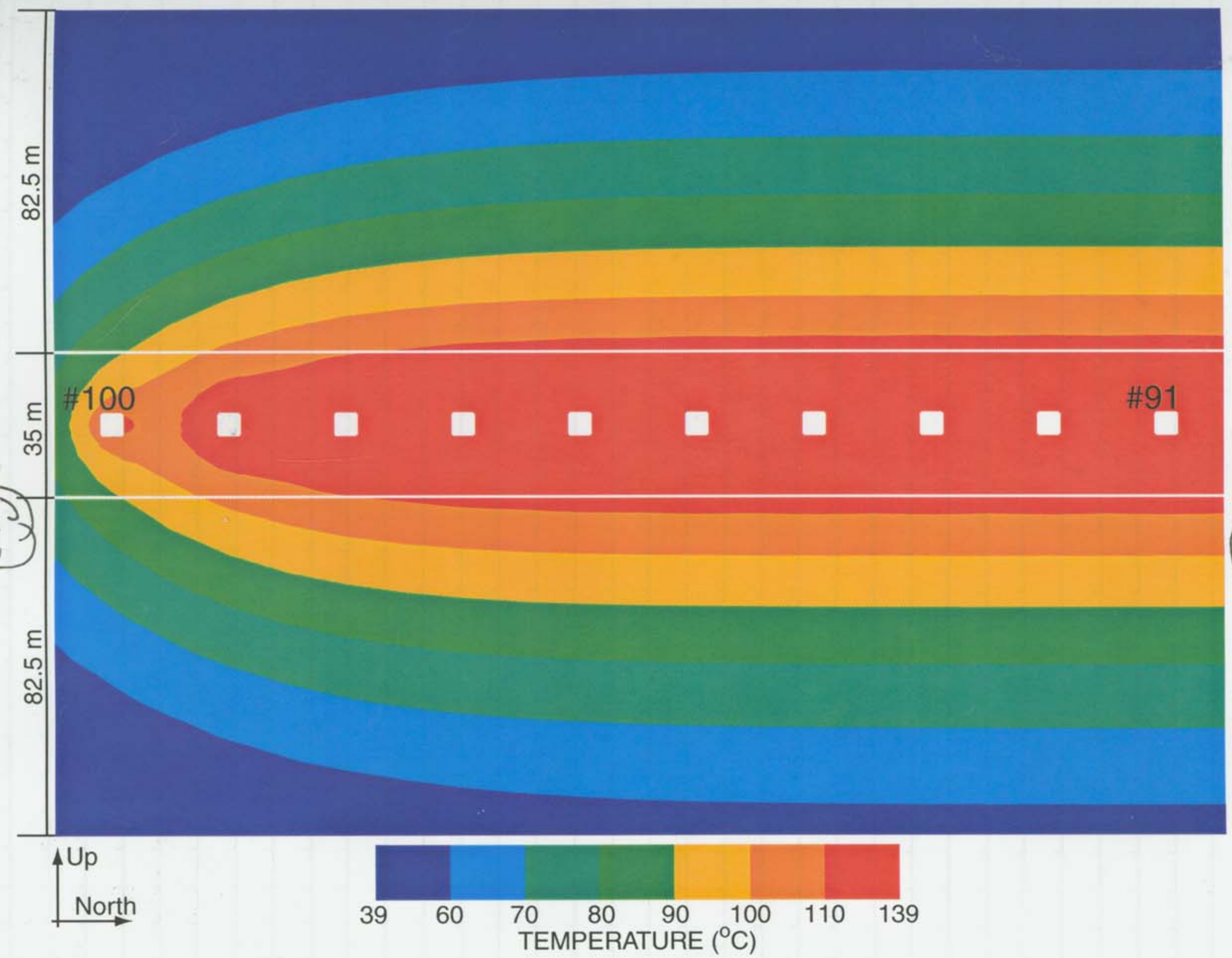
TITLE March 15 1999

From Page No.

Project No.

Book No.

pages 85-95 entries 3/15/99



Temperature distribution at 150 yr, over a 200-m high section of the repository-scale model from Drift #100 through Drift #91.

Witnessed & Understood by me,

Date

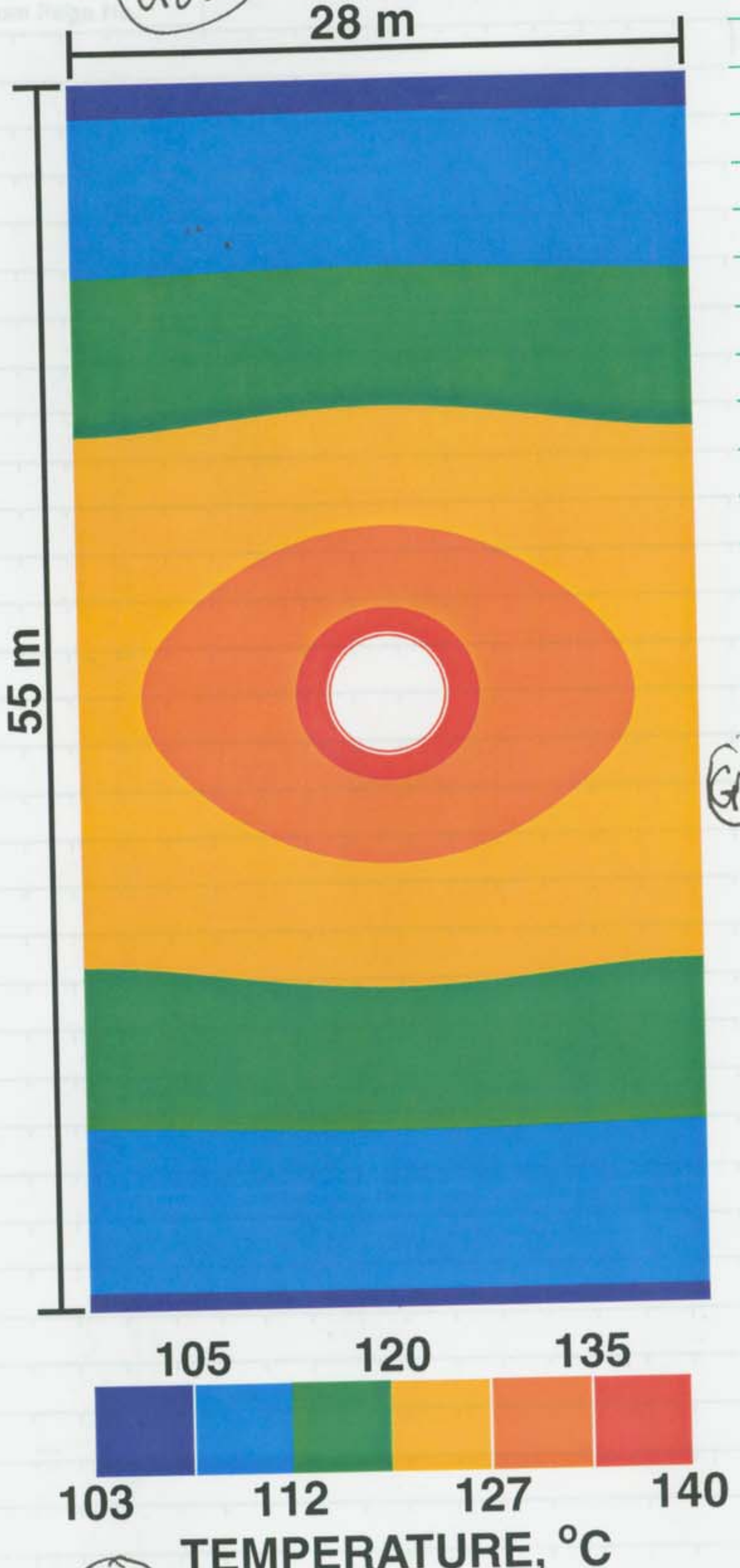
Invented by

Recorded by

Date

To Page No.

GWO



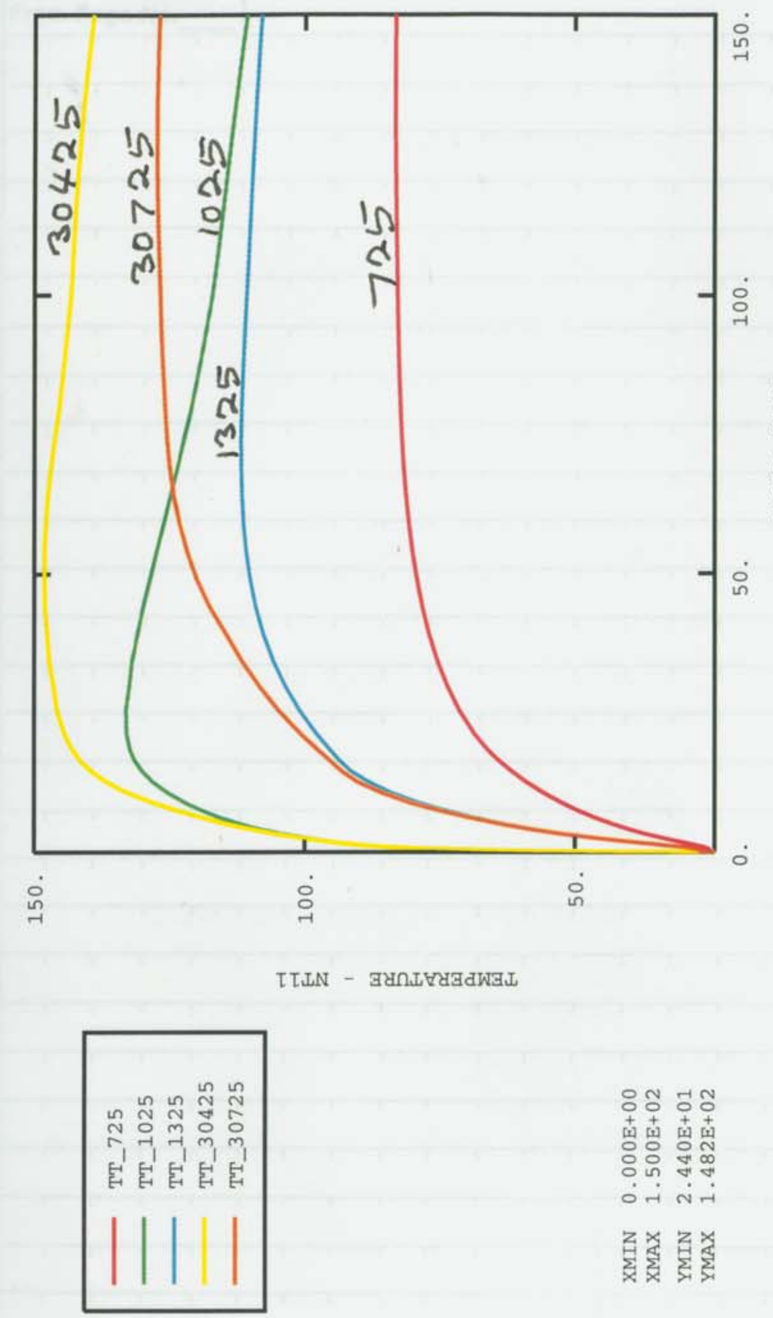
GWO

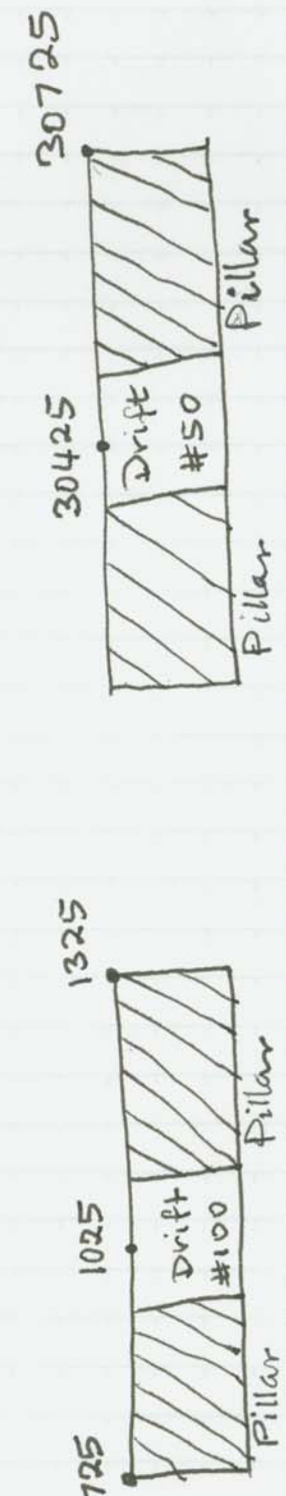
Temperature distribution at 150 yr within the 55-m high zone of interest ~~un~~ in the drift-scale models. Temperatures range from about 103°C to about 140°C. The 55-m high zone centered on the drift lies within the two inner contour zones of the figure on p. 86, in which temperatures also range from about 100°C to about 140°C.

GWO The two sets of results (p. 86 and this page) indicate consistency between the repository-scale and drift-scale models.

Witnessed & Understood by me, | Date | Invented by | Recorded by | Date

GWO



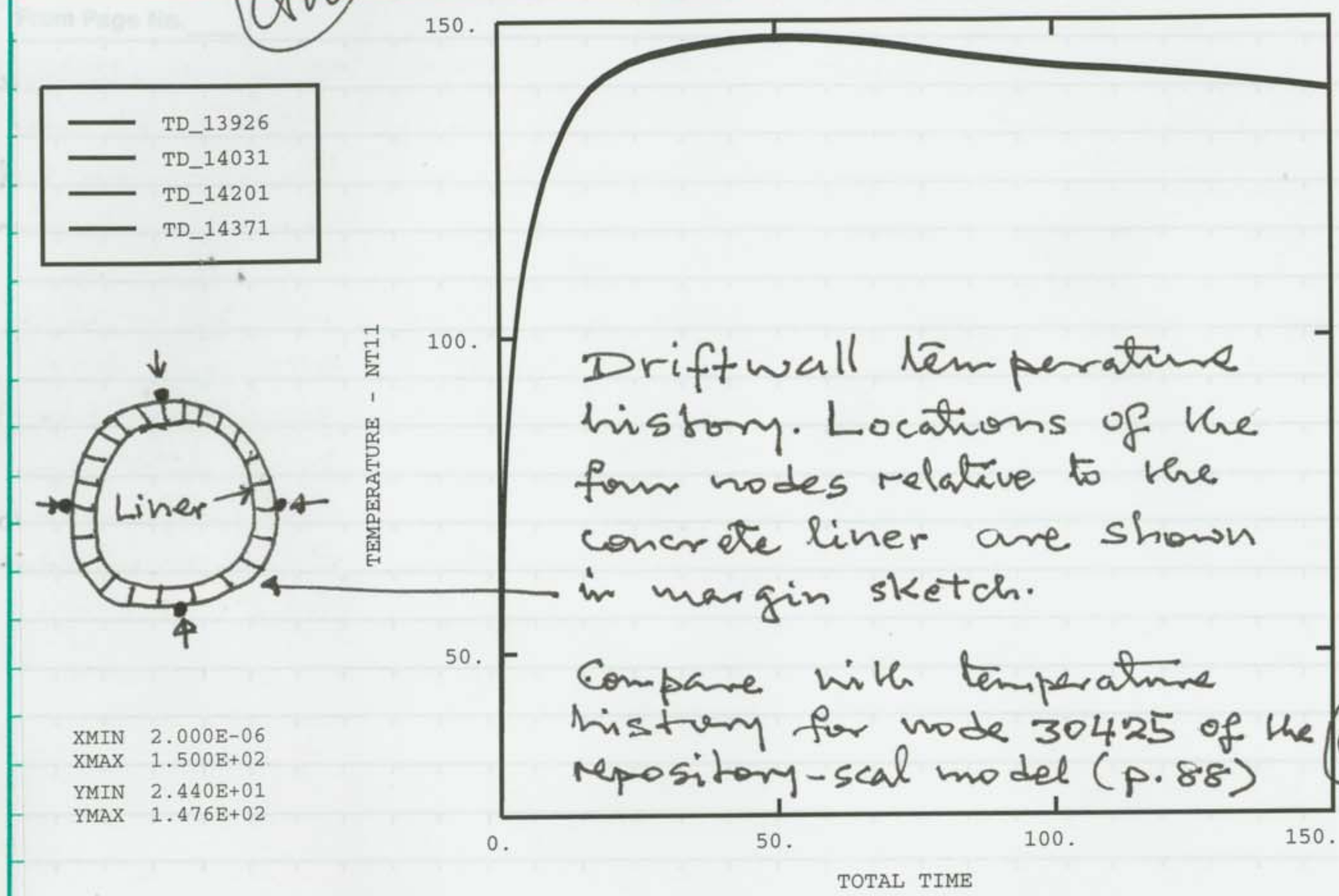


Temperature histories at 5 nodes from the repository-scale model. The locations of the nodes relative to Drifts #100 and #50 are illustrated. Drift #100 is an end drift, whereas #50 is an interior drift.

GWO 3/15/99

Witnessed & Understood by me, | Date | Invented by | Recorded by | Date

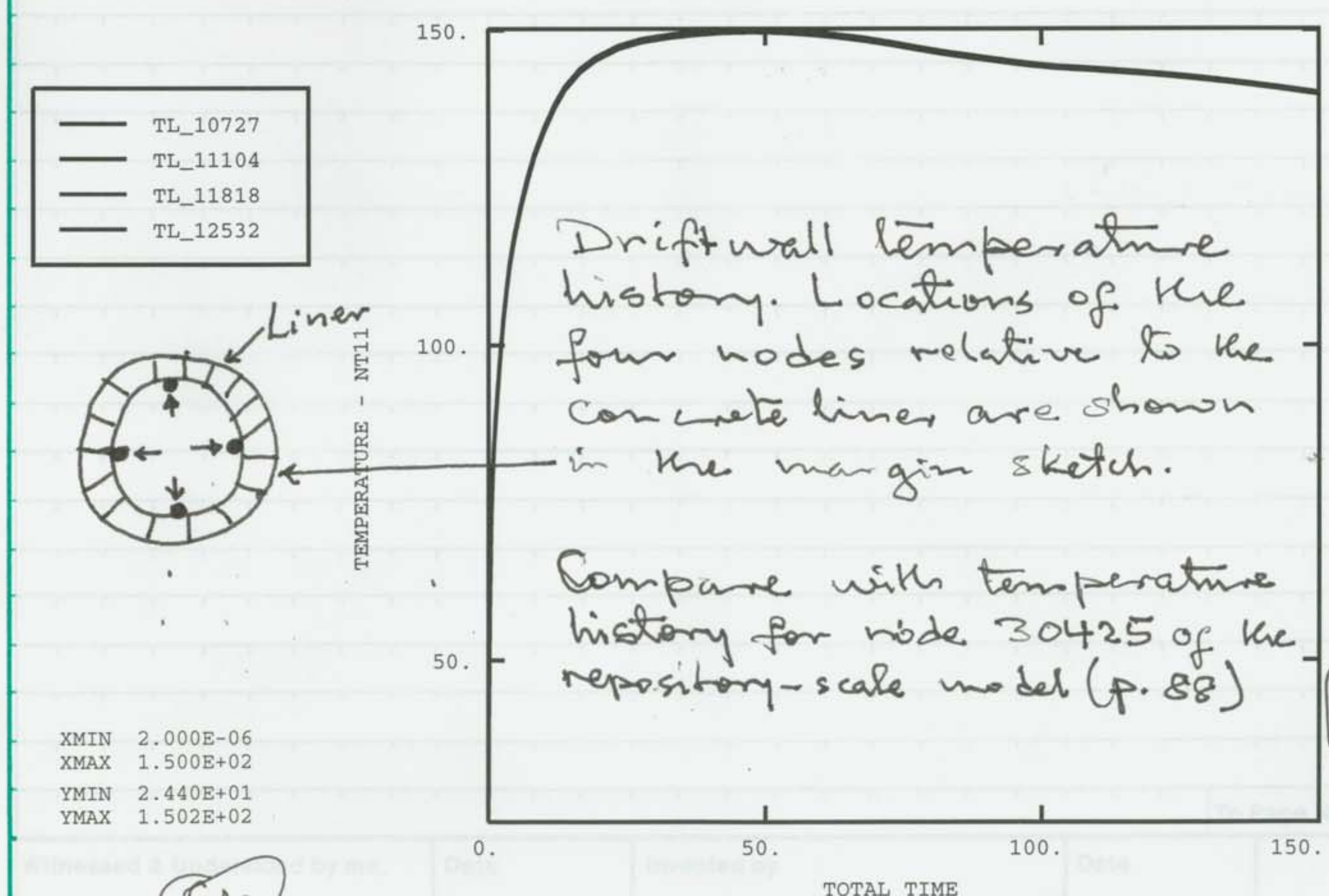
GW



Driftwall temperature history. Locations of the four nodes relative to the concrete liner are shown in margin sketch.

Compare with temperature history for node 30425 of the repository-scal model (p. 88)

GW



Driftwall temperature history. Locations of the four nodes relative to the concrete liner are shown in the margin sketch.

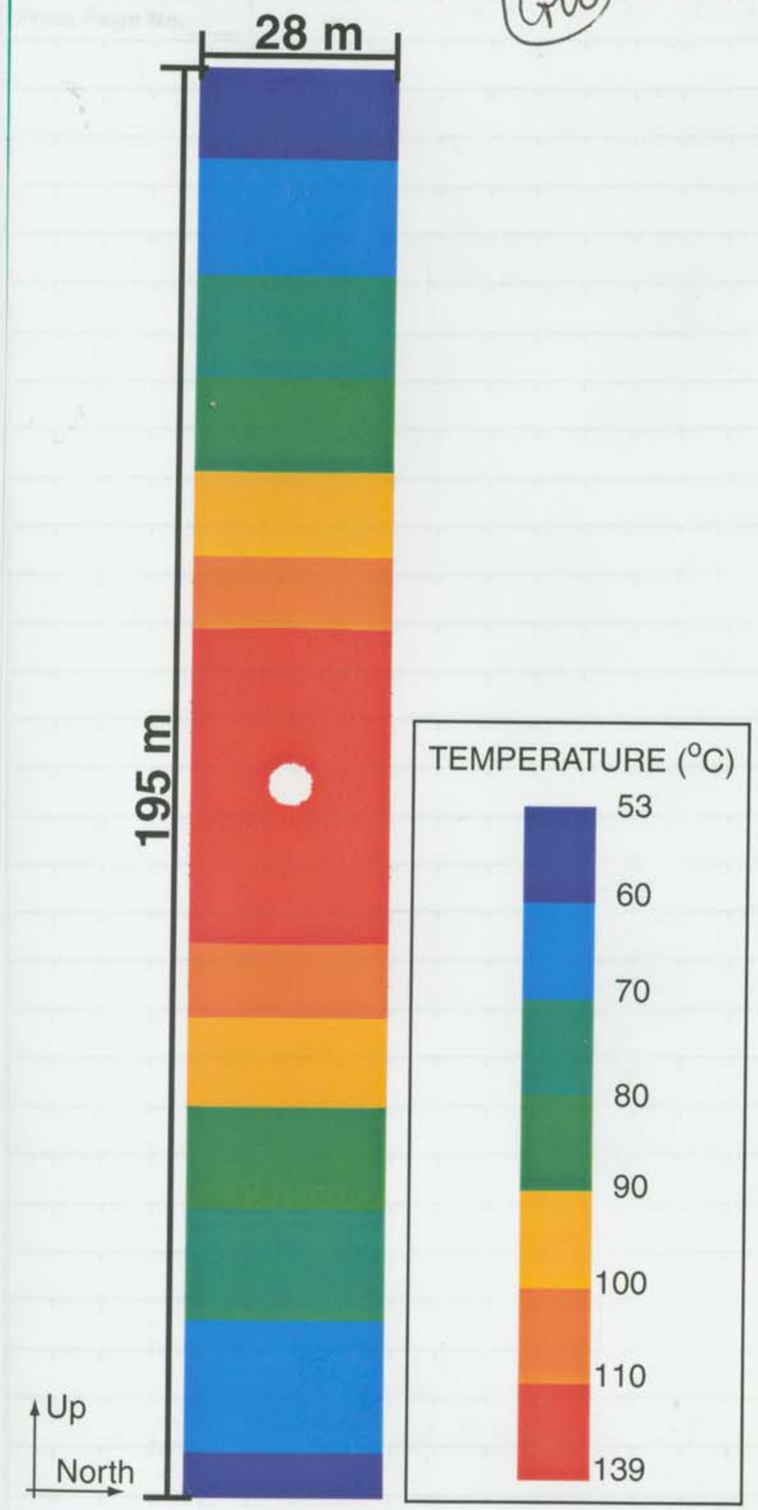
Compare with temperature history for node 30425 of the repository-scale model (p. 88)

GW

GW

Recorded by

GW



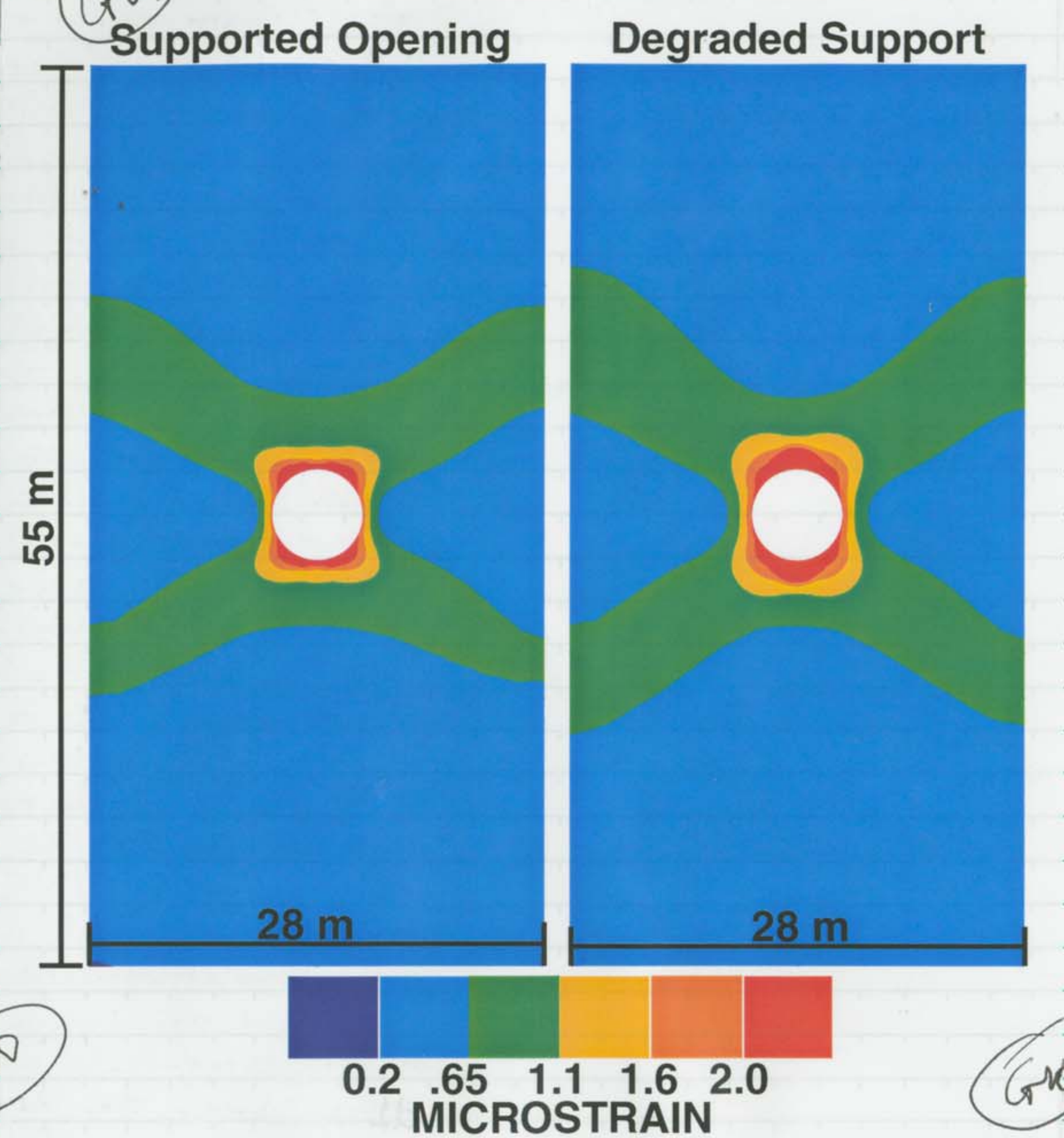
Temperature distribution at 150 yr over the full drift-scale model. The distribution is plotted following the same format used for the repository-scale model distribution shown on p. 85. Both plots cover a vertical interval of about 200 m centered on the drift axis. The temperature distribution from the drift-scale model should be closely similar to the repository-scale distribution ~~for~~ over a vertical (28-m wide) strip through one of the interior drifts, such as Drift #91 (p. 85). The two distributions appear to be identical. The only difference is that the lowest temperature is 53°C in the drift-scale model and 39°C in the repository-scale model. The lower minimum temperature in repository-scale model occurs near the top-south end of the model and is due to end cooling that is not accounted for in the drift-scale model.

GW

Project No.
Book No.

TITLE Γ^N Distributions From D1 Model

From Page No.



Inelastic strain (Γ^N) distributions from d1 (this page) and d2 (next page) models.

(1) Magnitudes of Γ^N in the pillars are higher in model d1 (highest Q area) than in model ~~d2~~ d2 (lowest Q area).

(2) Magnitudes of Γ^N close to the opening are higher in model d2 (lowest Q area) than in model d1 (highest Q area).

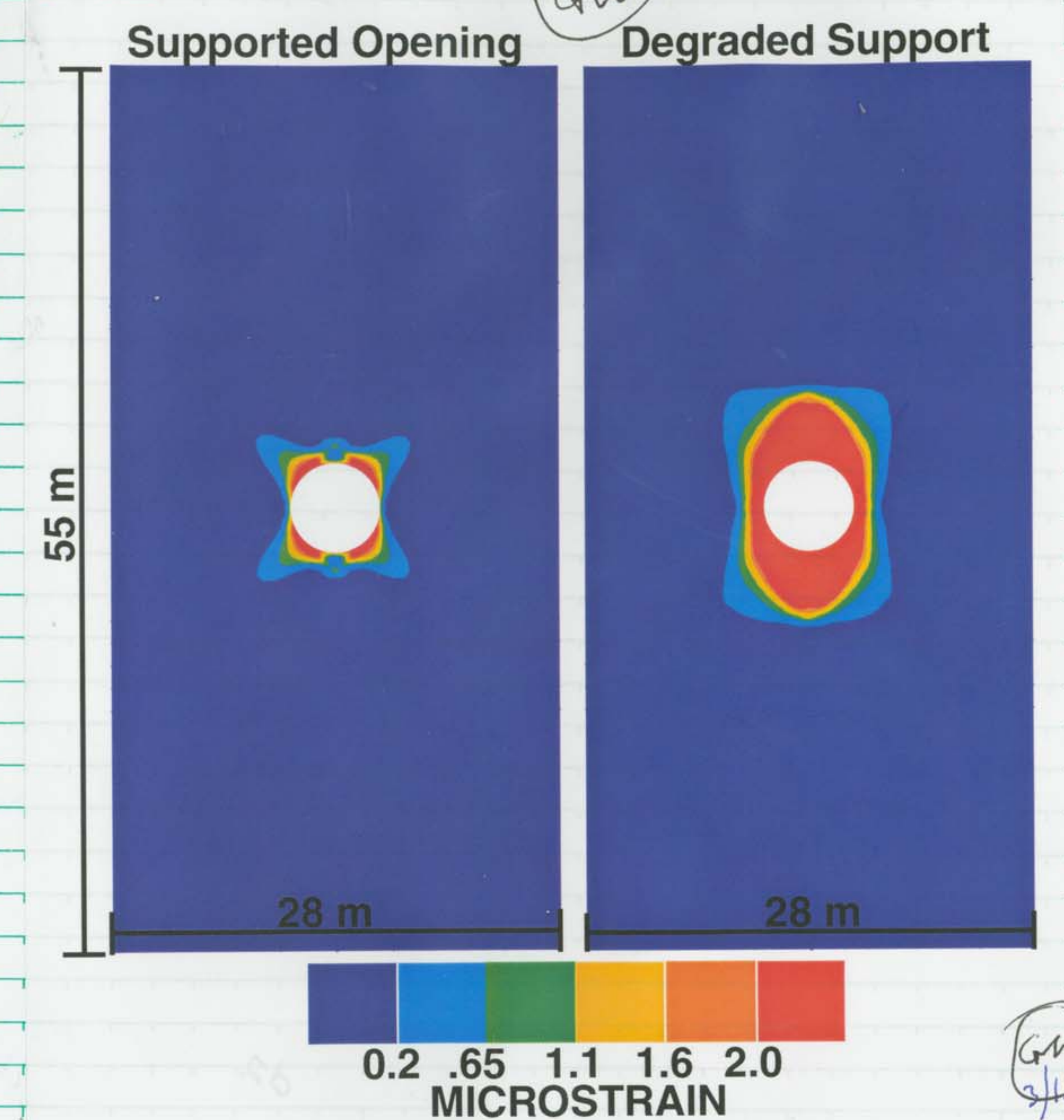
To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

TITLE Γ^N Distributions From D2 Model

Project No.
Book No.

From Page No.



GW 3/15/99

(3) Support degradation causes increased magnitudes of Γ^N everywhere, but much more in lowest-Q areas than in highest-Q areas.

(4) The high Γ^N values close to the opening in lowest-Q areas may indicate collapse of the opening in such areas.

(5) The stiffness of ~~the~~ concrete lining, which is greater than rock stiffness in lowest-Q areas (model d2) but smaller than rock in highest-Q areas (model d1) may have an effect on the Γ^N distribution.

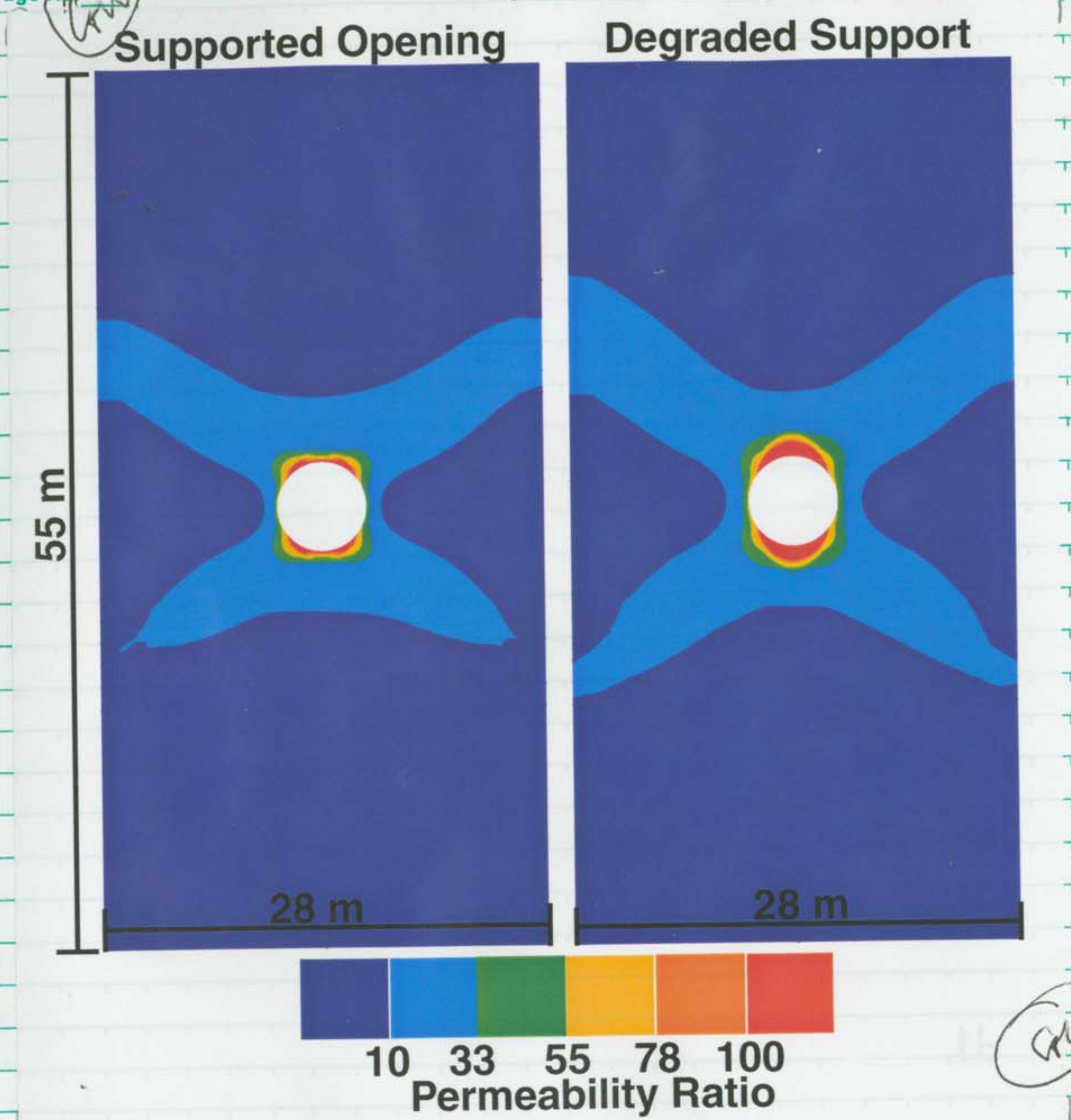
To Page No.

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

Project No.______
Book No.______

TITLE Permeability Changes from D1 Model

From Page No.



GND 3/15/99

Distributions of permeability-change ratio R_k (see p. 53) from model d1 (this page) and model d2 (next page).

(1) R_k distributions are similar to the corresponding distributions of Γ^N.

(2) R_k values in lowest-Q areas (model d2) are concentrated in the roof and floor areas, whereas ~~changes~~ permeability changes extend more

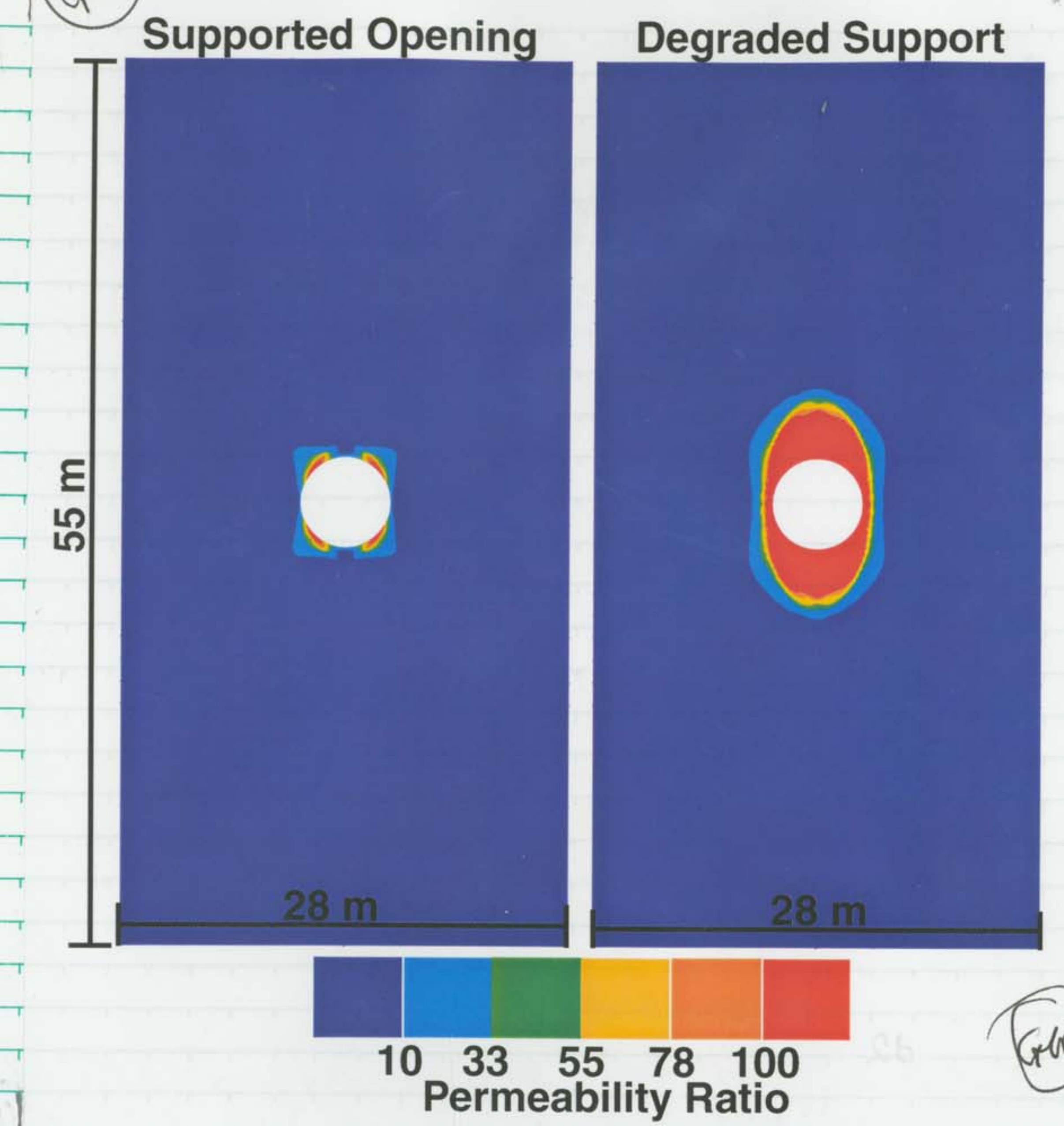
To Page No.

Witnessed & Understood by me, GND 3/15/99 | Date | Invented by | Date | Recorded by

TITLE Permeability Changes from D2 Model

Project No.______
Book No.______

From Page No.



GND

into the pillars in highest-Q areas.

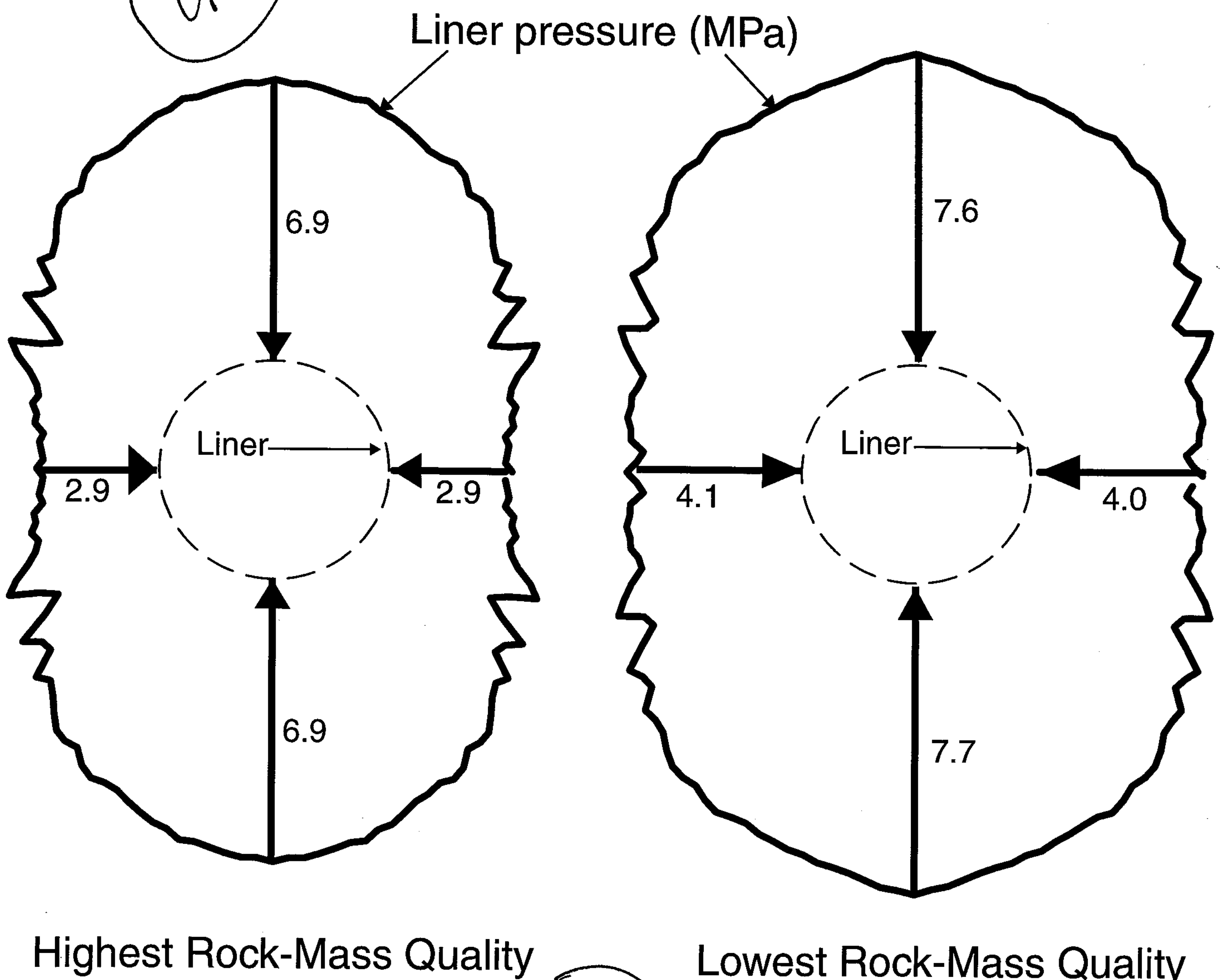
(3) The zones of relatively high R_k values near the roof and floor may not be significant because the rock within such zones may have collapsed into the openings.

(4) Support degradation has more severe effects on R_k distributions in the lowest-Q areas than in the highest-Q areas.

To Page No.

Witnessed & Understood by me, | Date | Invented by | Date | Recorded by

GW



GW

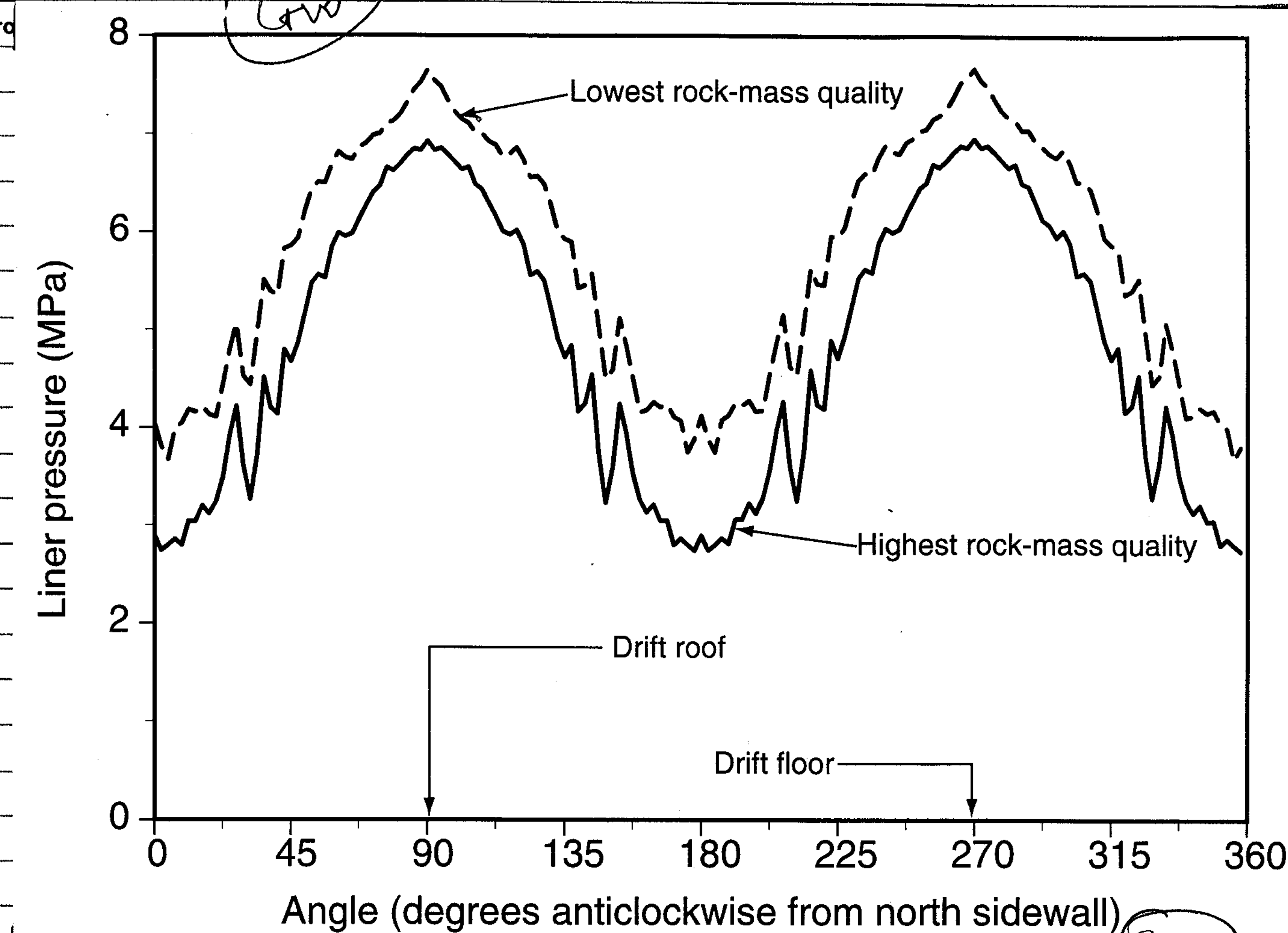
Distributions of liner pressure at 150 yr from models d1 (highest-Q area) and model d2 (lowest-Q area). Values of liner pressure are plotted as proportional radial distance from the liner on this page and as a function of angular distance on the next page.

(1) Liner pressure is generally higher in the lowest-Q area than in the highest-Q area.

(2) Liner pressure is greater in the roof and

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

GW



GW

floor of the liner than in the sidewalls. So potential liners may be subjected to nonhydrostatic loading.

(3) The liner pressures plotted here may have been influenced by the frictional properties applied to the rock-liner interface in the model. The liner was modeled as fully bonded to the rock in both model d1 and model d2.

(4) Possible effects of the applied rock-liner interface model will be investigated further.

Pages 85–95 entered by GW 3/15/99

GW 3/15/99

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	

Project No.______

Book No.______ TITLE______

From Page No.____

March 31 1999

Notes on this project are continued in CNWRA Scientific Notebook Number 321

This notebook (No. 263) is now closed.

3-31-99

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.

9-26-2000

MGFE

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

To Page No.______

Witnessed & Understood by me,	Date	Invented by	Date
		Recorded by	