

SAND--90-2542

DE92 017281

SAND90-2542
Unlimited Release
Printed December 1991

NORIA-SP—A Finite Element Computer Program for Analyzing Liquid Water Transport in Porous Media

P. L. Hopkins and R. R. Eaton

Fluid, Thermal, and Structural Sciences Department
Sandia National Laboratories
Albuquerque, New Mexico 87185

N. E. Bixler

Nuclear Energy Sciences and Materials Technology Department
Sandia National Laboratories
Albuquerque, New Mexico 87185

Abstract

NORIA-SP is a finite element computer program that solves a single nonlinear, parabolic, partial differential equation. The equation describes the transport of liquid water through partially saturated porous media. The numerical procedure uses the standard Galerkin finite element method to handle spatial discretization of two-dimensional domains with either planar symmetry or axisymmetry. Time integration is performed by a second-order predictor-corrector scheme that uses error estimates to automatically adjust time-step size to maintain uniform local time truncation error throughout the calculation. Thus, the user is only required to select an initial time-step size. Most material properties can be set either to constant values or defined as functions of the dependent and independent variables in user-supplied subroutines. This report is intended primarily as a user's manual but also briefly describes the theory of liquid transport in porous media and the numerical procedure used in NORIA-SP.

MASTER

25

The work contained in this report pertains to WBS Element 1.2.1.4.9.

Contents

Preface	vi
1 Introduction	1
2 Theory and Mathematical Model	2
2.1 Background	2
2.2 Conservation Equation	2
2.3 Transport Equation	3
2.4 Composite Fracture/Matrix Model	4
3 Galerkin Finite Element Formulation	6
3.1 Weak Form of the Partial Differential Equations	6
3.2 Finite Element Basis Functions	7
3.3 Galerkin's Method	10
3.4 Boundary Conditions	12
4 Time Integration Scheme	13
4.1 Strategy	13
4.2 Adams-Bashforth Predictor	13
4.3 Trapezoid-Rule Corrector	14
4.4 Newton Iteration Procedure	15
4.5 Automatic Time-Step Selection	16
4.6 Start-Up Procedure	17
5 Program Description	18
5.1 Organization	18
5.2 Mesh Generation	18
5.3 Boundary and Initial Conditions	20
5.4 Solution Procedure	21
5.5 Calculation of Derived Quantities	21
5.6 Plotting	21
6 Input Guide	22
6.1 Introduction	22
6.2 Header Card	23
6.3 SETUP Command Card	23
6.4 FORMKE Command Card	35
6.5 VELOCITY Command Card	35
6.6 OUTPUT Command Card	36
6.7 UNZIPP Command Card	37
6.8 RESTART Command Card	39

6.9	Program Termination Command Card	40
6.10	Input Deck Sequence	43
6.11	User Subroutines	44
6.12	Initial Conditions	48
6.13	Error Messages	49
6.14	Computer Requirements and Control Cards	52
7	Sample Calculations	54
7.1	Sample Problem 1	54
7.2	Sample Problem 2	62
8	Glossary	78
9	References	80
10	Appendix A	82
11	Appendix B	83

Figures

3.1	Finite Element Discretization of a Region	7
3.2	Quadrilateral Elements	9
6.1	Nomenclature for Generation of Nodal Points	29
6.2	Numbering of Element Nodes and Sides	32
7.1	Comparison of NORIA-SP Results and Analytical Solution for Fleming's Problem	63

Preface

A family of finite element computer programs has been developed at Sandia National Laboratories (SNL) that includes COYOTE (Gartling, 1982), NACHOS (Gartling, 1978), MARIAH (Gartling and Hickox, 1980), SAGUARO (Eaton *et al.*, 1983), NORIA (Bixler, 1985) and most recently, NORIA-SP. The original NORIA code solves a total of four transport equations simultaneously: liquid water, water vapor, air, and energy. Consequently, use of NORIA is computer-intensive. Since many of the applications for which NORIA is used are isothermal, we decided to “strip” the original four-equation version, leaving only the liquid water equation. This single-phase version is NORIA-SP. This manual includes much, but not all, of the background information included in the original NORIA manual. However, the primary intent of this document is to provide the user of NORIA-SP an accurate user’s manual. Consequently, the reader should refer to the NORIA manual if additional detail is required regarding the equation development and finite element methods used.

The single-equation version of the NORIA code (NORIA-SP) has been used most frequently for analyzing various hydrological scenarios for the potential underground nuclear waste repository at Yucca Mountain in western Nevada. These analyses are generally performed assuming a composite model to represent the fractured geologic media. In this model the material characteristics of the matrix and the fractures are area weighted to obtain equivalent material properties. Pressure equilibrium between the matrix and fractures is assumed so a single conservation equation can be solved. NORIA-SP is structured to accommodate the composite model. The equations for water velocities in both the rock matrix and the fractures are presented. To use the code for problems involving a single, nonfractured porous material, the user can simply set the area of the fractures to zero.

1 Introduction

The calculation of transient flow in porous media is of interest in radioactive waste disposal. The mathematical equations that describe flows in porous media are generally highly nonlinear and are thus not amenable to analytic solution except, perhaps, in the very simplest cases. Moreover, analytic solutions cannot be obtained when problem domains are irregular. The finite element method is a natural choice for solving problems with irregular boundaries, especially when Neumann (flux-type) boundary conditions are specified.

The solution procedure in NORIA-SP is transient; no provision has been made for obtaining steady solutions because such a steady solution procedure may not be convergent for the vast majority of analyses for which NORIA-SP would be used. However, it is always possible to construct a transient analysis that will approach the desired steady state after some elapsed time.

The following sections provide a theoretical background for and information on how to use NORIA-SP. Section 2 describes the equation that governs the transport of liquid water through a porous medium. Section 3 discusses the spatial discretization of the dependent variable in NORIA-SP, the Galerkin finite element formulation of the governing equation, and the enforcement of boundary conditions. Details of the time integration procedure are set out in Section 4. Sections 5 through 7 are directed primarily to the use of NORIA-SP: Section 5 gives an overview of the functions that are performed by NORIA-SP; Section 6 describes the input needed to execute NORIA-SP; and Section 7 illustrates the information in Section 6 by sample problems. Finally, Section 8 is a glossary of terms that may not be familiar to users who are not trained in finite element analysis and numerical methods.

2 Theory and Mathematical Model

2.1 Background

The theory of transport through porous media dates back to the late 1850s to the work of Darcy (1856). More recent and complete attempts to model such flows include the works of Richards (1931).

The mathematical model described in the following two subsections is relatively general. Even so, the following assumptions have been made:

- The single phase consists of a single component in liquid. Here, the liquid phase is assumed to be water but other constituents can be modeled equally well by NORIA-SP.
- All viscous flow (flow at high enough densities so that molecular effects are unimportant) is laminar and obeys Richards' equation (Freeze and Cherry, 1979), which is a form of Darcy's law for unsaturated media.
- The porosity of the matrix is uniform over each element. Up to twelve materials are allowed.

Aside from these restrictions, all properties are general. Each material property is specified in a user subroutine as a constant or as a function of the dependent or independent variables.

2.2 Conservation Equation

The basic equation that governs flow in porous media is a simple statement of conservation of mass:

$$\left(\frac{\partial \rho \Theta}{\partial p} \right) \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{q} = 0 \quad (2.1)$$

where

ρ = water density,

Θ = moisture content, *i.e.*, the volume fraction of matrix that is filled by water,

p = pore pressure in the liquid phase,

$\left(\frac{\partial \Theta}{\partial p} \right)$ = moisture capacitance,

$t =$ time, and

$\mathbf{q} =$ mass flux vector.

The terms on the left represent accumulation of mass; the term on the right represents net influx of mass.

In order to complete the governing Equation (2.1) and solve for pressure, a relationship must be chosen to specify how the flux depends on the unknown pressure. This relationship, which is constitutive in nature and thus depends on physical mechanisms, is discussed in the following two subsections.

2.3 Transport Equation

The mathematical model used in NORIA-SP describes the transport of liquid according to the pressure-driven Darcy flow equation:

$$\mathbf{q} = -\frac{\rho \mathbf{k}}{\mu} \cdot \nabla P, \quad (2.2)$$

where

$\rho =$ liquid density,

$\mathbf{k} =$ permeability of the porous matrix to flow in the liquid phase,

$\mu =$ dynamic viscosity of the liquid,

$P =$ effective pressure in the liquid phase, *i.e.*, $P = p + \rho g z = \rho g(\psi + z)$,

$p =$ pore pressure in the liquid phase,

$g =$ the acceleration caused by gravity,

$z =$ the vertical coordinate in a Cartesian or polar cylindrical coordinate system, *i.e.*, the z -axis is aligned with the direction of gravity and is opposite in sense, and

$\psi =$ pressure head, *i.e.*, pore pressure divided by ρg .

2.4 Composite Fracture/Matrix Model

NORIA-SP can be used to analyze the flow of water through fractured porous material. This is accomplished by formulating a single composite material through volume weighting of the material characteristics (permeability and moisture content) of the matrix and fracture materials. This volume weighting must be performed in the user subroutines FLUIDC and PERM as is demonstrated in the sample problem (Section 7). The development of this concept was formulated by Peters and Klavetter (1988). A major assumption of this model is one of pressure equilibrium between the fractures and surrounding matrix. The model is briefly described here, although it is recommended that those intending to implement this model refer to the above reference.

In the following discussions, the fracture porosity, Φ_f , is defined as the volume of fractures per unit volume of composite material, which includes both matrix and fractures. The volume of matrix per unit volume of composite material is $(1 - \Phi_f)$. Matrix porosity, Φ_m , is the porosity of matrix material alone, *i.e.*, the volume of pores per unit volume of matrix material. The composite, or total, porosity is the volume of void (from both fractures and matrix pores) per unit volume of composite material and is defined by

$$\Phi = \Phi_m(1 - \Phi_f) + \Phi_f. \quad (2.3)$$

The composite, or total, material permeability, K , is defined by volume weighting and is given by

$$K(c) = (1 - \Phi_f)K_m + \Phi_f K_f, \quad (2.4)$$

and the total moisture content, Θ , is the sum of the moisture contents in the matrix and fractures

$$\Theta = \Theta_m + \Theta_f = \Phi_m S_m + \Phi_f S_f \quad (2.5)$$

where S is saturation and subscripts m and f refer to the matrix and fractures, respectively. If the fracture porosity is defined as zero, the material properties reflect those of the matrix only.

Given the material properties described above, the pressure solution is found from Equations 2.1 and 2.2 independent of a matrix only ($\Phi_f = 0$) or a composite material ($\Phi_f \neq 0$) model. The computed pressure solution is then applied to define liquid flux through the matrix and, if the material is composite, to the fractures as well.

$$q_m = -\frac{\rho \mathbf{k}_m}{\mu} \cdot \nabla P, \quad (2.6)$$

$$q_f = -\frac{\rho \mathbf{k}_f}{\mu} \cdot \nabla P. \quad (2.7)$$

The linear, or pore, velocity of the liquid through the matrix is given by

$$v_m = \frac{q_m}{\Phi_m(S_m - S_{m,r})}, \quad (2.8)$$

and optionally through the fractures by

$$v_f = \frac{q_f}{\Phi_f(S_f - S_{f,r})}, \quad (2.9)$$

where S_r is the residual saturation.

3 Galerkin Finite Element Formulation

3.1 Weak Form of the Partial Differential Equations

The first step in applying the finite element method to a problem is to put the governing differential equation in weak form (Zienkiewicz, 1977). This step has three parts: (1) products of the governing differential equation, Equation (2.1), are formed with a set of weighting functions that are defined in the following two subsections; (2) the products are integrated over the problem domain; and (3) Green's theorem is used to reduce all second-order derivatives to first order. Applying the first two steps results in the following expression:

$$\int_{\Omega} \rho \frac{\partial \Theta}{\partial t} \psi_i dS = - \int_{\Omega} \nabla \cdot \mathbf{q} \psi_i dS, \quad (3.1)$$

where

ψ_i = the i th weighting function,

Ω = the area of the problem domain, and

dS = the differential of area.

Applying Green's theorem to Equation (3.1) leads to the following result:

$$\int_{\Omega} \rho \frac{\partial \Theta}{\partial t} \psi_i dS - \int_{\Omega} \mathbf{q} \cdot \nabla \psi_i dS + \int_{\partial \Omega} \mathbf{q} \cdot \mathbf{n} \psi_i ds = 0, \quad (3.2)$$

where

\mathbf{n} = the outward pointing unit normal vector along the perimeter of the problem domain, and

ds = the differential of length along the perimeter of the domain.

Equation (3.2) is the weak form of Equation (2.1) and is the equation that is solved by NORA SP. Equation (2.2) is used to evaluate \mathbf{q} .

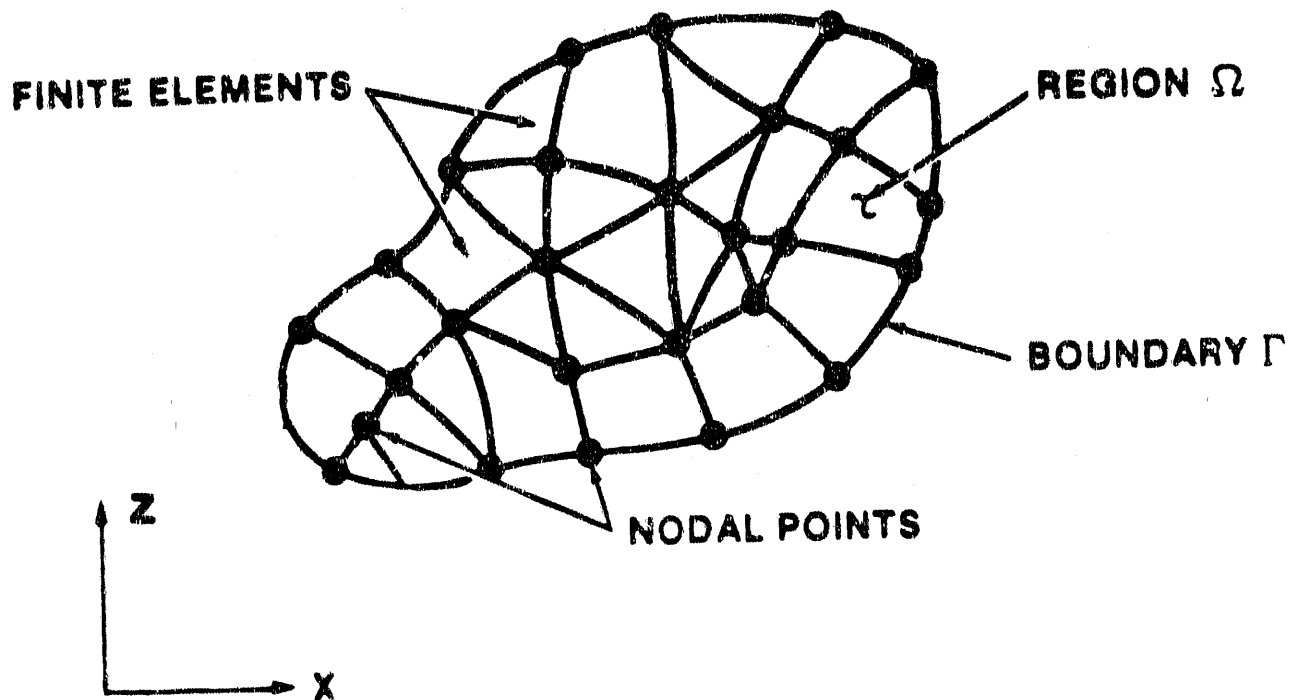


Figure 3.1. Finite Element Discretization of a Region

3.2 Finite Element Basis Functions

The heart of the finite element method is the representation of dependent variables by a set of basis functions, each of which spans only a small subdomain of the problem domain. Definition of the set of basis functions is accomplished by subdividing the problem domain into elements and defining a set of local basis functions on each element. The problem domain is subdivided so that it is the direct sum of the elements, as shown in Figure 3.1, except, perhaps, at curvilinear domain boundaries where element sides might only approximate the true boundary.

A large number of element types have been used in finite element analysis. Two types are available in NORIA-SP: (1) the eight-node isoparametric quadrilateral, and (2) the eight-node subparametric quadrilateral. The number of nodes in an element corresponds to the number of basis functions used to represent a dependent variable within that element. "Isoparametric" and "subparametric" refer to the way an element is mapped into a standard element.

Finite element basis functions are generally defined on standard elements. In the natural coordinate system, denoted by (ξ, η) , the standard eight-node quadrilateral element is a square with edges that fall on the lines $\xi = -1, +1$ and $\eta = -1, +1$ and with nodes that lie at corners and midway along sides (Figure 3.2). The basis functions for the eight-node quadrilateral are

$$\begin{aligned}
\phi_1 &= -\frac{1}{4}(1 - \xi)(1 - \eta)(1 + \xi + \eta) \\
\phi_2 &= -\frac{1}{4}(1 + \xi)(1 - \eta)(1 - \xi + \eta) \\
\phi_3 &= -\frac{1}{4}(1 + \xi)(1 + \eta)(1 - \xi - \eta) \\
\phi_4 &= -\frac{1}{4}(1 - \xi)(1 + \eta)(1 + \xi - \eta) \\
\phi_5 &= +\frac{1}{2}(1 - \xi^2)(1 - \eta) \\
\phi_6 &= +\frac{1}{2}(1 + \xi)(1 - \eta^2) \\
\phi_7 &= +\frac{1}{2}(1 - \xi^2)(1 + \eta) \\
\phi_8 &= +\frac{1}{2}(1 - \xi)(1 - \eta^2).
\end{aligned} \tag{3.3}$$

Here the subscripts refer to local node number, as shown in Figure 3.2.

The simplest conceivable problem domain is a square in the global coordinates, here either (x, z) or (r, z) . If this domain were subdivided into some number of uniform square elements, the mapping of an element from the original problem domain to the standard domain would be accomplished by translation and scaling. This is the simplest case of a subparametric mapping. Subparametric indicates that the mapping from the global domain to the standard domain is of lower order than the order of the basis functions themselves. For the eight-node quadrilateral element, the basis functions are of order two, *i.e.*, the basis functions are quadratic in the coordinates (ξ, η) . Thus, any first-order (linear) transformation is allowed in a subparametric mapping. On the other hand, an isoparametric mapping is of the same order as that of the basis functions themselves. Thus, any second-order (quadratic) transformation is allowed in an isoparametric mapping. In practice, this means that subparametric elements have straight sides and isoparametric elements have parabolic sides.

The finite element basis functions defined in Equation (3.3) have several properties that make them especially convenient:

- Each basis function is zero at all nodes except one, the node to which it corresponds, and it has a value of unity there.

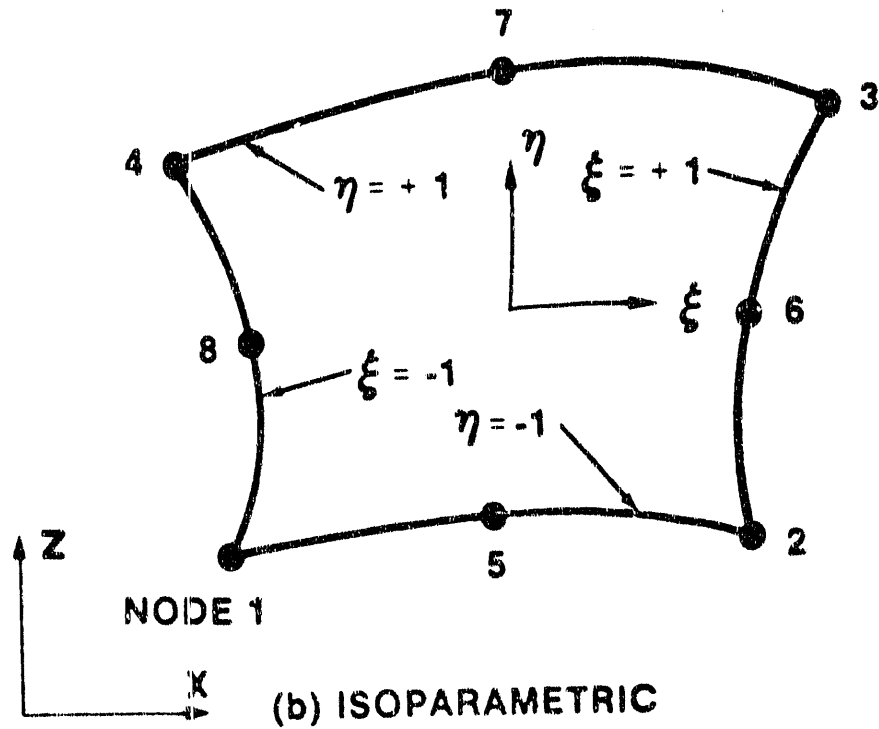
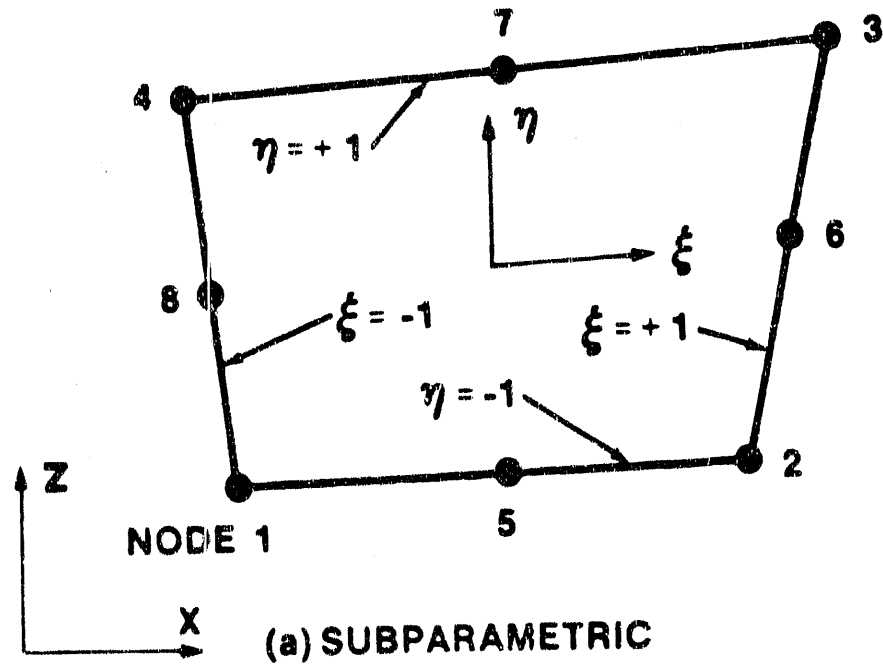


Figure 3.2. Quadrilateral Elements

- Each basis function is continuous over the entire problem domain; therefore, any variable expanded in the set of finite element basis functions is continuous over the problem domain.
- Each basis function is non-zero over at most a few elements, typically four quadrilateral elements.

The first property simplifies interpretation of results; the value of a dependent variable at a node point is simply the coefficient of the basis function corresponding to that node. The second property is mathematically essential in order to ensure the integrability of Equation (3.2). The third property means that resulting matrix equations will be sparse. Thus, special matrix-solvers can be used to reduce computational effort for direct elimination. The matrix-solver used here is a frontal method developed by Irons (1970) and generalized to handle asymmetric matrices by Gartling (1978).

3.3 Galerkin's Method

The preceding subsection defines the basis functions used to interpolate the dependent variables. Galerkin's method is to choose the same set of basis functions defined in Equation (3.3) as the weighting functions, ψ_i , used in Equation (3.2). The set of all the ϕ_i s, called the global set, is the union of the sets of local basis functions for each of the elements making up a finite element mesh. Thus, the global set of ψ_i s is identical to the global set of basis functions, the ϕ_i s.

The expansions of the dependent variable, P , in the global set of basis functions take the following form:

$$P = \sum_{j=1}^N P_j \phi_j. \quad (3.4)$$

where

N = the number of basis functions in the global set, which is the same as the number of node points in the finite element mesh.

P_j = a coefficient in the expansion of the effective liquid pressure, P .

To simplify the expansions of the complicated nonlinear terms in Equation (3.2), groups of properties and unknowns are expanded in a single expansion in ϕ_j . For example, the Darcy flow term in the liquid equation is expanded as follows:

$$\begin{aligned}
\int_{\Omega} \left[\frac{\rho \mathbf{k}}{\mu} \cdot \nabla P \right] \cdot \nabla \psi_i dS = \\
\int_{\Omega} \left[\sum_{j=1}^N \left(\frac{\rho \mathbf{k}}{\mu} \right)_j \phi_j \cdot \nabla \left(\sum_{k=1}^N P_k \phi_k \right) \right] \cdot \nabla \phi_i dS.
\end{aligned} \tag{3.5}$$

The quotient in parentheses followed by a subscript is evaluated at each of the node points of an element and expanded over that element using the same basis functions as those used in Equation (3.4). The advantage of the expansion shown in Equation (3.5) is that everything other than the basis functions can be taken outside of the integral so that element integrations can be performed once and for all.

$$\begin{aligned}
\int_{\Omega} \left[\frac{\rho \mathbf{k}}{\mu} \cdot \nabla P \right] \cdot \nabla \psi_i dS = \\
\sum_{j=1}^N \sum_{k=1}^N \left[\left(\frac{\rho \mathbf{k}}{\mu} \right)_j P_k \right] : \int_{\Omega} \phi_j \nabla \phi_k \nabla \phi_i dS
\end{aligned} \tag{3.6}$$

The final form of Equation (3.2) solved by NORIA-SP is shown below. The convention that repeated indices are summed is adopted here. Summed indices range from 1 to N .

$$\left(\rho \frac{\partial \Theta}{\partial t} \right)_j M_{ji} + \left(\frac{\rho \mathbf{k}}{\mu} \right)_j P_k : \mathbf{M}_{jki} + \mathbf{q} \cdot \mathbf{n} M_i = 0, \tag{3.7}$$

where the subscripted matrix quantities are defined below.

$$M_i = \int_{\partial\Omega} \phi_i ds$$

$$M_{ji} = \int_{\Omega} \phi_j \phi_i dS$$

$$\mathbf{M}_{jki} = \int_{\Omega} \phi_j \nabla \phi_k \nabla \phi_i dS$$

3.4 Boundary Conditions

Finite element boundary conditions are divided into two categories: essential and natural. An essential boundary condition is a Dirichlet one, *i.e.*, a boundary condition of the first kind; a natural boundary condition is any other type, *i.e.*, a Neumann (flux) or a Robin (mixed) boundary condition. A Robin boundary condition is one in which both a variable and its derivative appear.

Treatment of essential boundary conditions is entirely different than that of natural boundary conditions in the finite element method. Essential boundary conditions are imposed exactly by forcing coefficients in the basis function expansions to take on the required values. For example, if the pressure along a boundary were to be set equal to 1000 Pa, the coefficients of the basis functions used to expand pressure would be set to 1000 for each of the nodes on that boundary. On the other hand, natural boundary conditions are never imposed exactly in the finite element method. Instead, prescribed flux values are inserted in the appropriate boundary integral in Equation (3.2). For example, if a boundary were prescribed to be no flow, the boundary integral term would be set to zero there. No-flux boundary conditions are default in NORIA-SP. In the absence of any boundary condition specified by the user, no mass fluxes at all boundaries are presumed. Treatment of natural boundary conditions is one of the features that distinguishes finite element methods from finite difference techniques.

4 Time Integration Scheme

4.1 Strategy

The time integration scheme in NORIA-SP was chosen because it is quite robust, automatically adjusts time-step size to accommodate variations in natural time scale, and is quite accurate (local time truncation error is third order). It uses a predictor-corrector scheme coupled with a Newton iteration procedure to march ahead in time (Gresho *et al.*, 1979 and Bixler, 1989). The Newton iteration scheme is essential in NORIA-SP, given the highly nonlinear nature of the governing partial differential equation that it must solve. Furthermore, automatic time-step size adjustment is a requirement for solving porous flow problems because natural time scales commonly vary by several orders of magnitude. The Jacobian matrix for the Newton procedure is constructed using finite differences of each of the unknowns. Each step of the time integration procedure is described below. No steady state solution procedure is available in NORIA-SP. Steady solutions can generally be obtained by constructing a transient calculation that approaches a steady state after some elapsed time. The problem with general purpose steady-state solvers is that such solvers are not likely to be convergent for the majority of problems that would be analyzed using NORIA-SP.

4.2 Adams-Bashforth Predictor

The Adams-Bashforth predictor is described by Shampine and Gordon (1975) in their treatise on computer methods for initial value problems. The Adams-Bashforth predictor requires the dependent variables at the preceding timeplane and their rates of change at the two preceding timeplanes:

$$(P^p)_{n+1} = (P)_n + \frac{\Delta t_n}{2} \left[\left(2 + \frac{\Delta t_n}{\Delta t_{n-1}} \right) (\dot{P})_n - \frac{\Delta t_n}{\Delta t_{n-1}} (\dot{P})_{n-1} \right], \quad (4.1)$$

where

\dot{P} = the rate of change of P ,

$\Delta t_n = t_{n+1} - t_n$, and

p = a predicted quantity.

Estimates for the rates of change of pressure are made by the following formulas.

$$(\dot{P})_n = \frac{2}{\Delta t_{n-1}} [(P)_n - (P)_{n-1}] - (\dot{P})_{n-1} \quad (4.2)$$

$$(\dot{P})_{n+1} = \frac{\Delta t_{n-2}}{\Delta t_{n-1} + \Delta t_{n-2}} \left(\frac{(P)_n - (P)_{n-1}}{\Delta t_{n-1}} \right) + \frac{\Delta t_{n-1}}{\Delta t_{n-1} + \Delta t_{n-2}} \left(\frac{(P)_{n-1} - (P)_{n-2}}{\Delta t_{n-2}} \right) \quad (4.3)$$

Equation (4.2) is the common expression for rate of change using the trapezoid rule. However, it was found that using Equation (4.2) at timeplane n and at timeplane $n + 1$ produces temporal oscillations that become quite noticeable as steady state is approached. By using Equation (4.3) to estimate the rate of change of the dependent variables at timeplane $n + 1$, the oscillations are eliminated.

The Adams-Bashforth predictor cannot be used on the first two time steps because the dependent variables are not assigned values prior to timeplane zero, which is the initial condition. Thus a start-up procedure is required. The procedure used in NORIA-SP is described in Subsection 4.6.

4.3 Trapezoid-Rule Corrector

A trapezoid-rule (Crank-Nicolson) corrector step is taken after each predictor step to improve the predicted solution. In addition, using both schemes allows local time truncation error to be estimated so that time-step size can be adjusted. Automatic time-step size selection is described in Subsection 4.5.

Application of the trapezoid-rule step involves three parts.

- Time derivative in Equation (3.2) is replaced by a difference. For example,

$$\left(\frac{\partial \Theta}{\partial t} \right)_{n+1/2} \approx \frac{\Theta_{n+1} - \Theta_n}{\Delta t_n} \quad (4.4)$$

where the subscript $n + 1/2$ indicates that the time derivative is evaluated midway between timeplanes n and $n + 1$.

- The dependent variable and functions of the dependent variable in Equation (3.2) are evaluated midway between timeplanes. For example,

$$(P)_{n+1/2} \approx \frac{P_{n+1} + P_n}{2}, \quad (4.5)$$

$$(\mathbf{q})_{n+1/2} \approx \mathbf{q}(P_{n+1/2}). \quad (4.6)$$

- Equation (3.2) is solved for each of the coefficients, P_j , used in the expansion of pressure at the new timeplane.

The predictor step provides an estimate of the dependent variable at the $n + 1$ timeplane for use in Equations (4.4) and (4.5). If more than one Newton iteration is required at a given timeplane, the previous Newton result replaces the predicted solution at the new timeplane. Once the trapezoid-rule corrector step is complete, the size of the next time step is calculated, as described below, and another time step is initiated.

4.4 Newton Iteration Procedure

The equations generated in the trapezoid-rule step described above are usually highly nonlinear. As a result, an iterative method is needed to solve them. Newton's method is used in NORIA-SP because it has good convergence properties and exhibits a good domain of attraction. Newton's method requires the construction of a Jacobian matrix, which is accomplished numerically in NORIA-SP by the following steps: (1) calculating residuals, *i.e.*, the right side of Equation (3.2); (2) incrementing each of the finite element coefficients one at a time and recalculating the residuals; and (3) using the differences of residuals to estimate the rates of change with respect to the finite element coefficients. The computational work involved in constructing the Jacobian matrix is reduced by taking into account that the residuals in an element depend only on the nodal coefficients of that element. The resulting equations are then solved using a frontal method developed by Irons (1970) and generalized by Gartling (1978) to handle asymmetric matrices.

Several parameters control the operation of the Newton iteration process. The first parameter controls the size of the increments to the dependent variables used in calculating the Jacobian matrix. Normally, each of the dependent variables is incremented by 10^{-6} of the maximum absolute value in the field; for example, liquid pressure is incremented by 10^{-6} of the maximum value of P in the field. Three other parameters are used to control the number of Newton steps taken at a timeplane. Normally, only one Newton step is required; however, if the difference between the predicted and corrected solutions is too great, more than one Newton step may be taken. This difference is defined by a weighted *rms* norm and is given in the solution output:

$$rms = \sqrt{\sum_{j=1}^N \left(\frac{P_j - P_j^p}{P_{max}} \right)^2} . \quad (4.7)$$

Here,

max = the maximum absolute value of the variable at all node points, and

p = the predicted solution or, if one or more Newton steps have already been taken, the previous result of the Newton process.

Normally, Newton iteration is continued until $rms \leq 0.01$; however, no more than 3 Newton steps are allowed at a given timeplane. Also, if $rms \geq 0.2$, the time-step size is reduced, as described in the following subsection, and the time step is repeated. One additional parameter that controls the choice of time-step size is described in the next subsection. The values of these parameters are not easily, and ordinarily need not be, modified by the user. However, they may be modified when a problem so requires.

4.5 Automatic Time-Step Selection

Time-step size is varied at each timeplane to maintain local time truncation error as uniformly as possible. Taylor expansions are used to estimate local time truncation error for the Adams-Bashforth predictor and the trapezoid-rule corrector. The error estimate for the predictor is

$$\|P^p_{n+1} - P(t_{n+1})\| = \frac{1}{12} \left(2 + 3 \frac{\Delta t_{n-1}}{\Delta t_n} \right) \Delta t_n^3 \left\| \frac{\partial^3 P(t_{n+1})}{\partial t^3} \right\| + O(\Delta t^4). \quad (4.8)$$

Here, $P(t_{n+1})$ refers to the exact solution of the finite element Equation (3.2) at time t_{n+1} . The double vertical bars indicate the rms norm defined in Equation (4.7). The error estimate for the corrector is

$$\|P_{n+1} - P(t_{n+1})\| = \frac{1}{24} \Delta t_n^3 \left\| \frac{\partial^3 P(t_{n+1})}{\partial t^3} \right\| + O(\Delta t^4). \quad (4.9)$$

By combining Equations (4.8) and (4.9) to eliminate the norm of the third time derivative of P , and by using the triangle inequality to eliminate the exact solution $P(t_{n+1})$, the following inequality results:

$$d_{n+1} \equiv \|P_{n+1} - P(t_{n+1})\| \leq \frac{\|P_{n+1} - P^p_{n+1}\|}{3 + 6\Delta t_{n-1}/\Delta t_n} + O(\Delta t_n^4). \quad (4.10)$$

Here, d_{n+1} is the error in the corrected solution at timeplane $n + 1$. Finally, Equation (4.9) implies that

$$\frac{d_{n+2}}{d_{n+1}} = \left(\frac{\Delta t_{n+1}}{\Delta t_n} \right)^3 + O(\Delta t_n), \quad (4.11)$$

which can be solved for Δt_{n+1} to obtain

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\epsilon}{d_{n+1}} \right)^{1/3}. \quad (4.12)$$

The upper bound on d_{n+1} from Equation (4.10) is used in Equation (4.12) to establish a lower bound on Δt_{n+1} . Here, d_{n+2} has been set to ϵ , which is the desired value of the local time truncation error at the next time step. The default value of ϵ in NORIA-SP is 10^{-5} . Equation (4.12) is used to calculate time-step size for each time step beginning with $n = 4$. However, when the Δt_{n+1} is less than 80% of Δt_n , the solution at timeplane $n + 1$ is recalculated and Δt_n is replaced by the smaller value.

4.6 Start-Up Procedure

The Adams-Bashforth predictor cannot be used on the first two time steps because rates of change of the dependent variables are not known prior to the initial condition. As a result, a start-up procedure is required to initiate the time integrator. The one used here is to take two backward-difference steps before initiating the two-step time integration procedure. The reason backward difference is chosen instead of trapezoid-rule is that backward-difference steps help damp out discontinuities that may be present in the initial data. Luskin and Rannacher (1982) have shown that a few backward-difference steps taken at the beginning of a time integration procedure can actually enhance the accuracy of the overall results by damping singularities that might otherwise cause oscillations.

Automatic time-step size selection begins after the first predictor-corrector step, which is the third time step. The only exception is that if the user requests that output be printed at a particular time, time-step size may be limited so that results can be computed at that exact value of time. User-requested output is described further in Section 6.

5 Program Description

5.1 Organization

The program organization in NORIA-SP reflects the steps taken in setting up, solving, and evaluating the finite element analysis of water transport through a porous medium. NORIA-SP contains about 50 routines that can be categorized according to five primary groups of tasks. These program tasks are shown schematically in Appendix A and are described in Subsections 5.2-5.6.

Mesh Generation. Assign nodal point locations. Assign nodal points to finite elements and organize data for equation formulation. The finite element library in NORIA-SP consists of two element types.

Boundary and Initial Conditions. Assign boundary conditions to appropriate nodal points and select initial conditions.

Solution Procedure. Form element coefficient matrices. Use material property data to assemble local residuals and matrices. Assemble local residuals and matrices into global sets. Solve transient equations for dependent variables.

Calculation of Derived Quantities. At user-specified times during the calculation, certain derived quantities are calculated and reported.

Post-processing. NORIA-SP has no internal plotting capabilities. Instead, an output file in the EXODUS format (Mills-Curran *et al.*, 1988) is produced. This file can then be used as direct input to graphics packages such as TRINITY II (Glick and Gartling, 1988) and BLOT (Gilkey and Glick, 1989), or can be translated for use with other post-processing software.

5.2 Mesh Generation

Generation of node points in NORIA-SP is independent of element specification. As a result, a user may generate more nodes than are actually used in a problem so that he may experiment with node point location before selecting a mesh. This option is especially useful when a large or complicated domain is to be subdivided into elements.

For purposes of mesh generation, a problem domain is considered to be made up of regions determined by the user. Region boundaries are approximated by curves that can consist of linear, quadratic, or cubic mappings of a line into an (x,z) or (r,z) coordinate system, *i.e.*,

$$x = P_1(\xi), \quad z = P_2(\xi)$$

or

$$r = P_1(\xi), \quad z = P_2(\xi).$$

Here, P_1 and P_2 are polynomials that may be linear, quadratic, or cubic, and ξ is a parametric coordinate. This approach allows relatively complicated boundary shapes to be modeled easily and accurately.

The user locates node points along the boundaries of a region by specifying the number of nodes along each region boundary and by specifying a geometric factor that determines relative spacing. Node points within a region are generated automatically once the boundary nodes have been identified by an (I, J) numbering system. The user may specify node points along curves or at individual points. Node points may also be located by means of a subroutine supplied by the user.

The final step in mesh generation is to assign node points to elements. In the case of isoparametric elements, the element shape is determined by all of the nodes assigned to that element; in the case of subparametric elements, the element shape is determined by corner nodes. The midside nodes in subparametric elements are located midway on the line segment that connects adjacent corner nodes. In general, element boundaries that border a region do not coincide with the region boundary itself. However, isoparametric elements better approximate a curved boundary than do subparametric elements.

Two distinct errors are introduced in NORIA-SP when approximating curved boundaries for a problem domain. The first error is in the step of approximating problem domain boundaries by region boundaries. The second error is in the step of approximating region boundaries by element boundaries. However, these errors are generally minor and disappear altogether when problem domains have straight boundaries. Furthermore, the first error can be reduced by breaking up an irregular problem domain into more and more regions and can be completely eliminated by assigning boundary node locations individually, either by means of the point option mentioned above or the user-supplied subroutine. The second error is inherent in the finite element method, but, as shown by Strang and Fix (1973), with certain mild restrictions on the shapes of elements, this error is of no higher order than the error made by representing the solution in terms of the finite element basis functions.

Element Library

NORIA-SP uses static arrays that have been dimensioned to accommodate up to 5,000 elements; however, NORIA-SP can be redimensioned to handle larger meshes. There are two basic element types available in NORIA-SP.

Isoparametric Eight-Node Quadrilateral. The isoparametric eight-node quadrilateral (QUAD8/8) can have curved sides, as shown in Figure 3.2b, and uses the basis functions defined in Equation (3.3) to interpolate dependent variables and to map from the standard domain to a real domain.

Subparametric Eight-Node Quadrilateral. The subparametric eight-node quadrilateral (QUAD8/4) has straight sides and uses the basis functions defined in Equation (3.3) to interpolate dependent variables and bilinear basis functions, σ_j , to map from the standard domain to a real domain. Bilinear basis functions are defined as

$$\begin{aligned}\sigma_1 &= \frac{1}{4}(1 - \xi)(1 - \eta) \\ \sigma_2 &= \frac{1}{4}(1 + \xi)(1 - \eta) \\ \sigma_3 &= \frac{1}{4}(1 + \xi)(1 + \eta) \\ \sigma_4 &= \frac{1}{4}(1 - \xi)(1 + \eta).\end{aligned}\tag{5.1}$$

Isoparametric and subparametric transformations of an element are respectively defined as

$$x = \sum_{j=1}^M x_j \phi_j; \quad z = \sum_{j=1}^M z_j \phi_j;\tag{5.2}$$

$$x = \sum_{j=1}^M x_j \sigma_j; \quad z = \sum_{j=1}^M z_j \sigma_j.\tag{5.3}$$

Here, (x_j, z_j) are the nodal coordinates of an element in the real domain and M is the number of basis functions used in the expansion over an element.

Evaluation of the integrals in Equation (3.2) is carried out over standard elements by using the transformations defined in Equations (5.2) and (5.3). This evaluation requires that the Jacobian of the mapping be calculated. Integration is performed by nine-point Gaussian quadrature.

5.3 Boundary and Initial Conditions

Boundary conditions are specified on an element basis. Effective liquid pressure can be specified for individual nodal points or element sides. Use of the latter is restricted to cases where there is a uniform value along a side. Mass fluxes are specified as constants along element sides. Zero-mass-flux conditions are the default boundary conditions in NORIA SP, *i.e.*, no boundary condition need be specified for an impermeable surface. Moreover, any of the above types of boundary conditions may be specified as functions of any of the dependent or independent variables by means of user-supplied subroutines.

Two options exist in NORIA-SP for specification of initial conditions. The dependent variable may be specified in the input deck to be constant over each material. Arbitrary

initial conditions may be input by a user-supplied solution file written in the format specified in Section 6.

5.4 Solution Procedure

The solution procedure used in NORIA-SP is described in Section 4. The trapezoid-rule corrector is implemented on an element-by-element basis. The solution process is performed by Irons' frontal procedure (Irons, 1970), which has five distinct steps: (1) the elimination process is set up for the frontal solution procedure; (2) a residual vector and an element coefficient matrix are assembled for an element; (3) the local residual vector and the element coefficient matrix are respectively added to a global residual vector and coefficient matrix; (4) equations that have been fully summed are forward-eliminated; steps (2) through (4) are repeated until all elements have been treated; and (5) back substitution is carried out to produce the solution vector at the new timeplane. This process is repeated for each timeplane and may be repeated several times for a single timeplane, as described in Section 4.

5.5 Calculation of Derived Quantities

Quite frequently the user wishes to examine quantities derived from the basic field variables. These quantities include saturations, conductivities, liquid flux, and pore (linear) velocities in the matrix, and, if a composite model is used, in the fractures and total (composite) quantities as well. The variables reported are controlled by the printer flag on the solution control card (Section 6.7).

5.6 Plotting

NORIA-SP has no internal plotting capabilities. Instead, an output file in the EXODUS format is produced. This file can then be used as direct input to graphics packages such as TRINITY II (Glick and Gartling, 1988) and BLOT (Gilkey and Glick, 1989). Both BLOT and TRINITY II can generate plots of nodal point locations, finite element mesh, and outlines of the domain showing material boundaries, as well as contours, time histories, and profiles. Additionally, particle pathlines can be plotted using TRINITY II. Pressures, or any of the derived quantities reported during the calculation, may be included in the EXODUS file with the "post" option on the stop card (Section 6.9).

6 Input Guide

6.1 Introduction

The structure of an input deck for NORIA-SP corresponds to the steps required to formulate and solve a finite element problem. Through a series of command and data cards, the program is directed to such functions as generation of nodal points, construction of elements, construction of element-coefficient matrices, solution of the finite element equations, and postprocessing of solution field data. The actual sequence of commands to the program is quite flexible, although there are some obvious limitations to the order in which the operations can be carried out. In the following subsections, the command and data cards required by NORIA-SP are described in roughly the same order that they would normally appear in an input deck.

The conventions listed below are used in the description of input cards.

- Upper-case words imply a command or alphanumeric input value, *e.g.*, FORMKF. This notation is for descriptive purposes only; the actual input cards must contain lower-case characters.
- Lower-case words and symbols imply that an alphanumeric or numerical value of the specified variable is expected, *e.g.*, *xmax*.
- All variables are input in a free-field format and successive variables are separated by commas.
- [] indicates optional parameters that may be omitted by using successive commas in a variable list. If the omitted parameter is not followed by another parameter, no commas need be used.
- < > indicates the default value for an optional parameter.
- An asterisk (*) may be used to continue a variable list onto a second data card. It is placed after the last character on the card that is to be continued.
- A dollar sign (\$) may be used to end a data card so that the remaining space on the card may be used for comments.
- The contents of each input card are indicated by underlining.
- All quantities associated with a coordinate direction are expressed in terms of the planar (x, z) coordinate system. The corresponding quantities for axisymmetric problems are input by replacing the horizontal coordinate, x , with the radial coordinate, r .

- The user is free to select dimensional units; however, the units chosen must be consistent between all quantities defined in both the input deck and any user-supplied subroutines.
- When an input value is restricted to a certain range of values, that range is given; otherwise, consideration should be given to the applicable computer's accuracy and accepted values. Of course, the user is responsible for defining a physically meaningful problem.

The command cards are described in the following order:

- Header card
- SETUP command card
- FORMKF command card
- VELOCITY command card
- OUTPUT command card
- UNZIPP command card
- RESTART command card
- Program termination card

In the following subsections, the individual descriptions of the command cards are discussed, followed by descriptions of the input deck structure, user subroutines, initial conditions, error messages, and computer requirements.

6.2 Header Card

The header card must be the first card in a deck. A \$ must appear in the first column; the remaining 79 columns are available for a problem title. The header card has the following format.

\$ *problem title*

6.3 SETUP Command Card

The first task in formulating a finite element analysis using NORIA-SP is to specify the material properties and to define the finite element mesh and boundary conditions. The material properties are described in Section 2; the mesh and boundary conditions in Section 3. These functions are accomplished by means of the SETUP command and three sets of associated data cards.

The SETUP command card has the following format:

SETUP, [*iprint*], [*maxi*], [*order*]

where

iprint $\prec 2 \succ$ determines the amount of printout produced during the setup operation. Output increases with the value of *iprint*. For *iprint* = 1, no setup information is printed; *iprint* = 2 gives element definition (*c.g.*, type, material, coordinates) and boundary conditions (*c.g.*, type, location, value); *iprint* = 3 includes output from *iprint* = 2 plus grid input data for the problem geometry; *iprint* = 4 includes output from *iprint* = 3 plus coordinates and associated (*I, J*) values for each node both by (*I, J*) value and by element.

maxi $\prec 18 \succ$ is the maximum number of *I* rows of node points to be generated. *Maxi* need only be specified if there are more than 18 *I*-rows or more than 2770 *J*-rows. The limit on the maximum *I*- and *J*-rows is $I \cdot J \leq 50,000$.

order $\prec \succ$ determines the numbering of the elements. For the default (order left blank), the elements are numbered by increasing (*I, J*) values [*c.g.*, (1, 1), (2, 1), (3, 1), ..., (1, 2), (2, 2), ...]. For *order* = PRESCRIBED, the elements are numbered according to their order in the input list. The elements should be ordered so that the front width of the problem is minimized (Irons, 1970). For a basically rectangular shaped geometry, elements should be numbered across the dimension of the domain containing the fewest elements.

Following the SETUP command card, three sets of data cards are required. These data sets specify material properties, nodal point locations, and finite element mesh and boundary conditions. Each of the data sets is terminated by an END card. The third END card terminates the SETUP command and readies NORIA-SP for the next command.

Material Data Cards

Material data cards are of two types - one for fluid properties and one for matrix properties. The data card for fluid properties has the following form:

[*material name*], *number*, ρ , μ , *g*

where

material name is an optional alphanumeric material name for user reference.

number is an internal reference number for the material. The fluid must be number 1.

ρ is the density of the liquid.

μ is the dynamic viscosity of the liquid.

g is the acceleration caused by gravity.

The values defined must be consistent with the system of units chosen by the user. The default value for all parameters omitted on fluid property cards is zero.

Material properties for matrix materials are specified according to the following format:

[*material name*], *number*, P_0 , α

where

material name is an optional alphanumeric material name for user reference.

number is an internal reference number for the material. Matrix materials may use any or all of the numbers 2 through 13.

$P_0 < 0 >$ is the initial effective liquid pressure in a material, $P = p + \rho g z$ or $P = \rho g(\Psi + z)$ where $\Psi = p/\rho g$.

$\alpha < 0^\circ >$ is the angle, measured in degrees, from the x-axis to the first principal axis in the clockwise direction.

The material permeability, porosity, moisture content and capacitance are determined in user-supplied subroutines described in Subsection 6.11. The parameters α and P_0 must be set to constant values for each material. Default values are zero unless otherwise specified. An END command must follow the material property portion of the SETUP command.

END

Porous materials modeled by NORIA-SP can have either isotropic or orthotropic permeabilities. For isotropic materials (materials in which permeabilities are not functions of direction), permeability is defined in the user subroutine by $K_{11} = K_{22} = K$. The angle on the material property card, α , is either zero or left blank. For orthotropic materials (materials in which permeability is a function of direction), permeability is defined in the user subroutine along the principal axes: K_{11} is the value of permeability in the direction of the first principal axis; K_{22} is the value of permeability in the direction of the second principal axis; and α is the angle measured in the clockwise direction from the x-axis to the first principal axis. The angle α need be specified only when the principal axes are not aligned with the coordinate axes.

NORIA-SP does not contain any dimensional constants or properties; therefore, the user is free to choose any consistent set of units. Furthermore, it is possible to cast the governing equations (2.1) and (2.2) in dimensionless form. If a dimensionless form is used, some of the parameters in the material property cards would be replaced by dimensionless groups and some would be replaced by unity. This approach might be useful for large-scale parametric studies.

Nodal-Point Data

After material properties have been specified, nodal points for the finite element mesh are generated. Generation of nodal points and generation of elements are distinct operations in NORIA-SP. Nodal-point locations are calculated using quadrilateral regions. The boundaries of the regions are determined by polynomial mappings, as described in Subsection 5.2. The user defines each region by specifying a set of points that lie on the boundary of the region. Nodal points are identified by an (I, J) numbering system. The location of nodal points within a region is entirely controlled by the number of points along each of the boundaries of the region and by geometric parameters that control relative nodal spacing along each of the boundaries. Three data cards are needed to define the nodal points in a region.

$$\begin{aligned} & \underline{i_{min}, j_{min}, i_{max}, j_{max}, [q_1], [q_2], [q_3], [q_4], [POLAR], [x_0], [z_0]} \\ & \underline{x_1, x_2, x_3, x_4, [x_5], [x_6], \dots, [x_{12}]} \\ & \underline{z_1, z_2, z_3, z_4, [z_5], [z_6], \dots, [z_{12}]} \end{aligned}$$

where

i_{min}, j_{min} are the minimum values of I and J for the region being generated (Figure 6.1a).

i_{max} , j_{max} are the maximum values of I and J for the region being generated (Figure 6.1a). The difference between the maximum and minimum values determines the number of node points in the region.

$g_i < 1 >$ specifies a geometric factor for nodal spacing along the i th side of the region (Figure 6.1a). The geometric factors, shown in Figure 6.1a, are defined by

$$g_1 \equiv \frac{\Delta i_1}{\Delta i_2}, \quad (6.1)$$

$$g_2 \equiv \frac{\Delta j_1}{\Delta j_2}, \quad (6.2)$$

and Δm , for $m=i$ or j , is given by

$$\Delta m = L \cdot \left(\frac{g^{-1} - 1}{g^{-M} - 1} \right) \cdot g^{-(m-1)} \quad (6.3)$$

where

L = total length in the i or j direction of the region being generated,

M = total number of nodal points - 1,

m = index that goes from 1 to M .

The definitions for g_3 and g_4 are the same as those for g_1 and g_2 , respectively. The default values of unity give equal nodal spacing along a side. Geometric factors that are larger or smaller than unity give geometric nodal spacings (*i.e.*, the distances between consecutive nodes form a geometric progression with geometric factor g_i).

POLAR specifies the use of a polar rather than a Cartesian coordinate system. The default is a Cartesian coordinate system if this parameter is omitted.

$x_0 < 0 >$, $z_0 < 0 >$ designates the origin of a polar coordinate system.

x_1, z_1, x_{12}, z_{12} define the coordinates of the four corner and optional side points for each region. If a region is bounded by straight lines, only the coordinates of the four corners need be specified (*i.e.*, x_1, x_4 and z_1, z_4). If any of a region's sides are curved, then the appropriate side nodes must be specified as shown in Figure 6.1b. A quadratic mapping is used to define a boundary if one side point is specified; a cubic mapping is used to define a boundary if two side points are specified. Points should be nearly equally spaced along each side of a region.

There are no limits on the number of regions that may be used to define a set of nodal points. The only restriction is that $I_{max} \cdot J_{max} \leq 50,000$.

It is sometimes convenient to be able to position an individual nodal point or a string of nodes. The following data cards allow the user to implement these options.

$$\underline{\text{POINT}, i, j, x_1, z_1, [\text{POLAR}], [x_0], [z_0]}$$

where

i, j is the I, J specification for the point.

x_1, z_1 is the coordinate pair that defines the location of the point.

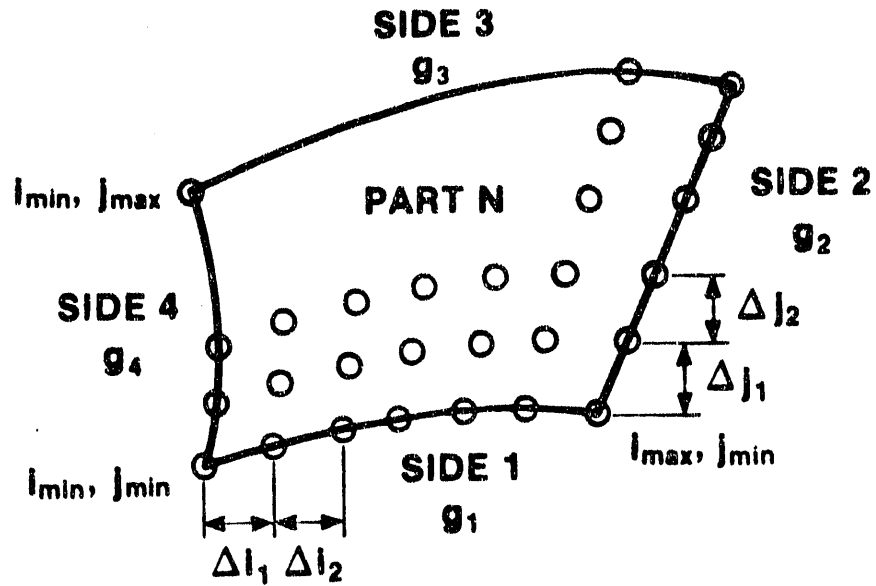
POLAR specifies the use of a polar rather than a Cartesian coordinate system.

$x_0 \prec 0 \succ, z_0 \prec 0 \succ$ designates the origin of a polar coordinate system.

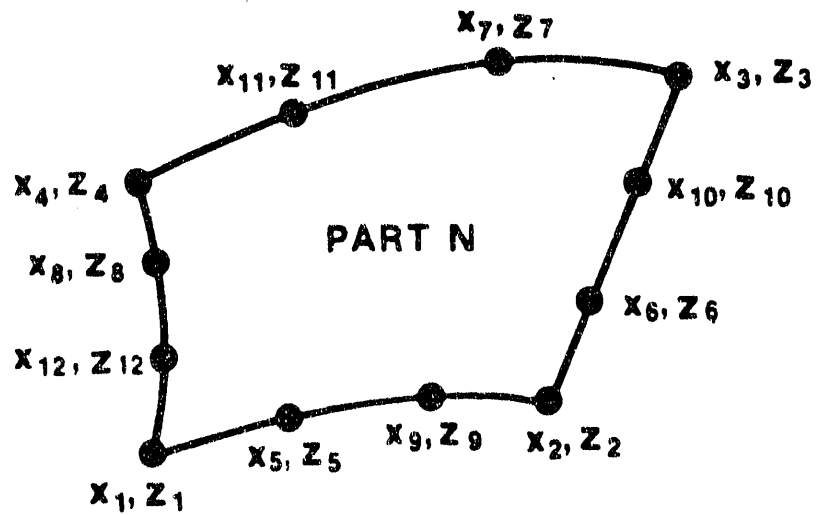
$$\underline{\text{ARC}, i_{min}, j_{min}, i_{max}, j_{max}, [g_1], [\text{POLAR}], [x_0], [z_0]}$$

$$\underline{x_1, x_2, [x_3], [x_4]}$$

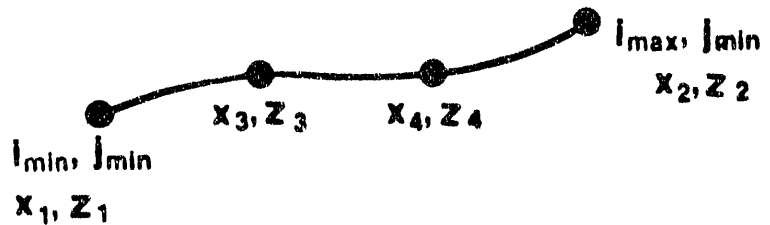
$$\underline{z_1, z_2, [z_3], [z_4]}$$



(a) I, J NOMENCLATURE



(b) COORDINATE NOMENCLATURE



(c) ARC NOMENCLATURE

where

i_{min}, j_{min} are the minimum values of I and J for the arc, as shown in Figure 6.1c.

i_{max}, j_{max} are the maximum values of I and J for the arc, as shown in Figure 6.1c.

Because a one-dimensional array of node points is generated by this command, either i_{min} must equal i_{max} or j_{min} must equal j_{max} . The difference between the maximum and minimum values determines the number of nodal points generated along the arc.

$g1 = 1$ specifies the geometric factor for nodal spacing along the arc. The gradient is defined either by Equation (6.1) or Equation (6.2), depending on whether I or J is being incremented along the arc.

POLAR specifies the use of a polar rather than a Cartesian coordinate system.

$x_0 = 0, z_0 = 0$ designates the origin of a polar coordinate system.

x_1, z_1, x_4, z_4 define the coordinates of the ends of the arc and optional intermediate points. If the arc is a straight line, only the first two sets of coordinates need be specified; if the arc is curved, either one or two intermediate points should be specified, as shown in Figure 6.1c.

There are no limits to the number of POINT and ARC data cards that may be used in generating a mesh. Both types of commands may appear anywhere within the nodal data portion of the SETUP command.

One further option exists for defining nodal point locations. Nodal coordinates can be defined by the user in a subroutine called EXTDEF. This subroutine provides a great deal of flexibility and is described in Subsection 6.11. Subroutine EXTDEF is accessed by placing the following card anywhere within the nodal data portion of the SETUP command.

EXTDEF

An END command must follow the nodal data.

END

Element and Boundary Data

Following the generation of nodal points, NORIA-SP is ready to accept element and boundary condition data. Because the nodal points are generated independently of the elements, selection of nodes to construct an element is very flexible. The process of constructing an element consists of identifying a group of nodal points to serve as corner and midside nodes of the element. This concept is apparent from the form of an element data card.

$$\underline{element\ type, mat, i_1, j_1, [i_2], [j_2], ..., [i_n], [j_n]}$$

where

element type is an alphanumeric name for the type of element. The allowed element types are described below.

mat is the matrix material number for the element. This number should correspond to one of the material numbers that appears on a matrix material property card.

$i_1, j_1, [i_2, j_2, \dots]$ are the I, J values for the node points in the element. The nodes are listed counterclockwise around an element, starting with any corner, as shown in Figure 6.2. In most situations, the list of I, J values may be significantly condensed. When I, J is specified only for the first node, the following values for the remaining nodes are assumed:

$$i_4 = i_8 = i_1$$

$$i_5 = i_7 = i_1 + 1$$

$$i_2 = i_3 = i_6 = i_1 + 2$$

$$j_2 = j_5 = j_1$$

$$j_6 = j_8 = j_1 + 1$$

$$j_3 = j_4 = j_7 = j_1 + 2.$$

When I, J values are specified for corner nodes only, the I, J values for each midside node are taken to be the average of the values at the adjacent corners. The coordinates of midside nodes are redefined as the midpoint of the adjacent corners.

An element's type is specified by the *element type* parameter. The following element types may be used in NORIA-SP.

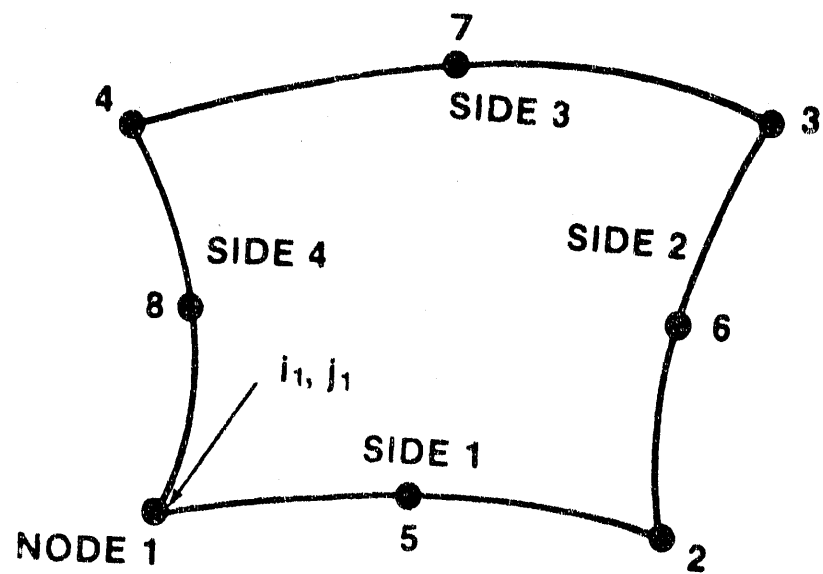


Figure 6.2. Numbering of Element Nodes and Sides

- QUAD8/4 is a subparametric quadrilateral with straight sides oriented arbitrarily.
- QUAD8/8 is a general isoparametric quadrilateral that uses a quadratic mapping to determine the shape of element sides.

In the generation of subparametric elements, physical coordinates (x, z) of midside nodes need not lie precisely on the element side because these coordinates are not used in element construction. The basis functions defined in Equation (3.3) are used to interpolate unknowns over quadrilateral elements:

Two points about the (I, J) identification of an element are noteworthy. Each element is identified internally by the (I, J) pair of the first node named on the element data card, *i.e.*, (i_1, j_1) . Because any corner node may be named first on an element data card, the internal identification may not be unique. The user must avoid assignment of a duplicate (I, J) identifier to any element that has an imposed boundary condition. This problem can always be avoided by selecting an appropriate nodal ordering on the relevant element data cards. The second point is that element connectivity is determined by nodal (I, J) values. The problem is that in some situations it is convenient to assign more than one (I, J) identifier to a single node point. The user must make sure that elements having a common node point use the same (I, J) identifier for that node. Otherwise, the elements will not be properly connected.

Boundary conditions are specified by element and may appear anywhere after the element to which the boundary condition applies has been defined and before the end of the element and boundary condition portion of the SETUP command. Essential boundary conditions can be specified as being constant along an element side or by node; natural boundary conditions can be specified only as being constant along element sides. The format for prescribing boundary conditions is

$$\underline{BC, b. c. type, i_1, j_1, side/node, value/curve no.}$$

where

b. c. type is an alphanumeric name for the type of boundary condition. The allowed types are described below.

i_1, j_1 is the I, J identification of the element to which the boundary condition applies, *i.e.*, the first I, J pair named on the element data card.

side/node identifies the side or node of the element to which the boundary condition is to be applied. The numbering of nodes and sides begins with the identifying node, *i.e.*, the first node named on the element data card, and proceeds counterclockwise, as shown in Figure 6.2.

value/curve no. is the numerical value of the applied boundary condition or the number of the CURVE subroutine from which the value of the boundary condition will be determined. CURVE subroutines are described further in Subsection 6.10.

Warning: A pressure boundary value of **exactly** zero will not be recognized as a fixed boundary.

The choices of *b, c, type* that may be used in NORIA-SP are

- PEFf specifies a constant value for effective liquid pressure at a node.
- PEFfSIDE specifies a constant value for effective liquid pressure along an element side.
- PEFfVARY specifies that effective liquid pressure along an element side is variable. A CURVE subroutine must be supplied by the user to determine how effective liquid pressure depends on other variables.
- ULSIDE specifies a constant value of liquid flux in the direction of the outward-pointing normal along an element side.
- ULVARY specifies that liquid flux in the direction of the outward-pointing normal is constant along an element side but depends on other variables there. A CURVE subroutine must be supplied by the user to implement this boundary condition.
- Impermeable surface requires no boundary condition to be specified.

Each of the above boundary conditions can be used with any of the above element types. NORIA SP will permit a total of six CURVE subroutines to be used. The format for these subroutines is described in Subsection 6.11.

In order to facilitate the specification of elements and boundary conditions, a looping feature is available in NORIA SP. This feature allows the definition of HLOOPs and JLOOPs (which are similar to FORTRAN DO loops) for incrementing data in the I and J directions. Nesting of the loops may be in either order, but no more than one HLOOP or one JLOOP may be in effect at once. All I or J values are given the same increment within the loop. The looping commands have the following form:

```

HLOOP, upass, inc
.....
JLOOP, upass, inc
.....
```

where

npass specifies the number of passes to be made through the loop.

inc specifies the increment to be added to the I or J values found within the loop. The *inc* parameter may be negative.

Element data cards, boundary condition data cards, or a combination of the two, follow the looping commands. I-looping and J-looping are terminated respectively by the following commands.

IEND

JEND

Values of I and J the first time through the loop are those given on the data cards that lie within the loop. These looping commands may appear at any point within the element and boundary condition portion of the SETUP command. An example of the use of this looping procedure is given in Section 7.2

6.4 FORMKF Command Card

After the completion of the SETUP command, the next task in the finite element formulation of a problem is to perform the element integrations necessary to solve the finite element Equation (3.2). This task is triggered by the FORMKF command:

FORMKF, [*symmetry*]

where

symmetry is an alphanumeric name that indicates the type of coordinate system desired. A two-dimensional Cartesian coordinate system with coordinates x and z is used in performing the element integrals if *symmetry* is omitted; a two-dimensional polar coordinate system with coordinates r and z is used in performing the element integrals if *symmetry* is set to AXISYM. In other words, planar symmetry is the default option, and axisymmetry is specified by setting *symmetry* =AXISYM.

6.5 VELOCITY Command Card

NORIA-SP includes the option of computing velocities by two methods. The value of *location* determines the calculation procedure used.

VELOCITY, [*location*]

where

location = GAUSS specifies the velocities to be calculated at the gauss points, using the gauss point pressure gradients, and then extrapolated to the corner nodal point locations. If *location* is left blank the velocities will be calculated at the gauss points.

location = NODAL specifies the velocities to be calculated at the nodal points using the nodal point pressure gradients.

6.6 OUTPUT Command Card

It is sometimes convenient to limit the amount of output generated during the solution of a problem by NORIA-SP. It may also be convenient to output values of the dependent variables at locations other than nodal points. Both of these features are available in NORIA-SP by using the OUTPUT command card. Selective printing of all variables is enforced by an OUTPUT command card with the following format:

OUTPUT, *type*, *n*₁, *n*₂, ..., *n*₅₀

where

type is an alphanumeric name that indicates how the subsequent data are to be interpreted. If *type* = SINGLE, the data that follow are taken to be element numbers. If *type* = STRING, the data that follow are interpreted as pairs of element numbers that define the lower and upper limits on a string of elements.

*n*₁, *n*₂, ..., *n*₅₀ is a list of element numbers that indicates which elements are to be included in the output list. A maximum of 50 individual element numbers or 25 element pairs may be specified on a single card.

There is no limit to the number of OUTPUT command cards that may be used in a NORIA-SP input deck. OUTPUT cards of both *types* may be mixed together in any order. However, all OUTPUT command cards must precede the UNZIP command card described in the following subsection. If no OUTPUT cards are included in the input deck, dependent variables are printed at every element in the finite element mesh.

Output at special points, *i.e.*, points that do not coincide with nodal points, may be requested by an OUTPUT card of the following format:

OUTPUT, POINTS, $x_1, z_1, x_2, z_2, \dots, x_{25}, z_{25}$

where

x_1, z_1-x_{25}, z_{25} is a list of x, z [or r, z] coordinates for the special points. A maximum of 25 points may be specified on a single command card.

NORIA-SP allows up to 50 special points to be specified during an analysis. Results at special points are printed in the output. The value of *iprt* on the UNZIP card controls the number of variables printed. The default values (*iprt* = 0) printed are pressure, saturation, and conductivity. Optional values (*iprt* > 0) are fluxes (matrix or total) and velocities (matrix). If the solution file (tape19) is to be saved (see the RESTART, NOSAVE command) and special points are requested, a solution file at the designated special points (tape17) is also saved. OUTPUT command cards that designate special points may be mixed with cards that limit output.

6.7 UNZIP Command Card

The UNZIP command card instructs NORIA-SP to begin assembling and integrating the finite element Equations (3.2). The UNZIP command card has the following format:

UNZIP, t_i , $[t_f]$, Δt_i , $[no. steps]$, $[init. cond.]$, $[nprint]$, $[tprint_1], \dots, [tprint_{45}]$, $[iprt]$

where

t_i is initial time. If t_i is set to TAPE, the initial time is read from a disk file called tape19.

$t_f < 10^{28} >$ is final time, *i.e.*, the time at which the time integration procedure will stop.

Δt_i is the initial time-step size. Δt_i also establishes the minimum time step allowed during the time integration procedure.

$no. steps < 1000 >$ is the maximum number of time steps to be taken.

init. cond. specifies the source of the initial conditions for the problem. If this parameter is omitted, initial conditions are taken from material property cards. If this parameter is set to TAPE, initial conditions are read from the file tape19, as explained below.

nprint is the number of timeplanes at which results are to be printed. Normally, results are printed at the final timeplane so the user need not request output then. If *nprint* is omitted, results will be printed only at the last timeplane of a calculation.

*tprint*₁–*tprint*₄₅ specify the times at which output is desired. A maximum of 45 times may be designated for output. The list of times must be in ascending order. Specifying one or more *tprints* will alter the selection of time-step size so that results can be computed at the exact values of time requested by the user.

iprt controls the parameters printed at the timeplanes defined above. The parameter values printed will include those described below for the defined *iprt* value in addition to those listed for lesser values of *iprt*.

<i>iprt</i>	parameter values printed
0 or blank (default)	PRESS, MSAT/TSAT (no composite/composite), MCOND11/TCOND11, MCOND22/TCOND22
1	MXFLUX/TXFLUX, MZFLUX/TZFLUX, MXVEL, MZVEL
2	if composite, MSAT, FSAT, MCOND11, MCOND22, FCOND11, FCOND22, MXFLUX, MZFLUX, FXFLUX, FZFLUX, FXVEL, FZVEL
3	PHEAD, MMC, MPERM11, MPERM22, and if composite, FMC, TMC, FPERM11, FPERM22, TPERM11, TPERM22

where the output quantities are as described below

COND	conductivity
FLUX	flux (Darcy velocity)
MC	moisture content
PERM	permeability
PRESS	effective (total) pressure
PHEAD	pressure head
SAI	saturation
VEL	pore water velocity

Modifiers

M	matrix
F	fracture
T	total (composite)

X	horizontal direction
Z	vertical direction
11	direction of first principal axis
22	direction of second principal axis

There is no need for multiple UNZIPP cards in NORIA-SP because time-step size is adjusted automatically by the program and because specification of timeplanes at which results are output is quite flexible. The UNZIPP card must be followed by an END card:

END

Initial conditions may be specified on material property cards or read from tape19, as described above. File tape19 can be created in one of two ways: it can be saved from a previous NORIA-SP run using the RESTART option described below or it can be written by another program. The appropriate format for the file is described in Subsection 6.14.

6.8 RESTART Command Card

NORIA-SP saves transient solution data on an external file (tape19) for continuing computations or postprocessing. In cases of large problems, or those with many timeplanes, this file can become large. The following command is used if it is desired not to save the solution data.

RESTART, NOSAVE

In order to restart from previously saved solution data, the following command card is used:

RESTART, RESET, [*nsteps*]

where

nsteps < 0 > indicates the timeplane number after which the solution file is to be positioned during the restart process. The solution file is rewound if *nsteps* is omitted. The solution file is positioned after the *nsteps* timeplane; the initial condition is counted as the first timeplane. Generally, if *n* time steps have already been taken in a transient analysis, the user should set *nsteps* $\geq n$ to continue the transient analysis.

The RESTART, NOSAVE command should occur after the UNZIPP command sequence. The RESTART, RESET command should immediately follow the header card.

A FORMKF command card must precede an UNZIPP command sequence when a transient solution is being continued via the restart option. In addition, *init. cond.* on the UNZIPP command card should be set to TAPE. It is usually desirable to set t_i to TAPE as well so that the initial time is automatically set to the last time computed during the previous run.

The RESTART commands direct NORIA-SP to collect solution data from the file `tape19`, which contains the solution data.

6.9 Program Termination Command Card

The appropriate termination command is a STOP command.

STOP

Another option that is available to NORIA-SP users is to produce NORIA-SP solution data in the form of an EXODUS database (Mills-Curran *et al.*, 1988) for postprocessing with graphics packages such as BLOT (Gilkey and Glick, 1989) and TRINITY II (Glick and Gartling, 1988). There are four commands to control the contents of the database: (1) the POST option on the stop command card flags the creation of an EXODUS database, (2) the SELECT command indicates which variables are to be included in the database, (3) the TIME or TIMEPLANE command identifies timeplanes to be included in the database, and (4) the END command completes the EXODUS file definition sequence. The format of these commands is shown below.

STOP, POST

SELECT, ALL

or

SELECT, *near*, *variable*₁, *variable*₂, ...

TIMEPLANE, ALL

or

TIMEPLANE, INCREMENT, *nstart*, *nstop*, *inc*

or

TIMEPLANE, SPECIFIED, *no.time*, *timeplane*₁, *timeplane*₂, ...

or

TIME, INCREMENT, *start*, *stop*, *inc*

or

TIME, SPECIFIED, *no.time*, *time*₁, *time*₂, ...

END

where

nvar is the number of alphanumeric names that follow on this line.

variable is an alphanumeric name that indicates quantities to be written to the EXODUS database. The valid responses and explanation are given below.

If variable is PRESSURE, the database includes

No fractures	Composite
PRESS,PHEAD	PRESS,PHEAD

If variable is SATURATION, the database includes

No fractures	Composite
MSAT	MSAT,FSAT,TSAT

If variable is CONDUCTIVITY, the database includes

No fractures	Composite
MCOND11,MCOND22	MCOND11,MCOND22,FCOND11,FCOND22, TCOND11,TCOND22

If variable is FLUX, the database includes

No fractures	Composite
MXFLUX,MZFLUX	MXFLUX,MZFLUX,FXFLUX,FZFLUX

If variable is VELOCITY, the database includes

No fractures	Composite
MXVEL,MZVEL	MXVEL,MZVEL,FXVEL,FZVEL,LXVEL,LZVEL

where

Physical quantities

COND conductivity

FLUX	flux (Darcy velocity)
PRESS	effective (total) pressure
PHEAD	pressure head
SAT	saturation
VEL	pore water velocity

Modifiers

M	matrix
F	fracture
T	total (composite)
X	horizontal direction
Z	vertical direction
L	larger of matrix/fracture (composite)
11	direction of principal axis
22	direction of secondary axis

nstart is the first timeplane to be included in the database.

nstop is the last timeplane to be included in the database.

inc is the increment to determine the intermediate timeplanes to be included in the database.

no. time is the number of timeplanes that follow on the input line.

timeplane_n is a list of timeplanes that are to be included in the database. A maximum of 49 timeplanes may be specified. If a timeplane number exceeds the number of solution timeplanes, the final solution timeplane will be included in the database.

start is the time of the first timeplane to be included in the database.

stop is the time of the last timeplane to be included in the database.

time is the time increment to determine the intermediate times to be included in the database.

time_n is a list of times that are to be included in the database. A maximum of 49 times may be specified. No interpolation of solution results is performed. For each time specified, the first timeplane encountered with a time greater than or equal to the specified time will be included in the EXODUS database. If a time exceeds the time of the last solution timeplane, the final solution timeplane will be included in the database.

For example, to include all variables at every twentieth timeplane over 1000 timeplanes in the EXODUS database, the following command sequence would be supplied:

```
STOP,POST  
SELECT,ALL  
TIMEPLANE,INCREMENT,1,1000,20  
END
```

Or, to include only saturation values at all timeplanes:

```
STOP,POST  
SELECT,1,SATURATION  
TIMEPLANE,ALL  
END
```

6.10 Input Deck Sequence

The order of commands in a NORIA-SP input deck is somewhat flexible. However, there are some obvious limitations to command sequence because some computations are necessary prerequisites to others. The following comments provide some guidelines.

- If SETUP, FORMKF, and UNZIP command cards are included in an input deck, they should always occur in the order listed here. However, in many cases, it is desirable to insert other command cards between these cards.
- If the RESTART, NOSAVE command is used it must follow an UNZIP command.
- If a transient calculation is being continued, the RESTART, RESET command should immediately follow the header card and should immediately precede both FORMKF and UNZIP commands. However, the SETUP command sequence may be inserted between the RESTART and the FORMKF commands. If the SETUP command sequence is included, it should be identical to the one used in the previous transient analysis.
- If OUTPUT command cards are included in the input deck, they should always follow the SETUP command card sequence and precede the UNZIP command card.
- If a STOP, POST command sequence is included in the input deck, it must follow the UNZIP command card.

6.11 User Subroutines

All problems executed using NORIA-SP require subroutines supplied by the user. NORIA-SP recognizes FLUIDC, PERM, CURVEN, and EXTDEF user-defined subroutines. Generally, only a subset of these is needed in a computation, although FLUIDC and PERM are required for all applications. Each of these subroutines is described below. NORIA-SP can use any set of units as long as they are consistent throughout the input deck and the user subroutines.

Moisture Content

Subroutine FLUIDC must be used to define material porosity and residual saturation, and to prescribe the functional dependence of moisture content. The material porosity value(s) indicate whether or not a composite (matrix and fracture) model is being applied. If the fracture "porosity" is nonzero, a composite model is assumed. FLUIDC is called once during the problem setup at the initial pressure to check for errors and set the composite model flag. The required argument list and array dimensions are shown below along with a description of each variable in the argument list. Examples of this subroutine are given in Section 7.

```
      SUBROUTINE FLUIDC (NELEM,MAT,NN,P,X,Z,TIME,RHOG,
1      CAP,THM,PHIM,SMR,THF,PHIF,SFR,TH,PHI)
      DIMENSION P(8),X(8),Z(8)
      DIMENSION CAP(8),THM(8),THF(8),TH(8)
C
C  VARIABLES PASSED TO ROUTINE:
C
C  NELEM      Element number
C  MAT        Material number of this element
C  NN         Number of nodes at which output is to be computed
C  P          Effective pressure at each node
C  X          x-coordinate value at each node
C  Z          z-coordinate value at each node
C  TIME       solution time
C  RHOG       product of water density and gravity
C
C  VARIABLES RETURNED FROM ROUTINE:
C
C  CAP        capacitance at each node
C  THM        matrix moisture content at each node
C  PHIM       matrix porosity for this element/material
C  SMR        matrix residual saturation for this element/material
C
```

```

C   Fracture and total variables need only be defined in the case of
C   a composite model (see Subsection 2.4).
C
C   THF      fracture moisture content at each node
C   PHIF     fracture "porosity" for this element/material (zero if no
C            composite model)
C   SFR      fracture residual saturation for this element/material
C   TH       total moisture content at each node
C   PHI      total "porosity" for this element/material
C

```

```

      RETURN
      END

```

Permeabilities

Subroutine PERM is used to prescribe the functional dependence of the permeabilities. PERM is called during the problem setup at the initial pressure to assure nonnegative values of permeability. The argument list and array dimensions are shown below along with a description of the variables. Examples of PERM are given in Subsections 7.1 and 7.2.

```

      SUBROUTINE PERM(NELEM,MAT,NN,P,X,Z,TIME,RHOG,
1                   AKM11,AKM22,AKF11,AKF22,AK11,AK22)
C
C      DIMENSION P(8),X(8),Z(8)
C      DIMENSION AKM11(8),AKM22(8),AKF11(8),AKF22(8),AK11(8),AK22(8)
C
C      VARIABLES PASSED TO ROUTINE:
C
C      NELEM    Element number
C      MAT      Material number of this element
C      NN       Number of nodes at which output is to be computed
C      P        Effective pressure at each node
C      X        x-coordinate value at each node
C      Z        z-coordinate value at each node
C      TIME     solution time
C      RHOG     product of water density and gravity
C

```

```

C  VARIABLES RETURNED FROM ROUTINE:
C
C  AKM11    matrix permeability in the first principal-axis direction
C           at each node
C  AKM11    matrix permeability in the second principal-axis direction
C           at each node
C
C  Fracture and total variables need only be defined in the case of
C  a composite model (see Subsection 2.4).
C
C  AKF11    fracture permeability in the first principal-axis direction
C           at each node
C  AKF11    fracture permeability in the second principal-axis direction
C           at each node
C  AK11     total permeability in the first principal-axis direction
C           at each node
C  AK11     total permeability in the second principal-axis direction
C           at each node
C
C           .
C           .
C           .

```

RETURN

END

Boundary Conditions

When boundary condition options PEEFVARY or ULVARY are used, the user must supply one or more CURVEN subroutines to define how the boundary condition depends on the independent or dependent variables. When option PEEFVARY is used, the subroutine returns effective pressures (P); when option ULVARY is used, the subroutine returns Darcy flux (VALUE). NORIA-SP will accept up to 6 CURVEN subroutines (n can be 1 through 6). The value of n is prescribed by the *value/curve no.* parameter on the BC card. CURVEN subroutines have the following form:

```

SUBROUTINE CURVEN(NELEM,MAT,NN,P,TIME,VALUE)
  DIMENSION P(8),X(8),Z(8)
  DIMENSION ISIDE1(3),ISIDE2(3),ISIDE3(3),ISIDE4(3)
  DATA (ISIDE1(I),I=1,3),(ISIDE2(I),I=1,3),(ISIDE3(I),I=1,3),
1      (ISIDE4(I),I=1,3) /1,5,2,2,6,3,3,7,4,4,8,1/

```

```

C
C  NATURAL BOUNDARY CONDITIONS:

```

```

C      THIS SUBROUTINE CALCULATES DARCY FLUX
C      OF LIQUID WATER,  IN UNITS OF M/S
C      FLUXES ARE POSITIVE IN
C      THE DIRECTION OF THE OUTWARD POINTING
C      NORMAL TO THE ELEMENT SIDE.  FOR
C      EXAMPLE
C
      VALUE=0.0
C
C      ESSENTIAL BOUNDARY CONDITIONS:  THIS
C      SUBROUTINE PLACES VALUES DIRECTLY
C      IN THE VECTORS CONTAINING THE
C      DEPENDENT VARIABLES.  FOR EXAMPLE
C
      DO 10 I=1,3
      P(ISIDE1(I))=F(TIME)
10  CONTINUE
      RETURN
      END

```

where

VALUE is the current value assigned for a flux-type boundary condition.

ISIDE_n is a vector that defines the node points that lie on side *n* of a quadrilateral element. These vectors are useful for assigning essential boundary conditions.

When essential boundary conditions are being assigned, *i.e.*, using PEFVARY, the boundary condition values are inserted directly in the appropriate vector of dependent variables as illustrated above. In the illustration, F(TIME) represents any function of time, but it could equally well be a function of any or all of the independent or dependent variables.

Mesh Generation

When an EXTDEF card is inserted in the nodal-point generation portion of the SETUP sequence of data cards, NORIA-SP calls the user-defined subroutine EXTDEF. This subroutine provides a great deal of flexibility in defining node point locations. EXTDEF has the following form:

```

SUBROUTINE EXTDEF(X,Z,MAXI)
DIMENSION X(MAXI,1),Z(MAXI,1)

```

```

C
C   THIS SUBROUTINE DEFINES NODAL COORDINATES,
C   X AND Z FOR EACH REQUIRED NODE POINT
C
      IMAX=11
      JMAX=21
      DO 10 I=1,IMAX
      DO 10 J=1,JMAX
      X(I,J)=F1(I,J)
      Z(I,J)=F2(I,J)
10  CONTINUE
      RETURN
      END

```

where

X is an array of nodal x-coordinates. The two subscripts are the (I,J) label for a given node point.

Z is an array of nodal z-coordinates. The two subscripts are the (I,J) label for a given node point.

MAXI is an integer that specifies the largest I value that can be used in the mesh. The value of MAXI is specified on the SETUP command card.

IMAX in this illustration is the number of I-rows in the problem domain.

JMAX in this illustration is the number of J-rows in the problem domain.

F1, F2 are arbitrary functions that determine the x- and z-coordinates for each (I,J) pair.

6.12 Initial Conditions

Initial conditions that can be prescribed as constant pressure over a given material may be specified on the solid material property cards. Nonuniform initial conditions may be supplied to NORIA-SP by use of the RESTART option (Subsection 6.8) and creating an external file (tape19). The statement in NORIA-SP that reads initial data from tape19 has the form:

```

      READ (19) TIME, PMAX, PMIN, NUMEL, ((US(I,J), I=1,8), J=1, NUMEL)

```

where

TIME is the value of time corresponding to the initial condition.

PMAX, PMIN are the respective maximum and minimum values of effective liquid pressure in the field.

NUMEL is the total number of elements in the domain.

US is a matrix that contains the effective pressures at each node of each element. The first subscript indicates local node number; the second subscript indicates element number.

6.13 Error Messages

NORIA-SP contains a host of error checks that test for bad input data, improper dimensions, *etc.* When an error is encountered, an error message is printed and execution is usually terminated with a STOP 1. Error messages in NORIA-SP are listed below.

- BCTIME-TIME CURVE NUMBER TOO LARGE—A time curve number greater than six was encountered.
- CONNECT—An invalid nodal point number was found in constructing the connectivity array for output to the EXODUS file.
- CONNECT—The maximum number of element blocks was exceeded. Either reduce the number of element blocks or increase MAXBLK.
- CONNECT—The maximum number of nodes was exceeded. Either reduce the number of elements or increase MAXGNOD.
- DRIVER-UNRECOGNIZED COMMAND—A command was either misspelled or was expected and not found.
- ELDATA-BC APPLIED TO AN UNDEFINED ELEMENT—A boundary condition was applied to an element not yet defined.
- ELDATA-BC ON IMPROPER SIDE OF ELEMENT—A boundary condition was specified for an improper element side.
- ELDATA-EXCESSIVE BOUNDARY CONDITIONS ON ELEMENT—More than eight boundary conditions have been applied to an element.
- ELDATA-INVALID FRACTURE PERMEABILITY (AKF11)—A negative value of fracture permeability in the direction of the principal axis was returned from PERM for the indicated material.

- **ELDATA-INVALID FRACTURE PERMEABILITY (AKM22)**—A negative value of fracture permeability in the direction of the secondary axis was returned from PERM for the indicated material.
- **ELDATA-INVALID FRACTURE MOISTURE CONTENT**—A negative value of fracture moisture content, or a value greater than the fracture porosity, was returned from FLUIDC for the indicated material.
- **ELDATA-INVALID MATRIX MOISTURE CONTENT**—A negative value of matrix moisture content, or a value greater than the matrix porosity, was returned from FLUIDC for the indicated material.
- **ELDATA-INVALID MATRIX PERMEABILITY (AKM11)**—A negative value of matrix permeability in the direction of the principal axis was returned from PERM for the indicated material.
- **ELDATA-INVALID MATRIX PERMEABILITY (AKM22)**—A negative value of matrix permeability in the direction of the secondary axis was returned from PERM for the indicated material.
- **ELDATA-INVALID MATRIX POROSITY**—A negative value of matrix porosity was returned from FLUIDC for the indicated material.
- **ELDATA-INVALID TOTAL MOISTURE CONTENT**—A negative value of total moisture content, or a value greater than the total porosity, was returned from FLUIDC for the indicated material.
- **ELDATA-INVALID TOTAL PERMEABILITY (AK11)**—A negative value of total permeability in the direction of the principal axis was returned from PERM for the indicated material.
- **ELDATA-INVALID TOTAL PERMEABILITY (AK22)**—A negative value of total permeability in the direction of the secondary axis was returned from PERM for the indicated material.
- **ELDATA-INVALID TOTAL POROSITY**—A negative value of total porosity was returned from FLUIDC for the indicated material.
- **ELDATA-LOOP PREVIOUSLY DEFINED**—Error in specification of looping. Check for third loop within two existing loops.
- **ELDATA-MAXIMUM NUMBER OF ELEMENTS EXCEEDED**—the dimension on the allowable number of elements has been exceeded. Reduce the number of elements used or redimension code.
- **ELDATA-UNDEFINED MATERIAL NUMBER**—a material number was used on an element specification card that was not defined in the material property/initial condition portion of the input deck.

- **ELDATA-UNRECOGNIZED BOUNDARY CONDITION**—Boundary condition type is in error. Check spelling.
- **ELDATA-UNRECOGNIZED COMMAND**—Erroneous element, boundary condition, or looping specification. Check spelling.
- **EXOWRT-UNRECOGNIZED WORD IN POST COMMAND**—Invalid option used in the POST command. Check spelling.
- **FFLD-END OF DATA**—An end-of-file was encountered on the input file. Check termination card.
- **FFLDSB-INPUT VARIABLE TOO LONG**—An input variable with more than ten characters was encountered.
- **FORMKFA-BAD ELEMENT JACOBIAN**—A negative element area was found. Check element coordinates and connectivity.
- **FORMKFA-TOO MANY TIME-VARYING BCS**—More than six time-varying boundary conditions were specified.
- **FORMKFP-BAD ELEMENT JACOBIAN**—A negative element area was found. Check element coordinates and connectivity.
- **FORMKFP-TOO MANY TIME-VARYING BCS**—More than six time-varying boundary conditions were specified.
- **MATREAD-MATERIAL NO. TOO LARGE**—a material was assigned a number greater than the allowable 2 through 13 for solid materials.
- **MATREAD-TOO MANY MATERIALS SPECIFIED**—More than 12 solid materials were specified.
- **NMESH-IJ MAX OR MIN EXCEEDS SPECIFIED VALUE**—A grid point was found with an i_{max} , j_{max} , i_{min} , or j_{min} that exceeded the specified value on the SETUP card.
- **PRESOLN-INSUFFICIENT STORAGE**—Insufficient storage for element connectivity. Reduce problem size or redimension code.
- **PRESOLN-INSUFFICIENT STORAGE FOR FRONT WIDTH**—Reduce problem size or redimension code.
- **PRINTER-UNRECOGNIZED COMMAND**—An OUTPUT command was used with an incomplete or misspelled parameter list.
- **QUAD-ZERO JACOBIAN**—A quadrilateral element with a zero area was found. Check element coordinates.

- **RESTART-UNRECOGNIZED COMMAND**—A restart command card was used with an incomplete or misspelled parameter list.
- **UNZIPP-COMMAND CARD MISSING**—A termination card is missing.
- **UNZIPP-MAXIMUMSTORAGE EXCEEDED**—Maximum active storage exceeded. Reduce problem size or redimension code.
- **UNZIPP-MISMATCH ON NUMBER OF ELEMENTS ON TAPE19**—The number of elements read from TAPE19 for initial conditions does not match the number of elements defined in the setup section of the input deck.
- **UNZIPP-TIME STEP SIZE TOO SMALL**—Time integration procedure attempted to use a time step smaller than the user-specified initial time step.
- **UNZIPP-ZERO PIVOT**—A zero was found on the diagonal of the equation being eliminated. Equation system is singular or element connectivity is in error.

6.14 Computer Requirements and Control Cards

The original version of NORIA was written for execution on the CRAY mainframes. NORIA-SP compiles with both the cft and cft77 compilers under CTSS and with the cf77 compiler under UNICOS. Minor changes will also allow a double precision version to compile on a VAX and other similar machines; however, the allowable problem size may need to be reduced (currently defined to be 5,000 elements). This reduction involves changing values defined in parameter statements.

The central processor time needed to run NORIA-SP is somewhat difficult to predict because, depending on the difficulty of the problem, more than one Newton iteration may be required per time step, and time steps are repeated when errors are too large. Therefore, it is generally advisable to set the time limit for a calculation to be a factor of one and one-half to two higher than would be expected for a specified number of time steps, especially when embarking on a new problem. As an example, on the Cray Y-MP, NORIA-SP requires about 10 seconds for compilation and start-up, and about 0.006 seconds per element per time step for solution of the differential equations. An additional 10 or 20 seconds should be added to the estimated execution time if significant postprocessing is to be performed, such as computing velocities.

The steps described below for code execution are generic (they are necessary for executing NORIA-SP on any computer system).

1. The NORIA-SP object file must be available either directly or by compiling the source code.

2. Your user subroutines must be compiled. Subroutines FLUIDC and PERM must be supplied. Some operating systems will not produce an executable with unresolved external references (*e.g.*, missing subroutines) in which case the user subroutines not needed for your particular application (*i.e.*, EXTDEF and CURVEN) must be supplied (they need not contain any executable statements). It is preferable to include all the user subroutines in one file.
3. To create the executable, the main program from step 1 must be linked with the user subroutines in step 2. Unresolved external references, if allowable, should only include the subroutines EXTDEF and CURVEN (where n is 1 through 6). Other system-supplied subroutines used in NORIA-SP are listed below.
 - (a) SAXPY -- adds a scalar multiple of a vector to another vector.
 - (b) SCOPY -- copies a vector into another vector.
 - (c) SDOT -- computes a dot product of two vectors.
 - (d) SECOND -- returns elapsed cpu time in seconds.
4. To execute, you must have your input deck in a local file named norin. The output listing will be in the local file norout, and if an EXODUS file is created, it will be called norex. The solution file is named tape19.

The steps described above are shown below for UNICOS assuming that the source code for NORIA-SP is in the file noriasp.f and all user-supplied subroutines are in the file usersub.f. The executable will be named noriasp.x.

1. Compile the main program: **cf77 -b noriasp.o noriasp.f**
2. Compile the user subroutines: **cf77 -b usersub.o usersub.f**
3. Create the executable: **cf77 -o noriasp.x noriasp.o,usersub.o**
4. Run the program: **noriasp.x**

7 Sample Calculations

A sample calculation is provided in this section to demonstrate the command cards described in Section 6. This example illustrates some, but not all, of the salient features of NORIA-SP. User manuals for SAGUARO (Eaton *et al.*, 1983), NORIA (Bixler, 1984), and MARIAH (Gartling and Hickox, 1980) may help familiarize the user with some of the features of NORIA-SP not demonstrated here.

7.1 Sample Problem 1

This sample allows comparison of results with the analytical solution presented by Fleming (1986). The problem consists of downward infiltration into an initially unsaturated, vertical, one-dimensional column of porous media. The prescribed time-dependent infiltration rate is defined in subroutine CURVEI. Saturation and conductivity are exponential functions of pore pressure and are calculated in subroutines FLUIDC and PERM respectively.

User Subroutines for Sample Problem 1

The user subroutine CURVEI is supplied in addition to the required FLUIDC and PERM subroutines to define the time-dependent boundary condition.

```
      subroutine fluidc(nelem,mat,nn,p,x,y,time,rhog,
1          cap,thm,phim,smr,thf,phif,sfr,th,phi)
c
c      dimension p(8),x(8),y(8),cap(8),thm(8),thf(8),th(8)
c
c      this subroutine calculates the moisture content and capacitance
c      units - kg,m,s
c
c      material 2, as specified by the Fleming formula
c
c      variables passed to routine:
c
c      nelem    element number
c      mat      material number of this element
c      nn       number of nodes at which output is to be computed
c      p        effective pressure at each node: rhog(phi + x)
c      x        x-coordinate value at each node
c      y        y-coordinate value at each node
c      time     solution time
c      rhog     product of water density and gravity
c
```

```

c  VARIABLES RETURNED FROM ROUTINE:
c
c  cap      capacitance at each node
c  thm      matrix moisture content at each node
c  phim     matrix porosity for this element/material
c  smr      matrix residual saturation for this element/material
c
c  Fracture and total variables need only be defined in the case of
c  a composite model.
c
c  thf      fracture moisture content at each node
c  phif     fracture 'porosity' for this element/material (zero if no
c           composite model)
c  sfr      fracture residual saturation for this element/material
c  th       total moisture content at each node
c  phi      total 'porosity' for this element/material (= phim if no
c           composite model)
c
c  Other variables used:
c
c  cond      matrix material conductivity
c  ph        matrix porosity
c  alpha,an,ce  model parameters
c
c  data  alpha,  an,  cond,  ce,  ph
c  1      /0.0046, 0.652, 1.9e-11, 2.7182818, 0.11 /
c for matrix material only ( no fractures) only define phim, smr, thm, and cap
c  phim=ph
c  smr = 0.0
c  do 100 j=1,nn
c  phm=p(j)/rhog - y(j)
c  if (phm.gt.-.000001) phm=-.000001
c  thm(j)=phim*ce**(alpha*phm)
c for composite model also define phif, sfr, phi, thf and th
c  phif = 1e-20
c  sfr = 0.0
c  phi = ph
c  thf(j)=1.0e-20
c  th(j)=thm(j)
c  compute the derivative of moisture content wrt psi / rhog
c  cap(j)=alpha*thm(j)/rhog
c  cap(j)=alpha*th(j)/rhog
c  100 continue

```

```

c
c      return
c      end
c
c
c      subroutine perm(nelem,mat,nn,p,x,y,time,rhog,
1          akm11,akm22,akf11,akf22,ak11,ak22)
c
c      dimension p(8),x(8),y(8),akm11(8),akm22(8),akf11(8),akf22(8),
1          ak11(10),ak22(8)
c
c      calculates permeability (m**2)
c      units - kgm, m,sec.
c
c
c      Variables passed to Routine:
c
c      nelem  Element number
c      mat    Material number of this element
c      nn     Number of nodes at which output is to be computed
c      p      Effective pressure at each node
c      x      x-coordinate value at each node
c      y      y-coordinate value at each node
c      time   solution time
c      rhog   product of water density and gravity
c
c      Variables returned from routine:
c
c      akm11  matrix permeability in the first principal-axis direction
c             at each node
c
c      akm22  matrix permeability in the second principal-axis direction
c             at each node
c
c      Fracture and total variables need only be defined in the
c      case of a composite model
c
c      akf11  fracture permeability in the first principal-axis
c             direction at each node
c
c      akf22  fracture permeability in the second principal-axis
c             direction at each node
c

```

```

c      ak11  total permeability in the first principal-axis
c            direction at each node
c
c      ak22  total permeability in the second principal-axis
c            direction at each node
c
c      Other variables used:
c
c      cond          matrix material conductivity
c      ph            matrix porosity
c      alpha,an,ce   model parameters
c
c      data  alpha,    an,    cond,        ce,    ph
1          /0.0046, 0.652, 1.9e-11, 2.7182818, 0.11 /
c      remember to change this data statement in subroutine curve1
      perms = cond*0.001/rhog
      do 20 j=1,nn
      phm=p(j)/rhog - y(j)
      if(phm .gt. -.000001) phm=-.000001
      akm11(j)=perms*ce** (alpha*(an+1)*phm)
      akm22(j)=akm11(j)
c for composite model (phif not equal to zero)
      akf11(j) = 1e-40
      akf22(j) = 1e-40
      ak11(j)  = akm11(j)
      ak22(j)  = akm22(j)
20  continue
      return
      end
c
      subroutine curve1 (nelem,mat,nn,p,time,value)
      dimension p(8)
      dimension iside1(3),iside2(3),iside3(3),iside4(3)
      data (iside1(i),i=1,3),(iside2(i),i=1,3),(iside3(i),i=1,3),
1        (iside4(i),i=1,3) /1,5,2,2,6,3,3,7,4,4,8,1/
      data  alpha,    an,    cond,        ce, ph
1        /0.0046, 0.652, 1.9e-11, 2.7182818, 0.11 /
c
      alambda = cond/ph
      value    = - cond*(1-ce**(-alpha*an*alambda*time))**(1/an)
      return
      end

```

Input Deck for Sample Problem 1

```
$ 1-d comparison with analytical model; SI units
setup , 1,3
water, 1, 1000.0, 0.001, 9.8
rock , 2, -2.45e7, 0.0
end
1,1,3,41          $ reg 1
0,1,1,0
-400,-400,0,0
end
jloop,20,2
quad8/4,2,1,1
jend
bc,ulvary ,1,39,3,1          $ velocity at left top
end
velocity,nodal
formkf ,
output,single,1,2,3,4,5,15,16,17,18,19,20
unzipp,0.0,.180e13,1.0e-4,2000, ,4,.0,.06e13,.12e13,.18e13
end
stop,post
select,1,saturation
times,specified,3,0.06e13,.12e13,0.18e13
end
```

Output File for Sample Problem 1

A total of 273 seconds of Cray X-MP computer time was required to compute this problem to a time of 1.8×10^{12} s. The total output file for this problem is too long for inclusion in this report. A partial listing is included. Most of the output nomenclature is self-descriptive; less apparent are rms, defined by equation 4.7, and newit, the number of Newton iterations.

Output for Sample Problem 1 (cont. 1)

the two-dimensional planar version of the element formulation was requested for this problem

time in formkf = 0.778 seconds

time in pre-front= 0.00777

properties of matrix system—

number of degrees of freedom— 103

max front width— 8

storage for max front— 152

buffer space— 194847

pressure fields were initialized by material with the following values—

material no. 2 initial effective pres= -0.2450000e+08

output for the transient problem was obtained over the interval—

initial time= 0.0000000e+00

final time= 0.1800000e+13

initial time step= 0.1000000e-03

integration was terminated when current time.ge.final time or when 2000 time steps were taken

=====

initial conditions are the solution at time step no. = 1

```

current time= 0.1000000e-03    time step= 0.1000000e-03    time step no.= 2
rms = 0.00000
max change in effec liq pres between time steps is 0.1410325e-18 in element 1 node 2

current time= 0.2000000e-03    time step= 0.1000000e-03    time step no.= 3
rms = 0.00000
max change in effec liq pres between time steps is 0.1410325e-18 in element 1 node 2

current time= 0.3000000e-03    time step= 0.1000000e-03    time step no.= 4
rms = 0.00000
newit= 1    rms= 0.00000
max change in effec liq pres between time steps is 0.1384989e-18 in element 1 node 2

current time= 0.1032280e+09    time step= 0.1032280e+09    time step no.= 5
rms = 0.00002
newit= 1    rms= 0.00002
max change in effec liq pres between time steps is 0.4463302e+03 in element 20 node 7

current time= 0.2533078e+09    time step= 0.1500798e+09    time step no.= 6
rms = 0.00005    rmsnew= 0.00005
newit= 1
max change in effec liq pres between time steps is 0.1447339e+04 in element 20 node 7

current time= 0.4334651e+09    time step= 0.1801572e+09    time step no.= 7
rms = 0.00009
newit= 1    rms= 0.00009
max change in effec liq pres between time steps is 0.2537220e+04 in element 20 node 7

current time= 0.5978903e+12    time step= 0.2984525e+10    time step no.= 614
rms = 0.00011
newit= 1    rms= 0.00011
max change in effec liq pres between time steps is 0.1663757e+04 in element 14 node 1

```

Output for Sample Problem 1 (cont. 2)

current time= 0.6000000e+12 time step= 0.2109661e+10 time step no.= 615
 rms = 0.00006
 newit= 1 rms= 0.00006
 max change in effec liq pres between time steps is 0.9616870e+03 in element 14 node 2

solution fields—

	—nodes—					
	-1-	-2-	-3-	-4-	-5-	-6-
element 1						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.6378453e-04	0.6378453e-04	0.5817820e-04	0.5817820e-04	0.6378453e-04	0.6091690e-04
TCOND11	0.2229171e-17	0.2229171e-17	0.1914862e-17	0.1914862e-17	0.2229171e-17	0.2229171e-17
TCOND22	0.2229171e-17	0.2229171e-17	0.1914862e-17	0.1914862e-17	0.2229171e-17	0.2229171e-17
element 2						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.5817820e-04	0.5817820e-04	0.5306463e-04	0.5306463e-04	0.5817820e-04	0.5556262e-04
TCOND11	0.1914862e-17	0.1914862e-17	0.1644871e-17	0.1644871e-17	0.1914862e-17	0.1914862e-17
TCOND22	0.1914862e-17	0.1914862e-17	0.1644871e-17	0.1644871e-17	0.1914862e-17	0.1914862e-17
element 3						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.5306463e-04	0.5306463e-04	0.4840053e-04	0.4840053e-04	0.5306463e-04	0.5067895e-04
TCOND11	0.1644871e-17	0.1644871e-17	0.1412947e-17	0.1412947e-17	0.1644871e-17	0.1644871e-17
TCOND22	0.1644871e-17	0.1644871e-17	0.1412947e-17	0.1412947e-17	0.1644871e-17	0.1644871e-17
element 4						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.4840053e-04	0.4840053e-04	0.4414637e-04	0.4414637e-04	0.4840053e-04	0.4622453e-04
TCOND11	0.1412947e-17	0.1412947e-17	0.1213724e-17	0.1213724e-17	0.1412947e-17	0.1412947e-17
TCOND22	0.1412947e-17	0.1412947e-17	0.1213724e-17	0.1213724e-17	0.1412947e-17	0.1412947e-17
element 5						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.4414637e-04	0.4414637e-04	0.4026613e-04	0.4026613e-04	0.4414637e-04	0.4216163e-04
TCOND11	0.1213724e-17	0.1213724e-17	0.1042592e-17	0.1042592e-17	0.1213724e-17	0.1213724e-17
TCOND22	0.1213724e-17	0.1213724e-17	0.1042592e-17	0.1042592e-17	0.1213724e-17	0.1213724e-17
element 15						
PRESS	-0.1745958e+08	-0.1745958e+08	-0.1252806e+08	-0.1252806e+08	-0.1745958e+08	-0.1521631e+08
TSAT	0.4792370e-03	0.4792370e-03	0.4424857e-02	0.4424857e-02	0.4792370e-03	0.1311806e-02
TCOND11	0.6237676e-16	0.6237676e-16	0.2453461e-14	0.2453461e-14	0.6237676e-16	0.6237676e-16
TCOND22	0.6237676e-16	0.6237676e-16	0.2453461e-14	0.2453461e-14	0.6237676e-16	0.6237676e-16
element 16						
PRESS	-0.1252806e+08	-0.1252806e+08	-0.9442894e+07	-0.9442894e+07	-0.1252806e+08	-0.1085159e+08
TSAT	0.4424857e-02	0.4424857e-02	0.1717366e-01	0.1717366e-01	0.4424857e-02	0.9282678e-02
TCOND11	0.2453461e-14	0.2453461e-14	0.2305403e-13	0.2305403e-13	0.2453461e-14	0.2453461e-14
TCOND22	0.2453461e-14	0.2453461e-14	0.2305403e-13	0.2305403e-13	0.2453461e-14	0.2453461e-14
element 17						
PRESS	-0.9442894e+07	-0.9442894e+07	-0.7498882e+07	-0.7498882e+07	-0.9442894e+07	-0.8369509e+07
TSAT	0.1717366e-01	0.1717366e-01	0.3901210e-01	0.3901210e-01	0.1717366e-01	0.2714547e-01
TCOND11	0.2305403e-13	0.2305403e-13	0.8941593e-13	0.8941593e-13	0.2305403e-13	0.2305403e-13
TCOND22	0.2305403e-13	0.2305403e-13	0.8941593e-13	0.8941593e-13	0.2305403e-13	0.2305403e-13
element 18						
PRESS	-0.7498882e+07	-0.7498882e+07	-0.6164404e+07	-0.6164404e+07	-0.7498882e+07	-0.6778519e+07
TSAT	0.3901210e-01	0.3901210e-01	0.6657037e-01	0.6657037e-01	0.3901210e-01	0.5224799e-01
TCOND11	0.8941593e-13	0.8941593e-13	0.2161791e-12	0.2161791e-12	0.8941593e-13	0.8941593e-13
TCOND22	0.8941593e-13	0.8941593e-13	0.2161791e-12	0.2161791e-12	0.8941593e-13	0.8941593e-13
element 19						
PRESS	-0.6164404e+07	-0.6164404e+07	-0.5155449e+07	-0.5155449e+07	-0.6164404e+07	-0.5629643e+07
TSAT	0.6657037e-01	0.6657037e-01	0.9749981e-01	0.9749981e-01	0.6657037e-01	0.8171765e-01
TCOND11	0.2161791e-12	0.2161791e-12	0.4060578e-12	0.4060578e-12	0.2161791e-12	0.2161791e-12
TCOND22	0.2161791e-12	0.2161791e-12	0.4060578e-12	0.4060578e-12	0.2161791e-12	0.2161791e-12
element 20						
PRESS	-0.5155449e+07	-0.5155449e+07	-0.4340105e+07	-0.4340105e+07	-0.5155449e+07	-0.4728694e+07
TSAT	0.9749981e-01	0.9749981e-01	0.1303944e+00	0.1303944e+00	0.9749981e-01	0.1137683e+00
TCOND11	0.4060578e-12	0.4060578e-12	0.6563888e-12	0.6563888e-12	0.4060578e-12	0.4060578e-12
TCOND22	0.4060578e-12	0.4060578e-12	0.6563888e-12	0.6563888e-12	0.4060578e-12	0.4060578e-12

Output for Sample Problem 1 (cont. 3)

current time= 0.120000e+13 time step= 0.363959e+09 time step no.= 860
 rmap= 0.00000 rmsnew= 0.00000
 newit= 1 rms= 0.00000
 max change in effect liq pres between time steps is 0.5194159e+02 in element 9 node 2

solution fields—

	---nodes---					
	-1-	-2-	-3-	-4-	-5-	-6-
element 1						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.6378453e-04	0.6378453e-04	0.5817821e-04	0.5817821e-04	0.6378453e-04	0.6091690e-04
TCOND11	0.2229171e-17	0.2229171e-17	0.1914863e-17	0.1914863e-17	0.2229171e-17	0.2229171e-17
TCOND22	0.2229171e-17	0.2229171e-17	0.1914863e-17	0.1914863e-17	0.2229171e-17	0.2229171e-17
element 2						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.5817821e-04	0.5817821e-04	0.5306473e-04	0.5306473e-04	0.5817821e-04	0.5556261e-04
TCOND11	0.1914863e-17	0.1914863e-17	0.1644876e-17	0.1644876e-17	0.1914863e-17	0.1914863e-17
TCOND22	0.1914863e-17	0.1914863e-17	0.1644876e-17	0.1644876e-17	0.1914863e-17	0.1914863e-17
element 3						
PRESS	-0.2450000e+08	-0.2450000e+08	-0.2449996e+08	-0.2449996e+08	-0.2450000e+08	-0.2450000e+08
TSAT	0.5306473e-04	0.5306473e-04	0.4840144e-04	0.4840144e-04	0.5306473e-04	0.5067888e-04
TCOND11	0.1644876e-17	0.1644876e-17	0.1412991e-17	0.1412991e-17	0.1644876e-17	0.1644876e-17
TCOND22	0.1644876e-17	0.1644876e-17	0.1412991e-17	0.1412991e-17	0.1644876e-17	0.1644876e-17
element 4						
PRESS	-0.2449996e+08	-0.2449996e+08	-0.2449996e+08	-0.2449996e+08	-0.2449996e+08	-0.2450003e+08
TSAT	0.4840144e-04	0.4840144e-04	0.4415462e-04	0.4415462e-04	0.4840144e-04	0.4622386e-04
TCOND11	0.1412991e-17	0.1412991e-17	0.1214099e-17	0.1214099e-17	0.1412991e-17	0.1412991e-17
TCOND22	0.1412991e-17	0.1412991e-17	0.1214099e-17	0.1214099e-17	0.1412991e-17	0.1412991e-17
element 5						
PRESS	-0.2449996e+08	-0.2449996e+08	-0.2449614e+08	-0.2449614e+08	-0.2449996e+08	-0.2450031e+08
TSAT	0.4415462e-04	0.4415462e-04	0.4033910e-04	0.4033910e-04	0.4415462e-04	0.4215547e-04
TCOND11	0.1214099e-17	0.1214099e-17	0.1045715e-17	0.1045715e-17	0.1214099e-17	0.1214099e-17
TCOND22	0.1214099e-17	0.1214099e-17	0.1045715e-17	0.1045715e-17	0.1214099e-17	0.1214099e-17
element 15						
PRESS	-0.6004220e+07	-0.6004220e+07	-0.5221809e+07	-0.5221809e+07	-0.6004220e+07	-0.5595813e+07
TSAT	0.1036944e+00	0.1036944e+00	0.1365516e+00	0.1365516e+00	0.1036944e+00	0.1199589e+00
TCOND11	0.4495533e-12	0.4495533e-12	0.7083757e-12	0.7083757e-12	0.4495533e-12	0.4495533e-12
TCOND22	0.4495533e-12	0.4495533e-12	0.7083757e-12	0.7083757e-12	0.4495533e-12	0.4495533e-12
element 16						
PRESS	-0.5221809e+07	-0.5221809e+07	-0.4554275e+07	-0.4554275e+07	-0.5221809e+07	-0.4876124e+07
TSAT	0.1365516e+00	0.1365516e+00	0.1703806e+00	0.1703806e+00	0.1365516e+00	0.1533868e+00
TCOND11	0.7083757e-12	0.7083757e-12	0.1021080e-11	0.1021080e-11	0.7083757e-12	0.7083757e-12
TCOND22	0.7083757e-12	0.7083757e-12	0.1021080e-11	0.1021080e-11	0.7083757e-12	0.7083757e-12
element 17						
PRESS	-0.4554275e+07	-0.4554275e+07	-0.3968326e+07	-0.3968326e+07	-0.4554275e+07	-0.4252600e+07
TSAT	0.1703806e+00	0.1703806e+00	0.2046031e+00	0.2046031e+00	0.1703806e+00	0.1874734e+00
TCOND11	0.1021080e-11	0.1021080e-11	0.1381594e-11	0.1381594e-11	0.1021080e-11	0.1021080e-11
TCOND22	0.1021080e-11	0.1021080e-11	0.1381594e-11	0.1381594e-11	0.1021080e-11	0.1021080e-11
element 18						
PRESS	-0.3968326e+07	-0.3968326e+07	-0.3443082e+07	-0.3443082e+07	-0.3968326e+07	-0.3699096e+07
TSAT	0.2046031e+00	0.2046031e+00	0.2387972e+00	0.2387972e+00	0.2046031e+00	0.2217266e+00
TCOND11	0.1381594e-11	0.1381594e-11	0.1783438e-11	0.1783438e-11	0.1381594e-11	0.1381594e-11
TCOND22	0.1381594e-11	0.1381594e-11	0.1783438e-11	0.1783438e-11	0.1381594e-11	0.1381594e-11
element 19						
PRESS	-0.3443082e+07	-0.3443082e+07	-0.2964639e+07	-0.2964639e+07	-0.3443082e+07	-0.3198685e+07
TSAT	0.2387972e+00	0.2387972e+00	0.2726503e+00	0.2726503e+00	0.2387972e+00	0.2557836e+00
TCOND11	0.1783438e-11	0.1783438e-11	0.2220111e-11	0.2220111e-11	0.1783438e-11	0.1783438e-11
TCOND22	0.1783438e-11	0.1783438e-11	0.2220111e-11	0.2220111e-11	0.1783438e-11	0.1783438e-11
element 20						
PRESS	-0.2964639e+07	-0.2964639e+07	-0.2523290e+07	-0.2523290e+07	-0.2964639e+07	-0.2739811e+07
TSAT	0.2726503e+00	0.2726503e+00	0.3059293e+00	0.3059293e+00	0.2726503e+00	0.2893744e+00
TCOND11	0.2220111e-11	0.2220111e-11	0.2685341e-11	0.2685341e-11	0.2220111e-11	0.2220111e-11
TCOND22	0.2220111e-11	0.2220111e-11	0.2685341e-11	0.2685341e-11	0.2220111e-11	0.2220111e-11

As a partial verification of the NORIA-SP code, the results of this sample are compared with the analytical solution of Fleming in Figure 7.1.

7.2 Sample Problem 2

This problem was part of the Performance Assessment Computational Exercise (PACE) (Barnard and Dockery, 1991 and Dykhuizen *et al.*, 1991). The object of the PACE exercise was to develop performance-assessment expertise among the participants of the Yucca Mountain underground nuclear waste project. The geometry for Sample Problem 2 consists of a two-dimensional cross section of Yucca Mountain. The initial state (zero infiltration) is perturbed at the top boundary with an influx of water of -3.1745×10^{-13} m/s (0.01 mm/yr). The original calculations required 10 hours of computer time on the CRAY X-MP Computer (1,090 time steps). To keep the computer requirements for Sample Problem 2 at a reasonable level, calculations were limited to 20 time steps. Nine different material regions extending from the local water table to near the surface of the mountain were modeled. Additional information on this problem can be found in Dykhuizen *et al.*, 1991.

User Subroutines for Sample Problem 2

The water infiltration at the top of the region is specified to be constant. No user subroutine is required for this quantity, as in sample 1; instead, it is specified directly in the input deck. The material permeability and moisture capacitance are functions of pore pressure as defined by subroutines FLUIDC and PERM for the nine geologic units.

```

      subroutine fluidc(nelem,mat,nn,p,x,z,time,rhog,
1          cap,thm,phim1,smr1,thf,phif1,sfr,th,phi)
c
      dimension p(8),x(8),z(8),cap(8),thm(8),thf(8),th(8)
      dimension smr(13),phim(13),phif(13),alpm(13),betm(13),alpp(13),
1          fc(13)
c
c      this subroutine calculates the moisture content and the
c      capacitance.  units - kg,m,s
c
c      VARIABLES PASSED TO ROUTINE:
c
c      nelem    element number
c      mat      material number of this element
c      nn       number of nodes at which output is to be computed
c      p        effective pressure at each node:  rhog(phi + x)

```

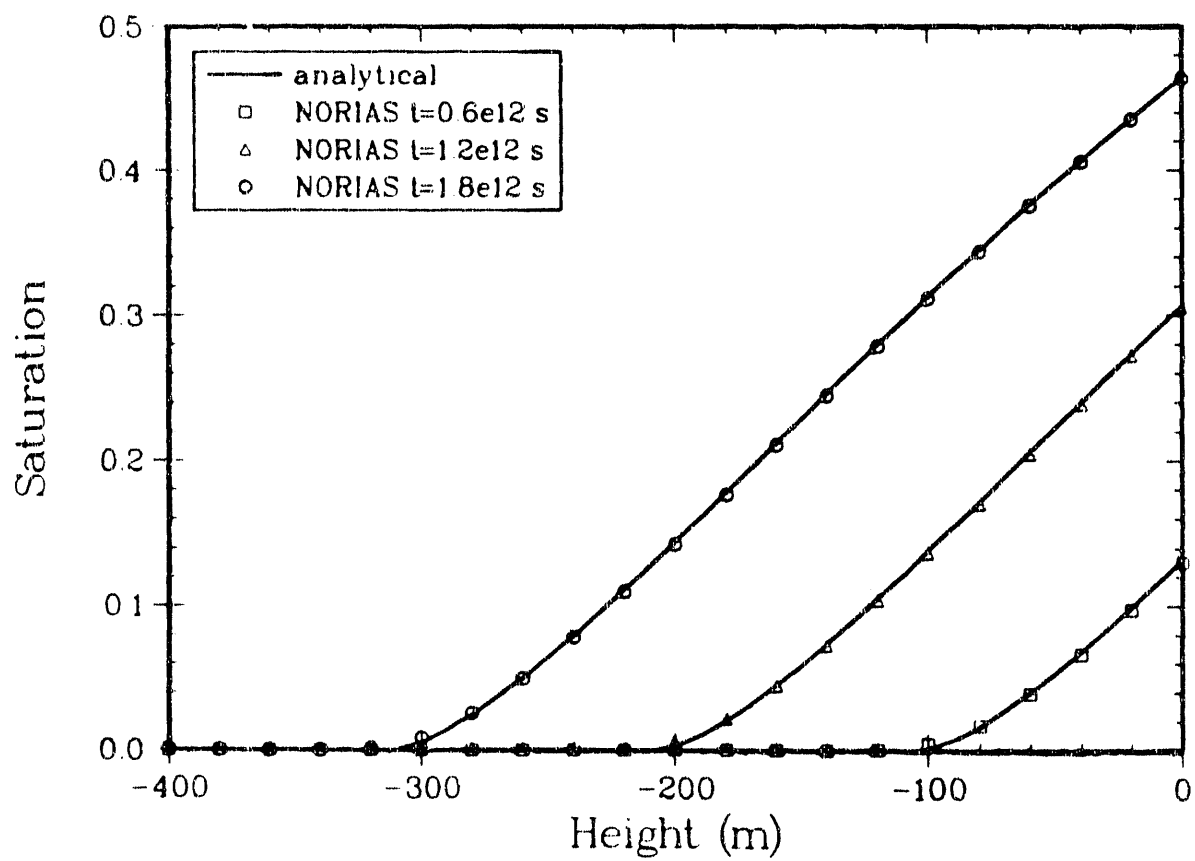


Figure 7.1. Comparison of NORIA-SP Results and Analytical Solution for Fleming's Problem

```

c      x      x-coordinate value at each node
c      z      z-coordinate value at each node
c      time    solution time
c      rhog    product of water density and gravity
c
c  VARIABLES RETURNED FROM ROUTINE:
c
c      cap      capacitance at each node
c      thm      matrix moisture content at each node
c      phim1    matrix porosity for this element/material
c      smr1     matrix residual saturation for this element/material
c
c  Fracture and total variables need only be defined in the case of
c  a composite model.
c
c      thf      fracture moisture content at each node
c      phif1    fracture 'porosity' for this element/material (must be set
c              to zero if no composite model)
c      sfr      fracture residual saturation for this element/material
c      th       total moisture content at each node
c      phi      total 'porosity' for this element/material (zero if no
c              composite model) ( $\phi = \phi_f + \phi_m * (1 - \phi_f)$ )
c
c  other variables used in this subroutine:
c
c      betp     water compressibility (beta prime)
c      alpp     rock compressibility (alpha prime)
c      fc       fracture compressibility
c      sm       saturation of matrix
c      sf       saturation of fracture
c
c
c  for matrix material only ( no fractures) only define phim, smr, thm, and cap
c  for composite model also define phif, sfr, phi, thf and th
c
c      *** material properties used in this subroutine:
c
c      mat 2, tcb
c      mat 3, tpt-tn
c      mat 4, tpt-tnv
c      mat 5, tpt-tv
c      mat 6, tpt-tm

```

```

c      mat 7, tpt-tml
c      mat 8, tpt-tdl
c      mat 9, tpt-td
c      mat 10, tpt-tm
c
c
c
c
c      rock properties for g4 and ue-25a #1
c
c      data(smr(i), i= 2,6) /0.125, 0.005, 0.150, 0.005, 0.100/
c      data(smr(i), i= 7,10)/0.050, 0.100, 0.150, 0.100/
c      data(phim(i), i= 2,6) /0.295, 0.36, 0.20, 0.04, 0.10/
c      data(phim(i), i= 7,10)/0.12, 0.08, 0.06, 0.10/
c      data(phif(i), i= 2,6) /1.8e-5, 9.0e-5, 6.6e-5, 2.0e-4, 3.0e-5/
c      data(phif(i), i= 7,10)/3.0e-5, 1.8e-5, 3.0e-5, 3.0e-5/
c      data(alpm(i), i= 2,6) /0.003, 0.02, 0.03, 0.002, 0.005/
c      data(alpm(i), i= 7,10)/0.01, 0.003, 0.004, 0.005/
c      data(betm(i), i= 2,6) /1.55, 1.2, 2.2, 1.7, 1.9/
c      data(betm(i), i= 7,10)/1.7, 1.8, 2.0, 1.9/
c      data(alpp(i), i= 2,6) /26.e-7, 5.8e-7, 12.e-7, 82.e-7, 6.2e-7/
c      data(alpp(i), i= 7,10)/26.e-7, 5.8e-7, 12.e-7, 82.e-7/
c      data(fc(i), i= 2,6) /2.8e-8, 12.e-8, 5.6e-8, 19.e-8, 132.e-8/
c      data(fc(i), i= 7,10)/2.8e-8, 12.e-8, 5.6e-8, 19.e-8/
c
c      saturation function statement (van Genuchten/Mualem)
c
c      satfu(sr,al,ap)=(1-sr)*(1+ap)**(-al) + sr
c
c      derivative of saturation function statement
c
c      chfu(sr,al,ap,bet,a,phm)=(1-sr)*al*(1.+ap)**(-al-1.)
c      1 *bet*a**bet*(-phm)**(bet-1)/rhog
c
c      total porosity
c
c      phi = phif(mat) + phim(mat) * (1 - phif(mat))
c
c      fracture properties
c
c      sfr=0.04
c      alpf=1.28
c      betf=4.23
c

```

```

c  matrix properties
c
c  water compressibility
    betp=9.8e-7
c
c  evaluate moisture content
c
    alm=(1.-1/betm(mat))
    alf=(1.-1./betf)
c
c  set values for use in computing fluxes and velocities
c
    phm1=phim(mat)
    phif1=phif(mat)
    smr1 =smr(mat)
c
c  loop over nodes
c
    do 100 j=1,nn
    phm=p(j)/rhog - z(j)
    if (phm.gt.-.000001) phm=-.000001
    apm=(alpm(mat)*(-phm)**betm(mat)
    apf=(alpf*      (-phm)**betf
    sm=satfu(smr(mat),alm,apm)
    sf=satfu(sfr      ,alf,apf)
c
c  convert saturation to moisture content
c
    thm(j)=sm*phim(mat)*(1.-phif(mat))
    thf(j)=sf*phif(mat)
    th(j)=thm(j) + thf(j)
c
c  incorporate these compressibility effects into the capacitance term
c
c  Note: the compressibilities are shown here for completeness but
c       are not used in this sample problem.
c
    watcomp=betp*th(j)/rhog
    rokcomp=alpp(mat)*th(j)/((phif(mat)+phim(mat))*(sm-phif(mat)
2      *(sm-sf))/rhog
    fracomp=fc(mat)*th(j)/((phif(mat)+phim(mat))*(sm-sf)/rhog
c
c  compute the derivative of saturation with respect to psi

```



```

c
      chm=chfu(smr(mat),alm,apm,betm(mat),alpm(mat),phm)
      chf=chfu(sfr      ,alf,apf,betf      ,alpf      ,phm)
c
c  convert to moisture content and sum terms for total capacitance
c
      cm1=chm*phim(mat)*(1.-phif(mat)) + chf*phif(mat)
c
c  for steady-flow analyses, use cm(j) = cm1 + constant
c  for transient analyses, use cm(j) = cm1 + watcomp + rokcomp + fracomp
c
      cap(j)=cm1 + watcomp + rokcomp + fracomp
c
      cap(j)=cm1 + 1.0e-5
100  continue
c
      return
      end
c
c
      subroutine perm(nelem,mat,nn,p,x,z,time,rhog,
1          akm11,akm22,akf11,akf22,ak11,ak22)
c
c  this subroutine calculates the permeability (m**2) of the
c  rock matrix and fractures in units of kg,m,s
c
      dimension p(8),x(8),z(8),akm11(8),akm22(8),akf11(8),akf22(8),
1          ak11(10),ak22(8)
c
      dimension phif(13),alpm(13),betm(13)
      dimension conm(13),conf(13),cratm(13),cratf(13)
c
c  Variables passed to Routine:
c
c  nelem  Element number
c  mat    Material number of this element
c  nn     Number of nodes at which output is to be computed
c  p      Effective pressure at each node
c  x      x-coordinate value at each node
c  z      z-coordinate value at each node
c  time   solution time
c  rhog   product of water density and gravity

```

```

c
c  VARIABLES RETURNED FROM ROUTINE:
c
c    akm11  matrix permeability in the first principal-axis direction
c           at each node
c
c    akm22  matrix permeability in the second principal-axis direction
c           at each node
c
c  Fracture and total variables need only be defined in the
c  case of a composite model
c
c    akf11  fracture permeability in the first principal-axis
c           direction at each node
c
c    akf22  fracture permeability in the second principal-axis
c           direction at each node
c
c    ak11   total permeability in the first principal-axis
c           direction at each node
c
c    ak22   total permeability in the second principal-axis
c           direction at each node
c
c
c  other variables used in this subroutine:
c
c    phm      matrix pressure head
c    cratm    conductivity ratio in rock matrix
c    cratf    conductivity ratio in fractures
c              ratio is of conductivity in direction 22 (second
c              principal axis) divided by conductivity in
c              direction 11 (first principal axis)
c    conm     saturated conductivity of matrix
c    conf     saturated conductivity of fractures
c
c
c
c    ***  material properties used in this subroutine:
c
c    mat  4, tcb
c    mat  5, tpt-tn
c    mat  6, tpt-tnv

```

```

c      mat 7, tpt-tv
c      mat 8, tpt-tm
c      mat 9, tpt-tml
c      mat 10, tpt-tdl
c      mat 11, tpt-td
c      mat 12, tpt-tm
c
c
c
c
c
c
c      rock properties for g4 and ue-25a #1
c
c      data(alpm(i), i= 2, 6)/ 0.003, 0.02, 0.03, 0.002, 0.005/
c      data(alpm(i), i= 7,10)/ 0.01, 0.003, 0.004, 0.005 /
c      data(betm(i), i= 2, 6)/ 1.55, 1.2, 2.2, 1.7, 1.9/
c      data(betm(i), i= 7,10)/ 1.7, 1.8, 2.0, 1.9 /
c      data(phif(i), i= 2, 6)/ 1.8e-5, 9.0e-5, 6.6e-5, 2.0e-4, 3.0e-5/
c      data(phif(i), i= 7,10)/ 3.0e-5, 1.8e-5, 3.0e-5, 3.0e-5 /
c      data(conm(i), i= 2, 6)/ 1.5e-11,3.0e-12, 2.4e-6, 3.0e-12, 2.e-11/
c      data(conm(i), i= 7,10)/ 2.0e-11,2.0e-12, 5.e-12, 2.0e-11/
c      data(conf(i), i= 2, 6)/ 3.0e-5, 8.0e-4, 4.0e-4, 4.0e-4, 4.0e-5/
c      data(conf(i), i= 7,10)/ 4.0e-5, 4.0e-5, 4.0e-5, 4.0e-5 /
c      data(cratm(i),i= 2, 6)/ 1.0, 1.0, 1.0, 1.0, 1.0/
c      data(cratm(i),i= 7,10)/ 1.0, 1.0, 1.0, 1.0 /
c      data(cratf(i),i= 2, 6)/ 1.0, 1.0, 1.0, 1.0, 1.0/
c      data(cratf(i),i= 7,10)/ 1.0, 1.0, 1.0, 1.0 /
c
c
c
c      conductivity function statement
c
c      confu(al,ap)=(1.+ap)**(-al/2.)*(1.-(ap/(1.+ap))**al)**2
c
c
c      fracture properties
c
c      alpf=1.28
c      betf=4.23
c
c      do 20 j=1,nn
c      phm=p(j)/rhog - z(j)
c

```

```

c  if pressure head becomes positive, permeability is calculated for
c  a very small negative pressure head (i.e., approximately equal
c  to the saturated value).
c
c      if(phm.gt. -.000001) phm=-.000001
c
c  calculate conductivity in the direction of the first
c  principal axis using Van Genuchten/Mualem formulations
c
c      apm=(alpm(mat)*(-phm))**betm(mat)
c      alm=(1-1./betm(mat))
c      apf=(alpf*      (-phm))**betf
c      alf=(1-1/betf)
c      com=confu(alm,apm)*conm(mat)
c      cof=confu(alf,apf)*conf(mat)
c
c  convert conductivity to bulk permeability
c  assume viscosity = 0.001
c
c      akm11(j)=(1.-phif(mat))*com*0.001/rhog
c      akf11(j)=phif(mat)*cof*0.001/rhog
c
c  calculate permeability for second principal axis
c  allowing for anisotropy
c
c      akm22(j)=akm11(j)*cratm(mat)
c      akf22(j)=akf11(j)*cratf(mat)
c
c  calculate permeability of equivalent combined porous media
c
c      ak11(j)=akm11(j)+akf11(j)
c      ak22(j)=akm22(j)+akf22(j)
20  continue
    return
    end

```

Input Deck for Sample Problem 2

The input deck for the two-dimensional PACE problem is shown below. SI units are used throughout. The nine solid material property cards define a constant pressure initial condition throughout the domain. The bottom boundary is held at a pressure near

saturation ($\psi = 0$) to simulate the water table. The top boundary has a constant flux of -3.1745×10^{-13} m/s (0.01 mm/yr). Note that velocities are to be computed at the nodal points. The calculation is to start at time zero and continue for 20 iterations or until a time of 1.0×10^5 s, whichever condition is met first. The initial time step is 1.0×10^{-4} . The default output is to be printed at a calculated time of 1.0×10^5 s. An EXODUS file is to be produced with saturation values only at a time of 1,000 s.

```
$ two-dimensional 9 layer region; q=0.01mm/yr
setup, 1, 61                                $ imax=61
water,      1, 1000.0, 0.001, 9.8
tcb ,       2, 7151050., 0.0
tpt-tn,     3, 7151050., 0.0
tpt-tnv,    4, 7151050., 0.0
tpt-tv,     5, 7151050., 0.0
tpt-tm,     6, 7151050., 0.0
tpt-tml,    7, 7151050., 0.0
tpt-tdl,    8, 7151050., 0.0
tpt-td,     9, 7151050., 0.0
tpt-tm,    10, 7151050., 0.0
end
  1,  1, 61, 19, 1.06, 1.10, 1.06, 1.10      $ ***** region 1 *****
0.0, 923.1, 923.1, 0.0                      $ tcb
729.7, 729.7, 784.2, 841.2
  1, 19, 61, 25, 1.06, 1.00, 1.06, 1.00      $ ***** region 2 *****
0.0, 923.1, 923.1, 0.0                      $ tpt-tn
841.2, 784.2, 787.2, 850.9
  1, 25, 61, 31, 1.06, 1.00, 1.06, 1.10      $ ***** region 3 *****
0.0, 923.1, 923.1, 0.0                      $ tpt-tnv
850.9, 787.2, 797.3, 860.1
  1, 31, 61, 37, 1.06, 1.00, 1.06, 1.00      $ ***** region 4 *****
0.0, 923.1, 923.1, 0.0                      $ tpt-tv
860.1, 797.3, 810.7, 868.6
  1, 37, 61, 47, 1.06, 0.98, 1.06, 0.98      $ ***** region 5 *****
0.0, 923.1, 923.1, 0.0                      $ tpt-tm
868.6, 810.7, 871.1, 930.2
  1, 47, 61, 61, 1.06, 1.10, 1.06, 1.10      $ ***** region 6 *****
0.0, 923.1, 923.1, 0.0                      $ tpt-tml
930.2, 871.1, 1006.1, 1082.9
  1, 61, 61, 69, 1.06, 0.98, 1.06, 0.98      $ ***** region 7 *****
0.0, 923.1, 923.1, 0.0                      $ tpt-tdl
1082.9, 1006.1, 1073.7, 1148.2
  1, 69, 61, 75, 1.06, 0.98, 1.06, 0.98      $ ***** region 8 *****
0.0, 923.1, 923.1, 0.0                      $ tpt-td
```

```

1148.2, 1073.7, 1093.6, 1183.2
  1, 75, 61, 85, 1.06, 1.10, 1.06, 1.10
0.0, 923.1, 923.1, 0.0
1183.2, 1093.6, 1116.4, 1200.6
end
jloop, 9, 2
iloop, 30, 2
quad8/4, 2, 1, 1
iend
jend
jloop, 3, 2
iloop, 30, 2
quad8/4, 3, 1, 19
iend
jend
jloop, 3, 2
iloop, 30, 2
quad8/4, 4, 1, 25
iend
jend
jloop, 3, 2
iloop, 30, 2
quad8/4, 5, 1, 31
iend
jend
jloop, 5, 2
iloop, 30, 2
quad8/4, 6, 1, 37
iend
jend
jloop, 7, 2
iloop, 30, 2
quad8/4, 7, 1, 47
iend
jend
jloop, 4, 2
iloop, 30, 2
quad8/4, 8, 1, 61
iend
jend
jloop, 3, 2
iloop, 30, 2
quad8/4, 9, 1, 69

```

```

$ ***** region 9 *****
$ tpt-tm

```

```

$ ***** region 1 *****
$ tcb

```

```

$ ***** region 2 *****
$ tpt-tn

```

```

$ ***** region 3 *****
$ tcb-tnv

```

```

$ ***** region 4 *****
$ tcb-tv

```

```

$ ***** region 5 *****
$ tcb-tm

```

```

$ ***** region 6 *****
$ tcb-tml

```

```

$ ***** region 7 *****
$ tcb-tdl

```

```

$ ***** region 8 *****
$ tcb-td

```

```

iend
jend
jloop, 5, 2          $ ***** region 9 *****
iloop, 30, 2         $ tcb-tm
quad8/4, 10, 1, 75
iend
jend
iloop, 30, 2          $ ***** top surface bc *****
bc, ulside, 1, 83, 3, -3.1745e-13 $ infiltration rate in m/s
iend
iloop, 30, 2          $ ***** bottom surface bc *****
bc, peffside, 1, 1, 1, +7151050. $ pressure head approx. zero at water table
iend
end
velocity,nodal
formkfkf
output, single, 1,2,3,4,5,1255,1256,1257,1258,1259,1260
unzipp,0.0 , 1.0e15, 0.1e-3,20, , 1, 1e15
end
stop, post
select,1,saturation
time,specified,1,1e3
end
$
$ pace exercise g4 to ue-25a#1
$ 30 horizontal elements and 42 vertical elements.
$ Isotropic conductivity in all units.
$ First principal conductivity axis 0-degrees clockwise from horizontal.
$ Water table at bottom boundary.
$ Infiltration rate = 0.01 mm/yr at top.
$ Run with NOR_sub.pace
$ In UNZIPPP, eps = 0.00001 for convergence limits.
$ In FLUIDC, compressibilities set for steady-state results only.
$ Input lines for additional solid materials:

```

Output File for Sample Problem 2

The total output file for this problem is quite lengthy (500 lines). A partial listing is included. The portion of the file which echoes the input data has not been included.

two-dimensional 9 layer region; $q=0.01\text{mm/yr}$

material properties

—fluid—

no.	name	density	viscosity	gravity
1	water	1000.0	$0.10000\text{e}-02$	9.8000

—solid(s)—

no.	name	initial effective pressure	alpha (degrees)
2	tcb	$0.71511\text{e}+07$	0.00000
3	tpt-tn	$0.71511\text{e}+07$	0.00000
4	tpt-tnv	$0.71511\text{e}+07$	0.00000
5	tpt-tv	$0.71511\text{e}+07$	0.00000
6	tpt-tm	$0.71511\text{e}+07$	0.00000
7	tpt-tml	$0.71511\text{e}+07$	0.00000
8	tpt-tdl	$0.71511\text{e}+07$	0.00000
9	tpt-td	$0.71511\text{e}+07$	0.00000
10	tpt-tm	$0.71511\text{e}+07$	0.00000

—data for 1260 elements have been read—

time in setup = 2.471 seconds

velocities will be computed at nodal points.

the two-dimensional planar version of the element formulation was requested

time in formkf = 51.071 seconds

time in pre-front= 11.70941

properties of matrix system—

number of degrees of freedom— 3925
max front width— 66
storage for max front— 4560
buffer space— 190439

Output for Sample Problem 2 (cont. 1)

pressure fields were initialized by material with the following values—

```
material no. 2 initial effective pres= 0.7151050e+07
material no. 3 initial effective pres= 0.7151050e+07
material no. 4 initial effective pres= 0.7151050e+07
material no. 5 initial effective pres= 0.7151050e+07
material no. 6 initial effective pres= 0.7151050e+07
material no. 7 initial effective pres= 0.7151050e+07
material no. 8 initial effective pres= 0.7151050e+07
material no. 9 initial effective pres= 0.7151050e+07
material no. 10 initial effective pres= 0.7151050e+07
```

output for the transient problem was obtained over the interval—

initial time= 0.000000e+00

final time= 0.100000e+16

initial time step= 0.100000e-03

integration was terminated when current time.ge.final time or when 20 time steps were taken

=====

initial conditions are the solution at time step no. = 1

```
current time= 0.100000e-03    time step= 0.100000e-03    time step no.= 2
rms = 0.00000
max change in effec liq pres between time steps is 0.1172093e-10 in element 1231 node 4

current time= 0.200000e-03    time step= 0.100000e-03    time step no.= 3
rms = 0.00000
max change in effec liq pres between time steps is 0.1172093e-10 in element 1231 node 4

current time= 0.300000e-03    time step= 0.100000e-03    time step no.= 4
rms = 0.00000
newit= 1    rms= 0.00000
max change in effec liq pres between time steps is 0.1172093e-10 in element 1231 node 4

current time= 0.1032280e+09    time step= 0.1032280e+09    time step no.= 5
rms = 0.00001
newit= 1    rms= 0.00001
max change in effec liq pres between time steps is 0.1209291e+02 in element 1231 node 4

current time= 0.3042646e+09    time step= 0.2010365e+09    time step no.= 6
rms = 0.00001
newit= 1    rms= 0.00001
max change in effec liq pres between time steps is 0.2295688e+02 in element 1231 node 4

current time= 0.6753037e+09    time step= 0.3710391e+09    time step no.= 7
rms = 0.00000
newit= 1    rms= 0.00000
max change in effec liq pres between time steps is 0.4092201e-01 in element 1231 node 4

current time= 0.5867809e+10    time step= 0.5192505e+10    time step no.= 8
rms = 0.00001
newit= 1    rms= 0.00001
max change in effec liq pres between time steps is 0.7462501e+01 in element 1231 node 4
```

Output for Sample Problem 2 (cont. 2)

```

current time= 0.1707719e+11    time step= 0.1120938e+11    time step no.= 9
rms = 0.00002
newit= 1    rms= 0.00002
max change in effec liq pres between time steps is 0.3082391e+02 in element 1231 node 4

current time= 0.3424880e+11    time step= 0.1717161e+11    time step no.= 10
rms = 0.00004
newit= 1    rms= 0.00004
max change in effec liq pres between time steps is 0.6174092e+02 in element 1231 node 4

current time= 0.5637185e+11    time step= 0.2212305e+11    time step no.= 11
rms = 0.00006
newit= 1    rms= 0.00006
max change in effec liq pres between time steps is 0.8564204e+02 in element 1231 node 4

current time= 0.8299852e+11    time step= 0.2662667e+11    time step no.= 12
rms = 0.00006
newit= 1    rms= 0.00006
max change in effec liq pres between time steps is 0.1015963e+03 in element 1231 node 4

current time= 0.1140899e+12    time step= 0.3109138e+11    time step no.= 13
rms = 0.00007
newit= 1    rms= 0.00007
max change in effec liq pres between time steps is 0.1111378e+03 in element 1231 node 4

current time= 0.1500795e+12    time step= 0.3598961e+11    time step no.= 14
rms = 0.00007
newit= 1    rms= 0.00007
max change in effec liq pres between time steps is 0.1167152e+03 in element 1231 node 4

current time= 0.1917833e+12    time step= 0.4170379e+11    time step no.= 15
rms = 0.00006
newit= 1    rms= 0.00006
max change in effec liq pres between time steps is 0.1196681e+03 in element 1231 node 4

current time= 0.2403725e+12    time step= 0.4858920e+11    time step no.= 16
rms = 0.00006
newit= 1    rms= 0.00006
max change in effec liq pres between time steps is 0.1206088e+03 in element 1231 node 4

current time= 0.2972936e+12    time step= 0.5692109e+11    time step no.= 17
rms = 0.00006
newit= 1    rms= 0.00006
max change in effec liq pres between time steps is 0.1196011e+03 in element 1231 node 4

current time= 0.3641080e+12    time step= 0.6681446e+11    time step no.= 18
rms = 0.00006
newit= 1    rms= 0.00006
max change in effec liq pres between time steps is 0.1166005e+03 in element 1231 node 4

current time= 0.4423010e+12    time step= 0.7819297e+11    time step no.= 19
rms = 0.00006
newit= 1    rms= 0.00006
max change in effec liq pres between time steps is 0.1119923e+03 in element 1231 node 4

```

Output for Sample Problem 2 (cont. 3)

current time= 0.6385893e+12 time step= 0.1053172e+12 time step no.= 21
 rmasp= 0.00007 rmasew= 0.00007
 newit= 1 rms= 0.00007
 max change in effec liq pres between time steps is 0.1025567e+03 in element 1231 node

extended core required for this problem—

vector for condensed eqns— 273900
 total ecs— 320521

number of data blocks used in solution— 3

time in unzipp = 694.615 seconds

—nodes—

	-1-	-2-	-3-	-4-	-5-	-6-
element 1						
PRESS	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07
TSAT	0.1000000e+01	0.1000000e+01	0.9953013e+00	0.9948549e+00	0.1000000e+01	0.9983461e+00
TCOND11	0.5549719e-09	0.5549719e-09	0.8993448e-11	0.8820222e-11		
TCOND22	0.5549719e-09	0.5549719e-09	0.8993448e-11	0.8820222e-11		
element 2						
PRESS	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07
TSAT	0.1000000e+01	0.1000000e+01	0.9956868e+00	0.9953013e+00	0.1000000e+01	0.9984794e+00
TCOND11	0.5549719e-09	0.5549719e-09	0.9153392e-11	0.8993448e-11		
TCOND22	0.5549719e-09	0.5549719e-09	0.9153392e-11	0.8993448e-11		
element 3						
PRESS	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07
TSAT	0.1000000e+01	0.1000000e+01	0.9960203e+00	0.9956868e+00	0.1000000e+01	0.9985946e+00
TCOND11	0.5549719e-09	0.5549719e-09	0.9300692e-11	0.9153392e-11		
TCOND22	0.5549719e-09	0.5549719e-09	0.9300692e-11	0.9153392e-11		
element 4						
PRESS	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07
TSAT	0.1000000e+01	0.1000000e+01	0.9963091e+00	0.9960203e+00	0.1000000e+01	0.9986944e+00
TCOND11	0.5549719e-09	0.5549719e-09	0.9436009e-11	0.9300692e-11		
TCOND22	0.5549719e-09	0.5549719e-09	0.9436009e-11	0.9300692e-11		
element 5						
PRESS	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07	0.7151050e+07
TSAT	0.1000000e+01	0.1000000e+01	0.9965598e+00	0.9963091e+00	0.1000000e+01	0.9987808e+00
TCOND11	0.5549719e-09	0.5549719e-09	0.9560026e-11	0.9436009e-11		
TCOND22	0.5549719e-09	0.5549719e-09	0.9560026e-11	0.9436009e-11		
element 1255						
PRESS	0.7151606e+07	0.7151606e+07	0.7172180e+07	0.7172235e+07	0.7151606e+07	0.7155552e+07
TSAT	0.5415467e+00	0.5420171e+00	0.5413071e+00	0.5408466e+00	0.5417818e+00	0.5411447e+00
TCOND11	0.1770243e-12	0.1779733e-12	0.1765427e-12	0.1756198e-12		
TCOND22	0.1770243e-12	0.1779733e-12	0.1765427e-12	0.1756198e-12		
element 1256						
PRESS	0.7151606e+07	0.7151607e+07	0.7172131e+07	0.7172180e+07	0.7151606e+07	0.7155549e+07
TSAT	0.5420171e+00	0.5424365e+00	0.5417178e+00	0.5413071e+00	0.5422267e+00	0.5415606e+00
TCOND11	0.1779733e-12	0.1788230e-12	0.1773690e-12	0.1765427e-12		
TCOND22	0.1779733e-12	0.1788230e-12	0.1773690e-12	0.1765427e-12		
element 1257						
PRESS	0.7151607e+07	0.7151607e+07	0.7172087e+07	0.7172131e+07	0.7151607e+07	0.7155547e+07
TSAT	0.5424365e+00	0.5428105e+00	0.5420838e+00	0.5417178e+00	0.5426234e+00	0.5419314e+00
TCOND11	0.1788230e-12	0.1795833e-12	0.1781083e-12	0.1773690e-12		
TCOND22	0.1788230e-12	0.1795833e-12	0.1781083e-12	0.1773690e-12		
element 1258						
PRESS	0.7151607e+07	0.7151608e+07	0.7172048e+07	0.7172087e+07	0.7151607e+07	0.7155544e+07
TSAT	0.5428105e+00	0.5431438e+00	0.5424101e+00	0.5420838e+00	0.5429770e+00	0.5422618e+00
TCOND11	0.1795833e-12	0.1802633e-12	0.1787694e-12	0.1781083e-12		
TCOND22	0.1795833e-12	0.1802633e-12	0.1787694e-12	0.1781083e-12		
element 1259						
PRESS	0.7151608e+07	0.7151611e+07	0.7171976e+07	0.7172048e+07	0.7151607e+07	0.7155527e+07
TSAT	0.5431438e+00	0.5434409e+00	0.5426978e+00	0.5424101e+00	0.5432921e+00	0.5425550e+00
TCOND11	0.1802633e-12	0.1808713e-12	0.1793539e-12	0.1787694e-12		
TCOND22	0.1802633e-12	0.1808713e-12	0.1793539e-12	0.1787694e-12		
element 1260						
PRESS	0.7151611e+07	0.7151579e+07	0.7170955e+07	0.7171976e+07	0.7151590e+07	0.7154368e+07
TSAT	0.5434409e+00	0.5437029e+00	0.5428755e+00	0.5426978e+00	0.5435715e+00	0.5427717e+00
TCOND11	0.1808713e-12	0.1814089e-12	0.1797159e-12	0.1793539e-12		
TCOND22	0.1808713e-12	0.1814089e-12	0.1797159e-12	0.1793539e-12		

8 Glossary

Basis function—one of a set of functions used to expand an arbitrary function. In the finite element method, the basis functions are defined on elements and dependent variables are expanded in them [see Equation (3.5) in Subsection 3.3].

Bilinear—refers to functions that are linear in each of two variables.

Effective pressure—an expression for pressure defined following Equation (2.2) in Subsection 2.3.

Element coefficient matrix—a matrix that contains the contributions to the global coefficient matrix, that is, the Jacobian matrix used in the Newton iteration procedure, from an individual element.

Element connectivity—a term used to describe the way in which the finite elements are interconnected. Two adjacent elements may be connected, *i.e.*, may allow mass transfer through the common side, or may not be connected depending on the way nodal points are assigned to the elements.

Gauss point—a quadrature point in a scheme devised by Gauss. (See “Quadrature”.)

Global coefficient matrix—a term used here for the Jacobian matrix used in the Newton iteration procedure.

Natural coordinate system—a term describing the coordinate system in the mapped domain where a standard element lies (see “Standard element”). Here, the natural coordinate system is a rectangular Cartesian coordinate system and the mapped domain is defined by a subparametric or an isoparametric mapping of an element in the original problem domain.

Newton iteration procedure—a well-known iteration procedure for solving a set of nonlinear equations. The basis for the iteration procedure is a Taylor expansion that is truncated after the linear term. Applying Newton’s iteration procedure to a set of nonlinear equations involves solving a system of linear equations at each iteration. A Newton step is the same as a Newton iteration.

Predictor-corrector—a general category of numerical time integration schemes that involves two steps. The first step is to predict the solution at the next timeplane. The prediction is based on a knowledge of how the variables have behaved over past time steps. The corrector step uses the predicted values at the new timeplane to improve upon the solution. The corrector step improves both the accuracy and stability of the overall time integration procedure.

Principal axes—the axes used to define an orthotropic property. The directions of the two principal axes are chosen so that the tensor used to define the orthotropic property is diagonal.

Quadrature—a general category of methods that serve to estimate the values of integrals.

Residual—an equation that measures the degree to which one of the governing equations is satisfied in some integral sense. Normally, residual equations are set equal to zero; however, in practice they can only be set equal to approximately zero if they contain nonlinear terms.

Residual moisture content—the moisture content below which moisture is not free to flow by pressure or density gradient, or by gravity.

Standard element—an element in the isoparametrically or subparametrically mapped domain. Standard elements have simple shapes (quadrilaterals in NORIA-SP).

Weak form—a term that originated from the topic of variational calculus. There are several weak forms of a differential equation. One is the integral of the product of the differential equation with an arbitrary function, which is referred to as a weighting function. Other weak forms result from the application of Green's theorem to this integral equation.

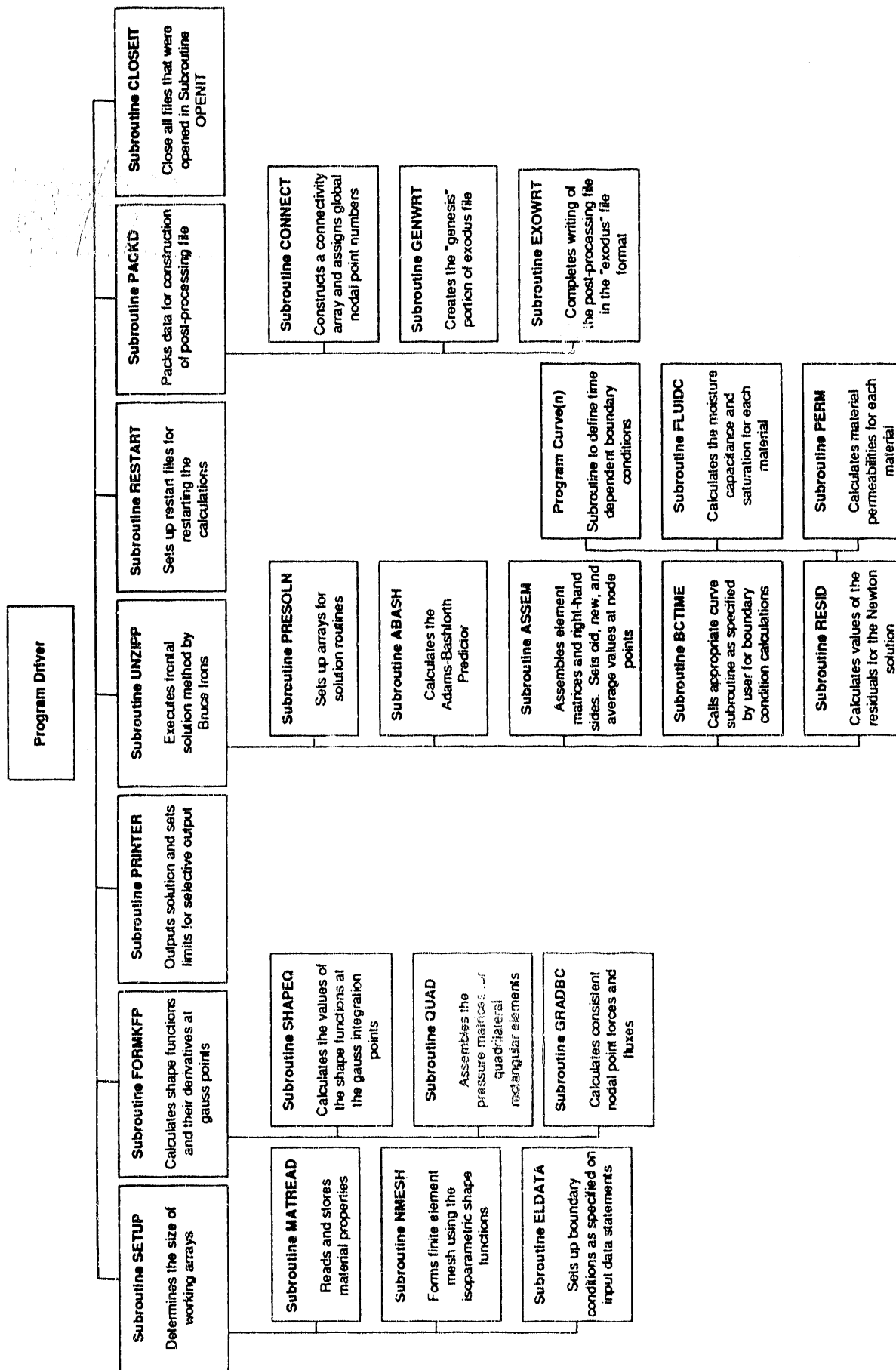
Weighting function—an arbitrary function that appears in the weak form of a differential equation. (See “Weak form”.)

9 References

- Barnard, R. W., and Dockery, H. A., Eds, Technical Summary of the Performance Assessment Computational Exercises for 1991 (PACE-90), Volume 1: 'Nominal Configuration' Hydrogeologic Parameters and Computational Results, *Sandia National Laboratories*, SAND90-2726, 1991. (NNA.910523.0001)
- Bixler, N. E. NORIA—A Finite Element Computer Program for Analyzing Water, Vapor, Air, and Energy Transport in Porous Media, *Sandia National Laboratories*, SAND84-2057, 1985. (NNA.870721.0002)
- Bixler, N. E. "An Improved Time Integrator for Finite Element Analysis," *Comm. in Appl. Num. Meth.*, 5 pp. 69–78, 1989. (NNA.910813.0008)
- Darcy, H. *Les Fontaines Publiques de la Ville Dijon*, Victor Dalmont, Paris, France, 1856. (NNA.900720.0059)
- Dykhuizen, R. C., Eaton, R. R., Hopkins, P. L., and Martinez, M. J. PACE-90 Water and Solute Transport Calculations for 0.01, 0.1, and 0.5 mm/yr Infiltration into Yucca Mountain, *Sandia National Laboratories*, SAND90-3165, 1991. (NNA.911202.0032)
- Eaton, R. R., Gartling, D. K., and Larson, D. E. SAGUARO—A Finite Element Computer Program for Partially Saturated Porous Flow Problems, *Sandia National Laboratories*, SAND82-2772, 1983. (NN1.881007.0040)
- Fleming, J. F., Parlange, J. Y., and Hograth, W. L. "Scaling of Flux and Water Content Relations: Comparison of Optimal and Exact Result," *Soil Science*, 137, No. 6 pp. 464–468, 1986. (NNA.900308.0329)
- Freeze, R. A. and Cherry, J. A. *Groundwater*, Prentice-Hall Inc., Englewood Cliffs, N.J. 1979. (NNA.870406.0444)
- Gartling, D. K. NACHOS—A Finite Element Computer Program for Incompressible Flow Problems: Part I—Theoretical Background, *Sandia National Laboratories*, SAND77-1333, 1978. (NNA.910923.0001)
- Gartling, D. K. NACHOS—A Finite Element Computer Program for Incompressible Flow Problems: Part II—User's Manual, *Sandia National Laboratories*, SAND77-1334, 1978. (NNA.910923.0002)
- Gartling, D. K. and Hickox, C. E. MARIAH—A Finite Element Computer Program for Incompressible Porous Flow Problems: Theoretical Background, *Sandia National Laboratories*, SAND79-1622, 1982. (NNA.HQS.880517.2681)
- Gartling, D. K. and Hickox, C. E. MARIAH—A Finite Element Computer Program for Incompressible Porous Flow Problems: User's Manual, *Sandia National Laboratories*, SAND79-1623 (revised), 1982. (NNA.910809.0016)

- Gartling, D. K.** COYOTE—A Finite Element Program for Nonlinear Heat Conduction Problems, *Sandia National Laboratories*, SAND77-1332 (revised), 1982. (HQS.880517.2294)
- Glick, J. H. and Gartling, D. K.** TRINITY II—A Postprocessing Program for Two-Dimensional Finite Element Data, *Sandia National Laboratories*, SAND88-0896, 1988. (NNA.890713.0212)
- Gilkey, A. P. and Glick J. H.** BLOT—A Mesh and Curve Plot Program for the Output of a Finite Element Analysis, *Sandia National Laboratories*, SAND88-1432, 1989. (NNA.910816.0057)
- Gresho, P. M., Lee, R. L., and Sani, R. L.** On the Time Dependent Solution of the Incompressible Navier-Stokes Equations in Two and Three Dimensions, *Recent Advances in Numerical Methods in Fluids, Volume I*, Pineridge Press, Swansea, U. K., pp 27-79, 1980. (NNA.910813.003)
- Irons, B. M.** A Frontal Solution Program for Finite Element Analysis, *Int. J. Num. Meth. Engng.*, **2** pp. 5-32, 1970. (NNA.910819.0001)
- Luskin, M. and Rannacher, R.** On the Smoothing Property of the Crank-Nicolson Scheme, *Applicable Analysis*, **14** pp. 117-135, 1982. (NNA.910819.0002)
- Mills-Curran, W. C., Gilkey, A. P., and Flanagan, D. F.** EXODUS: A Finite Element File Format for Pre- and Postprocessing, *Sandia National Laboratories*, SAND87-2997, 1988. (NNA.910813.0001)
- Peters, R. R. and Klavetter, E. A.** A Continuum Model for Water Movement in an Unsaturated Fractured Rock Mass, *Water Resources Res.*, **24** pp. 416-430, 1988. (NNA.890523.0139)
- Richards, L. A.** Capillary Conduction of Liquids Through Porous Mediums, *Physics*, **1** pp. 318-333, 1931. (NNA.890522.0282)
- Shampine, L. and Gordon, M.** *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. Freeman and Company, San Francisco, CA, pp 75-94, 1975. (NNA.911007.0003)
- Strang, G. and Fix, G. J.** *An Analysis of the Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1973. (NNA.910923.0004)
- Zienkiewicz, O. C.** *The Finite Element Method*, McGraw-Hill, Berkshire, England, 1977. (NNA.910327.0066)

10 Appendix A



11 Appendix B

Information from the Reference Information Base Used in this Report

This report contains no information from the Reference Information Base.

Candidate Information for the Reference Information Base

This report contains no candidate information for the Reference Information Base.

Candidate Information for the Site & Engineering Properties Data Base

This report contains no candidate information for the Site and Engineering Properties Data Base.

DISTRIBUTION LIST

- | | |
|--|--|
| <p>1 J. W. Bartlett, Director (RW-1)
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> | <p>1 S. J. Brocoun (RW-22)
Analysis and Verification Division
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> |
| <p>1 F. G. Peters, Deputy Director (RW-2)
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> | <p>1 D. D. Shelor (RW-30)
Office of Systems and Compliance
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> |
| <p>1 T. H. Isaacs (RW-4)
Office of Strategic Planning
and International Programs
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> | <p>1 J. Roberts (RW-33)
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> |
| <p>1 J. D. Saltzman (RW-5)
Office of External Relations
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> | <p>1 G. J. Parker (RW-332)
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> |
| <p>1 Samuel Rousso (RW-10)
Office of Program and Resources
Management
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> | <p>1 Associate Director (RW-40)
Office of Storage and Transportation
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> |
| <p>1 J. C. Bresee (RW-10)
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> | <p>1 Associate Director (RW-50)
Office of Contract Business
Management
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> |
| <p>1 C. P. Gertz (RW-20)
Office of Geologic Disposal
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> | <p>1 C. G. Russomanno (RW-52)
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
1000 Independence Avenue, S.W.
Washington, D.C. 20585</p> |

- 1 D. U. Deere, Chairman
Nuclear Waste Technical
Review Board
1100 Wilson Blvd. #910
Arlington, VA 22209-2297
- 1 Dr. John E. Cantlon
Nuclear Waste Technical Review Board
1795 Bramble Dr.
East Lansing, MI 48823
- 1 Dr. Melvin W. Carter
Nuclear Waste Technical Review Board
4621 Ellisbury Dr., N.E.
Atlanta, GA 30332
- 1 Dr. Donald Langmuir
Nuclear Waste Technical Review Board
109 So. Lookout Mountain Cr.
Golden, CO 80401
- 1 Dr. D. Warner North
Nuclear Waste Technical Review Board
Decision Focus, Inc.
4984 El Camino Real
Los Altos, CA 94062
- 1 Dr. Dennis L. Price
Nuclear Waste Technical Review Board
1011 Evergreen Way
Blacksburg, VA 24060
- 1 Dr. Ellis D. Verink
Nuclear Waste Technical Review Board
4401 N.W. 18th Place
Gainesville, FL 32605
- 5 C. P. Gertz, Project Manager
Yucca Mountain Project Office
U.S. Department of Energy
P.O. Box 98608--MS 523
Las Vegas, NV 89193-8608
- 1 C. L. West, Director
Office of External Affairs
DOE Field Office, Nevada
U.S. Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-85180
- 12 Technical Information Officer
DOE Field Office, Nevada
U.S. Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518
- 1 P. K. Fitzsimmons, Director
Health Physics & Environmental
Division
DOE Field Office, Nevada
U.S. Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518
- 1 E. R. Elle, Director
Environmental Protection Division
DOE Field Office, Nevada
U.S. Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518
- 1 Repository Licensing & Quality
Assurance Project Directorate
Division of Waste Management
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
- 1 Senior Project Manager for Yucca
Mountain Repository Project Branch
Division of Waste Management
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
- 1 NRC Document Control Desk
Division of Waste Management
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
- 1 P. T. Prestholt
NRC Site Representative
301 E. Stewart Ave.
Las Vegas, NV 89101
- 1 E. P. Binnall
Field Systems Group Leader
Building 50B/4235
Lawrence Berkeley Laboratory
Berkeley, CA 94720
- 1 Center for Nuclear Waste
Regulatory Analyses
6220 Culebra Road
Drawer 28510
San Antonio, TX 78284
- 3 L. J. Jardine
Technical Project Officer for YMP
Mail Stop L-204
Lawrence Livermore National
Laboratory
P.O. Box 808
Livermore, CA 94550

- 4 R. J. Herbst
Technical Project Officer for YMP
N-5, Mail Stop J521
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545
- 1 H. N. Kalia
Exploratory Shaft Test Manager
Los Alamos National Laboratory
Mail Stop 527
101 Convention Center Dr.
Suite 820
Las Vegas, NV 89109
- 1 J. F. Divine
Assistant Director for
Engineering Geology
U.S. Geological Survey
106 National Center
12201 Sunrise Valley Dr.
Reston, VA 22092
- 6 L. R. Hayes
Technical Project Officer
Yucca Mountain Project Branch--MS 425
U.S. Geological Survey
P.O. Box 25046
Denver, CO 80225
- 1 V. R. Schneider
Asst. Chief Hydrologist--MS 414
Office of Program Coordination
& Technical Support
U.S. Geological Survey
12201 Sunrise Valley Drive
Reston, VA 22092
- 1 R. B. Raup, Jr.
Geological Division Coordinator
MS 913
Yucca Mountain Project
U.S. Geological Survey
P.O. Box 25046
Denver, CO 80225
- 1 D. H. Appel, Chief
Hydrologic Investigations Program
MS 421
U.S. Geological Survey
P.O. Box 25046
Denver, CO 80225
- 1 E. J. Helley
Branch of Western Regional Geology
MS 427
U.S. Geological Survey
345 Middlefield Road
Menlo Park, CA 94025
- 1 Chief
Nevada Operations Office
U.S. Geological Survey
101 Convention Center Drive
Suite 860, MS 509
Las Vegas, NV 89109
- 1 D. Zesiger
U.S. Geological Survey
101 Convention Center Dr.
Suite 860 - MS509
Las Vegas, NV 89109
- 1 R. V. Watkins, Chief
Project Planning and Management
U.S. Geological Survey
P.O. Box 25046
421 Federal Center
Denver, CO 80225
- 1 A. L. Flint
U.S. Geological Survey
MS 721
P.O. Box 327
Mercury, NV 89023
- 1 D. A. Beck
U.S. Geological Survey
1500 E. Tropicana, Suite 201
Las Vegas, NV 89119
- 1 P. A. Glancy
U.S. Geological Survey
Federal Building, Room 224
Carson City, NV 89701
- 1 Sherman S. C. Wu
Branch of Astrogeology
U.S. Geological Survey
2255 N. Gemini Dr.
Flagstaff, AZ 86001
- 1 J. H. Sass
Branch of Tectonophysics
U.S. Geological Survey
2255 N. Gemini Dr.
Flagstaff, AZ 86001

- 1 DeWayne A. Campbell
Technical Project Officer for YMP
Bureau of Reclamation
Code D-3790
P.O. Box 25007
Denver, CO 80225
- 1 S. M. Dash
Science Applications International
Corp.
14062 Denver West Parkway, Suite 255
Golden, CO 80401
- 1 K. W. Causseaux
NHP Reports Chief
U.S. Geological Survey
421 Federal Center
P.O. Box 25046
Denver, CO 80225
- 1 V. M. Glanzman
U.S. Geological Survey
913 Federal Center
P.O. Box 25046
Denver, CO 80225
- 1 J. H. Nelson
Technical Project Officer for YMP
Science Applications International
Corp.
101 Convention Center Dr.
Suite 407
Las Vegas, NV 89109
- 2 SAIC-T&MSS Library
Science Applications International
Corp.
101 Convention Center Dr.
Suite 407
Las Vegas, NV 89109
- 1 Elaine Ezra
YMP GIS Project Manager
EG&G Energy Measurements, Inc.
Mail Stop D-12
P.O. Box 1912
Las Vegas, NV 89125
- 1 R. E. Jackson, Program Manager
Roy F. Weston, Inc.
955 L'Enfant Plaza, Southwest
Washington, D.C. 20024
- 1 Technical Information Center
Roy F. Weston, Inc.
955 L'Enfant Plaza, Southwest
Washington, D.C. 20024
- 1 D. Hedges, Vice President,
Quality Assurance
Roy F. Weston, Inc.
4425 Spring Mountain Road, Suite 300
Las Vegas, Nevada 89102
- 1 D. L. Fraser, General Manager
Reynolds Electrical & Engineering Co.
Mail Stop 555
P.O. Box 98521
Las Vegas, NV 89193-8521
- 1 R. F. Pritchett
Technical Project Officer for YMP
Reynolds Electrical & Engineering Co.
MS 408
P.O. Box 98521
Las Vegas, NV 89193-8521
- 1 B. W. Colston
General Manager & President
Las Vegas Branch
Raytheon Services Nevada
Mail Stop 416
P.O. Box 95487
Las Vegas, NV 89193-5487
- 1 R. L. Bullock
Technical Project Officer for YMP
Raytheon Services Nevada
Suite P250, MS 403
101 Convention Center Dr.
Las Vegas, NV 89109
- 1 R. E. Lowder
Technical Project Officer for YMP
MAC Technical Services
101 Convention Center Drive
Suite 1100
Las Vegas, NV 89109
- 1 C. K. Hastings, Manager
PASS Program
Pacific Northwest Laboratories
P.O. Box 999
Richland, WA 99352

1 A. T. Tamura
Science and Technology Division
Office of Scientific and Technical
Information
U.S. Department of Energy
P.O. Box 62
Oak Ridge, TN 37831

1 Carlos G. Bell, Jr.
Professor of Civil Engineering
Civil and Mechanical Engineering
Department
University of Nevada, Las Vegas
4505 South Maryland Parkway
Las Vegas, NV 89154

1 C. F. Costa, Director
Nuclear Radiation Assessment
Division
U.S. Environmental Protection
Agency
Environmental Monitoring Systems
Laboratory
P.O. Box 93478
Las Vegas, NV 89193-3478

1 ONWI Library
Battelle Columbus Laboratory
Office of Nuclear Waste Isolation
505 King Avenue
Columbus, OH 43201

1 T. Hay, Executive Assistant
Office of the Governor
State of Nevada
Capitol Complex
Carson City, NV 89710

3 R. R. Loux, Jr.
Executive Director
Nuclear Waste Project Office
State of Nevada
Evergreen Center, Suite 252
1802 North Carson Street
Carson City, NV 89710

1 C. H. Johnson
Technical Program Manager
Nuclear Waste Project Office
State of Nevada
Evergreen Center, Suite 252
1802 North Carson Street
Carson City, NV 89710

1 John Fordham
Water Resources Center
Desert Research Institute
P.O. Box 60220
Reno, NV 89506

1 Dr. Martin Mifflin
Water Resources Center
Desert Research Institute
2505 Chandler Avenue
Suite 1
Las Vegas, NV 89120

1 Eric Anderson
Mountain West Research-Southwest
Inc.
2901 N. Central Ave. #1000
Phoenix, AZ 85012-2730

1 Department of Comprehensive Planning
Clark County
225 Bridger Avenue, 7th Floor
Las Vegas, NV 89155

1 Planning Department
Nye County
P.O. Box 153
Tonopah, NV 89049

1 Lincoln County Commission
Lincoln County
P.O. Box 90
Pioche, NV 89043

5 Judy Foremaster
City of Caliente
P.O. Box 158
Caliente, NV 89008

1 Economic Development Department
City of Las Vegas
400 East Stewart Avenue
Las Vegas, NV 89101

1 Community Planning & Development
City of North Las Vegas
P.O. Box 4086
North Las Vegas, NV 89030

1 Director of Community Planning
City of Boulder City
P.O. Box 367
Boulder City, NV 89005

1	Commission of the European Communities 200 Rue de la Loi B-1049 Brussels BELGIUM	1	1510	J. C. Cummings
		1	1511	J. S. Rottler
		10	1511	R. R. Eaton
		1	1511	D. K. Gartling
		5	1511	P. L. Hopkins
		1	1511	M. J. Martinez
		1	1511	D. F. McTigue
		1	1511	A. J. Russo
		1	1512	A. C. Ratzel
		1	1513	R. D. Skocypec
		1	1513	R. C. Dykhuizen
		1	1513	C. E. Hickox
		1	1561	C. M. Stone
		1	1562	J. R. Asay
		5	3141	S. A. Landenberger
		8	3145	Document Processing for DOE/OSTI
		3	3151	G. C. Claycomb
		20	6341	WMT Library
		1	6410	D. J. McCloskey, Actg.
		2	6424	N. E. Bixler
		1	8245	R. J. Kee
		1	8523-2	Central Technical Files
2	M. J. Dorsey, Librarian YMP Research and Study Center Reynolds Electrical & Engineering Co., Inc. MS 407 P.O. Box 98521 Las Vegas, NV 89193-8521			
1	Amy Anderson Argonne National Laboratory Building 362 9700 So. Cass Ave. Argonne, IL 60439			
1	6300	T. O. Hunter		
1	6310	T. E. Blejwas, Actg.		
1	6310A	L. E. Shephard		
1	6312	F. W. Bingham		
1	6312	H. A. Hockery		
1	6312	P. G. Kaplan		
3	6312	F. C. Lauffer		
1	6312	M. L. Wilson		
1	6313	L. S. Costin		
1	6313	M. E. Fewell		
1	6313	S. R. Sobolik		
1	6313	A. H. Treadway		
1	6315	F. B. Nimick, Actg.		
1	6315	R. J. Glass		
1	6315	M. D. Siegel		
1	6316	R. P. Sandoval		
2	6318	R. J. Macer for 100/12149/SAND90-2542/QA		
1	6319	R. R. Richards		

END

**DATE
FILMED**

9 / 1 / 92