Title

FINITE ELEMENT ANALYSIS OF LAMINAR/ TURBULENT FLOW IN POROUS AND FRACTURED MEDIA

Authors

Elsworth, D.
Sitar, N.
Goodman, R.E.

Publication Date

1985

LBL-19105
UC-11

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

EARTH SCIENCES DIVISION

FINITE ELEMENT ANALYSIS OF LAMINAR/TURBULENT FLOW IN POROUS AND FRACTURED MEDIA

D. Elsworth, N. Sitar and R.E. Goodman

January 1985

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

FINITE ELEMENT ANALYSIS OF LAMINAR/TURBULENT FLOW IN POROUS AND FRACTURED MEDIA

by

Derek Elsworth
Nicholas Sitar
Richard E. Goodman

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

January, 1985

ACKNOWLEDGMENT

Financial support for this research was provided by the American Electric Power Corporation and the Earth Sciences Division of the Lawrence Berkeley Laboratory, through U.S. Department of Energy Contract No. DE-AC03-76SF00098.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The rational design of structures involving rock, requires ability to predict changes in the hydrologic regime caused by construction and operation of engineered structures. Such predictions will only be valid if realistic flow laws are used to characterise the behaviour at all flow velocities of interest. If high velocities are probable, then these flow laws must encompass turbulent conditions.

The effect of turbulent flow under boundary conditions of constant head produces pore pressures in excess of those produced by Darcian flow. This effect has been shown to have a detrimental effect on structures (Goodman and Elsworth, 1984) where increased pore pressures provide excess destabilising forces. Conversely, the existence of a nonlinear flow relationship linking discharge and potential gradient is a useful asset when attempting to reduce the ambiguity of pump tests in rock masses (Elsworth, 1984a).

This report synthesises existing characterisations of nonlinear laminar/turbulent flow in rock fissures and rockfill into a coherent numerical form. In this form, nonlinear flow laws have to be introduced into the finite element method to solve the resulting nonlinear differential equations.

2. HYDRAULIC CHARACTERISTICS OF FLUID FLOW

For application to porous media flow with low flow velocities throughout the domain, a linear Darcian relationship is usually adequate to define the conductive characteristics. In this form, the driving total head gradient is related to discharge by a coefficient of permeability hydraulic conductivity. For Darcian flow, the hydraulic conductivity is constant, irrespective of the driving gradient or velocity of flow. For nonlinear flow, this is no longer the case. The conductivity tensor is a function of potential gradient which, in turn, relates directly to the macroscopic flow velocity.

For the cases of nonlinear flow considered in this context, the relationships for irrotational potential flow may be identified symbolically as

$$\{q\} = [T]\{\nabla\phi\} \quad \text{Linear flow} \qquad (2.01)$$

$$\{q\} = [T(\nabla\phi)]\{\nabla\phi\} \quad \text{Nonlinear flow} \qquad (2.02)$$

where q is a vector containing discharge terms, $\nabla\phi$ is a vector containing gradient terms and T is a transmissivity tensor. The terms within the transmissivity tensor, T, for the two above expressions are only equivalent over the linear, laminar range of flow velocities.

Typical applications for nonlinear flow theory include characterisation of flow in rock fissure networks, coarse gravels and rockfill structures. These materials have a relatively open structure with large voids. The voids present zones of extremely high conductivity that are capable of returning high flow velocities under relatively modest potential gradients. The individual parameters necessary to describe flow in fissure networks and rockfill structures will be treated separately herein.

2.1. Fissure Flow

For tractable analysis of fractured rock masses, the conductivity of the matrix material is commonly ignored in formulation of the problem. This simplification is reasonable for steady state analysis since the ratios of matrix to fissure conductivities are commonly low.

2.1.1. Linear flow

Analytical treatment of fissure flow is possible if the conduit is idealised as two parallel plates. This analogy is relatively consistent for saturated, laminar, incompressible flow where fluid transmission within the wall rock is negligible. Wall roughness must be slight in relation to the mean fissure aperture. The parallel plate analogy yields the transmissivity of a fissure as

$$T = \frac{gb^3}{12\nu} \qquad (2.03)$$

where T = transmissivity, g = gravitational acceleration, b = mean fissure aperture and ν = fluid kinematic viscosity. Much experimental work has been completed to increase the fidelity of this relationship for fissures with significant wall roughness. However, for the work presented here, the characterisation studies performed by Louis (1969) are utilised without modification.

The geometry of a single, plane section of fissure is illustrated in Figure 2.1. From experimental studies using simulated rock fissure profiles, it was possible to formulate the conductivity terms for non tapering flow within the laminar regime. The characterisations are a function of the fissure relative roughness (k/2b) where k represents the wall double amplitude roughness.

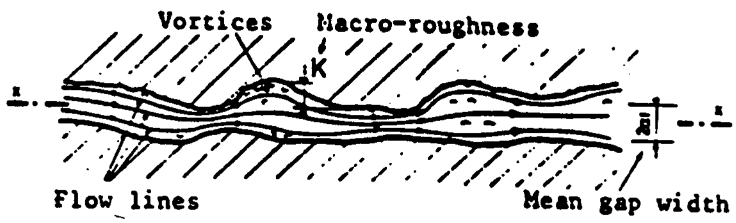

FIGURE 2.1 The geometry of a single fissure. (after Louis, 1969)

RANGE OF VALIDITY	HYDRAULIC CONDUCTIVITY (LT^{-1})
$0 < k/2b < 0.033$	$\frac{gb^2}{12\nu}$
$0.033 < k/2b < 0.5$	$\frac{gb^2}{12\nu(1+8.8(k/2b)^{3/2})}$

TABLE 2.1 Equivalent hydraulic conductivities (Louis, 1969)

The relationships documented in Table 2.1 have been validated against recently available data (Iwai, 1976). For more detailed treatment of this aspect, the interested reader is referred to Elsworth (1984b).

2.1.2 Nonlinear flow

The nonlinearity inferred in this context is that discharge rate is no longer a unique function of hydraulic gradient. The transmissivity tensor for the fissure is consequently a nonlinear function of gradient and may be expressed symbolically as

$$\{q\} = [T(\nabla\phi)]\{\nabla\phi\}$$

The nonlinearity results from the mixed interference of inertial and kinetic effects, both of which may be substantial for high velocity flow.

Inertial effects result from spatial accelerations within the flow domain. Accelerations result from flow convergence and may be envisaged to occur adjacent to well bores or in fissures tapering in the direction of flow. This effect is not exhibited in plane, non-tapering flow situations.

The severity of inertial effects may be quantified by recourse to the dimensionless inertial loss factor (η). This parameter (Iwai, 1976) is defined as

$$\eta = \frac{|v \;\partial v/\partial x_1|}{|-g\,\partial\phi/\partial x_1|} \qquad (2.04)$$

and may be evaluated for any given flow configuration. It has been found that inertial effects are only of significance if the inertial factor (η) is in excess of 0.5. If η values are maintained below this threshold, inertial effects may be neglected.

Kinetic effects result from head losses at increased flow velocities being dependent on velocity head in addition to the potential head. For low velocity flows, the kinetic head is small in comparison with potential head and may be neglected. However, for flow in fissure systems, kinetic losses may account for a significant portion of the total losses.

In addition to kinetic losses, turbulent flow may be manifest at increased flow velocities. The onset of turbulence may be indexed by recourse to the critical Reynolds number. For flow in rock fissures, the Reynolds number may be defined as,

$$R_e = \frac{2bv}{\nu} \qquad (2.05)$$

where v = velocity, b = mean fissure aperture and ν = fluid kinematic viscosity. For rock fissures, the critical Reynolds number commonly lies between 100 and 2300 depending on joint wall relative roughness.

In testing the hydraulic conductivity of fissures it is not possible to separate the effects of kinetic and turbulent loss explicitly since both effects may be manifest simultaneously and to varying degrees. A nonlinear criterion may be identified to describe this phenomenon. Neglecting first the effect of turbulence, the form of this relationship may be deduced from consideration of the summed head losses for a real fluid. The head loss (h_L) may be stated according to the Bernoulli equation as

$$\Delta\left(\frac{p}{\gamma_w}\right) + \Delta z + \Delta\left(\frac{v^2}{2g}\right) = \Delta h_L \qquad (2.06)$$

where P = pressure, γ_w = unit weight of fluid, z = elevation head, v = velocity and g = gravitational acceleration. Considering total head to be equal to $\phi = (\frac{p}{\gamma_w} + Z)$ then

$$\frac{\Delta\phi}{\Delta L} + \frac{1}{\Delta L}\frac{\Delta(v^2)}{2g} = \frac{\Delta h_L}{\Delta L} \tag{2.07}$$

and L = an arbitrary unit length in the direction of flow. Assuming first that the velocity may be deduced from Darcy's law neglecting velocity effects, Darcy's law may be stated as

$$\frac{\Delta\phi}{\Delta L} = \frac{v}{K} \tag{2.08}$$

where K = hydraulic conductivity. On substitution into (2.07)

$$\frac{v}{K} + \frac{1}{\Delta L 2g}\Delta(v^2) = \frac{\Delta h_L}{\Delta L} \tag{2.09}$$

Thus head loss per unit length is a function of both velocity and velocity to the second power; K is an experimentally deduced coefficient.

Two flow laws have been proposed to relate flow velocity to hydraulic gradient in a nonlinear manner. These relationships are reported in recent literature (Volker (1969), Trollope (1971)). The first of these is the Forchheimer law using a polynominal expression to describe the velocity dependent gradient ($\nabla\phi$) as

$$\nabla\phi = \underline{a}v + \underline{b}v^2 \tag{2.10}$$

where $\underline{a}$ and $\underline{b}$ are constants and v is the flow velocity in the direction of maximum gradient. Constants $\underline{a}$ and $\underline{b}$ are determined experimentally and are properties of the fluid and transmitting medium. The constants are only applicable over a given range of flow velocities, outside which, revised parameters must be substituted. For low velocity flows, constant a is much larger than b and the expression reduces to the normal Darcy law. The validity of this relationship has been inferred from both experimental results and from manipulation of the Navier-Stokes equations.

A second relationship between hydraulic gradient and flow velocity is the Missbach law in the form of a power function

$$\nabla\phi = \underline{c}v^{m} \qquad (2.11)$$

where c is a proportionality constant and m is an exponent ranging between 1 and 2. The magnitudes of both c and m vary with flow velocity although both may be approximately constant over a given range. The validity of the law has been confirmed for fissure flow in simulated rock fractures by Louis (1969). These flow laws enable the conductance of a fissure of known geometry to be determined uniquely as a function of flow velocity. The Missbach law in equation (2.11) may be rewritten as

$$v = -K\nabla\phi^{\alpha} \qquad (2.12)$$

where K represents the hydraulic conductivity of the fissure; α is equal to unity for laminar flow and 1/2 for turbulent flow in a rough walled fracture. Suitable values of K and the range of validity of the laminar regime are obtained from the work of Louis (1969). Fissure conductivity is a function of aperture and fissure wall relative roughness. A critical Reynolds number is ascribed for any given fissure relative roughness to delineate the transition from laminar to turbulent flow. The empirical results for the constants K and α are shown in Table 2.2. The range of validity of these parameters as a function of the two dimensionless parameters of relative roughness (k/2b) and Reynolds number are shown in Figure 2.2. The form of nonlinear conductivity is shown in Figure 2.3 as a function of gradient beyond a threshold gradient for a number of representative fissure apertures.

The two phases of flow in the laminar and turbulent regime give

Laminar	$v = K_L \nabla\phi$	$\nabla\phi = \dfrac{v}{K_L}$	(2.13)
Turbulent	$v = K_T(\nabla\phi)^{1/2}$	$\nabla\phi = \dfrac{v^2}{K_T^2}$	(2.14)

The Missbach relationship implies that flow is linear until the onset of turbulence. Beyond this threshold, the effect of kinetic head in contributing to total head loss is considered. The implication of this is that head loss due to velocity effects is only considered beyond the turbulent threshold rather than as a continuous function of velocity as in the Forchheimer relationship. The Missbach relationship is analogous to the Forchheimer relationship where the coefficients a and b may be substituted as

Laminar	$a = \dfrac{1}{K_L}$	$b = 0$	(2.15)
Turbulent	$a = 0$	$b = \dfrac{1}{K_T^2}$	(2.16)

Throughout this work, the Missbach flow law is used to characterise the full flow regime. This is the form in which the only authoritative experimental data on the subject is reported (Louis, 1969).

2.2 Nonlinear Flow in Rockfill

The large open voids in rockfill can yield flow velocities in the turbulent regime at gradients of engineering significance. Unlike

HYDRAULIC ZONE	HYDRAULIC CONDUCTIVITY (LT^{-1})	EXPONENT (α)
1	$\frac{gb^2}{12\nu}$	1.0
2	$\frac{1}{b}[\frac{g}{.079}\frac{2}{\nu}b^3]^{4/7}$	4/7
3	$4\ g^{1/2}\log[\frac{3.7}{k/D_h}]b^{1/2}$	1/2
4	$\frac{gb^2}{12\nu(1+8.8(k/D_h)^{3/2})}$	1.0
5	$4\ g^{1/2}\log[\frac{1.9}{(k/D_h)}]b^{1/2}$	1/2

N.B. $D_h = 2b$

TABLE 2.2 Equivalent hydraulic conductivities

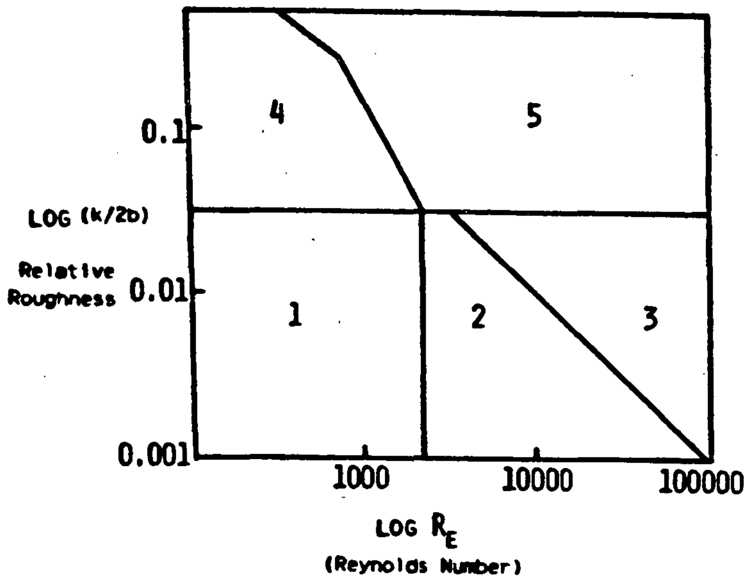

FIGURE 2.2 Hydraulic flow regions

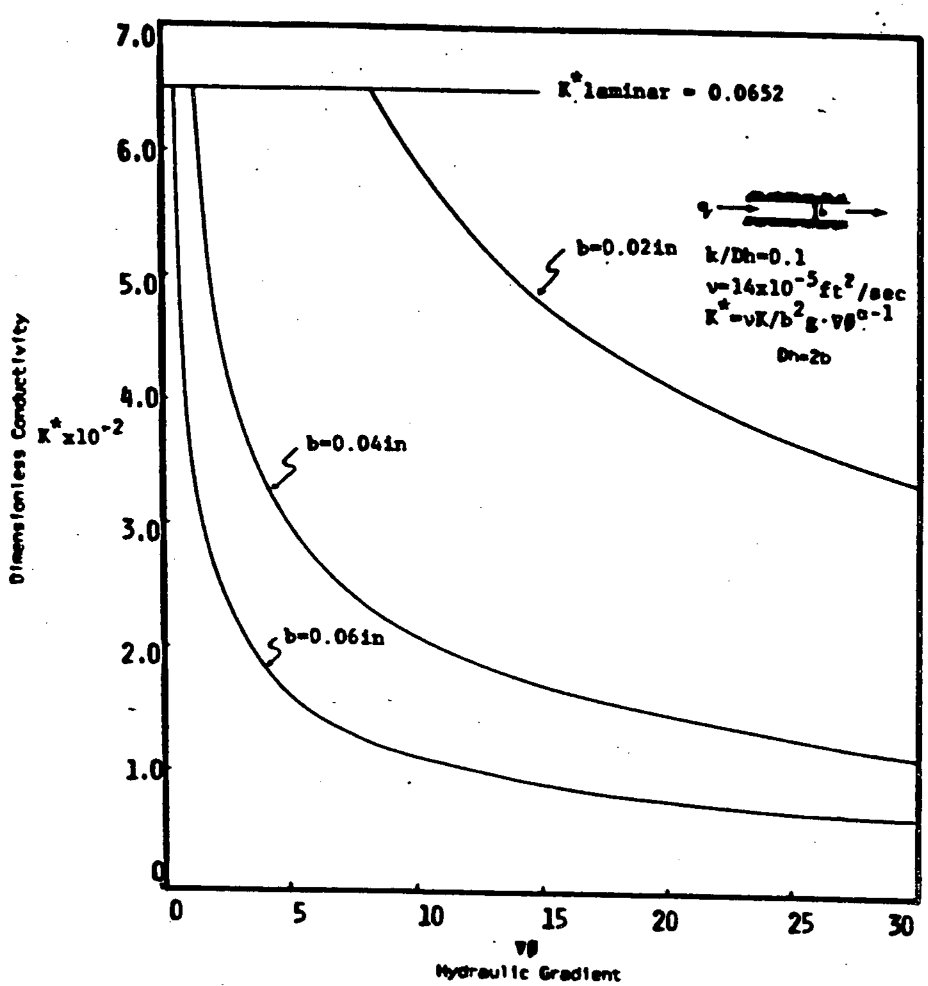

FIGURE 2.3 Variation in dimensionless conductivity with hydraulic gradient

fissure flow, it is reasonable to assume that flow in rockfill will always be turbulent. Characterisations of the conductivity of rockfills of different porosity are available in the literature (Leps, 1973). The relationship between bulk flow velocity and driving gradient may be given as

$$v = nWm^{0.5}(\nabla\phi)^{0.54} \qquad (2.17)$$

where n = rockfill porosity, m = mean hydraulic radius in dimensions of length and W = a constant of dimensions $L^{1/2}T^{-T}$ relating to the angularity of the constituent rock particles. These parameters may be tabulated for a reasonable range of mean rock sizes (Leps, 1973).

ROCK SIZE (in)	m (in)	$m^{0.5}$ ($in^{0.5}$)	$Wm^{0.5}$ (in/sec)
0.75	0.09	0.30	10
2	0.24	0.49	16
6	0.75	0.87	28
8	0.96	0.98	32
24	3.11	1.76	58
48	6.43	2.54	84

TABLE 2.3 Representative values of the coefficients required to characterise rockfill hydraulic conductivity (after Leps, 1973).

Equation (2.17) may be used similarly to the nonlinear relationships previously reported for fissure flow.

3. NUMERICAL FORMULATION FOR NONLINEAR FLOW

To solve problem geometries not amenable to analytical treatment, it is necessary to formulate the problem of nonlinear fluid flow for numerical solution. Presently available domain methods are extremely powerful and enable solution of complex geometries with multiple porous media interfaces. The finite element (FEM) and finite difference (FDM) methods are both in wide use, but because the finite element technique more easily handles complicated geometries, only it is addressed here. For treatment of nonlinear fluid flow by the finite difference method, the interested reader is referred to Elsworth (1984b).

3.1 Governing Differential Equations

In the following, specific treatment will be limited to flow within a two dimensional fissure as shown in Figure 3.1. This two dimensional geometry is, however, entirely general and may equally well be used to represent the cross sectional or plan projection of a rockfill aquifer.

Flow is assumed incompressible and restricted to within the plane of analysis (x_1, x_2). Total heads are considered throughout the analysis.

A constitutive relation is combined with continuity constraints to produce the governing differential equation as follows:

Constitutive relation -

$$v = -\underline{K}\nabla\phi \qquad (3.01)$$

where v = flow velocity

ϕ = total head $(z + p/\gamma_w)$

$\underline{K}$ = nonlinear hydraulic conductivity = $K(\nabla\phi_s)^{\alpha-1}$

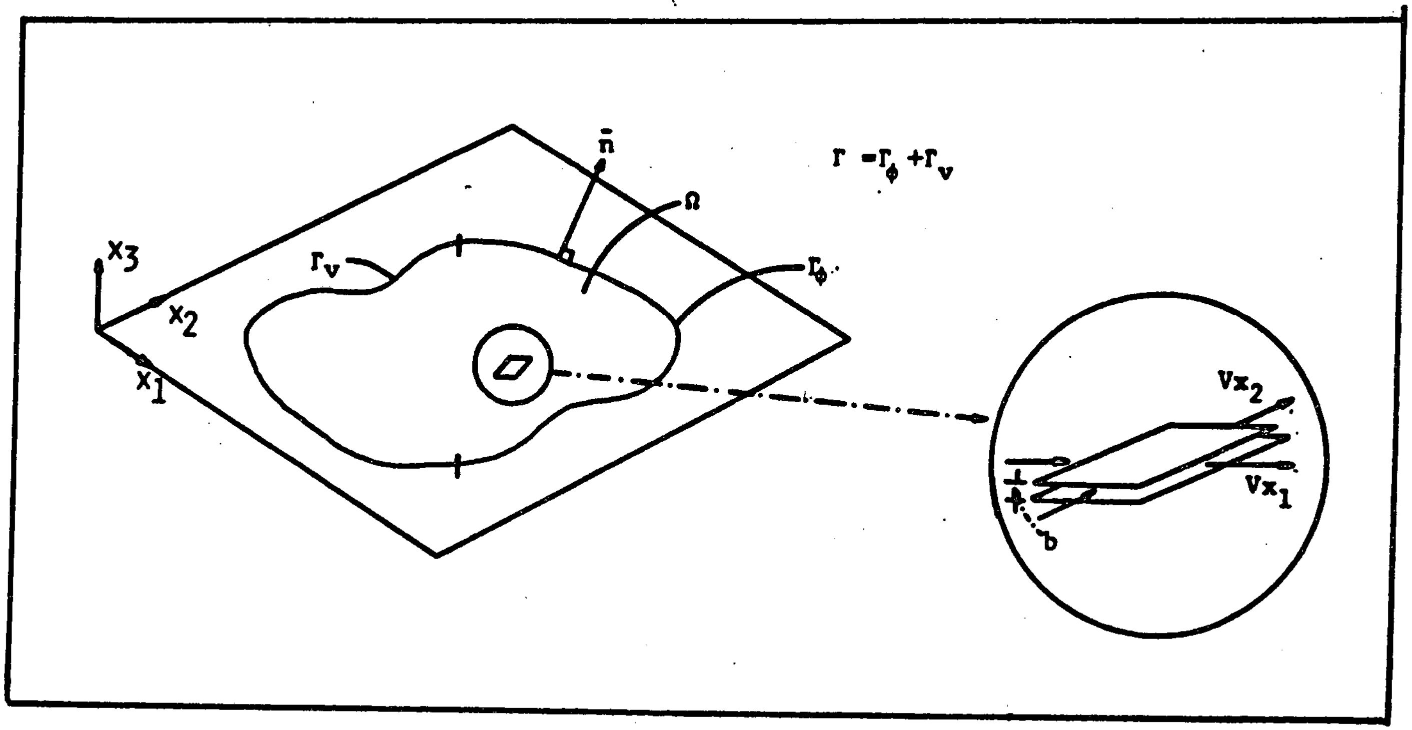

FIGURE 3.1 Flow domain for a plane fracture.

K, α = conductance term and exponent (Louis, 1969, Leps, 1973)

$\nabla\phi_s$ = maximum gradient within the plane of analysis

∇ = linear differential operator $(\partial/\partial x_1, \partial/\partial x_2)$

Continuity condition

$$-\nabla^T v - Q = 0 \qquad (3.02)$$

where Q = accumulation rate of fluid per unit volume and T identifies the matrix transpose.

For a domain within which fissure aperture may be expected to vary in a stepwise manner, the continuity condition may be restated as:

$$-b\nabla^T v - bQ = 0 \qquad (3.03)$$

Combining the constitutive relation (3.01) with continuity yields

$$\nabla^T b\underline{K}\nabla\phi - bQ = 0 \qquad (3.04)$$

Subject to the boundary conditions prescribed for the domain

$$\phi = \overline{\phi} \text{ on } \Gamma_\phi \qquad \overline{\phi} = \text{prescribed head}$$

$$bv_n = b\overline{v} \text{ on } \Gamma_q \qquad \overline{v} = \text{prescribed flux}$$

The flux normal to the boundary = $\overline{v} = \nabla^T\overline{n}$ where $\overline{n}$ is a vector of direction cosines representing the unit outward normal to the system. The boundary conditions must similarly satisfy continuity. Therefore, in generality

$$(b\underline{K}\nabla\phi)^T - b\overline{v} = 0 \qquad (3.05)$$

The two relationships (3.04) and (3.05) completely define the problem of turbulent flow where both are nonlinear in ϕ.

3.2 The Finite Element Method

The governing differential equations stated previously may be combined with the appropriate boundary conditions and integrated by parts to yield

$$\int_{\Omega} b\nabla^T \xi \underline{K} \nabla \phi d\Omega - \int_{\Omega} b\xi Q d\Omega - \int_{\Gamma_q} b\xi q d\Gamma = 0 \qquad (3.06)$$

where ξ is an unknown function varying within the domain, Q is the internal flux and q is the externally applied flux on the boundary Γ_q.

For the Galerkin formulation where the trial function used is $\phi = \Sigma N_i \phi_i$ and $\xi = N_i$, the boundary value problem reduces to:

$$[\int_{\Omega} b\nabla^T N_i \underline{K} \nabla N_j d\Omega]\phi - \int_{\Omega} bN_i Q d\Omega - \int_{\Gamma_q} bN_i q d\Gamma = 0 \qquad (3.07)$$

where $\underline{K}$ is given in equation (3.01).

The leftmost identity comprises a geometric conductance matrix where all terms are known physical properties of the domain excepting $\nabla\phi_s$ in $\underline{K}$. The resulting set of equations are therefore nonlinear for the case of turbulent flow where $\alpha \neq 1$.

Solution of equation 3.07 requires that the flow domain is discretised into a number of elements. Iterative manipulation of the resulting set of equations is required to obtain an elemental conductivity tensor compatible with the local flow velocity. A direct iterative solution is invoked, the iterations being purely explicit in time.

The finite element code discussed in this document utilises linear functional variation of total head. Quadrilateral or triangular elements may be used to discretise the domain of interest. The step by step procedure involved in the solution of a single problem is given in Table 3.1.

TABLE 3.1

Summary of the solution method for steady state laminar/turbulent flow in fissure networks

1. Define input. Geometry, element apertures (b), boundary conditions and convergence tolerance.
2. Form initial conductance matrix for laminar flow:

   $$\underline{q} = \underline{T}\,\underline{\phi}$$

3. Rearrange for boundary conditions
4. Solve system for $\underline{\phi}$: $\underline{\phi} = \underline{T}^{-1}\,\underline{q}$
5. From potential distribution (ϕ_u), reevaluate gradient dependant conductivities and reassemble conductance matrix:

   $$\underline{q} = \underline{T}(\underline{\phi})$$

6. Solve for potential(ϕ) unknowns
7. Repeat steps 5 and 6 until convergence tolerance is met
8. Output solution and stop

TABLE 3.1 Summary of the solution method for steady state laminar/turbulent flow in fissure networks.

4. VALIDATION STUDIES

The coding reported in this work has been validated against a number of analytical solutions. These solutions are applicable to fissure flow and flow in rockfill subject to axisymmetric flow conditions.

For the case of a horizontal fissure drained by a single well, the analytical solution has been developed by Amadei (1983) and is reported in Elsworth (1984b). Comparison of the numerical model with the analytical result is included in Figure 4.1 for an axisymmetric mesh containing 100 radial elements. Excellent agreement is exhibited between the two solutions for two different fissure apertures. The solution paths taken in the numerical work for both constant head and constant discharge boundary conditions are shown in Figure 4.2. The final discharges and heads are shown to be identical, regardless of the solution paths taken.

For fissure flow, the head distribution resulting from flow in a tapering fissure is shown in Figure 4.3. Representing the tapering fissure numerically by a series of constant aperture elements is shown to be entirely adequate for conditions of minor taper.

Finally, for axisymmetric flow in an unconfined rockfill reservoir, the results of analytical and numerical solution are presented in Figure 4.4. Once again, 100 radial elements were used to discretise the axisymmetric flow domain.

Analytically, the variation in head over the axisymmetric flow domain may be shown to be equivalent to

$$h(r) = \left[\frac{(h_w^\beta - h_e^\beta)}{(r_w^\gamma - r_e^\gamma)} \cdot (r^\gamma - r_e^\gamma) + h_e^\beta \right]^{1/\beta}$$

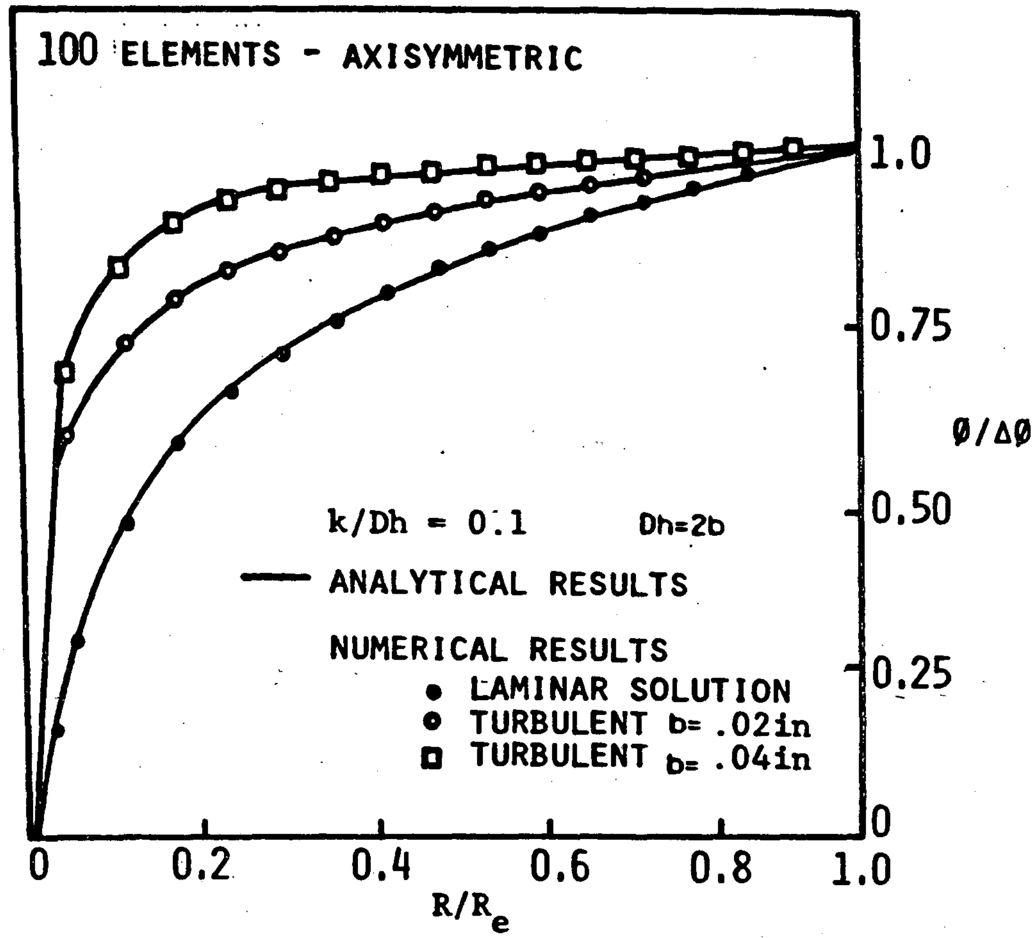

FIGURE 4.1 Validation of numerical solution against analytical results for laminar and turbulent flow - axisymmetric case.

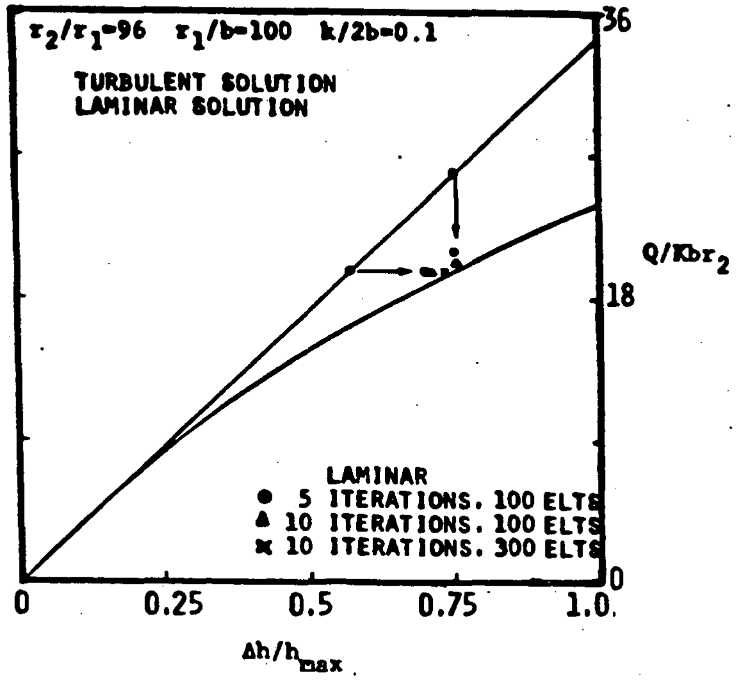

FIGURE 4.2 Variation of discharge as a function of driving head for a well intersecting a single horizontal fissure.

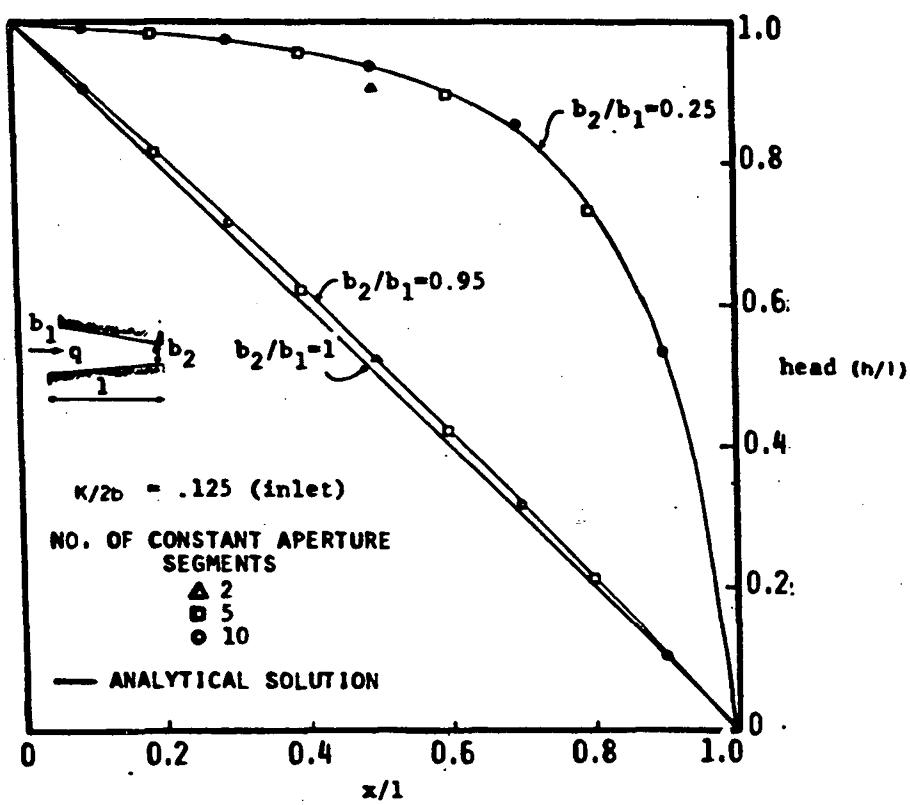

FIGURE 4.3 Head loss along a tapered fissure. Numerical and analytical results.

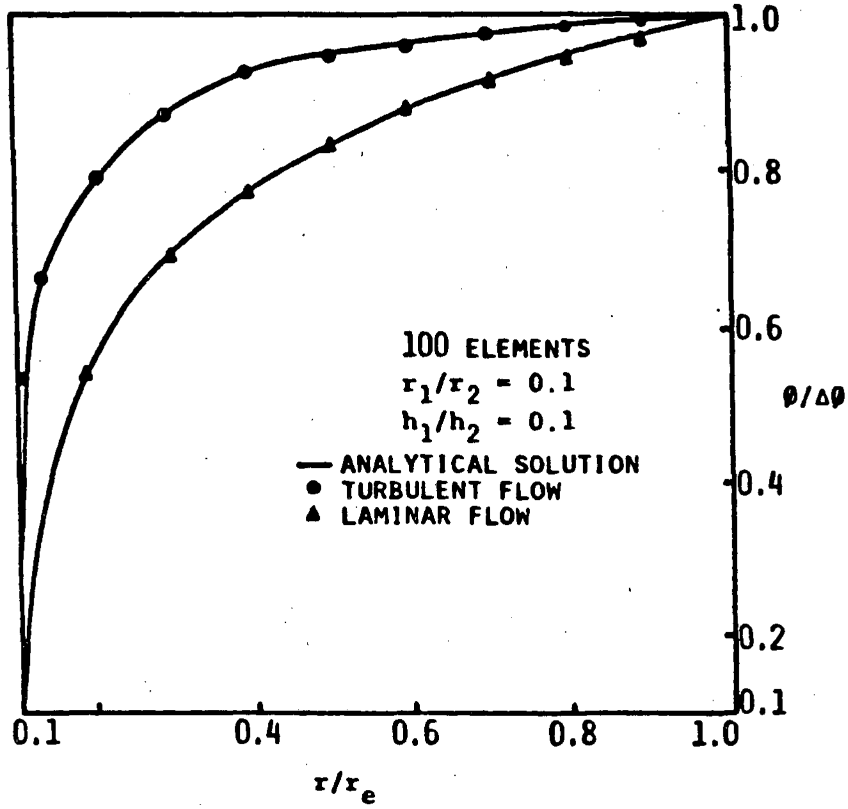

FIGURE 4.4 Head versus radial distance from well for unconfined flow in rockfill.

where h and r refer to total heads and radii from the well center and subscripts w and e refer to the wellbore and external boundaries, respectively. The coefficients β and γ are functions of the exponent α, which for rockfill is equal to 0.54.

$$\text{Thus} \qquad \beta = \frac{1+\alpha}{\alpha}$$

$$\text{and} \qquad \gamma = \frac{\alpha-1}{\alpha}$$

In assembling this solution, it is assumed that the Dupuit-Forchheimer assumption may be invoked. The results presented in Figure 4.4 show excellent agreement between the analytical and numerical solutions.

It is concluded that the numerical formulation, proposed and developed herein, is capable of modelling mixed laminar/turbulent flow in fissure systems and porous media.

5. REFERENCES

Amedei, M. (1983). Rock Anisotropy and the Theory of Stress Measurements, Springer-Verlog, Berlin, 478 pp.

Goodman, R.E. and Elsworth, D. (1984). Performance Evaluation of a Prototype Dam Undercut by a Non-Extensive Foundation Crack. Report Submitted to the American Electric Power Corporation. September, 91 pages.

Elsworth, D. (1984a). Interpretation of Single Borehole, Constant Pressure Pump Tests Using Nonlinear Flow Laws. Report Submitted to the Earth Sciences Division, Lawrence Berkeley Laboratory, Berkeley, California. September, 34 pages.

Elsworth, D. (1984b). Laminar and Turbulent Flow in Rock Fissures and Fissure Networks. Ph.D. thesis, Department of Civil Engineering, University of California, Berkeley, 191 pages.

Iwai, K. (1976). Fundamental Studies of Flow Through a Single Fracture. Ph.D. Thesis, Department of Civil Engineering, University of California, Berkeley.

Leps, T.M. (1973). Flow Through Rockfill. In: Embankment Dam Engineering. Casagrande Volume. Wiley, New York, pp. 87-107.

Louis, C. (1969). A Study of Groundwater Flow in Jointed Rock and its Influence on the Stability of Rock Masses. Imperial College Rock Mechanics Research Report, No. 10, September, 90 pages.

Trollope, D.H., Stark, K.P. and Volker, R.E. (1971). Complex Flow Through Porous Media. Australian Geomechanics Journal, Vol. 91, No. 1.

Volker, R.E. (1969). Nonlinear Flow Through Porous Media by Finite Elements. Journal of the Hydraulics Division, ASCE, HY6, November, pp. 2093-2114.

APPENDICES

APPENDIX 1 SUBROUTINE ORGANISATION

The coding reported in this document consists of twelve slave subroutines coordinated by a main master subroutine. The identities of the component subroutines and their respective tasks are summarised in the following:

SUBROUTINE	PURPOSE
MAIN	Sets the dimension of the blank common array and defines input (IN) and output (IOUT) logical operator numbers.
FLOW	Reads the problem description and assigns storage within the blank common. Controls the direct iteration sequencing for turbulent flow analysis.
CENTR	Acts as the main calling subroutine for the program. Calls input subroutine, controls generation of the elemental and global conductivity matrices and arranges solution of equations.
INPUT	Responsible for reading and generating input data as appropriate.
ELEM	Coordinates generation of elemental conductance matrices making appropriate tansformations for fissure in three dimensional space.
QUAD	Assembles the elemental conductivity matrix.
QUADS	Assembles the elemental conductivity matrix by isoparametric formulation.
TRIAG	Evaluates geometric properties of triangular elements using Cramer's rule.
SOLVE	Equation solution package using Gaussian elimination.
STDM	Evaluates the geometric components of the conductivity matrix for isoparametric formulation.
NLIN	Evaluates the gradient dependent hydraulic conductivity tensor for each element under applied nodal heads.
TFORM	Transforms an arbitrarily oriented fissure element in three dimensional space to two dimensional space.
RFORM	Re-transforms the two dimensional finite element to true three dimensional space.

APPENDIX 2 NONLIN USER'S MANUAL

PROGRAM INDENTIFICATION

NONLIN: Analysis of nonlinear (laminar/turbulent) potential flow in three dimensional networks of rock fissures and two dimensional rockfill structures. October, 1984.

Developed by: D. Elsworth
Source deck by: N. Sitar
University of California, Berkeley

CONTENTS

A2.1 Program Description

NONLIN models steady state flow of three dimensional networks of rock fissures and two dimensional plane and axisymmetric rockfill structures. Boundary and internal conditions of constant flux or head may be prescribed. Solutions to linear and nonlinear flow problems are possible with the coding as currently developed. The nonlinearity refers to a gradient dependent hydraulic conductivity for each element within the mesh.

The present version contains two dimensional elements with a linear shape function. Formulations for both isoparametric and direct evaluation of the geometric conductivity matrices are included.

A2.2 Program Capacity

A blank common of dimension <u>n</u> is supplied to allow variable dimensioning of all arrays to facilitate dynamic storage allocation techniques. The maximum size for n attempted on the IBM 3081 comprised 62,000 locations. Under this allocation, a total of 900 elements and 975 nodes were accommodated.

A minimum value of n for any situation is given by

$$n = 13 + 11*NUMNP + 9*NUMEL + 4*NUMAT + NUMAT*IBAN$$

where

NUMNP = Maximum number of nodal points

NUMEL = Maximum number of elements

NUMAT = Maximum number of materials (Equal to NUMEL for turbulent flow)

IBAN = Maximum semi bandwidth

A2.3 Program Input Data

The following sections describe the necessary sequence of cards which define a given structure to be analysed.

I. PROBLEM IDENTIFICATION CARD (20I4) - One card

Columns	Variable	Description
1-80	-	Problem title

II. PROBLEM SIZE CARD (10I5) - One card

Columns	Variable	Description
1-5	NUMNP	Total number of nodal points in structure
6-10	NUMEL	Total number of elements in structure
11-15	NC	Total number of constant head nodes
16-20	NC	Total number of point flux nodes
21-25	NUMAT	Total number of material types
26-30	ICHK	Check for type of flow 0 = Unconfined flow within a horizontal section 1 = Confined flow within a horizontal section or within arbitrarily inclined fissures 2 = Flow within a vertical section
31-35	ILIST	Dummy parameter for printed output 0 = No listing of input data 1 = Listing of input data
36-40	NINT	Order of Gaussian integration 0 = Non-isoparametric quadrilaterals and triangular elements 2 = Isoparametric quadrilaterals
41-45	IRAD	Flow geometry 0 = Plane flow 1 = Axisymmetric flow
46-50	IFLOW	Flow law type 0 = Linear flow 1 = Nonlinear fissure flow 2 = Nonlinear rockfill flow

III. NODAL POINT DATA (2I5, 3F10.0)

Columns	Variable	Description
1-5	N	Node number
6-10	INC	Increment for nodal point generation
11-20	X	X coordinate of node
21-30	Y	Y coordinate of node
31-40	Z	Z coordinate of node

NB. System of coordinates use a right hand cartesian system. The program has the capability of incrementing between specified nodes. If INC is left blank it is set equal to unity.

IV. NODAL HEAD DATA (I5, F10.0) NC cards

Columns	Variable	Description
1-5	N	Node number
6-15	HEAD	Head prescribed at node N

NB. There must be NC card supplied with this option.

V. NODAL POINT FLUX (I5, F10.0) NF cards

Columns	Variable	Description
1-5	N	Node number
6-15	FLUX	Flux prescribed at node N

NB1. Positive values signify recharge into the system.
Negative values signify discharge out of the system.

NB2. There must be NF card supplied with this option.

VI. ELEMENT DATA (6I5, 2F10.0, I5)

Columns	Variable	Description
1-5	M	Element number
6-10	IND	Node number of Ith node of the element
11-15	JND	Node number of Jth node of the element
16-20	KND	Node number of Kth node of the element
21-25	LND	Node number of Lth node of the element
26-30	MND	Material number for the element
41-50	Q	Distributed flux for the element
51-55	INC	Increment for node generation of each element

NB1. For triangular elements, LND=KND and this should be set as input.

NB2. MND may left blank to zero if only one material type is present.

NB3. Nodes must always be given in counterclockwise order with respect to the global system.

NB4. Where elements are generated, the material number is also incremented if NUMAT = NUMEL.

VII. MATERIAL PROPERTIES (I5, 2F10.0)

Columns	Variable	Description
1-5	N	Material number
6-10	CON2	Value of fissure aperture (b) for fissure flow Value of effective conductivity ($nWm^{3/2}$) for rockfill. See Table 2.3.

NB1. Where NUMAT = NUMEL, the program will increment between specified elements.

VIII. TURBULENT FLOW DATA (I5, 2F10.0, E10.0) One card

Columns	Variable	Description
1-5	ITURBM	Maximum number of iterations for nonlinear flow analysis.
6-15	GRAV	Acceleration due to gravity
16-25	ZO	Fissure absolute roughness (k) (Tables 2.1 and 2.2)
26-35	RNU	Fluid kinematic viscosity (ν) (Tables 2.1 and 2.2)

IX. BLANK CARD - One card

APPENDIX 3 - LISTING OF NONLIN

```
C **************************************************************************
C
C    STEADY STATE GALERKIN FINITE ELEMENT SOLUTION
C
C
C    ************************************************************
C    *                                                          *
C    *    N. SITAR  UCB FEB. 1983 VERSION                       *
C    *    D. ELSWORTH UPDATE FOR LAMINAR/TURBULENT FLOW         *
C    *        IN FRACTURED MEDIA (3D)                           *
C    *                                                          *
C    ************************************************************
C
C ***************
C    INPUT DATA
C ***************
C
C 1.  IDENTIFICATION CARD   (20A4)
C       COL.  1-80 -- ANY CHARACTERS
C
C 2.  PROBLEM SIZE INFORMATION   (10I5)
C       COL.  1-5    NUMNP-NUMBER OF NODAL POINTS
C             6-10   NUMEL-NUMBER OF ELEMENTS
C             11-15  NC-NUMBER OF CONSTANT HEAD NODES
C             16-20  NF-NUMBER OF POINT FLUX NODES
C             21-25  NUMAT-NUMAT OF MATERIAL TYPES
C             26-30  ICHK-CHECK FOR TYPE OF FLOW
C                        0=UNCONFINED FLOW
C                        1=CONFINED FLOW
C                        2=VERTICAL SECTION
C             31-35  ILIST-DUMMY PARAMETER
C                        0=NO LISTING OF INPUT DATA
C                        1=LISTING OF INPUT DATA
C             36-40  NINT-ORDER OF GAUSSIAN INTEGRATION
C                        0= 4CST QUADRILATERALS AND TRIANGLES
C                        2= ISOPARAMETRIC ELEMENTS
C             41-45  IRAD-FLOW GEOMETRY
C                        0= NON AXISYMMETRIC FLOW
C                        1= AXISYMMETRIC FLOW
C             46-50  IFLOW-FLOW TYPE
C                        0=LINEAR FLOW
C                        1=NONLINEAR FISSURE FLOW
C                        2=NONLINEAR ROCKFILL FLOW
C
C 3.  NODAL POINT DATA   (2I5,3F10.0)
C     * PROGRAM WILL GENERATE DATA BETWEEN SPECIFIED NODES *
C     * ICDE MUST BE CONSTANT DURING DATA GENERATION *
C     * INC MAY BE LEFT BLANK IF EQUAL TO UNITY *
C       COL.  1-5    N-NODE NUMBER
C             6-10   INC-INCREMENT FOR NODE GENERATION
C             11-20  X-X COORDINATE OF NODE
C             21-30  Y-Y COORDINATE OF NODE (VERTICAL FOR ICHK=2)
C             31-40  Z-Z COORDINATE OF NODE
C
C 4.  NODAL HEAD DATA  (I5,F10.0)
C       COL.  1-5    N-NODE NUMBER
C             6-15   HEAD AT NODE N
C
C 5.  POINT FLUX DATA   (I5,D10.2)
C     * WILL NEED NF/5 CARDS (OMIT IF NF=0) *
C     * POSITIVE VALUES SIGNIFY RECHARGE INTO SYSTEM *
C     * NEGATIVE VALUES SIGNIFY DISCHARGE OUT OF SYSTEM *
C       COL.  1-5   N-NODE NUMBER
C       COL.  6-15  DUMMY-FLUX AT THE NODE N
C
C 6.  ELEMENT DATA   (6I5,2F10.0,I5)
C     * PROGRAM WILL GENERATE OMMITTED ELEMENTS *
C     * MND MUST BE CONSTANT DURING DATA GENERATION *
C     * INC SET TO UNITY IF BLANK *
C     *** NODES MUST BE GIVEN IN COUNTERCLOCKWISE ORDER ***
C     *
C       COL.  1-5    M-ELEMENT NUMBER
C             6-10   IND-NODE NUMBER OF ITH NODE OF ELEMENT
C             11-15  JND-NODE NUMBER OF JTH NODE OF ELEMENT
C             16-20  KND-NODE NUMBER OF KTH NODE OF ELEMENT
C             21-25  LND-NODE NUMBER OF LTH NODE OF ELEMENT
C                  ** SET KND=LND FOR TRIANGULAR ELEMENTS **
C             26-30  MND-MATERIAL NUMBER
C                  ** CAN BE BLANK OR ZERO FOR MATERIAL #1 **
C             31-40  ALPHA-ANGLE BETWEEN PRINCIPAL CONDUCTIVITY
C                         DIRECTION AND X AXIS
C             41-50  Q-DISTRIBUTED FLUX
C             51-55  INC-INCREMENT FOR NODE GENERATION
C
C 7.  MATERIAL PROPERTIES DATA    (I5,2F10.0)
C     * WILL NEED NUMAT CARDS *
C       COL.  1-5    N-MATERIAL NUMBER
C             6-15   CON1-VALUE OF MAXIMUM CONDUCTIVITY
C             16-25  CON2-VALUE OF MINIMUM CONDUCTIVITY
C                  ** FOR TURBULENT OPTION NUMAT = NUMEL  **
C                  ** CON1 - SET BLANK                   **
C                  ** CON2 - FISSURE APERTURE            **
C                  ** INCREMENTED BETWEEN ELEMENTS       **
C
C
C 8.  TURBULENT FLOW DATA  (I5,2F10.0,1E10.0)
C     * WILL NEED 1 CARD *
C     * INSERT BLANK CARD FOR NO TURBULENT FLOW *
C       COL   1-5    ITURBM-MAXIMUM NUMBER OF ITERATIONS
C             6-15   GRAV  -GRAVITATIONAL ACCELERATION
C             16-25  ZO    -FISSURE ABSOLUTE ROUGHNESS
C             26-35  RNU   -KINEMATIC VISCOSITY
C
C **************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON S(60000)
      COMMON/IO/ IN,IOUT
      COMMON/FRAC/XT(4),YT(4),II(4)
      NSCM=60000
      IN=5
```

```
      IOUT=6
      CALL FLOW(S,NSCM)
      WRITE(IOUT,600)
  600 FORMAT (' ',///,'*****      END   OF   PROBLEM       *****')
      WRITE (IOUT,601)
  601 FORMAT ('1')
      STOP
      END
C
C
      SUBROUTINE FLOW (S,NSCM)
C
C     THIS SUBROUTINE READS THE PROBLEM DESCRIPTION AND ASSIGNS STORAGE
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(1)
      COMMON/IO/ IN,IOUT
      COMMON/CONST1/ TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),I1(4)
C
C     ****************************************************************
C
      READ (IN,500) TITLE
      READ (IN,501) NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT,IRAD,IFLOW
      WRITE (IOUT,600) TITLE
      IF (ICHK-1) 1,2,3
    1 WRITE (IOUT,611)
      GO TO 4
    2 WRITE (IOUT,612)
      GO TO 4
    3 WRITE (IOUT,613)
      GO TO 4
    4 CONTINUE
      IF(IRAD.EQ.1) WRITE(IOUT,616)
      IF(IFLOW.EQ.1) WRITE(IOUT,617)
      IF(IFLOW.EQ.2) WRITE(IOUT,618)
      WRITE (IOUT,601) NUMNP,NC,NF,NUMEL,NUMAT,NINT
      IF (NINT.EQ.0) WRITE (IOUT,614)
      IF (NINT.GT.0) WRITE (IOUT,615)
C
C     ****************************************************************
C
C     ASSIGN STORAGE LOCATION
C
      L01=1
      L02=L01+2*NUMNP
      L03=L02+2*NUMNP
      L04=L03+2*NUMNP
      L05=L04+NUMEL
      L06=L05+NUMEL
      L07=L06+NUMEL
      L08=L07+NUMEL
      L09=L08+2*NUMAT
      L10=L09+2*NUMAT
      L11=L10+2*NUMEL
      L12=L11+2*NUMEL
      L13=L12+2*NUMNP
      L14=L13+2*NUMNP
      L15=L14+NUMEL
      L16=L15+NUMNP
C
C
C     CALL CENTRAL CONTROL ROUTINE
C
C
C###################################################################
      ITURB = 0
  100 CONTINUE
C
      CALL CENTR (S(L01),S(L02),S(L03),S(L04),S(L05),S(L06),S(L07),
     &    S(L08),S(L09),S(L10),S(L11),S(L12),S(L13),S(L14),S(L15),
     &    S(L16),NSCM,L16)
C
      IF(ITURB.LT.ITURBM) GO TO 100
C###################################################################
C
C
C
  500 FORMAT (20A4)
  501 FORMAT (10I5)
  600 FORMAT ('1',' ',20A4,////)
  601 FORMAT (////,' ','NUMBER OF NODES',17X,I5,
     &            /,' ','NUMBER OF CONSTANT HEAD NODES',3X,I5,
     &            /,' ','NUMBER OF NODES WITH FLUX',7X,I5,
     &            /,' ','NUMBER OF ELEMENTS',14X,I5,
     &            /,' ','NUMBER OF MATERIALS',13X,I5,
     &            /,' ','ORDER OF INTEGRATION',12X,I5,///)
  611 FORMAT (' ','*****  UNCONFINED FLOW IN A HORIZONTAL PLANE *****')
  612 FORMAT (' ','*****  CONFINED FLOW *****')
  613 FORMAT (' ','*****  VERTICAL CROSSECTION *****')
  614 FORMAT (' ','*  4CST QUADRILATERAL ELEMENTS *')
  615 FORMAT (' ','* LINEAR ISOPARAMETRIC QUADRILATERALS  *')
  616 FORMAT (' ','**** AXISYMMETRIC FLOW REGIME ****')
  617 FORMAT (' ','**** FISSURE FLOW PROBLEM ****')
  618 FORMAT (' ','**** NONLINEAR FLOW IN ROCKFILL ****')
      RETURN
      END
C
C
C **********
C
      SUBROUTINE CENTR(X,Y,Z,IND,JND,KND,LND,CON1,
     &                 CON2,ALPHA,Q,XR,HEAD,MND,ICDE,SS,NSCM,L15)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IO/ IN,IOUT
      COMMON/CONST1/ TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),I1(4)
      DIMENSION X(1),Y(1),Z(1),IND(1),JND(1),KND(1),LND(1),CON1(1),
```

```
     &          CON2(1),ALPHA(1),Q(1),XR(1),HEAD(1),MND(1),
     &          SS(NUMNP,1),ICOE(1)
      REWIND 1
      REWIND 2
C
C     ******************************************************************
C
C     READ NODE COORDINATES AND ELEMENT DATA GENERATING THE MISSING VALUE
C
C
C#################################################################
      IF(ITURB.NE.0) GO TO 100
C
      CALL INPUT (X,Y,Z,IND,JND,KND,LND,MND,CON1,CON2,Q,XR,ALPHA,
     &           HEAD,ICOE)
C
  100 CONTINUE
      IF(ITURBM.EQ.0) GO TO 300
C
      CALL NLIN(X,Y,Z,IND,JND,KND,LND,CON1,CON2,XR)
C
      DO 200 I=1,NUMNP
      READ (2) XR(I)
  200 CONTINUE
  300 CONTINUE
C
C#################################################################
C
C DETERMINED BAND WIDTH
C
      IBAN=0
      DO 4 I=1,NUMEL
      IF (IABS(IND(I)-JND(I)).GE.IBAN) IBAN=IABS(IND(I)-JND(I))+1
      IF (IABS(IND(I)-KND(I)).GE.IBAN) IBAN=IABS(IND(I)-KND(I))+1
      IF (IABS(IND(I)-LND(I)).GE.IBAN) IBAN=IABS(IND(I)-LND(I))+1
      IF (IABS(JND(I)-KND(I)).GE.IBAN) IBAN=IABS(JND(I)-KND(I))+1
      IF (IABS(JND(I)-LND(I)).GE.IBAN) IBAN=IABS(JND(I)-LND(I))+1
      IF (IABS(KND(I)-LND(I)).GE.IBAN) IBAN=IABS(KND(I)-LND(I))+1
4     CONTINUE
      L16=L15+2*NUMNP*IBAN
      IF (NSCM-L16) 1,2,2
 1    LOW=L16-NSCM
      WRITE(IOUT,600) LOW
      STOP
 2    CONTINUE
C
C ******************************************************************
C
C GENERATE STIFFNESS MATRICES
C
      DO 9 JJ=1,IBAN
      DO 9 II=1,NUMNP
      SS(II,JJ)=0.0D0
9     CONTINUE
C
C
       CALL ELEM(X,Y,Z,IND,JND,KND,LND,MND,CON1,CON2,ALPHA,Q,XR,HEAD,
     &          SS,NSCM,L14)
C
C ALTER STIFFNESS TO ACCOUNT FOR CONSTANT HEAD VALUES INCLUDING THE
C   ALTERING OF EQUATIONS, OUTPUT EQUATIONS FOR CONSTANT HEAD NODES
C
      DO 20 K=1,NUMNP
      IF (ICOE(K).EQ.0) GO TO 20
      WRITE (1) SS(K,1)
      DO 19 L=2,IBAN
      JL=K+L-1
      IF (JL.GT.NUMNP) GO TO 18
      WRITE (1) SS(K,L)
 18   JK=K-L+1
      IF (JK.LT.1) GO TO 19
      WRITE (1) SS(JK,L)
 19   CONTINUE
 20   CONTINUE
C
C
C
      DO 8 K=1,NUMNP
      IF (ICOE(K).EQ.0) GO TO 8
      XR(K)=HEAD(K)
      SS(K,1)=1.0D0
      DO 7 L=2,IBAN
      JL=K+L-1
      IF (JL.GT.NUMNP) GO TO 6
      XR(JL)=XR(JL)-SS(K,L)*XR(K)
 6    SS(K,L)=0.0D0
      JK=K-L+1
      IF (JK.LT.1) GO TO 7
      XR(JK)=XR(JK)-SS(JK,L)*XR(K)
      SS(JK,L)=0.0D0
 7    CONTINUE
 8    CONTINUE
C
C ******************************************************************
C
C SOLVE EQUATIONS BY GAUSSIAN ELIMINATION
C
C
      CALL SOLVE (SS,XR,IBAN)
C
C
C ******************************************************************
C
C OUTPUT OF SOLUTION
C
      IF (ICHK.GT.0) GO TO 11
      DO 10 II=1,NUMNP
      XR(II)=DSQRT(XR(II))
 10   CONTINUE
 11   IF (ICHK.NE.2) GO TO 12
```

```
      WRITE (IOUT,603)
      DO 14 II=1,NUMNP
      PHEAD=XR(II)-Y(II)
14    WRITE (IOUT,604) II,XR(II),Y(II),PHEAD
C#############################################################
12    IF(ITURB.EQ.1.OR.ITURB.EQ.ITURBM) GO TO 21
      GO TO 13
C#############################################################
21    WRITE (IOUT,601)
      WRITE (IOUT,602) (M,XR(M),M=1,NUMNP)
      IF(ITURBM.EQ.0) GO TO 13
      IF(ITURB.EQ.ITURBM) WRITE(IOUT,608)
      IF(ITURB.EQ.ITURBM) WRITE(IOUT,602) (M,CON1(M),M=1,NUMEL)
      IF(ITURB.EQ.ITURBM) WRITE(IOUT,609)
      IF(ITURB.EQ.ITURBM) WRITE(IOUT,602) (M,CON2(M),M=1,NUMEL)
13    CONTINUE
C
C  COMPUTE THE FLUX AT CONSTANT HEAD NODES
C
      REWIND 1
C############################################################
      IF(ITURB.EQ.1.OR.ITURB.EQ.ITURBM) GO TO 22
      RETURN
C############################################################
22    BALIN=0.D0
      BALOU=0.D0
      WRITE (IOUT,605)
      H2 = 1.0D0
      DO 17 K=1,NUMNP
      IF (ICDE(K).EQ.0) GO TO 17
      READ (1) TEMP
      IF (ICHK.EQ.0) H2 = XR(K)
      FLUX=XR(K)*TEMP*H2
      DO 16 L=2,IBAN
      JL=K+L-1
      IF (JL.GT.NUMNP) GO TO 15
      READ (1) TEMP
      IF (ICHK.EQ.0) H2 = XR(JL)
      FLUX=FLUX+XR(JL)*TEMP*H2
15    JK=K-L+1
      IF (JK.LT.1) GO TO 16
      READ (1) TEMP
      IF (ICHK.EQ.0) H2 = XR(JK)
      FLUX=FLUX+XR(JK)*TEMP*H2
16    CONTINUE
      IF (FLUX.LT.0.D0) BALOU=BALOU+FLUX
      IF (FLUX.GT.0.D0) BALIN=BALIN+FLUX
      WRITE (IOUT,606) K,FLUX
17    CONTINUE
      TBAL=BALOU+BALIN
      WRITE (IOUT,607) BALIN,BALOU,TBAL
600   FORMAT (' ','ARRAY STORAGE TOO SMALL BY ',I5,2X,'WORDS')
601   FORMAT ('1','PROBLEM SOLUTION',/,' ','****************',//,
     &        ' ','NODE NUMBER',8X,'STEADY STATE HEAD',//)
602   FORMAT (' ',5(I5,2X,D12.5,5X),/)
603   FORMAT ('1','PROBLEM SOLUTION - VERTICAL SECTION',/,' ',
     &             '*************************************',//,' ',
     &        'NODE NUMBER',5X,'TOTAL HEAD',8X,'ELEV. HEAD',8X,
     &        'PRESS. HEAD',//)
604   FORMAT (' ',I6,10X,3(D10.3,8X))
605   FORMAT ('1','FLUX AT CONSTANT HEAD NODES',/,' ',
     &             '***************************',//,' ',
     &             'NODE NUMBER',8X,'FLUX',//)
606   FORMAT (' ',I6,10X,D12.5)
607   FORMAT (/,' ','TOTAL FLUX IN=',E10.3,5X,'TOTAL FLUX OUT=',E10.3,
     &        5X,'MASS BALANCE (IN - OUT)=',E10.3)
608   FORMAT ('1',' FINAL ELEMENTAL TRANSMISSIVITIES',/' ',' **********',
     &        '***********************',//,' ','ELEMENT NUMBER',8X,
     &         'TRANSMISSIVITY',//)
609   FORMAT ('1',' FINAL ELEMENTAL VELOCITIES ',/' ',' ***********',
     &        '**************** ',//,' ','ELEMENT NUMBER',8X,
     &        'VELOCITY',//)
      RETURN
      END
C
C
C **********
C
      SUBROUTINE INPUT(X,Y,Z,IND,JND,KND,LND,MND,CON1,CON2,Q,XR,
     &                 ALPHA,HEAD,ICDE)
C
C     THIS SUBROUTINE READS AND GENERATES NODAL AND ELEMENT DATA
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IO/ IN,IOUT
      COMMON/CONST1/ TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),II(4)
      DIMENSION X(1),Y(1),Z(1),IND(1),JND(1),KND(1),LND(1),MND(1),
     &          CON1(1),CON2(1),Q(1),XR(1),ALPHA(1),HEAD(1),ICDE(1)
C
C     ***********************************************************
C
C READ AND PRINT NODAL POINT DATA
C
      L=1
      READ (IN,500) N,INC,X(N),Y(N),Z(N)
1     READ (IN,500) N,INC,X(N),Y(N),Z(N)
      IF(INC.EQ.0) INC = 1
      L=L+INC
      IF(L.GT.N) L = N
C
C CHECK FOR NEED TO INTERNALLY GENERATE NODAL POINT DATA
C
      IF (N-L) 5,4,2
2     LM1=L-INC
      DUM=DFLOAT(N-LM1)
      DUM=DUM/DFLOAT(INC)
      DX=(X(N)-X(LM1))/DUM
      DY=(Y(N)-Y(LM1))/DUM
```

```
      DZ=(Z(N)-Z(LM1))/DUM
      DH=(HEAD(N)-HEAD(LM1))/DUM
3     LM1=L-INC
      X(L)=X(LM1)+DX
      Y(L)=Y(LM1)+DY
      Z(L)=Z(LM1)+DZ
      ICDE(L)=ICDE(LM1)
      HEAD(L)=HEAD(LM1)+DH
      L=L+INC
      IF (N-L) 5,4,3
4     IF (NUMNP-N) 5,6,1
5     WRITE (IOUT,600) N
      STOP
C
C READ IN HEAD BOUNDARY CONDITIONS
C
6     CONTINUE
      DO 20 I=1,NUMNP
20    ICDE(I) = 0
      IF(NC.EQ.0) GO TO 21
      DO 21 I=1,NC
      READ(IN,504) N,DUMMY
      ICDE(N) = 1
      HEAD(N) = DUMMY
21    CONTINUE
C
C READ IN POINT FLUX
C
      DO 30 I=1,NUMNP
30    XR(I)=0.0D0
      IF (NF.EQ.0) GO TO 32
      DO 31 I=1,NF
      READ (IN,503) N,DUMMY
31    XR(N)=DUMMY
32    IF (ILIST.LE.0) GO TO 7
      WRITE (IOUT,601)
      WRITE (IOUT,602) (M,X(M),Y(M),Z(M),XR(M),M=1,NUMNP)
      WRITE (IOUT,606)
      DO 22 I=1,NUMNP
      IF (ICDE(I).EQ.1) WRITE (IOUT,607) I,HEAD(I)
22    CONTINUE
C
C     ADJUST RIGHTHAND SIDE IN CASE OF UNCONFINED FLOW
C
7     IF (ICHK.GT.0) GO TO 24
      DO 23 I=1,NUMNP
      HEAD(I)=HEAD(I)*HEAD(I)
      XR(I)=2.0D0*XR(I)
23    CONTINUE
24    CONTINUE
C
C WRITE TO TAPE 2
C
      DO 120 I=1,NUMNP
      WRITE (2) XR(I)
```

```
  120 CONTINUE
      REWIND 2
C
C*********************************************************************
C
C READ AND PRINT ELEMENT DATA
C
   14 CONTINUE
C
      N=0
8     READ (IN,501) M,IND(M),JND(M),KND(M),LND(M),MND(M),ALPHA(M),Q(M),
     .INC
      IF(INC.EQ.0) INC = 1
9     N=N+1
C
C CHECK FOR NEED TO INTERNALLY GENERATE ELEMENTAL DATA
C
      IF (MND(M).EQ.0) MND(M)=1
      IF (N.EQ.M) GO TO 10
      NM1=N-1
      IND(N)=IND(NM1)+INC
      JND(N)=JND(NM1)+INC
      KND(N)=KND(NM1)+INC
      LND(N)=LND(NM1)+INC
      MND(N)=MND(NM1)
      Q(N)=Q(NM1)
      ALPHA(N)=ALPHA(NM1)
      IF(NUMAT.EQ.NUMEL) MND(N) = MND(NM1) + 1
C
   15 CONTINUE
C
      GO TO 9
10    IF (NUMEL-M) 11,300,8
11    WRITE (IOUT,603) M
      STOP
C
C READ IN MATERIAL PROPERTIES
C
C#################################################################
  300 IF(NUMAT.NE.NUMEL) GO TO 12
      N = 0
   52 READ(IN,502) M,CON1(M),CON2(M)
      IF(CON1(M).EQ.0.0D0) CON1(M)=1.0D0
   51 N = N + 1
      IF(N.EQ.M) GO TO 50
      NM1 = N - 1
      CON1(N) = CON1(NM1)
      CON2(N) = CON2(NM1)
      GO TO 51
   50 IF(NUMAT-M) 100,100,52
12    READ (IN,502) (J,CON1(J),CON2(J),I=1,NUMAT)
  100 CONTINUE
C
C
      IF (ILIST.LE.0) GO TO 13
```

```
C **********
C
      SUBROUTINE ELEM (X,Y,Z,IND,JND,KND,LND,MND,CON1,CON2,ALPHA,Q,XR,
     &                 HEAD,SS,NSCM,L14)
C
C THIS SUBROUTINE GENERATES ELEMENT STIFFNESS MATRICES AND GLOBAL STIF
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/IO/IN,IOUT
      COMMON/CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),I1(4)
      DIMENSION X(1),Y(1),Z(1),IND(1),JND(1),KND(1),LND(1),MND(1),
     &          CON1(1),CON2(1),ALPHA(1),Q(1),XR(1),HEAD(1),
     &          SS(NUMNP,1),SQ(5,5),R(3,3)
      ISTOP=0
      IOLSTP=0
C
C ********************************************************************
C
C DO ONCE FOR EACH ELEMENT
C
      DO 17 LL=1,NUMEL
C
C ROTATE CONDUCTIVITIES TO UNIVERSAL COORDINATE SYSTEM
C
      I=MND(LL)
C###################################################################
      C1 = CON1(I)
      C2 = CON2(I)
      IF(IFLOW.GT.0)  C2 = CON1(I)
      PX=C1*(DCOS(ALPHA(LL))**2.0D0)+C2*DSIN(ALPHA(LL))**2.0D0
      PY=C1*(DSIN(ALPHA(LL))**2.0D0)+C2*DCOS(ALPHA(LL))**2.0D0
      PXY=(C1-C2)*DCOS(ALPHA(LL))*DSIN(ALPHA(LL))
C###################################################################
      I=IND(LL)
      J=JND(LL)
      K=KND(LL)
      L=LND(LL)
C
C     TRANSFORM PLANE ELEMENT TO LOCAL X-Y COORDINATE SYSTEM
C
      CALL TFORM(I,J,K,L,X,Y,Z)
C
C     DETERMINE THE ELEMENT TYPE AND ELEMENT SHAPE
C     IF NINT=0 4CST QUADRILATERAL OR TRIANGLE IS ASSUMED
C
      NODES=4
      IF (NINT.GT.0) GO TO 20
      IF (KND(LL).EQ.LND(LL)) GO TO 2
C
C     GENERATE THE CORRESPONDING STIFFNESS MATRICES
C
C
      CALL QUAD (LL,I,J,K,L,X,Y,PX,PY,PXY,SQ,AREATO,ISTOP)
```

```
      WRITE (IOUT,604)
      WRITE (IOUT,605) (N,IND(N),JND(N),KND(N),LND(N),MND(N),Q(N),
     .                  ALPHA(N),N=1,NUMEL)
      WRITE (IOUT,610)
      WRITE (IOUT,611) (M,CON1(M),CON2(M),M=1,NUMAT)
13    CONTINUE
      IF (ICHK.GT.0) GO TO 26
      DO 25 I=1,NUMEL
      Q(I)=2.0D0*Q(I)
25    CONTINUE
26    CONTINUE
C###################################################################
C-----READ TURBULENT FLOW PARAMETERS ----------
      READ(IN,520) ITURBM,GRAV,AROUGH,RNU
      WRITE(IOUT,620) ITURBM,GRAV,RNU,AROUGH
      ITURB = 0
C------------------------------------------------
  520 FORMAT(I5,2F10.0,1E10.0)
  620 FORMAT('1','     TURBULENT FLOW ANALYSIS    ',/
     .'  --------------------------------------------',/
     .'    MAXIMUM NO. OF ITERATIONS            ',I7,/
     .'    GRAVITATIONAL ACCELERATION       ',D10.5,/
     .'    KINEMATIC VISCOSITY              ',D10.5,/
     .'    FISSURE ABSOLUTE ROUGHNESS       ',D10.5,/)
C------------------------------------------------
C###################################################################
500   FORMAT (2I5,3F10.0)
501   FORMAT (6I5,2F10.2,I5)
502   FORMAT (I5,2F10.0)
503   FORMAT (I5,D10.2)
504   FORMAT (I5,F10.0)
600   FORMAT (' ','ERROR IN NODAL DATA AT NODE',2X,I5,/,' ','CHECK THE
     &         REST OF THE INPUT DATA FOR SIMILAR ERRORS')
601   FORMAT ('1','NODAL POINT DATA',/,' ','****************',//,
     &        ' ','NODE NUMBER',8X,'X COORDINATE',8X,'Y COORDINATE',
     &        8X,'Z COORDINATE',9X,'POINT FLUX',//)
602   FORMAT (' ',2X,I5,7X,F15.3,5X,F15.3,5X,F15.3,5X,D15.3)
603   FORMAT ('1','ERROR IN ELEMENT INPUT AT ELEMENT',I5,/,' ',
     &        'CHECK THE REST OF THE INPUT DATA FOR OTHER ERRORS')
604   FORMAT ('1','ELEMENT DATA',/,' ','************',//,' ','ELEMENT',
     &        ' NUMBER',8X,'NPI',5X,'NPJ',5X,'NPK',5X,'NPL',5X,
     &        'MATERIAL NO.',8X,'FLUX',9X,'CONDUCTIVITY ANGLE',///)
605   FORMAT (' ',3X,I5,12X,I5,3X,I5,3X,I5,3X,I5,7X,I5,10X,D10.3,2X,
     &        F15.1)
606   FORMAT ('1','CONSTANT HEADS',/,' ','**************',//,
     &        ' ','NODE NO.',8X,'HEAD',//)
607   FORMAT (' ',I5,5X,F12.3)
610   FORMAT ('1','MATERIAL PROPERTIES',/,' ','*******************',//,
     &        ' ','MATERIAL NUMBER',8X,'CONDUCTIVITY 1',6X,
     &        'CONDUCTIVITY 2',//)
611   FORMAT (' ',4X,I5,13X,D15.8,5X,D15.8)
      RETURN
      END
C
C
```

```
      IF (ISTOP.EQ.0) GO TO 3
      IOLSTP=1
      ISTOP=0
      WRITE (IOUT,600) LL
      GO TO 17
C
C     TRIANGULAR ELEMENTS
C
 2    NODES=3
      XI=X(I)
      XJ=X(J)
      XK=X(K)
      YI=Y(I)
      YJ=Y(J)
      YK=Y(K)
      KK=0
      CALL TRIAG(LL,KK,PX,PY,PXY,XI,XJ,XK,YI,YJ,YK,AREA2,R,ISTOP)
      IF (ISTOP.EQ.0) GO TO 30
      IOLSTP=1
      ISTOP=0
      WRITE (IOUT,600) LL
      GO TO 17
 30   AREATO=AREA2*.5D0
      DO 18 II=1,3
      DO 18 JJ=1,3
      SQ (II,JJ)=R(II,JJ)
 18   CONTINUE
      DO 4 II=1,4
      SQ(4,II)=0.D0
      SQ(II,4)=0.D0
 4    CONTINUE
      GO TO 3
C
C     ISOPARAMETRIC ELEMENT
C
 20   CALL QUADS (LL,I,J,K,L,PX,PY,PXY,X,Y,XR,Q,SQ,ISTOP)
      IF (ISTOP.EQ.0) GO TO 3
      IOLSTP=1
      ISTOP=0
      WRITE(IOUT,600) LL
      GO TO 17
C
C ADD TO CONDENSED GLOBAL STIFFNESS MATRIX
C
 3    FNODE=DFLOAT(NODES)
      DO 16 II=1,NODES
      GO TO (5,6,7,8),II
 5    N=I
      GO TO 9
 6    N=J
      GO TO 9
 7    N=K
      GO TO 9
 6    N=L
 9    DO 15 JJ=1,NODES
      GO TO (10,11,12,13),JJ
 10   M=I
      GO TO 14
 11   M=J
      GO TO 14
 12   M=K
      GO TO 14
 13   M=L
 14   KL=M-N+1
      IF (KL.LE.0) GO TO 15
      SS(N,KL)=SS(N,KL)+SQ(II,JJ)
 15   CONTINUE
C
C     CONTRIBUTION BY DISRIBUTED FLUX - 4CST AND TRIANGULAR ELTS ONLY
C
      IF (NINT.GT.0) GO TO 16
      XR(N)=XR(N)+(Q(LL)*AREATO)/FNODE
 16   CONTINUE
C
C     RETURN ELEMENT TO GLOBAL COORDINATES X-Y-Z
C
      CALL RFORM(I,J,K,L,X,Y)
C
 17   CONTINUE
C     *STOP IF THERE IS ERROR IN ELEMENT OR NODAL DATA*
      IF (IOLSTP.EQ.1) STOP
      RETURN
 600  FORMAT (' ','ERROR IN ELEMENT NUMBER ',I5)
      END
C
C
C **********
C
      SUBROUTINE QUAD(LL,I,J,K,L,X,Y,PX,PY,PXY,SQ,AREATO,ISTOP)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/IO/IN,IOUT
      COMMON/CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      DIMENSION X(1),Y(1),R(3,3),SQ(5,5)
      DO 7 II=1,5
      DO 7 JJ=1,5
      SQ(II,JJ)=0.0D0
 7    CONTINUE
C
C LOCATE CENTROID
C
      XX=0.0D0
      YY=0.0D0
      XX=(X(I)+X(J)+X(K)+X(L))*.25D0
      YY=(Y(I)+Y(J)+Y(K)+Y(L))*.25D0
      AREATO=0.0D0
C
C DETERMINE ELEMENT STIFFNESS MATRIX FOR QUADRILATERAL ELEMENT
C
      DO 1 KK=1,4
```

```
      GO TO (2,3,4,5),KK
2     XI=X(I)
      XJ=X(J)
      YI=Y(I)
      YJ=Y(J)
      GO TO 6
3     XI=X(J)
      XJ=X(K)
      YI=Y(J)
      YJ=Y(K)
      GO TO 6
4     XI=X(K)
      XJ=X(L)
      YI=Y(K)
      YJ=Y(L)
      GO TO 6
5     XI=X(L)
      XJ=X(I)
      YI=Y(L)
      YJ=Y(I)
6     CONTINUE
C
C
      CALL TRIAG (LL,KK,PX,PY,PXY,XI,XJ,XX,YI,YJ,YY,AREA2,R,ISTOP)
C
C     RETURN IF INPUT DATA ERROR
C
      IF (ISTOP.EQ.1) GO TO 1
C
C GENERATE QUADRILATERAL STIFFNESS MATRIX
C
      KM=KK+1
      IF (KK.EQ.4) KM=1
      SQ(KK,KK)=SQ(KK,KK)+R(1,1)
      SQ(KK,KM)=SQ(KK,KM)+R(1,2)
      SQ(KK,5)=SQ(KK,5)+R(1,3)
      SQ(KM,KK)=SQ(KM,KK)+R(2,1)
      SQ(KM,KM)=SQ(KM,KM)+R(2,2)
      SQ(KM,5)=SQ(KM,5)+R(2,3)
      SQ(5,KK)=SQ(5,KK)+R(3,1)
      SQ(5,KM)=SQ(5,KM)+R(3,2)
      SQ(5,5)=SQ(5,5)+R(3,3)
      AREATO=AREATO+AREA2*.5DO
1     CONTINUE
C
      IF (ISTOP.EQ.1) RETURN
C
C ELIMINATE CENTER POINT
C
      DO 8 II=1,4
      DO 8 JJ=1,4
      FACT=SQ(JJ,5)/SQ(5,5)
      SQ(II,JJ)=SQ(II,JJ)-SQ(II,5)*FACT
8     CONTINUE
C
      RETURN
      END
C
C
C **********
C
      SUBROUTINE TRIAG (LL,KK,PX,PY,PXY,XI,XJ,XK,YI,YJ,YK,AREA2,R,ISTOP)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/IO/IN,IOUT
      COMMON/CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),II(4)
      DIMENSION R(3,3), A(3), B(3)
C
      A(1)=YJ-YK
      A(2)=YK-YI
      A(3)=YI-YJ
      B(1)=XK-XJ
      B(2)=XI-XK
      B(3)=XJ-XI
C
C CALCULATE DETERMINANT - 2*AREA OF TRIANGLE
C
      AREA2=B(3)*A(2)-B(2)*A(3)
C
C     CHECK FOR INPUT DATA ERRORS
C
      IF (AREA2.LE.0.DO)  GO TO 2
C
C
      AK=PX/(2.0DO*AREA2)
      BK=PY/(2.0DO*AREA2)
      CK=PXY/(2.0DO*AREA2)
C
C ELEMENTAL STIFFNESS MATRIX- (R)
C
      DO 1 I=1,3
      DO 1 J=1,3
      R(I,J)=AK*A(I)*A(J)+BK*B(I)*B(J)+CK*(A(I)*B(J)+A(J)*B(I))
1     CONTINUE
      RETURN
2     ISTOP=1
      RETURN
      END
C
C
C **********
C
      SUBROUTINE SOLVE (SS,XR,IBAN)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/IO/IN,IOUT
      COMMON/CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),II(4)
      DIMENSION SS(NUMNP,1),XR(1)
```

```
C
C####################
      RAD = 1.0D0
C####################
      D(1,1)= PX
      D(1,2)= PXY
      D(2,1)=PXY
      D(2,2)=PY
C
C     CALCULATE ELEMENT STIFFNESS
C
      DO 1 JJ=1,5
      DO 1 II=1,5
 1    SQ(II,JJ)=0.0D0
      DO 6 LX=1,NINT
      RI=XG(LX,NINT)
      DO 6 LY=1,NINT
      SI=XG(LY,NINT)
C
C
C     EVALUATE DERIVATIVES (B) AND DETERMINANT DET
C
      CALL STDM (LL,I,J,K,L,RI,SI,X,Y,XR,Q,B,DET,ISTOP,RAD)
C
C     ADD CONTRIBUTIONS TO THE ELEMENT STIFFNESS
C
      IF (ISTOP.EQ.1) RETURN
      WT=WGT(LX,NINT)*WGT(LY,NINT)*DET*RAD
      DO 5 JJ=1,4
      DO 2 KK=1,2
      DB(KK)=0.D0
      DO 2 LK=1,2
      DB(KK)=DB(KK)+D(KK,LK)*B(LK,JJ)
 2    CONTINUE
      DO 4 II=JJ,4
      STIFF=0.0D0
      DO 3 LK=1,2
      STIFF=STIFF+B(LK,II)*DB(LK)
 3    CONTINUE
      SQ(II,JJ)=SQ(II,JJ)+STIFF*WT
 4    CONTINUE
 5    CONTINUE
 6    CONTINUE
      DO 7 JJ=1,4
      DO 7 II=1,4
      SQ(JJ,II)=SQ(II,JJ)
 7    CONTINUE
      RETURN
      END
C
      SUBROUTINE STDM(LL,I,J,K,L,RI,SI,X,Y,XR,Q,B,DET,ISTOP,RAD)
C
C     THIS SUBROUTINE EVALUATES THE DISPLACEMENT MATRIX B AT A POINT
C     (R,S) FOR A QUADRILATERAL ELEMENT
C
```

```
C
C REDUCTION
C
      DO 3 N=1,NUMNP
      DO 2 M=2,IBAN
      IF (SS(N,M).EQ.0.0D0) GO TO 2
      C=SS(N,M)/SS(N,1)
      I=N+M-1
      IF (I.GT.NUMNP) GO TO 2
      J=0
      DO 1 K=M,IBAN
      J=J+1
 1    IF (SS(N,K).NE.0.0D0) SS(I,J)=SS(I,J)-C*SS(N,K)
      SS(N,M)=C
      XR(I)=XR(I)-SS(N,M)*XR(N)
 2    CONTINUE
      XR(N)=XR(N)/SS(N,1)
 3    CONTINUE
C
C BACK SUBSTITUTION
C
      N=NUMNP
 4    DO 5 K=2,IBAN
      L=N+K-1
      IF (L.GT.NUMNP) GO TO 6
 5    IF (SS(N,K).NE.0.0D0) XR(N)=XR(N)-SS(N,K)*XR(L)
 6    N=N-1
      IF (N.GT.0) GO TO 4
      RETURN
      END
      SUBROUTINE QUADS (LL,I,J,K,L,PX,PY,PXY,X,Y,XR,Q,SQ,ISTOP)
C
C     THIS SUBROUTINE CREATES THE ISOPARAMETRIC ELEMENT STIFFNESS MATRIX
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /IO/IN,IOUT
      COMMON /CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),II(4)
      DIMENSION X(1),Y(1),XR(1),Q(1),SQ(5,5),D(2,2),B(2,4),XG(4,4),
     &          WGT(4,4),DB(4)
C
C     MATRIX XG CONTAINS THE GAUSS-LEGENDRE INTERPOLATION POINTS
C
      DATA XG/0.0D0,0.0D0,0.0D0,0.0D0,-.577350269189600,
     &       .577350269189600, 0.0D0, 0.0D0,-.774596669241500,
     &       0.0D0, .774596669241500, 0.0D0, -.861136311594150D0,
     &       -.339981043584900, .339981043584900,.861136311594150D0/
C
C     MATRIX WGT STORES INTEGRATION WEIGHTING FACTORS
C
      DATA WGT/ 2.D0, 0.D0, 0.D0, 0.D0, 1.D0, 1.D0, 0.D0, 0.D0,
     &      .555555555555556D0, .888888888888889D0, .555555555555556D0, 0.D0,
     &      .347854845137500, .652145154862500, .652145154862500,
     &      .347854845137500/
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /IO/IN,IOUT
      COMMON /CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,NINT,ICHK,ILIST
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),II(4)
      DIMENSION X(1),Y(1),XR(1),Q(1),B(2,4),XJ(2,2),XJI(2,2),P(2,4),H(4)
      RP=1.0D0+RI
      SP=1.0D0+SI
      RM=1.0D0-RI
      SM=1.0D0-SI
C
C     INTERPOLATION FUNCTIONS
C
      H(1)=0.25D0*RP*SP
      H(2)=0.25D0*RM*SP
      H(3)=0.25D0*RM*SM
      H(4)=0.25D0*RP*SM
C
C     NATURAL COORDINATE DERIVATIVES OF THE INTERPOLATION FUNCTION
C
C     1. WITH RESPECT TO R
C
      P(1,1)=0.25D0*SP
      P(1,2)=-P(1,1)
      P(1,3)=-.25D0*SM
      P(1,4)=-P(1,3)
C
C     2. WITH RESPECT TO S
C
      P(2,1)=.25D0*RP
      P(2,2)=0.25D0*RM
      P(2,3)=-P(2,2)
      P(2,4)=-P(2,1)
C
C     EVALUATE THE JACOBIAN AT POINT (R,S)
C
      DO 2 II=1,2
      XJ(II,1)=P(II,1)*X(I)+P(II,2)*X(J)+P(II,3)*X(K)+P(II,4)*X(L)
      XJ(II,2)=P(II,1)*Y(I)+P(II,2)*Y(J)+P(II,3)*Y(K)+P(II,4)*Y(L)
    2 CONTINUE
C
C     COMPUTE THE DETERMINANT OF THE JACOBIAN AT  (R,S)
C
      DET=XJ(1,1)*XJ(2,2)-XJ(2,1)*XJ(1,2)
      IF (DET.GT.0.00000001) GO TO 3
      WRITE (IOUT,600) LL
      ISTOP=1
      RETURN
C
C     COMPUTE THE INVERSE OF THE JACOBIAN
C
    3 DUM=1.0D0/DET
      XJI(1,1)=XJ(2,2)*DUM
      XJI(1,2)=XJ(1,2)*DUM
      XJI(2,1)=XJ(2,1)*DUM
      XJI(2,2)=XJ(2,2)*DUM
C
C     EVALUATE GLOBAL DERIVATIVE OPERATOR B
C
      DO 4 KK=1,4
      B(1,KK)=0.0D0
      B(2,KK)=0.0D0
      DO 4 II=1,2
      B(1,KK)=XJI(1,II)*P(II,KK)+B(1,KK)
      B(2,KK)=XJI(2,II)*P(II,KK)+B(2,KK)
    4 CONTINUE
C
C     ADD DISTRIBUTED FLUX TERMS
C
      DO 10 II=1,4
      GO TO (5,6,7,8), II
    5 N=I
      GO TO 9
    6 N=J
      GO TO 9
    7 N=K
      GO TO 9
    8 N=L
    9 XR(N)=XR(N)+Q(LL)*H(II)*DET
   10 CONTINUE
C#######################################################
C     COMPUTE RADIUS AT POINT (R,S)
      IF(IRAD.NE.1)RETURN
      RAD = H(1)*X(I) + H(2)*X(J) + H(3)*X(K) + H(4)*X(L)
C#######################################################
  600 FORMAT (' ','***** ERROR- ZERO DETERMINANT IN ELT.',I6)
      RETURN
      END
C##############################################################
C-------------------------------------------------
      SUBROUTINE NLIN(X,Y,Z,II,JJ,KK,LL,CON1,CON2,PHI)
C-------------------------------------------------------
C-----GENERALISED SUBROUTINE FOR NON LAMINAR FLUID FLOW
C     IN A HORIZONTAL FISSURE - FINITE ELEMENT VERSION
C     FOR QUADRILATERAL AND TRIANGULAR FINITE ELEMENTS
C-------------------------------------------------------
C     DEREK ELSWORTH - JULY 1983
C-------------------------------------------------------
C     NUMEL   - MAXIMUM NUMBER OF ELEMENTS
C     X,Y,Z   - NODAL COORDINATES
C     II,JJ,KK,LL  - LOCAL NODAL NUMBERING (COUNTERCLOCKWISE)
C     PHI      - NODAL POTENTIAL
C     CON1     - ELEMENT HYDRAULIC CONDUCTIVITY
C     CON2     - ELEMENT FISSURE APERTURE
C-------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/IO/IN,IOUT
      COMMON/CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),II(4)
```

```
      DIMENSION X(1),Y(1),Z(1),II(1),JJ(1),KK(1),LL(1),PHI(1),CON1(1),
     .          CON2(1)
C------------------------------------------------
      ITURB = ITURB + 1
      G = GRAV
      DKM = 0.0D0
      DO 200 I=1,NUMEL
      CI = CON1(I)
      NI = II(I)
      NJ = JJ(I)
      NK = KK(I)
      NL = LL(I)
C------------------------------------------------
      CALL TFORM(NI,NJ,NK,NL,X,Y,Z)
C------------------------------------------------
      IF(NL.EQ.NK) GO TO 100
C-----EVALUATE GRADIENTS OVER QUADRILATERAL ELEMENT
      XIJ = X(NJ)-X(NI) + X(NK)-X(NL)
      XJK = X(NK)-X(NJ) + X(NL)-X(NI)
      YIJ = Y(NJ)-Y(NI) + Y(NK)-Y(NL)
      YJK = Y(NK)-Y(NJ) + Y(NL)-Y(NI)
      DX = XIJ*XIJ + YIJ*YIJ
      DX = DSQRT(DX)
      DY = XJK*XJK + YJK*YJK
      DY = DSQRT(DY)
      GX = (PHI(NJ)-PHI(NI)+PHI(NK)-PHI(NL))/DX
      GY = (PHI(NK)-PHI(NJ)+PHI(NL)-PHI(NI))/DY
      GO TO 150
C-----EVALUATE GRADIENTS OVER TRIANGULAR ELEMENT
  100 DT = X(NI)*(Y(NJ)-Y(NK))-Y(NI)*(X(NJ)-X(NK))+
     .     (X(NJ)*Y(NK)-X(NK)*Y(NJ))
      GX = PHI(NI)*(Y(NJ)-Y(NK))-Y(NI)*(PHI(NJ)-PHI(NK))+
     .     (PHI(NJ)*Y(NK)-PHI(NK)*Y(NJ))
      GY =-PHI(NI)*(X(NJ)-X(NK))+X(NI)*(PHI(NJ)-PHI(NK))+
     .     (PHI(NK)*X(NJ)-PHI(NJ)*X(NK))
      GX = GX/DT
      GY = GY/DT
C------------------------------------------------
  150 GXY = (GX*GX+GY*GY)**0.5D0
C------------------------------------------------
C-----EVALUATE CONDUCTIVITY COEFFICIENTS -----
      IF(IFLOW.NE.2) GO TO 180
C-----ROCKFILL CONDUCTIVITY OPTION  ----------
      B = CON2(I)
      E = -0.44
      Z1 = 1.0
      GO TO 800
  180 CONTINUE
C-----FISSURE FLOW OPTION -------------------
C-----APERTURE DEPENDENT TERM            -----
      Z1 = CON2(I)
      ROUGH = AROUGH/(2.0D0*Z1)
      A1 = (2.0D0*CON1(I)*GXY)/RNU
      A2 = 1.0D05/(0.333D0+ROUGH/0.15D-02)
      A3 = 0.19D01/ROUGH
      A4 = 0.384D03*(0.1D01 + .88D01*ROUGH**0.15D01)*(DLOG10(A3))**2.D0
      IF(ROUGH.LE.0.33D-01) GO TO 300
      GO TO 600
C-----ZONE 1 -- LAMINAR -----------------------
  300 CONTINUE
      IF(ITURB.EQ.1) GO TO 350
      IF(A1.GT.0.23D04) GO TO 400
  350 B = G*Z1*Z1/(0.12D02*RNU)
      E = 0.0D0
      GO TO 800
C-----ZONE 2 -- TURBULENT ---------------------
  400 CONTINUE
      IF(A1.GT.A2) GO TO 500
      B = (((G/.79D-01)*(2.0D0/RNU)**0.25D0*Z1*Z1*Z1)**0.5710D0)/Z1
      E = -0.429D0
      GO TO 800
C-----ZONE 3 -- COMPLETELY ROUGH --------------
  500 CONTINUE
      B1 = .37D01/ROUGH
      B = DSQRT(G)*0.4D01*DLOG10(B1)*DSQRT(Z1)
      E = -0.5D0
      GO TO 800
C-----ZONE 4 -- LAMINAR ----------------------
  600 CONTINUE
      IF(ITURB.EQ.1) GO TO 650
      IF(A1.GT.A4) GO TO 700
  650 B = (G*Z1*Z1)/(12.0D0*RNU*(1.0D0 + 8.8D0*ROUGH**1.5D0))
      E = 0.0D0
      GO TO 800
C-----ZONE 5 -- TURBULENT --------------------
  700 CONTINUE
      B1 = 1.9D0/ROUGH
      B2 = Z1*G
      B = 4.0D0*DSQRT(B2)*DLOG10(B1)
      E = -0.5D0
  800 CONTINUE
C-----EVALUATE GRADIENT DEPENDENT TERM  --------
      IF(GXY.EQ.0.0D0) GO TO 830
      IF(E.EQ.0.0D0) GO TO 830
      GDME = GXY**E
      GO TO 840
  830 GDME = 1.0D0
  840 CONTINUE
      CON1(I) = B*GDME*Z1
      IF(ITURB.EQ.ITURBM) CON2(I) = CON1(I)*GXY/Z1
C------------------------------------------
      DK = DABS(CON1(I)-CI)
      IF(DK.GT.DKM) DKM = DK
C------------------------------------------
      CALL RFORM(NI,NJ,NK,NL,X,Y)
C------------------------------------------
  200 CONTINUE
      WRITE(IOUT,2000) ITURB,DKM
 2000 FORMAT(/,' AT ITERATION NUMBER ',I2,'  DELTA T =',D12.5,//)
      IF(DKM.LE.1.0D-05) ITURB = ITURBM
```

```
      END
C
C##################################################################
C
      SUBROUTINE RFORM(I,J,K,L,X,Y)
C-----RETURN ORIGINAL COORDINATES TO ELEMENT -------
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/IO/IN,IOUT
      COMMON/CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),I1(4)
      DIMENSION X(1),Y(1)
C
      X(I) = XT(1)
      X(J) = XT(2)
      X(K) = XT(3)
      X(L) = XT(4)
      Y(I) = YT(1)
      Y(J) = YT(2)
      Y(K) = YT(3)
      Y(L) = YT(4)
      RETURN
      END
```

```
      RETURN
      END
C##################################################################
C
      SUBROUTINE TFORM(I,J,K,L,X,Y,Z)
C-----TO TRANSFORM ELEMENT FROM GLOBAL TO LOCAL COORDINATES
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/IO/IN,IOUT
      COMMON/CONST1/TITLE(20),NUMNP,NUMEL,NC,NF,NUMAT,ICHK,ILIST,NINT
      COMMON/TURB/ITURB,ITURBM,GRAV,RNU,AROUGH,IRAD,IFLOW
      COMMON/FRAC/XT(4),YT(4),I1(4)
      DIMENSION X(1),Y(1),Z(1)
C
      I1(3) = K
      I1(4) = L
C-----EVALUATE LOCAL COORDINATES -------
      XIJ = X(J) - X(I)
      YIJ = Y(J) - Y(I)
      ZIJ = Z(J) - Z(I)
      DIJ = DSQRT(XIJ*XIJ + YIJ*YIJ + ZIJ*ZIJ)
      DCOS1 = XIJ/DIJ
      DCOS2 = YIJ/DIJ
      DCOS3 = ZIJ/DIJ
C-----EXCHANGE STORAGE LOCATIONS -----
      XT(1) = X(I)
      XT(2) = X(J)
      XT(3) = X(K)
      XT(4) = X(L)
      YT(1) = Y(I)
      YT(2) = Y(J)
      YT(3) = Y(K)
      YT(4) = Y(L)
C
      X(I) = 0.0D0
      Y(I) = 0.0D0
      X(J) = DIJ
      Y(J) = 0.0D0
C
      NN = 4
      IF(K.EQ.L) NN = 3
C
      DO 100 II=3,NN
      IT = I1(II)
      XIJ = X(IT) - XT(1)
      YIJ = Y(IT) - YT(1)
      ZIJ = Z(IT) - Z(I)
      DIJ = DSQRT(XIJ*XIJ + YIJ*YIJ + ZIJ*ZIJ)
      DCOS11 = XIJ/DIJ
      DCOS22 = YIJ/DIJ
      DCOS33 = ZIJ/DIJ
      COSA = DCOS1*DCOS11 + DCOS2*DCOS22 + DCOS3*DCOS33
      X(IT) = DIJ*COSA
      Y(IT) = DSQRT(DIJ*DIJ - X(IT)*X(IT))
  100 CONTINUE
      RETURN
```

APPENDIX 4 - LISTING OF DATA FOR AXISYMMETRIC FISSURE FLOW EXAMPLE

```
AXISYMMETRIC TEST EXAMPLE - RE/RI=100
 202  100    4        100    1    0    0          1
   1    2         1.     -.3140
 201    2      101.0  -31.7140
   2    2         1.      .3140
 202    2      101.0   31.7140
   1
   2
 201       1.00
 202       1.00
   1    1    3    4    2    1                           2
 100  199  201  202  200  100                           2
   1         1.      0.0060
 100                 0.0060
  10        32.        .006   .14E-06
```

APPENDIX 5 - LISTING OF DATA FOR AXISYMMETRIC ROCKFILL FLOW EXAMPLE

```
AXISYMMETRIC TEST EXAMPLE - RE/RI=100  ROCKFILL FLOW, CONFINED.
  202  100    4        100    1     0     0         2
    1    2          1.     -.3140
  201    2       101.0  -31.7140
    2    2          1.      .3140
  202    2       101.0   31.7140
    1
    2
  201       1.C0
  202       1.00
    1    1    3    4    .2     1                          2
  100  199  201  202  200   100                          2
    1         1.          2.0
  100                     2.0
   10        32.        .006    .14E-06
```

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720