
Fluid Flow and Solute Transport Modeling Through Three-Dimensional Networks of Variably Saturated Discrete Fractures

Prepared by T. C. Rasmussen, D. D. Evans

Department of Hydrology and Water Resources
University of Arizona

Prepared for
U.S. Nuclear Regulatory
Commission

HYDROLOGY DOCUMENT NUMBER 546

89020804917

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability of responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

NOTICE

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 1717 H Street, N.W.
Washington, DC 20555
2. The Superintendent of Documents, U.S. Government Printing Office, Post Office Box 37082,
Washington, DC 20013-7082
3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances*.

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free, to the extent of supply, upon written request to the Division of Information Support Services, Distribution Section, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

Fluid Flow and Solute Transport Modeling Through Three-Dimensional Networks of Variably Saturated Discrete Fractures

Manuscript Completed: November 1988
Date Published: January 1989

Prepared by
T. C. Rasmussen, D. D. Evans

T. J. Nicholson, NRC Project Manager

Department of Hydrology and Water Resources
University of Arizona
Tucson, AZ 85721

Prepared for
Division of Engineering
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555
NRC FIN D1662

OTHER REPORTS IN THIS SERIES ARE:

Evans, D.D., 1983, Unsaturated Flow and Transport Through Fractured Rock - Related to High-Level Waste Repositories, NUREG/CR-3206, 231 pp.

Schrauf, T.W. and D.D. Evans, 1984, Relationship Between the Gas Conductivity and Geometry of a Natural Fracture, NUREG/CR-3680, 131 pp.

Huang, C. and D.D. Evans, 1985, A 3-Dimensional Computer Model to Simulate Fluid Flow and Contaminant Transport Through a Rock Fracture System, NUREG/CR-4042, 109 pp.

Green, R.T. and D.D. Evans, 1987, Radionuclide Transport as Vapor Through Unsaturated Fractured Rock, NUREG-CR-4654, 163 pp.

Rasmussen, T.C. and D.D. Evans, 1987, Unsaturated Flow and Transport Through Fractured Rock - Related to High-Level Waste Repositories, NUREG/CR-4655, 474 pp.

Yeh, T.C.J., T.C. Rasmussen and D.D. Evans, 1988, Simulation of Liquid and Vapor Movement in Unsaturated Fractured Rock at the Apache Leap Tuff Site: Models and Strategies, NUREG/CR-5097, 73 pp.

Weber, D.S. and D.D. Evans, 1988, Stable Isotopes of Authigenic Minerals in Variably-Saturated Fractured Tuff, NUREG/CR-5255, 70 pp.

ABSTRACT

The boundary integral method is used to estimate hydraulic and solute transport properties of unsaturated, fractured rock by solving the boundary value problem within intersecting fracture planes. Flow through both impermeable and permeable rock is determined using two and three dimensional formulations, respectively. Synthetic fracture networks are created to perform sensitivity studies, results of which show that: (1) The global hydraulic conductivity is linearly dependent on the product of fracture transmissivity and density for fractures of infinite length; (2) The effect of correlation between fracture length and transmissivity is to increase the global hydraulic conductivity; and (3) Simulated flow through a fractured permeable matrix compare favorably with analytic results.

Flow through variably saturated fractures is modeled using a constant capillary head within individual fractures. A simulated free surface compares favorably with an approximate analytic solution and with laboratory results. Simulations indicate zones of water under both positive and negative pressure, as well as regions of air-filled voids. Travel times and breakthrough curves are determined by integrating the inverse velocity over a streamline, and then summing over all streamlines. Faster travel times are noted as fracture saturation decreases for the fracture network examined.

TABLE OF CONTENTS

	<u>Page</u>
EXECUTIVE SUMMARY	1
1. INTRODUCTION	4
1.1 Formulation of Flow Through Fractured Rock	4
1.1.1 Intra-Fracture Flow and Solute Transport	6
1.1.2 Inter-Fracture Flow and Solute Transport	7
1.1.3 Supra-Fracture Flow and Solute Transport	10
1.2 Proposed DFN Models	12
1.2.1 Steady Saturated Flow Through Discrete Fracture Networks	13
1.2.2 Steady Flow Through Variably Saturated Fractures	13
1.2.3 Solute Transport Through Variably Saturated Fractures	14
1.2.4 Computer Simulation Models of Fracture Flow and Transport	14
2. NUMERICAL SIMULATION OF STEADY FLUID FLOW	16
2.1 Fracture Flow Hydraulics	16
2.2 Boundary Integral Method	21
2.3 Applications to Steady Flow Through Discrete Fractures	29
2.3.1 Flow Through a Single Fracture	29
2.3.2 Flow Through Serial Fractures	35
2.3.3 Flow Through Fractures with Internal Intersections	42
2.3.4 Influence of Fracture Density and Spacing	44
2.3.5 Influence of Fracture Transmissivity and Length Correlations	46
2.4 Coupled Fracture-Matrix Flow	48
2.4.1 Two Dimensional Porous Medium Application	49
2.4.2 Three Dimensional Porous Medium Application	54
2.4.3 Summary of Three Dimensional Coupled Fracture-Matrix Flow	58

TABLE OF CONTENTS (Continued)

	<u>Page</u>
3. NUMERICAL SIMULATION OF STEADY FLOW THROUGH VARIABLY SATURATED FRACTURES	59
3.1 Generation of Synthetic Moisture Characteristic and Unsaturated Transmissivity Functions for Discrete Fractures	59
3.1.1 Theory of Flow Through Capillaries	61
3.1.2 Fracture Moisture Characteristic Function	65
3.1.3 Fracture Unsaturated Transmissivity Function	67
3.2 Boundary Integral Solution of Free Surface Problem	68
3.2.1 Comparison of Boundary Integral Results with Analytic and Laboratory Results	70
3.2.2 Application to Fracture Networks	79
4. SOLUTE TRANSPORT THROUGH UNSATURATED FRACTURED ROCK	86
4.1 Travel Time and Breakthrough Curve Calculation	86
4.1.1 Travel Time and Breakthrough Curve Calculation Using Boundary Integral Method	91
4.1.2 Specification of Potential and Stream Intersections	92
4.1.3 Application to Saturated and Variably Saturated Flow	93
4.2 Effects of Sorption and Solute Retardation	99
4.3 Effects of Matrix Diffusion	103
5. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	115
5.1 Summary	115
5.2 Conclusions	116
5.3 Recommendations	117

TABLE OF CONTENTS (Continued)

	<u>Page</u>
APPENDIX A: COMPUTER SIMULATION MODELS	118
A.1 BIM: Fluid Flow Solver Using the Boundary Integral Method	119
A.2 FRACGEN: Discrete Fracture Network Generator	135
A.2.1 Creation of Individual Fractures	135
A.2.2 Definition of Global and Sample Volumes, and Interior Surfaces	136
A.2.3 Number of Fracture Sets and Fracture Density	138
A.2.4 Fracture Location	138
A.2.5 Fracture Orientation	138
A.2.6 Fracture Areal Extent	138
A.2.7 Fracture Transmissivity	140
A.2.8 Determination of Fracture Intersections	140
A.2.9 Truncation of Fractures	142
A.2.10 Removal of Isolated and Dead-End Fractures	142
A.2.11 Examples of Program Inputs and Outputs	142
A.3 BIM2D: Boundary Integral Method for Two Dimensional Fracture Flow	162
A.4 BIM3D: Boundary Integral Method for Three Dimensional Coupled Fracture-Matrix Flow	170
GLOSSARY	180
LIST OF REFERENCES	183

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Page</u>
1.1 Nesting of fluid flow and solute transport processes in fractured rock.	5
1.2 Site and bond percolation models for three densities of site and bond occupancy, p . Note that networks remain fully connected until the occupancy drops below a critical threshold density, p'	8
2.1 Flow domain and boundary conditions for two dimensional (A) and three dimensional (B) porous media. Symbols are defined in text.	17
2.2 Fluxes, q_f and q_m , and geometric properties, w_f and a_m , for a single fracture and rock face intersecting a boundary.	19
2.3 Measured relationship between hydraulic and mass balance fracture aperture	21
2.4 Boundary discretization schemes for two dimensional planar fractures (A) and three dimensional rock matrix (B).	24
2.5 Linear interpolation functions for one (A) and two (B) dimensional boundaries.	26
2.6 Multiple flow domain geometry.	27
2.7 Discretization schemes for boundary integral method simulation study.	29
2.8 Serial fracture discretization scheme. Coarse discretization shown.	35
2.9 Four experiments conducted to evaluate the effect of fracture lengths, orientations and distances of boundaries. Views are plan showing exterior no flow boundaries and lines of intersection with other fractures for two parallel fractures (A), two perpendicular fractures (B), two parallel fractures of disparate lengths (C), and three parallel fractures (D).	43
2.10 Four realizations of fracture networks used to evaluate network hydraulic conductivity.	47
2.11 Fracture surface discretization geometry showing net flux term, q_m , representing flow between the fracture and the matrix.	49

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>	<u>Page</u>
2.12 Two dimensional flow geometry showing no flow and constant head boundaries for fractures perpendicular (A), diagonal (B) and parallel (C) to the direction of the mean head gradient.	51
2.13 Definition of aspect ratio as proportion of distance between nodes to width of fracture.	51
2.14 Results of simulations performed using the two dimensional flow geometry for fractures fully (A) and partially (B) dividing the flow domain.	53
2.15 Three dimensional flow geometry showing fracture (A) and matrix (B) boundary surface discretization strategies.	55
2.16 Results of simulations performed using the three dimensional flow geometry for fractures fully (A) and partially (B) dividing the flow domain.	57
3.1 Macroscopic and microscopic formulations of unsaturated flow through fractured rock.	60
3.2 Surface tension forces acting upon a gas-liquid-solid interface.	61
3.3 Geometric properties of an inclined fracture in contact with a liquid surface at its base.	66
3.4 Moisture characteristic curves for planar fractures with constant capillary head at various orientations. . . .	66
3.5 Conceptual model of zone of saturation within an unsaturated fracture.	67
3.6 Procedure for locating nodal positions: (1) Determine which nodes satisfy $h_p < h_c$, solid circles; (2) For these nodes, locate new temporary boundary nodes on the same isohead contour and the contour of $h_p = h_c$, open circles; (3) Update nodal position by finding midpoint between initial and temporary nodal positions, crosses. . .	68
3.7 Circular constant head source in a planar fracture.	70
3.8 Graphical representation of superimposed flow fields resulting from a circular source and a gravitational field	73
3.9 Laboratory flow visualization experiment.	74

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>	<u>Page</u>
3.10 Flow visualization experiment for two input pressure head boundary conditions.	76
3.11 Unsaturated flow domain geometry for circular source.	77
3.12 Simulated free surface position for three input pressure head boundary conditions.	78
3.13 Flow geometry and boundary conditions for fracture intersecting two other fractures.	80
3.14 Contours of total head within the plane of a horizontal fracture.	81
3.15 Contours of total head and zone where pressure heads are negative within the plane of a vertical fracture allowing no air entry.	83
3.16 Contours of total head and free surface position after successive iterations and after the final iteration within the plane of a vertical fracture allowing air entry.	84
3.17 Contours of total head and interface position after the final iteration within the plane of a vertical fracture allowing air entry. A capillary head of 10 m was used.	85
4.1 Finite element and boundary integral approximations of fluid streamlines. Note sharp change in direction at edge of element in finite element approximation.	87
4.2 Streamline showing two endpoints and velocity at a point along the streamline.	88
4.3 Translation in time of solute concentration curve ignoring molecular diffusion.	89
4.4 Geometry of flow between two boundaries. Also indicated are limiting streamlines.	90
4.5 Boundary conditions for stream functions.	91
4.6 Streamline and head contour discretization scheme.	92
4.7 Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a square flow domain (A) and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.	94

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>	<u>Page</u>
4.8 Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a horizontal fracture with two intersecting fractures (A) and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.	96
4.9 Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a vertical fracture with two intersecting fractures (A) and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.	97
4.10 Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a vertical fracture with two intersecting fractures and a capillary head of 10 m (A) and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.	98
4.11 Calculated breakthrough curves at the outflow boundary for the flow domain of Figure 4.8 with a retardation coefficient of 2.	101
4.12 Calculated breakthrough curves at the outflow boundary for the flow domain of Figure 4.9 with a retardation coefficient of 2.	101
4.13 Calculated breakthrough curves at the outflow boundary for the flow domain of Figure 4.10 with a retardation coefficient of 2.	102
4.14 Conceptual model of flow through an individual fracture of semi-infinite areal extent.	103
4.15 Analytic solutions of distance and time plots of solute concentration resulting from a step inflow of solute past an impermeable rock matrix.	106
4.16 Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with an effective porosity of 0.001.	107
4.17 Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of time since injection of a step inflow of solute past a rock matrix with an effective porosity of 0.001.	108

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>	<u>Page</u>
4.18 Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with an effective porosity of 0.01.	109
4.19 Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of time since injection of a step inflow of solute past a rock matrix with an effective porosity of 0.01.	110
4.20 Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with an effective porosity of 0.1.	111
4.21 Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of time since injection of a step inflow of solute past a rock matrix with an effective porosity of 0.1.	112
4.22 Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with an effective porosity of 0.1 and an increased velocity from 10 to 100 m/yr.	113
4.23 Relative concentration at outflow surface of flow domain shown in Figure 4.7 in response to a step injection of tracer at the inflow boundary for three values of the parameter $\theta(D^*)^{1/2}$ at two velocities, 0.08 m/s (A) and 0.16 m/s (B).	114
A.1 Relative positions of global, sample and interior volumes.	137
A.2 Position and orientation parameters for discrete fractures.	139
A.3 Four possible outcomes of non-parallel fractures: Fracture planes intersect but neither discrete fracture intersects the line of intersection (A); Only one fracture intersects the line of intersection (B); Both fractures intersect the line of intersection, but not each other (C); and Both fractures intersect line of intersection and each other.	141

LIST OF TABLES

<u>Table</u>	<u>Page</u>
2.1 Results for simulation experiment using coarse discretization interval, twelve nodes total. Fracture transmissivity is 1. Bold faced values are assigned boundary conditions.	31
2.2 Results for simulation experiment using coarse discretization interval, twelve nodes total. Fracture transmissivity is 5. Bold faced values are assigned boundary conditions.	32
2.3 Results for simulation experiment using fine discretization interval, twenty four nodes total. Fracture transmissivity is 1. Bold faced values are assigned boundary conditions.	33
2.4 Results for simulation experiment using fine discretization interval, twenty four nodes total. Fracture transmissivity is 5. Bold faced values are assigned boundary conditions.	34
2.5 Results for simulation experiment using coarse discretization interval, twelve nodes total. Transmissivity of both fractures is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.	36
2.6: Results for simulation experiment using coarse discretization scheme, twelve nodes total. Transmissivity of first fracture is 5; of second is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.	37
2.7 Results for simulation experiment using fine discretization interval, twenty four nodes total. Transmissivity of both fractures is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.	38
2.8 Results for simulation experiment using fine discretization interval, twenty four nodes total. Transmissivity of first fracture is 5; of second is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.	40
2.9 Calculated and exact flow rates for variable geometries within the plane of a fracture.	44

LIST OF TABLES (Continued)

<u>Table</u>	<u>Page</u>
2.10 Effect of fracture density and fracture transmissivity on global hydraulic conductivity. A unit vertical gradient was applied across two ends of a (50 x 50 x 50) m ³ rock cube. Fracture density is length of fractures (m) per (50 x 50) m ² rock cube face.	45
2.11 Comparison of global hydraulic conductivity calculated using the boundary integral model for uncorrelated and perfectly correlated lengths and transmissivities.	47
2.12 Fracture parameters for flow experiments.	50
2.13 Two dimensional precision of boundary integral method as a function of aspect ratio, defined here as the ratio of the distance between nodes to the width of the flow domain.	50
2.14 Three dimensional precision of boundary integral method as a function of aspect ratio, defined here as the ratio of the distance between nodes to the width of the flow domain.	56
2.15 Analytic and simulation results for variable fracture-matrix permeability ratios using three dimensional boundary integral method. Fully bisecting fractures.	56
3.1 Results of flow visualization experiments.	75
3.2 Results of computer simulation experiments. Flow geometry is presented as Figure 3.11. A unit fracture transmissivity is used.	79
3.3 Simulation results for (1) horizontal flow, (2) vertical flow with no capillary head, and (3) vertical flow with capillary head.	82
4.1 Calculated breakthrough times for streamlines in horizontal and vertical fractures.	95
A.1 Opening menu for program BIM	119
A.2 Sample input data for program BIM.	120
A.3 Sample data output from Option 3 from program BIM using input data from Table A.2.	121
A.4 Fracture Network Parameters: Specified for interior volumes m greater than 1, fractures one through j and boundary surfaces 1 through k	143

LIST OF TABLES (Continued)

<u>Table</u>	<u>Page</u>
A.5 Fracture Network Parameters: Specified for interior volumes m greater than 1, generated for fractures one through j and specified for boundary surfaces 1 through k	143
A.6 Sample program input.	144
A.7 Sample program output for data set presented in Table A.6.	145
A.8 Sample Input Data for Program BIM3D.	170

ACKNOWLEDGMENTS

Many individuals have provided assistance in the completion of this document, especially discussions with Professors T.C. Jim Yeh and Shlomo Neuman. Drafting of figures within this report was performed by Priscilla J. Sheets. Funding for this report was provided by the U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Division of Engineering under contract NRC-04-86-114, FIN D1662. The NRC project technical monitor was Mr. Thomas J. Nicholson, who provided us with direction and encouragement during the development of this document.

EXECUTIVE SUMMARY

Introduction

A computer model based on the boundary integral method is formulated to investigate hydraulic and solute transport properties of unsaturated, fractured rock. The model is applied to networks of discrete fractures for the purpose of estimating steady fluid flow rates and breakthrough curves of entrained solutes. The model formulations idealize fractures as finite discrete discontinuities within a rock matrix. Flow and transport through a fractured rock matrix is divided into three components: (1) Intra-fracture (e.g., flow within a single fracture); (2) Inter-fracture (e.g., flow within and between fractures within a fracture network); and (3) Supra-fracture (e.g., coupled fracture and matrix flow), each of which is governed by unique parameters.

Fracture Flow Conceptualization

Intra-fracture flow and transport is used to describe the movement of water and solutes through individual fractures, neglecting inflows and outflows from other fractures or the rock matrix which bounds the fracture. Intra-fracture processes include the influence of fracture surface roughness, channeling, and tortuosity on fluid flow and solute transport. Inter-fracture processes are related to the effects of fracture networks, such as the influence of fracture orientations, areal extent, densities and locations. Supra-fracture processes are used to describe the effects of coupled matrix-fracture systems, especially solute retardation and matrix diffusion due to geochemical processes.

The treatment of fractures as finite discrete continuities offers several advantages and disadvantages over alternate conceptualizations which idealize the porous medium as an equivalent continuum. While the discrete fracture network (DFN) approach used here requires detailed understanding of the physical and hydraulic properties of the fractures at the site of interest, a formidable task in most situations, the approach offers the ability to understand in greater detail and to evaluate with greater precision the processes which govern movement in unsaturated fractured rock.

DFN Flow Model Parameterization

The DFN approach employed here to simulate fluid flow and solute transport uses a two dimensional flow equation within fracture planes. The hydraulic parameters of interest within the fracture plane are (1) the fracture transmissivity and (2) the matric suction at which the fracture desaturates, termed the capillary suction. The assumption is made that only two zones are present within a fracture, a desaturated, air-filled region, and a saturated, water-filled region. In the first region, the liquid transmissivity and relative saturation are assumed to be zero, while the transmissivity within the second region assumes a constant value. An equilibrium air-water interface is assumed to exist between the desaturated and saturated regions where the matric suction in the liquid phase is equal to the capillary suction.

Application of Boundary Integral Method

The boundary integral method is used to solve the boundary value problem for the geometric and hydraulic properties expected in fractured rock. The boundary integral method takes advantage of Gauss' theorem which reduces the problem of determining unknown head and flow rates within a flow domain with uniform hydraulic properties to a problem along only the boundary of the flow domain. For saturated two-dimensional flow within a single fracture, nodes along the exterior rim of the fracture are required. For intersecting fractures planes, additional nodes are required along the finite lines of intersection. Solution of the flow problem for saturated conditions can be performed in a single iteration. Solution of the flow problem for unsaturated flow requires that an iterative scheme be employed for determining the position of the air-water interface.

Influence of Fracture Network Geometry

Synthetic fracture networks are created using planar fractures of finite areal extent embedded within a three dimensional rock matrix. Once the fracture network geometry is created, sensitivity analyses are performed to determine the effects of variable fracture network geometric properties on inferred network hydraulic conductivity. The network hydraulic conductivity of saturated fractures is determined for various geometric parameters, such as fracture orientation and density. Results of the sensitivity studies show that: (1) The global saturated hydraulic conductivity for the fracture network is linearly dependent on the product of fracture transmissivity and density for fractures of which fully penetrate the rock volume; and (2) The effect of increasing correlation between fractures of finite variable length and transmissivity is to increase the global hydraulic conductivity.

Supra-Fracture Analysis

A three-dimensional flow model is used to calculate flow through a permeable matrix with embedded permeable fractures. Exterior and interior surfaces are discretized using boundary elements to account for flow between fractures and the matrix, and between the matrix and fractures and the exterior boundaries. Results using the three dimensional coupled fracture-matrix flow regime compare favorably with analytic results.

Variably Saturated Fracture Flow

An important part of conceptualizing fluid travel times and paths is the ability to locate the interface which separates the water and air-filled regions within a fracture. Flow through variably saturated fracture networks is modeled by assigning a constant capillary suction to individual fractures. The air-water interface is found using an iterative procedure which locates nodal points at the intersection of constant total head and pressure head contours. The simulated air-water interface compares favorably with an approximate analytic solution and with laboratory results. Simulations indicate the presence of zones of water under both positive and negative potential, as well as regions of air-filled voids.

The implications of the existence of saturated regions within fractures (which may be under either positive or negative potential) are twofold: (1) Equivalent continuum models which assume uniform potentials within a variably saturated vertical fracture may provide inaccurate predictions of flow velocities; and (2) Regions of saturation will be present in vertical fractures at ambient suctions less than the capillary suction of the fractures, and also in the lowermost portion of fractures under a wide range of ambient suctions. Positive fluid potentials in fractures can lead to the enhanced movement of water from the fracture into the matrix, and thus attenuate fracture flow.

Travel Times and Breakthrough Curves

Travel times and breakthrough curves are determined for steady flow conditions by integrating the inverse velocity along a streamline, and then summing over all streamlines. The boundary integral method is used to determine the velocity distribution at discrete points along each streamline. The travel time is determined by dividing the distance between points along the streamline by the averaged velocity between the points. The total travel time is the summation of all travel times between nodes along an individual streamline. The location of points used to perform the travel time summation is determined by finding the intersection of streamlines with contours of constant total head. The intersection is found by using an iterative scheme in conjunction with the assumption that streamlines and contours of total head intersect at right angles. For the fracture network examined, travel times decrease as the matrix suction increases, or, equivalently, as the relative saturation of the fracture decreases. Most of the decrease in travel times is through regions of the fracture which are under negative potential, while the travel times within the positive potential zone are either unaffected or substantially increased.

Matrix Diffusion and Retardation

The effects of retardation and matrix diffusion due to sorption and migration into the rock matrix, respectively, are shown to delay and to attenuate solute breakthrough curves. A method for demonstrating the appropriateness of using a constant matrix diffusion attenuation coefficient is introduced which is based upon determining under what conditions the time rate of change of the attenuation coefficient is negligibly small.

Conclusion

A discrete fracture network (DFN) model is used to investigate the influence of variable fluid saturation and fracture network geometric properties on fluid flow rates, travel times and solute breakthrough curves. In contrast to equivalent porous media models which generally neglect fracture geometric properties and assume uniform saturation within individual fractures, the DFN formulation accounts for more complex flow processes. By focusing on fracture geometries and the position of an air-water interface within a fracture, more realistic estimates of flow and transport properties are obtained.

CHAPTER 1

INTRODUCTION

The ability to characterize the movement of fluids and solutes through fractured rock media of low hydraulic conductivity is a necessary precondition for identifying sites suitable for the isolation of hazardous materials. Sites located in media of low hydraulic conductivity are attractive because potential travel times from the zone of emplacement may be long compared to the rate at which the hazardous materials, especially high-level nuclear waste (HLW) and any by-products, decay.

The estimation of travel times from the containment zone to the accessible environment requires that a conceptual-physical model be formulated and solved using either analytical or numerical techniques, or both. The conceptual-physical model must be rational and subject to experimental verification. In addition, if a numerical technique is employed to determine travel times, the algorithm must be accurate and efficient. Lacking these properties, the derived information may not provide reliable estimates of the containment capacity of the site.

To provide experimental verification of the conceptual-physical model, the Apache Leap Test Site has been developed by the University of Arizona under contract with the U.S. Nuclear Regulatory Commission. The site is also used to investigate characterization procedures for fluid flow and solute transport in unsaturated, fractured rock. Nine boreholes have been installed at the site which is located near Superior, Arizona, in slightly welded volcanic tuff. A series of hydraulic, pneumatic, and tracer tests have been and are being performed at three meter intervals along the boreholes. To help design and interpret these tests a sampling and modeling strategy must be applied. The purpose of this document is to provide a methodology which can be used to design and interpret the field testing activities.

1.1 Formulation of Flow Through Fractured Rock

The characterization of ground water flow and the attendant transport of dissolved solutes in fractured rock is conceptualized as having three distinct components, termed intra-, inter-, and supra-fracture flow. Each component is described using different physical-chemical processes and parameters, the understanding and quantification of each component being necessary for the complete characterization of a flow system. Figure 1.1 illustrates the scope of the relative processes.

It is proposed that intra-fracture flow be used to describe the movement of water through individual fractures, neglecting inflows and outflows from other fractures or the rock matrix which bounds the fracture. It is assumed that the walls of the fracture are impermeable, i.e., there are no sources or sinks from other fractures or from the matrix. The walls of the fracture can be assumed to be parallel with a finite, non-zero aperture, or a distribution of fracture apertures can be used to describe intra-fracture aperture variability. Montazer and Wilson (1984) and Wang and Narasimhan (1985) present formulations using circular regions of pendular water to account for aperture variability.

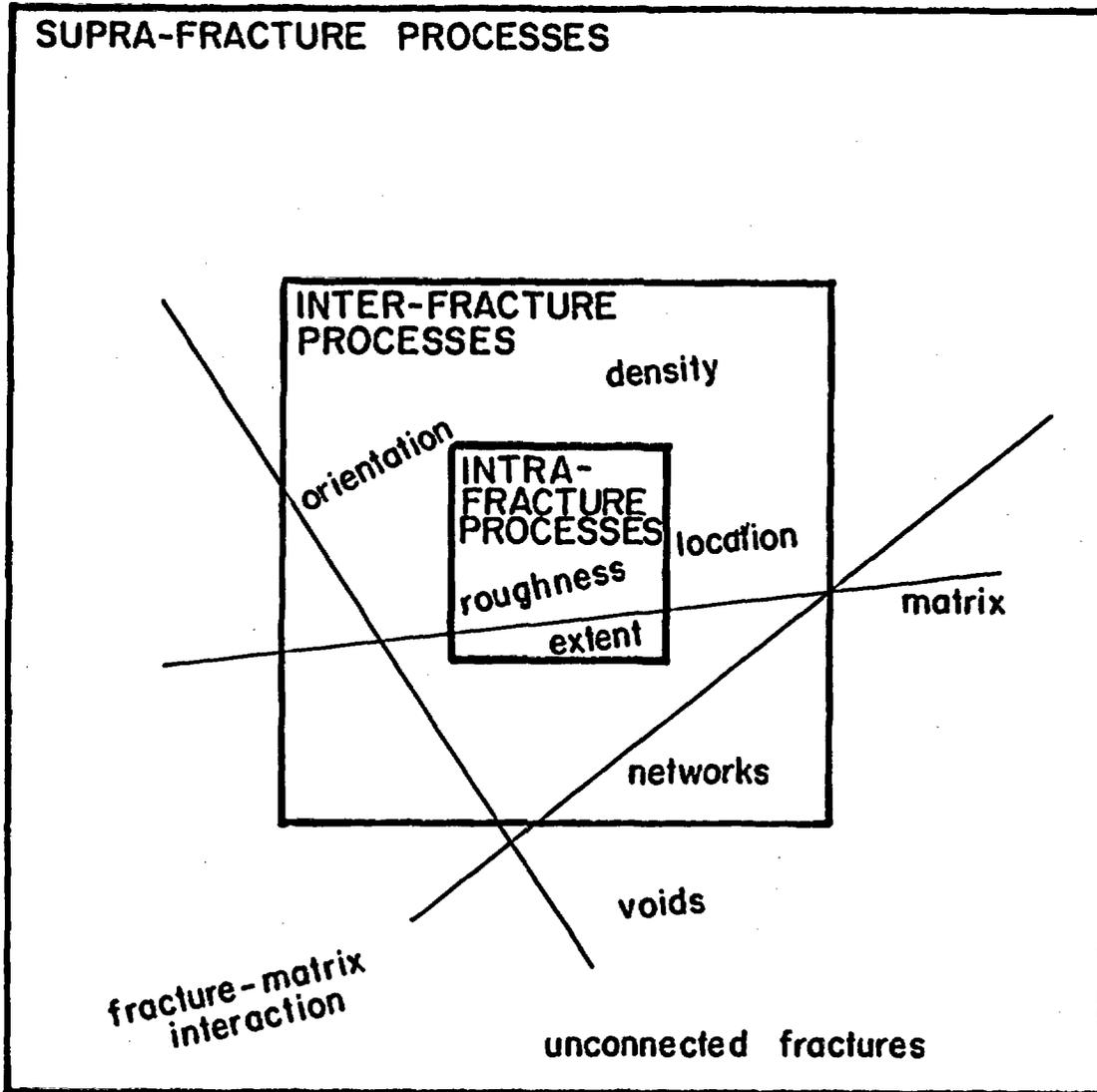


Figure 1.1: Nesting of fluid flow and solute transport processes in fractured rock.

Unlike intra-fracture flow which only incorporates fluid flow and solute transport through individual fractures, inter-fracture flow incorporates hydraulic factors associated with flow through a network of fractures, neglecting inflows and outflows from the rock matrix. The incorporation of additional dimensions in the modeling of fracture networks results in more connections between two points or surfaces than if flow through a single fracture is considered. By including alternate flow paths, a more likely fracture connection between two points or surfaces may occur.

Supra-fracture flow and transport, refers to the movement of water and solutes through a fractured rock matrix of non-zero permeability. The evaluation of the importance of supra-fracture flow requires that the interactions between the matrix and the embedded fractures be evaluated. Wang and Narasimhan (1985) have investigated the effects of sorption on fracture surfaces, as well as the movement through the matrix around fracture asperities. Other important factors affecting ground-water travel times and fluxes in the unsaturated zone are the saturated and relative hydraulic conductivities, the moisture release curve, and the stratigraphy or variation of these properties over space. At any point in a geologic medium, the saturated hydraulic conductivity generally remains constant, only changing due to variable stress loading and temperature changes. Unsaturated hydraulic conductivity, however, can vary with water content over time at a point, even at constant stress levels and temperatures. To determine solute travel times through fractures embedded in a porous matrix the effects of sorption and matrix diffusion must be considered. Other processes, such as chemical precipitation and dissolution, chelation, colloid formation and movement, radioactive decay and the attendant production of decay and/or degradation products, and volatilization of the solute, may also be important in controlling solute transport.

1.1.1 Intra-Fracture Flow and Solute Transport

Fluid flow through individual fractures has been studied in the laboratory by Sharp (1970), Iwai (1976), Schrauf and Evans (1986), Kilbury et al. (1986) who demonstrate that a linear relationship exists between the flow rate and the applied fluid gradient, as long as flow is laminar. Laboratory and field tests, along with simulation models, have documented the effects of fracture roughness (Schrauf and Evans, 1986), tortuosity (Tsang, 1984), and channels (Tsang and Tsang, 1987) within individual fractures on the measured fluid flow in response to an applied fluid gradient.

A description of physical processes affecting solute dispersion is provided by Neretnieks (1983) and the effect of flow channels within fractures is described by Tsang and Tsang (1987). Analytical solutions developed by Tang et al. (1981), Sudicky and Frind (1982, 1984), Rasmuson (1985), Rasmuson and Neretnieks (1986), and Moreno and Rasmuson (1986) have yielded important results concerning the physical processes of dispersion, retardation and diffusion within individual fractures. Rasmuson et al. (1982) and Neretnieks and Rasmuson (1984) have presented an integrated finite difference model for simulating the movement of radionuclides in a stream tube with arbitrary velocity. In addition,

the effect of local permeability perturbations on observed dispersion within porous media has been described by Neuman et al. (1987). Laboratory studies of dispersion in a natural rock fracture conducted by Moreno et al. (1985) are the only results available for model calibration.

Fluid flow through variably saturated fractures has received less attention than single phase fracture flow. Wang and Narasimhan (1985) proposed a phase constriction factor which accounts for zones of pendular water around fracture asperities that increase in size as the fluid suction decreases. Validation of this model has not been performed to date using field or laboratory flow studies, although fracture surface mapping studies by Myer et al. (1986) to determine the fracture void geometry suggest that the phase constriction factor may be a viable descriptor of fracture hydraulic properties. Evans and Rasmussen (1988) describe ongoing laboratory and studies which describe the influence of fluid suction on fluid flux and velocity, as well as solute transport processes.

1.1.2 Inter-Fracture Flow and Solute Transport

A quantitative means for estimating the hydraulic properties of a discrete fracture network using information about fracture density, aperture, orientation, and assuming infinite fracture length was presented by Snow (1965, 1969), which provides an estimate of the equivalent porous medium hydraulic conductivity tensor using data easily gathered from boreholes or mines. While Snow assumed that fractures are of infinite length, many networks consist of fractures which are of finite length, and so other methods must be used to evaluate the hydraulic properties of a fractured rock mass.

Percolation theory provides a semianalytic means for estimating the conductivity of a medium. The theory can be used to describe the effect point interconnection variability has on the interconnections at longer distances. Recent literature (Castellani et al., 1981; Goldman and Wolf, 1983; Hughes and Ninham, 1983; Kesten, 1982; Orbach, 1986; Rodrigues and Tondeur, 1981) investigates the connectiveness of a system of pores that have a specified probability for intersecting neighboring pores. Two types of percolation networks have been proposed; site and bond networks. A site percolation network has been described by Castellani et al. (1981) as a periodic lattice of sites with each site being occupied with a probability, p , and empty with probability, $(1-p)$, independent of the status of its neighbors. A cluster of sites is defined as a group of neighboring occupied sites. As the percolation parameter, p , increases in value, there exists a percolation threshold, $p = p'$, at which any site within the lattice will be connected with every other point to form an infinite cluster.

Unlike site percolation models, bond percolation networks have been described by Hughes and Ninham (1983) as a periodic lattice of points connected by bonds which are assigned at random, and independently of each other. The bonds are assigned with a probability, p , of being occupied and probability $(1-p)$ of being vacant. For an infinite lattice, this assignment is equivalent to removing a fraction $(1-p)$ of all bonds at random. Two sites are called connected if there exists at

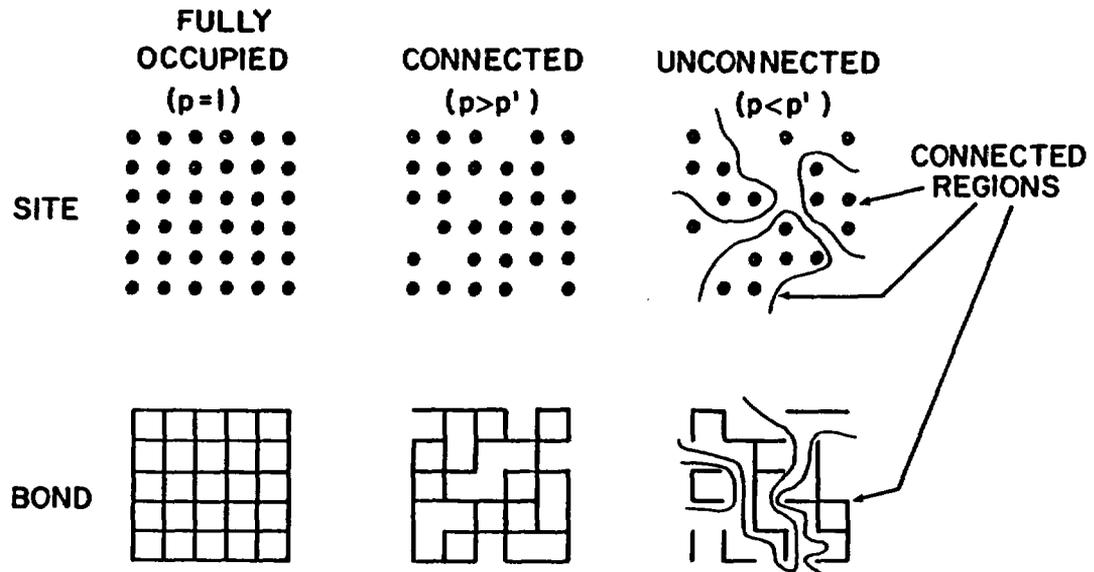


Figure 1.2: Site and bond percolation models for three densities of site and bond occupancy, p . Note that networks remain fully connected until the occupancy drops below a critical threshold density, p' .

least one path between them consisting solely of occupied bonds. In each realization of the bond problem there will be connected clusters of sites, linked together by occupied bonds, and bounded by vacant bonds. Similar to the site problem, there will be a bond percolation threshold which defines the probability, p' , at which an infinite cluster will develop. Figure 1.2 illustrates the difference between site and bond models for three probabilities.

Investigations of dispersion within randomly connected fractal fracture networks have also been performed. Ross (1986) showed that small pulses of contaminant tend to grow as the square root of distance traveled for networks with few high transmissivity fractures. For networks with frequent highly transmissive fractures the small pulses of contaminant tend to grow linearly with distance traveled.

Lacking exact analytic techniques to evaluate the permeability of fractured rocks, a number of researchers have developed computer simulation

models for relating local hydraulic properties to global rock permeability. For example, Silliman (1986) used a stochastic approach related to percolation theory for the purpose of comparing the effective permeability of a rock mass with the minimum permeability of flow paths that cross a given volume within the subsurface. His results, using a three-dimensional, nearest-neighbor, site-problem study, showed that the minimum permeability of flow paths between any two points is greater than the permeability calculated using globally-averaged properties.

Long and Witherspoon (1985) have also used simulation techniques to evaluate the permeability of a fractured rock mass. Their study showed that the interconnection between given fracture sets is a complex function of fracture density and fracture extent. Other researchers (Smith and Schwartz, 1984; Long et al., 1985; Huang and Evans, 1985; Reeves et al., 1986) have proposed additional one, two, and three dimensional discrete fracture network (DFN) simulation models.

Smith and Schwartz (1984) present a two dimensional DFN model in which flow and mass transport occurs through two orthogonal fracture sets. In this formulation, one fracture set is perpendicular to the imposed potential gradient, while the second set provides connective routes between discontinuous fractures of the first set. They found that the second fracture set allows transport through pathways which would not have been available otherwise. If this is true, then as more dimensions are provided, it may be possible that flow and mass transport are increased due to the additional pathways provided. They also report that the addition of alternate fracture paths in the second dimension results in an increase in macroscopic dispersion, an earlier initial breakthrough time, and a delay of the final arrival time.

Long et al. (1985) extended a two dimensional DFN model (Long et al., 1982) to three dimensions by analytically solving for boundary conditions within circular disks of uniform aperture. While the formulation is currently restricted to applications involving a limited number of circular disks of uniform aperture, the ability to solve for flow and pressure heads provides a solution for complete three dimensional flow, allowing for validation with other simulation models.

Huang and Evans (1985) proposed a conceptual and numerical technique to simulate a wide range of field conditions. The proposed methodology uses a three dimensional formulation of the fracture network with a one dimensional flow tube formulation within the plane of the fracture. The model is used to provide estimates of global hydraulic conductivity within a rock mass for specified distributions of fracture orientations, apertures, and densities. Boundary conditions are specified by defining the pressure head along the exterior boundary of the simulated rock mass. Steady, saturated flow conditions are assumed. The model is also used to calculate solute travel times and breakthrough curves. Mass transport of solutes is simulated using a piston flow approximation. A major deficiency of the Huang and Evans model is the simplification of flow through an individual fracture to a one dimensional formulation which does not incorporate interaction between multiple sources and sinks within a fracture, or the influence of orientation of the fracture-fracture intersections (Rasmussen, 1987).

An early attempt to characterize fluid flow through unsaturated porous media was proposed by Fatt (1956a,b,c) who used bundles of capillary tubes. The technique estimated unsaturated hydraulic properties of porous media by applying capillary theory to fluid flow through a model incorporating a network of tubes interconnected at regular intervals. By combining a distribution of tube sizes (obtained from pore size distributions) with capillary theory, the unsaturated wetting phase saturation was calculated, along with the associated unsaturated relative hydraulic and pneumatic conductivities. Also, by using various geometric networks Fatt was able to demonstrate the effect of pore size irregularities on unsaturated hydraulic properties.

1.1.3 Supra-Fracture Flow and Transport

Fluid flow through unsaturated fractured rock is formulated by defining physical and hydraulic parameters for the rock matrix and for fractures embedded within the rock. The physical and hydraulic properties of the matrix include the pore size distribution, the dependence of the hydraulic conductivity on water content and fluid potential, the moisture characteristic curve, and the pneumatic conductivity. The physical properties of the fractures include the orientation, areal extent, fracture center location, and shape. The hydraulic property of interest for fluid flow within fractures is the fracture transmissivity which will vary due to geochemical processes, such as dissolution and precipitation, and variable saturation.

Because properties of only a few fractures can be identified in field-scale problems, statistical techniques must be employed to characterize sets of fractures. Frequency distributions are obtained for each fracture parameter which best approximate observed characteristics of the fracture network at the site of interest. A number of approximation techniques have been developed for the purpose of modeling fluid flow through both a porous rock matrix and fractures embedded within the matrix. The techniques are usually solved numerically, but in some circumstances solutions can be obtained using analytic techniques.

A widely used approach for predicting fluid flow and travel times through low permeability fractured rock is to assume that fluid flow through such media behaves in a manner similar to flow through porous media. The equivalent porous medium (EPM) formulation assumes that a uniformly porous material consists of a large number of microscopic flow routes which, upon averaging, provides consistent macroscopic parameters that are used for modeling ground water flow and solute transport (El-Kadi and Brutsaert, 1985). The EPM formulation is attractive because of the widespread availability of numerical algorithms to solve problems of this type. In general, solutions for problems related to flow through both porous media can be grouped into three broad categories, i.e., analog, analytic, and numeric.

Analog solutions can be obtained using sandboxes or electric analog models (Karplus, 1958). Analytic (or, equivalently, closed form) solutions are available for a number of flow problems (see, for example: Muskat, 1946; Carslaw and Jaeger, 1959; Churchill, 1974 Philip, 1985; Waechter and Philip, 1985; Wheatcraft and Winterberg, 1985). In many circumstances, the analytic expressions are appropriate only for uniform

material properties. To account for variable hydraulic parameters, analytic stochastic models have been developed (see: Gelhar and Axness, 1983; Mantoglou and Gelhar, 1987a,b,c; Yeh et al., 1985a,b,c). Gelhar and Axness (1983) used stochastic theory to estimate the effective hydraulic conductivity tensor for conditions of saturated flow in a statistically anisotropic medium with arbitrary orientation of the major axes of mean flow. The effective (or large-scale) hydraulic conductivity tensor is shown to be of second rank and symmetric. Parameters for the stochastic model include the mean, variance and covariance function of the natural logarithm of the local scale saturated hydraulic conductivity, as well as the orientation of the mean flow direction.

Numeric models include finite difference models (such as by: Travis, 1984; Reeves et al., 1986; Pruess, 1987), finite element models (see, for example: Davis and Neuman, 1983; Wang and Narasimhan, 1985; Allen and Murphy, 1986; Huyakorn et al., 1984, 1985, 1986; Noorishad and Mehran, 1982; Pinder and Gray, 1977; Segerlind, 1984), and boundary element models (for example those by: Brebbia, 1978, 1981a,b, 1984; Brebbia and Ferrante, 1979; Brebbia and Maier, 1985; Brebbia and Noye, 1985; Brebbia et al., 1984a,b; Cheng, 1984; Elsworth, 1986, 1987; Lafe and Cheng, 1987; Lafe et al., 1981; Lennon et al., 1979a,b; Liggett and Liu, 1979a,b, 1983; Liu et al., 1981; Shapiro and Anderson, 1983; Andersson and Dverstorp, 1987).

Various applications of the EPM concept to fractured media have been made. One application shows that the EPM formulation is valid if there are sufficient fractures for statistical averaging of flow paths (Neuzil and Tracy, 1981; Long et al., 1982). The scale of statistical averaging which is required to obtain a sufficient number may be large, especially when fracture densities are low (Sagar and Runchal, 1982). In practice the size of the rock volume for which the hydraulic parameter is estimated is enlarged to a size which will result in consistent hydraulic parameters (see e.g., Smith and Schwartz, 1984; Witherspoon et al., 1979).

Dual porosity models have been developed for circumstances when substantial flow through both a rock matrix and rock fractures occur simultaneously. Unlike single-porosity EPM models which lump matrix and fracture properties into a single parameter, dual-porosity models differentiate between fracture and matrix flow by solving two sets of flow equations using a coupling parameter to represent flow between the matrix and fractures. This technique has been used for saturated conditions (see, for example: Bibby, 1981; Huyakorn et al., 1983; Moench, 1984), as well as to generate composite unsaturated hydraulic conductivity curves of fractured rock (Wang and Narasimhan, 1985, Tsang and Pruess, 1987).

Simulating flow through unsaturated porous media has been well described in the literature (see, for example: Andersson and Shapiro, 1983; Bresler and Dagan, 1982a-b; Cooley, 1983; Dagan and Bresler, 1982; Huyakorn et al., 1983a,b,c, 1984, 1985, 1986; Nielsen et al., 1986; Pollock, 1986; Ross, 1984). Flow through saturated, fractured rock has also received considerable attention (for example: Neuzil and Tracy, 1981; Sagar and Runchal, 1982; Castillo et al., 1972; Chen, 1986; Ross, 1986; Hsieh and Neuman, 1985; Hsieh et al., 1985).

Attention to flow in variably saturated fractured rocks has received less attention. Wang and Narasimhan (1985) used a measured fracture aperture distribution to derive a theoretical relationship between fracture hydraulic conductivity and pressure head which considers the influence of asperities. With this theoretical relationship they used a numerical model to simulate flow in regularly spaced fractures of infinite areal extent bounded by matrix blocks. The model provides estimates of changes in saturation levels, permeabilities, darcian flow velocities in fractures and at fracture-matrix interfaces, and in effective fracture-matrix flow areas.

Tsang and Pruess (1987) have also investigated coupled fracture and matrix flow through unsaturated tuff. Their model employs a finite difference mesh to discretize the region surrounding a high level nuclear repository. Coupled vapor and fluid flow of water is investigated by assigning an effective permeability between nodes which is a composite of both matrix and fracture permeabilities. Complex fracture orientations are not investigated, nor are differential saturation levels between the rock matrix and embedded fractures.

Stochastic theory has also been used by Yeh et al. (1985a,b,c) to describe steady unsaturated flow in a heterogeneous medium. It is demonstrated that the effective hydraulic conductivity of a statistically anisotropic medium has tensorial properties, and also that the anisotropy of the hydraulic conductivity is dependent upon the moisture content of the medium. This results from the variation in pore size distributions within the soil medium and the correlation structure of the variation. Extensions of these results to fractured media are possible if macroscopic (i.e., large scale) hydraulic properties of the fractured rock can be determined.

1.2 Proposed DFN Models

Of immediate concern is the demonstration of an ability to define the important geometric and physical characteristics of a fractured rock medium which influence the hydraulic and transport properties of variably saturated fractured rock. Specific goals include the determination of critical geometric properties of fractures with respect to bulk hydraulic properties, the influence of fluid suction on the relative saturation and hydraulic conductivity of fractures, and the behavior of fluid and solute travel times and breakthrough curves in variably saturated fractures incorporating sorption and matrix diffusion.

In order to meet these goals, this study is organized into three broad areas of study. The first study, presented in Chapter 2, focuses on fluid flow through individual and networks of discrete fractures. The second, presented in Chapter 3, focuses on fluid flow through variably saturated discrete fractures, while the last study, presented in Chapter 4, investigates the effect of sorption, matrix diffusion and variable saturation on solute travel times and breakthrough curves resulting from flow through discrete fractures. Appendix A provides the computer models used in the simulation studies.

1.2.1 Steady Saturated Flow Through Discrete Fracture Networks

The objectives of this study are to identify how geometric properties of saturated fractures control the bulk hydraulic properties of fractured rock. In particular, the effects of fracture orientation, density, length and transmissivity on the global hydraulic conductivity (i.e., the effective hydraulic conductivity evaluated at a scale which incorporates flow through multiple fractures) are evaluated for both flow within impermeable rock, as well as for coupled flow through fractured permeable rock.

In order to achieve the stated objectives, numerical simulation using the boundary integral method is used to model steady, saturated fluid flow through networks of discrete fractures. The boundary integral method requires that flow domain boundaries be discretized into finite line segments for two-dimensional flow through a fracture, and into finite planar elements for three-dimensional flow through a rock matrix. Simulation results demonstrate that fracture orientation with respect to the mean direction of the fluid gradient plays an important role in determining the magnitude of the fluid flow, as does the length of fractures, and the correlation between fracture lengths and transmissivities.

As opposed to previous studies, described above, which reduce individual two-dimensional fractures to one-dimensional line segments, the formulation presented here provides a more complete geometric representation of individual fractures by maintaining a two-dimensional fracture geometry. Simulation results derived from the proposed methodology provide additional evidence for determining the conditions under which equivalent porous media models are suitable for characterizing steady saturated flow through discrete fractures.

1.2.2 Steady Flow Through Variably Saturated Fractures

The objectives of this study are to evaluate the effects of variable fluid suction and fracture orientation on the hydraulic properties of fractures. Specifically, the behavior of a free surface within a fracture which delimits the air-water interface is investigated under conditions of variable fracture orientation and fluid suction. Also, an important objective is the determination of regions of saturation within individual fractures, along with the regions of positive, negative, and undefined hydraulic head. (Unsaturated regions will correspond to regions with an undefined head, while saturated regions will contain both positive and negative heads.)

In order to determine the hydraulic head distribution, as well as the region of saturation, the boundary integral method is used to discretize the saturated region and to solve the hydraulic head distribution within the plane of a fracture. To determine the position of the free surface, a constant capillary head is assigned to each fracture and the location of nodes along the free surface are adjusted until the calculated hydraulic pressure head at the air-water interface is equal to the capillary head. Simulation studies indicate that regions of saturation are limited to the regions immediately around a source, extending below the source to a water table. Below the water table is a region of

positive fluid pressure heads. The position of the water table is defined by the location of the outlet from the fracture.

The methodology proposed here is superior to the existing methods, presented above, in that constant or slowly varying fluid suction within the plane of individual fractures are not assumed. Instead, large variations in head will exist, assuming values ranging from large suction to large positive pressure heads, dependent upon location within the fracture and the orientation of the fracture. The accurate definition of fracture head variation, as well as the definition of the saturated regions within a fracture, are important factors in determining the suitability of using existing equivalent porous media models for characterizing fluid flow through unsaturated rock.

1.2.3 Solute Transport Through Variably Saturated Fractures

The objectives of the third study are to develop methodologies for calculating travel times and breakthrough curves for water and solutes in variably saturated fractured rock. Travel times and release rates are important characterization properties of repository performance (USNRC, 1987). The developed methodologies must also incorporate transport processes which ameliorate the effects of contaminant release, such as sorption and matrix diffusion, as well as time dependent inputs.

The proposed methodologies consist of determining the integrated inverse velocity along streamlines. The effect of flow path variation on the shape of breakthrough curves is considered by discretizing calculated fluxes into unique, one-dimensional streamlines orthogonal to potential lines. Velocity variation along a streamline is incorporated by discretization along a streamline, and by accumulating travel times between discretization points. Simulation results show that velocity variations along and between streamlines result in substantial variability in the calculated travel time. Decreased travel times are noted as the relative saturation is decreased.

The boundary integral method provides superior estimates of travel times because, unlike finite element and finite difference methods which use piecewise interpolation functions of various orders over each spatially discretized interval, the boundary integral method defines smooth functions of velocity and streamline functions which are continuous in all derivatives within the discretized flow domain.

1.2.4 Computer Simulation Models of Fracture Flow and Transport

Four computer models, written in FORTRAN-77, are developed to simulate fluid flow and solute transport through discrete fractures. Presented in Appendix A, the programs provide the ability to generate fracture networks using synthetic or field data, as well as to solve for fluid head and flow rates within fractures by discretizing the fracture boundaries and the matrix flow domain using the boundary integral method.

Program BIM provides estimates of steady flow rates, hydraulic head distributions, travel times and breakthrough curves for discrete fracture networks, incorporating both saturated and variably saturated flow. Program FRACGEN is used to determine the global hydraulic conductivity

of a fractured rock mass by generating finite fractures within a specified rock volume and then solving for the finite lines of intersections between fractures and between fractures and the rock volume boundary. The program uses site-specific geometric data, or can generate synthetic fractures using distribution of fracture parameters including fracture orientation, length, and density. Once a fracture network has been generated using program FRACGEN, Program BIM2D is used to discretize the fracture network and then solve for steady fluid flow and transport using the boundary integral method. Program BIM2D is limited to applications involving an impermeable rock matrix. Program BIM3D is used to investigate coupled flow through a fracture network embedded within a permeable matrix.

CHAPTER 2

NUMERICAL SIMULATION OF STEADY FLUID FLOW

A numerical procedure is presented for obtaining estimates of steady fluid flow through saturated discrete fractures. The procedure uses the boundary integral method to discretize and solve the boundary value problem for hydraulic head and fluxes within discrete fractures. Applications of the method include flow through single and interconnected fractures which are embedded within an impermeable matrix, as well as subsequent applications to fractures contained within a permeable matrix. For fracture flow within an impermeable matrix, the effect of fracture orientation, density, and fracture transmissivity distributions are evaluated for their effects on the global three-dimensional hydraulic conductivity, which is a measure on a macroscopic scale of the hydraulic properties of the fractured rock.

2.1 Fracture Flow Hydraulics

Steady fluid flow through a porous medium is governed by equations which incorporate a mass balance constraint. For flow through fractures and the rock matrix, the mass balance equations are, respectively:

$$(2.1a) \quad \nabla \cdot \underline{q}_f(\underline{x}) = 0 \quad \text{in } \Omega$$

and

$$(2.1b) \quad \nabla \cdot \underline{q}_m(\underline{x}) = 0 \quad \text{in } R$$

where

∇	gradient operator, 1/m;
\underline{q}_f	darciian flux through a planar fracture, m^2/s ;
\underline{q}_m	darciian flux through the rock matrix, m/s ;
\underline{x}	position vector, m ;
Ω	two dimensional planar fracture flow domain; and
R	three dimensional spatial matrix flow domain.

The relationship between fluid flux and the force driving fluid flow is defined using the tensorial form of Darcy's law, in two and three dimensions, respectively:

$$(2.2a) \quad \underline{q}_f(\underline{x}) = - \underline{T}(\underline{x}) \nabla h(\underline{x})$$

and

$$(2.2b) \quad \underline{q}_m(\underline{x}) = - \underline{K}(\underline{x}) \nabla h(\underline{x})$$

where

h	hydraulic head, m ;
\underline{K}	hydraulic conductivity, m/s ; and
\underline{T}	fracture transmissivity, m^2/s .

Assigned head and flux boundary conditions are, respectively (Figure 2.1):

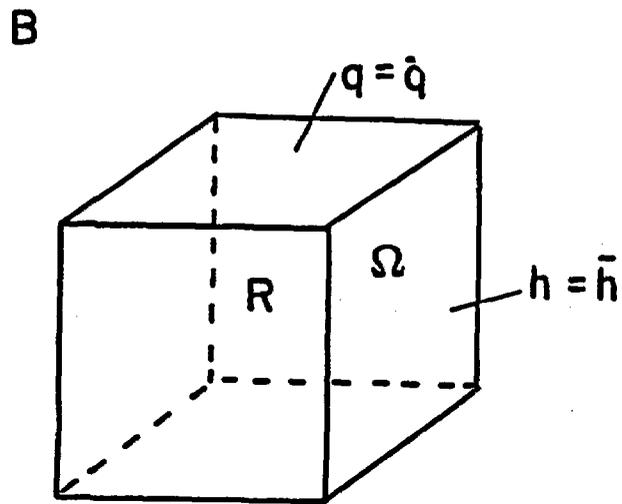
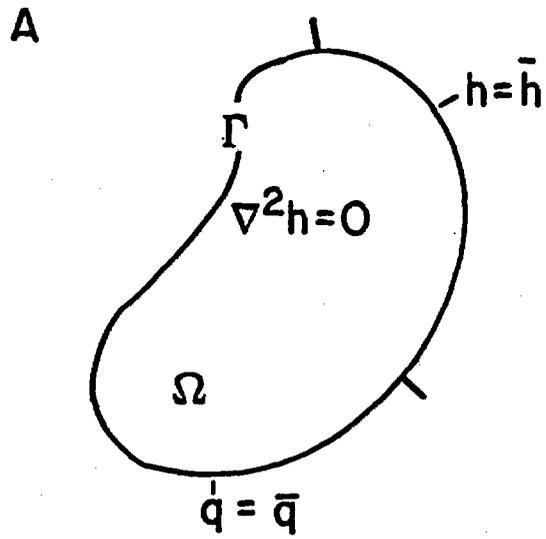


Figure 2.1: Flow domain and boundary conditions for two dimensional (A) and three dimensional (B) porous media. Symbols are defined in text.

$$(2.3a) \quad h(\underline{x}) = \bar{h}(\underline{x}) \quad \text{on } \Gamma \text{ or } \Omega$$

or

$$(2.3b) \quad \underline{q}(\underline{x}) = \bar{\underline{q}}(\underline{x}) \underline{n}(\underline{x}) \quad \text{on } \Gamma \text{ or } \Omega$$

where

\bar{h} assigned head boundary condition, m;
 $\bar{\underline{q}}$ assigned flux boundary condition, m/s;
 \underline{n} outwardly directed unit vector normal to boundary; and
 Γ one dimensional linear domain.

Total fluid flow across a boundary is calculated by integrating the darcian velocity over the boundary, or:

$$(2.4a) \quad Q_f = \int \underline{q}_f(\underline{x}) \underline{n}(\underline{x}) d\Gamma$$

and

$$(2.4b) \quad Q_m = \int \underline{q}_m(\underline{x}) \underline{n}(\underline{x}) d\Omega$$

where Q_f and Q_m are fracture and matrix flow rates (m^3/s), respectively. For a rock mass incorporating both a porous rock matrix and embedded fractures oriented parallel to the gradient, the total fluid flow is the sum of fracture and matrix flow components:

$$(2.5) \quad Q = \int \underline{q}_f(\underline{x}) \underline{n}(\underline{x}) d\Gamma + \int \underline{q}_m(\underline{x}) \underline{n}(\underline{x}) d\Omega$$

For constant darcian fluxes over the fracture boundaries and a fracture boundary lying upon a matrix boundary (Figure 2.2), Equation 2.5 can be reduced to:

$$(2.6) \quad Q = q_f w_f + q_m A_m$$

where

$$q_f = \underline{q}_f(\underline{x}) \underline{n}(\underline{x})$$

$$q_m = \underline{q}_m(\underline{x}) \underline{n}(\underline{x})$$

and

w_f extent of the fracture intersecting the matrix boundary, m;

A_m area of the matrix boundary, m.

The mean darcian flux over the cross sectional area, q (m/s), for the conditions of Equation 2.6 is calculated using:

$$(2.7) \quad q = Q / A$$

where A is the total surface area of rock. By noting that the matrix area, A_m , is not appreciably different than the total surface area, A , and that the extent of fractures can be related to the total area using a density measure, i.e.:

$$(2.8) \quad d_f = w_f / A$$

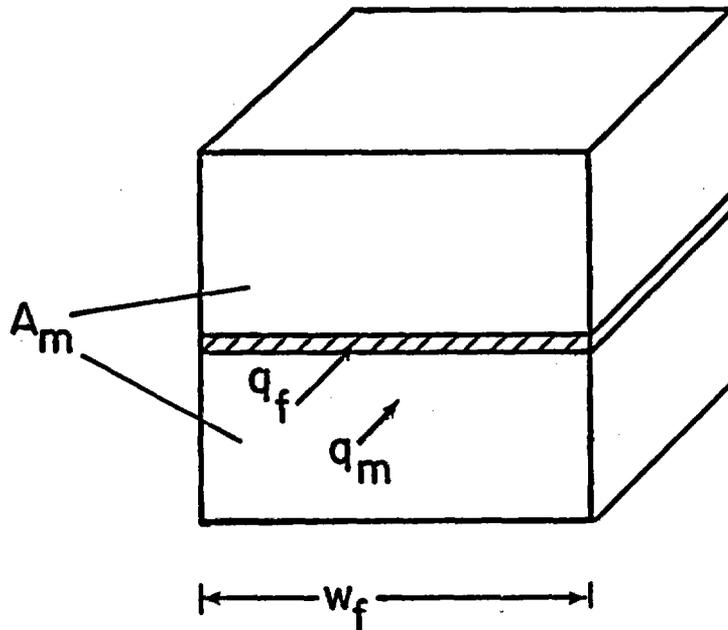


Figure 2.2: Fluxes, q_f and q_m , and geometric properties, w_f and A_m , for a single fracture and rock face intersecting a boundary.

where d_f is the fracture extent per unit rock surface (1/m), the following relationship is obtained:

$$(2.9) \quad q = d_f q_f + q_m$$

The intrinsic permeability of a porous medium is related to the hydraulic conductivity for an isotropic medium with constant viscosity and fluid specific weight, using:

$$(2.10a) \quad T = k_f \gamma / \mu$$

and

$$(2.10b) \quad K = k_m \gamma / \mu$$

where

- k_f fracture permeability, m^3 ;
- k_m matrix permeability, m^2 ;
- γ specific weight, Pa/m; and
- μ dynamic viscosity, Pa s.

Substituting Equations 2.2 and 2.10 into Equation 2.9 and assuming a equivalent hydraulic gradient in both the fracture and the matrix

yields:

$$(2.11) \quad q = -(d_f k_f + k_m) \gamma/\mu Vh = -k \gamma/\mu Vh$$

or

$$(2.12) \quad k = d_f k_f + k_m$$

where k is the bulk permeability of the fractured rock. The fracture permeability can be determined using injection tests such as those devised by Kilbury et al. (1986). The fracture density can be measured using exposed rock surfaces and borehole core samples, while the matrix permeability can be measured using unfractured rock samples or borehole tests in unfractured intervals.

In some circumstances, fluid flow within a fracture can be assumed to obey Poiseuille's law which relates the intrinsic permeability of a fracture to the hydraulic aperture of the fracture. This assumption has been examined in several studies (Iwai, 1976; Schrauf and Evans, 1986; Kilbury et al., 1986; Witherspoon et al., 1980). The relationship between the fracture permeability and the fracture aperture, e (m), is:

$$(2.13) \quad k_f = e^3 / 12$$

Equation 2.13 is appropriate for the case of a planar fracture with constant cross-sectional area. For a fracture with variable aperture, Smith et al. (1987) demonstrate that Equation 2.13 does not provide estimates of fracture aperture which compare with estimates made using tracers or with a volume balance calculation. Schrauf and Evans (1986) show that the volume balance calculation provides a larger estimate of aperture than Equation 2.13 (Figure 2.3). Given these ambiguous relationships between fracture permeability and aperture as measured by different methods, the aperture defined by Equation 2.13 is not used in this study. Instead, a fracture transmissivity and permeability are used to relate the hydraulic gradient to darcian fluxes.

A global hydraulic conductivity, \underline{K}_g (m/s), for an assemblage of fractures by noting that:

$$(2.14a) \quad \underline{K}_g = \underline{Q} / [\underline{A} Vh(\underline{x})]$$

and

$$(2.14b) \quad \underline{K}_g = \underline{d}_f \underline{T} + \underline{K}$$

These two relationships provide alternate methods for experimentally determining the bulk hydraulic properties of a fractured rock medium. The first method assumes an equivalent porous media exists for the fractured rock so that macroscopic properties can be determined using existing porous medium hydraulic testing procedures (Hsieh and Neuman, 1985). The second formulation uses field data about fracture spacing and transmissivity, in conjunction with laboratory or field estimates of matrix hydraulic conductivity, to provide an estimate of the macroscopic hydraulic properties of the fractured rock. Section 2.3 provides simulation studies which compare these formulations.

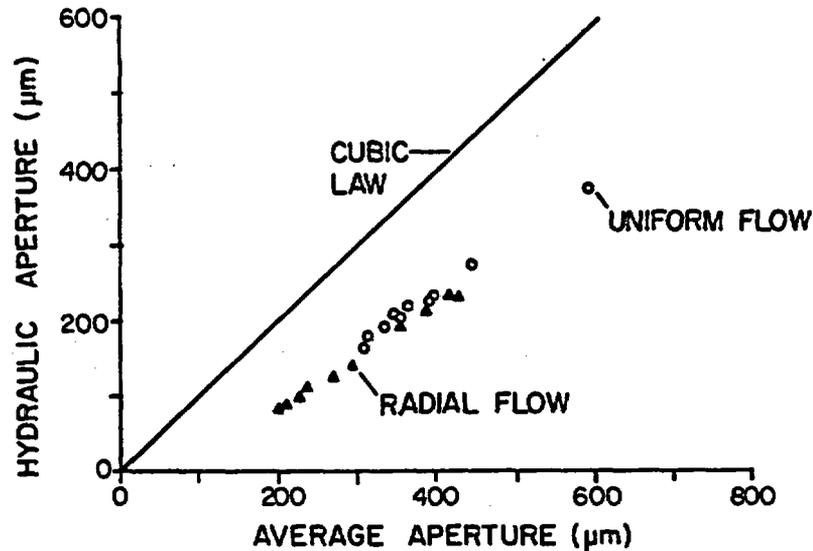


Figure 2.3: Measured relationship between hydraulic and mass balance fracture aperture.

2.2 Boundary Integral Method

The boundary integral method has been widely applied within the field of fluid hydraulics and subsurface flow modeling (Lafe et al., 1981), normally for two dimensional applications. A few three dimensional applications have been reported (Liggett and Liu, 1983; Brebbia et al., 1984; Huyakorn and Pinder, 1983), as well as applications to flow through fractured rock (Shapiro and Andersson, 1983; Elsworth, 1986, 1987). Such studies have examined flow through homogenous media, or through layered media with homogeneous properties within each layer. Recent advances have also provided the ability to examine two dimensional flow through heterogeneous media (Lafe and Cheng, 1987).

In order to determine fluid head within a prescribed flow domain subject to arbitrary boundary conditions, Gauss's formula can be used to solve Laplace's equation in two and three dimensions, respectively:

$$(2.15a) \quad \int V^2 h \, d\Omega = \int \partial h / \partial \underline{n} \, d\Gamma$$

and

$$(2.15b) \quad \int V^2 h \, dR = \int \partial h / \partial \underline{n} \, d\Omega$$

where \underline{n} is the direction normal to boundary. If, instead of Vh , we use an arbitrary weighting function fVg in Equation 2.15, we have Green's first identity:

$$(2.16a) \quad \int (fV^2g + VfVg) \, d\Omega = \int f \, \partial g / \partial \underline{n} \, d\Gamma$$

and

$$(2.16b) \quad \int (fV^2g + VfVg) \, dR = \int f \, \partial g / \partial \underline{n} \, d\Omega$$

Due to symmetry in the middle term, interchange of f and g and subtraction of terms yields Green's second identity:

$$(2.17a) \quad \int (fV^2g - gV^2f) \, d\Omega = \int (f \, \partial g / \partial \underline{n} - g \, \partial f / \partial \underline{n}) \, d\Gamma$$

and

$$(2.17b) \quad \int (fV^2g - gV^2f) \, dR = \int (f \, \partial g / \partial \underline{n} - g \, \partial f / \partial \underline{n}) \, d\Omega$$

Assigning $h(\underline{x})$ to g and $h^*(\underline{x}, \underline{x}^*)$ to f results in:

$$(2.18a) \quad \int (h^*(\underline{x}, \underline{x}^*) V^2h(\underline{x}) - h(\underline{x}) V^2h^*(\underline{x}, \underline{x}^*)) \, d\Gamma = \\ \int (h^*(\underline{x}, \underline{x}^*) \, \partial h(\underline{x}) / \partial \underline{n} - h(\underline{x}) \, \partial h^*(\underline{x}, \underline{x}^*) / \partial \underline{n}) \, d\Gamma$$

and

$$(2.18b) \quad \int (h^*(\underline{x}, \underline{x}^*) V^2h(\underline{x}) - h(\underline{x}) V^2h^*(\underline{x}, \underline{x}^*)) \, dR = \\ \int (h^*(\underline{x}, \underline{x}^*) \, \partial h(\underline{x}) / \partial \underline{n} - h(\underline{x}) \, \partial h^*(\underline{x}, \underline{x}^*) / \partial \underline{n}) \, d\Omega$$

where

$h^*(\underline{x}, \underline{x}^*)$ weighted residual function, m ; and
 \underline{x} position vector of weighted residual function.

An approximate solution for fluid head and flux which minimizes the error between the true and estimated head and flux over the flow domain is obtained by assigning the weighted residual statement (Brebbia et al., 1984) in both two and three dimensions, respectively:

$$(2.19a) \quad \int [h^*(\underline{x}, \underline{x}^*) V^2h(\underline{x})] \, d\Omega = 0$$

and

$$(2.19b) \quad \int [h^*(\underline{x}, \underline{x}^*) V^2h(\underline{x})] \, dR = 0$$

The weighted residual function is dependent on position within the flow domain and on the physical and fluid properties within the flow domain. The first half of the integral on the left-hand side of Equation 2.18 is equal to zero (from Equation 2.19), while the remaining half is calculated by noting that (Brebbia et al., 1984):

$$(2.20a) \quad V^2h^*(\underline{x}, \underline{x}^*) = -2\pi \Delta(\underline{x}, \underline{x}^*)$$

and

$$(2.20b) \quad \nabla^2 h^*(\underline{x}, \underline{x}^*) = -4\pi \Delta(\underline{x}, \underline{x}^*)$$

in two and three dimensions, respectively, where $\Delta(\underline{x}, \underline{x}^*)$ is the Dirac delta, with properties:

$$(2.21) \quad \Delta(\underline{x}, \underline{x}^*) = \begin{cases} 0 & \underline{x}^* \neq \underline{x} \\ \infty & \underline{x}^* = \underline{x} \end{cases}$$

Substituting Equation 2.20 into Equation 2.18 and noting that:

$$(2.22a) \quad \int h(\underline{x}) \Delta(\underline{x}, \underline{x}^*) d\Omega = h(\underline{x}^*)$$

and

$$(2.22b) \quad \int h(\underline{x}) \Delta(\underline{x}, \underline{x}^*) dR = h(\underline{x}^*)$$

yields:

$$(2.23a) \quad c(\underline{x}) h(\underline{x}) + \int h(\underline{x}) q^*(\underline{x}, \underline{x}^*) d\Gamma = \int q(\underline{x}) h^*(\underline{x}, \underline{x}^*) d\Gamma$$

and

$$(2.23b) \quad c(\underline{x}) h(\underline{x}) + \int h(\underline{x}) q^*(\underline{x}, \underline{x}^*) d\Omega = \int q(\underline{x}) h^*(\underline{x}, \underline{x}^*) d\Omega$$

where $q(\underline{x}) = \partial h(\underline{x}) / \partial n$ and $q^*(\underline{x}, \underline{x}^*) = \partial h^*(\underline{x}, \underline{x}^*) / \partial n$. For positions internal to the flow domain, $c(\underline{x}^*)$ equals 2π in two dimensions and 4π in three dimensions. For positions along a one-dimensional boundary (Brebbia et al., 1984; Elsworth, 1986):

$$(2.24) \quad c(\underline{x}) = \delta$$

where δ is the interior angle at location \underline{x} . By discretizing the boundaries of the flow domain, a relationship between flow and head can be used which replaces the integral terms of Equation 2.23 with numerical summations for nodes along the boundaries (Figure 2.4):

$$(2.25a) \quad c(\underline{x}) h(\underline{x}) + \sum_{\Gamma} h(\underline{x}) q^*(\underline{x}, \underline{x}^*) = \sum_{\Gamma} q(\underline{x}) h^*(\underline{x}, \underline{x}^*)$$

and

$$(2.25b) \quad c(\underline{x}) h(\underline{x}) + \sum_{\Omega} h(\underline{x}) q^*(\underline{x}, \underline{x}^*) = \sum_{\Omega} q(\underline{x}) h^*(\underline{x}, \underline{x}^*)$$

Equation 2.25 can be written more concisely as (Brebbia, 1978):

$$(2.26) \quad \underline{A} \underline{h} = \underline{B} \underline{q}$$

where

- \underline{h} head on boundary, located at discrete boundary positions, m ;
- \underline{q} flux normal to boundary, located at discrete boundary positions, m/s ; and
- $\underline{A}, \underline{B}$ boundary integral coefficient matrices, with

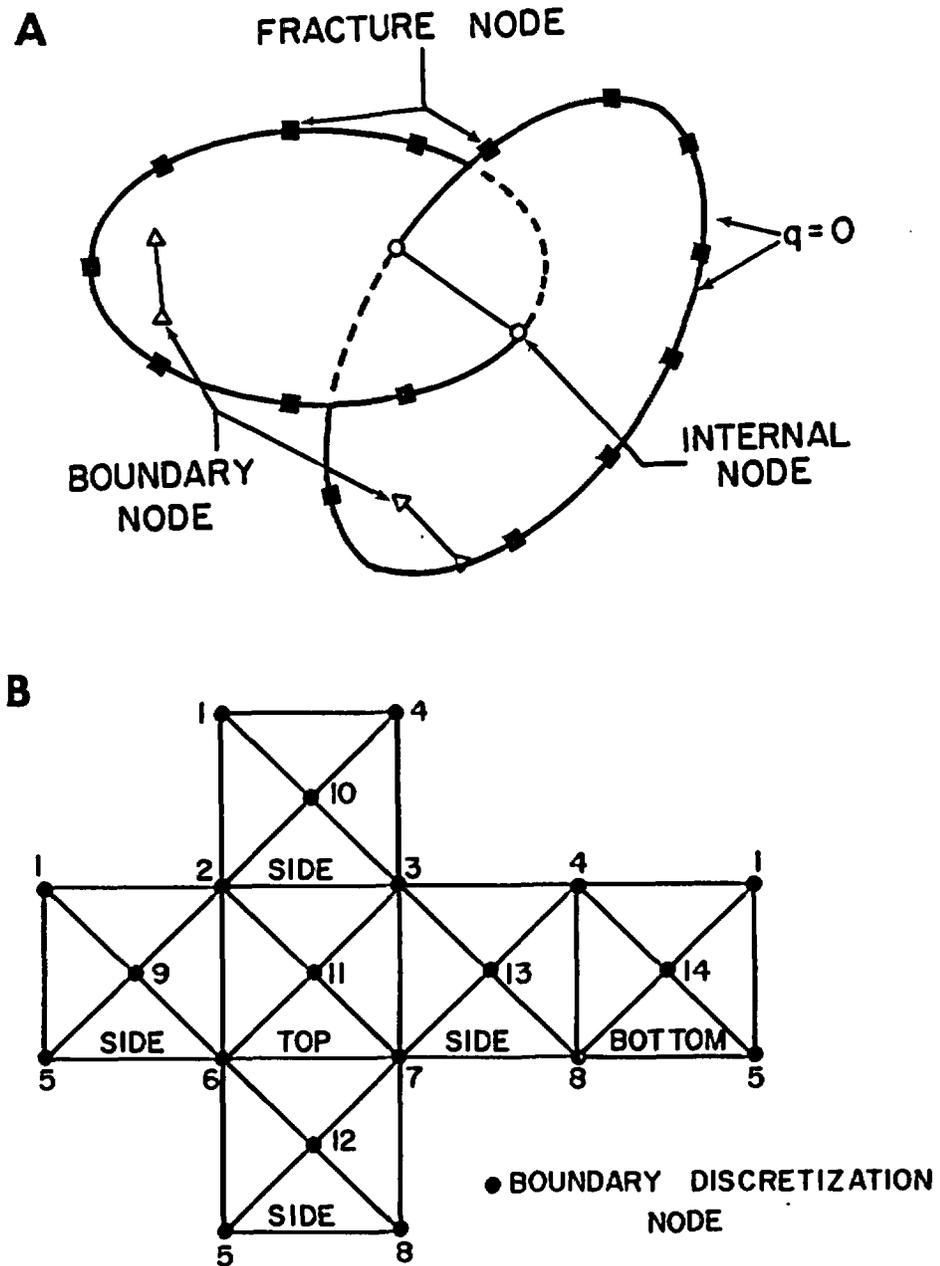


Figure 2.4: Boundary discretization schemes for two dimensional planar fractures (A) and three dimensional rock matrix (B).

$$(2.27a) \quad \underline{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3m} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mm} \end{bmatrix}$$

and

$$(2.27b) \quad \underline{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1m} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2m} \\ b_{31} & b_{32} & b_{33} & \dots & b_{3m} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ b_{m1} & b_{m2} & b_{m3} & \dots & b_{mm} \end{bmatrix}$$

where m is the number of discretization intervals along the boundary.

For constant heads and fluxes along discrete boundary segments, the elements of \underline{A} and \underline{B} can be found for two and three dimensional flow domains using Equation 2.25:

$$(2.28a) \quad a_{ij} = \int h^*(\underline{x}, \underline{x}^*) d\Gamma$$

$$(2.28b) \quad b_{ij} = \int q^*(\underline{x}, \underline{x}^*) d\Gamma$$

and

$$(2.28c) \quad a_{ij} = \int h^*(\underline{x}, \underline{x}^*) d\Omega$$

$$(2.28d) \quad b_{ij} = \int q^*(\underline{x}, \underline{x}^*) d\Omega$$

where i and j are indices corresponding to the position in the \underline{A} and \underline{B} matrices, and 2.28a and 2.28c include δ when $i=j$.

Linear interpolation of head and flux along elements will more accurately account for variations along the boundary than constant interpolation functions. Higher order interpolation will also improve the accuracy of the procedure, at the expense of numerical convenience. For a linear head and flux variation along an element, the head and flux at any point along the element is determined for two and three dimensional flow domains using, respectively (Figure 2.5):

$$(2.29a) \quad h(e) = [h_1(1-e) + h_2(1+e)] / 2$$

$$(2.29b) \quad q(e) = [q_1(1-e) + q_2(1+e)] / 2$$

and

$$(2.29c) \quad h(e_1, e_2) = h_1 e_1 + h_2 e_2 + h_3(1-e_1-e_2)$$

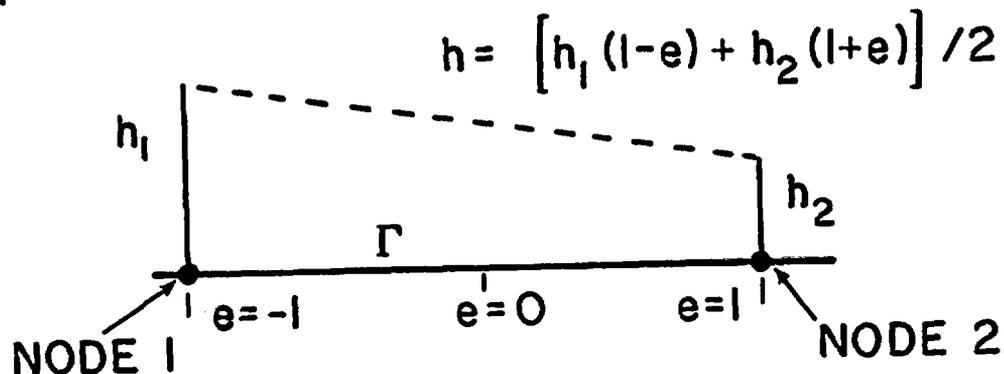
$$(2.29d) \quad q(e_1, e_2) = q_1 e_1 + q_2 e_2 + q_3(1-e_1-e_2)$$

where

- h, q head and flux at any position along a boundary line or surface element;
- e linear interpolation coordinate ($-1 < e < 1$) along boundary line element;

e_1, e_2 linear interpolation coordinates ($0 < e_1 < 1$; $0 < e_2 < 1$) along boundary surface element;
 h_1, h_2 head at endpoints of boundary line element;
 q_1, q_2 flux at endpoints of boundary line element;
 h_1, h_2, h_3 head at corners of triangular boundary surface element; and
 q_1, q_2, q_3 flux at corners of triangular boundary surface element;

A



B

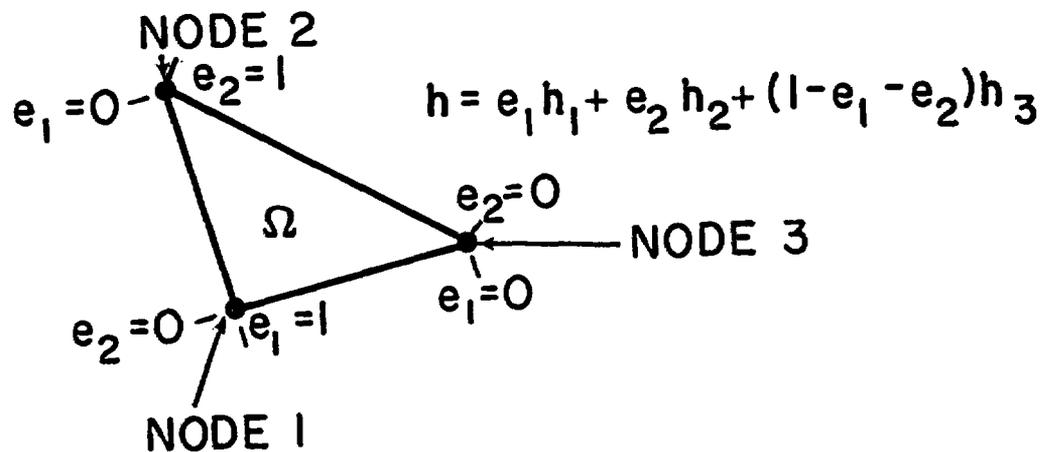


Figure 2.5: Linear interpolation functions for one (A) and two (B) dimensional boundaries.

The only remaining requirement is to determine the weighted residual functions, $h^*(\underline{x}, \underline{x}^*)$ and $q^*(\underline{x}, \underline{x}^*)$. For two and three dimensional homogeneous isotropic flow domains with constant transmissivity and hydraulic conductivity, the functions are taken to be the fundamental solutions, or, respectively:

$$(2.30a) \quad h^*(\underline{x}, \underline{x}^*) = \ln(T^{1/2}/r) / 2\pi T$$

$$(2.30b) \quad q^*(\underline{x}, \underline{x}^*) = \partial r / \partial n / 2\pi r$$

and

$$(2.60c) \quad h^*(\underline{x}, \underline{x}^*) = -1/4\pi K r$$

$$(2.30d) \quad q^*(\underline{x}, \underline{x}^*) = \partial(1/r) / \partial n / 4\pi$$

where r is the scalar distance between \underline{x} and \underline{x}^* . Huyakorn and Pinder (1983, p. 317) present analytic solutions to the integration of Equation 2.30 for two dimensional flow, as required by Equation 2.28. For conditions of three dimensional flow the integrations are performed numerically using gaussian integration over triangular areas (Cowper, 1973).

For flow between two intersecting fractures, additional internal boundary elements are introduced to represent the line of intersection between the fracture planes (Figure 2.6). From mass balance considerations and equivalence of heads along the boundary, a larger matrix can be formed which is composed of sub-matrices of flow within each fracture plus the flow across the line of intersection.

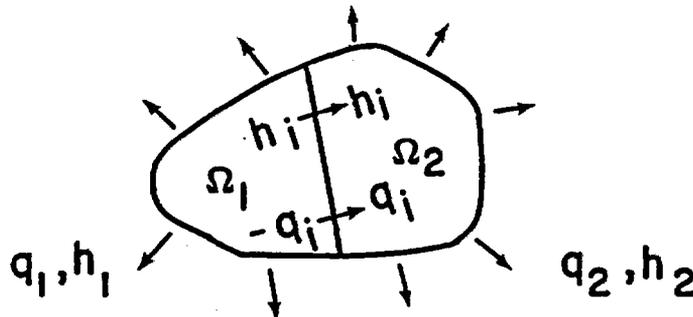


Figure 2.6: Multiple flow domain geometry.

The pure-fracture flow matrix equation in each fracture is:

$$(2.31a) \quad \begin{bmatrix} \underline{A}_1 & \underline{A}_{11} \end{bmatrix} \begin{pmatrix} q_1 \\ q_i \end{pmatrix} = \begin{bmatrix} \underline{B}_1 & \underline{B}_{11} \end{bmatrix} \begin{pmatrix} h_1 \\ h_i \end{pmatrix}$$

and

$$(2.31b) \quad \begin{bmatrix} \underline{A}_{21} & \underline{A}_2 \end{bmatrix} \begin{pmatrix} -q_i \\ q_2 \end{pmatrix} = \begin{bmatrix} \underline{B}_{21} & \underline{B}_2 \end{bmatrix} \begin{pmatrix} h_i \\ h_2 \end{pmatrix}$$

where

- $\underline{A}_1, \underline{B}_1$ boundary integral coefficient matrices between elements along exterior surfaces of fracture 1;
- $\underline{A}_2, \underline{B}_2$ boundary integral coefficient matrices between elements along exterior surfaces of fracture 2;
- $\underline{A}_{11}, \underline{B}_{11}$ boundary integral coefficient matrices between elements along exterior surfaces of fracture 1 and interfacial elements;
- $\underline{A}_{21}, \underline{B}_{21}$ boundary integral coefficient matrices between elements along exterior surfaces of fracture 2 and interfacial elements;
- h_1, q_1 flow and heads along exterior boundary of fracture 1;
- h_2, q_2 flow and heads along exterior boundary of fracture 2; and
- h_i, q_i flow and heads along interfacial boundary between fractures.

Combining Equations 2.31a and 2.31b yields:

$$(2.32) \quad \begin{bmatrix} \underline{A}_1 & \underline{A}_{11} & \underline{0} \\ \underline{0} & -\underline{A}_{21} & \underline{A}_2 \end{bmatrix} \begin{pmatrix} q_1 \\ q_i \\ q_2 \end{pmatrix} = \begin{bmatrix} \underline{B}_1 & \underline{B}_{11} & \underline{0} \\ \underline{0} & \underline{B}_{21} & \underline{B}_2 \end{bmatrix} \begin{pmatrix} h_1 \\ h_i \\ h_2 \end{pmatrix}$$

Equation 2.32 is appropriate for two intersecting fractures. Networks of intersecting fractures require more interfacial elements, with a matrix structure composed of blocks of non-zero elements, alternating with blocks of zero elements corresponding to nodes which do not lie in the same fracture plane. Equation 2.32 is solved by selecting appropriate boundary conditions at all non-interfacial nodes (either prescribed head or flux) and by reducing the resulting global set of equations to the form:

$$(2.33) \quad \underline{U} \underline{u} = \underline{V} \underline{v} = \underline{v}'$$

where

- $\underline{U}, \underline{V}$ square matrices of known boundary integral coefficients;
- \underline{u} vector of unknown boundary conditions;
- \underline{v} vector of known boundary conditions; and
- \underline{v}' vector of known coefficients and boundary conditions.

Equation 2.33 can be solved using gaussian elimination or other direct solvers.

Once the unknown head and fluxes are determined, the head at any position internal to the flow domain can be calculated using:

$$(2.34) \quad h_i = \sum_j (q_j a_{ij} - h_j b_{ij})$$

The location of streamlines can be determined by noting that streamlines correspond to constant stream function contours. Within any two dimensional flow field, the stream function is defined using the Cauchy-Riemann conditions:

$$(2.35a) \quad \partial s / \partial y = q_x = - T \partial h / \partial x$$

and

$$(2.35b) \quad \partial s / \partial x = - q_y = T \partial h / \partial y$$

where

s stream function, m^3/s ;
 x, y orthogonal cartesian coordinates in two dimensions, m ;
 q_x, q_y flux components, m^2/s ; and
 K isotropic hydraulic conductivity scalar.

For boundary surfaces, a stream function can be calculated as:

$$(2.36) \quad s_i = s_{i-1} + \int q_i \, d\tau$$

$$= s_{i-1} + l_i (q_i + q_{i-1}) / 2$$

where s_i is the stream function at position i , and l_i is the length of the boundary segment between nodes $i-1$ and i .

2.3 Applications for Steady Flow Through Discrete Fractures

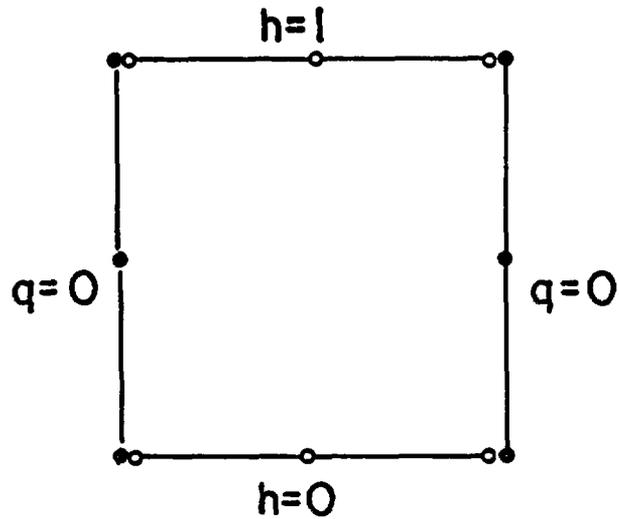
An examination of the boundary integral method is first performed for a variety of simplified steady fluid flow examples. Once confidence is gained in the ability to estimate heads and flow in simple systems, applications to more complex examples are made. The numerical precision of the method is also evaluated when analytic or other results are available. The methods are implemented in FORTRAN-77, and are presented in Appendix A as programs BIM, a general purpose, multidimensional boundary integral solver, BIM2D, which solves the two-dimensional boundary integral problem, and BIM3D, which solves the three-dimensional boundary integral problem.

2.3.1 Flow Through a Single Fracture

Steady flow through a single square fracture with a constant unit fracture transmissivity is estimated for uniform one-dimensional flow in response to a unit head gradient. Figure 2.7 illustrates the flow field configuration. Notice that nodes are located along the boundary of the flow domain, and double nodes are closely spaced at locations where the boundary conditions change rapidly. The close spacing is also required to prevent mixed boundary conditions between two nodes.

Unknown head and flow rates along the boundaries are computed using FORTRAN program BIM which incorporates linear variation in head and flow between nodes. (Documentation for program BIM is presented in Appendix A). Values of the stream function are calculated using Equation 2.34.

A) COARSE DISCRETIZATION



- Constant head nodes
- Constant flux nodes

B) FINE DISCRETIZATION

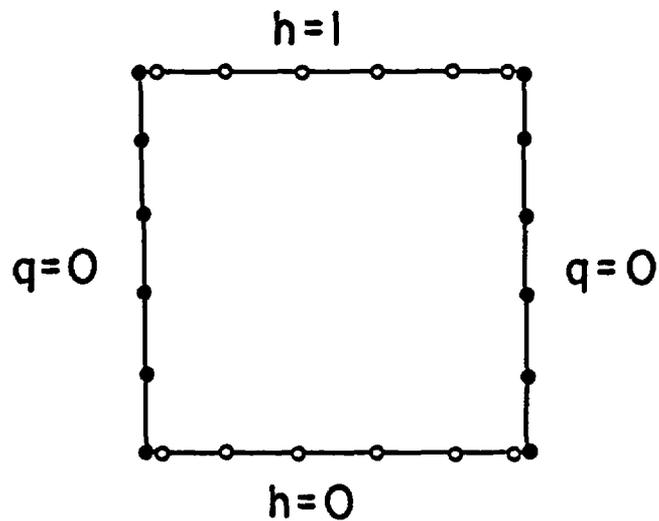


Figure 2.7: Discretization schemes for boundary integral method simulation study.

Table 2.1 presents results for a coarse discretization scheme using twelve nodes to represent the flow domain. A unit transmissivity value is used in this example. The errors in estimated unknown head and flow values are also presented in the table; the largest error in head being 0.0012 and the largest error in flow being 0.0006. Note that the error in head is small except near zones where boundary conditions change from constant flux to constant head.

 Table 2.1: Results for simulation experiment using coarse discretization interval, twelve nodes total. Fracture transmissivity is 1. Bold faced values are assigned boundary conditions.

Node x	Location y	Head h	Flow q	Stream s	Head/Flow Error
1.0000	0.0000	0.00121	0.0000	-0.00005	0.0012 (h)
1.0000	0.5000	0.50000	0.0000	-0.00005	0.0000 (h)
1.0000	1.0000	0.99879	0.0000	-0.00005	-0.0012 (h)
0.9999	1.0000	1.00000	1.0003	0.00000	0.0003 (q)
0.5000	1.0000	1.00000	1.0006	0.50011	0.0006 (q)
0.0001	1.0000	1.00000	1.0003	1.00023	0.0003 (q)
0.0000	1.0000	0.99879	0.0000	1.00028	-0.0012 (h)
0.0000	0.5000	0.50000	0.0000	1.00028	0.0000 (h)
0.0000	0.0000	0.00121	0.0000	1.00028	0.0012 (h)
0.0001	0.0000	0.00000	-1.0003	1.00023	-0.0003 (q)
0.5000	0.0000	0.00000	-1.0006	0.50011	-0.0006 (q)
0.9999	0.0000	0.00000	-1.0003	0.00000	-0.0003 (q)

 A second simulation using the coarse discretization mesh is performed with the transmissivity value increased from one to five. The results, presented in Table 2.2, show that:

- o The total flow increases five-fold;
- o The maximum error in estimated heads is equivalent to the errors in Table 2.1; and
- o The maximum error in estimated flows is increased approximately five-fold, to 0.0030.

The two simulations presented above are repeated using a finer discretization which doubles the number of nodes along the boundary from twelve to twenty-four. Table 2.3 presents the results for the finer discretization problem for a unit transmissivity and Table 2.4 presents results for a transmissivity of five. The resulting heads and flows are not appreciably different from those estimated using the coarse discretization interval. The maximum error in estimated head is reduced from 0.0012 to 0.0004, but the maximum error in estimated flux increased from 0.0030 to 0.0045 for the fracture with a transmissivity of five.

Table 2.2: Results for simulation experiment using coarse discretization interval, twelve nodes total. Fracture transmissivity is 5. Bold faced values are assigned boundary conditions.

Node Location		Head	Flow	Stream	Head/Flow
x	y	h	q	s	Error
1.0000	0.0000	0.00121	0.0000	-0.00025	0.0012 (h)
1.0000	0.5000	0.50000	0.0000	-0.00025	0.0000 (h)
1.0000	1.0000	0.99879	0.0000	-0.00025	-0.0012 (h)
0.9999	1.0000	1.00000	5.0016	0.00000	0.0016 (q)
0.5000	1.0000	1.00000	5.0030	2.50057	0.0030 (q)
0.0001	1.0000	1.00000	5.0016	5.00114	0.0016 (q)
0.0000	1.0000	0.99879	0.0000	5.00139	-0.0012 (h)
0.0000	0.5000	0.50000	0.0000	5.00139	0.0000 (h)
0.0000	0.0000	0.00121	0.0000	5.00139	0.0012 (h)
0.0001	0.0000	0.00000	-5.0016	5.00114	-0.0016 (q)
0.5000	0.0000	0.00000	-5.0030	2.50057	-0.0030 (q)
0.9999	0.0000	0.00000	-5.0016	0.00000	-0.0016 (q)

Table 2.3: Results for simulation experiment using fine discretization interval, twenty four nodes total. Fracture transmissivity is 1. Bold faced values are assigned boundary conditions.

Node Location		Head	Flow	Stream	Head/Flow
x	y	h	q	s	Error
1.0000	0.0000	0.00044	0.0000	-0.00005	0.0004 (h)
1.0000	0.2000	0.19997	0.0000	-0.00005	0.0000 (h)
1.0000	0.4000	0.39999	0.0000	-0.00005	0.0000 (h)
1.0000	0.6000	0.60001	0.0000	-0.00005	0.0000 (h)
1.0000	0.8000	0.80003	0.0000	-0.00005	0.0000 (h)
1.0000	1.0000	0.99956	0.0000	-0.00005	-0.0004 (h)
0.9999	1.0000	1.00000	1.0009	0.00000	0.0009 (q)
0.8000	1.0000	1.00000	0.9999	0.20003	-0.0001 (q)
0.6000	1.0000	1.00000	1.0000	0.40002	0.0000 (q)
0.4000	1.0000	1.00000	1.0000	0.60002	0.0000 (q)
0.2000	1.0000	1.00000	0.9999	0.80001	-0.0001 (q)
0.0001	1.0000	1.00000	1.0009	1.00004	0.0009 (q)
0.0000	1.0000	0.99956	0.0000	1.00009	-0.0004 (h)
0.0000	0.8000	0.80003	0.0000	1.00009	0.0000 (h)
0.0000	0.6000	0.60001	0.0000	1.00009	0.0000 (h)
0.0000	0.4000	0.39999	0.0000	1.00009	0.0000 (h)
0.0000	0.2000	0.19997	0.0000	1.00009	0.0000 (h)
0.0000	0.0000	0.00044	0.0000	1.00009	0.0004 (h)
0.0001	0.0000	0.00000	-1.0009	1.00004	-0.0009 (q)
0.2000	0.0000	0.00000	-0.9999	0.80001	0.0001 (q)
0.4000	0.0000	0.00000	-1.0000	0.60002	0.0000 (q)
0.6000	0.0000	0.00000	-1.0000	0.40002	0.0000 (q)
0.8000	0.0000	0.00000	-0.9999	0.20003	0.0001 (q)
0.9999	0.0000	0.00000	-1.0009	0.00000	-0.0009 (q)

Table 2.4: Results for simulation experiment using fine discretization interval, twenty four nodes total. Fracture transmissivity is 5. Bold faced values are assigned boundary conditions.

Node	Location	Head	Flow	Stream	Head/Flow
x	y	h	q	s	Error
1.0000	0.0000	0.00044	0.0000	-0.00025	0.0004 (h)
1.0000	0.2000	0.19997	0.0000	-0.00025	0.0000 (h)
1.0000	0.4000	0.39999	0.0000	-0.00025	0.0000 (h)
1.0000	0.6000	0.60001	0.0000	-0.00025	0.0000 (h)
1.0000	0.8000	0.80003	0.0000	-0.00025	0.0000 (h)
1.0000	1.0000	0.99956	0.0000	-0.00025	-0.0004 (h)
0.9999	1.0000	1.00000	5.0045	0.00000	0.0045 (q)
0.8000	1.0000	1.00000	4.9999	1.00016	-0.0001 (q)
0.6000	1.0000	1.00000	5.0000	2.00011	0.0000 (q)
0.4000	1.0000	1.00000	5.0000	3.00009	0.0000 (q)
0.2000	1.0000	1.00000	4.9999	4.00004	-0.0001 (q)
0.0001	1.0000	1.00000	5.0045	5.00020	0.0045 (q)
0.0000	1.0000	0.99956	0.0000	5.00045	-0.0004 (h)
0.0000	0.8000	0.80003	0.0000	5.00045	0.0000 (h)
0.0000	0.6000	0.60001	0.0000	5.00045	0.0000 (h)
0.0000	0.4000	0.39999	0.0000	5.00045	0.0000 (h)
0.0000	0.2000	0.19997	0.0000	5.00045	0.0000 (h)
0.0000	0.0000	0.00044	0.0000	5.00045	0.0004 (h)
0.0001	0.0000	0.00000	-5.0045	5.00020	-0.0045 (q)
0.2000	0.0000	0.00000	-4.9999	4.00004	0.0001 (q)
0.4000	0.0000	0.00000	-5.0000	3.00009	0.0000 (q)
0.6000	0.0000	0.00000	-5.0000	2.00011	0.0000 (q)
0.8000	0.0000	0.00000	-4.9999	1.00016	0.0001 (q)
0.9999	0.0000	0.00000	-5.0045	0.00000	-0.0045 (q)

2.3.2 Flow Through Serial Fractures

Flow through a series of fractures with constant transmissivity within individual fractures but with varying transmissivities between fractures is estimated for steady flow in response to a unit head gradient. Figure 2.8 illustrates the flow field configuration. The configuration is also appropriate for two zones within a single fracture having differing transmissivities.

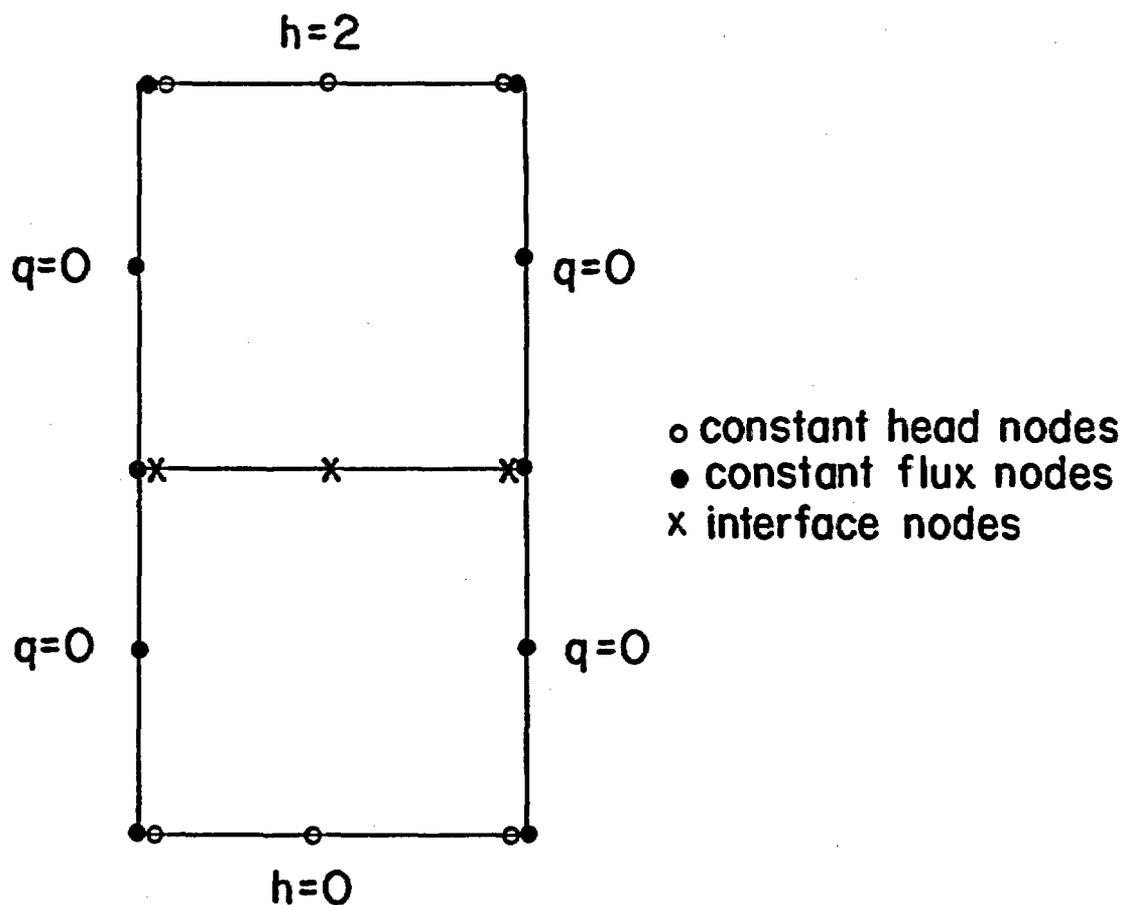


Figure 2.8: Serial fracture discretization scheme. Coarse discretization shown.

Table 2.5 presents simulation results using FORTRAN program BIM for two coarsely discretized fractures with equal unit transmissivities. Table 2.6 presents results for the same discretization but with one fracture having a transmissivity five times that of the other. Tables 2.7 and 2.8 repeat the simulation experiments using a finer discretization interval. Note that the approximation errors are small in all cases; the largest single error being 0.0020 associated the calculated head value for two coarsely discretized fractures with differing transmissivities. No measurable mass balance errors are present for any of the simulations.

 Table 2.5: Results for simulation experiment using coarse discretization interval, twelve nodes total. Transmissivity of both fractures is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.

#	Node Location x y	Head h	Flow q	Stream s	Head/Flow Error
1	1.0000 2.0000	1.99879	0.0000	-0.00005	-0.0012 (h)
	1.0000 1.5000	1.50000	0.0000	-0.00005	0.0000 (h)
	1.0000 1.0000	1.00121	0.0000	-0.00005	0.0012 (h)
	0.9999 1.0000	<u>1.00000</u>	<u>-1.0003</u>	0.00000	-0.0003 (q)
	0.5000 1.0000	<u>1.00000</u>	<u>-1.0006</u>	0.50011	-0.0006 (q)
	0.0001 1.0000	<u>1.00000</u>	<u>-1.0003</u>	1.00023	-0.0003 (q)
	0.0000 1.0000	1.00121	0.0000	1.00028	0.0012 (h)
	0.0000 1.5000	1.50000	0.0000	1.00028	0.0000 (h)
	0.0000 2.0000	1.99879	0.0000	1.00028	-0.0012 (h)
	0.0001 2.0000	2.00000	1.0003	1.00023	0.0003 (q)
	0.5000 2.0000	2.00000	1.0006	0.50011	0.0006 (q)
	0.9999 2.0000	2.00000	1.0003	0.00000	0.0003 (q)
2	1.0000 0.0000	0.00121	0.0000	-0.00005	0.0012 (h)
	1.0000 0.5000	0.50000	0.0000	-0.00005	0.0000 (h)
	1.0000 1.0000	0.99879	0.0000	-0.00005	-0.0012 (h)
	0.9999 1.0000	<u>1.00000</u>	<u>1.0003</u>	0.00000	0.0003 (q)
	0.5000 1.0000	<u>1.00000</u>	<u>1.0006</u>	0.50011	0.0006 (q)
	0.0001 1.0000	<u>1.00000</u>	<u>1.0003</u>	1.00023	0.0003 (q)
	0.0000 1.0000	0.99879	0.0000	1.00028	-0.0012 (h)
	0.0000 0.5000	0.50000	0.0000	1.00028	0.0000 (h)
	0.0000 0.0000	0.00121	0.0000	1.00028	0.0012 (h)
	0.0001 0.0000	0.00000	-1.0003	1.00023	-0.0003 (q)
	0.5000 0.0000	0.00000	-1.0006	0.50011	-0.0006 (q)
	0.9999 0.0000	0.00000	-1.0003	0.00000	-0.0003 (q)

Table 2.6: Results for simulation experiment using coarse discretization scheme, twelve nodes total. Transmissivity of first fracture is 5; of second is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.

#	Node Location		Head	Flow	Stream	Head/Flow
	x	y	h	q	s	Error
1	1.0000	2.0000	1.99960	0.0000	-0.00008	-0.0004 (h)
	1.0000	1.5000	1.83333	0.0000	-0.00008	0.0000 (h)
	1.0000	1.0000	1.66707	0.0000	-0.00008	0.0004 (h)
	0.9999	1.0000	<u>1.66667</u>	<u>-1.6672</u>	0.00000	-0.0005 (q)
	0.5000	1.0000	<u>1.66667</u>	<u>-1.6676</u>	0.83352	-0.0009 (q)
	0.0001	1.0000	<u>1.66667</u>	<u>-1.6672</u>	1.66705	-0.0005 (q)
	0.0000	1.0000	1.66707	0.0000	1.66713	0.0004 (h)
	0.0000	1.5000	1.83333	0.0000	1.66713	0.0000 (h)
	0.0000	2.0000	1.99960	0.0000	1.66713	-0.0004 (h)
	0.0001	2.0000	2.00000	1.6672	1.66705	0.0005 (q)
	0.5000	2.0000	2.00000	1.6676	0.83352	0.0009 (q)
	0.9999	2.0000	2.00000	1.6672	0.00000	0.0005 (q)
	2	1.0000	0.0000	0.00201	0.0000	-0.00008
1.0000		0.5000	0.83333	0.0000	-0.00008	0.0000 (h)
1.0000		1.0000	1.66465	0.0000	-0.00008	-0.0020 (h)
0.9999		1.0000	<u>1.66667</u>	<u>1.6672</u>	0.00000	0.0005 (q)
0.5000		1.0000	<u>1.66667</u>	<u>1.6676</u>	0.83352	0.0009 (q)
0.0001		1.0000	<u>1.66667</u>	<u>1.6672</u>	1.66705	0.0005 (q)
0.0000		1.0000	1.66465	0.0000	1.66713	-0.0020 (h)
0.0000		0.5000	0.83333	0.0000	1.66713	0.0000 (h)
0.0000		0.0000	0.00201	0.0000	1.66713	0.0020 (h)
0.0001		0.0000	0.00000	-1.0009	1.66705	-0.0009 (q)
0.5000		0.0000	0.00000	-0.9999	0.83352	0.0001 (q)
0.9999		0.0000	0.00000	-1.0009	0.00000	0.0009 (q)

Table 2.7: Results for simulation experiment using fine discretization interval, twenty four nodes total. Transmissivity of both fractures is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.

#	Node Location		Head	Flow	Stream	Head/Flow
	x	y	h	q	s	Error
1	1.0000	2.0000	1.99956	0.0000	-0.00005	-0.0004 (h)
	1.0000	1.8000	1.80003	0.0000	-0.00005	0.0000 (h)
	1.0000	1.6000	1.60001	0.0000	-0.00005	0.0000 (h)
	1.0000	1.4000	1.39999	0.0000	-0.00005	0.0000 (h)
	1.0000	1.2000	1.19997	0.0000	-0.00005	0.0000 (h)
	1.0000	1.0000	1.00044	0.0000	-0.00005	+0.0004 (h)
	0.9999	1.0000	<u>1.00000</u>	<u>-1.0009</u>	0.00000	-0.0009 (q)
	0.8000	1.0000	<u>1.00000</u>	<u>-0.9999</u>	0.20003	0.0001 (q)
	0.6000	1.0000	<u>1.00000</u>	<u>-1.0000</u>	0.40002	0.0000 (q)
	0.4000	1.0000	<u>1.00000</u>	<u>-1.0000</u>	0.60002	0.0000 (q)
	0.2000	1.0000	<u>1.00000</u>	<u>-0.9999</u>	0.80001	0.0001 (q)
	0.0001	1.0000	<u>1.00000</u>	<u>-1.0009</u>	1.00004	0.0009 (q)
	0.0000	1.0000	1.00044	0.0000	1.00009	0.0004 (h)
	0.0000	1.2000	1.19997	0.0000	1.00009	0.0000 (h)
	0.0000	1.4000	1.39999	0.0000	1.00009	0.0000 (h)
	0.0000	1.6000	1.60001	0.0000	1.00009	0.0000 (h)
	0.0000	1.8000	1.80003	0.0000	1.00009	0.0000 (h)
	0.0000	2.0000	1.99956	0.0000	1.00009	-0.0004 (h)
	0.0001	2.0000	<u>2.00000</u>	1.0009	1.00004	0.0009 (q)
	0.2000	2.0000	<u>2.00000</u>	0.9999	0.80001	-0.0001 (q)
	0.4000	2.0000	<u>2.00000</u>	1.0000	0.60002	0.0000 (q)
	0.6000	2.0000	<u>2.00000</u>	1.0000	0.40002	0.0000 (q)
	0.8000	2.0000	<u>2.00000</u>	0.9999	0.20003	-0.0001 (q)
	0.9999	2.0000	<u>2.00000</u>	1.0009	0.00000	0.0009 (q)

- - - - -
 Table 2.7 (Continued):
 - - - - -

#	Node Location		Head	Flow	Stream	Head/Flow
	x	y	h	q	s	Error
2	1.0000	0.0000	0.00044	0.0000	-0.00005	0.0004 (h)
	1.0000	0.2000	0.19997	0.0000	-0.00005	0.0000 (h)
	1.0000	0.4000	0.39999	0.0000	-0.00005	0.0000 (h)
	1.0000	0.6000	0.60001	0.0000	-0.00005	0.0000 (h)
	1.0000	0.8000	0.80003	0.0000	-0.00005	0.0000 (h)
	1.0000	1.0000	0.99956	0.0000	-0.00005	-0.0004 (h)
	0.9999	1.0000	1.00000	1.0009	0.00000	0.0009 (q)
	0.8000	1.0000	1.00000	0.9999	0.20003	-0.0001 (q)
	0.6000	1.0000	1.00000	1.0000	0.40002	0.0000 (q)
	0.4000	1.0000	1.00000	1.0000	0.60002	0.0000 (q)
	0.2000	1.0000	1.00000	0.9999	0.80001	-0.0001 (q)
	0.0001	1.0000	1.00000	1.0009	1.00004	0.0009 (q)
	0.0000	1.0000	0.99956	0.0000	1.00009	-0.0004 (h)
	0.0000	0.8000	0.80003	0.0000	1.00009	0.0000 (h)
	0.0000	0.6000	0.60001	0.0000	1.00009	0.0000 (h)
	0.0000	0.4000	0.39999	0.0000	1.00009	0.0000 (h)
	0.0000	0.2000	0.19997	0.0000	1.00009	0.0000 (h)
	0.0000	0.0000	0.00044	0.0000	1.00009	0.0004 (h)
	0.0001	0.0000	0.00000	-1.0009	1.00004	-0.0009 (q)
	0.2000	0.0000	0.00000	-0.9999	0.80001	0.0001 (q)
	0.4000	0.0000	0.00000	-1.0000	0.60002	0.0000 (q)
	0.6000	0.0000	0.00000	-1.0000	0.40002	0.0000 (q)
	0.8000	0.0000	0.00000	-0.9999	0.20003	0.0001 (q)
	0.9999	0.0000	0.00000	-1.0009	0.00000	-0.0009 (q)

Table 2.8: Results for simulation experiment using fine discretization interval, twenty four nodes total. Transmissivity of first fracture is 5; of second is 1. Bold faced values are assigned boundary conditions. Underlined values are shared between fractures.

#	Node Location		Head	Flow	Stream	Head/Flow
	x	y	h	q	s	Error
1	1.0000	2.0000	1.99985	0.0000	-0.00008	-0.0001 (h)
	1.0000	1.8000	1.93334	0.0000	-0.00008	0.0000 (h)
	1.0000	1.6000	1.86667	0.0000	-0.00008	0.0000 (h)
	1.0000	1.4000	1.80000	0.0000	-0.00008	0.0000 (h)
	1.0000	1.2000	1.73332	0.0000	-0.00008	0.0000 (h)
	1.0000	1.0000	1.66681	0.0000	-0.00008	0.0001 (h)
	0.9999	1.0000	<u>1.66667</u>	<u>-1.6682</u>	0.00000	-0.0015 (q)
	0.8000	1.0000	<u>1.66667</u>	<u>-1.6665</u>	0.33339	0.0002 (q)
	0.6000	1.0000	<u>1.66667</u>	<u>-1.6666</u>	0.66670	0.0001 (q)
	0.4000	1.0000	<u>1.66667</u>	<u>-1.6666</u>	1.00003	0.0001 (q)
	0.2000	1.0000	<u>1.66667</u>	<u>-1.6665</u>	1.33335	0.0002 (q)
	0.0001	1.0000	<u>1.66667</u>	<u>-1.6682</u>	1.66673	-0.0015 (q)
	0.0000	1.0000	1.66681	0.0000	1.66682	0.0001 (h)
	0.0000	1.2000	1.73332	0.0000	1.66682	0.0000 (h)
	0.0000	1.4000	1.80000	0.0000	1.66682	0.0000 (h)
	0.0000	1.6000	1.86667	0.0000	1.66682	0.0000 (h)
	0.0000	1.8000	1.93334	0.0000	1.66682	0.0000 (h)
	0.0000	2.0000	1.99985	0.0000	1.66682	-0.0001 (h)
	0.0001	2.0000	2.00000	1.6682	1.66673	0.0015 (q)
	0.2000	2.0000	2.00000	1.6665	1.33335	-0.0002 (q)
	0.4000	2.0000	2.00000	1.6666	1.00003	-0.0001 (q)
	0.6000	2.0000	2.00000	1.6666	0.66670	-0.0001 (q)
	0.8000	2.0000	2.00000	1.6665	0.33339	-0.0002 (q)
	0.9999	2.0000	2.00000	1.6682	0.00000	0.0015 (q)

Table 2.8 (Continued):

#	Node Location		Head	Flow	Stream	Head/Flow
	x	y	h	q	s	Error
2	1.0000	0.0000	0.00073	0.0000	-0.00008	0.0007 (h)
	1.0000	0.2000	0.33328	0.0000	-0.00008	0.0000 (h)
	1.0000	0.4000	0.66665	0.0000	-0.00008	0.0000 (h)
	1.0000	0.6000	1.00002	0.0000	-0.00008	0.0000 (h)
	1.0000	0.8000	1.33339	0.0000	-0.00008	0.0000 (h)
	1.0000	1.0000	1.66594	0.0000	-0.00008	-0.0007 (h)
	0.9999	1.0000	1.66667	1.6682	0.00000	0.0015 (q)
	0.8000	1.0000	1.66667	1.6665	0.33339	-0.0002 (q)
	0.6000	1.0000	1.66667	1.6666	0.66670	-0.0001 (q)
	0.4000	1.0000	1.66667	1.6666	1.00003	-0.0001 (q)
	0.2000	1.0000	1.66667	1.6665	1.33335	-0.0002 (q)
	0.0001	1.0000	1.66667	1.6682	1.66673	0.0015 (q)
	0.0000	1.0000	1.66594	0.0000	1.66682	-0.0007 (h)
	0.0000	0.8000	1.33339	0.0000	1.66682	0.0000 (h)
	0.0000	0.6000	1.00002	0.0000	1.66682	0.0000 (h)
	0.0000	0.4000	0.66665	0.0000	1.66682	0.0000 (h)
	0.0000	0.2000	0.33328	0.0000	1.66682	0.0000 (h)
	0.0000	0.0000	0.00073	0.0000	1.66682	0.0007 (h)
	0.0001	0.0000	0.00000	-1.0009	1.66673	-0.0009 (q)
	0.2000	0.0000	0.00000	-0.9999	1.33335	-0.0001 (q)
	0.4000	0.0000	0.00000	-1.0000	1.00003	0.0000 (q)
	0.6000	0.0000	0.00000	-1.0000	0.66670	0.0000 (q)
	0.8000	0.0000	0.00000	-0.9999	0.33339	-0.0001 (q)
	0.9999	0.0000	0.00000	-1.0009	0.00000	-0.0009 (q)

2.2.3 Flow Through Fractures with Internal Intersections

Flow experiments are performed to investigate the effects of fractures intersecting along line segments internal to individual fractures. In all, four experiments are performed with three scenarios per experiment. In the first experiment, two parallel unit length constant head boundaries are aligned parallel to each at a unit distance apart (Figure 2.9a). A unit difference in potential is maintained between the two boundaries. The three scenarios examined are:

- o An exterior no flow boundary is first placed so that both of the constant head boundaries intersect but do not cross the no flow boundary;
- o A scenario with the length of the no flow boundaries being three times as long as the constant head boundaries; and
- o A scenario with the no flow boundary length being five times as long as the constant head boundaries.

In the second experiment, the three scenarios are again repeated, only this time the unit length constant head boundaries are now perpendicular to each other with the midpoints of the boundaries located a unit distance apart. Figure 2.9b illustrates the geometry of the boundaries. For the third experiment, fractures of disparate lengths are compared. Again the boundaries are successively displaced away from the flow region, as shown in Figure 2.9c. The final experiment uses multiple sources within an individual fracture plane. In this case three unit length sources are aligned parallel to each other. Each source is assigned a potential of $(-1,0,1)$ respectively. Figure 2.9d presents the flow geometry.

Table 2.9 presents the flow test results using program BIM for the simulation experiments. The table shows that as the outer fracture boundaries are moved away from the fracture-fracture intersections an increase in flow rates is observed. Analytic solutions for all of the experiments are not available. Analytic solutions for some of the experiments are presented in the table which show that the simulation results do not substantially deviate from the exact solution.

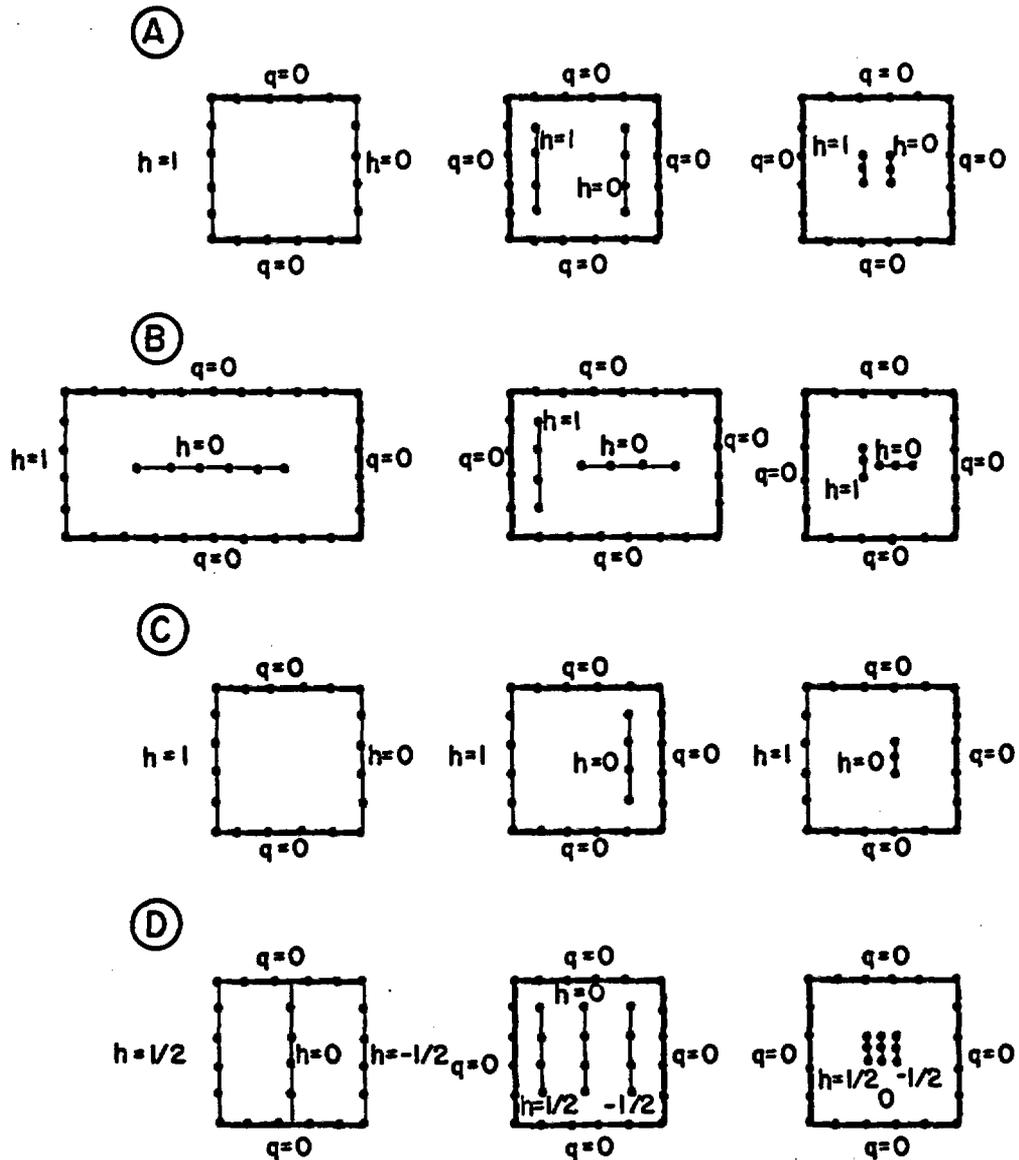


Figure 2.9: Four experiments conducted to evaluate the effect of fracture lengths, orientations and distances of boundaries. Views are plan showing exterior no flow boundaries and lines of intersection with other fractures for two parallel fractures (A), two perpendicular fractures (B), two parallel fractures of disparate lengths (C), and three parallel fractures (D).

Table 2.9: Calculated and exact flow rates for variable geometries within the plane of a fracture.

Flow System	Computed	Exact
Two Parallel Sources:		
Fully-bisecting	1.01	1.00
Near-boundary	2.11	-
Far-boundary	2.42	-
Two Perpendicular Sources:		
Fully-bisecting	1.54	-
Near-boundary	2.77	-
Far-boundary	2.80	-
Two Disparate Length Sources:		
(5:5) ratio	1.01	1.00
(5:3) ratio	1.56	-
(5:1) ratio	3.21	-
Three Parallel Sources:		
Fully-bisecting	0.33	0.33
Near-boundary	0.50	-
Far-boundary	0.50	-

2.3.4 Influence of Fracture Density and Spacing

Additional simulation experiments are performed to evaluate the increase in global hydraulic conductivity as a function of fracture density and orientation. (The global or network hydraulic conductivity of a rock block is the equivalent flow rate per unit head gradient per unit rock area.) Fractures are synthetically generated within a rock matrix by defining fracture centers, orientations, and areal extent for each fracture. The fractures are organized into sets with common orientations within each set. Intersections between generated fractures are then found. The resulting network of fractures and fracture intersections are assembled within a specified sample volume such that fractures and intersections exterior to the sample volume are removed. Exterior portions of fractures and intersections that lie partly inside and partly outside the sample volume are truncated at the boundary.

Table 2.10 presents simulation results which indicates that the global hydraulic conductivity increases linearly as a function of fracture density, d_f , and fracture transmissivity, T_f , for sets of infinite fractures which lie parallel to each other.

Table 2.10: Effect of fracture density and fracture transmissivity on global hydraulic conductivity. A unit vertical gradient was applied across two ends of a (50 x 50 x 50) m³ rock cube. Fracture density is length of fractures (m) per (50 x 50) m² rock cube face.

Fracture Density (1/m)	Fracture Transmissivity (m ² /s)	Global Hydraulic Conductivity Calculated (m/s)	Global Hydraulic Conductivity Simulated (m/s)	Error (Percent)
0.001	0.01	0.0000100	0.0000104	4.
0.001	0.10	0.0001000	0.0001000	0.0
0.001	1.00	0.0010000	0.0010008	0.08
0.01	0.01	0.0001000	0.0001000	0.0
0.01	0.10	0.0010000	0.0010008	0.08
0.01	1.00	0.0100000	0.0100088	0.088
0.10	0.01	0.0010000	0.0010008	0.08
0.10	0.10	0.0100000	0.0100088	0.088
0.10	1.00	0.1000000	0.1000864	0.0864

The flow through a network of fractures, Q, can be calculated using:

$$(2.37) \quad Q = A T_f d_f [I - a^T a] i$$

where

- A flow area perpendicular to flow direction;
- I identity matrix;
- a vector of direction cosines; and
- i hydraulic gradient.

The inferred global hydraulic conductivity is:

$$(2.38) \quad K = Q / A i = T_f d_f [I - a^T a]$$

For a unit hydraulic gradient with a fracture set oriented parallel to the gradient and also perpendicular to an injection surface, the estimated global hydraulic conductivity is:

$$(2.39) \quad K = T_f d_f$$

Simulated global hydraulic conductivity values presented in Table 2.10 agree closely with values calculated from Equation 2.37. The greatest single error is four percent corresponding to the value with the least number of significant decimal places. The calculated error for the case with the greatest significance is 0.0864 percent.

2.3.5 Influence of Fracture Transmissivity and Length Correlations

The influence of fracture transmissivity and length on estimated global hydraulic conductivities is also examined. In particular, the potential for long fractures with high transmissivities to greatly influence the estimated hydraulic conductivity is investigated. Program FRACGEN was used to generate four fracture networks in which the fracture lengths are both correlated and uncorrelated with fracture transmissivity. Input parameters consist of:

- o A fracture generating volume (assigned a value of $100 \times 100 \times 100 \text{ m}^3$); and
- o Three fracture sets, each with an arbitrary fracture transmissivity of $10 \times 10^{-6} \text{ m}^2/\text{s}$, and with:
 - † Set one having 48 fractures within the generating volume, a mean length of 10 m, a vertical orientation, and an east-west strike;
 - † Set two having twelve fractures within the generating volume, a mean length of 20 m, and a horizontal orientation; and
 - † Set three having three fractures within the generating volume, a mean length of 40 m, a vertical orientation and a north-south strike.

Program BIM2D was used to solve for total fluid flow through a ($50 \times 50 \times 50 \text{ m}^3$) sample volume within the global generating volume for a unit gradient in the vertical direction. Figure 2.10 illustrates in two dimensions the three dimensional fracture networks. Table 2.11 presents simulation results using program BIM2D for both uncorrelated and correlated parameters.

The simulation results indicate that an increase in the estimated global hydraulic conductivity results from an increase in correlation for both a configuration where one fracture connects two boundaries, as shown in the first and second networks, and also when flow must pass through three fractures between surfaces, as shown in the third and fourth networks. The dependence of fracture connectivity on fracture length is evident in these realizations, in that the fractures which provide the opportunity for flow are always those which are longer than the mean fracture length. As a result, a high correlation between fracture length and transmissivity results in increases in network hydraulic conductivity.

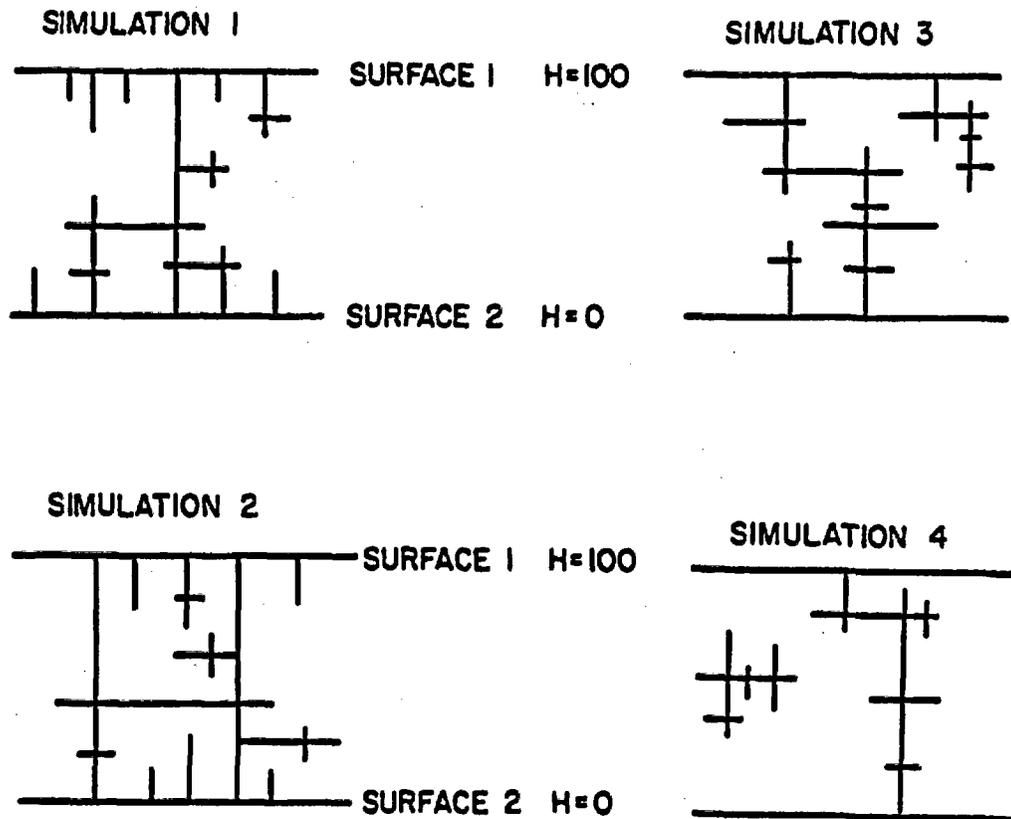


Figure 2.10: Four realizations of fracture networks used to evaluate network hydraulic conductivity.

Table 2.11: Comparison of global hydraulic conductivity calculated using the boundary integral model for uncorrelated and perfectly correlated lengths and transmissivities.

Network Number	Uncorrelated	Correlated	Ratio
1	0.246e-8	0.276e-7	11.2
2	0.318e-8	0.327e-7	10.3
3	0.666e-8	0.278e-7	4.2
4	0.529e-8	0.230e-6	43.5
mean	0.440e-8	0.795e-7	18.1

2.4 Coupled Fracture-Matrix Flow

For many geologic settings the rock matrix can not be assumed to be impermeable. A sensitivity analysis is implemented using the boundary integral method to determine the relative contribution to total liquid flow through a rock block by both flow through the rock matrix and by flow through discrete fractures embedded within the rock matrix. To develop a methodology for simulating fluid flow through a porous rock matrix with embedded fractures, a three dimensional boundary integral method is examined for its accuracy. The method allows the discretization of the outer surface of a rock volume, along with internal fractures. Simultaneous fluid flow between boundaries on the rock surface and through fractures is determined by coupling source and sink terms along the interface between the fracture walls and the rock matrix. Fracture flow which incorporates flow into and out of the fracture from the matrix across the walls of the fracture is governed by the Poisson equation:

$$(2.40) \quad V^*(q_f) + q_m = V^*(\underline{K}Vh) + q_m = 0$$

where q_m is the areal source term, $m^3/s/m^2$, which accounts for the net flow into the fracture from the matrix through both walls of the fracture. Flow between the fracture and the matrix is spatially variable over the fracture. The boundary integral method can be employed to evaluate the magnitude of the flow and the head distribution within the fracture and surrounding matrix by reforming the weighted residual statement of Equation 2.19a as:

$$(2.41) \quad \int [h^*(\underline{x}, \underline{x}^*) v^2 h(\underline{x})] d\Omega = \int [h^*(\underline{x}, \underline{x}^*) q_m(\underline{x})] d\Omega$$

The fracture flow matrix equation thus becomes:

$$(2.42) \quad \underline{F} + \underline{A} \underline{h} = \underline{B} \underline{q}$$

where \underline{F} is the contribution to fluid flow from the rock matrix into the fracture, integrated over all triangular fracture elements on the surface of the fracture (Figure 2.11). The value of \underline{F} is calculated from:

$$(2.43) \quad \underline{F} = \sum_i [\sum_j w_j (q_m h^*(\underline{x}, \underline{x}^*))_j] A_i$$

where

- i index over all fracture area elements;
- j counter over all numerical integration points on the fracture area element;
- w_j gaussian integration weighting factor for integration points (from Cowper, 22); and
- A_i area of fracture element.

Flow through the matrix is solved using the three dimensional boundary integral equation:

$$(2.44) \quad A_m h_m = b_m q_m$$

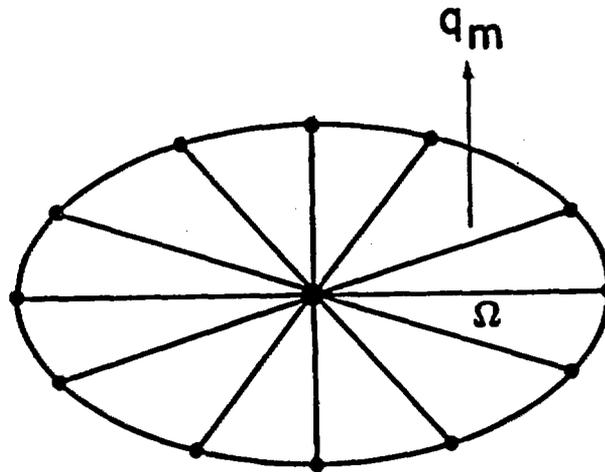


Figure 2.11: Fracture surface discretization geometry showing net flux term, q_m , representing flow between the fracture and the matrix.

where the subscripts denote matrix flow components calculated using three dimensional formulation. Equations 2.42 and 2.44 are coupled using mass balances between interface nodes and the equivalence of heads for fracture and matrix nodes.

2.4.1 Two Dimensional Porous Medium Application

Before proceeding with an analysis of flow through two dimensional planar fractures embedded within a three dimensional porous matrix, it is instructive to examine a simplified example of flow through a one dimensional linear fracture embedded within a two dimensional porous medium. In this example a square with unit length sides is constructed so that two opposing sides are assigned no flow boundary conditions, the top surface is assigned a unit hydraulic head, and the bottom surface is assigned a hydraulic head of zero. The resulting global gradient is only in the vertical direction, and the magnitude of this global vertical gradient, J_z , is one. The matrix permeability, k_m , of the interior flow region is assigned a unit value and the matrix region is assigned a width, b_m , of 0.99. The center of a fracture with an arbitrary aperture, b_f , of 0.01 is placed in the center of the unit square and length, rotation and fracture permeability, k_f , parameters are assigned.

Fracture orientation is allowed to vary in such a manner that the fracture is either parallel, perpendicular, or diagonal to the direction of flow. Also, fracture length is allowed to vary from fully to partially dividing the flow domain. Finally, k_f/k_m is varied from 10^{-5} to 10^5 . Table 2.12 summarizes the flow test parameters.

 Table 2.12: Fracture parameters for flow experiments.

Simulation:	1	2	3	4	5	6
Length:	1.	1.4	1.	0.5	0.7	0.5
Rotation:	0°	45°	90°	0°	45°	90°

ALL SIMULATIONS:

Fracture center is located at center of flow domain.
 Fracture aperture, b_f , is 0.01. Matrix width, b_m , is 0.99.
 Fracture permeability-matrix permeability ratios of
 ($10^{-5}, 10^{-3}, 10^{-1}, 1, 10^1, 10^3, 10^5$) used for each simulation.

 To discretize the flow domain, nodes are placed around the periphery of the unit square, as well as around the rim of the linear fracture. The location of nodes along the perimeter of the matrix domain are selected so that linear head and flow changes are reproduced. (Errors may be introduced if the true head and flow are varying in a nonlinear manner and a linear interpolation function is used.) From experience, it has been determined that nodes must be placed at corners and near where fractures and matrix boundaries are closest to each other. Figure 2.12 illustrates the flow domain geometry and nodal discretization schemes.

The number of nodes required to discretize the fracture surface is determined by comparing calculated flows with flows obtained using an analytic solution for a known problem. In this case, a flow regime with a fracture perpendicular to the direction of the gradient completely bisects the flow domain. The aspect ratio of the fracture is the proportion between the distance between nodes to the width of the flow domain (Figure 2.13). For an aspect ratio near one, good accuracy is expected, with decreasing accuracy as the ratio departs from unity. Table 2.13 presents simulation results for two dimensional flow error analysis. It can be observed that the accuracy of the procedure decreases rapidly when the aspect ratio is greater than 10:1. An aspect ratio less than 10:1 is used in the simulation experiments which follow.

 Table 2.13: Two dimensional precision of boundary integral method as a function of aspect ratio, defined here as the ratio of the distance between nodes to the width of the flow domain.

Aspect Ratio	1:1	5:1	10:1	20:1	25:1
2-D Error (percent)	0.000	0.003	0.20	5.9	25.4

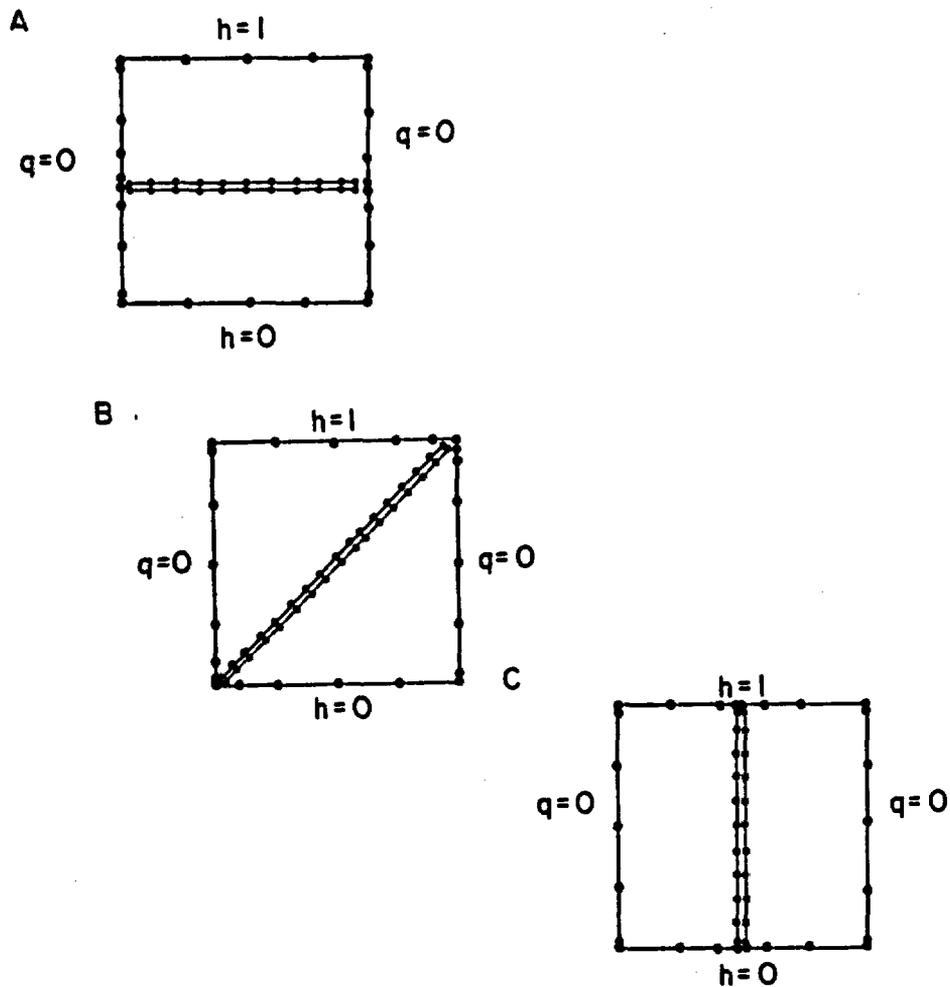


Figure 2.12: Two dimensional flow geometry showing no flow and constant head boundaries for fractures perpendicular (A), diagonal (B) and parallel (C) to the direction of the mean head gradient.

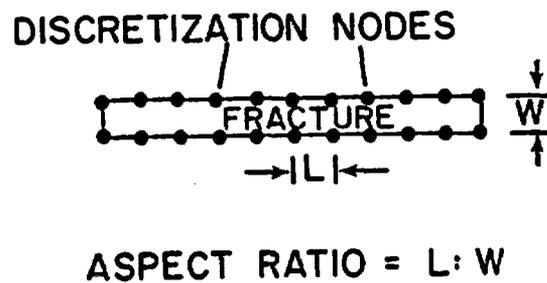


Figure 2.13: Definition of aspect ratio as proportion of distance between nodes to width of fracture.

Once the geometry, boundary conditions, and the number and location of boundary nodes have been established, simulations can be performed to determine the global vertical hydraulic conductivity, K_z , defined as the measured flux through the domain, Q , divided by the unit area and also the unit hydraulic gradient imposed across the domain, J_z :

$$(2.45) \quad K_z = Q / (A J_z)$$

This measure of the hydraulic conductivity is consistent with the interpretation of a flow test in which heads and gradients within the interior region can not be examined. The global hydraulic conductivity can also be related to k_z by noting:

$$(2.46) \quad K_z = k_z \gamma / \mu$$

where γ is the specific weight of water, and μ is the dynamic viscosity of water.

Figure 2.14 presents two dimensional simulation results between the global permeability in the vertical direction, k_z , to the ratio of fracture versus matrix permeabilities, k_f/k_m . It can be concluded that for flow simulations with a fracture which fully bisects the flow domain, (i.e., simulations 1, 2, and 3) that:

o K_z is directly related to k_m when:

† The fracture fully divides the flow domain perpendicular to the direction of flow, and k_f is high; and

† The fracture fully divides the flow domain parallel to the direction of flow, and the k_f is low.

k_z can be calculated from the harmonic average of k_f and k_m when the fracture is perpendicular to the direction of flow. The harmonic average, k_h , is calculated using:

$$(2.47a) \quad k_h = (b_f + b_m) / (b_f/k_f + b_m/k_m)$$

o K_z is directly related to k_f when:

† The fracture fully divides the flow domain perpendicular to the direction of flow, and k_f is low; and

† The fracture fully divides the flow domain parallel to the direction of flow, and k_f is high.

k_z can be calculated from the arithmetic average of k_f and k_m when the fracture is parallel to the direction of flow. The arithmetic average, k_a , is calculated using:

$$(2.47b) \quad k_a = (b_f k_f + b_m k_m) / (b_f + b_m)$$

o K_z is directly related to k_f when the fracture fully divides the flow domain diagonal to the direction of flow. k_z can be calculated from the geometric average of k_h and k_a when the fracture is

TWO DIMENSIONAL FLOW SIMULATION RESULTS

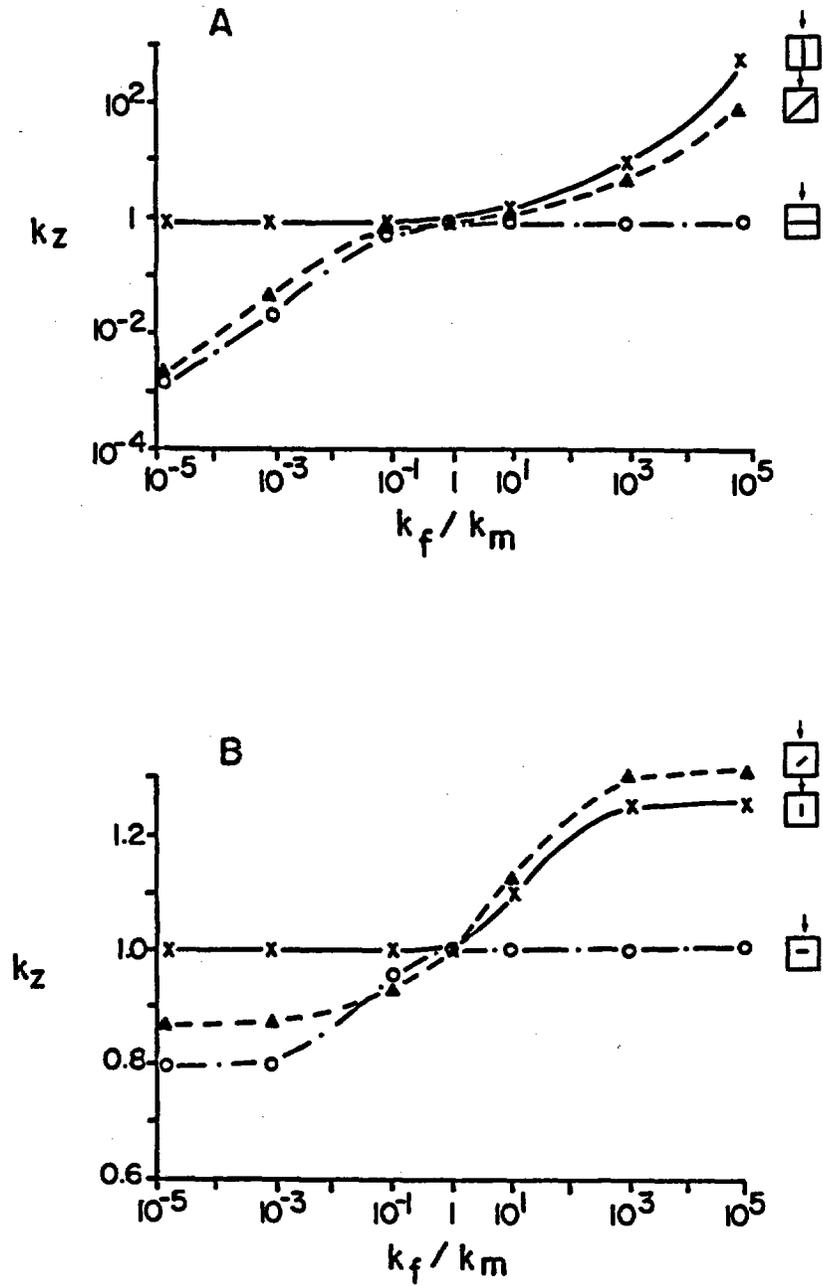


Figure 2.14: Results of simulations performed using the two dimensional flow geometry for fractures fully (A) and partially (B) dividing the flow domain.

diagonal to the direction of flow. The geometric average, k_g , is calculated using (Bear, 1979):

$$(2.47c) \quad k_g = 1 / (\cos^2 a / k_h + \sin^2 a / k_a)$$

where a is fracture orientation ($a = 0$ for a perpendicular fracture).

For a fracture with a length one-half of the flow width (i.e., simulations 4, 5, and 6) the effects of fracture orientation are similar to those for a fully divided flow domain, but the effects are reduced.

2.4.2 Three Dimensional Porous Medium Application

To evaluate the boundary integral procedure for its ability to estimate three dimensional flow properties of a fractured medium, a simplified three dimensional flow domain is investigated by defining a cube with unit length sides, four of which are no flow boundaries and the top and bottom are assigned a constant head of one and zero, respectively (Figure 2.15). Again, the matrix permeability is assigned a value of unity and the center of a fracture is located at the center of the flow domain and the fracture is allowed to vary in length and orientation, as described in Table 2.12.

An analysis of the effect of nodal density on simulation error is performed in a manner similar to that performed for two dimensional flow. The aspect ratio is again defined as the distance between nodes divided by the thickness of the flow domain. Results for three dimensional aspect ratio error analyses are presented in Table 2.14. An aspect ratio less than 10:1 (corresponding to an error of less than five percent) is used in the simulation experiments described below.

Results for experiments conducted using the same conditions (i.e., simulations 1 through 6; presented previously in Table 2.12) as for the two dimensional experiments are presented in Figure 2.16. Table 2.15 compares simulation and analytic results using Equation 2.47. Note that the simulation results compare favorably except for the case when the fractures are placed diagonally across the flow domain and a large fracture/matrix permeability ratio is used. In this case, the error is attributed to the positioning of one end of the fracture directly upon the upper flow surface, with no intervening matrix. By providing a direct connection between the upper flow boundary and the fracture, an estimated global permeability higher than the theoretical value is to be expected.

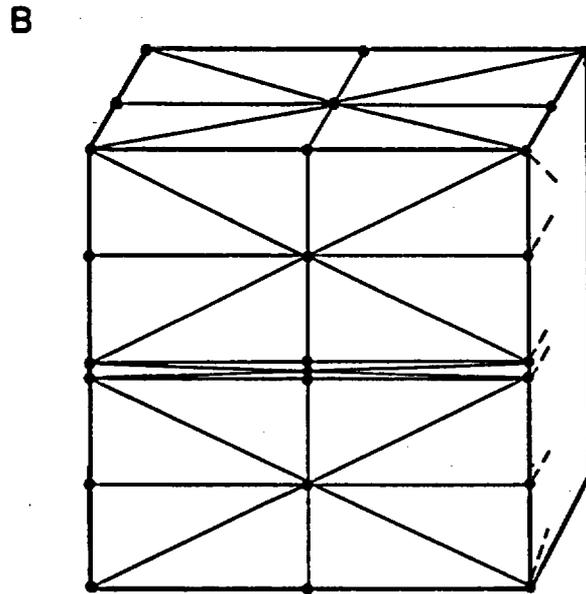
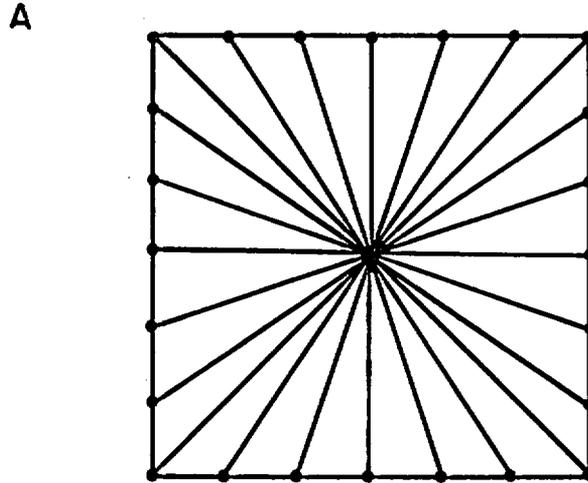


Figure 2.15: Three dimensional flow geometry showing fracture (A) and matrix (B) boundary surface discretization strategies.

Table 2.14: Three dimensional precision of boundary integral method as a function of aspect ratio, defined here as the ratio of the distance between nodes to the width of the flow domain.

Aspect Ratio	1:1	5:1	10:1	20:1	25:1
3-D Error (percent)	0.000	0.484	4.70	7.2	17.8

Table 2.15: Analytic and simulation results for variable fracture-matrix permeability ratios using three dimensional boundary integral method. Fully bisecting fractures.

k_f/k_m	Perpendicular		Parallel		Diagonal	
	k_h	k_z	k_a	k_z	k_g	k_z
10^{-5}	0.001	0.0018	0.99	0.98	0.002	0.003
10^{-3}	0.091	0.023	0.99	0.99	0.17	0.07
10^{-1}	0.92	0.90	0.99	0.99	0.95	0.96
10^0	1.	1.	1.	1.	1.	0.99
10^1	1.01	1.02	1.09	1.20	1.05	1.07
10^3	1.01	1.02	10.1	9.79	1.84	6.23
10^5	1.01	1.02	1000.	1013.	2.02	12.40

THREE-DIMENSIONAL FLOW SIMULATION RESULTS

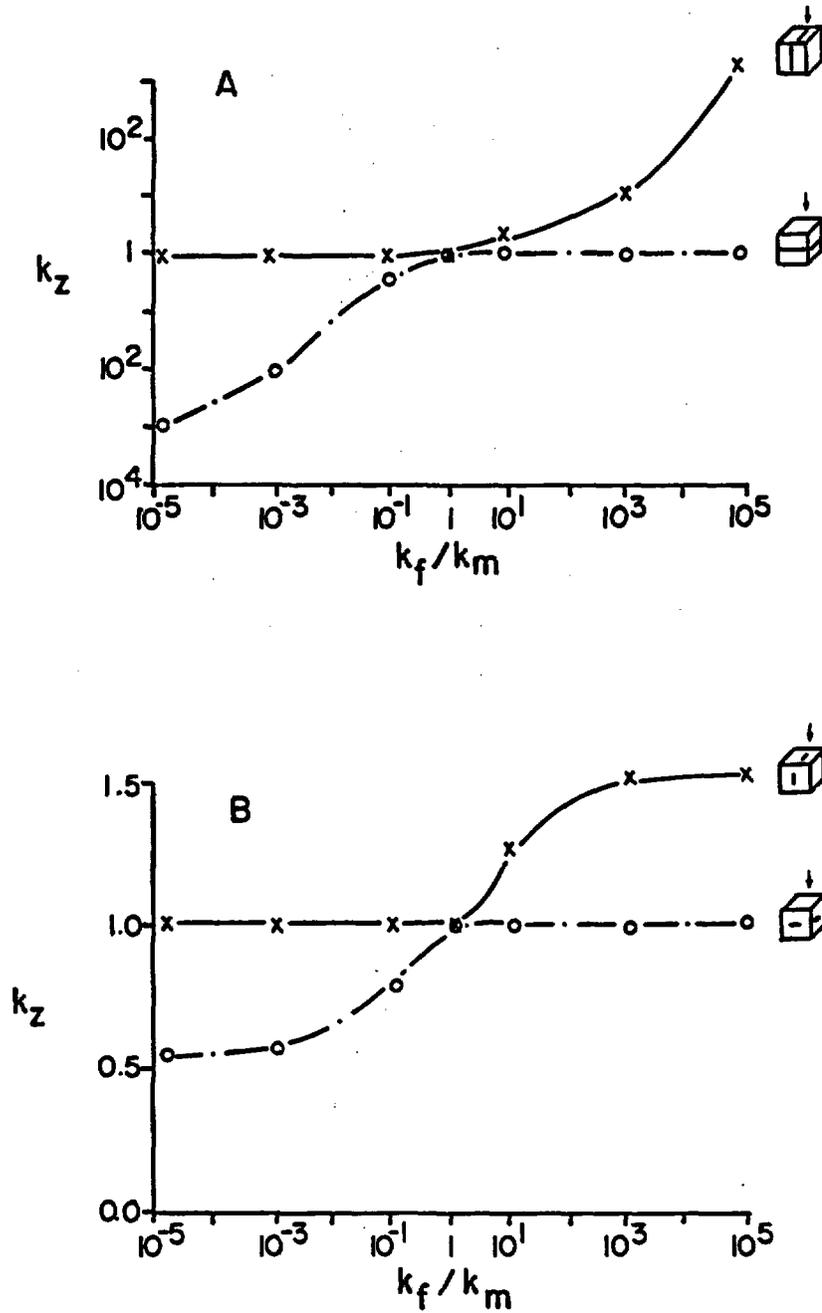


Figure 2.16: Results of simulations performed using the three dimensional flow geometry for fractures fully (A) and partially (B) dividing the flow domain.

2.4.3 Summary of Three Dimensional Coupled Fracture-Matrix Flow

Hydraulic flow properties of a combined fracture-matrix system are characterized using a proposed three dimensional boundary integral procedure. The procedure accounts for flow through fractures embedded within a porous matrix. The only discretization required is along the boundaries of the fractures and the exterior surface of the matrix block being examined. An average unit vertical head gradient is induced across the block and the resultant flow is used to estimate the global vertical hydraulic conductivity, K_z .

The effect of variations in fracture permeability on K_z are simulated by allowing the fracture permeability to vary from 10^{-5} to 10^5 relative to the matrix permeability. The variations in fracture permeability can result from geochemical processes such as dissolution and precipitation of minerals, or the result of variable water saturation levels within the fracture. It is shown for both two and three dimensional flow that the vertical global hydraulic conductivity is a function of the length of intervening fractures and their orientation.

When fractures are of infinite areal extent and the permeability/matrix permeability ratio, k_f/k_m , is large, K_z is directly related to k_f when all the fractures are oriented vertically, but is unaffected by the fractures if they are all oriented horizontally. For small k_f/k_m ratios, the roles of vertical and horizontal fractures reverse, with horizontal fractures controlling the magnitude of K_z , and vertical fractures having no effect.

CHAPTER 3

NUMERICAL SIMULATION OF STEADY FLOW THROUGH VARIABLY SATURATED FRACTURES

Conceptual models for liquid flow through discrete fractures under conditions of variable fluid saturation have been proposed by Montazer and Wilson (1984), Wang and Narasimhan (1985), Tsang and Pruess (1987) and Peters and Klavetter (1988). Their models quantify the variation in fracture permeability as a function of variable fracture water content. The resultant functions are then used to assign macroscopic parameters in regional models of flow through a fractured rock medium by assuming a uniform permeability over discretization intervals which may be larger than the size of individual fractures (Figure 3.1).

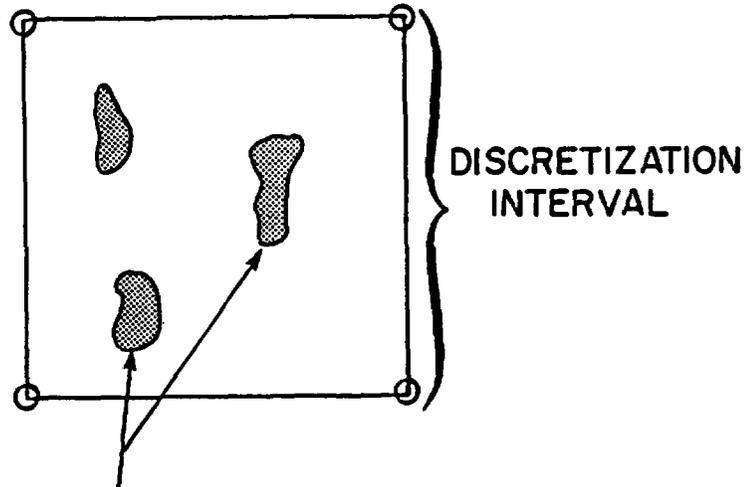
The purpose of the studies presented in this chapter is to present a formulation which provides for variable fluid potential within individual fractures at a scale which allows for a finer resolution than previous models. The methodology accounts for large variations in water contents within vertical fractures, as well as demonstrates that regions within an individual fracture may be under positive pressure while other regions are under negative pressure. The macro-variability of water content within fractures is investigated by using a free surface to represent a discrete air-water interface. Applications using this representation have been presented by Bear and Dagan (1964), Neuman and Witherspoon (1970), Pinder and Gray (1977), Liu et al. (1981), Huyakorn and Pinder (1983), and Liggett and Liu (1983).

This chapter first presents the theory used to generate synthetic moisture characteristic and unsaturated transmissivity functions for fractures. The synthetic functions are required due to the lack of available data which can be used to specify actual functions. A subsequent section presents the free surface formulation and the boundary method procedure developed here to solve for the air-water interface in fractures. A final section presents simulation results for flow through fractures of arbitrary orientation and fluid saturation.

3.1 Generation of Synthetic Moisture Characteristic and Unsaturated Transmissivity Functions for Discrete Fractures

The influence of fluid saturation and potential on fracture transmissivity has not been quantified using laboratory or field testing methods. Tsang and Pruess (1987) present a hypothetical relationship between fracture permeability and relative saturation, which is closely related to a form proposed for soils (Van Genuchten, 1978, 1980). While this parametric form has the advantage of application to a wide range of curves, the parameters in the model are not amenable to field testing techniques. Existing research activities (Evans and Rasmussen, 1988) are directed toward generating representative moisture characteristic and unsaturated transmissivity functions for discrete fractures using laboratory methods. Until such time as realistic functions become available, synthetic functions must be generated based on relevant physical theories and observed statistical properties of fractures.

MACROSCOPIC



REGIONS OF SATURATION

MICROSCOPIC

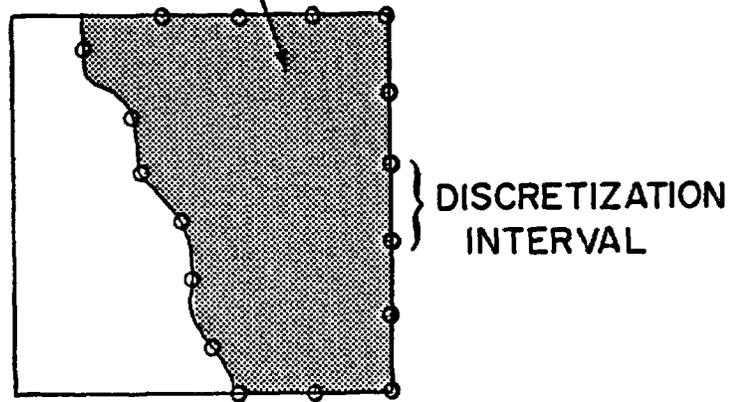


Figure 3.1: Macroscopic and microscopic formulations of unsaturated flow through fractured rock.

The following sections first present an overview of capillary theory, which is a fundamental description of the forces acting upon fluids contained within small cavities. Subsequent sections describe how synthetic hydraulic properties of discrete fractures are generated.

3.1.1 Theory of Flow Through Capillaries

Capillary theory relates the height of rise of a liquid in a tube to the diameter of the tube. The relationship is derived from the free energy states of the liquid-solid, liquid-gas and solid-gas interfaces (Richards, 1931). The force acting upon the interfacial junction is the free energy difference per unit length perpendicular to the junction (Figure 3.2). Equivalently, the force is the free energy difference per unit area multiplied by the length of the junction. The force is acting to minimize the sum of the free energies of the system. The junction will adjust until the sum of the energies are minimized or until an equal and opposing force is encountered. Because the solid may not deform, the location of the junction will move along the two-dimensional surface of the solid, normal to the fluid interface junction.

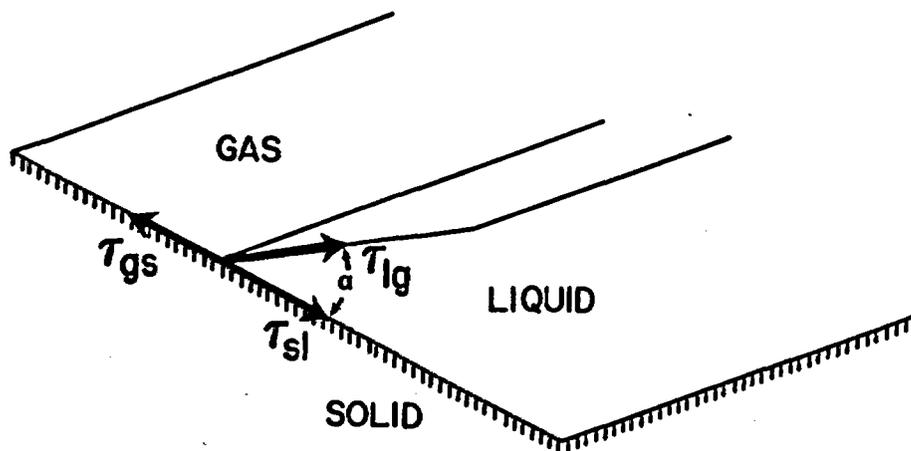


Figure 3.2: Surface tension forces acting upon a gas-liquid-solid interface.

When the junction is a ring in a hollow tube, the force is directed toward the drained end of the tube. In this case, the magnitude of the force can be computed as the product of the free energy surface density difference and the length of the junction, or:

$$(3.1) \quad F = \tau L$$

where

- F force acting upon the junction, N;
- τ free energy surface density difference, J/m^2 ; and
- L length of the junction, m.

The free energy surface density difference is obtained by noting that the force acting upon the junction of the interfaces is the sum of the individual forces acting at the junction (Hillel, 1971, p. 41):

$$(3.2) \quad F = (\tau_{lg} \cos \alpha + \tau_{sl} - \tau_{sg}) L$$

where

- τ_{lg} liquid-gas interface free energy surface density, J/m^2 ;
- τ_{sl} solid-liquid interface free energy surface density, J/m^2 ;
- τ_{sg} gas-solid interface free energy surface density, J/m^2 ; and
- α meniscus angle, $^\circ$, with the solid.

The free energy surface density difference can be set equal to:

$$(3.3) \quad \tau = \tau_{lg} \cos \alpha + \tau_{sl} - \tau_{sg}$$

By also noting that the length of the junction is the circumference of a circle, Equation 3.1 can be expressed as:

$$(3.4) \quad F = \tau (2 \pi r)$$

where r is the radius of the capillary tube, m. If the capillary tube is suspended vertically with the bottom of the capillary immersed in water, then a gravitational force is directed downward. The magnitude of the gravitational force is the mass of the fluid multiplied by the gravitational constant, or:

$$(3.5) \quad F = V \gamma$$

where V is the volume of liquid within the capillary tube, m^3 , and γ is the specific weight of the liquid, Pa/m . By noting that the volume of the liquid within the capillary tube can be approximated by a cylinder, Equation 3.5 is replaced with:

$$(3.6) \quad F = (\pi h_c r^2)$$

where h_c is the height of rise of liquid within the capillary tube, m. The opposing forces must balance at equilibrium, allowing Equations 3.4 and 3.6 to be set equal. Solving for the height of rise yields:

$$(3.7) \quad h_c = 2 \tau / r \gamma$$

Incorporating Equation 3.3 in 3.7 yields:

$$(3.8) \quad h_c = 2 (\tau_{lg} \cos \alpha + \tau_{sl} - \tau_{sg}) / r$$

Many authors (e.g., Richards, 1931; Hillel, 1971; Marshall and Holmes, 1979) neglect the solid-liquid and solid-gas interface terms:

$$(3.9) \quad \tau_{sl} = \tau_{sg} = 0$$

which allows Equation 3.8 to be reduced to:

$$(3.10) \quad h_c = 2 \tau_{lg} \cos \alpha / r$$

It is further assumed that the contact angle is zero, appropriate for surfaces such as clean glass, resulting in the expression:

$$(3.11) \quad h_c = 2 \tau_{lg} / r \gamma$$

For most applications (i.e., temperatures near 20°C), the constants in Equation 3.11 can be assigned the values of:

$$(3.12) \quad \tau_{lg} = 0.07275 \text{ Pa m}$$

and

$$(3.13) \quad \gamma = 9806 \text{ Pa/m}$$

yielding:

$$(3.14) \quad h_c = 0.1484 \times 10^{-4} / r$$

Equation 3.14 can be generalized for the case where an arbitrary fluid potential is imposed at the bottom of the capillary tube. The height of rise from the bottom of the capillary tube, h (m) is calculated as the sum of the capillary head calculated using Equation 3.14 and the pressure head at the bottom of the capillary tube, h_p (m), outside of the capillary tube:

$$(3.15) \quad h = h_c + h_p$$

When the pressure head imposed at the bottom of the capillary tube is negative (i.e., a suction is imposed), the height of rise will be less than the height calculated using capillary theory. When the suction exceeds the capillary head, the capillary tube will be completely drained. To determine the suction required to drain the capillary, the height of rise in Equation 3.15 is set equal to zero, yielding:

$$(3.16) \quad h_c = - h_p$$

For capillary tubes inclined at an angle, β , from the vertical the height of rise within the tube can be calculated by first noting that the volume within the tube is approximately:

$$(3.17) \quad V = h_c r^2 / \cos \beta$$

The gravitational force opposing the capillary force is calculated as:

$$(3.18) \quad F = \pi h_c r^2 \gamma / \cos \beta$$

At equilibrium, the gravitational force equals the capillary force directed up the tube. Setting the forces equal and solving for the height of rise results in:

$$(3.19) \quad h_c = 2 \tau_{lg} \cos \beta / r \gamma$$

It should be noted that Equations 3.10 and 3.19 are equivalent except for the addition of the cosine coefficient which accounts for gravitational forces which are exerted as a function of the orientation of the tube.

Similar equations for a geometry characterized by parallel plates offset by a constant aperture are next derived. In this case, it will be shown that the aperture is used to determine the height of rise in place of the radius in the capillary tube geometry. Like a capillary tube, a fracture will not fill unless the ambient pressure head is less negative than the capillary height of rise. The force acting to draw water up the fracture is equal to the length of the fracture, multiplied by two to indicate that both walls are acting upon the water, and further multiplied by the free energy surface density difference:

$$(3.20) \quad F = 2 \tau L$$

where L is the length of the fracture, m. If the fracture is set in water at an arbitrary angle with respect to the vertical, then the gravitational force is the mass of the fluid multiplied by the gravitational constant and the cosine of the angle from horizontal, or:

$$(3.21) \quad F = V \gamma / \cos \beta$$

where V is now the volume of liquid within the fracture, m³. By noting that the volume of the liquid within the fracture can be approximated by a square prism, Equation 3.21 is replaced by:

$$(3.22) \quad F = h_c e L \gamma / \cos \beta$$

Solving for the vertical height of rise in a fracture and substituting known constants yields:

$$(3.23) \quad h_c = 0.1484 \times 10^{-4} / e$$

Equation 3.23 is valid for all fracture inclinations. The total head at any point within the fracture is the sum of the gravitational and pressure forces (exclusive of osmotic, thermal and other forces), expressed in terms of hydraulic head:

$$(3.24) \quad h_t = h_g + h_p$$

where

h_t total head, m;
 h_g elevation head, m; and
 h_p fluid pressure head, m, from:

$$(3.25) \quad h_p = h - h_c$$

where h is the pressure head exclusive of capillary head, m , and h_c is the capillary head, m .

3.1.2 Fracture Moisture Characteristic Function

The relationship between the water content of a fracture and the fluid potential is the moisture characteristic curve. For a horizontal fracture with a uniform aperture and one end immersed in a fluid reservoir maintained at constant total head, h_t , the relationship is a step function; the fracture will be entirely filled at for positive pressure plus capillary heads (i.e., $h = h_p + h_c > 0$), and the fracture will be completely drained otherwise:

$$(3.26) \quad \theta/\theta_s = \begin{cases} 0 & h_p < -h_c \\ 1 & h_p \geq -h_c \end{cases}$$

where θ_s is the saturated water content, dimensionless. The relationship between water content and fluid potential for a vertical fracture with the lower edge of the fracture immersed in a reservoir maintained at a constant head is a ramp function:

- o The fracture will be entirely filled when the total pressure head equals the height of the fracture, H ; and
- o The fracture will be drained when the total pressure plus capillary heads are negative.

The ramp function can be expressed as a function of the total head applied at the lower end of the fracture, h_t :

$$(3.27) \quad \theta/\theta_s = \begin{cases} 0 & h_t < -h_c \\ h_t/(H-h_c) & -h_c \leq h_t \leq H-h_c \\ 1 & h_t \geq H-h_c \end{cases}$$

where H is the height of the fracture. For inclined fractures, Equation 3.27 is rewritten as:

$$(3.28) \quad \theta/\theta_s = \begin{cases} 0 & h_t < -h_c \\ h_t/(H'-h_c) & -h_c \leq h_t \leq H'-h_c \\ 1 & h_t \geq H'-h_c \end{cases}$$

where $H' = H \sin \beta$ and β is the dip of the fracture from the horizontal, degrees (Figure 3.3). Figure 3.4 illustrates a hypothetical characteristic curve for horizontal, vertical and inclined fractures.

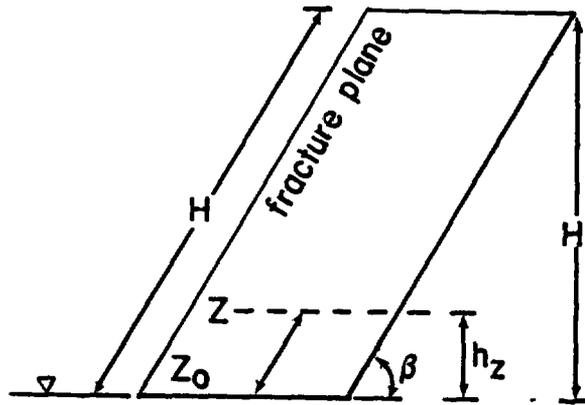


Figure 3.3: Geometric properties of an inclined fracture in contact with a liquid surface at its base.

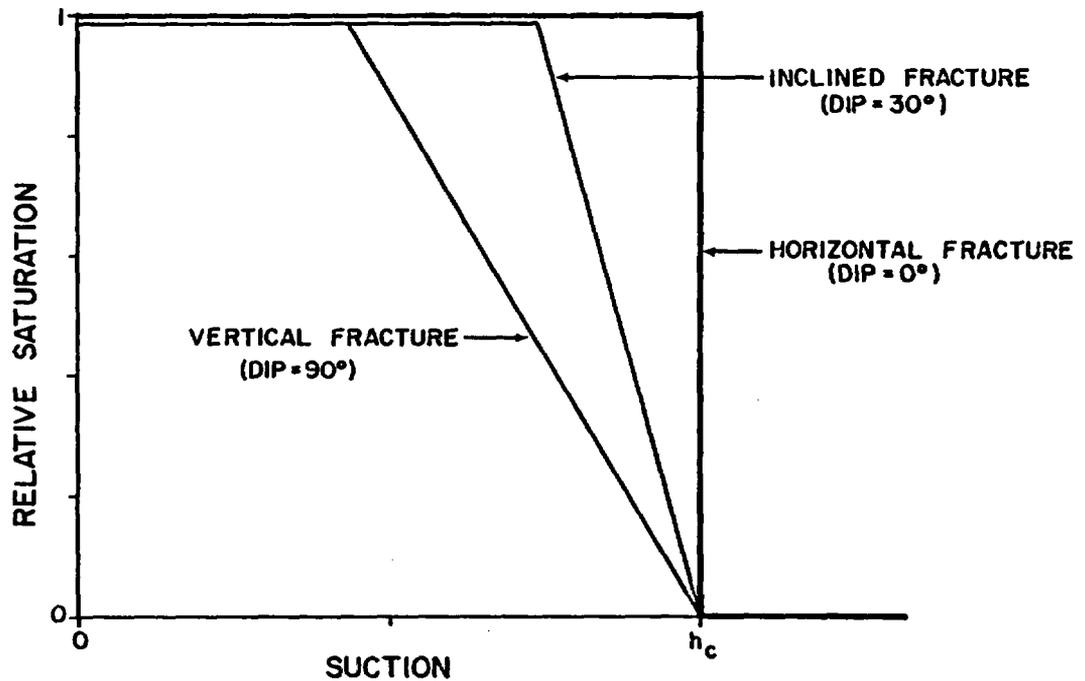


Figure 3.4: Moisture characteristic curves for planar fractures with constant capillary head at various orientations.

3.1.3 Fracture Unsaturated Transmissivity Function

For conditions of variable fracture saturation, the assumption is made that an air-water interface exists within the fracture, such that two flow fields can be defined within an individual fracture (Figure 3.5), which is exact for fractures with a constant aperture. When the fracture aperture is not constant, it is further assumed that large apertures which lie on the wetted side of the interface remain saturated even if the pressure head within the fracture is more negative than the capillary head. In addition, isolated small apertures which lie on the opposite side of the interface are assumed to remain unsaturated. The pressure head at the interface is equal to a capillary head which is assumed constant for the entire fracture. From Equations 3.24 and 3.25, the pressure head exclusive of the capillary head, h , on either side of an air-water interface is equal to zero, or:

$$(3.29) \quad h = h_t - h_g + h_c = 0$$

and the residual pressure head within the saturated domain interior is:

$$(3.30) \quad h = h_t - h_g + h_c > 0$$

The capillary head, h_c , is determined using Equation 3.23 while the gravitational head, h_g , is the elevation difference between the point under consideration, z (m) and an arbitrary reference elevation, z_0 (m):

$$(3.31) \quad h_g = z - z_0$$

If z is used as a local coordinate within the plane of the fracture, Equation 3.31 must be adjusted by the dip of the fracture, β , or:

$$(3.32) \quad h_g = (z - z_0) \sin \beta$$

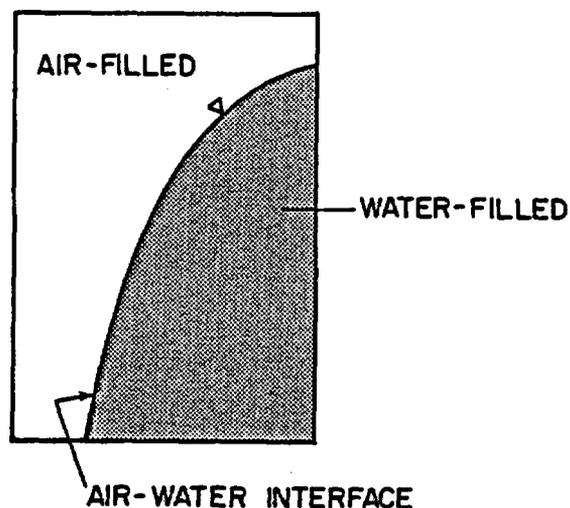


Figure 3.5: Conceptual model of zone of saturation within an unsaturated fracture.

3.2 Boundary Integral Solution of Free Surface Problem

The position of the equilibrium air-water interface is found by first locating temporary nodes which move the interface incrementally toward the equilibrium interface. The interface is moved incrementally because the fluid pressure distribution within the flow region will change as the boundary conditions change. The methodology for determining the temporary interface consists of (Figure 3.6):

- o Determining the fluid pressure at every boundary node using:

$$(3.33) \quad h_p = h - h_z = h - (z - z_0) \sin\beta$$

- o Determining which nodes are beyond the air-water interface by testing whether the fluid pressure is less than the sum of the capillary and air pressure heads, or:

$$(3.34) \quad h_p < (h_c + h_a)$$

- o For those nodes which satisfy Equation 3.34, a new temporary position is calculated such that:

- † The total head at the temporary location, h' , is equal to the total head at the previous location, h^0 :

$$(3.35) \quad h' = h^0$$

- † The pressure head at the temporary location, h_p' , is equal to the interfacial pressure head, h_p , or:

$$(3.36) \quad h_p' = h_p$$

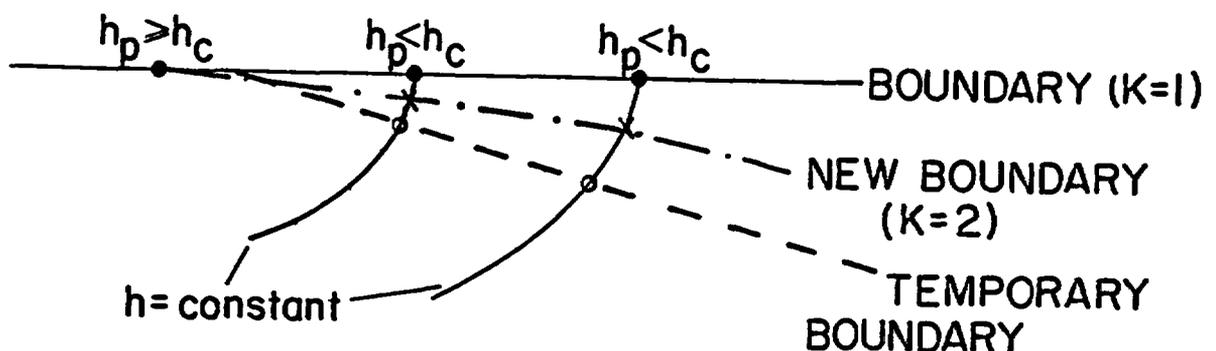


Figure 3.6: Procedure for locating nodal positions: (1) Determine which nodes satisfy $h_p < h_c$, solid circles; (2) For these nodes, locate new temporary boundary nodes on the same isohead contour and the contour of $h_p = h_c$, open circles; (3) Update nodal position by finding midpoint between initial and temporary nodal positions, crosses.

The position of the temporary location is calculated by using a Newton-Raphson iteration scheme to incrementally arrive at the new position. The iterative scheme calculates displacements in the z direction, dz, by using the chain rule:

$$(3.37) \quad dz = \frac{\partial h/\partial x \, dh_p - \partial h_p/\partial x \, dh}{\partial h/\partial x \, \partial h_p/\partial z - \partial h/\partial z \, \partial h_p/\partial x}$$

where

$$(3.38a) \quad dh = h' - h^0 = 0$$

and

$$(3.38b) \quad dh_p = h_p' - h_p^0 = (h_p^0 + h_p)/2 - h_p^0 = (h_p - h_p^0)/2$$

The displacement in the x direction, dx, is calculated using:

$$(3.39) \quad dx = (dh - \partial h/\partial z \, dz) / \partial h/\partial x$$

The values of the changes in (h, h_p) with respect to (x, z) are determined using:

$$(3.40a) \quad \partial h/\partial z = [h(x, z+dz) - h(x, z)] / dz$$

$$(3.40b) \quad \partial h/\partial x = [h(x+dx, z) - h(x, z)] / dx$$

$$(3.40c) \quad \partial h_p/\partial z = [h_p(x, z+dz) - h_p(x, z)] / dz$$

and

$$(3.40d) \quad \partial h_p/\partial x = [h_p(x+dx, z) - h_p(x, z)] / dx$$

where dx and dz are small relative to the offset desired.

- o The updated position of node i after iteration k is calculated using half of the calculated displacement:

$$(3.41a) \quad x(i, k) = x(i, k-1) + dx(i, k)/2$$

and

$$(3.41b) \quad z(i, k) = z(i, k-1) + dz(i, k)/2$$

where dx(i, k) and dz(i, k) are the calculated horizontal and vertical displacements for node i during iteration k, respectively.

Once the temporary positions for all nodes has been determined, the boundary value problem is recomputed for the new geometry. Because the original boundary conditions are still used, only the position of some of the nodes must be changed. If, after updating the pressure head at all boundary nodes, the pressure head at any of the nodes are less than the sum of the capillary plus atmospheric pressure heads (i.e., Equation 3.34), the methodology presented above is repeated. Iteration is

stopped when the change in pressure head, dh_p , from Equation 3.38b, is less than a desired stopping criterion. Appendix A presents the source code and users manual for the FORTRAN program BIM which implements the procedure.

3.2.1 Comparison of Boundary Integral Results with Analytic and Laboratory Results

In order to validate the application of the boundary integral method to unsaturated flow problems, a simplified flow example is constructed such that a small circular ring is maintained at a constant total head within a vertical fracture (Figure 3.7).

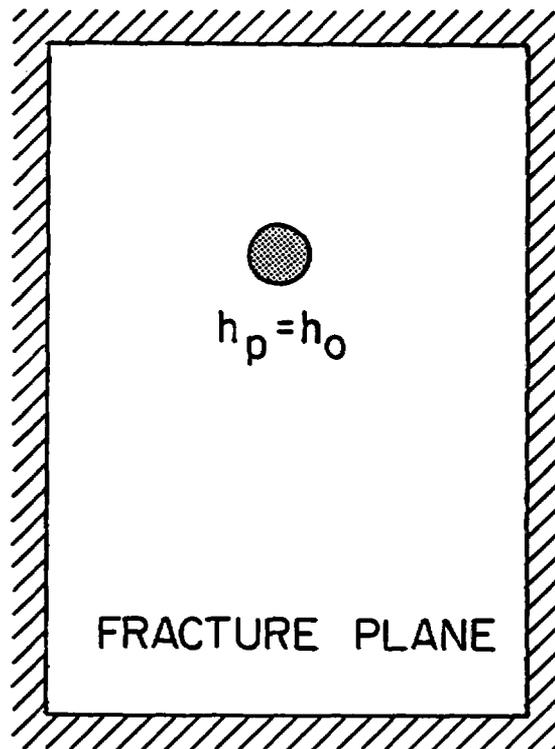


Figure 3.7: Circular constant head source in a planar fracture.

An approximate analytic solution to this problem consists of superimposing two driving forces, a uniform gravitational field directed in the (-z) direction, and a radial flow field directed in a radial (+r) direction, where:

$$(3.42) \quad r^2 = x^2 + z^2$$

and x is orthogonal to z. A pressure head, h_0 is imposed along the outer rim of a circular opening within the flow domain:

$$(3.43) \quad h_p = h_0 \quad \text{at } r = r_0$$

where r_0 is the radius of the opening. The velocity of water resulting from gravitational forces is:

$$(3.44a) \quad \begin{aligned} v_x &= -K \partial h / \partial x = -K \partial [h_z + h_p] / \partial x \\ &= -K \partial [(z-z_0) \sin\beta + h_p] / \partial x \\ &= -K \partial h_p / \partial x \end{aligned}$$

and

$$(3.44b) \quad \begin{aligned} v_z &= -K \partial h / \partial z = -K \partial [h_z + h_p] / \partial z \\ &= -K \partial [(z-z_0) \sin\beta + h_p] / \partial z \\ &= -K [\sin\beta + \partial h_p / \partial z] \end{aligned}$$

where z_0 is an arbitrary reference elevation, located at the center of the radial source in this example, and K is the hydraulic conductivity of the flow domain, assumed constant. The fluid pressure is constant for steady gravitational flow within a fracture of uniform hydraulic properties and with no sources or sinks. Equation 3.44 becomes:

$$(3.45a) \quad v_x = 0$$

and

$$(3.45b) \quad v_z = -K \sin\beta$$

The velocity of water resulting from the radial flow field is:

$$(3.46) \quad v_r = Q / A = Q / 2\pi r e$$

where

- Q injected flow rate through the circular source;
- A cross-sectional area through which radial flow occurs; and
- e thickness of the flow domain, i.e., the fracture aperture.

The velocity at any point is the superposition of the gravitational and radial flow solutions, or:

$$(3.47a) \quad v_x = 0 + Q / 2\pi r e = Q / 2\pi (x^2 + z^2)^{1/2} e$$

and

$$(3.47b) \quad \begin{aligned} v_z &= -K \sin\beta + Q / 2\pi r e \\ &= -K \sin\beta + Q / 2\pi (x^2 + z^2)^{1/2} e \end{aligned}$$

Figure 3.8 presents the configuration of the flow field resulting from this formulation. Of interest to this analysis is the height of rise of the fluid above the internal source, as well as the width of the flow emanating from the internal source at large distances below the source. The height of rise is calculated by noting that the position of the source is directly above the internal source (i.e., $x = 0$) and that the radial and gravitational velocities in the z -direction at this point are equal and opposite in magnitude, or:

$$(3.48a) \quad v_z = -v_r$$

or

$$(3.49b) \quad K \sin\beta = Q / 2\pi z' e$$

and

$$(3.49) \quad z' = Q / 2\pi e K \sin\beta = Q / 2\pi T \sin\beta$$

where z' is the height above the center of the radial source where the stagnation point occurs, and T is the domain transmissivity. The width of the flow domain emanating from the radial source is calculated at long distances by noting that the pressure gradient resulting from the radial source approaches zero at large distances. Thus, the velocity resulting from the radial source is zero, leaving only the gravitational driving force, or:

$$(3.50) \quad v_z = -K \sin\beta = Q / A = Q / x' e$$

and

$$(3.51) \quad x' = -Q / T \sin\beta$$

where x' is the width of the flow domain emanating from the radial source. It is interesting to note that the ratio of x' to z' is:

$$(3.52) \quad x'/z' = [-Q / T \sin\beta] / [Q / 2\pi T \sin\beta] = -2\pi$$

The location of the dividing streamline between the saturated and unsaturated zones is (Kovacs, 1981):

$$(3.53) \quad z = x \cotan(x/z')$$

To evaluate the accuracy of the numerical and analytic formulations presented above, a flow visualization experiment was performed using two glass panels separated by metal shims. The dimensions of the glass panels are (0.240 x 1.200 x 0.006 m). Two metal shim strips (1.200 x 0.010 x 0.0001 m) were placed between the glass panels along the sides of the longest dimension, and then clamped using six clamps along each side. Twenty seven metal shim pieces (0.010 x 0.010 x 0.0001 m) were placed regularly within the space between the two glass panels for the purpose of providing a uniform aperture between the two panels. Figure 3.9 illustrates the geometry of the flow visualization experiment.

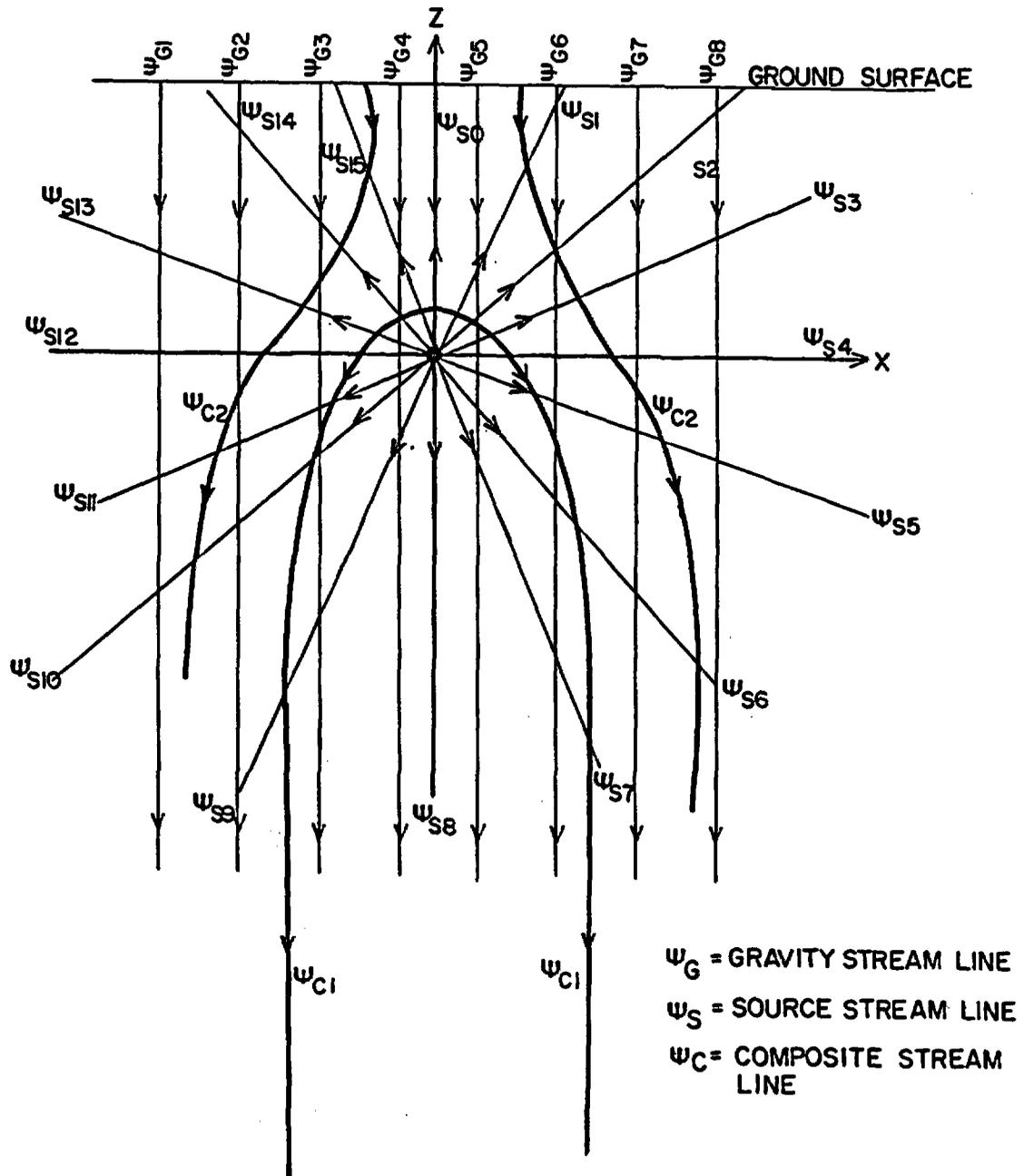


Figure 3.8: Graphical representation of superimposed flow fields resulting from a circular source and a gravitational field.

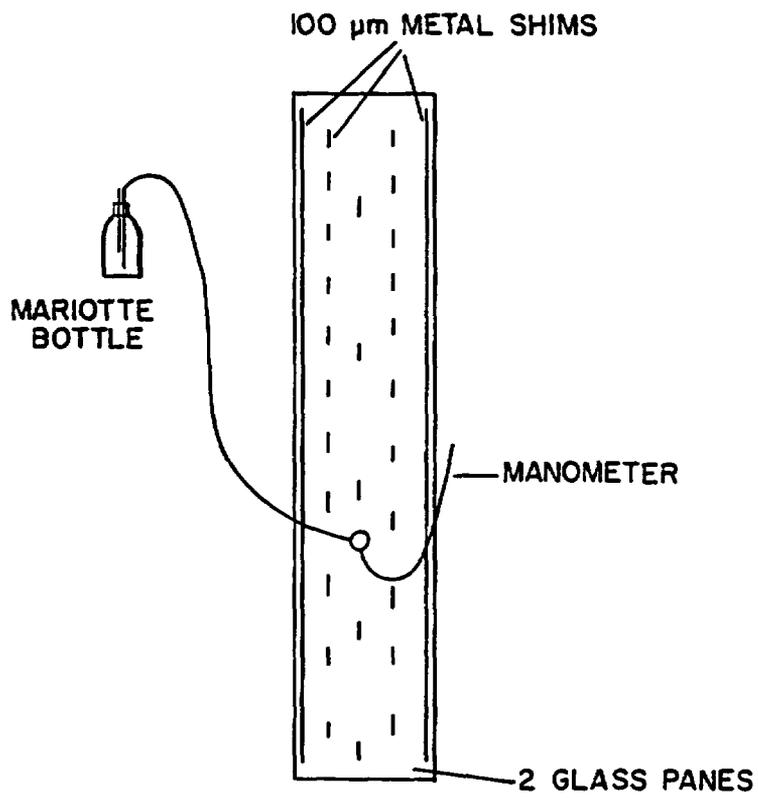


Figure 3.9: Laboratory flow visualization experiment.

To provide a constant head of water at a point between the panels, a mariotte bottle was connected with tubing and a stopper to the interior of a 0.010 m diameter hole drilled through one of the glass panels. The flow rate was monitored by measuring the time required to drain 100 ml from the mariotte bottle. Pressure at the flow inlet of the glass panel was obtained using a water manometer connected to the same stopper. The flow rate was controlled by raising or lowering the mariotte bottle. During each flow experiment, the flow rate, inlet pressure, height of rise above the inlet, and the asymptotic width of the flow field below the inlet were measured.

Table 3.1 reports the test results and Figure 3.10 demonstrates the results for two input pressure head boundary conditions. The experimental results indicate that to within a maximum fifteen percent error the expected ratio of flow height to flow width is accurate. Variability in results are due to aperture irregularities and the inclusion of small air pockets within the flow domain. The effect of air pockets is to increase the flow width. Assuming that the air pockets do not conduct flow, and that the transmissivity of the saturated fracture remains constant, the flow width is:

$$(3.54) \quad x' = (- Q / T \sin\beta) / w_a$$

where w_a is the relative proportion of the air pockets width per unit flow width. Neither the effects of air pockets on the flow height, nor the effects of aperture irregularity on both the flow height and width are easily quantified. Simulation studies are required to evaluate these effects, but are beyond the scope of this study.

 Table 3.1: Results of flow visualization experiments.

Flow Rate (ml/s)	Height (mm)	Width (mm)	Error ¹ (%)
1.101	25	146	7
0.980	12	79	- 5
0.927	10	71	-13
0.899	15	80	15
0.700	10	56	11
0.640	7	47	- 7

¹ Error calculated using $100(1 - \text{Width}/\text{Height}/2 \text{ pi})$

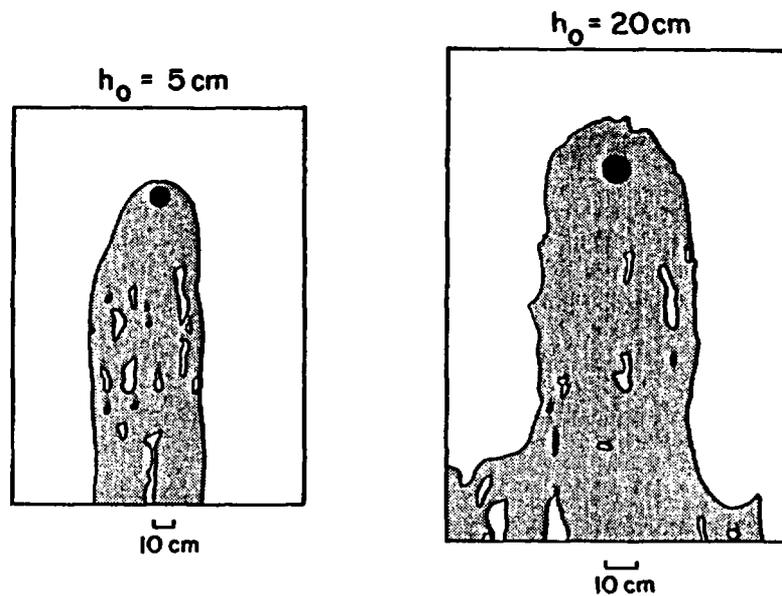


Figure 3.10: Flow visualization experiment for two input pressure head boundary conditions.

A comparison of the simulation model with the analytic and laboratory model results was performed using four input pressures, 10, 25, 40 and 50 mm. Figure 3.11 illustrates the geometry and nodal configuration of the flow domain. A (100 x 100 mm) square flow domain was constructed using 24 nodes around the perimeter. The top and two sides were assigned no flow boundary conditions (i.e., $q = 0$) and the bottom was assigned a zero total head (i.e., $h = 0$). A small (2 mm diameter) circular source region with a constant head was placed at a point 80 mm above the bottom of the flow domain. Eight nodes were used to construct the circular source. A unit transmissivity (i.e., $T = 1 \text{ mm}^2/\text{s}$) was assigned to the flow domain. Figure 3.12 illustrates the final air-water interface for the three input pressure head boundary conditions. Simulation results are summarized in Table 3.2.

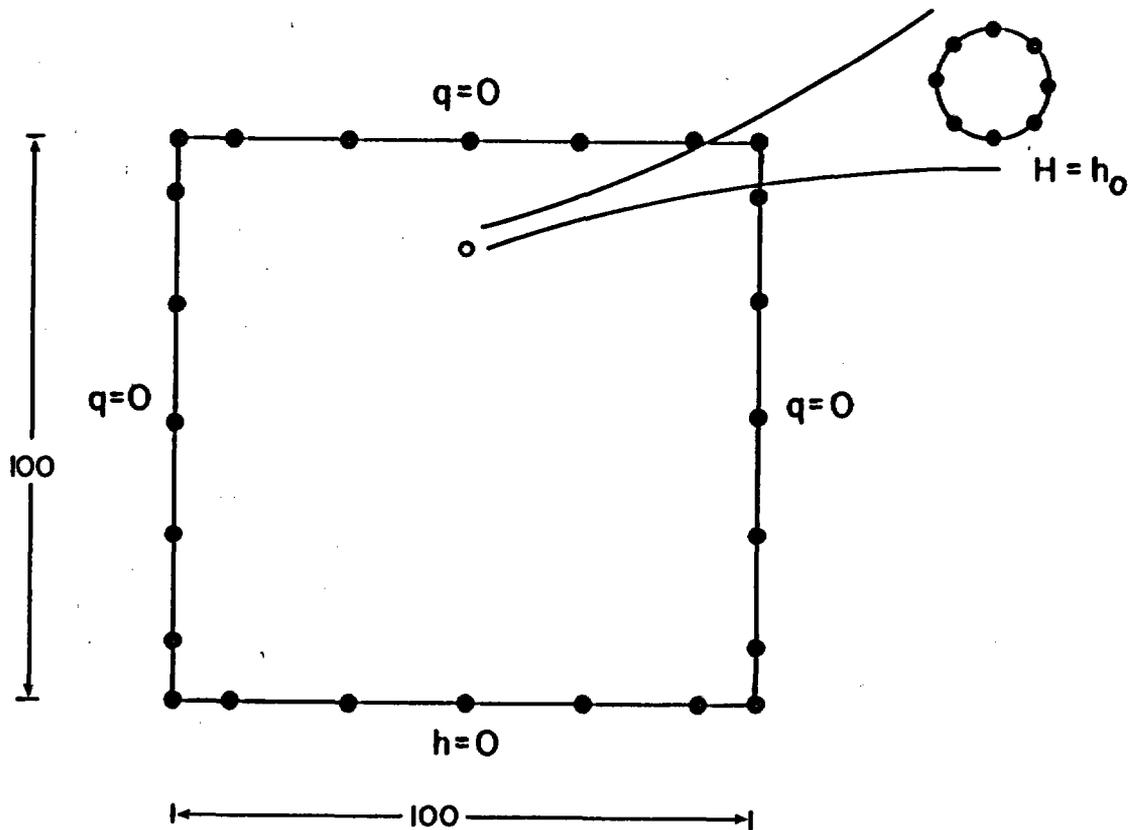


Figure 3.11: Unsaturated flow domain geometry for circular source.

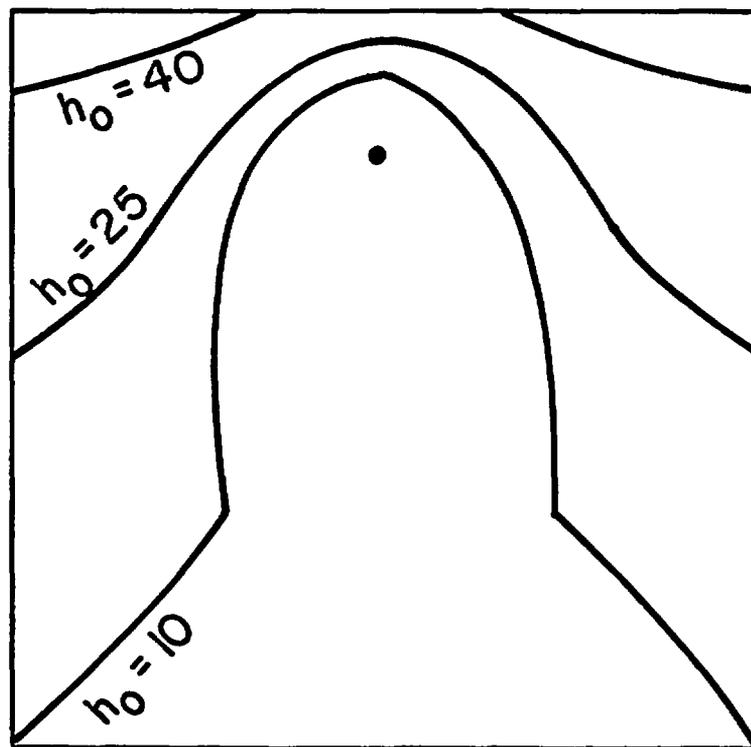


Figure 3.12: Simulated free surface position for three input pressure head boundary conditions.

Table 3.2: Results of computer simulation experiments. Flow geometry is presented as Figure 3.11. A unit fracture transmissivity is used.

Flow Rate (mm ³ /s)	Pressure (mm)	Height (mm)	Width (mm)	Error ¹
62.844	50	> 20	> 100	-
62.531	40	> 20	> 100	-
57.344	25	15.7	> 100	-
40.576	10	9.9	42.1	32 %

¹ Error calculated using $100(1 - \text{Width}/\text{Height}/2 \pi)$

The simulation results can be compared to analytic and laboratory results in two ways:

- o The observed flow rate divided by the unit transmissivity (i.e., $40.576 / 1 = 40.6$) is approximately equal to the calculated flow width (i.e., 42.1). One explanation for the small discrepancy may be that a unit gradient may not have been achieved at the location where the flow width was measured.
- o The ratio of the flow width to the flow height is smaller by 32 percent of the analytic results. This result can be attributed to the difficulty in numerically locating the node which lies immediately above the source. The numerical difficulty stems from the fact that the node lies at a stagnation point within the flow domain and a unique fluid gradient does not exist at that point.

3.2.2 Applications to Fracture Networks

To demonstrate the application of the boundary integral method to unsaturated media, a flow example is created using a single square fracture with dimensions of (100 x 100 m). The fracture is intersected by two other fractures forming linear slits of length 20 m. External nodes are placed along the rim of the fracture and internal nodes are placed along the lines of intersection between the fractures. External nodes are maintained as zero flow boundaries, and internal nodes are maintained at constant heads of 80 mm for the fracture intersecting at (x,z) coordinates between (60,80) and (80,80) and at 20 mm for the fracture intersecting between (20,20) and (20,40). The geometry of the flow example is indicated in Figure 3.13. Figure 3.14 presents the total and pressure heads for a flow example which assumes flow through a horizontal fracture. The fluid flux in this example amounts to 27.7 mm²/s.

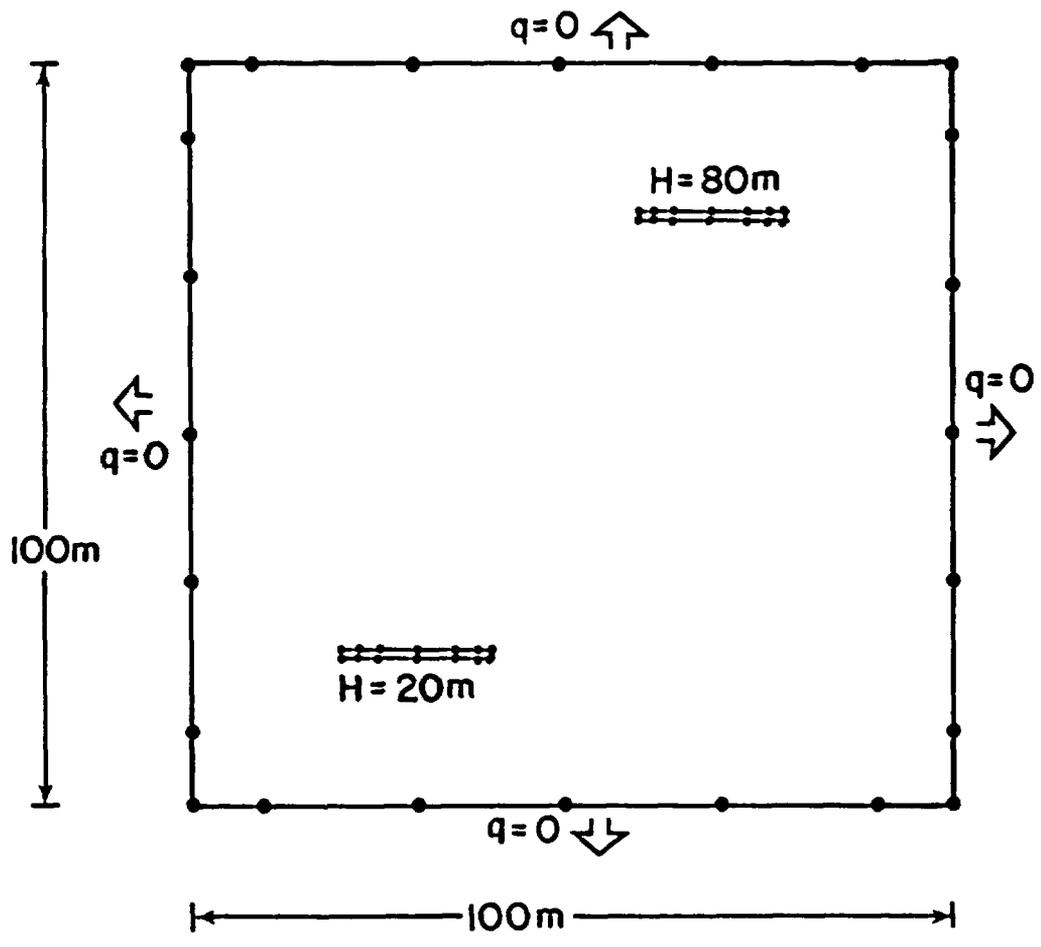


Figure 3.13: Flow geometry and boundary conditions for fracture intersecting two other fractures.

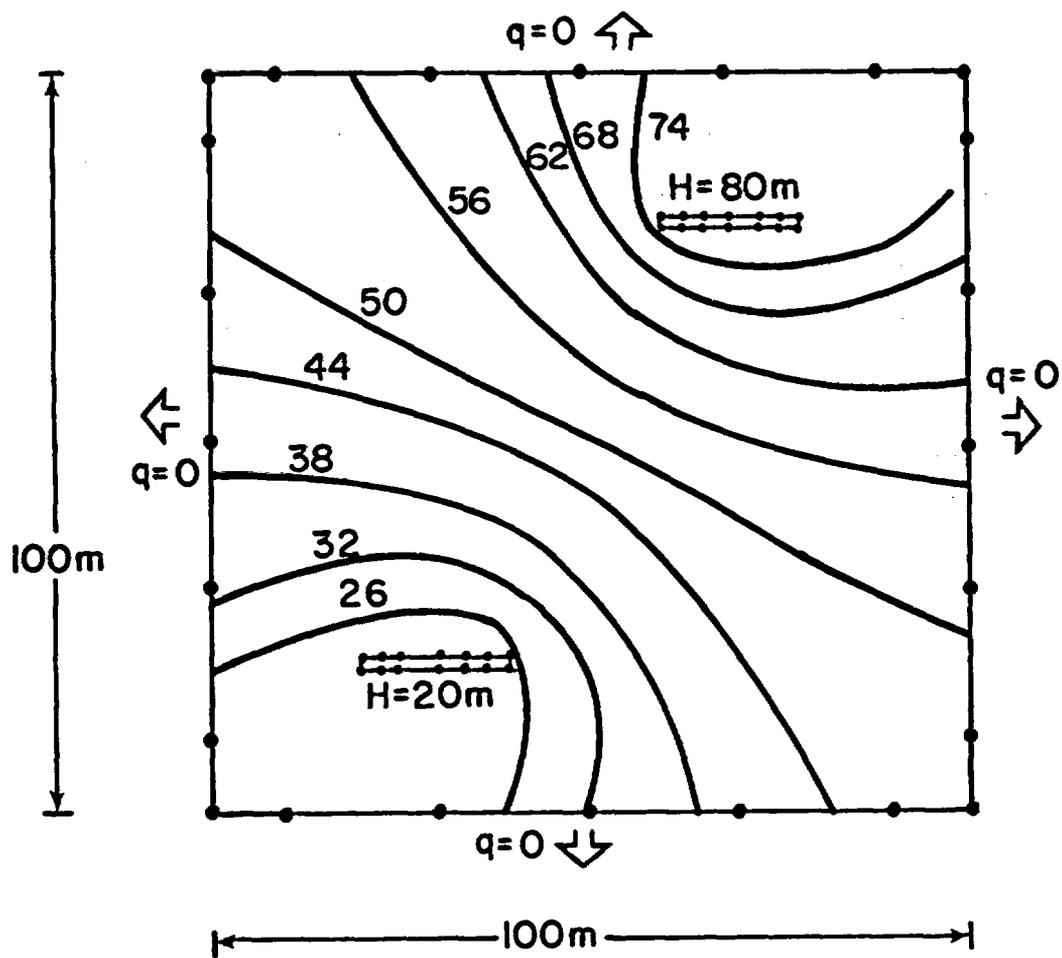


Figure 3.14: Contours of total head within the plane of a horizontal fracture.

Figure 3.15 presents the results of a simulation for flow through a vertical fracture with no capillary forces and no air entry. Also indicated is the region under negative pressure which would drain if an air entry route were available. If air is allowed to enter the fracture, then the equilibrium interface position can be calculated. The location of this interface after each iteration is shown in Figure 3.16 along with the final position. Compared to the previous example, the fluid flux has been reduced by 38 percent, to 17.21 mm²/s.

Flow through a vertical fracture with a capillary pressure head of 10 m is presented in Figure 3.17. Because of the increased flow area, an increase of 25 percent is observed over the previous example, to 21.46 mm²/s. Table 3.3 summarizes simulation parameters and flow results.

 Table 3.3: Simulation results for (1) horizontal flow, (2) vertical flow with no capillary head, and (3) vertical flow with capillary head.

Simulation:	1	2	3
Orientation:	Horizontal	Vertical	Vertical
Capillary Head (mm):	0.0	0.0	10.0
BIM Flux (mm ² /s):	27.7	17.2	21.4

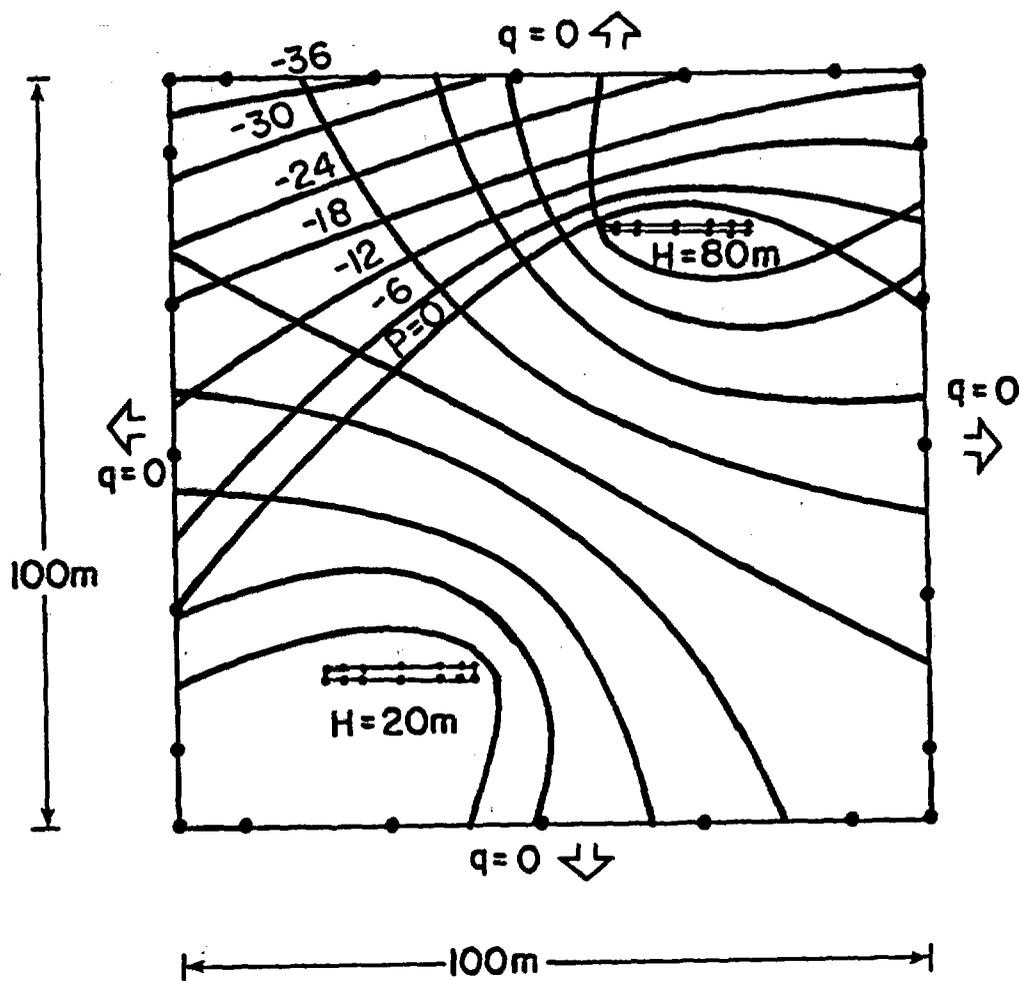
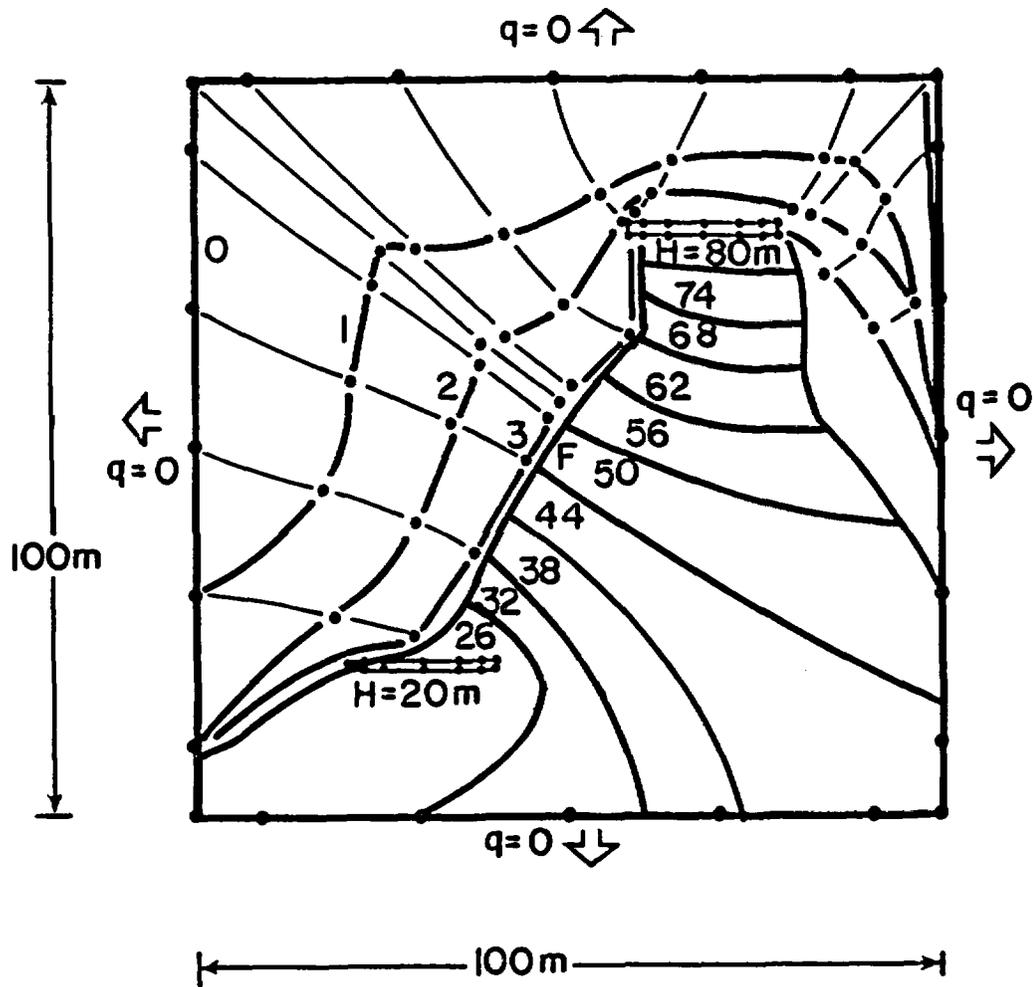
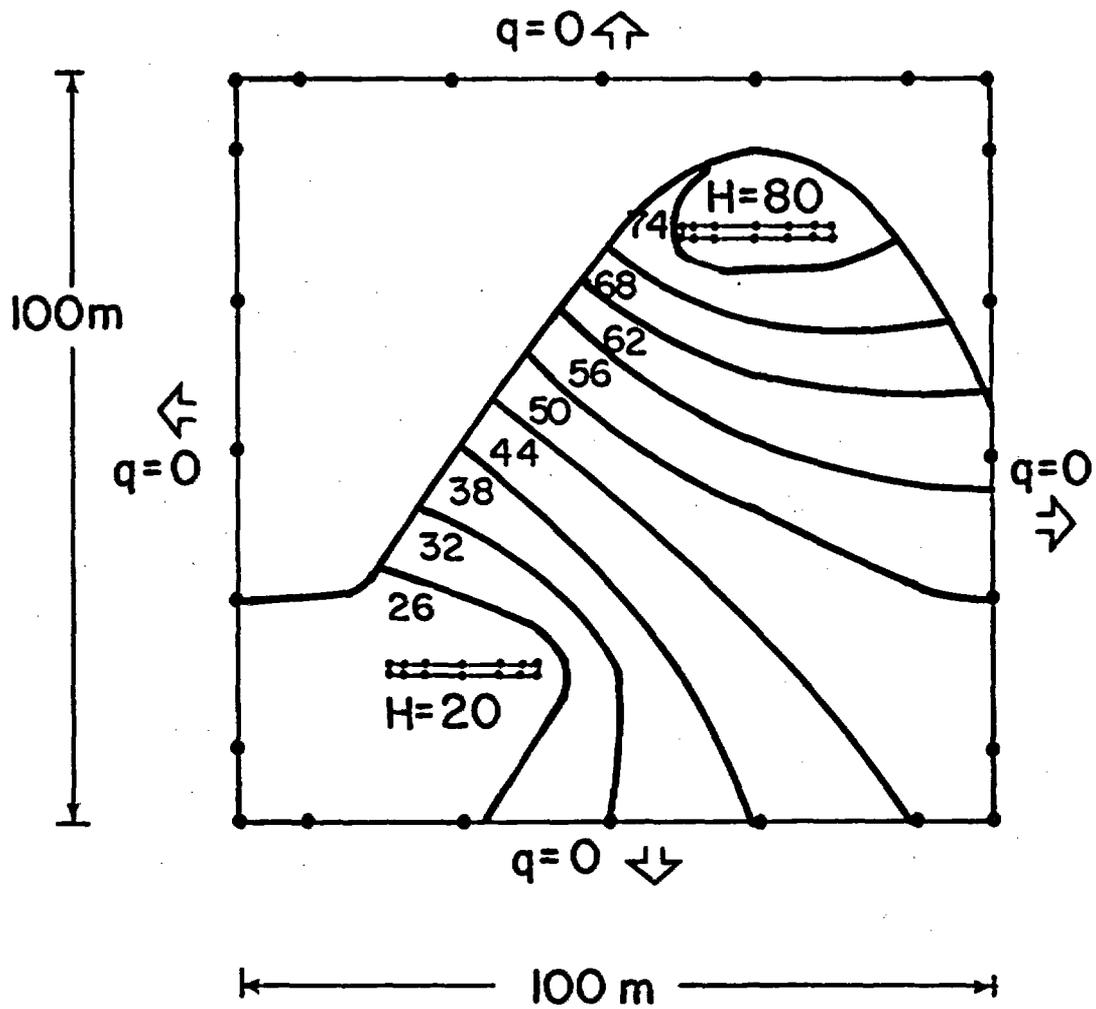


Figure 3.15: Contours of total head and zone where pressure heads are negative within the plane of a vertical fracture allowing no air entry.



VERTICAL FRACTURE
Capillary Head = 0m

Figure 3.16: Contours of total head and free surface position after successive iterations and after the final iteration within the plane of a vertical fracture allowing air entry.



VERTICAL FRACTURE
Capillary Head=10m

Figure 3.17: Contours of total head and interface position after the final iteration within the plane of a vertical fracture allowing air entry. A capillary head of 10 m was used.

CHAPTER 4

SOLUTE TRANSPORT THROUGH UNSATURATED FRACTURED ROCK

The ability to calculate travel times and breakthrough curves for flow through unsaturated discrete fracture networks is important for predicting the confinement capability of materials stored in such a medium. In addition, solute interactions with fracture surfaces and with the matrix which surrounds the fracture may be important sources of solute attenuation. In this chapter, the boundary integral method is extended to provide estimates of travel times and breakthrough curves within unsaturated fractured rock. The method incorporates variable velocities by integrating the inverse fluid velocity along flow paths between two boundaries, and by integrating the resultant travel time for individual flow paths over all flow paths intersecting the downstream boundary. While the method neglects molecular diffusion within the flow domain, hydrodynamic dispersion is incorporated by accounting for variable velocity profiles within individual fractures.

It is advantageous to use the boundary integral method over other methods because the boundary integral method provides the ability to calculate smoothly varying hydraulic heads and velocities at points internal to the flow domain. In other methods, interpolation functions must be used which may limit the accuracy of calculated heads and velocities, especially at boundaries between elements which discretize the interior of the flow domain (Figure 4.1).

4.1 Travel Time and Breakthrough Curve Calculation

Estimates of fluid travel times are obtained by first determining the fluid velocity within a fracture and then relating fluid velocity to travel times. Equation 2.1 presented an expression which relates the darcian velocity vector to a hydraulic conductivity tensor and the hydraulic head gradient. Equation 2.1 is repeated here as:

$$(4.1) \quad \underline{q}(\underline{x}) = - \underline{K} \nabla h(\underline{x})$$

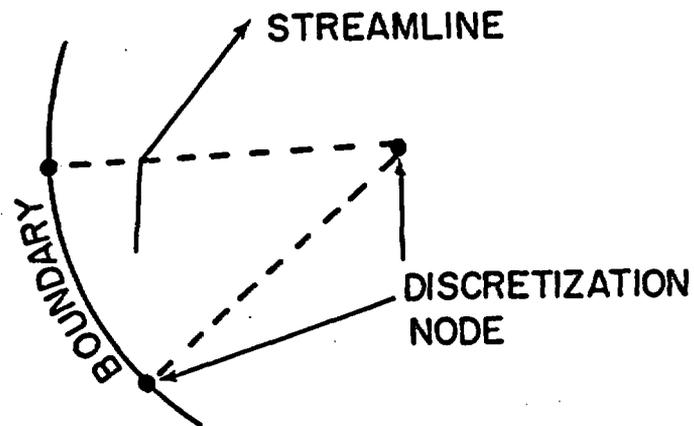
where

\underline{q}	darcian flow velocity, m/s;
\underline{x}	position, m;
\underline{K}	hydraulic conductivity, m/s;
∇	gradient operator, 1/m; and
h	hydraulic head, m.

Equation 4.1 is valid for one, two, and three dimensional flow, but is used in this chapter for two dimensional flow fields. The estimation of travel time between two points along a flow path requires that the integral of inverse fluid velocity over the one-dimensional flow path (within a two dimensional flow domain) be evaluated (Figure 4.2):

$$(4.2) \quad t_t = \int_{\underline{x}_1}^{\underline{x}_2} \frac{1}{v(\underline{x})} d\underline{x} \quad s = \text{constant}$$

FINITE ELEMENT



BOUNDARY INTEGRAL

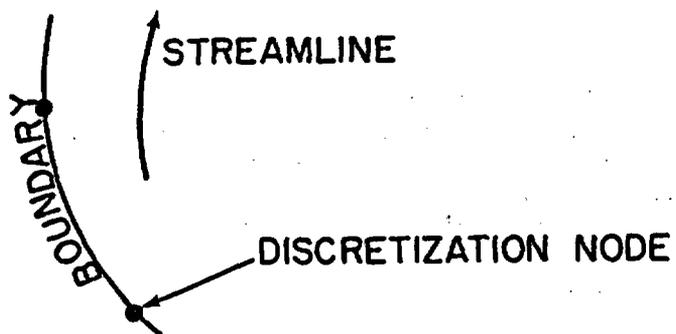


Figure 4.1: Finite element and boundary integral approximations of fluid streamlines. Note sharp change in direction at edge of element in finite element approximation.

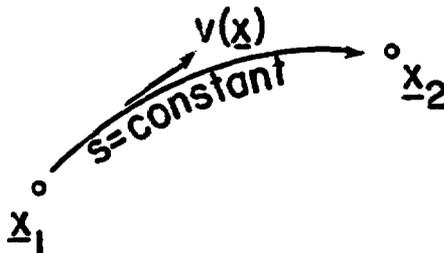


Figure 4.2: Streamline showing two endpoints and velocity at a point along the streamline.

where

- t_t travel time between two points, s;
- \underline{x}_1 position of initial point, m;
- \underline{x}_2 position of final point, m;
- \underline{v} fluid velocity, m/s; and
- s stream function, m^2/s .

In a porous medium the porosity is used to relate the darcian to the fluid velocity. Within a fracture, however, the porosity is equal to unity and the darcian and fluid velocities are equal. Equation 4.2 incorporates spatially variable velocity and hydraulic conductivity fields and assumes no diffusion or dispersion and that the initial and final points lie on the same flow path. Within any two dimensional flow field, the stream function is defined using the Cauchy-Riemann conditions:

$$(4.3a) \quad \partial s / \partial y = v_x = - T \partial h / \partial x$$

$$(4.3b) \quad \partial s / \partial x = - v_y = T \partial h / \partial y$$

where x and y are orthogonal cartesian coordinates in two dimensions and T is the isotropic transmissivity. For anisotropic flow, lines of constant stream functions are no longer orthogonal to the lines of constant hydraulic head. In this case, Equation 4.3 can be either expanded to include a transmissivity tensor, i.e.:

$$(4.4a) \quad \partial s / \partial y = v_x = - (T_{xx} \partial h / \partial x + T_{xy} \partial h / \partial y)$$

$$(4.4b) \quad \partial s / \partial x = - v_y = (T_{yx} \partial h / \partial x + T_{yy} \partial h / \partial y)$$

where T_{xx} , T_{xy} , T_{yx} and T_{yy} are the elements of \underline{T} with $T_{xy} = T_{yx}$, or a pseudopotential function can be defined such that the pseudopotential function is orthogonal to the stream function (Matanga, 1988). While the latter formulation aids graphical construction of flow nets, Equation 4.4 is used in this analysis because the graphical construction of flow nets is unnecessary. The stream function formulation (also called the dual formulation) has been used by Frind and Matanga (1985),

Frind et al. (1985) to model contaminant transport from landfills. In addition, Hull and Koslow (1986) and Philip (1988) have used the streamline formulation to route solutes through fracture junctions.

A breakthrough curve is generated for conditions of steady flow and arbitrary input concentrations of a conservative tracer (i.e., a tracer which travels at the same velocity as the water) by noting that along a flow path the concentration of the tracer will be equal to the inflow concentration delayed by the travel time, or (Figure 4.3):

$$(4.5) \quad C_2(s,t) = C_1(s,t-t_t)$$

where

- t time, s;
- C_2 downstream concentration, kg/m^3 ; and
- C_1 input concentration, kg/m^3 .

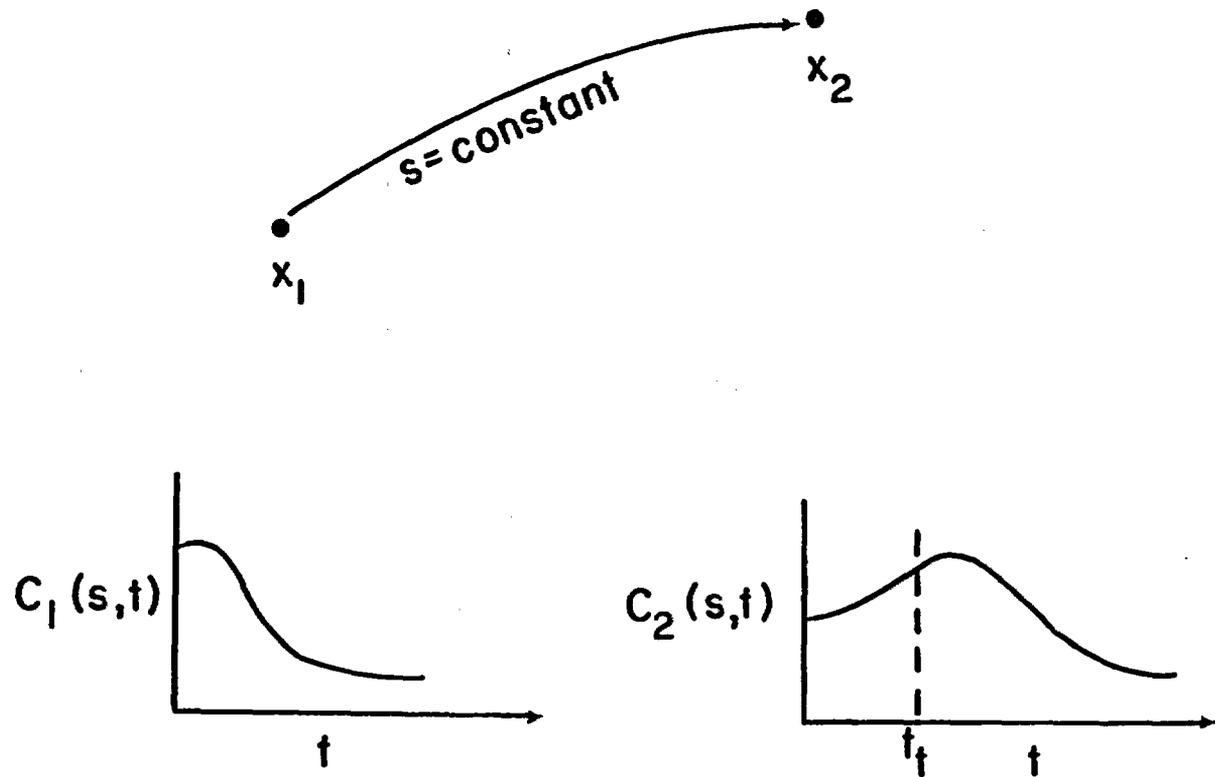


Figure 4.3: Translation in time of solute concentration curve ignoring molecular diffusion.

In general, the breakthrough curve at a line boundary resulting from an arbitrary input from a second line boundary is the sum of breakthrough curves averaged over all flow paths (i.e., over all stream functions) intersecting the downstream boundary (Figure 4.4):

$$\begin{aligned}
 (4.6) \quad \bar{c}(t) &= \int_{s_1}^{s_2} \bar{c}_2(s, t) ds / (s_2 - s_1) \\
 &= \int_{s_1}^{s_2} \bar{c}_1(s, t - t_t(s)) ds / (s_2 - s_1) \\
 &= \int_{s_1}^{s_2} \bar{c}_1(s, t - \int_{x_1}^{x_2} 1/v(\underline{x}) dx) ds / (s_2 - s_1)
 \end{aligned}$$

where s_1 and s_2 are the bounding flow paths, m/s, on the downstream boundary.

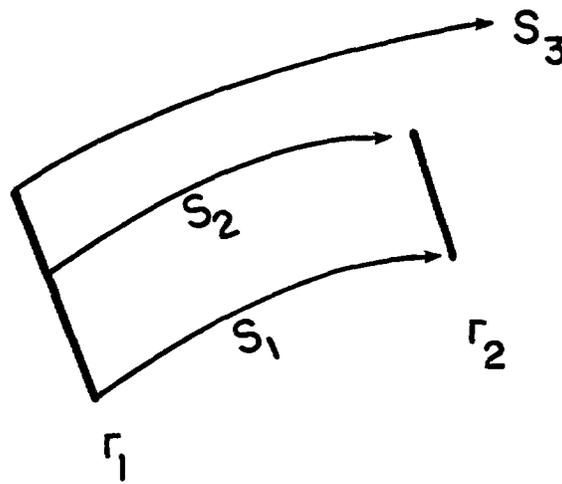


Figure 4.4: Geometry of flow between two boundaries. Also indicated are limiting streamlines.

4.1.1 Travel Time and Breakthrough Curve Calculations Using Boundary Integral Method

The boundary integral method is used to calculate travel times and breakthrough curves by assuming the existence of a constant aperture within the plane of the fracture. The stream function is solved using the Laplace equation:

$$(4.7) \quad \nabla^2 s = 0$$

Equation (4.7) can be solved for specific boundary conditions using the boundary integral method (Figure 4.5):

$$(4.8) \quad \underline{A} \underline{s} = \underline{B} \underline{s}'$$

where

$\underline{A}, \underline{B}$ boundary integral coefficient matrices; and
 \underline{s}' derivative of stream function w.r.t. the outward directed boundary normal, computed using the Cauchy-Riemann condition.

The \underline{A} and \underline{B} matrices are identical to those used for solving the head and flux boundary integral problem, avoiding the need for recomputing these matrices. The boundary conditions are imposed by equating the stream function with cumulative discharge:

$$(4.9a) \quad s_i = s_{i-1} + \int q_i d\Gamma$$

or

$$(4.9b) \quad s_i = s_{i-1} + l_i (q_i + q_{i-1})/2$$

where

Γ one dimensional flow domain boundary;
 s_i stream function value at node i ;
 l_i length of boundary segment between nodes $i-1$ and i ; and
 q_i discharge at node i .

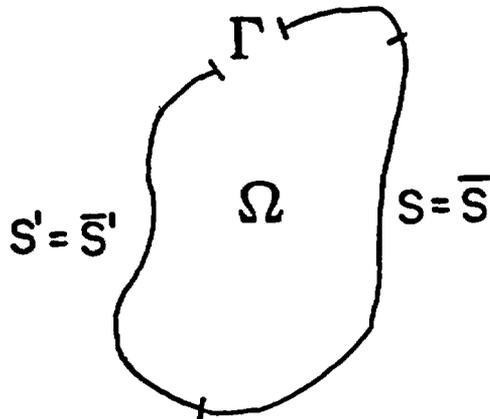


Figure 4.5: Boundary conditions for stream functions.

4.1.2 Specification of Potential and Stream Intersections

The determination of the location of streamlines using Equation 4.6 is a necessary first step prior to performing the integration of Equation 4.2. In order to provide a continuous representation of the velocity field, the velocity along the streamline must be evaluated. Instead of the continuous representation, however, an approximation is made by discretizing the streamlines into segments so that equal potential differences are found along each segment. The location of the segment endpoints are determined by finding the intersection of the streamlines with potential contours. The range of stream functions is divided into ten equal intervals, as is the range of potential functions, as shown in Figure 4.6. The (x,z) locations of the intersections of the stream contours with the potential functions are found using a Newton-Raphson procedure. The method is similar to that developed in the previous chapter for unsaturated flow. In this case, however, rather than solving for the intersection of the isohead contour with a desired pressure contour, the goal is to find the location of the intersections of the desired stream and isohead contours. The iterative procedure incrementally approaches the intersection using:

$$(4.10a) \quad x(i,k) = x(i,k-1) + dx(i,k)/2$$

and

$$(4.10b) \quad z(i,k) = z(i,k-1) + dz(i,k)/2$$

where $dx(i,k)$ and $dz(i,k)$ are the horizontal and vertical displacements for node i during iteration k , respectively.

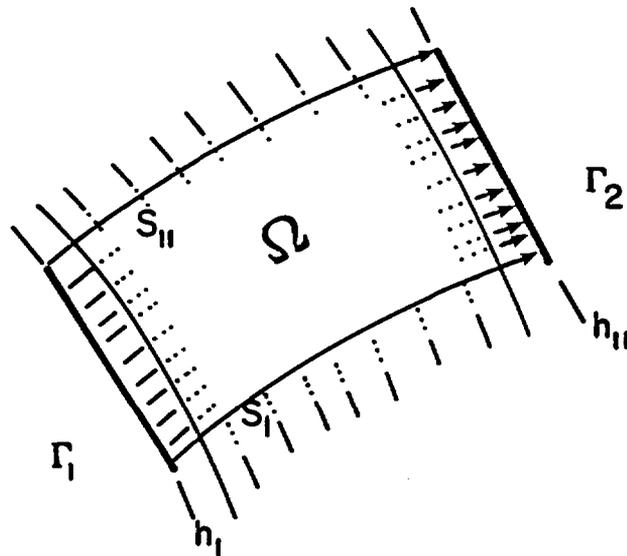


Figure 4.6: Streamline and head contour discretization scheme.

The displacements are found using:

$$(4.11a) \quad dz = \frac{\partial h / \partial x \, ds - \partial s / \partial x \, dh}{\partial h / \partial x \, \partial s / \partial z - \partial h / \partial z \, \partial s / \partial x}$$

and

$$(4.11b) \quad dx = \frac{dh - \partial h / \partial z \, dz}{\partial h / \partial x}$$

where s is the stream function value and h is the potential function value. The values of the changes in (h,s) with respect to (x,z) are determined using:

$$(4.12a) \quad \partial h / \partial z = [h(x, z+dz) - h(x, z)] / dz$$

$$(4.12b) \quad \partial h / \partial x = [h(x+dx, z) - h(x, z)] / dx$$

$$(4.12c) \quad \partial s / \partial z = [s(x, z+dz) - s(x, z)] / dz$$

and

$$(4.12d) \quad \partial s / \partial x = [s(x+dx, z) - s(x, z)] / dx$$

where dx and dz are small relative to the offset desired. Once the location corresponding to the total head and stream of interest is found, the next intersection along the stream line is identified. The velocity between the two intersections is obtained by finding the mean velocity between the two intersections. A harmonic average velocity, \bar{v} , is used:

$$(4.13) \quad \bar{v} = [v(i) v(i-1)] / [v(i) + v(i-1)]$$

4.1.3 Application to Saturated and Variably Saturated Flow

The first application is made to a single square planar fracture with constant unit transmissivity and a unit aperture. Two opposing boundaries are imposed no flow boundaries. The other two opposing boundaries are held at constant heads with a difference in total head equal to the distance between the two boundaries. Because the velocity field is uniform, the expected breakthrough curve for a step injection of a conservative solute assuming no dispersion due to diffusion should be a step function. Figure 4.7 illustrates the flow geometry, boundary conditions, computed potential and stream contours, and the resultant breakthrough curve. Note that the computed and expected breakthrough agree to within one percent except for streamtubes which lie along the boundary of the sample. Due to the approximation function used by the boundary integral method, the largest errors are to be expected in this region. If instead of computing velocities along boundaries using the Newton-Raphson method, the velocities are computed using head differences between nodes, a superior estimate of the breakthrough curve is obtained, the error being reduced to less than three percent.

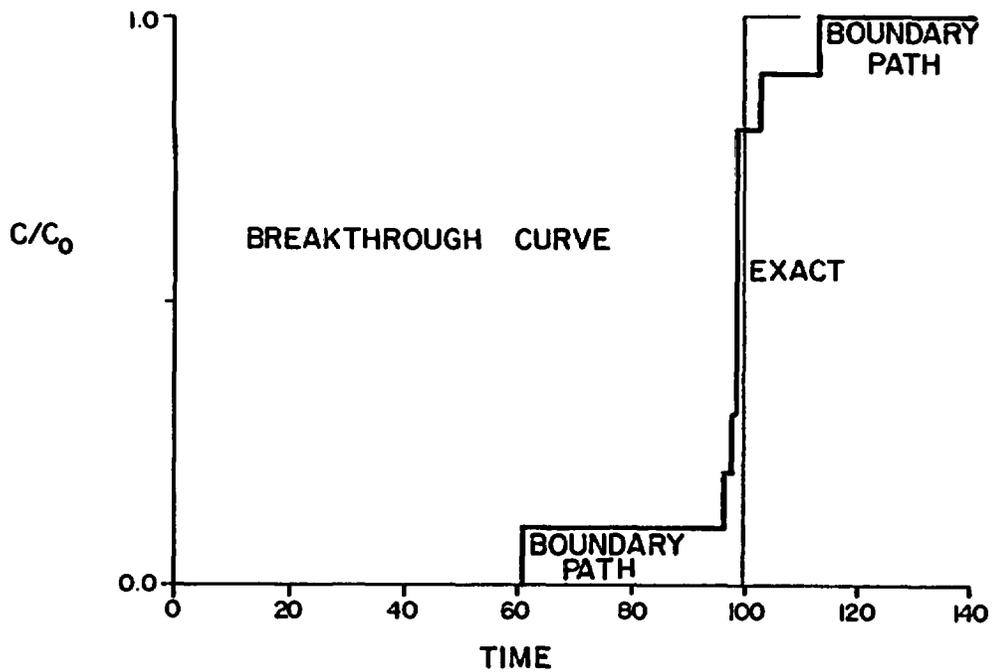
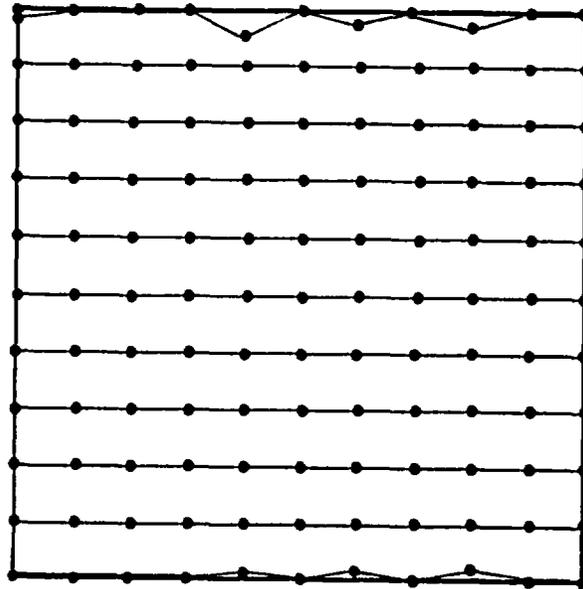


Figure 4.7: Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a square flow domain and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.

Three additional applications are made to a saturated, horizontal fracture, and two vertical, variably saturated fractures. One of the vertical fractures is assigned no capillary head, while the other is assigned a capillary head of 10 m. These applications are identical to those examined in Section 4.3 of the previous chapter. Figures 4.8 through 4.10 illustrate the flow geometries, boundary conditions, computed potential and stream contours, and resultant breakthrough curves for the three applications. Table 4.1 presents the breakthrough curves in tabular form. It is to be noted that substantial tailing of the step injection of a conservative tracer occurs. The tailing results from the two stagnation points located at opposing corners behind the constant head boundaries. Velocities at the stagnation points will be zero and any stream line passing nearby will be affected by a reduction in velocity and a concomitant increase in the travel time. This effect is not noticeable in the simulated problem due to the coarse discretization intervals used.

 Table 4.1: Calculated breakthrough times for streamlines in horizontal and vertical fractures.

Streamline	Horizontal	Vertical
1	274.5	349.2
2	221.6	239.9
3	126.5	144.8
4	99.3	101.0
5	81.6	83.9
6	81.9	80.8
7	90.3	85.7
8	100.5	81.5
9	131.1	87.1
10	248.7	89.3
11	275.1	121.8

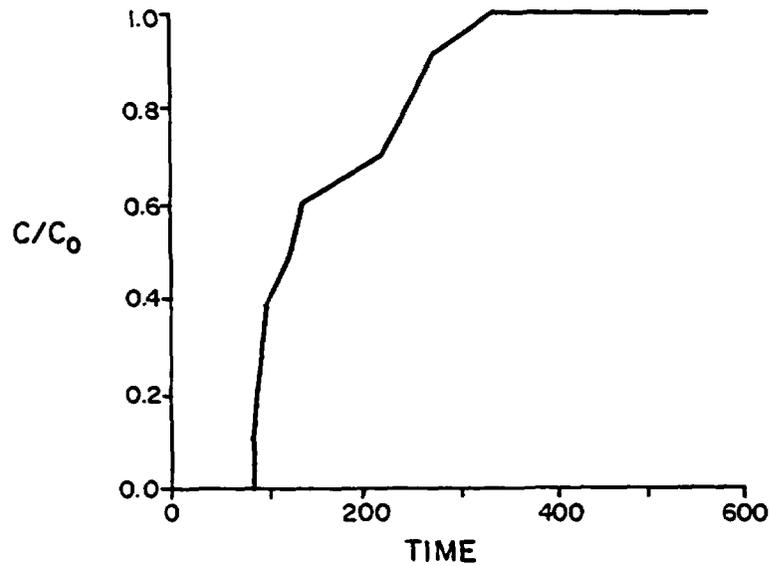
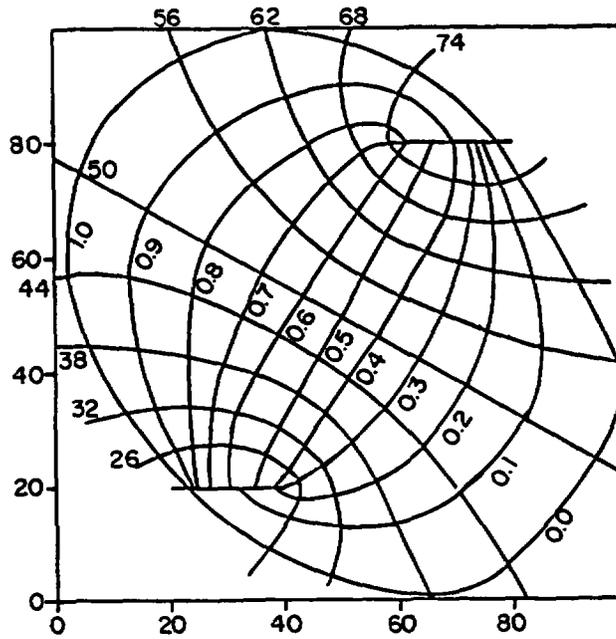


Figure 4.8: Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a horizontal fracture with two intersecting fractures and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.

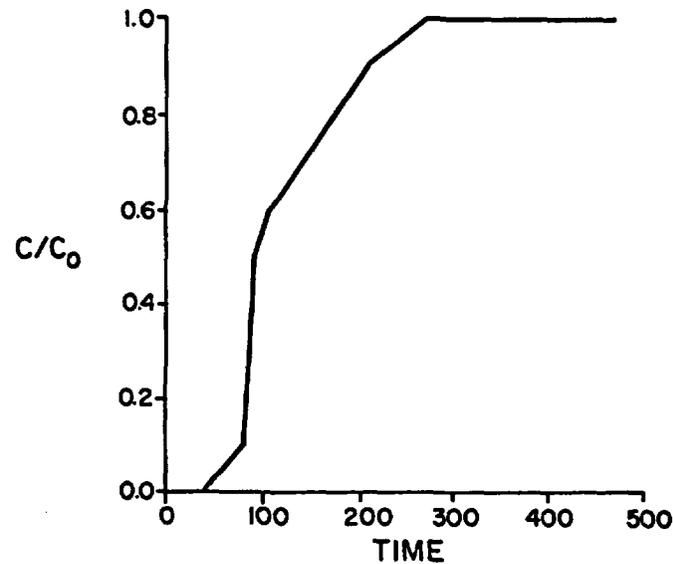
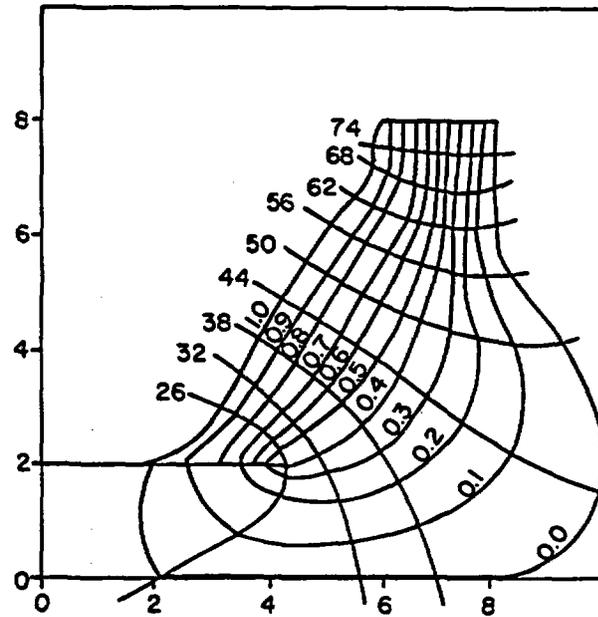


Figure 4.9: Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a vertical fracture with two intersecting fractures and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.

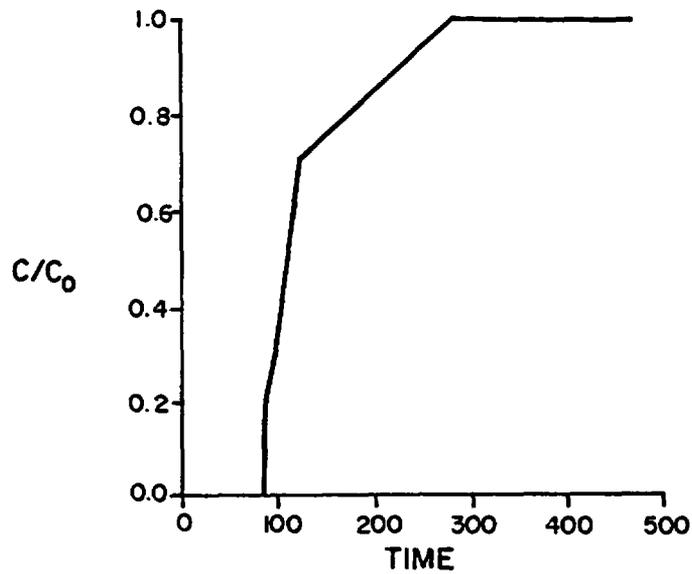
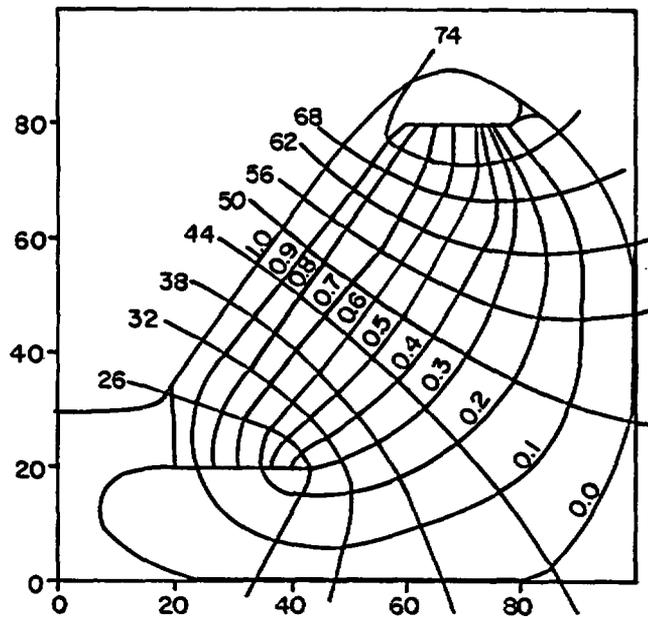


Figure 4.10: Flow geometry, boundary conditions, calculated total head contours, and calculated stream function contours for a vertical fracture with two intersecting fractures and a capillary head of 10 m and calculated breakthrough curves at the outflow boundary for a step injection at the inflow boundary.

4.2 Effects of Sorption and Solute Retardation

Sorption of solutes to fracture surfaces will result in the retardation of the breakthrough curve. At any point the solute velocity can be modeled by assuming that the solute velocity can be linearly related to the fluid velocity using a retardation coefficient:

$$(4.14) \quad v_s = v / R$$

where

- v_s mean solute velocity, m/s;
- v mean fluid velocity, m/s; and
- R retardation factor, dimensionless.

The use of a retardation coefficient to model sorption onto fracture surfaces is appropriate when fast reversible adsorption is present with a linear isotherm (Jennings, 1987). It is interesting to note the behavior of the retardation coefficient:

- o If the solute is conservative (i.e., no sorption) the coefficient is set to one;
- o If the solute is subject to instantaneous, reversible sorption, the coefficient is set a number greater than one;
- o If the solute is excluded from boundary layers where the fluid velocity is less than the mean velocity (such as due to anion exclusion), the coefficient may be less than one.

For slow reversible adsorption with a linear isotherm, a first-order sorption rate constant can be used to model the process (Valocchi, 1986):

$$(4.15) \quad M_s = K (C_f - C_m)$$

where

- M_s mass of solute sorbed per unit time, kg/s;
- K sorption rate constant, kg/s;
- C_f concentration of the solute in the fracture fluid, dimensionless; and
- C_m concentration of the solute sorbed onto the matrix, dimensionless.

When sorption and desorption occur at different rates, such as for irreversible reactions, two rate parameters are required:

$$(4.16a) \quad M_s = K^+ (C_f - C_s)$$

and

$$(4.16b) \quad M_s = K^- (C_f - C_s)$$

where K^+ and K^- are sorption and desorption rate constants, kg/s, respectively. The process of linear instantaneous sorption can be modeled using the retardation coefficient which results from a Freundlich isotherm:

$$(4.17) \quad S = K_d C^n$$

where

- S mass of solute species adsorbed or precipitated on the solids per unit fracture area, kg/m^2 ;
- K_d distribution coefficient, or the mass of solute on the solid phase per unit fracture area divided by the concentration of solute in solution, m;
- C solute concentration, kg/m^3 ; and
- n exponent, equal to one for a linear isotherm, dimensionless.

The retardation coefficient is applied in the computer simulation model by assuming that the retardation coefficient is everywhere constant within the medium and that the arrival of a concentration change at the point of observation can be shifted by dividing the travel time by the retardation factor. In this case, Equation (4.14) is modified to yield:

$$(4.18) \quad v_s = v / R = q / (e R)$$

where

- v_s solute velocity, m/s;
- v fluid velocity, m/s; and
- R retardation coefficient, dimensionless, equal to:

$$(4.19) \quad R = 1 + A K_d$$

where A is the surface area to volume ratio for the fracture, $1/\text{m}$. For a planar fracture:

$$(4.20) \quad A = 1 / e$$

so that Equation 4.19 becomes:

$$(4.21) \quad R = 1 + K_d / e$$

This formulation results in a delay of the breakthrough curve. Figures 4.11 through 4.13 present simulated breakthrough curves for the horizontal and vertical fractures described in Section 3.2.2 using a retardation coefficient of 2. Note that the solution is a trivial case of doubling the travel time between the influent and effluent nodes.

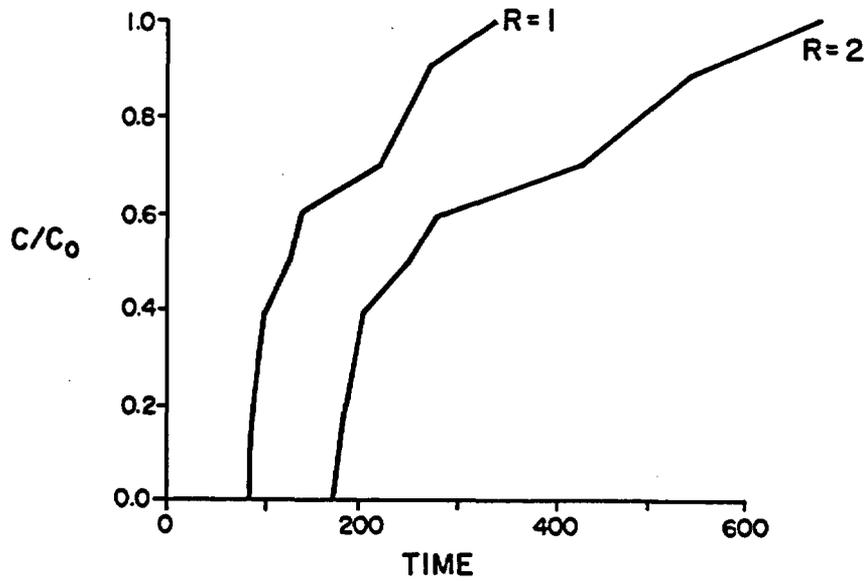


Figure 4.11: Calculated breakthrough curves at the outflow boundary for the flow domain of Figure 4.8 with a retardation coefficient of 2.

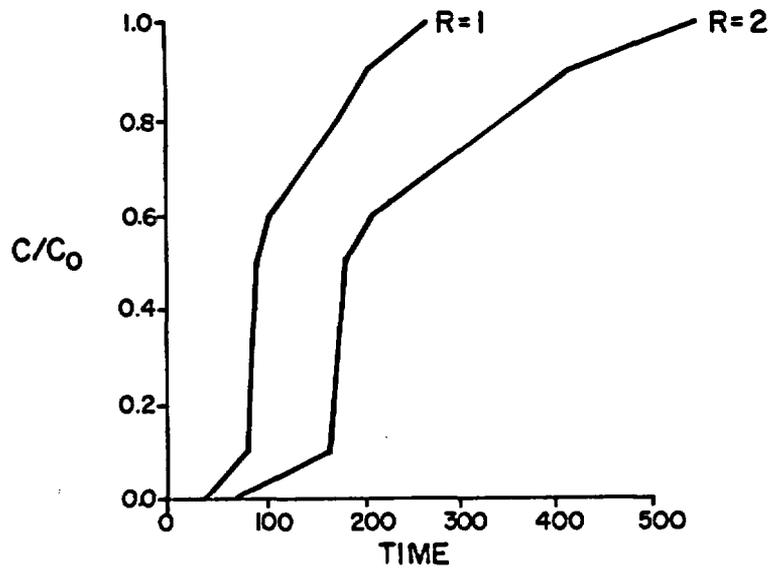


Figure 4.12: Calculated breakthrough curves at the outflow boundary for the flow domain of Figure 4.9 with a retardation coefficient of 2.

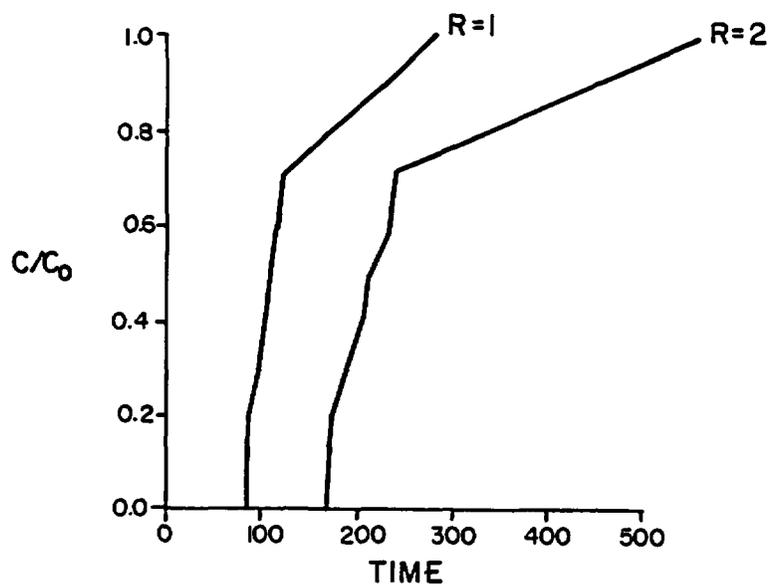


Figure 4.13: Calculated breakthrough curves at the outflow boundary for the flow domain of Figure 4.10 with a retardation coefficient of 2.

4.3 Effects of Matrix Diffusion

Matrix diffusion is the movement of a solute into a rock matrix from a fracture which can be described using Fick's first law (Grisak and Pickens, 1981). Loss of solute into the matrix can not be modeled using the retardation coefficient because the assumption of instantaneous equilibration is violated. Instead, a decaying rate of loss is generally observed. By assuming a flow system, illustrated in Figure 4.14, in which a step injection of solute is made into a fracture, an analytic relationship between solute concentrations as a function of time and distance along the fracture and matrix porosity and diffusivity can be established (Grisak and Pickens, 1981):

$$(4.22) \quad a(x,y,t) = C(x,y,t) / C_0 \\ = \text{erfc}[\{(\theta D^*/ve)x + y/2\} / (D^*(t-x/v))^{1/2}]$$

where

- $a(x,y,t)$ attenuation coefficient, dimensionless;
- $C(x,y,t)$ solute concentration at any point (x,y) within the matrix or fracture at time $(t > x/v)$, dimensionless;
- C_0 initial solute concentration, $C(0,0,t)$ for $\forall t$, dimensionless;
- $\text{erfc}(x)$ complementary error function, $1 - \text{erf}(x)$, dimensionless;
- θ effective matrix porosity, dimensionless;
- D^* effective molecular diffusion coefficient of the solute, m^2/s ;
- v mean fluid velocity in the fracture, m/s ;
- x distance from source, parallel to fracture, m ;
- y distance from source, perpendicular to fracture, m ; and
- t time from beginning of injection, s .

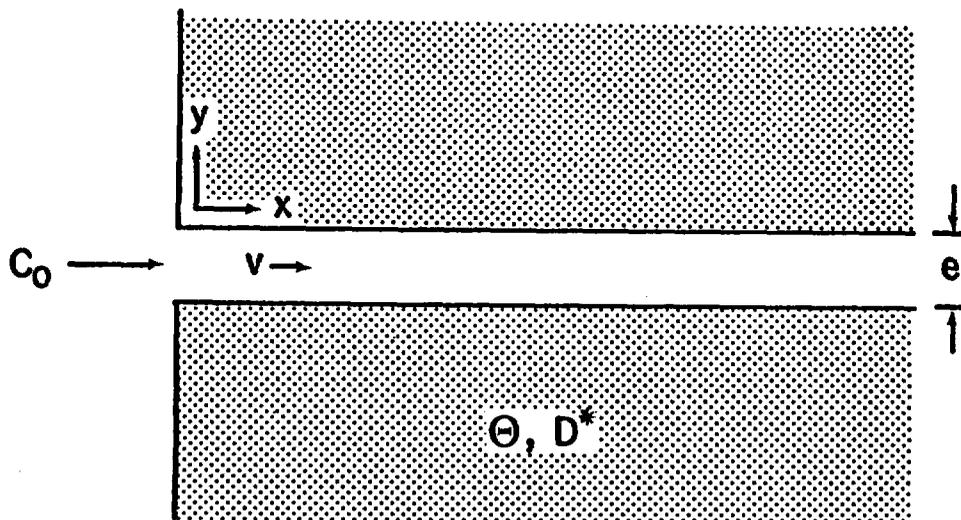


Figure 4.14: Conceptual model of flow through an individual fracture of semi-infinite areal extent.

Of interest to this analysis is the behavior of the solute concentration at a point within a fracture as a function of time. If the rate of change of the solute concentration is low, then a constant solute concentration can be modeled using the attenuation coefficient. This will only occur if the attenuation is very nearly a constant, i.e.:

$$(4.23) \quad \partial a(x, y, t) / \partial t = 0$$

To determine under what conditions Equation 4.23 is valid, an analytic expression is derived. Combining Equations 4.22 and 4.23 yields:

$$(4.24) \quad \partial \operatorname{erfc}[\left(\frac{\Theta D^*}{ve}x + y/2\right) / (D^*(t-x/v))^{1/2}] / \partial t = 0$$

This derivative can be calculated by noting that (Abramowitz and Stegun, 1972, Eq. 7.2.8):

$$(4.25) \quad \partial(i^n \operatorname{erfc}(z)) / \partial z = -i^{(n-1)} \operatorname{erfc}(z) \quad (n = 0, 1, \dots)$$

where i is the square root of minus one. In our case, $n = 0$. Using op. cit. (Eq. 7.2.1):

$$(4.25) \quad \partial(\operatorname{erfc}(z))/\partial z = -1/i \operatorname{erfc}(z) = -2 \exp(-z^2) / \pi^{1/2}$$

By also noting that:

$$(4.26) \quad \partial a / \partial t = \partial(\operatorname{erfc}(z)) / \partial z \partial z / \partial t$$

We can substitute Equation 4.25 into Equation 4.26 to obtain:

$$(4.27) \quad \partial a / \partial t = [-2 \exp(-z^2) / \pi^{1/2}] \partial z / \partial t$$

We can find the derivative of z with respect to time by noting that:

$$(4.28) \quad z = \left(\frac{\Theta D^*}{ve}x + y/2\right) / (D^*(t-x/v))^{1/2}$$

Taking the derivative yields:

$$(4.29) \quad \partial z / \partial t = -z / 2 (t - x/v)$$

Substituting Equation 4.29 into Equation 4.27 results in:

$$(4.30) \quad \partial a / \partial t = z \exp(-z^2) / [\pi^{1/2} (t - x/v)]$$

where z is given by Equation 4.28. Equation 4.30 relates the rate of change of the attenuation coefficient at any point within the fracture or matrix to physical properties.

Sensitivity analyses can be used to determine under what conditions the attenuation coefficient is constant with respect to time. Such a sensitivity analysis is presented in Figures 4.15 through 4.22. For all the analyses an effective solute diffusion coefficient of $0.003 \text{ m}^2/\text{yr}$ and an aperture of 100 um was used. The relative concentration and the time rate of change of the relative concentration are plotted as functions of both distance and time.

Figure 4.15 shows the movement of a solute pulse contained in a solvent moving at a rate of 10 m/yr through a fracture bounded by an impervious matrix. Note that the fluid pulse is a step function for all distances and times. Figures 4.16 and 4.17 show the movement in both space and time for a solvent moving at a rate of 10 m/yr through a fracture bounded by a matrix with an effective porosity of 0.001. Note here that the sharp front of the solute has been attenuated. Note also that the time rate of change of the relative concentration is highest near the leading edge of the solute step function, and becomes minor once the leading edge of the step function moves past the observation point. Figures 4.18 and 4.19 are for identical parameters as for the previous figures, except that the matrix porosity is increased to 0.01. Again, the time rate of change of the relative concentration is small except in the proximity of the leading edge of the solute step function. Figures 4.20 and 4.21 present results for a matrix porosity of 0.1. Figure 4.22 presents results for a flow velocity increased to 100 m/yr for through a matrix with an effective porosity of 0.1. Of interest is the observation that the time rate of change of the relative concentration is very high at early time, but is diminishing at later times. From these limited scenarios, it is demonstrated that under specific conditions the mass flux into the rock matrix is very nearly a constant for short time intervals, allowing the use of an attenuation factor which is variable in space but constant in time.

Once the limitations of the use of the solute attenuation coefficient have been determined, the effect of matrix diffusion can be investigated by assuming slow diffusion into an immobile liquid phase in the rock matrix (Rasmussen, 1982). Matrix diffusion is modeled assuming that a constant solute flux into the rock matrix exists for short time intervals. A reduction of the solute concentration is calculated by using a convolution summation to calculate the attenuation coefficient at each time step. The convolution is:

$$(4.31) \quad a(x,t) = \sum_{i=0}^t a(x,i) C(x_0,t-i)$$

where a is the time and spatially dependent solute attenuation coefficient which accounts for diffusion into the rock matrix, dimensionless, obtained from Equation 4.22. As demonstrated in the sensitivity studies, above, Equation 4.31 will not accurately calculate the breakthrough curve when the attenuation coefficient changes quickly over time. To evaluate whether the coefficient is constant, the value of the derivative of the constant with respect to time should be small. The derivative is calculated using Equation 4.30. As long as the derivative is small, there is no need to update the matrix diffusion attenuation coefficient within each time step.

Figure 4.23 presents simulation results for three values of the parameter $\theta(D^*)^{1/2}$. The figures were generated using the square flow domain of Figure 4.7 at two velocities, 0.08 m/s, and 0.16 m/s over a distance of 100 m in response to a step injection of solute at the inflow boundary.

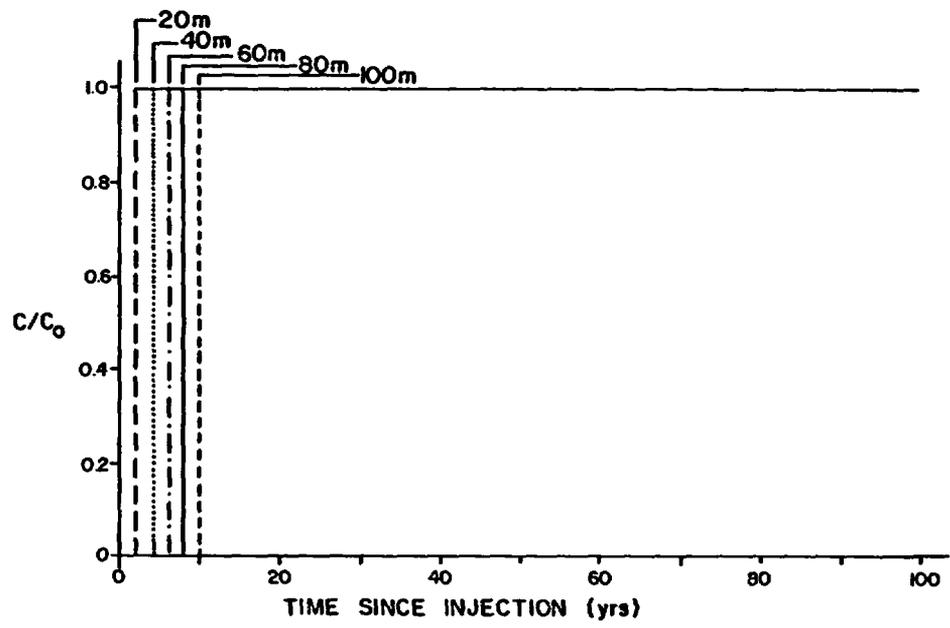
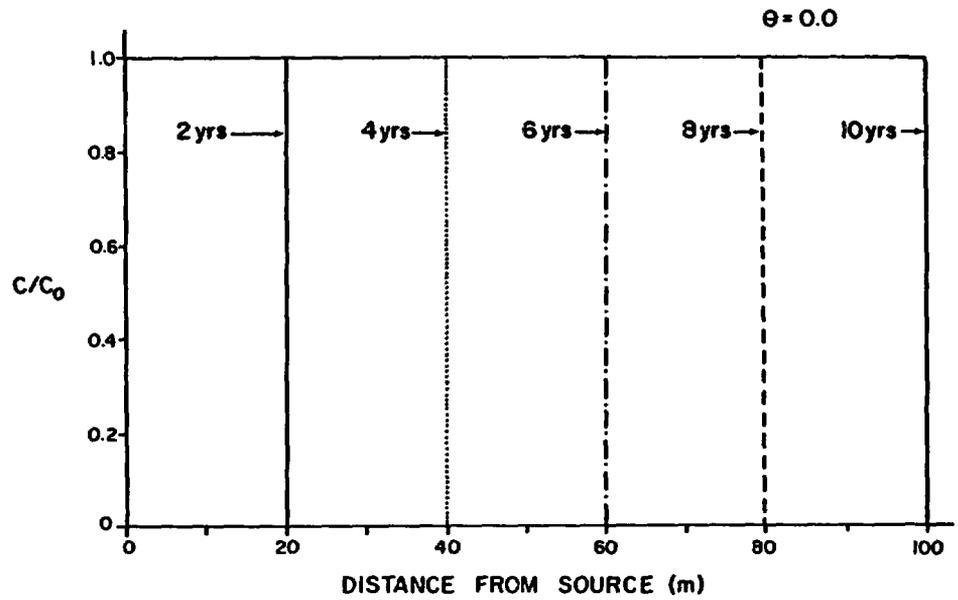


Figure 4.15: Analytic solutions of distance and time plots of solute concentration resulting from a step inflow of solute past an impermeable rock matrix.

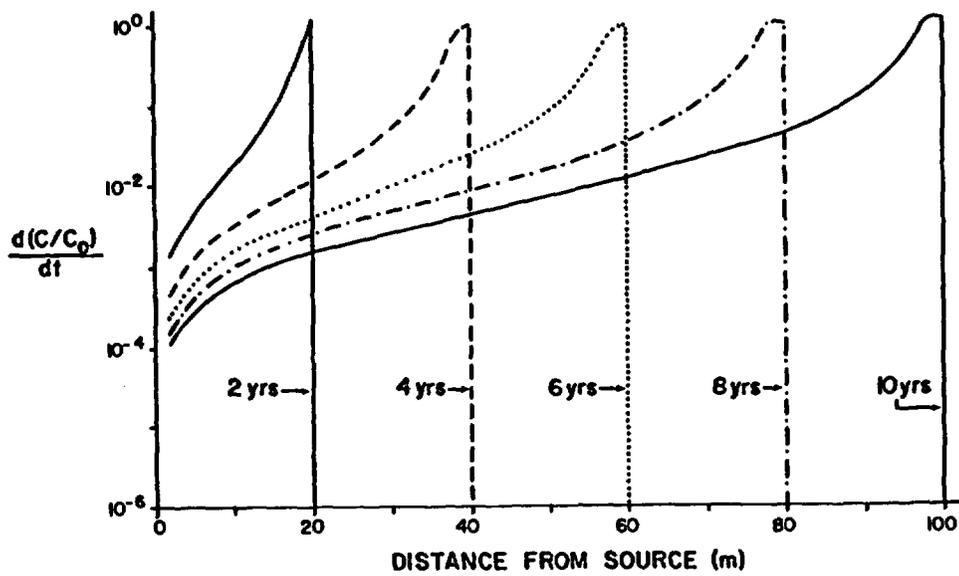
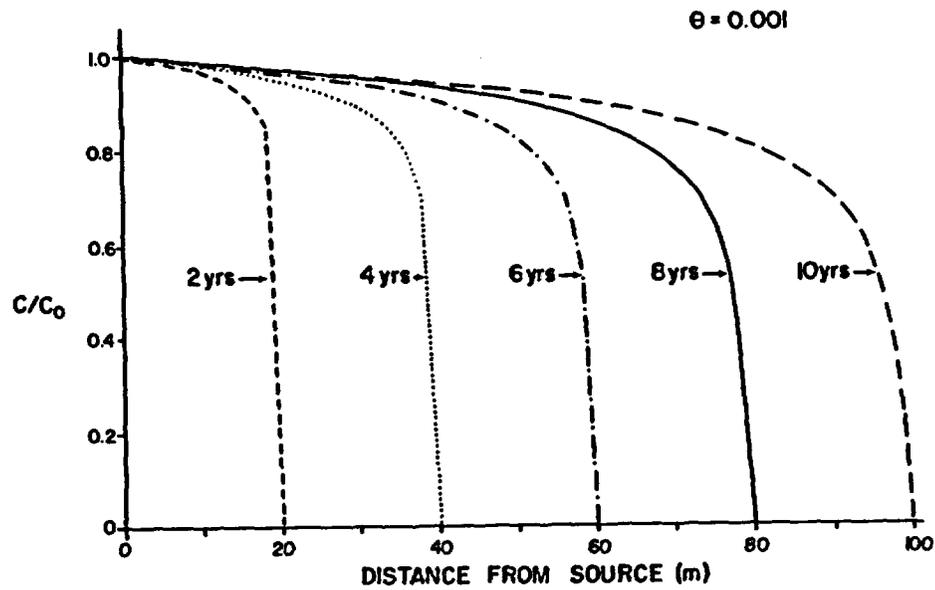


Figure 4.16: Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with a porosity of 0.001.

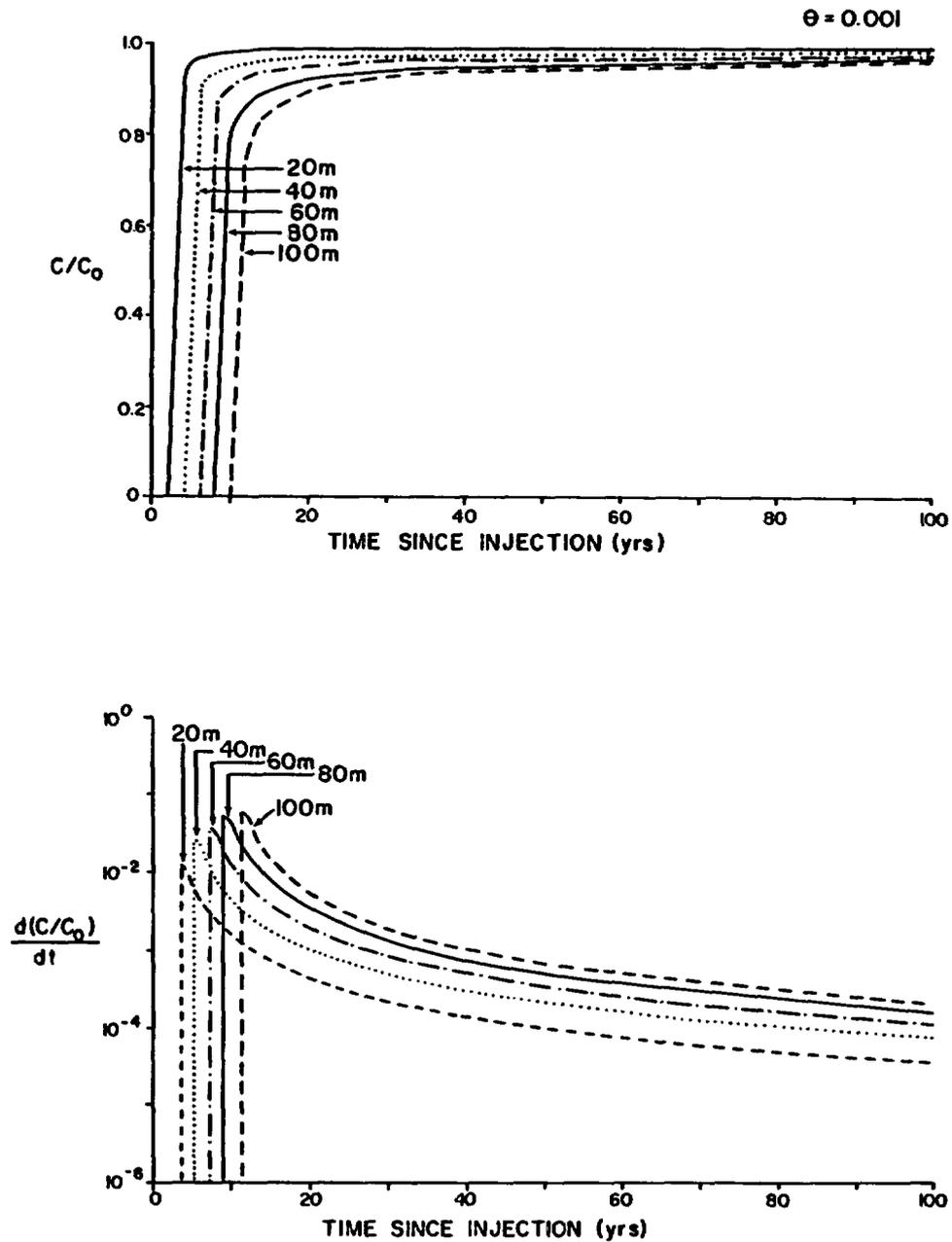


Figure 4.17: Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of time since injection of a step inflow of solute past a rock matrix with an effective porosity of 0.001.

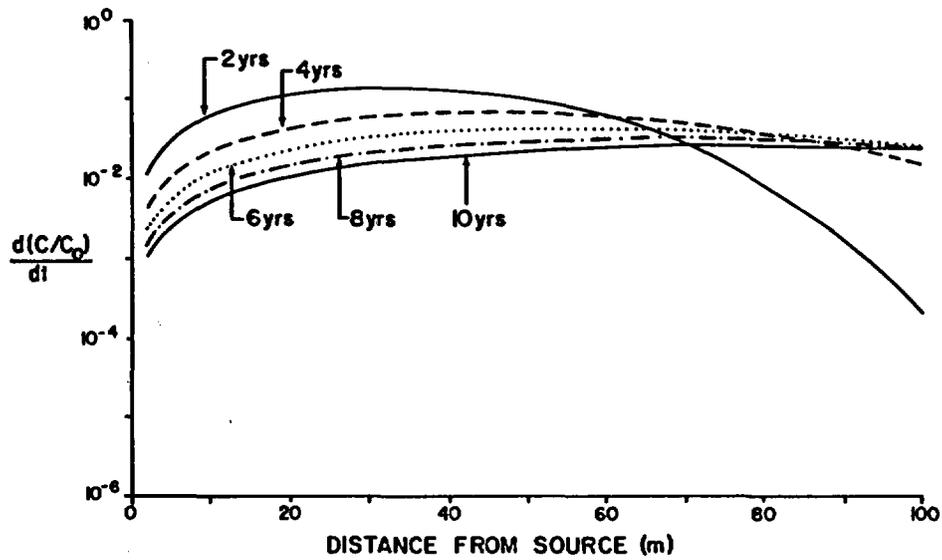
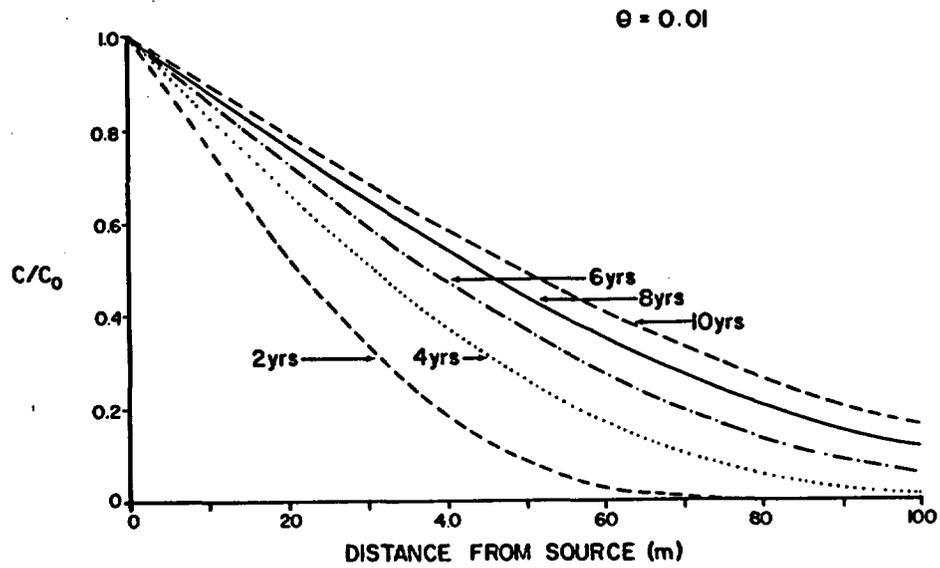


Figure 4.18: Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with a porosity of 0.01.

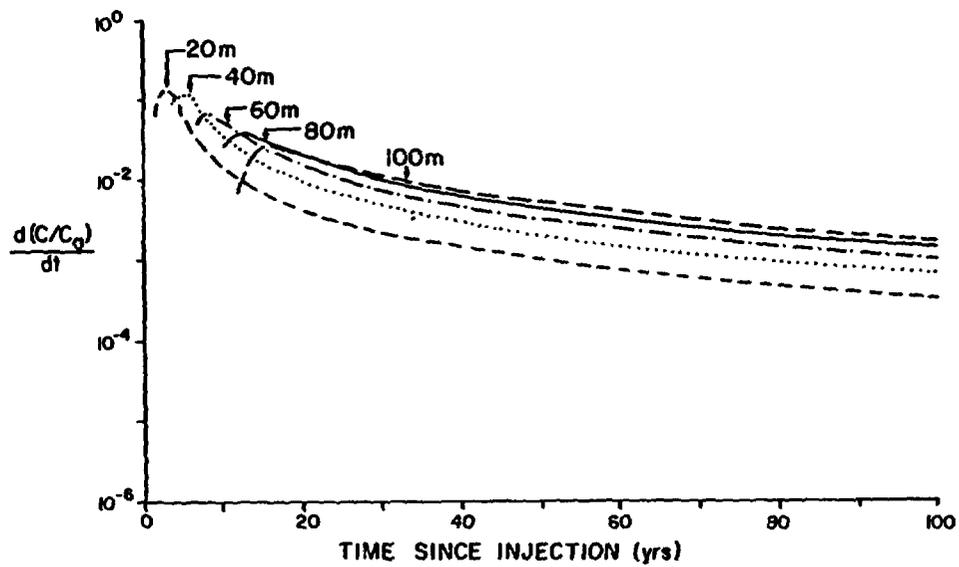
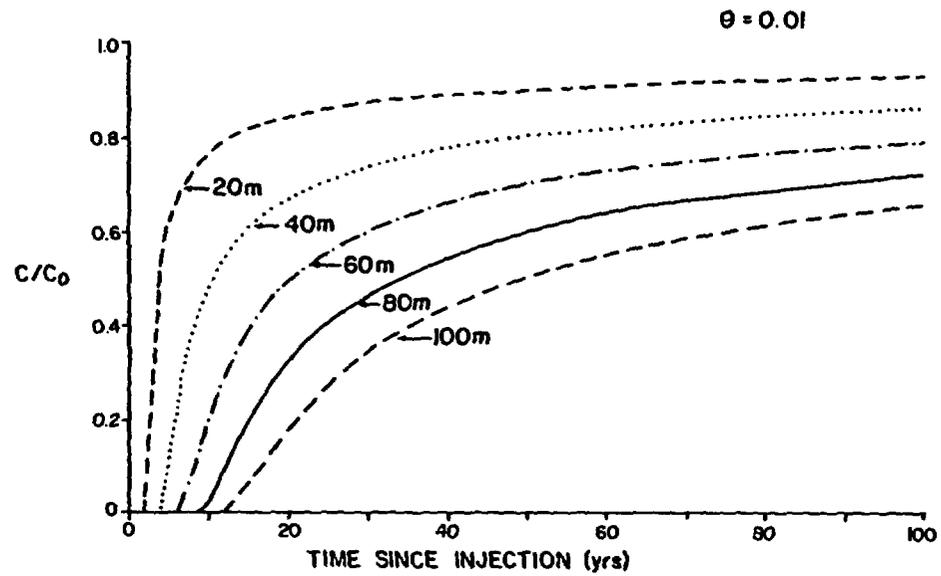


Figure 4.19: Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of time since injection of a step inflow of solute past a rock matrix with an effective porosity of 0.01.

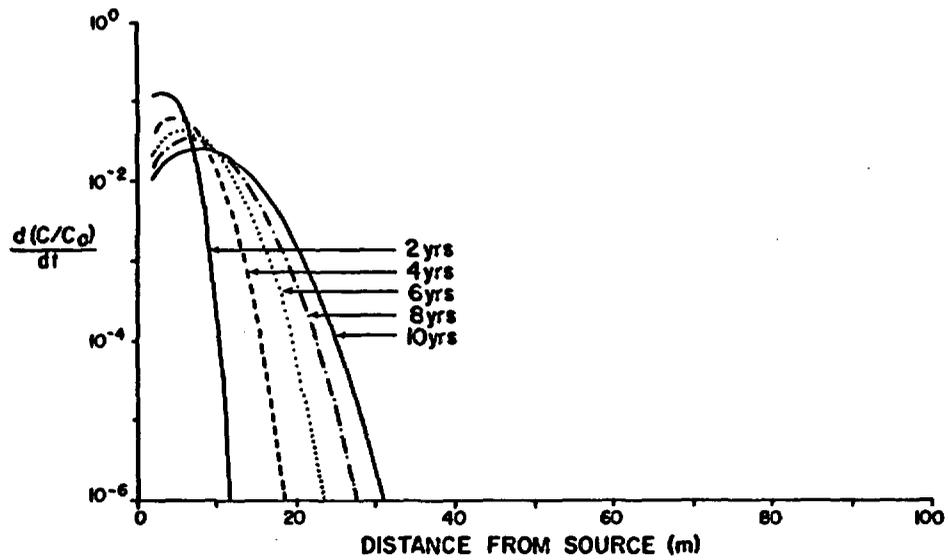
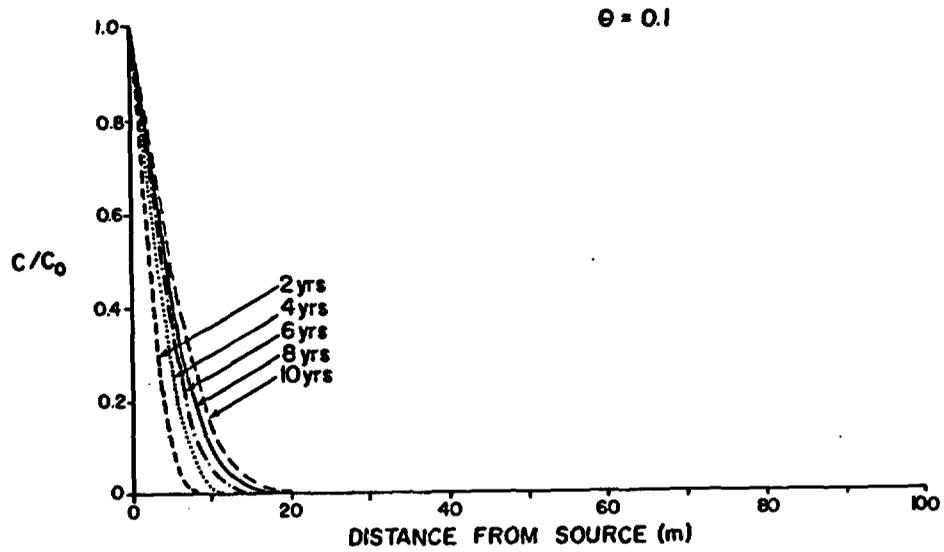


Figure 4.20: Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with a porosity of 0.1.

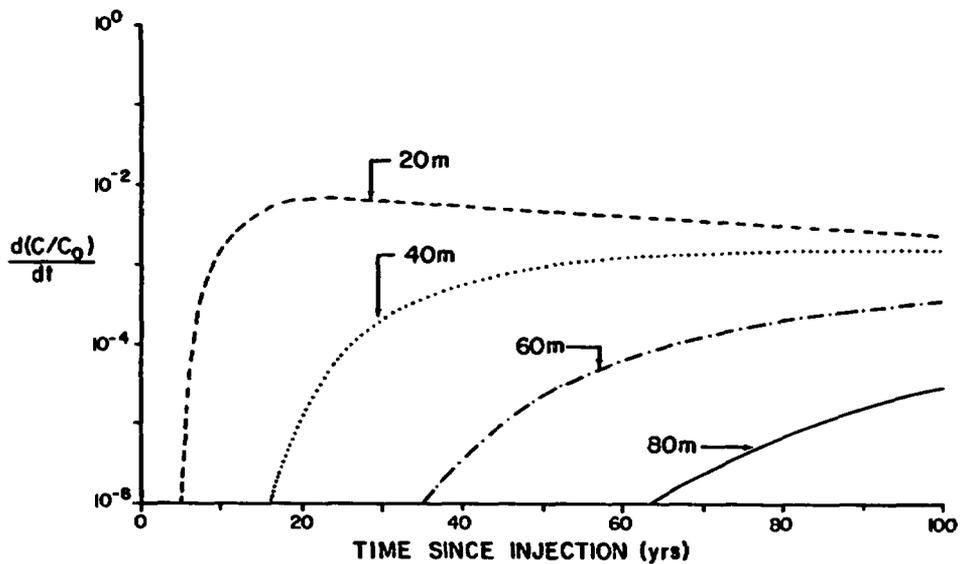
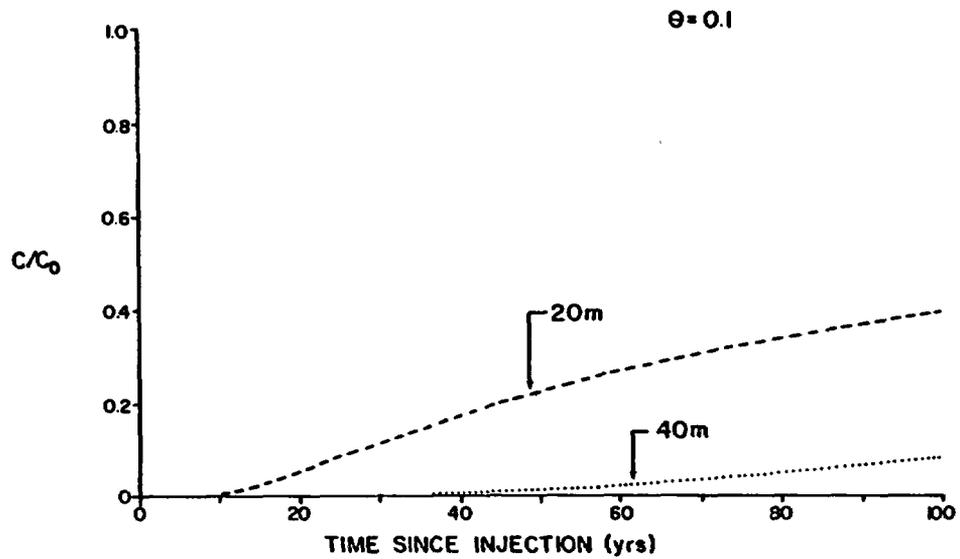


Figure 4.21: Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of time since injection of a step inflow of solute past a rock matrix with an effective porosity of 0.1.

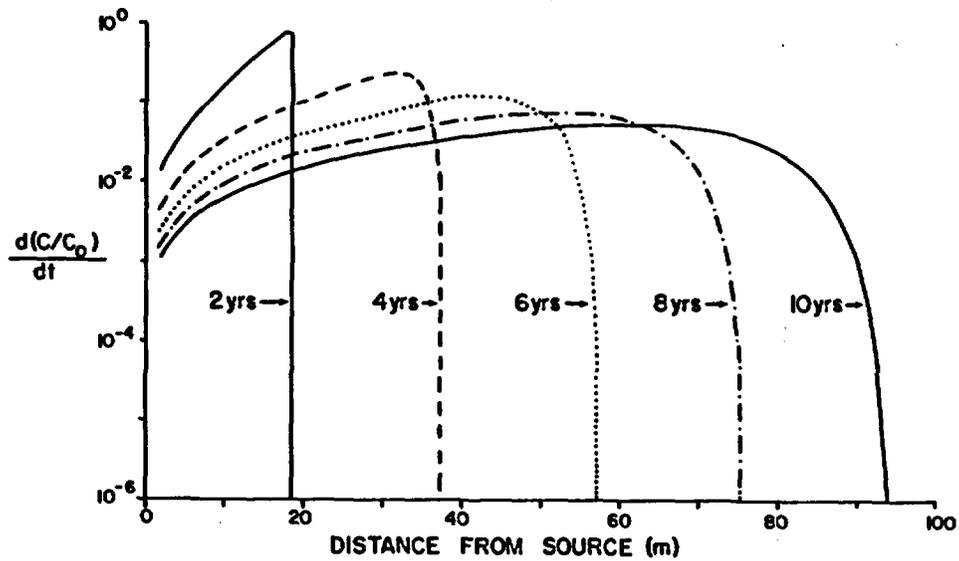
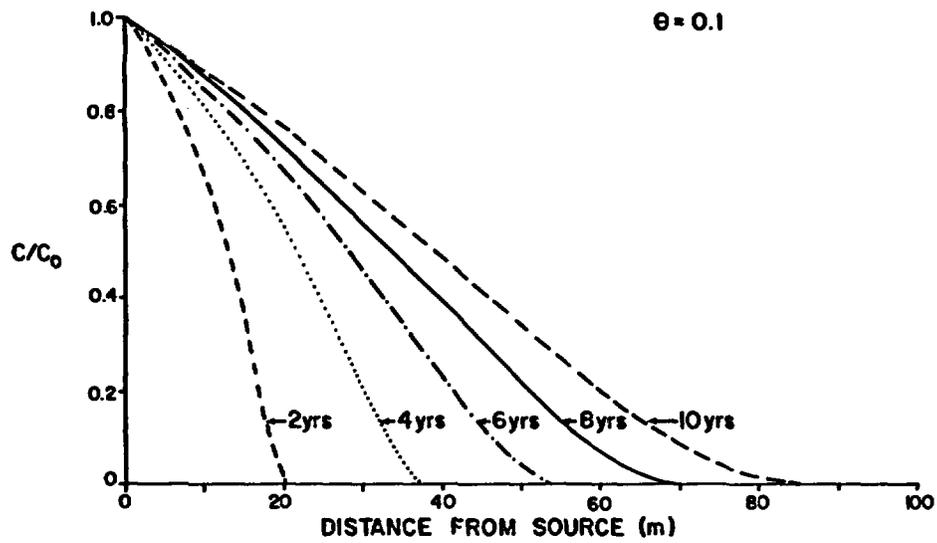


Figure 4.22: Analytic solutions of solute concentration and the time rate of change of the solute concentration as a function of distance from the source resulting from a step inflow of solute past a rock matrix with an effective porosity of 0.1 and an increased velocity from 10 to 100 m/yr.

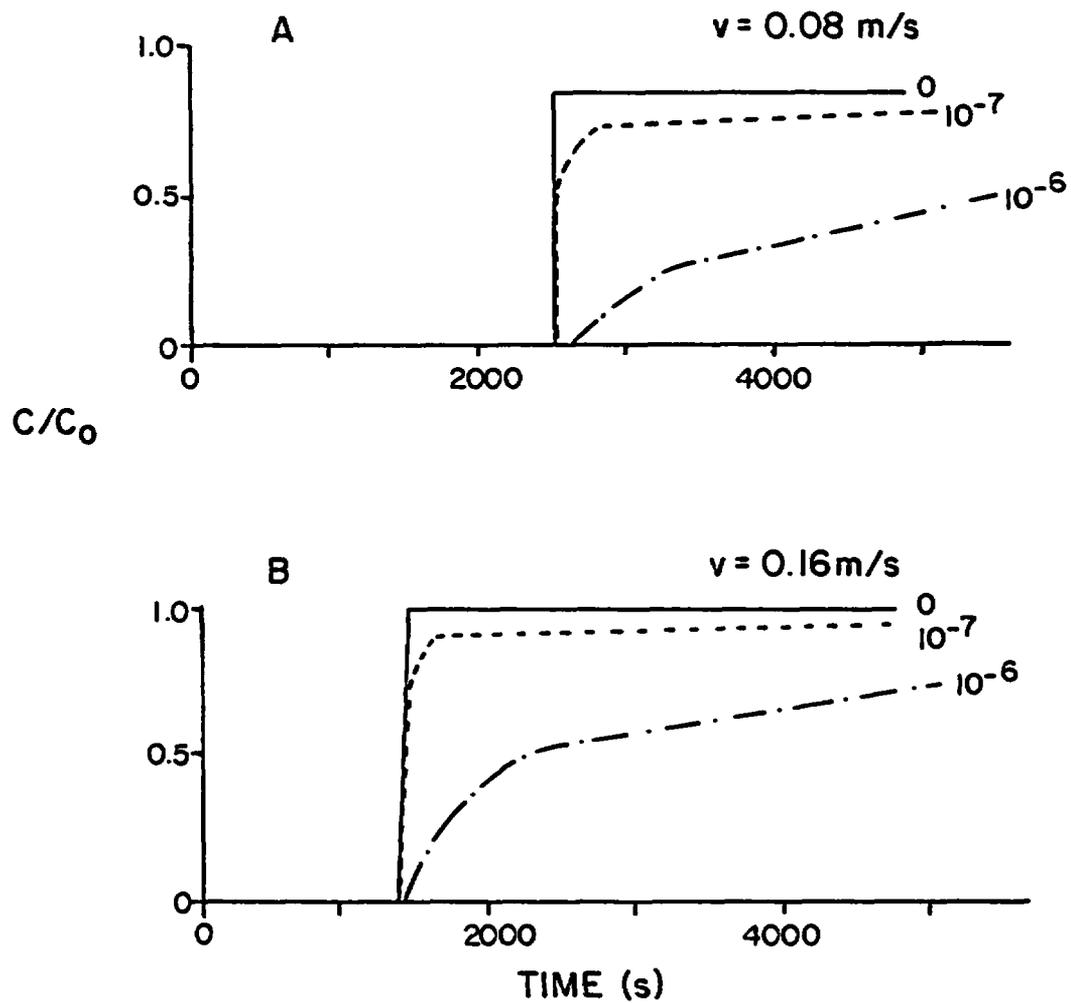


Figure 4.23: Relative concentration at outflow surface of flow domain shown in Figure 4.7 in response to a step injection of tracer for three values of the parameter $\theta(D^*)^{1/2}$ at two velocities, 0.08 m/s (A) and 0.16 m/s (B).

CHAPTER 5

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary

Flow and transport through fractured media has been variously simulated in other studies by using equivalent porous media models, dual (or double) porosity models, and discrete fracture network models. The modeling of discrete fracture networks has progressed from one-dimensional bundle of tubes models, to two-dimensional fracture networks, and recently to fully three-dimensional simulation models. The model presented in this work uses the discrete fracture network formulation to generate fractures within a three dimensional space, and to solve both a two-dimensional flow field within a fracture network embedded in an impermeable matrix, as well as a three-dimensional flow field using fractures embedded in a permeable matrix. Extensions to unsaturated flow and solute transport are made assuming an impermeable, porous matrix.

Flow and transport processes through the fracture network are subdivided into three components, termed intra-fracture, inter-fracture, and supra-fracture processes. Intra-fracture processes occur within individual fractures, while inter-fracture processes result from flow and transport through networks of discrete fractures. Supra-fracture processes occur as the result of interactions with the rock matrix. Such interactions include matrix diffusion and retardation, and flow through the matrix. The quantification of the components proceeds from the demonstrated ability to solve a flow problem within an individual fracture incorporating unsaturated conditions and solute breakthrough curves, to the ability to solve a flow problem through a network of intersecting fractures, and finally to a problem incorporating flow through both a fracture network and the matrix surrounding the fractures.

The simulation of flow and transport is based on the boundary integral method which provides a methodology for discretizing the boundary of each fracture and to solve for fluid flow within the two dimensional fracture flow domain. Extensions to combined matrix and fracture flow are made. Fracture network hydraulic conductivity is calculated by stochastically generating a network of fractures. Each fracture within the network is defined by its center, orientation, lengths in two directions, and transmissivity. The intersections between fractures are found numerically, and mass balance equations are used to determine the flow between intersections. The mass balance equations are solved for specified boundary conditions. The proposed models employ Darcy's law to relate fluid flow to a potential field, capillary theory to relate fracture saturation to a potential, and the concept of stream lines to calculate travel times. The models are implemented using computer code which is executable on a PC-compatible micro-computer.

5.2 Conclusions

Sensitivity studies have been performed to evaluate the influence of specific fracture parameters on network hydraulic conductivity, as well as on solute breakthrough curves. From these studies the following conclusions are made:

- o Steady fluid flow through individual fractures can be calculated using the boundary integral method for various boundary geometries. Calculated errors vary from 0.06 to 0.30 percent for fine and coarse discretization schemes, respectively.
- o Steady flow through serial fractures can be calculated with a maximum error of 0.20 percent for a coarse discretization scheme.
- o Steady flow through a fracture with internally intersecting fractures is demonstrated. A maximum 1.0 percent error was present for simulations where an analytic solution was available.
- o Flow through fracture networks containing fractures of infinite length is linearly dependent on fracture density (i.e., fracture length per unit area normal to the hydraulic gradient) and fracture transmissivity.
- o The effect of correlation between fracture length and fracture transmissivity on estimated network hydraulic conductivity indicates that increasing the correlation between fracture length and fracture transmissivity results in an increased network conductivity for all four realizations examined.
- o A three dimensional boundary integral equation is shown to accurately represent combined flow through both a permeable matrix and an embedded fracture. Estimated global hydraulic conductivities are calculated for variable fracture to matrix permeability ratios. It is shown that estimated errors increase as the aspect ratio increases.
- o Extension of the boundary integral method to unsaturated fracture flow is performed by assuming a constant capillary potential within individual fractures. A mobile (or moving) interface formulation is used to position the air-water interface along the zero pressure head surface. A Newton-Raphson scheme is used to locate nodal positions. Comparison of simulation results is made with an analytic solution, along with laboratory results obtained from a flow visualization experiment.
- o Simulations of unsaturated flow through a fracture network demonstrates the presence of zones of water under large positive pressures, along with zones of air-filled voids.
- o Retardation and matrix diffusion are shown to delay and to attenuate, respectively, solute breakthrough curves. The retardation coefficient is used to model instantaneous sorption, while matrix diffusion is used to model slow diffusion into a porous rock matrix.

5.3 Recommendations

Based upon demonstrated modeling capabilities and the sensitivity studies which have been performed, the following recommendations are made:

- o It is recommended that calibration of the model be performed using field tests and comparison with porous media and analytic stochastic models. Once calibration has been performed, the computer simulation model can be used to provide parameter sensitivity estimates for flow through a fractured rock mass of arbitrary matrix permeability.
- o Additional needs for model calibration include the determination of the relation between pressure heads in fractures and the unsaturated fracture hydraulic conductivity. Such tests are required to validate the hypothesized relationship between hydraulic conductivity and pressure head. The testing of travel times and breakthrough curves within fracture networks at various pressure heads need to be performed for model validation. Observations for fracture length estimation also need to be performed to determine field-scale values of this parameter.
- o It is recommended that several features be included in the methodology to more completely characterize fluid flow and solute transport through variably-saturated fractured rock. In particular, a two-dimensional representation of fracture transmissivity variation within the plane of the fracture should be included. The formulation of the two-dimensional representation of an fracture transmissivity would more completely account for head distributions within the plane of the fracture, as well as tortuosity, streamlines and dispersivity coefficients.
- o An additional feature which should be incorporated within the methodology is the computation of breakthrough curves resulting from flow between the rock matrix and the fractures. For networks of fractures within a rock mass at large negative pressure heads, flow will predominantly occur through the matrix, drained fractures serving to inhibit the flow of water from one block to another.
- o The inclusion of transient fluid flow should also be examined. The effects of pulsed, step, and cyclic boundary conditions should be evaluated to determine the effect of fractured rock on peak flow rates, especially with respect to depth below a source.

APPENDIX A

COMPUTER SIMULATION MODELS

This appendix presents four computer simulation models used in this study for investigating the effects of geometric properties of fractures on fluid flow and solute transport through variably saturated fractured rock. Simulation inputs and outputs are presented, along with source code listings of the programs. It should be noted that the programs are being continually updated as improved algorithms are implemented. Updated versions of the programs, as well as source code on PC-compatible floppy disks are available from the author. The programs are written in FORTRAN-77 for implementation on a Definicon DSI-32 68000 processing board, but can be used in any environment supporting FORTRAN-77 with minor modifications.

The programs presented here were prepared in support of research activities and are not intended for other uses. Neither the United States Government, the University of Arizona nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any portion of these programs or represents that its use by such third party would not infringe privately owned rights.

A.1 BIM: Fluid Flow Solver Using the Boundary Integral Method

This section presents the data entry requirements, sample data output, and the program listing for a computer simulation model used to model saturated and unsaturated flow through networks of discrete fractures. Table A.1 presents the opening menu, indicating the various options available. Option 1 requires an input data file, an example of which is provided in Table A.2. Options 3 through 6 provide output data files, an example of which is presented as Table A.3 for the sample input of Table A.2. Options 7 and 8 store and retrieve intermediate output, and should be executed after or instead of data entry, respectively. Option 9 is implemented in order to return to the operating system. A listing of the program is presented following Table A.3.

.....
Table A.1: Opening menu for program BIM
.....

BOUNDARY INTEGRAL METHOD

Implemented by Todd C. Rasmussen
Department of Hydrology, Univ. of AZ

- 1 - Input from Data File
- 2 - Display Boundary Equation Matrix
- 3 - Find Potentials and Streamlines at Boundary Points
- 4 - Find Potentials and Streamlines at Interior Points
- 5 - Find Free Surface
- 6 - Calculate Breakthrough Curves
- 7 - Write Backup File
- 8 - Read Backup File

- 9 - Exit to DOS

Enter Selection:
.....

Table A.2: Sample input data for program BIM.

1			
1	24		
5.			
100.	0.	1	
100.	20.	1	
100.	40.	1	
100.	60.	1	
100.	80.	1	
100.	100.	1	
99.99	100.		100.
80.	100.		100.
60.	100.		100.
40.	100.		100.
20.	100.		100.
0.01	100.		100.
0.	100.	1	
0.	80.	1	
0.	60.	1	
0.	40.	1	
0.	20.	1	
0.	0.	1	
0.01	0.		0.
20.	0.		0.
40.	0.		0.
60.	0.		0.
80.	0.		0.
99.99	0.		0.

Table A.3: Sample data output from Option 3 from program BIM using input data from Table A.2.

i	n	x	y	u	q	d	t
1	1	100.000	.000	.044	.000	-.006	-.025
1	2	100.000	20.000	19.997	.000	.000	-.025
1	3	100.000	40.000	39.999	.000	.000	-.025
1	4	100.000	60.000	60.001	.000	.000	-.025
1	5	100.000	80.000	80.003	.000	.000	-.025
1	6	100.000	100.000	99.956	.000	.006	-.025
1	7	99.990	100.000	100.000	5.007	50.045	.000
1	8	80.000	100.000	100.000	5.000	99.986	100.016
1	9	60.000	100.000	100.000	5.000	99.997	200.011
1	10	40.000	100.000	100.000	5.000	99.997	300.009
1	11	20.000	100.000	100.000	5.000	99.986	400.004
1	12	.010	100.000	100.000	5.007	50.045	500.020
1	13	.000	100.000	99.956	.000	.006	500.045
1	14	.000	80.000	80.003	.000	.000	500.045
1	15	.000	60.000	60.001	.000	.000	500.045
1	16	.000	40.000	39.999	.000	.000	500.045
1	17	.000	20.000	19.997	.000	.000	500.045
1	18	.000	.000	.044	.000	-.006	500.045
1	19	.010	.000	.000	-5.007	-50.045	500.020
1	20	20.000	.000	.000	-5.000	-99.986	400.004
1	21	40.000	.000	.000	-5.000	-99.997	300.009
1	22	60.000	.000	.000	-5.000	-99.997	200.011
1	23	80.000	.000	.000	-5.000	-99.986	100.016
1	24	99.990	.000	.000	-5.007	-50.045	.000

Mass balance: .0000

```

*****
*           Boundary Integral Program           *
*   Solves system of equations of the form (A h = B q)   *
*****
The program presented here was prepared in support of research activities and is not intended for other use. Neither the United States Government, the University of Arizona nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any program of this program or represents that its use by such third party would not infringe privately owned rights.

PROGRAM him
$INCLUDE j:common

OPEN(8,file='flow.out',status='new')
OPEN(9,file='flow.res',status='new')

10 CALL cls(2)

WRITE(*,101)
READ(*,*,err=10) i

IF (i .EQ. 1) THEN
  CALL cls(2)
  CALL input
  CALL build
  CALL pack

ELSEIF (i .EQ. 2) THEN
  CALL cls(2)
  CALL plot

ELSEIF (i .EQ. 3) THEN
  CALL cls(2)
  CALL gauss
  CALL unpack
  CALL show
  CALL cls(2)
  CALL stream
  CALL pack
  CALL gauss
  CALL unpack
  CALL show

ELSEIF (i .EQ. 4) THEN
  CALL cls(2)
  CALL points
  CALL pointers

ELSEIF (i .EQ. 5) THEN
  CALL cls(2)
  CALL contract
  CALL cls(2)
  CALL build
  CALL pack

ELSEIF (i .EQ. 6) THEN
  CALL cls(2)
  CALL tracer

```

```

ELSEIF (i .EQ. 7) THEN
  OPEN(7,file='flow.dat',form='binary')
  WRITE(7) nn,nd,n,nj,nn,nc,t,total,x,y,u,v,q,r,kode,g,h,ikode
  CLOSE(7)

ELSEIF (i .EQ. 8) THEN
  OPEN(7,file='flow.dat',form='binary')
  READ (7) nn,nd,n,nj,nn,nc,t,total,x,y,u,v,q,r,kode,g,h,ikode
  CLOSE(7)

ELSEIF (i .EQ. 9) THEN
  CALL cls(2)
  CLOSE(8)
  CLOSE(9)
  STOP
ENDIF

GOTO 10

101 FORMAT (
. //15x,' 1 - Input from Data File',
. /15x,' 2 - Display Boundary Equation Matrix',
. /15x,' 3 - Find Potentials and Streamlines at Boundary Points',
. /15x,' 4 - Find Potentials and Streamlines at Interior Points',
. /15x,' 5 - Find Free Surfaces',
. /15x,' 6 - Calculate Breakthrough Curves',
. /15x,' 7 - Write Backup File',
. /15x,' 8 - Read Backup File',
. //15x,' 9 - Exit to DOS',
.///15x,'          Enter Selection: '$)
END

```

```

* .....
*           Pause output and clear screen
* .....

SUBROUTINE cls(n)

CHARACTER wait

IF (n .EQ. 1) THEN
  WRITE(*,101) CHAR(27)
  READ (*,102) wait
ENDIF

WRITE(*,103) CHAR(27)

IF (n .EQ. 2) WRITE(*,104)

RETURN
101 FORMAT(A1,'[25;25H','Press <RETURN> key to continue ...')
102 FORMAT(A1)
103 FORMAT(A1,'[2J')
104 FORMAT(/15x,'          BOUNDARY INTEGRAL METHOD',
.         /15x,' Implemented by Todd C. Rasmussen',
.         /15x,' Department of Hydrology, Univ. of AZ')
END

```

```

* .....
*           Read Data and Build Global Arrays
* .....

SUBROUTINE input

$INCLUDE j:common

CHARACTER*10 readfile, chr1, chr2
CHARACTER*80 chr

* Determine input file name
WRITE(*,101)
READ (*,102) readfile
OPEN (1,file=readfile)

* Echo input data?
WRITE(*,103)
READ (*,104) list

* Read number of domains
READ (1,105) nd
IF (nd .GT. mx2) THEN
  WRITE(*,108)
  CALL cls(1)
  STOP
ENDIF

* Input data for all domains
DO 20 i = 1, nd
  READ (1,105) nm(i),(nc(j,i),j=1,nm(i))
  IF (list.EQ.1) WRITE(*,105) nm(i),(nc(j,i),j=1,nm(i))

  n(i) = nc(nm(i),i)
  IF (n(i) .GT. mx3) THEN
    WRITE(*,108)
    CALL cls(1)
    STOP
  ENDIF

  READ (1,109) t(i)
  IF (list.EQ.1) WRITE(*,109) t(i)

  DO 10 j = 1, n(i)
    READ (1,107) i,j,x(j,i),y(j,i),kode(j,i),u(j,i)
10    IF (list.EQ.1) WRITE(*,107) i,j,x(j,i),y(j,i),kode(j,i),u(j,i)

    IF (list.EQ.1) CALL cls(1)

* Find total number of nodes.
IF (i .EQ. 1) nj(i) = 0
20 IF (i .NE. 1) nj(i) = n(i-1) + nj(i-1)

```

* Done with data file, close it.

```
CLOSE(1)

nn = nj(nd) + n(nd)
IF (nn .GT. mx1) THEN
  WRITE(*,108)
  CALL cls(1)
  STOP
ENDIF

RETURN

101 FORMAT (//10x,' Enter name of input data file: '$)
102 FORMAT (A10)
103 FORMAT (/10x,' Echo of input data?: '$)
104 FORMAT (I1)
105 FORMAT (16I5)
106 FORMAT (2F10.2,15,F10.2)
107 FORMAT (2I5,2F10.2,15,F10.2)
108 FORMAT (//10x,' Problem exceeds memory capacity')
109 FORMAT (8F10.2)
END.
```

```
*****
*           Compute Quadrature Weighting Functions           *
*****

SUBROUTINE build

$INCLUDE j:comson

* Clear G and H
DO 10 i = 1, nd
DO 10 j = 1, n(i)
DO 10 k = 1, n(i)
  h(j,k,i) = 0.
  g(j,k,i) = 0.

* Compute G and H
DO 30 i = 1, nd
IF (t(i) .NE. 0) THEN
DO 20 j = 1, n(i)
DO 20 k = 1, n(i)
  l = next(k,nc(1,i),nm(i))
  m = last(k,nc(1,i),nm(i))
  IF ((j .NE. k) .AND. (j .NE. l)) THEN
    CALL integral (t(i),x(j,i),y(j,i),x(k,i),y(k,i),x(l,i),y(l,i),
    a1,a2,b1,b2)
    h(j,k,i) = h(j,k,i) + a1
    g(j,k,i) = g(j,k,i) + b1
    h(j,l,i) = h(j,l,i) + a2
    g(j,l,i) = g(j,l,i) + b2
    h(j,j,i) = h(j,j,i) - a1 - a2
  ELSE
    ax = x(l,i) - x(k,i)
    ay = y(l,i) - y(k,i)
    sr = DSQRT(ax*ax + ay*ay)
    b1 = sr * (1.5 - DLOG(sr/DSQRT(t(i)))) / 2. / t(i)
    b2 = sr * (0.5 - DLOG(sr/DSQRT(t(i)))) / 2. / t(i)
    IF (k .NE. j) THEN
      g(j,k,i) = g(j,k,i) + b2
      g(j,l,i) = g(j,l,i) + b1
    ELSE
      g(j,k,i) = g(j,k,i) + b1
      g(j,l,i) = g(j,l,i) + b2
    ENDIF
  ENDIF
ENDIF
20 CONTINUE
ENDIF
30 CONTINUE
RETURN
END
```

```

*****
* Find influence function between two line segments *
*****

```

```

SUBROUTINE integral (t,x,y,x1,y1,x2,y2,a1,a2,b1,b2)

IMPLICIT REAL*8(A-H,O-Z)

DIMENSION f(6), w(6)

DATA f /0.125233408511469, 0.3678314989998180, 0.587317954286617,
.      0.769902674194305, 0.904117256370475, 0.981560634246719/

DATA w /0.249147045813403, 0.233492536538355, 0.203167426723066,
.      0.160078328543346, 0.108939325995318, 0.047175336386512/

ax = (x2 - x1) / 2.
ay = (y2 - y1) / 2.
bx = (x2 + x1) / 2.
by = (y2 + y1) / 2.

IF (ax .NE. 0) THEN
  ta = ay / ax
  dist = ABS ((ta*x - y + y1 - ta*x1) / DSQRT(ta*ta + 1))
ELSE
  dist = ABS (x - x1)
ENDIF

IF ( (x1-x)*(y2-y) .LT. (x2-x)*(y1-y) ) dist = -dist

a1 = 0.
a2 = 0.
b1 = 0.
b2 = 0.

DO 10 i = 1, 6
  gi = f(i)
  DO 10 j = 1, 2
    IF (j .EQ. 2) gi = -gi
    xx = x - (ax * gi + bx)
    yy = y - (ay * gi + by)
    ra = DSQRT (xx*xx + yy*yy)
    ar = DSQRT (ax*ax + ay*ay)
    g = ar * w(i) * DLOG(ra/DSQRT(t)) / 2. / t
    h = ar * w(i) * dist / (ra*ra) / 2.
    a1 = a1 + h * (gi - 1.)
    a2 = a2 - h * (gi + 1.)
    b1 = b1 + g * (gi - 1.)
    b2 = b2 - g * (gi + 1.)
  10
RETURN
END

```

```

*****
* Find subsequent node *
*****

```

```

INTEGER function next (j,nc,m)

DIMENSION nc(*)

next = j + 1
IF (j .EQ. nc(1)) THEN
  next = 1
ELSEIF (m .GT. 1) THEN
  DO 10 k = 2, m
10  IF (j .EQ. nc(k)) next = nc(k-1) + 1
  ENDDIF

RETURN
END

```

```

*****
* Find previous node *
*****

```

```

INTEGER function last (j,nc,m)

DIMENSION nc(*)

last = j - 1
IF (j .EQ. 1) THEN
  last = nc(1)
ELSEIF (m .GT. 1) THEN
  DO 10 k = 2, m
10  IF (j .EQ. nc(k-1)+1) last = nc(k)
  ENDDIF

RETURN
END

```

```

.....
*                               *
*           Create Global Matrix           *
*                               *
.....

```

SUBROUTINE pack

\$INCLUDE j:common

* Combine H with Q to form vector of knowns
* and transfer from Q to QQ and G to GG

```

DO 5 i = 1, mx1
  qq(i) = 0.
DO 5 j = 1, mx1
5  gg(i,j) = 0.

DO 50 i = 1, nd
  l = nj(i)
DO 50 k = 1, n(i)
  m = kode(k,i)
  mj = MOD(-m,128)
  mk = -m/128
  ml = nj(mk)+mj

  IF (m .GT. 0) THEN
    DO 10 j = 1, n(i)
      qq(l+j) = qq(l+j) - g(j,k,i) * u(k,i)
10  gg(l+j,l+k) = - h(j,k,i)

    ELSEIF (m .EQ. 0) THEN
      DO 20 j = 1, n(i)
        qq(l+j) = qq(l+j) + h(j,k,i) * u(k,i)
20  gg(l+j,l+k) = g(j,k,i)

    ELSEIF (i .LT. mk) THEN
      DO 30 j = 1, n(i)
        gg(l+j,ml) = -h(j,k,i)
30  gg(l+j,l+k) = g(j,k,i)

    ELSE
      DO 40 j = 1, n(i)
        gg(l+j,ml) = -g(j,k,i)
40  gg(l+j,l+k) = -h(j,k,i)
    ENDIF
50 CONTINUE

RETURN
END

```

```

.....
*                               *
*           Plot matrix                       *
*                               *
.....

```

SUBROUTINE plot

\$INCLUDE j:common

CHARACTER c(200),d,s,b

DATA s,b / ' ', 'q' /
DATA k / 1 /

```

CALL cls(0)
DO 20 i = 1, nn
  DO 10 j = 1, nn
    IF (gg(i,j) .GT. 0.) THEN
      c(j)='+'
    ELSEIF (gg(i,j) .LT. 0.) THEN
      c(j)='- '
    ELSE
      c(j)='.'
    ENDIF
10  CONTINUE

  IF (qq(i) .NE. 0) THEN
    d = 'x'
  ELSE
    d = 'o'
  ENDIF

  WRITE (*,100) (c(j),j=1,nn),s,b,s,d

  IF (i .EQ. (nj(k) + n(k))) THEN
    CALL cls(1)
    k = k + 1
  ENDIF
20 CONTINUE

RETURN
100 FORMAT(1x,200a1)
END

```

```

*****
* Gaussian elimination
*****

```

```

SUBROUTINE gauss

```

```

$INCLUDE j:common

```

```

DATA error / 1.D-6 /

```

```

DO 50 l = 1, nn-1

```

```

* Exchange rows if zero in diagonal
IF (ABS(gg(l,l)) .LT. error) THEN
DO 20 k = l+1, nn

```

```

IF (ABS(qg(k,l)) .GT. error) THEN
DO 10 j = l, nn
c = gg(l,j)
gg(l,j) = gg(k,j)
10 gg(k,j) = c

```

```

c = qq(l)
qq(l) = qq(k)
qq(k) = c
GOTO 30

```

```

ENDIF
20 CONTINUE

```

```

* Can't find non-zero to exchange with, singular matrix
WRITE (*,*) ' * * Singularity * * In row', l
STOP
ENDIF

```

```

* Divide row by diagonal coefficient

```

```

30 c = gg(l,l)
qq(l) = qq(l) / c
DO 40 k = l+1, nn
40 gg(l,k) = gg(l,k) / c

```

```

* Eliminate unknown Q(L) from row J

```

```

DO 50 j = l+1, nn
c = gg(j,l)
qq(j) = qq(j) - c * qq(l)
DO 50 k = l+1, nn
50 gg(j,k) = gg(j,k) - c * gg(l,k)

```

```

* Compute last unknown

```

```

IF (ABS(gg(nn,nn)) .LE. error) THEN
WRITE (*,*) ' * * Singularity * * In row', nn
STOP
ELSE
qq(nn) = qq(nn) / gg(nn,nn)
ENDIF

```

```

* Back substitute

```

```

DO 60 j = 1, nn-1
l = nn-j
DO 60 k = l+1, nn
60 qq(l) = qq(l) - gg(l,k) * qq(k)

```

```

RETURN
END

```

```

.....
*           Reduce Global Equation           *
.....

```

```
SUBROUTINE unpack
```

```
$INCLUDE j:common
```

```

DO 10 i = 1, nd
DO 10 j = 1, n(i)
  m = kode(j,i)
  nj = MOD(-m,128)
  mk = -m/128
  k = nj(mk)+mj
  l = nj(i) + j

```

```

IF (m .GT. 0) THEN
  q(j,i) = u(j,i)
  u(j,i) = qq(l)

```

```

ELSEIF (m .EQ. 0) THEN
  q(j,i) = qq(l)

```

```

ELSEIF (i .LT. mk) THEN
  q(j,i) = qq(l)
  u(j,i) = qq(k)

```

```

ELSE
  q(j,i) = -qq(k)
  u(j,i) = qq(l)
ENDIF

```

```

10 CONTINUE
RETURN
END

```

```

.....
*           Display Solution                 *
.....

```

```
SUBROUTINE show
```

```
$INCLUDE j:common
```

```

ki = 1
sum = 0.
WRITE(8,601)

```

```

DO 20 i = 1, nd
DO 10 j = 1, n(i)

```

```

* Fix page break
IF (MOD(j-1,20) .EQ. 0) THEN
  IF (j .NE. 1) CALL cls(1)
  IF (j .EQ. 1) CALL cls(0)
  WRITE (*,601)
ENDIF

```

```

* Adjust discharges by length
l = next(j,nc(1,i),nm(i))
k = last(j,nc(1,i),nm(i))
d1 = DSQRT(DABS(x(j,i)-x(l,i))**2+DABS(y(j,i)-y(l,i))**2)/2.
dk = DSQRT(DABS(x(j,i)-x(k,i))**2+DABS(y(j,i)-y(k,i))**2)/2.
d = d1*(3.*q(j,i)+q(l,i))/4.+dk*(3.*q(j,i)+q(k,i))/4.
IF (j .EQ. 1) THEN
  total(j,i) = dk * (q(j,i)+q(k,i))
ELSE
  total(j,i) = total(j-1,i) + dk * (q(j,i)+q(k,i))
ENDIF
sum = sum + d

```

```

* Display locations, potentials, discharges, and mass balance
WRITE (*,603) i,j,x(j,i),y(j,i),u(j,i),q(j,i),d,total(j,i)
WRITE (8,603) i,j,x(j,i),y(j,i),u(j,i),q(j,i),d,total(j,i)

```

```

IF (j .EQ. nc(ki,i)) THEN
  total(j,i) = 0.
  ki = ki + 1
ENDIF

```

```
10 CONTINUE
```

```

WRITE (*,604) sum
WRITE (8,604) sum

```

```
20 CALL cls(1)
```

```

RETURN
601 FORMAT (4x,'i',4x,'n',7x,'x',9x,'y',9x,'u',9x,'q',9x,'d',9x,'t'/)
603 FORMAT (2i5,7f10.3)
604 FORMAT ('/' Mass balance: ',f10.4)
END

```



```

* Again
  x1 = x0
  y1 = y0 + delta
  h2 = 0.
  s2 = 0.

  DO 40 k = 1, n(i)
    l = next(k,nc(1,i),nm(i))
    CALL integral(t(i),x1,y1,x(k,i),y(k,i),x(1,i),y(1,i),
      a1,a2,b1,b2)
    h2=h2+((b1*r(k,i)+b2*r(1,i))-(a1*v(k,i)+a2*v(1,i)))/(2.*pi)
40   s2=s2+((b1*q(k,i)+b2*q(1,i))-(a1*u(k,i)+a2*u(1,i)))/(2.*pi)

  dhdy = (h2-h1)/delta
  dsdy = (s2-s1)/delta

* Calculate Jump using Newton-Raphson
  dh = h1 - h1
  ds = s1 - s1

  dy = (dhdx*ds - dsdx*dh) / (dhdx*dady - dhdy*dadx)
  dx = (dh - dhdy*dy) / dhdx

  x0 = x0 + dx/del
  y0 = y0 + dy/del

  IF (x0 .GT. 200. .OR. x0 .LT. -100. .OR.
    y0 .GT. 200. .OR. y0 .LT. -100) THEN
    del = 2. * del
    GOTO 5
  ENDIF

* Stopping Criteria
  IF ((DABS(dh) .GT. error .OR.
    DABS(ds) .GT. error) .AND. iter .NE. 50) THEN
    iter = iter + 1
    GOTO 10
  ENDIF
50  WRITE (*,102) j,iter,hi,si,x0,y0
    WRITE (9,102) j,iter,hi,si,x0,y0
  CLOSE(1)
  CALL cls(1)
  RETURN
101 FORMAT (15,2f10.3)
102 FORMAT (10x,15,4f10.3)
201 FORMAT (//14x,'# Iter   - u -   - v -   - x -   - y -')
  END

```

```

* * * * *
*           Compute potential at interior points
* * * * *

SUBROUTINE pointers

$INCLUDE j:common

CHARACTER*10 readfile
pi = DACOS(-1.DO)

OPEN(1,file='flow.in2')

WRITE(*,201)
WRITE(9,201)

READ (1,101) num
DO 20 j = 1, num
  READ (1,101) i, cx, cy
  h1 = 0.
  s1 = 0.

  DO 10 k = 1, n(i)
    l = next(k,nc(1,i),nm(i))
    CALL integral(t(i),cx,cy,x(k,i),y(k,i),x(1,i),y(1,i),
      a1,a2,b1,b2)
    h1=h1+((b1*r(k,i)+b2*r(1,i))-(a1*v(k,i)+a2*v(1,i)))/(2.*pi)
10   s1=s1+((b1*q(k,i)+b2*q(1,i))-(a1*u(k,i)+a2*u(1,i)))/(2.*pi)

  WRITE (*,102) i,cx,cy,h1,s1
20  WRITE (9,102) i,cx,cy,h1,s1

CLOSE(1)
CALL cls(1)

RETURN
101 FORMAT (15,2f10.3)
102 FORMAT (10x,15,4f10.3)
201 FORMAT (//10x,' Domain   - x -   - y -   - u -   - v -')
  END

```

```

.....
* Find Free Surface
.....

```

SUBROUTINE contract

INCLUDE j:common

```

DATA error / 1.D-1 /
DATA eps / 1.D-9 /
pi = DACOS(-1.D0)

```

```

DO 3 i = 1, n(i)
3 WRITE(*,113) i, x(i,1), y(i,1), v(i,1), r(i,1), v(i,1)-y(i,1)
113 format(i10,6f10.3)
CALL cis(i)

```

```

WRITE(*,201)
READ (*,*) angl, head

```

angl = DSIN(angl * pi / 180.)

```

CALL cis(0)
WRITE(*,202)
WRITE(9,202)

```

```

DO 50 i = 1, nd
DO 50 j = 1, n(i)

```

```

IF (ikode(j,i) .EQ. 1 .AND. DABS(r(j,i)) .LE. eps) THEN
hi = v(j,i) - angl*y(j,i) + head
IF (hi .LT. 0) THEN
hi = 0.
si = v(j,i)
delta = 0.25
del = 1.
iter = 1
5 x0 = (4.*x(j,i) + 50.) / 5.
y0 = (4.*y(j,i) + 50.) / 5.

```

* Find head and stream in center of domain

```

10 xi = x0
yi = y0
h1 = - angl*yi + head
s1 = 0.
DO 20 k = 1, n(i)
l = next(k,nc(1,i),nm(i))
CALL integral(t(i),xi,yi,x(k,i),y(k,i),x(l,i),y(l,i),
. a1,a2,b1,b2)
. tmp = ((b1*r(k,i)+b2*r(l,i))-
. (a1*v(k,i)+a2*v(l,i)))/(2.*pi)
. h1 = h1 + tmp
20 s1 = s1 + tmp

```

* Find head and stream in two other directions

```

xi = x0 + delta
yi = y0
h2 = - angl*yi + head
s2 = 0.

```

```

DO 30 k = 1, n(i)
l = next(k,nc(1,i),nm(i))
CALL integral(t(i),xi,yi,x(k,i),y(k,i),x(l,i),y(l,i),
. a1,a2,b1,b2)
. tmp = ((b1*r(k,i)+b2*r(l,i))-
. (a1*v(k,i)+a2*v(l,i)))/(2.*pi)
. h2 = h2 + tmp
30 s2 = s2 + tmp

```

```

dhdx = (h2-h1)/delta
dedx = (s2-s1)/delta

```

* Again

```

xi = x0
yi = y0 + delta
h2 = - angl*yi + head
s2 = 0.

```

```

DO 40 k = 1, n(i)
l = next(k,nc(1,i),nm(i))
CALL integral(t(i),xi,yi,x(k,i),y(k,i),x(l,i),y(l,i),
. a1,a2,b1,b2)
. tmp = ((b1*r(k,i)+b2*r(l,i))-
. (a1*v(k,i)+a2*v(l,i)))/(2.*pi)
. h2 = h2 + tmp
40 s2 = s2 + tmp

```

```

dhdy = (h2-h1)/delta
dedy = (s2-s1)/delta

```

* Calculate Jump using Newton-Raphson

```

dh = h1 - h1
ds = s1 - s1

```

```

dy = (dhdx*ds - dedx*dh) / (dhdx*dedy - dhdy*dedx)
dx = (dh - dhdy*dy) / dhdx

```

```

x0 = x0 + dx/del
y0 = y0 + dy/del

```

```

IF (x0 .GT. 200. .OR. x0 .LT. -100. .OR.
. y0 .GT. 200. .OR. y0 .LT. -100) THEN
del = 2. * del
GOTO 5
ENDIF

```

```

* Stopping Criteria
  IF ((DABS(dh) .GT. error .OR.
      DABS(ds) .GT. error) .AND. iter .NE. 50) THEN
      iter = iter + 1
      GOTO 10
  ENDIF

99  WRITE (*,102) j,iter,dh,ds,x0,y0
     WRITE (9,102) j,iter,dh,ds,x0,y0

     x(j,i) = (x(j,i) + x0) / 2.
     y(j,i) = (y(j,i) + y0) / 2.
  ENDIF
ENDIF
50 CONTINUE

DO 60 i = 1, nd
DO 60 j = 1, n(i)
  kode(j,i) = ikode(j,i)
  u(j,i) = v(j,i)
60  IF (kode(j,i) .EQ. 1) u(j,i) = r(j,i)

CALL c1a(1)
RETURN
102 FORMAT (10X,2I5,6F10.3)
201 FORMAT (///10X,' Enter dip and head: ',§)
202 FORMAT (//14X,'# Iter   - h -   - p -   - x -   - y -')

END

```

```

*****
* Calculate Breakthrough Curves
*****

SUBROUTINE tracer

$INCLUDE j:common

CHARACTER*10 readfile
REAL*8 tt(11)

DATA error, delta, ddelta / 1.D-3, 1.D-1, 1. /
pi = DACOS(-1.D0)

* Find maximum/minimum flow/head
flowmax = 0.
flowmin = 1.D12

headmax = 0.
headmin = 1.D12

DO 60 i = 1, nd

WRITE (*,301)
WRITE (9,301)
301 FORMAT(' flow head iter      x0      y0      distance',
. ' velocity time')

DO 5 j = 1, n(i)
  IF (flowmax .LT. u(j,i)) flowmax = u(j,i)
  IF (flowmin .GT. u(j,i)) flowmin = u(j,i)
  IF (headmax .LT. v(j,i)) headmax = v(j,i)
  IF (headmin .GT. v(j,i)) headmin = v(j,i)
5

istep = 10

headinc = (headmax-headmin) / DBLE(istep)
flowinc = (flowmax-flowmin) / DBLE(istep)

hdelta = headinc/10.
fdelta = flowinc/10.

si = flowmin + fdelta

DO 40 j = 1, istep+1

  IF (j .EQ. 2) THEN
    si = flowmin + flowinc
  ELSEIF (j .EQ. istep+1) THEN
    si = flowmax - fdelta
  ENDIF

  hi = headmin + hdelta

```

```

time = 0.
DO 35 jk = 1, istep+1
  IF (jk .EQ. 2) THEN
    hi = headmin + heading
  ELSEIF (jk .EQ. istep+1) THEN
    hi = headmax - hdelta
  ENDIF

  iter = 1
  del = 1.

10  x0 = 50.
   y0 = 50.

15  xi = x0
   yi = y0

   h1 = 0.
   s1 = 0.

  DO 20 k = 1, n(i)
    l = next(k,nc(1,i),nm(i))
    CALL integral(t(i),xi,yi,x(k,i),y(k,i),x(1,i),y(1,i),
      a1,a2,b1,b2)
    h1=h1+((b1*r(k,i)+b2*r(1,i))-(a1*v(k,i)+a2*v(1,i)))/(2.*pi)
20  s1=s1+((b1*q(k,i)+b2*q(1,i))-(a1*u(k,i)+a2*u(1,i)))/(2.*pi)

   xi = x0 + delta

   h2 = 0.
   s2 = 0.

  DO 25 k = 1, n(i)
    l = next(k,nc(1,i),nm(i))
    CALL integral(t(i),xi,yi,x(k,i),y(k,i),x(1,i),y(1,i),
      a1,a2,b1,b2)
    h2=h2+((b1*r(k,i)+b2*r(1,i))-(a1*v(k,i)+a2*v(1,i)))/(2.*pi)
25  s2=s2+((b1*q(k,i)+b2*q(1,i))-(a1*u(k,i)+a2*u(1,i)))/(2.*pi)

  dhdx = (h2-h1)/delta
  dedx = (s2-s1)/delta

  xi = x0
  yi = y0 + delta

  h2 = 0.
  s2 = 0.

  DO 30 k = 1, n(i)
    l = next(k,nc(1,i),nm(i))
    CALL integral(t(i),xi,yi,x(k,i),y(k,i),x(1,i),y(1,i),
      a1,a2,b1,b2)
    h2=h2+((b1*r(k,i)+b2*r(1,i))-(a1*v(k,i)+a2*v(1,i)))/(2.*pi)
30  s2=s2+((b1*q(k,i)+b2*q(1,i))-(a1*u(k,i)+a2*u(1,i)))/(2.*pi)

  dhdy = (h2-h1)/delta
  dedy = (s2-s1)/delta

  dh = hi - h1
  ds = si - s1

  dy = (dhdx*ds - dedx*dh) / (dhdx*dedy - dhdy*dedx)
  dx = (dh - dhdy*dy) / dhdx

  x0 = x0 + dx/del
  y0 = y0 + dy/del

  IF (x0 .GT. 200. .OR. x0 .LT. -100. .OR.
    y0 .GT. 200. .OR. y0 .LT. -100) THEN
    del = 2. * del
    GOTO 10
  ENDIF

  IF ((DABS(dh) .GT. error .OR.
    DABS(ds) .GT. error) .AND. iter .NE. 50) THEN
    iter = iter + 1
    GOTO 15
  ENDIF

  v1 = DSQRT(dhdy**2 + dhdx**2)
  dist = DSQRT((x0-x0)**2 + (y0-y0)**2)

  IF (jk .EQ. 2) THEN
    time = time + dist / v1 * heading / (heading - hdelta)
  ELSEIF (jk .EQ. istep+1) THEN
    time = time + dist / vo * heading / (heading - hdelta)
  ELSEIF (jk .NE. 1) THEN
    time = time + dist * (v1+vo) / (2.*vo*v1)
  ENDIF

  vo = v1

  x0 = x0
  y0 = y0

  WRITE (*,102) j,jk,iter,x0,y0,dist,v1,time
  WRITE (9,102) j,jk,iter,x0,y0,dist,v1,time

35  hi = hi + heading

   tt(j) = time
40  si = si + flowinc

  CALL c1s(1)

45 WRITE (*,100)
  READ (*,*) retard

  IF (retard .LT. 0.) GOTO 60

```

```

WRITE (*,109)
READ (*,*) atten

WRITE (*,110)
READ (*,*) to, delta

WRITE (9,112) retard, atten, to, delta

time = to
DO 55 jt = 1, 100
  time = time + delta
  sumatt = 0.

  DO 50 j = 1, istep + 1
    IF (retard*tt(j) .GE. time) THEN
      att = 0.
    ELSE
      att = erfc(atten * retard * tt(j))
        / DSQRT(time - retard * tt(j))
      sumatt = sumatt + att
    ENDIF
50 CONTINUE

WRITE (*,111) jt, time, sumatt/DBLE(istep+1)
55 WRITE (9,111) jt, time, sumatt/DBLE(istep+1)

CALL cls(1)
GOTO 45
60 CONTINUE

RETURN
100 FORMAT (/// ' Enter retardation coefficient: ', $)
109 FORMAT (/// ' Enter matrix diff. coefficient (negative to end): ', $)
110 FORMAT (/// ' Enter initial time and time step: ', $)
111 FORMAT (110, 2f12.6)
112 FORMAT (5x, ' Retard, Atten Coef, To, Delta = ', 4G12.3)
101 FORMAT (15, 2f10.3)
102 FORMAT (3i5, 6f10.3)
201 FORMAT (//14x, '# Iter   - u -   - v -   - x -   - y -')
END

```

REAL*8 FUNCTION erfc(x)

IMPLICIT REAL*8 (A-H,O-Z)

DATA p,a,b,c,d,e / 0.327591100, 0.254829592, -0.284496736,
. 1.421413741, -1.453152027, 1.061405429 /

t = 1/(1+p*x)

erfc = (a*t + t*(b*t + t*(c*t + t*(d*t + t*e*t)))) / DEXP(x*x)

RETURN

END

* COMMON BLOCK DECLARATIONS *

IMPLICIT REAL*8(A-H,O-Z)

* PARAMETERS:

- * mx1 = maximum size of solver matrix
- * mx2 = maximum number of domains
- * mx3 = maximum number of nodes per domain

PARAMETER (mx1=100,mx2=5,mx3=60)

COMMON nn, nd, n(mx2), nj(mx2), nm(mx2), nc(mx2,mx2), t(mx2),
. total(mx3,mx2), x(mx3,mx2), y(mx3,mx2), u(mx3,mx2), v(mx3,mx2),
. q(mx3,mx2), r(mx3,mx2), kode(mx3,mx2), g(mx3,mx3,mx2),
. h(mx3,mx3,mx2), gg(mx1,mx1), qq(mx1), ikode(mx3,mx2)

A.2 FRACGEN: Discrete Fracture Network Generator

This section presents a computer simulation model, FRACGEN, used to generate networks of discrete fractures. The simulation model was written for the purpose of providing analysis and interpretations of the hydraulic properties of fractured rock masses. FRACGEN represents fractures as two dimensional finite planes with a fixed thickness. The simulated fracture network is a collection of a number of individual finite planes within a three dimensional global volume. The global volume is composed of six exterior surfaces which are defined by the user. Internal surfaces can also be specified to represent boreholes or mine shafts, adits and drifts.

The simulation model is used to generate the planar fractures within the three-dimensional global volume using physically-based parameters. A smaller sample volume is then extracted from within the global volume to remove effects of undersampling near boundaries. Isolated and dead-end fractures are also removed. The computer model is able to generate synthetic fractures using descriptive statistical data inputs on the size of the generating volume, number of fracture sets, fracture density, areal extent, orientation, and transmissivity. The model also allows the user to input observed fractures from a field site as an option. The model then solves for intersections between fractures, and between fractures and surfaces.

FRACGEN is based on code originally developed by Huang and Evans (1985). The original code generated three-dimensional networks of discrete fractures for only a single fracture set. The fracture network was then reduced to a set of nodes with no account taken for the physical geometry of individual fractures. The newly revised code allows for the incorporation of multiple fracture sets, as well as for discretization of fractures and surfaces. The discretization of boundaries is required when the boundary integral method is used to solve two and three dimensional flow problems.

A.2.1 Creation of Individual Fractures

Fractures are defined as finite planar features within a three-dimensional volume. The fractures can either be defined using input data obtained from field observations, or synthetic fractures can be generated using statistical methods. Estimation techniques for obtaining the relevant physical parameters are presented in Appendix A. If the computer simulation model is used to generate individual fractures, a pseudo-random number generator is used in conjunction with specified frequency distributions (i.e., normal, log-normal, uniform and exponential) for fracture parameters. Pseudo-random numbers are generated using a linear congruential generator (LCG) of the form:

$$(A.1) \quad s(i+1) = (a s(i) + c) \text{ MOD } m$$

where

$s(i)$ pseudo-random variable for random number i ;
 a, c multiplier and increment;
 MOD modulus operator; and
 m modulus.

For a full-period LCG (i.e., the number of generated random numbers is equal to the modulus), the value of the multiplier and the increment are limited to:

$$(A.2a) \quad a = 4n - 1 \quad n = 1, 2, \dots m/4$$

and

$$(A.2b) \quad c = 2p - 1 \quad p = 1, 2, \dots m/2$$

The value of the modulus is normally limited by the number of bits in the largest unsigned integer declaration. For a language which accepts unsigned integers and allows register overflow, the largest value of the modulus is:

$$(A.3a) \quad m = 2^j$$

where j is the number of bits of the largest unsigned integer declaration. For a computer language which does not allow implicit overflow, the modulus operator is required and the value of j must be halved. In addition, if unsigned integers are not allowed, the value of j must be decremented by one. For standard FORTRAN the largest value of the modulus is computed by:

$$(A.3b) \quad m = 2^{(j/2 - 1)}$$

For languages with integer declarations limited to four bytes, (i.e., j equal to 32 bits) the largest modulus is 32768. By specifying different seed numbers, multipliers or increments, a large number of unique fracture networks can be generated.

A.2.2 Definition of Global and Sample Volumes, and Interior Surfaces

The global volume is the region within fractures are defined. The global coordinate system is a three dimensional Cartesian coordinate system used to locate points within the global volume. The coordinate system is defined by specifying a center location, $R_0(x,y,z)$, and the dimensions of the global volume, (L_x, L_y, L_z) . Once the global coordinate system has been defined and the initial fracture network has been generated, a sample volume within the global volume is specified. The sample volume is defined as a sub-region within the global volume. The ability to specify a sample volume is necessary in order to remove the effects of undersampling near boundaries, to provide the capability for investigating scale effects, and to evaluate the spatial and directional variability of network properties.

The sample coordinate system is also a three dimensional Cartesian coordinate system used to describe the intersections (or nodal points) contained inside a rectangular block of sample. The coordinate system is defined by an origin at the center of the bottom of the rectangular block, $r_0(x,y,z)$. The sample coordinate system need not be oriented in the same direction as the global coordinate system, allowing the axes to be rotated (a,b) degrees with respect to the original coordinate system. The dimensions of the sample volume, (l_x, l_y, l_z) , should be smaller than

the global volume so that undersampling near boundaries can be avoided. The undersampling occurs because only fractures whose centers lie within the global volume are generated. Fractures whose centers are located outside of the global volume but which would have extended into the global volume are not sampled. Near the global volume boundary the density of fractures will fall to one-half of the density near the center.

The transformation from the global coordinate system to the sample coordinate system is given by:

$$(A.4) \quad \begin{matrix} x \\ y \\ z \end{matrix} = \begin{pmatrix} \cos\alpha \cos\beta & \sin\alpha \cos\beta & \sin\beta \\ -\sin\alpha & \cos\alpha & 0 \\ -\cos\alpha \sin\beta & -\sin\alpha \sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}$$

where

- x, y, z axes of the sample cartesian coordinate system, m;
- X, Y, Z axes of the global cartesian coordinate system, m; and
- X_0, Y_0, Z_0 origin of the axes of the global coordinate system, m.

Interior surfaces, such as boreholes, mines, etc. are defined by similarly specifying a center position, three volume dimensions, and two rotation dimensions. The interior surfaces are allowed to intersect each other, as well as the external sample volume surfaces. Currently, up to three interior volumes, composed of eighteen interior surfaces, are possible. Figure A.1 illustrates the geometric properties of the global and sample volumes, and interior surfaces.

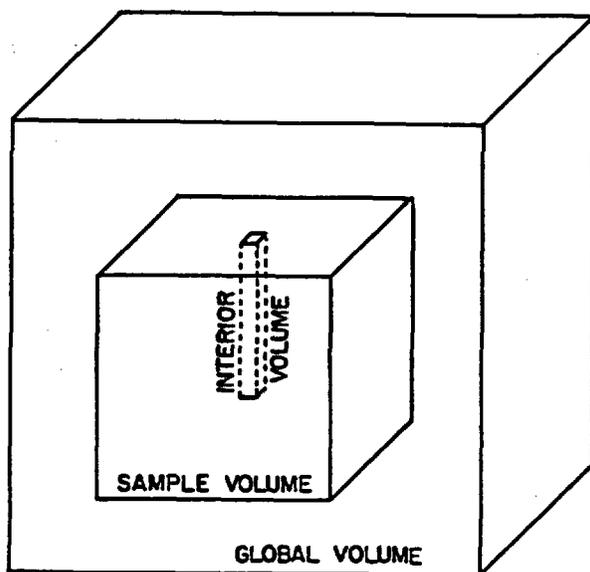


Figure A.1: Relative positions of global, sample and interior volumes.

A.2.3 Number of Fracture Sets and Fracture Density

Fractures are usually observed to form sets, with unique orientations, densities, and transmissivities for each set. To allow for the stratification of observed fractures into distinct sets, the simulation program allows the generation of families of fractures with user specified parameters for each set. If fractures are not synthetically generated, known fractures are input without regard to set membership. Currently, up to six fracture sets are possible. For each fracture set, the number of fractures within the global volume must be specified. This number can be computed as the global volume divided by the fracture density. Only integer values are permitted. For all fracture sets the total number of synthetic or observed fractures is currently limited to one hundred.

A.2.4 Fracture Location

Fracture centers for individual fractures are found by designating a point (R_0) having a global coordinate of (X_0, Y_0, Z_0) , as denoted on Figure A.2. The location of fracture centers within the global volume are found by assuming that they occur according to a Poisson process. This process results in a uniform probability of fracture centers for any location within the global volume and an exponential distribution of distances between fracture centers.

A.2.5 Fracture Orientation

The orientation of the fracture plane is defined by two angles of rotation (A and B) which are used to specify a vector normal to the fracture plane with the tail of the vector located at R_0 . The first angle is the horizontal angle measured counter-clockwise from the +X axis. The second angle is the elevation angle in the plane of R_0 T and measured from the XY plane (Figure A.2). The equation of an infinite plane encompassing the finite fracture is:

$$(A.5) \quad a(x - x_0) + b(y - y_0) + c(z - z_0) = 0$$

where

$$\begin{aligned} a &= \cos(\alpha)\cos(\beta) \\ b &= \cos(\alpha)\sin(\beta) \\ c &= \sin(\alpha) \end{aligned}$$

The two rotation angles can be related to the strike and dip of a fracture set using an appropriate transformation. Deviations from the mean value of the orientation parameters are used to provide perturbations distributed about the central tendency.

A.2.6 Fracture Areal Extent

The areal extent of each fracture requires that information about the shape and length be provided. Currently, a circle, square, rectangle or ellipse may be selected. The areal extent of the fracture is defined using characteristic lengths appropriate for the shape. Both an ellipse and a rectangle require two characteristic lengths, i.e., the major and minor axis lengths. In addition, the direction of the major axis, C, is also needed to specify the orientation of each shape.

The bounding regions of regular fractures are defined by converting the global coordinates into a local coordinate system. The fracture coordinate system is a two dimensional Cartesian coordinate system defined at the center of each fracture or boundary surface, and is used to find the intersections among fractures and boundary surfaces. For every surface, there is one such coordinate system defined. Within the local coordinate system Equation (A.6a) defines the region of an ellipse and Equation (A.6b) defines the region of a rectangle:

$$(A.6a) \quad x^2 / r_1^2 + y^2 / r_2^2 = 1$$

where

$$(A.6b) \quad x \leq r_1 \quad \text{and} \quad y \leq r_2$$

Figure A.2 illustrates the geometric properties of individual fractures. While a circle and square are special cases of an ellipse and rectangle, respectively, FRACGEN can be extended to generate other shapes as long as the boundaries can be expressed by analytic functions.

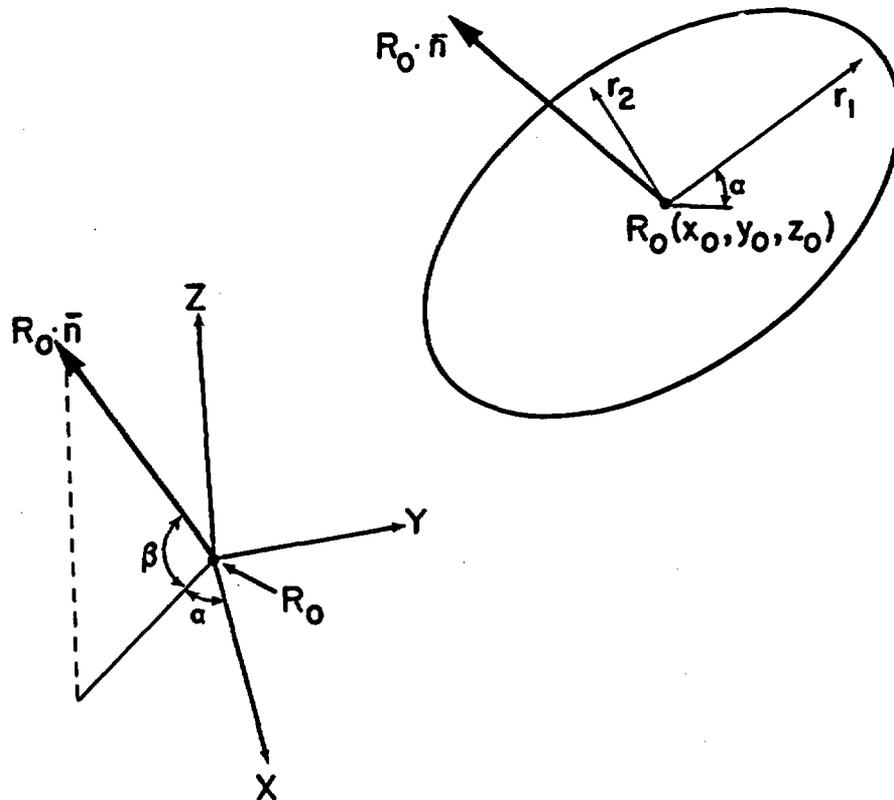


Figure A.2: Position and orientation parameters for discrete fractures.

A.2.7 Fracture Transmissivity

The intrinsic fracture transmissivity can be defined for each fracture, or the transmissivity can be generated using statistical distributions. The model treats the intrinsic fracture transmissivity as a constant, but allows the relative transmissivity to vary as a function of fluid potential. The relationship between the relative transmissivity and fluid potential is dependent upon the intra-fracture transmissivity distribution, which may or may not be the same as the inter-fracture transmissivity distribution. To account for this difference, a functional relationship between the relative transmissivity and fluid potential is specified using input parameters.

A.2.8 Determination of Fracture Intersections

Once the individual fractures have been generated, the fractures are further manipulated so that intersections between fractures and between fractures and surfaces can be found. Also, isolated and dead end fractures are eliminated. Intersections between fractures are called internal intersects, while intersections between fracture planes and boundary surfaces are called external intersects. The lines of intersection are obtained by first finding the intersecting line between two infinite planes containing the two finite fractures, or the finite fracture and a finite surface. The infinite line of intersection is next truncated to a finite segment such that it is common to both finite surfaces. The equation of an infinite plane containing the first finite fracture is:

$$(A.9a) \quad a_1 x + b_1 y + c_1 z = d_1$$

Similarly, the equation for a second plane containing the second fracture is:

$$(A.9b) \quad a_2 x + b_2 y + c_2 z = d_2$$

The equation of a line common to both planes is given as:

$$(A.10) \quad \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} X_1 \\ Y_1 \\ Z_1 \end{Bmatrix} u + \begin{Bmatrix} X_2 \\ Y_2 \\ Z_2 \end{Bmatrix}$$

where u is a scalar. As long as:

$$(A.11) \quad E = c_2 b_1 - c_1 b_2 \neq 0$$

then:

$$(A.12a) \quad \begin{Bmatrix} X_1 \\ Y_1 \\ Z_1 \end{Bmatrix} = \begin{Bmatrix} 1 \\ (a_2 c_1 - a_1 c_2) / E \\ (b_2 a_1 - b_1 a_2) / E \end{Bmatrix}$$

and

$$(A.12b) \quad \begin{Bmatrix} X_2 \\ Y_2 \\ Z_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ (c_2 d_1 - c_1 d_2) / E \\ (d_2 b_1 - d_1 b_2) / E \end{Bmatrix}$$

If the value of E in Equation (A.11) is equal to zero, then Equations (A.12) may be estimated using the other two cofactors as the basis function. Once the two vectors have been found, the line of intersection of the two planes can be represented by two distinct points on the line. This is accomplished by choosing two different values of t . Assume that points T_1 and T_2 are two distinct points on the line of intersection. The procedure for truncating this line to a finite line segment common to both fractures is:

- (1) Transform T_1 and T_2 to local coordinates defined on Fracture 1 and find the two boundary point intersects, P_{11} and P_{12} , between the line and the boundary of Fracture 1, if they exist.
- (2) Similar to step 1, find the two boundary points, P_{21} and P_{22} , representing the intersection between the line of intersection and Fracture 2. If the line does not intersect either one of the two fracture boundaries, then the two fractures do not share a common line (Figures A.3A and B).
- (3) If the line intersects both fractures, then Points P_{11} and P_{12} are checked to see if they are contained within the boundary of Fracture 2. Similarly, Points P_{21} and P_{22} are checked on Fracture 1. If the two fractures share a common line segment, then two of the four points should be common to both fracture regions (Figures A.3C and D). These two points are two end points of the finite line segment.

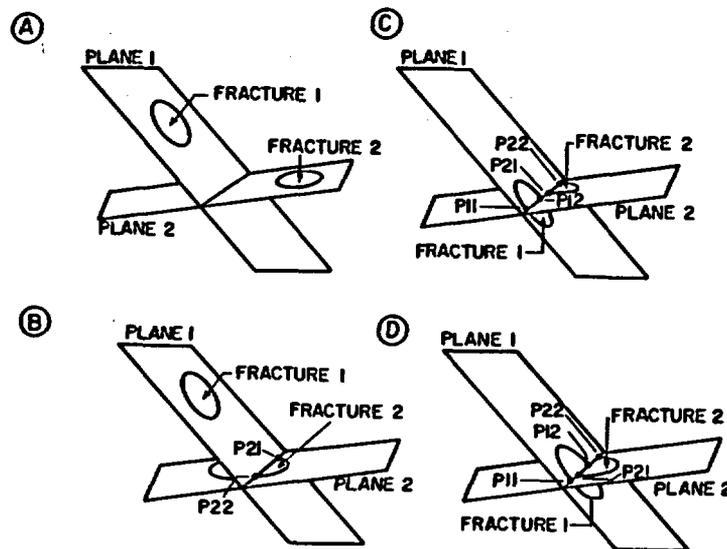


Figure A.3: Four possible outcomes of non-parallel fractures: Fracture planes intersect but neither discrete fracture intersects the line of intersection (A); Only one fracture intersects the line of intersection (B); Both fractures intersect the line of intersection, but not each other (C); and Both fractures intersect line of intersection and each other.

A.2.9 Truncation of Fractures

Within an isolated sample, the finite line segments between fractures are truncated to lie within the boundary of the sample. To accomplish the truncation, internal and external intersects are expressed in sample coordinates and the segments of the intersects which lie outside of the sample volume are truncated. If the intersect lies entirely outside of the sample volume, then that intersect is eliminated.

A.2.10 Removal of Isolated and Dead-End Fractures

Once the fracture network is assembled, isolated and dead-end fractures are removed to eliminate non-contributing flow routes. Isolated fractures are fractures which do not intersect any boundary surfaces or other fractures. Dead-end fractures are fractures which only intersect one boundary surface or one other fracture. While the removal of dead-end fractures will cause a decrease in the computed macroscopic dispersion coefficient, the effect will be in a conservative direction. That is, the exclusion of dead-end fractures will cause a decrease in the travel time from an injection point to an observation point. In addition, chemical interactions with the host rock will result in a higher total mass flux than when the dead-end fractures are included.

A.2.11 Examples of Program Inputs and Outputs

Table A.4 presents a description of input data for FRACGEN when data about individual fractures are available. Table A.5 describes the input data requirements when statistical data about fracture sets is available. Table A.6 provides sample input data for the option requiring statistical data. Also included is information about the sample volume and the boundary conditions imposed upon the boundaries of the sample volume. Table A.7 displays sample program output for the input data of Table A.6. Information about generated fractures as well as the location of the endpoints of fracture-fracture and fracture-boundary intersections are displayed.

Table A.4: Fracture Network Parameters: Specified for interior volumes
m greater than 1, fractures one through j and boundary sur-
faces 1 through k.

Input Variable:	Attribute	Variable Name
Sample Volume	Center	CORO(1,1) i = 1,2,3
	Dimension	COSZ(1,1) i = 1,2,3
	Orientation	COAL(1,1) i = 1,2
Interior Volume	Center	CORO(1,m) i = 1,2,3
	Dimension	COSZ(1,m) i = 1,2,3
	Orientation	COAL(1,m) i = 1,2
Fracture	Center	XINP(1,j) i = 1,2,3
	Orientation	XINP(1,j) i = 4,5,6
	Areal Extent	XINP(1,j) i = 7,8
	Transmissivity	XINP(9,j)
	Shape	XINP(10,j)
Boundary Conditions	Type	NBTYPE(k)
	Value	BVALUE(k)

Table A.5: Fracture Network Parameters: Specified for interior volumes
m greater than 1, generated for fractures one through j and
specified for boundary surfaces 1 through k.

Input Variable:	Attribute	Variable Name
Global Volume	Dimension	FRX,FRY,FRZ
Sample Volume	Center	CORO(1,1) i = 1,2,3
	Dimension	COSZ(1,1) i = 1,2,3
	Orientation	COAL(1,1) i = 1,2
Interior Volume	Center	CORO(1,m) i = 1,2,3
	Dimension	COSZ(1,m) i = 1,2,3
	Orientation	COAL(1,m) i = 1,2
Fracture	Number of sets	NSET
	Number in set	I
	Orientation	FRstrk,FRdip,SDstrk,SDdip
	Areal extent	FRlth,SDlth
	Transmissivity	FRtran,SDtran
Boundary Conditions	Shape	FRshp
	Type	NBTYPE(k)
	Value	BVALUE(k)

Table A.6: Sample program input.

Fracture Set: 1

Number of Fractures:	20		
Size of Prism (x,y,z):	100.000	100.000	100.000
(X-Y) Rotation:	.000	Deviation:	.000
(X-Z) Rotation:	.000	Deviation:	.000
Fracture Length:	10.000	Deviation:	.000
Transmissivity:	10.000	Deviation:	.000

Fracture Set: 2

Number of Fractures:	20		
Size of Prism (x,y,z):	100.000	100.000	100.000
(X-Y) Rotation:	90.000	Deviation:	.000
(X-Z) Rotation:	.000	Deviation:	.000
Fracture Length:	10.000	Deviation:	.000
Transmissivity:	10.000	Deviation:	.000

Boundary Surface Number: 1

Boundary size (meters):	100.00	100.00	100.00
Boundary center (meters):	50.00	50.00	50.00
Boundary orientation (degrees):	.00	.00	

#	Side	Type	Value
1	1	1	1.000
1	2	1	1.000
1	3	1	1.000
1	4	1	1.000
1	5	1	1.000
1	6	1	1.000

Table A.7: Sample program output for data set presented in Table A.6.

#	FRACTURE ATTRIBUTES					
	- Location (x,y,z) -			- Orientation -		
1	73.575	52.112	12.161	.000	.000	.000
2	46.707	47.388	31.094	.000	.000	.000
3	69.925	3.174	92.789	.000	.000	.000
4	86.038	85.107	70.621	.000	.000	.000
5	80.740	33.472	19.424	.000	.000	.000
6	16.153	89.221	76.614	.000	.000	.000
7	82.535	54.187	67.410	.000	.000	.000
8	97.342	75.464	84.464	.000	.000	.000
9	65.689	82.166	8.768	.000	.000	.000
10	90.262	4.614	69.962	.000	.000	.000
11	24.503	16.223	97.855	.000	.000	.000
12	41.068	53.711	8.853	.000	.000	.000
13	31.363	60.046	88.113	.000	.000	.000
14	55.545	34.448	39.542	.000	.000	.000
15	27.017	25.525	41.971	.000	.000	.000
16	36.282	65.137	14.224	.000	.000	.000
17	14.859	30.066	.516	.000	.000	.000
18	28.421	98.193	80.191	.000	.000	.000
19	13.748	93.433	94.620	.000	.000	.000
20	75.174	48.804	5.264	.000	.000	.000
21	25.735	67.871	50.552	90.000	.000	.000
22	43.887	.549	69.974	90.000	.000	.000
23	29.019	71.765	29.520	90.000	.000	.000
24	49.698	15.247	77.737	90.000	.000	.000
25	64.700	60.547	49.673	90.000	.000	.000
26	2.371	94.067	72.696	90.000	.000	.000
27	39.737	90.051	78.714	90.000	.000	.000
28	87.479	62.280	62.881	90.000	.000	.000
29	18.985	37.561	22.415	90.000	.000	.000
30	11.929	69.714	77.274	90.000	.000	.000
31	41.580	16.772	65.421	90.000	.000	.000
32	40.274	56.116	4.642	90.000	.000	.000
33	5.667	99.463	36.343	90.000	.000	.000
34	70.682	96.265	42.960	90.000	.000	.000
35	36.563	99.670	61.868	90.000	.000	.000
36	35.757	.574	72.977	90.000	.000	.000
37	97.757	26.514	63.589	90.000	.000	.000
38	29.959	87.378	49.893	90.000	.000	.000
39	63.333	92.383	65.494	90.000	.000	.000
40	75.504	89.343	91.629	90.000	.000	.000

 Table A.7: (Continued)

#	FRACTURE ATTRIBUTES (Continued)			
	- Length -		Trans.	Shape (1=elliptic)
1	2.550	2.550	10.000	.000
2	18.031	18.031	10.000	.000
3	9.466	9.466	10.000	.000
4	5.164	5.164	10.000	.000
5	3.806	3.806	10.000	.000
6	9.573	9.573	10.000	.000
7	7.229	7.229	10.000	.000
8	7.705	7.705	10.000	.000
9	2.831	2.831	10.000	.000
10	1.788	1.788	10.000	.000
11	8.128	8.128	10.000	.000
12	1.278	1.278	10.000	.000
13	43.665	43.665	10.000	.000
14	2.596	2.596	10.000	.000
15	4.220	4.220	10.000	.000
16	8.957	8.957	10.000	.000
17	1.660	1.660	10.000	.000
18	6.161	6.161	10.000	.000
19	1.886	1.886	10.000	.000
20	5.738	5.738	10.000	.000
21	3.711	3.711	10.000	.000
22	1.367	1.367	10.000	.000
23	8.668	8.668	10.000	.000
24	10.336	10.336	10.000	.000
25	7.756	7.756	10.000	.000
26	1.198	1.198	10.000	.000
27	9.966	9.966	10.000	.000
28	2.533	2.533	10.000	.000
29	21.714	21.714	10.000	.000
30	9.602	9.602	10.000	.000
31	3.849	3.849	10.000	.000
32	2.434	2.434	10.000	.000
33	2.908	2.908	10.000	.000
34	1.805	1.805	10.000	.000
35	4.390	4.390	10.000	.000
36	18.316	18.316	10.000	.000
37	2.735	2.735	10.000	.000
38	6.581	6.581	10.000	.000
39	10.203	10.203	10.000	.000
40	.294	.294	10.000	.000

Table A.7: (Continued)

	BOUNDARY SURFACES							
	Location			Orientation			Size	
100.000	50.000	50.000	.000	.000	.000	50.000	50.000	.000
.000	50.000	50.000	180.000	.000	.000	50.000	50.000	.000
50.000	100.000	50.000	90.000	.000	.000	50.000	50.000	.000
50.000	.000	50.000	270.000	.000	.000	50.000	50.000	.000
50.000	50.000	100.000	.000	90.000	.000	50.000	50.000	.000
50.000	50.000	.000	.000	270.000	.000	50.000	50.000	.000

Endpoints of Fracture-Fracture Intersections

N	Location		Location		Location	
1	31.363	90.051	68.748	31.363	90.051	88.680
2	31.363	87.378	44.448	31.363	87.378	56.474
3	36.282	71.765	20.852	36.282	71.765	23.181

Fracture-Fracture Intersections by Fracture Number

F	#	F	F	F	F	F	...
13	2	27	38				
16	1	23					
23	1	16					
27	1	13					
38	1	13					

Nodal Number by Fracture Number

F	#	N	N	N	N	N	...
13	2	1	2				
16	1	3					
23	1	3					
27	1	1					
38	1	2					

Table A.7: (Continued)

EXTERNAL NODES

B	Location		Location		Location	
1	100.000	26.514	60.855	100.000	26.514	66.324
2	.000	37.561	.701	.000	37.561	44.130
3	31.363	100.000	44.448	31.363	100.000	100.000
4	28.421	100.000	74.030	28.421	100.000	86.352
5	69.925	.000	83.323	69.925	.000	100.000
6	69.925	12.640	100.000	69.925	.000	100.000
7	24.503	24.351	100.000	24.503	8.095	100.000
8	31.363	100.000	100.000	31.363	16.381	100.000
9	14.859	31.726	.000	14.859	28.406	.000
10	75.174	54.542	.000	75.174	43.065	.000

Fracture Number by Boundary Surface Number

S	#	F	F	F	F	F	...
1	1	37					
2	1	29					
3	2	13	18				
4	1	3					
5	3	3	11	13			
6	2	17	20				

Node Number by Boundary Surface Number

S	#	B	B	B	B	B	...
1	1	1					
2	1	2					
3	2	3	4				
4	1	5					
5	3	6	7	8			
6	2	9	10				

Table A.7: (Continued)

Boundary Surface Number by Fracture Number

F	#	S	S	S	S	S	...
3	2	4	5				
11	1	5					
13	2	3	5				
17	1	6					
18	1	3					
20	1	6					
29	1	2					
37	1	1					

Boundary Node Number by Fracture Number

F	#	B	B	B	B	B	...
3	2	5	6				
11	1	7					
13	2	3	8				
17	1	9					
18	1	4					
20	1	10					
29	1	2					
37	1	1					

```

*****
*          FRACGEN: FRACTURE GENERATING Model          *
*          (Last Modification - March 18, 1987)        *
*****

PROGRAM FRACGEN
$INCLUDE j:common.a
CHARACTER*10 readfile

* Write initial screen.
CALL cls(0)
PRINT 202, CHAR(27)
CALL cls(1)

10 CALL cls(2)
WRITE(*,200)
READ (*,100) itype

IF (itype .EQ. 1) THEN
  CALL cls(2)
  WRITE(*,201)
  READ (*,101) readfile
  OPEN (1,FILE=readfile,status='old',err=10)
  READ (1,100) nfrct
  READ (1,102) ((xinp(j,i),j=1,10),i=1,nfrct)
  CLOSE(1)
  CALL showfractures

ELSEIF (itype .EQ. 2) THEN
  CALL generate
  CALL showfractures

ELSEIF (itype .EQ. 3) THEN
  CALL size

ELSEIF (itype .EQ. 4) THEN
  CALL buildfractures
  CALL findfractures
  CALL listfractures
  OPEN(9,file='savefile',form='binary',status='new')
  WRITE(9) cosZ,cor0,coal,corA,nfrct,nnode,nfr,ncf,xinp,nfn,ndcf,
  nfrn,ndfr,xx1,xx2,xi1,xi2,index,jtotal,ncut,xcut,nbond,nct,nctn,
  ndct,nd
  CLOSE(9)

ELSEIF (itype .EQ. 9) THEN
  CALL cls(2)
  STOP
ENDIF

GOTO 10
100 FORMAT (8i10)
101 FORMAT (a10)
102 FORMAT (10f10.3)

```

```

200 FORMAT (/15x,'          MAIN MENU'
  .        /15x,' 1 - Input Known Fractures',
  .        /15x,' 2 - Generate Random Fractures',
  .        /15x,' 3 - Define Boundary Sizes',
  .        /15x,' 4 - Generate Fractures and Combine Network',
  .        //15x,' 9 - Exit to DOS',
  .        //15x,'          Enter Selection: '$)
201 FORMAT (/10x,' Enter name of data file: '$)
202 FORMAT(A1,'[2J',////
  . //10x,'          FRACGEN'
  . //10x,' Program to Generate Discrete Fracture Networks'
  . //10x,' Originally developed by Chi-Hua Huang (6/84)'
  . //10x,' Substantially modified by Todd C. Rasmussen (6/86)'
  . /10x,' Department of Hydrology & Water Resources'
  . /10x,' University of Arizona, Tucson 85721')
END

```

The program presented here was prepared in support of research activities and is not intended for other uses. Neither the United States Government, the University of Arizona nor any of their employees, make any warranty, expressed or implied, or assume any legal liability or responsibility for any third party's use, or the results of such use, of any portion of this program or represents that its use by such third party would not infringe previously owned rights.

```

*****
*          Pause output and clear screen          *
*****
SUBROUTINE cls(n)
IF (n .EQ. 1) THEN
  PRINT 101, CHAR(27)
  READ (*,102)
ENDIF
PRINT 103, CHAR(27)
IF (n .EQ. 2) PRINT 104
RETURN
101 FORMAT(A1,'[25;25H','Press <RETURN> key to continue ...'$)
102 FORMAT(A1)
103 FORMAT(A1,'[2J')
104 FORMAT(/15x,' DISCRETE FRACTURE NETWORK MODEL',
  .        /15x,' Implemented by Todd C. Rasmussen',
  .        /15x,' Department of Hydrology, Univ. of AZ')
END

```

```

* * * * *
*          Generate Fracture Attributes          *
* * * * *
SUBROUTINE generate
INCLUDE j:common.a
CHARACTER*10 readfile
common /prime/ mult, incr
DATA mult,iseed,incr / 123,456,789 /

CALL cls(2)
WRITE(*,200)
READ (*,100) readfile
OPEN (1,file=readfile)

* Determine number of fracture sets
READ (1,101) nsets

* Read in fracture network parameters for each fracture set
nfrct = 0
CALL cls(0)
DO 30 nset = 1, nsets
  READ (1,101) i
  READ (1,102) FRX, FRY, FRZ
  READ (1,102) FRstrk, FRdip, FRlth, FRtran
  READ (1,102) SDstrk, SDdip, SDlth, SDtran

* Echo input parameters
  WRITE (*,204) nset,i,frx, fry, frz, frstrk, sdstrk,
    .   frdip, sddip, frlth, sddlth, frtran, sdtran
  CALL cls(1)

* Generate fractures
DO 20 j = nfrct+1, nfrct+i

* Poisson locations
  xinp(1,j) = FRX * RANDUfn(iseed)
  xinp(2,j) = FRY * RANDUfn(iseed)
  xinp(3,j) = FRZ * RANDUfn(iseed)

* Spherical-normal orientations
  xinp(4,j) = FRstrk + SDstrk * RANDnrn(iseed)
  xinp(5,j) = FRdip + SDdip * RANDnrn(iseed)
  xinp(6,j) = 0.0
  DO 15 k = 4, 6
    IF (xinp(k,j) .LT. 0.) xinp(k,j) = xinp(k,j) + 360.
  15  IF (xinp(k,j) .GE. 360.) xinp(k,j) = xinp(k,j) - 360.

* Exponential lengths
  xinp(7,j) = - FRlth * DLOG(RANDUfn(iseed))
  xinp(8,j) = xinp(7,j)

* Log-normal transmissivities
  SQtran = FRtran * FRtran
  CVtran = SDtran * SDtran / SQtran
  AVtran = DLOG(SQtran/(1+CVtran))/2.
  VRtran = DLOG(1+CVtran)
  xinp(9,j) = EXP(AVtran+SQRT(VRtran) * RANDnrn(iseed))

* Shape is square.
  20  xinp(10,j) = 0.
  30  nfrct = nfrct + 1
CLOSE(1)

```

```

OPEN (9,file='data.out',status='new')
WRITE(9,102) xinp
CLOSE(9)

99 RETURN
100 FORMAT (a10)
101 FORMAT (8i10)
102 FORMAT (10f10.4)
200 FORMAT (//10x,' Enter name of input file: '$)
204 FORMAT(' Fracture Set: ',I3
.////' Number of Fractures: ',I5
.// ' Size of Prism (x,y,z): ',3F10.3
.// ' (X-Y) Rotation: ',F10.3,' Deviation: ',F10.3
.// ' (X-Z) Rotation: ',F10.3,' Deviation: ',F10.3
.// ' Fracture Length: ',F10.3,' Deviation: ',F10.3
.// ' Transmissivity: ',F10.3,' Deviation: ',F10.3//)
END

```

```

* * * * *
*          Generate uniform (0,1) pseudo-random numbers          *
*          An exponentially distributed r.v. is generated by:      *
*          RANDEXP = - mean * ALOG(RANDUfn(iseed))                *
* * * * *
FUNCTION RANDUfn(iseed)
IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
common /prime/ mult, incr
iseed = MOD(mult*iseed + incr,32768)
RANDUfn = FLOAT(iseed)/32768.
RETURN
END

* * * * *
*          Generate normal (0,1) numbers using the Polar method  *
*          A log-normally distributed r.v. is generated by:      *
*          RANDlgn = EXP(mean + stdev * RANDnrn(iseed))          *
* * * * *
FUNCTION RANDnrn(iseed)
IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
1  u = 2. * RANDUfn(iseed) - 1.
  v = 2. * RANDUfn(iseed) - 1.
  sum = u * u + v * v
  IF (sum .GE. 1.) GOTO 1
  RANDnrn = u * SQRT(-2. * DLOG(sum)/sum)
RETURN
END

```



```

        ndfr(j1,nfr(j1)) = nnode
        nfr(j2) = nfr(j2) + 1
        nfrn(j2,nfr(j2)) = j1
        ndfr(j2,nfr(j2)) = nnode
    ENDIF
  ENDIF
25 CONTINUE

*****
* Switch to boundaries
*****
DO 45 j1 = 1,ncut*jtotal
  nct(j1) = 0
  DO 45 j2 = 1,nfrot
    i2 = INT(xinp(10,j2))
    CALL intersect (4,xcut(1,j1),i2,xinp(1,j2),icheck,t1)
    IF (icheck .EQ. 2) THEN
      nbond = nbond + 1
      IF (nbond .GT. mx1) THEN
        PRINT *, ' Number of Allowed Boundary Nodes Exceeded'
        CALL cls(1)
        RETURN
      ENDIF
    * Convert from global to sample volume.
    DO 35 i = 1, 2
      DO 30 j = 1, 3
        t2(j) = t1(j,i)
30      t1(j,i) = 0.
      DO 35 j = 1, 3
        DO 35 k = 1, 3
35      t1(j,i) = t1(j,i) + cosA(k,j,1)
          * (t2(k) - cosO(k,1) + cosz(k,1)/2.)
        w = 0.
        DO 40 j = 1,3
          xx1(nbond,j) = t1(j,1)
          xx2(nbond,j) = t1(j,2)
40      nct(j1) = nct(j1) + 1
          nctn(j1,nct(j1)) = j2
          ndct(j1,nct(j1)) = nbond
          ncf(j2) = ncf(j2) + 1
          ncfn(j2,ncf(j2)) = j1
          ndcf(j2,ncf(j2)) = nbond
        ENDIF
45 CONTINUE
    RETURN
  END

```

```

*****
* Find line of intersection between two planar fractures. *
* The line segment is represented by its two end points. *
*****
SUBROUTINE intersect (n1,x1,n2,x2,icheck,p)
  IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
  DIMENSION x1(9),x2(9),p(3,2),tg1(3,2),tg2(3,2),t1(2,2),
    t12(2,2),tx(3,2),tp1(2,2),tp2(2,2),tout(3,4)
  DATA error /1,E-9/

  icheck = 0

  * Do they intersect?
  CALL supint (x1,x2,ic,tx)

  IF (ic .EQ. 0) RETURN

  * Convert to local coordinates
  CALL supgtl(2,tx,t1,x1)
  CALL supgtl(2,tx,t12,x2)

  * Fracture 1
  IF (N1 .EQ. 1) THEN
    CALL supcir(t1,x1(7),x1(8),ic1,tp1)
  ELSE
    CALL supret(t1,x1(7),x1(8),ic1,tp1)
  ENDIF

  * Fracture 2
  IF (N2 .EQ. 1) THEN
    CALL supcir(t12,x2(7),x2(8),ic2,tp2)
  ELSE
    CALL supret(t12,x2(7),x2(8),ic2,tp2)
  ENDIF

  IF ((ic1*ic2) .EQ. 0) RETURN

  * Convert to global
  CALL supltg (2,tp1,tg1,x1)
  CALL supltg (2,tp2,tg2,x2)

  * Convert to local
  CALL supgtl (2,tg1,tp1,x2)
  CALL supgtl (2,tg2,tp2,x1)

  DO 30 j = 1, 2
    iic = 0
    xx = tp1(1,j)
    yy = tp1(2,j)
    r1 = x2(7)
    r2 = x2(8)
    IF (N2 .EQ. 1) THEN
      IF (((xx*xx)/(r1*r1) + (yy*yy)/(r2*r2)) .LE. 1.) iic = 1
    ELSE
      IF ((DABS(xx) .LE. r1) .AND. (DABS(yy) .LE. r2)) iic = 1
    ENDIF
  END DO

```

```

      ENDIF
      IF (iic .EQ. 1) THEN
        icheck = icheck + 1
        DO 10 k = 1, 3
10      tout(k,icheck) = tg1(k,j)
      ENDIF
      iic = 0
      xx = tp2(1,j)
      yy = tp2(2,j)
      r1 = x1(7)
      r2 = x1(8)
      IF (N1 .EQ. 1) THEN
        IF (((xx*xx)/(r1*r1) + (yy*yy)/(r2*r2)) .LE. 1.) iic = 1
      ELSE
        IF ((DABS(xx) .LE. r1) .AND. (DABS(yy) .LE. r2)) iic = 1
      ENDIF
      IF (iic .EQ. 1) THEN
        icheck = icheck + 1
        DO 20 k = 1, 3
20      tout(k,icheck) = tg2(k,j)
      ENDIF
30      CONTINUE

      jc = 2
      IF (icheck .GT. 2) THEN
        iic = 1
        DO 40 j = 2, icheck
          IF ((DABS(tout(1,j) - tout(1,1)) .GT. error) .OR.
            (DABS(tout(2,j) - tout(2,1)) .GT. error) .OR.
            (DABS(tout(3,j) - tout(3,1)) .GT. error)) THEN
            iic = iic + 1
            jc = j
          ENDIF
40      CONTINUE
      IF (iic .NE. 1) THEN
        icheck = 2
      ELSE
        icheck = 0
      ENDIF
      ENDIF

      DO 50 j = 1, 3
        p(j,1) = tout(j,1)
50      p(j,2) = tout(j,jc)

      RETURN
      END

```

```

*****
* Truncates a finite line segment at the boundary of the sample *
* volume. Only the part of the line segment within the *
* sample volume is returned. *
*****
      SUBROUTINE truncate (tq,icheck)
$INCLUDE j:common.a
      DIMENSION s(6),tp(3,2),tq(3,2),tt(3)

      DO 10 i = 1, 3
        s(2*i-1) = -cosz(i,1) / 2.
10      s(2*i) = cosz(i,1) / 2.

      DO 20 i = 1,2
        DO 20 i1 = 1,3
          w = 0.
          DO 15 i2 = 1,3
15          w = w + corA(i2,i1,1) * (tq(i2,i) - cor0(i2,1))
20          tp(i1,i) = w

      icheck = 0
      DO 30 i = 1,3
        i1 = 2*i
        i2 = i1 - 1
30      IF (((tp(i,1) .GT. s(i1)) .AND. (tp(i,2) .GT. s(i1))) .OR.
          ((tp(i,1) .LT. s(i2)) .AND. (tp(i,2) .LT. s(i2)))) icheck = 1

      IF (icheck .EQ. 0) THEN
        DO 50 i = 1,3
          ij = 0
          i1 = i*2
          i2 = i*2-1
          ip1 = i+1
          ip2 = i+2
          IF(ip1 .GT. 3) ip1 = ip1-3
          IF(ip2 .GT. 3) ip2 = ip2-3
          IF((tp(i,1) .GT. s(i1)) .OR. (tp(i,2) .GT. s(i1))) THEN
            IF((tp(i,1) .GT. tp(i,2)) .AND. (tp(i,1) .GT. s(i1))) THEN
              tt(i) = s(i1)
              ij = 1
            ENDIF
            IF((tp(i,2) .GT. tp(i,1)) .AND. (tp(i,2) .GT. s(i1))) THEN
              tt(i) = s(i1)
              ij = 2
            ENDIF
            tt(ip1) = tp(ip1,1)+(tt(i)-tp(i,1))*(tp(ip1,2)-tp(ip1,1))
              / (tp(i,2)-tp(i,1))
            tt(ip2) = tp(ip2,1)+(tt(i)-tp(i,1))*(tp(ip2,2)-tp(ip2,1))
              / (tp(i,2)-tp(i,1))
          IF( (tt(ip1) .GT. s(ip1*2))
            .OR. (tt(ip1) .LT. s(ip1*2-1))
            .OR. (tt(ip2) .GT. s(ip2*2))
            .OR. (tt(ip2) .LT. s(ip2*2-1))) THEN
            icheck = 1
          ELSE

```

```

        icheck = 0
        DO 35 ik = 1,3
35      tp(ik,ij) = tt(ik)
        ENDIF
        ENDIF
        IF((tp(i,1) .LE. s(i2)) .OR. (tp(i,2) .LE. s(i2))) THEN
        IF((tp(i,2) .GT. tp(i,1)) .AND. (tp(i,1) .LE. s(i2))) THEN
            tt(i) = s(i2)
            ij = 1
        ENDIF
        IF((tp(i,1) .GT. tp(i,2)) .AND. (tp(i,2) .LE. s(i2))) THEN
            tt(i) = s(i2)
            ij = 2
        ENDIF
        tt(ip1) = tp(ip1,1)+(tt(i)-tp(i,1))*(tp(ip1,2)-tp(ip1,1))
            / (tp(i,2)-tp(i,1))
        tt(ip2) = tp(ip2,1)+(tt(i)-tp(i,1))*(tp(ip2,2)-tp(ip2,1))
            / (tp(i,2)-tp(i,1))
        IF ((tt(ip1) .GT. s(ip1*2)) .OR. (tt(ip1) .LT. s(ip1*2-1))
        .OR. (tt(ip2) .GT. s(ip2*2)) .OR. (tt(ip2) .LT. s(ip2*2-1))) THEN
            icheck = 1
        ELSE
            icheck = 0
            DO 40 ik = 1,3
40      tp(ik,ij) = tt(ik)
            ENDIF
        ENDIF
50      CONTINUE
        ENDIF
        DO 60 i = 1,2
        DO 60 i1 = 1,3
            w = 0.
            DO 55 i2 = 1,3
55      w = w + corA(i2,i1,1) * tp(i2,i)
60      tq(i1,i) = w + cor0(i1,1)

        RETURN
        END

```

```

* * * * *
* Find line of intersection between two planar fractures. *
* * * * *
SUBROUTINE supint(xan,xbn,ick,p)
IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-W)
DIMENSION xa(4),xb(4),r1(3),r2(3),p(3,2),xan(9),xbn(9)

DATA error / 1.E-9/
DATA radian / 0.0174532925199433/

xan4 = xan(4) * radian
xan5 = xan(5) * radian

xa(1) = COS(xan5) * COS(xan4)
xa(2) = COS(xan5) * SIN(xan4)
xa(3) = SIN(xan5)

xbn4 = xbn(4) * radian
xbn5 = xbn(5) * radian

xb(1) = COS(xbn5) * COS(xbn4)
xb(2) = COS(xbn5) * SIN(xbn4)
xb(3) = SIN(xbn5)

IF ((DABS(xa(1)*xb(2)-xb(1)*xa(2)) .LT. error) .AND.
. (DABS(xa(2)*xb(3)-xb(2)*xa(3)) .LT. error) .AND.
. (DABS(xa(3)*xb(1)-xb(3)*xa(1)) .LT. error)) THEN
    ick = 0
ELSE
    ick = 1

    xa(4) = xa(1)*xan(1) + xa(2)*xan(2) + xa(3)*xan(3)
    xb(4) = xb(1)*xbn(1) + xb(2)*xbn(2) + xb(3)*xbn(3)

    r00 = xa(1)*xb(2)-xb(1)*xa(2)

    IF (DABS(r00) .GT. error) THEN
        r1(1) = (xb(3)*xa(2)-xa(3)*xb(2))/r00
        r1(2) = (xb(1)*xa(3)-xa(1)*xb(3))/r00
        r1(3) = 1.

        r2(1) = (xb(2)*xa(4)-xa(2)*xb(4))/r00
        r2(2) = (xb(4)*xa(1)-xa(4)*xb(1))/r00
        r2(3) = 0.
    ELSE
        r00 = xa(2)*xb(3)-xb(2)*xa(3)
        IF (DABS(r00) .GT. error) THEN
            r1(1) = 1.
            r1(2) = (xb(1)*xa(3)-xa(1)*xb(3))/r00
            r1(3) = (xb(1)*xa(2)-xa(1)*xb(2))/r00

            r2(1) = 0.
            r2(2) = (xb(3)*xa(4)-xa(3)*xb(4))/r00
            r2(3) = (xb(4)*xa(2)-xa(4)*xb(2))/r00
        ELSE

```

```

      r00 = xa(3)*xb(1)-xb(3)*xa(1)

      r1(1) = (xb(3)*xa(2)-xa(3)*xb(2))/r00
      r1(2) = 1.
      r1(3) = (xb(2)*xa(1)-xa(2)*xb(1))/r00

      r2(1) = (xb(4)*xa(3)-xa(4)*xb(3))/r00
      r2(2) = 0.
      r2(3) = (xb(1)*xa(4)-xa(1)*xb(4))/r00
    ENDIF
  ENDIF
  DO 10 j = 1,3
    p(j,1) = r2(j) + r1(j)
10    p(j,2) = r2(j) - r1(j)
  ENDIF

  RETURN
  END

* * * * *
* Used to perform coordinate transformation from local 2-D region
* (defined for each fracture) to the global 3-D region
* (defined using the center of the sample region).
* * * * *
SUBROUTINE supltg(ni,xyl,xyag,xinp)
  IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
  DIMENSION xyl(2,ni),xyag(3,ni),xa2(3),xa3(3),xinp(9)
  DATA radian / 0.0174532925199433/

  alpha = xinp(4) * radian
  beta = xinp(5) * radian
  theta = xinp(6) * radian

  xa2(1) = SIN(beta)*COS(alpha)
  xa2(2) = SIN(beta)*SIN(alpha)
  xa2(3) = -COS(beta)

  xa3(1) = -SIN(alpha)
  xa3(2) = COS(alpha)
  xa3(3) = 0.0

  DO 10 n = 1,ni
    xb = xyl(1,n)*COS(theta) - xyl(2,n)*SIN(theta)
    yb = xyl(1,n)*SIN(theta) + xyl(2,n)*COS(theta)
    DO 10 j = 1, 3
10    xyag(j,n) = xinp(j) + xb*xa2(j) + yb*xa3(j)

  RETURN
  END

```

```

* * * * *
* Used to perform coordinate transformation from global 3-D region
* (defined using the sample region) to the local 2-D region
* (defined for each individual fracture plane).
* * * * *
SUBROUTINE supgtl(ni,tp,xyl,xcn)
  IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
  DIMENSION xyl(2,ni),tp(3,ni),xcn(9),xc2(3),xc3(3)
  DATA error / 1.D-6/
  DATA radian / 0.0174532925199433/

  alpha = xcn(4) * radian
  beta = xcn(5) * radian
  theta = xcn(6) * radian

  xc2(1) = SIN(beta)*COS(alpha)
  xc2(2) = SIN(beta)*SIN(alpha)
  xc2(3) = -COS(beta)

  xc3(1) = -SIN(alpha)
  xc3(2) = COS(alpha)
  xc3(3) = 0.0

  DO 10 n = 1, ni
    IF (DABS(xc2(3)) .GT. error) THEN
      xb = (tp(3,n)-xcn(3))/xc2(3)
    ELSE
      xb = (xc3(2)*(tp(1,n)-xcn(1))-xc3(1)*(tp(2,n)-xcn(2)))
        / (xc3(2)*xc2(1)-xc3(1)*xc2(2))
    ENDIF
    IF (DABS(xc3(1)) .GT. error) THEN
      yb = (tp(1,n)-xcn(1)-xb*xc2(1))/xc3(1)
    ELSE
      yb = (tp(2,n)-xcn(2)-xb*xc2(2))/xc3(2)
    ENDIF
10    xyl(1,n) = xb*COS(theta) + yb*SIN(theta)
    xyl(2,n) = -xb*SIN(theta) + yb*COS(theta)

  RETURN
  END

```

```

*****
* Find point of intersection between a line and the *
* boundary of a rectangle *
*****
SUBROUTINE supret(p,r1,r2,ic,cx)
IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
DIMENSION p(2,2), cx(2,2), ct(2,4), xx(4), yy(4)

error = 1.E-9

IF((DABS(r1) .LT. error) .OR. (DABS(r2) .LT. error)) THEN
  ic = 0
  RETURN
ENDIF

a = p(2,2) - p(2,1)
b = p(1,1) - p(1,2)
c = p(2,1)*p(1,2) - p(1,1)*p(2,2)

ic = 0

IF (DABS(a) .LT. error) THEN
  y = -c / b
  IF (DABS(y) .LE. r2) THEN
    ic = 1
    cx(1,1) = r1
    cx(1,2) = -r1
    cx(2,1) = y
    cx(2,2) = y
  ELSE
    ic = 0
  ENDIF
ELSE
  ELSE
    IF (DABS(b) .LT. error) THEN
      x = -c / a
      IF (DABS(x) .LE. r1) THEN
        ic = 1
        cx(1,1) = x
        cx(1,2) = x
        cx(2,1) = r2
        cx(2,2) = -r2
      ELSE
        ic = 0
      ENDIF
    ELSE
      ELSE
        xx(1) = r1
        yy(1) = (-a*r1-c)/b
        xx(2) = -r1
        yy(2) = (a*r1-c)/b
        yy(3) = r2
        xx(3) = (-b*r2-c)/a
        yy(4) = -r2
        xx(4) = (b*r2-c)/a
        icct = 0
        DO 10 ic = 1,4

```

```

IF((DABS(xx(ic)).LE. r1) .AND. (DABS(yy(ic)).LE. r2)) THEN
  icct = icct + 1
  ct(1,icct) = xx(ic)
  ct(2,icct) = yy(ic)
ENDIF
ENDIF
10 CONTINUE
IF (icct .EQ. 0) THEN
  ic = 0
ELSE
  cx(1,1) = ct(1,1)
  cx(2,1) = ct(2,1)
  jct = 1
  DO 15 j = 2,icct
    IF (ct(1,j) .NE. ct(1,1)) THEN
      cx(1,2) = ct(1,j)
      cx(2,2) = ct(2,j)
      jct = 2
    ENDIF
  ENDIF
15 CONTINUE
ic = 1
IF (jct .EQ. 1) THEN
  ic = 0
ENDIF
ENDIF
ENDIF
RETURN
END

```

```

*****
* Find point of intersection between a line and the *
* boundary of an ellipse *
*****
SUBROUTINE supcir(p,r1,r2,ic,ct)
IMPLICIT REAL*8(A-H,O-Z), INTEGER*4(I-N)
DIMENSION p(2,2), ct(2,2)

error = 1.E-9

IF((DABS(r1) .LT. error) .OR. (DABS(r2) .LT. error)) THEN
  ic = 0
  RETURN
ENDIF

a = p(2,2) - p(2,1)
b = p(1,1) - p(1,2)
c = p(2,1)*p(1,2) - p(1,1)*p(2,2)
IF (DABS(b) .GT. error) THEN
  r2b = r2*r2*b*b
  ax = 1./(r1*r1) + a*a / r2b
  bx = 2.*a*c / r2b
  cx = c*c / r2b - 1
  bac = bx*bx - 4.*ax*cx
  IF (bac .GT. 0) THEN
    ic = 1
    ct(1,1) = (-bx + DSQRT(bac)) / (2.*ax)
    ct(2,1) = (- a*ct(1,1) - c) / b
    ct(1,2) = (-bx - DSQRT(bac)) / (2.*ax)
    ct(2,2) = (- a*ct(1,2) - c) / b
  ELSE
    ic = 0
  ENDIF
ELSE
  r1a = r1*r1*a*a
  ax = 1./(r2*r2) + b*b / r1a
  bx = 2.*b*c / r1a
  cx = c*c / r1a - 1
  bac = bx*bx - 4.*ax*cx
  IF (bac .GT. 0) THEN
    ic = 1
    ct(2,1) = (-bx + DSQRT(bac)) / (2.*ax)
    ct(1,1) = (- b*ct(2,1) - c) / a
    ct(2,2) = (-bx - DSQRT(bac)) / (2.*ax)
    ct(1,2) = (- b*ct(2,2) - c) / a
  ELSE
    ic = 0
  ENDIF
ENDIF
RETURN
END

```

```

*****
* Find contributing fractures *
*****
SUBROUTINE findfractures
$INCLUDE j:common.a
DIMENSION itemp(mx1)

DO 5 i = 1, nfrct
  index(i) = 0
  5 itemp(i) = 0

* Find fractures connected to boundary surfaces
DO 15 i = 1, nout*jtotal
  IF (nct(i) .NE. 0) THEN
    DO 10 j = 1, nct(i)
      k = nctn(i,j)
      IF (itemp(k) .EQ. 0) THEN
        index(k) = i
        itemp(k) = 1
      ELSEIF (index(k) .NE. i) THEN
        itemp(k) = 2
      ENDIF
    10 CONTINUE
  15 CONTINUE

* Find fractures connected to fractures connected to boundaries
20 itest = 0
DO 40 i = 1, nfrct

  IF (itemp(i) .EQ. 0) THEN
    DO 25 j = 1, nfr(i)
      k = nfrn(i,j)
      IF (itemp(k) .EQ. 1) THEN
        itest = 1
        index(i) = index(k)
      ELSEIF (itemp(k) .EQ. 2) THEN
        itest = 1
        itemp(i) = 2
      ENDIF
    25 CONTINUE

    ELSEIF (itemp(i) .EQ. 1) THEN
      DO 30 j = 1, nfr(i)
        k = nfrn(i,j)
        IF (itemp(k) .EQ. 0) THEN
          itest = 1
          itemp(k) = 1
          index(k) = index(i)
        ELSEIF (itemp(k) .EQ. 1 .AND. index(i) .NE. index(k)) THEN
          itest = 1
          itemp(k) = 2
          itemp(i) = 2
        ELSEIF (itemp(k) .EQ. 2) THEN

```

```

        itest = 1
        itemp(i) = 2
    ENDIF
30    CONTINUE

    ELSEIF (itemp(i) .EQ. 2) THEN
        DO 35 j = 1, nfr(i)
            k = nfrn(i,j)
            IF (itemp(k) .NE. 2) THEN
                itest = 1
                itemp(k) = 2
            ENDIF
        ENDIF
35    CONTINUE

    ENDIF
40 CONTINUE

    IF (itest .EQ. 1) GOTO 20

*****
* Find fractures in a network with more than one outlet
*
    nd = 0
    DO 45 i = 1, nfrct
        IF (itemp(i) .EQ. 2) THEN
            nd = nd + 1
            index(nd) = i
        ENDIF
    ENDIF
45 CONTINUE
    RETURN
    END

*****
* Common file for FRACGEN program
*****

    IMPLICIT REAL*8(A-H,O-Z)
    PARAMETER (mx0=200,mx1=100,mx2=12,mx3=2,mx4=100,mx5=20)

* mx0: maximum number of boundary nodes in global network
* mx1: maximum number of fractures
* mx2: maximum number of boundary surfaces (6*mx3)
* mx3: maximum number of boundaries
* mx4: maximum number of boundary nodes in local networks
* mx5: maximum number of intersections

    COMMON /a/ oosZ(3,mx3),cor0(3,mx3),coal(2,mx3),corA(3,3,mx3),
    . nfrct,node,nfr(mx1),ncf(mx1),xinp(10,mx1),
    . ncfn(mx1,mx5),ndcf(mx1,mx5),nfrn(mx1,mx5),ndfr(mx1,mx5),
    . xx1(mx1,3),xx2(mx1,3),xi1(mx1,3),xi2(mx1,3),index(mx1),
    . jtotal,ncut,xcut(9,mx2),nbond,ncf(mx2),nctn(mx2,mx5),
    . ndct(mx2,mx5),nd

```

```

*****
* Print out nodal representation
*****

    SUBROUTINE listfractures
$INCLUDE j:common.a
    CHARACTER c(200),d,s,b
    DATA s,b / ' ', 'q' /
    DATA epsilon / 1.E-6 /

    CALL cls(0)
    IF (nnode .NE. 0) THEN
* Display FRACTURE-FRACTURE table by NNODE #
        icount = 0
        DO 10 k = 1, nnode
            IF (icount .EQ. 0) WRITE(*,201)
            WRITE(*,101) k,(xi1(k,i),i=1,3),(xi2(k,i),i=1,3)
            icount = icount + 1
            IF (icount .GE. 20) THEN
                CALL cls(1)
                icount = 0
            ENDIF
        ENDIF
10    CONTINUE
        IF (icount .NE. 0) CALL cls(1)
    ENDIF

    IF (nfrct .NE. 0) THEN
* Display FRACTURE-FRACTURE table by NFRCT #
        icount = 0
        DO 20 k = 1, nfrct
            IF (nfr(k) .GT. 0) THEN
                IF (icount .EQ. 0) WRITE(*,202)
                WRITE(*,102) k,nfr(k),(nfrn(k,i),i=1,nfr(k))
                icount = icount + 1
                IF (icount .GE. 21) THEN
                    CALL cls(1)
                    icount = 0
                ENDIF
            ENDIF
        ENDIF
20    CONTINUE
        IF (icount .NE. 0) CALL cls(1)

* Display FRACTURE-NNODE table by NFRCT #
        icount = 0
        DO 30 k = 1, nfrct
            IF (nfr(k) .GT. 0) THEN
                IF (icount .EQ. 0) WRITE(*,203)
                WRITE(*,102) k,nfr(k),(ndfr(k,i),i=1,nfr(k))
                icount = icount + 1
                IF (icount .GE. 21) THEN
                    CALL cls(1)
                    icount = 0
                ENDIF
            ENDIF
        ENDIF
30    CONTINUE
        IF (icount .NE. 0) CALL cls(1)
    ENDIF

```

```

ENDIF
IF (nbond .NE. 0) THEN
* Display FRACTURE-BOUNDARY table by NBOND #
icount = 0
DO 40 k = 1, nbond
IF (icount .EQ. 0) WRITE(*,204)
WRITE(*,101) k,(xx1(k,i),i=1,3),(xx2(k,i),i=1,3)
icount = icount + 1
IF (icount .GE. 20) THEN
CALL cls(1)
icount = 0
ENDIF
40 CONTINUE
IF (icount .NE. 0) CALL cls(1)
ENDIF

* Display BOUNDARY-FRACTURE table by NCUT #
IF (jtotal .NE. 0) THEN
icount = 0
DO 50 k = 1, ncut*jtotal
IF (nct(k) .GT. 0) THEN
IF (icount .EQ. 0) WRITE(*,205)
WRITE(*,102) k,nct(k),(nctn(k,i),i=1,nct(k))
icount = icount + 1
IF (icount .GE. 21) THEN
CALL cls(1)
icount = 0
ENDIF
ENDIF
50 CONTINUE
IF (icount .NE. 0) CALL cls(1)

* Display BOUNDARY-NBOND table by NCUT #
icount = 0
DO 60 k = 1, ncut*jtotal
IF (nct(k) .GT. 0) THEN
IF (icount .EQ. 0) WRITE(*,206)
WRITE(*,102) k,nct(k),(nctn(k,i),i=1,nct(k))
icount = icount + 1
IF (icount .GE. 21) THEN
CALL cls(1)
icount = 0
ENDIF
ENDIF
60 CONTINUE
IF (icount .NE. 0) CALL cls(1)
ENDIF

IF (nfrct .NE. 0) THEN
* Display FRACTURE-BOUNDARY table by NFRCT #
icount = 0
DO 70 k = 1, nfrct
IF (ncf(k) .GT. 0) THEN
IF (icount .EQ. 0) WRITE(*,207)
WRITE(*,102) k,ncf(k),(ncfn(k,i),i=1,ncf(k))
icount = icount + 1
IF (icount .GE. 21) THEN
CALL cls(1)
icount = 0
ENDIF
ENDIF
70 CONTINUE
IF (icount .NE. 0) CALL cls(1)

* Display FRACTURE-NBOND table by NFRCT #
icount = 0
DO 80 k = 1, nfrct
IF (ncf(k) .GT. 0) THEN
IF (icount .EQ. 0) WRITE(*,208)
WRITE(*,102) k,ncf(k),(ndcf(k,i),i=1,ncf(k))
icount = icount + 1
IF (icount .GE. 21) THEN
CALL cls(1)
icount = 0
ENDIF
ENDIF
80 CONTINUE
IF (icount .NE. 0) CALL cls(1)
ENDIF
RETURN
100 FORMAT(1x,200A1)
101 FORMAT(13,6F10.3)
102 FORMAT(16I5)

201 FORMAT(27x,'INTERNAL NODES '
. //4x,'N',
. - - - Location - - - - - Location - - -'
. /' (x) (y) (z) (x) (y) (z)')
202 FORMAT(27x,'INTERNAL NODES '
. //4x,'F' $ F F F F F ...')
203 FORMAT(27x,'INTERNAL NODES '
. //4x,'F' $ N N N N N ...')
204 FORMAT(27x,'EXTERNAL NODES '
. //x,'B' - - - Location - - - - - Location - - -'
. /10x,'(x) (y) (z) (x) (y) (z)')
205 FORMAT(27x,'EXTERNAL NODES '
. //4x,'S' $ F F F F F ...')
206 FORMAT(27x,'EXTERNAL NODES '
. //4x,'S' $ B B B B B ...')
207 FORMAT(27x,'EXTERNAL NODES '
. //4x,'F' $ S S S S S ...')
208 FORMAT(27x,'EXTERNAL NODES '
. //4x,'F' $ B B B B B ...')
END

```

A.3 BIM2D: Boundary Integral Method for Two Dimensional Fracture Flow

This section presents the source code for the boundary integral method applied to two dimensional fracture flow through a discrete fracture network (BIM2D). Chapter 2 describes how the program calculates steady fluid flow for saturated fractures. Chapter 3 describes the technique used to solve for steady unsaturated flow. Chapter 4 presents the methodology used for calculating tracer breakthrough curves. Inputs to the program were described in Section A.2.

```

*****
*           2-D BIM Discrete Fracture Network Model           *
*           (Last Modification - August 23, 1987)             *
*****
PROGRAM BIM2D
$INCLUDE j:common.a
$INCLUDE j:common.b
CHARACTER*10 readfile

* Write initial screen.
PRINT 202, CHAR(27)
OPEN(9, file='savefile', form='binary')
READ(9) cos2, cor0, coal, corA, nfrct, nnode, nfr, ncf, xinp, ncfm,
.      ndcf, nfrn, ndfr, xx1, xx2, xi1, xi2, index, jtotal, ncut,
.      xcut, nbond, nct, nctn, ndct, nd
CLOSE(9)
CALL cls(1)
10 CALL cls(2)
WRITE(*,200)
READ(*,100) itype
IF (itype .EQ. 1) THEN
  CALL boundary
ELSEIF (itype .EQ. 2) THEN
  CALL buildmatrices
  CALL global
  CALL gauss
  CALL display
ELSEIF (itype .EQ. 3) THEN
  CALL points
ELSEIF (itype .EQ. 9) THEN
  CALL cls(2)
STOP
ENDIF
GOTO 10
100 FORMAT (8i10)
101 FORMAT (a10)
102 FORMAT (10f10.3)
200 FORMAT (//15x,'          MAIN MENU'
.      //15x,' 1 - Assign Boundary Conditions',
.      //15x,' 2 - Generate and Solve BEIM Matrix',
.      //15x,' 3 - Solve at Interior Points',
.      //15x,' 9 - Exit to DOS',
.      //15x,'          Enter Selection: '$)
201 FORMAT (//10x,' Enter name of data file: '$)

202 FORMAT(A1,'[2J',////
. //10x,'          PROGRAM TO ESTIMATE HYDRAULIC PROPERTIES'
. //10x,'          OF DISCRETE FRACTURE NETWORKS'
. //10x,'          Developed by Todd C. Rasmussen (6/86)'
. //10x,'          Department of Hydrology & Water Resources'
. //10x,'          University of Arizona, Tucson 85721')
END

```

The program presented here was prepared in support of research activities and is not intended for other uses. Neither the United States Government, the University of Arizona nor any of their employees, make any warranty, expressed or implied, or assume any legal liability or responsibility for any third party's use, or the results of such use, of any portion of this program or represents that its use by such third party would not infringe privately owned rights.

```

*****
*           Pause output and clear screen           *
*****
SUBROUTINE cls(n)
  IF (n .EQ. 1) THEN
    WRITE(*,101) CHAR(27)
    READ (*,102)
  ENDIF
  WRITE(*,103) CHAR(27)
  IF (n .EQ. 2) WRITE(*,104)
  RETURN
101 FORMAT(A1,'[25;25H','Press <RETURN> key to continue ...'$)
102 FORMAT(A1)
103 FORMAT(A1,'[2J')
104 FORMAT(/15x,' DISCRETE FRACTURE NETWORK MODEL',
.         /15x,' Implemented by Todd C. Rasmussen',
.         /15x,' Department of Hydrology, Univ. of AZ')
END

```

```

*****
*           Assign boundary conditions               *
*****
SUBROUTINE boundary
$INCLUDE j:common.a
$INCLUDE j:common.b

  CALL cls(0)
  n = 0
  WRITE(*,200)
  DO 10 j = 1, jtotal
    DO 10 k = 1, ncut
      n = n + 1
      WRITE(*,201) j, k
10    READ (*,*) nctype(m), bvalue(m)

  RETURN
200 FORMAT (/10x,' Enter boundary type and value',
.         /10x,' * 0 = prescribed flux',
.         /10x,' * 1 = prescribed total head',
.         /10x,' * 2 = constant pressure head'//)
201 FORMAT (2i5,10x,'(i,k): '$)
END

```

```

*****
*           Build Boundary Element Matrix           *
*****
SUBROUTINE buildmatrices
$INCLUDE j:common.a
$INCLUDE j:common.b
  DIMENSION t1(2), t2(3)
  DIMENSION xsign(8), ysign(8)
  DATA xsign / 1, 1, 0,-1,-1,-1, 0, 1/
  DATA ysign / 0, 1, 1, 1, 0,-1,-1,-1/
  DATA iseed / 12345 /

  nn = 0
*****
* Loop through all contributing fractures
*
  IF (nd .NE. 0) THEN
    DO 60 i = 1, nd
      ii = index(i)
      r1 = xinp(7,ii)
      r2 = xinp(8,ii)
* Elliptic vs. rectangular fractures
      IF (xinp(10,ii) .EQ. 1.) THEN
        nc(1,i) = 8
        r12 = DSQRT(1/(1/(r1*r1) + 1/(r2*r2)))
        DO 5 j = 1,8,2
          x(j,i) = xsign(j) * r1
          y(j,i) = ysign(j) * r2
          x(j+1,i) = xsign(j+1) * r12
          y(j+1,i) = ysign(j+1) * r12
5        ELSE
          nc(1,i) = 8
          DO 10 j = 1, 8
            x(j,i) = xsign(j) * r1
            y(j,i) = ysign(j) * r2
10        ENDIF
        nm(i) = 1
*****
* Loop through all intersecting fractures
*
        IF (nfr(ii) .NE. 0) THEN
          DO 37 j = 1, nfr(ii)
            nm(i) = nm(i) + 1
            nc(nm(i),i) = nc(nm(i)-1,i) + 2
            DO 20 m = 1, 2
              jj = nc(nm(i)-1,i) + m
              IF (m .EQ. 1) THEN
                DO 15 k = 1, 3
                  t2(k) = xi1(ndfr(ii,j),k)
15              ELSE
                DO 17 k = 1, 3
                  t2(k) = xi2(ndfr(ii,j),k)
17              ENDIF
            CALL supgtl(1,t2,t1,xinp(1,ii))

```

```

      x(jj,i) = t1(1)
      y(jj,i) = t1(2)
      IF (INT(xinp(10,ii)) .EQ. 1) THEN
        IF (((t1(1)*t1(1))/(r1*r1)+(t1(2)*t1(2))/(r2*r2)).EQ.1.)
          CALL insertnode(t1(1),t1(2),i)
        ELSE
          IF ((DABS(t1(1)) .EQ. r1) .OR. (DABS(t1(2)) .EQ. r2))
            CALL insertnode(t1(1),t1(2),i)
          ENDDIF
        20 CONTINUE
      * Find ends of fractures
      IF (i .NE. 1) THEN
        DO 35 k = 1, i-1
          IF (index(k) .EQ. nfrn(ii,j)) THEN
            DO 30 l = 1, nfr(index(k))
              IF (nfr(index(k),l) .EQ. ii) THEN
                DO 25 m = 1, 2
                  jj = nc(nm(i)-1,i) + m
                  kk = nc(l,k) + m
                  u(jj,i) = 0.
                  u(kk,k) = 0.
                  kode(jj,i) = -(128*k + kk)
                25 kode(kk,k) = -(128*i + jj)
              ENDDIF
            30 CONTINUE
          ENDDIF
        35 CONTINUE
      ENDDIF
    37 CONTINUE
  ENDDIF

*****
* Loop through all intersecting boundaries
*
  IF (ncf(ii) .NE. 0) THEN
    DO 45 j = 1, ncf(ii)
      nn(i) = nn(i) + 1
      nc(nm(i),i) = nc(nm(i)-1,i) + 2
      DO 40 m = 1, 2
        jj = nc(nm(i)-1,i) + m
        IF (m .EQ. 1) THEN
          DO 40 k = 1, 3
            40 t2(k) = xx1(ncf(ii,j),k)
          ELSE
            DO 42 k = 1, 3
              42 t2(k) = xx2(ncf(ii,j),k)
            ENDDIF
          CALL supgt1(1,t2,t1,xinp(1,ii))
          x(jj,i) = t1(1)
          y(jj,i) = t1(2)
        * Apply boundary conditions
        IF (nbtype(ncfn(ii,j)) .EQ. 0) THEN
          kode(jj,i) = 1
          u(jj,i) = bvalue(ncfn(ii,j))
        ELSEIF (nbtype(ncfn(ii,j)) .EQ. 1) THEN
          kode(jj,i) = 0
          u(jj,i) = bvalue(ncfn(ii,j))
        ELSE
          kode(jj,i) = 0
          u(jj,i) = bvalue(ncfn(ii,j)) + x(jj,i)
        ENDDIF
      45 CALL insertnode (t1(1),t1(2),i)
    ENDDIF

    DO 50 j = 1, nc(1,i)
      kode(j,i) = 1
    50 u(j,i) = 0.

*****
* Find total number of nodes.
*
    nj(i) = nn
    n(i) = nc(nm(i),i)
    nn = nn + n(i)
    IF (nn .GT. mx0) THEN
      PRINT *, ' Total # of nodes exceeded: ', nn
      CALL cis(1)
      RETURN
    ENDDIF
  60 CONTINUE
  ENDDIF
  RETURN
  END

```

```

*****
*   Insert nodes along circumference of fracture   *
*****
SUBROUTINE insertnode (xx,yy,i)
$INCLUDE j:common.a
$INCLUDE j:common.b

  IF (yy .NE. 0) thetaB = xx/yy
  IF (yy .EQ. 0) thetaB = 1.D10

  IF (xx .GE. 0 .AND. yy .GT. 0) jcase = 1
  IF (xx .LT. 0 .AND. yy .GE. 0) jcase = 2
  IF (xx .LE. 0 .AND. yy .LT. 0) jcase = 3
  IF (xx .GT. 0 .AND. yy .LE. 0) jcase = 4

  DO 20 k = 1, nc(1,i)
    IF (y(k,i) .NE. 0) thetaF = x(k,i)/y(k,i)
    IF (y(k,i) .EQ. 0) thetaF = 1.D10

    IF (x(k,i) .GE. 0 .AND. y(k,i) .GT. 0) icase = 1
    IF (x(k,i) .LT. 0 .AND. y(k,i) .GE. 0) icase = 2
    IF (x(k,i) .LE. 0 .AND. y(k,i) .LT. 0) icase = 3
    IF (x(k,i) .GT. 0 .AND. y(k,i) .LE. 0) icase = 4

    IF (icase .EQ. jcase .AND. thetaB .GT. thetaF) THEN
      DO 10 j = nc(na(i),i), k, -1
        kode(j+1,i) = kode(j,i)
        IF (kode(j,i) .LT. 0) THEN
          mk = -kode(j,i) / 128
          mj = MOD(-kode(j,i),128)
          kode(mj,mk) = kode(mj,mk) - 1
        ENDIF
        u(j+1,i) = u(j,i)
        q(j+1,i) = q(j,i)
        x(j+1,i) = x(j,i)
        y(j+1,i) = y(j,i)
      10 DO 15 j = 1, na(i)
        nc(j,i) = nc(j,i) + 1
        x(k,i) = xx
        y(k,i) = yy
        RETURN
      ENDIF
    20 CONTINUE
    RETURN
  END

```

```

*****
*   Used to perform coordinate transformation from global 3-D region   *
*   (defined using the sample region) to the local 2-D region       *
*   (defined for each individual fracture plane).                     *
*****
SUBROUTINE supgtl(ni,tp,xyl,xcn)
IMPLICIT REAL*8(A-N,O-Z), INTEGER*4(I-N)
DIMENSION xyl(2,ni),tp(3,ni),xcn(9),xc2(3),xc3(3)
DATA error / 1.D-6/
DATA radian / 0.0174532925199433/

alpha = xcn(4) * radian
beta  = xcn(5) * radian
theta = xcn(6) * radian

xc2(1) = SIN(beta)*COS(alpha)
xc2(2) = SIN(beta)*SIN(alpha)
xc2(3) = -COS(beta)

xc3(1) = -SIN(alpha)
xc3(2) = COS(alpha)
xc3(3) = 0.0

DO 10 n = 1, ni
  IF (DABS(xc2(3)) .GT. error) THEN
    xb = (tp(3,n)-xcn(3))/xc2(3)
  ELSE
    xb = (xc3(2)*(tp(1,n)-xcn(1))-xc3(1)*(tp(2,n)-xcn(2)))
      / (xc3(2)*xc2(1)-xc3(1)*xc2(2))
  ENDIF
  IF (DABS(xc3(1)) .GT. error) THEN
    yb = (tp(1,n)-xcn(1)-xb*xc2(1))/xc3(1)
  ELSE
    yb = (tp(2,n)-xcn(2)-xb*xc2(2))/xc3(2)
  ENDIF
  xyl(1,n) = xb*COS(theta) + yb*SIN(theta)
  10 xyl(2,n) = -xb*SIN(theta) + yb*COS(theta)

RETURN
END

```

```

*****
*   Boundary Element program adapted from Brebbia (1978)   *
*   Solves system of equations of the form H U = G Q      *
*****
      SUBROUTINE global
$INCLUDE j:common.a
$INCLUDE j:common.b

* Clear QQ and GG
      DO 25 i = 1, mx0
        qq(i) = 0.
        DO 25 j = 1, mx0
          25   gg(i,j) = 0.

* Loop through all domains
      DO 50 i = 1, nd
        t = xinp(9,index(i))

* Clear G and H
        DO 10 j = 1, n(i)
          DO 10 k = 1, n(i)
            h(j,k) = 0.
          10   g(j,k) = 0.

        DO 20 j = 1, n(i)
          DO 20 k = 1, n(i)
            l = next(k,nc(1,i),nm(i))
            m = last(k,nc(1,i),nm(i))
            IF ((j .NE. k) .AND. (j .NE. 1)) THEN
              CALL integral (t,x(j,i),y(j,i),x(k,i),y(k,i),x(1,i),y(1,i),
                a1,a2,b1,b2)
              h(j,k) = h(j,k) + a1
              g(j,k) = g(j,k) + b1
              h(j,1) = h(j,1) + a2
              g(j,1) = g(j,1) + b2
              h(j,j) = h(j,j) - a1 - a2
            ELSE
              ax = x(1,i) - x(k,i)
              ay = y(1,i) - y(k,i)
              sr = DSQRT(ax*ax + ay*ay)
              b1 = sr * (1.5 - DLOG(sr/DSQRT(t))) / 2. / t
              b2 = sr * (0.5 - DLOG(ax/DSQRT(t))) / 2. / t
              IF (k .NE. j) THEN
                g(j,k) = g(j,k) + b2
                g(j,1) = g(j,1) + b1
              ELSE
                g(j,k) = g(j,k) + b1
                g(j,1) = g(j,1) + b2
              ENDIF
            ENDIF
          20   CONTINUE

        l = nj(i)
        DO 50 k = 1, n(i)
          m = kode(k,i)

```

```

      nj = MOD(-m,128)
      mk = -m/128
      ml = nj(mk)+mj
      IF (m .GT. 0) THEN
        DO 30 j = 1, n(i)
          qq(1+j) = qq(1+j) - g(j,k) * u(k,i)
          gg(1+j,1+k) = - h(j,k)
        30   ELSEIF (m .EQ. 0) THEN
          DO 35 j = 1, n(i)
            qq(1+j) = qq(1+j) + h(j,k) * u(k,i)
            gg(1+j,1+k) = g(j,k)
        35   ELSEIF (i .LT. mk) THEN
          DO 40 j = 1, n(i)
            gg(1+j,ml) = -h(j,k)
            gg(1+j,1+k) = g(j,k)
        40   ELSE
          DO 45 j = 1, n(i)
            gg(1+j,ml) = -g(j,k)
            gg(1+j,1+k) = -h(j,k)
        45   ENDIF
      50   CONTINUE
      RETURN
      END

```

```

* * * * *
* Compute off-diagonal elements of G and H by
* numerical integration along boundary elements.
* * * * *
SUBROUTINE integral (t,x,y,x1,y1,x2,y2,a1,a2,b1,b2)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION f(6), w(6)
DATA f /0.125233408511469, 0.367831498998180, 0.587317954286617,
. 0.769902674194305, 0.904117256370475, 0.981560634246719/
DATA w /0.249147045813403, 0.233492536538355, 0.203167426723066,
. 0.160078328543346, 0.106939325995318, 0.047175336386512/

ax = (x2 - x1) / 2.
ay = (y2 - y1) / 2.
bx = (x2 + x1) / 2.
by = (y2 + y1) / 2.
IF (ax .NE. 0) THEN
  ta = ay / ax
  dist = DABS((ta*x - y + y1 - ta*x1) / DSQRT(ta*ta + 1))
ELSE
  dist = DABS(x - x1)
ENDIF
IF ((x1-x)*(y2-y) .LT. (x2-x)*(y1-y)) dist = -dist
a1 = 0.
a2 = 0.
b1 = 0.
b2 = 0.

DO 10 i = 1, 6
  qi = f(i)
  DO 10 j = 1, 2
    IF (j .EQ. 2) qi = -qi
    xx = x - (ax * qi + bx)
    yy = y - (ay * qi + by)
    ra = DSQRT(xx*xx + yy*yy)
    ar = DSQRT(ax*ax + ay*ay)

    g = ar * w(i) * DLOG(ra/DSQRT(t)) / 2. / t
    h = ar * w(i) * dist / (ra*ra) / 2.

    a1 = a1 + h * (qi - 1.)
    a2 = a2 - h * (qi + 1.)
    b1 = b1 + g * (qi - 1.)
10  b2 = b2 - g * (qi + 1.)

RETURN
END

```

```

* * * * *
* Find subsequent node
* * * * *
INTEGER function next (j,nc,m)
DIMENSION nc(*)
next = j + 1
IF (j .EQ. nc(1)) THEN
  next = 1
ELSEIF (m .GT. 1) THEN
  DO 10 k = 2, m
10  IF (j .EQ. nc(k)) next = nc(k-1) + 1
ENDIF
RETURN
END

* * * * *
* Find previous node
* * * * *
INTEGER function last (j,nc,m)
DIMENSION nc(*)
last = j - 1
IF (j .EQ. 1) THEN
  last = nc(1)
ELSEIF (m .GT. 1) THEN
  DO 10 k = 2, m
10  IF (j .EQ. nc(k-1)+1) last = nc(k)
ENDIF
RETURN
END

```

```

.....
* Gaussian elimination
.....
SUBROUTINE gauss
$INCLUDE j:common.a
$INCLUDE j:common.b
DATA epsilon / 1.D-10/

ierrow = 0
DO 50 l = 1, nm-1
* Exchange rows if zero in diagonal
IF (DABS(qg(l,l)) .LT. epsilon) THEN
DO 20 k = l+1, nm
IF (DABS(qg(k,l)) .GT. epsilon) THEN
DO 10 j = l, nm
c = qg(l,j)
qg(l,j) = qg(k,j)
10 qg(k,j) = c
c = qg(l)
qg(l) = qg(k)
qg(k) = c
GO TO 30
ENDIF
20 CONTINUE
* Can't find non-zero to exchange with, singular matrix
PRINT *, ' * * Singularity * * In row', l
CALL cls(1)
RETURN
ENDIF
* Divide row by diagonal coefficient
30 c = qg(l,l)
qg(l) = qg(l) / c
DO 40 k = l+1, nm
40 qg(l,k) = qg(l,k) / c
* Eliminate unknown Q(L) from row J
DO 50 j = l+1, nm
c = qg(j,l)
qg(j) = qg(j) - c * qg(l)
DO 50 k = l+1, nm
50 qg(j,k) = qg(j,k) - c * qg(l,k)
* Compute last unknown
IF (DABS(qg(nm,nm)) .LE. epsilon) THEN
PRINT *, ' * * Singularity * * In row', nm, qg(nm,nm)
CALL cls(1)
RETURN
ELSE
qg(nm) = qg(nm) / qg(nm,nm)
ENDIF
* Back substitute
DO 60 j = 1, nm-1
l = nm-j
DO 60 k = l+1, nm
60 qg(l) = qg(l) - qg(l,k) * qg(k)
RETURN
END

```

```

.....
* Reduce and Display Global Equation
.....
SUBROUTINE display
$INCLUDE j:common.a
$INCLUDE j:common.b

sum = 0.
DO 30 i = 1, nd
DO 10 j = 1, n(i)

m = node(j,i)
nj = MOD(-m,128)
mk = -m/128
k = nj(mk)*mj
l = nj(i) + j

IF (m .GT. 0) THEN
q(j,i) = u(j,i)
u(j,i) = q(i)
ELSEIF (m .EQ. 0) THEN
q(j,i) = q(i)
ELSEIF (i .LT. mk) THEN
q(j,i) = q(i)
u(j,i) = q(k)
ELSE
q(j,i) = -q(k)
u(j,i) = q(l)
ENDIF
10 CONTINUE

* Display results
jj = 1
sum = 0.
OPEN(1,file='output')
WRITE (1,601) index(i)
DO 20 j = 1, n(i)

* Fix page break
IF (MOD(j-1,20) .EQ. 0) THEN
IF (j .NE. 1) CALL cls(1)
IF (j .EQ. 1) CALL cls(0)
WRITE (*,601) index(i)
ENDIF

* Adjust discharges by length
l = next(j,nc(1,i),nm(i))
k = last(j,nc(1,i),nm(i))
dl = DSQRT(DABS(x(j,i)-x(l,i))**2 + DABS(y(j,i)-y(l,i))**2)
dk = DSQRT(DABS(x(j,i)-x(k,i))**2 + DABS(y(j,i)-y(k,i))**2)
d = (3.*q(j,i)*(dl+dk) + dl*q(l,i) + dk*q(k,i))/8.
sum = sum + d
IF (j .EQ. nc(jj,i)+1) THEN
sum = 0.
jj = jj + 1
ENDIF

```

```

      sum = sum + d
* Display locations, potentials, discharges, and mass balance
  WRITE (*,602) i,j,jj,x(j,i),y(j,i),u(j,i),q(j,i),d,sum
20  WRITE (1,602) i,j,jj,x(j,i),y(j,i),u(j,i),q(j,i),d,sum
  WRITE (*,603) sum
  WRITE (1,603) sum
30  CALL c1s(1)
    CLOSE(1)

    RETURN
601  FORMAT (' Fracture: ',i5,
. /4x,'i',4x,'n',4x,'m',7x,'x',9x,'y',9x,'u',9x,'q',9x,'d',9x,'s'//)
602  FORMAT (3i5,6f10.3)
603  FORMAT (' Mass balance: ',f10.4)
    END

```

```

* * * * *
*           Compute potential at interior points
* * * * *
      SUBROUTINE points
$INCLUDE j:common.a
$INCLUDE j:common.b
      pi = DACOS(-1.D0)

* Input domain and location
1  CALL c1s(0)
  WRITE(*,201)
  READ (*,*) i
  IF (i .GT. nd) GOTO 1
  IF (i .EQ. 0) RETURN
  t = xinp(9,index(i))
2  WRITE(*,202)
  READ (*,*) cx, cy
  IF (cx .EQ. -99.) GOTO 1
  sum = 0.
  DO 10 k = 1, n(i)
    i = next(k,nc(1,i),na(i))
    CALL integral(t,cx,cy,x(k,i),y(k,i),x(1,i),y(1,i),a1,a2,b1,b2)
10  sum=sum+((b1*q(k,i)+b2*q(1,i))-(a1*u(k,i)+a2*u(1,i)))/(2.*pi)
  WRITE (*,203) nd, cx, cy, sum
  GOTO 2
201  FORMAT (' Enter Domain Number ("0" to end): '$)
202  FORMAT (' Enter (x,y) coordinates ("-99.,-99." to end): '$)
203  FORMAT (' Domain: ',i3,' (x,y): (' ,2f10.3,' ) Potential: ',f10.3)
    END

```

```

* * * * *
*           COMMON file for SIM2D
* * * * *

COMMON /b/ nbtype(mx2),bvalue(mx2),
.          nn,n(mx4),nj(mx4),nm(mx4),nc(mx4,mx4),x(mx4,mx4),
.          y(mx4,mx4),u(mx4,mx4),q(mx4,mx4),h(mx4,mx4),
.          g(mx4,mx4),h(mx4,mx4),qq(mx0),gg(mx0,mx0)

```

A.4 BIM3D: Boundary Integral Method for Three Dimensional Coupled Fracture-Matrix Flow

This section presents the source code for the boundary integral method applied to three dimensional flow through a discrete fracture network embedded within a porous matrix (BIM3D). Chapter 2 describes how the program calculates steady fluid flow for saturated fractures. Inputs to the program are presented as Table A.8.

Table A.8: Sample Input Data for Program BIM3D.

1				
35				
0.	0.	0.	1	0.
0.5	0.	0.	1	0.
0.5	1.	0.	1	0.
0.	1.	0.	1	0.
0.25	0.5	0.	1	0.
0.	0.	1.	1	0.
0.5	0.	1.	1	0.
0.5	1.	1.	1	0.
0.	1.	1.	1	0.
0.25	0.5	1.	1	0.
0.	0.	0.	1	0.
0.5	0.	0.	1	0.
0.5	0.	1.	1	0.
0.	0.	1.	1	0.
0.25	0.	0.5	1	0.
0.	1.	0.	1	0.
0.5	1.	0.	1	0.
0.5	1.	1.	1	0.
0.	1.	1.	1	0.
0.25	1.	0.5	1	0.
0.	0.	0.		0.
0.	0.	1.		0.
0.	1.	1.		0.
0.	1.	0.		0.
0.	0.5	0.5		0.
0.5	0.	0.		5.
0.5	0.	1.		5.
0.5	1.	1.		5.
0.5	1.	0.		5.
0.5	0.75	0.5		5.
0.5	0.50	1.		5.
0.5	0.50	0.		5.
0.5	0.25	0.5		5.
0.5	0.50	0.	1	0.
0.5	0.50	1.	1	0.

Table A.8 (Continued):

1.

30		
1	4	5
2	1	5
34	2	5
3	34	5
4	3	5
6	7	10
7	35	10
35	8	10
8	9	10
9	6	10
11	12	15
12	13	15
13	14	15
14	11	15
16	19	20
17	16	20
18	17	20
19	18	20
21	22	25
22	23	25
23	24	25
24	21	25
32	29	30
31	32	30
28	31	30
29	28	30
32	31	33
26	32	33
27	26	33
31	27	33

Table A.8 (Continued):

30				
0.	0.	0.	1	0.
0.5	0.	0.	1	0.
0.5	0.5	0.	1	0.
0.	0.5	0.	1	0.
0.25	0.25	0.	1	0.
0.	0.	1.	1	0.
0.5	0.	1.	1	0.
0.5	0.5	1.	1	0.
0.	0.5	1.	1	0.
0.25	0.25	1.	1	0.
0.	0.	0.	1	0.
0.5	0.	0.	1	0.
0.5	0.	1.	1	0.
0.	0.	1.	1	0.
0.25	0.	0.5	1	0.
0.	0.5	0.	1	0.
0.5	0.5	0.	1	0.
0.5	0.5	1.	1	0.
0.	0.5	1.	1	0.
0.25	0.5	0.5	1	0.
0.	0.	0.		5.
0.	0.	1.		5.
0.	0.5	1.		5.
0.	0.5	0.		5.
0.	0.25	0.5		5.
0.5	0.	0.	-154	0.
0.5	0.	1.	-155	0.
0.5	0.5	1.	-159	0.
0.5	0.5	0.	-160	0.
0.5	0.25	0.5	-161	0.

Table A.8 (Continued):

100.

24		
1	4	5
2	1	5
3	2	5
4	3	5
6	7	10
7	8	10
8	9	10
9	6	10
11	12	15
12	13	15
13	14	15
14	11	15
16	19	20
17	16	20
18	17	20
19	18	20
21	22	25
22	23	25
23	24	25
24	21	25
27	26	30
26	29	30
29	28	30
28	27	30

```

* * * * *
* 3-D Boundary Element program adapted from Khabib (1978) *
* Solves system of equations of the form H U = G Q *
* * * * *
PROGRAM BIN3D
$INCLUDE j:common
CHARACTER*10 readfile
CHARACTER*80 chr

* Read Data and Build Global Arrays

  1 CALL cls(2)
* Determine input file name
  WRITE(*,101)
  READ (*,102) readfile
  OPEN (1,file=readfile,ERR=99)

* Echo input data?
  WRITE(*,103)
  READ (*,104) list

* Read number of domains
  READ (1,105) nd
  IF (list.EQ.1) WRITE(*,105) nd

  IF (nd .GT. mx2) THEN
    WRITE(*,108)
    CALL cls(1)
    GOTO 1
  ENDIF

* Input data for all domains
  DO 30 i = 1, nd
    READ (1,105) n(i)
    IF (list.EQ.1) WRITE(*,105) n(i)

    IF (n(i) .GT. mx3) THEN
      WRITE(*,108)
      CALL cls(1)
      GOTO 1
    ENDIF

    DO 10 j = 1, n(i)
      READ (1,106) x(j,i),y(j,i),z(j,i),kode(j,i),u(j,i)
10    IF (list.EQ.1) WRITE(*,107) i,j,x(j,i),y(j,i),z(j,i),
      .      kode(j,i),u(j,i)

    READ (1,109) t(i)
    IF (list.EQ.1) WRITE(*,109) t(i)
    IF (list.EQ.1) CALL cls(1)

    READ (1,105) m0(i)
    IF (list.EQ.1) WRITE(*,105) m0(i)

```

```

IF (m0(i) .GT. mx3) THEN
  WRITE(*,108)
  CALL cls(1)
  GOTO 1
ENDIF

DO 20 j = 1, m0(i)
  READ (1,105) m1(j,i), (mm(k,j,i), k=1,m1(j,i))
20  IF (list.EQ.1) WRITE(*,105) m1(j,i), (mm(k,j,i), k=1,m1(j,i))

* Find total number of nodes.
  IF (i .EQ. 1) nj(i) = 0
  IF (i .NE. 1) nj(i) = n(i-1) + nj(i-1)
30  IF (list.EQ.1) CALL cls(1)

* Done with data file, close it.
  CLOSE(1)
  nn = nj(nd) + n(nd)

  IF (nn .GT. mx1) THEN
    WRITE(*,108)
    CALL cls(1)
    GOTO 1
  ENDIF

  CALL cls(2)
  CALL build
  CALL pack
  IF (list.EQ.1) CALL plot
  CALL gauss
  CALL unpack
  CALL show
  GOTO 1

99 STOP
101 FORMAT (//10x,' Enter name of input data file: '$)
102 FORMAT (A10)
103 FORMAT ( /10x,' Echo of input data? (1 = yes): '$)
104 FORMAT (I1)
105 FORMAT (16I5)
106 FORMAT (3E10.2,15,2E10.2)
107 FORMAT (2I5,3E10.2,15,2E10.2)
108 FORMAT (//10x,' Problem exceeds memory capacity')
109 FORMAT (8E10.2)
END

```

The program presented here was prepared in support of research activities and is not intended for other uses. Neither the United States Government, the University of Arizona nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any portion of this program of computers that in use by such third party would not infringe privately owned rights.


```

*****
* Find influence function between two line segments *
*****
SUBROUTINE integral (x,y,z,x1,y1,z1,x2,y2,z2,x3,y3,z3,a1,a2,a3,
. b1,b2,b3,area,g,w)
IMPLICIT REAL*8 (a-h,o-s)
DIMENSION g(3,13), w(13)

pi = DACOS(-1.00)
twopi = 2. * pi
a1 = 0.
a2 = 0.
a3 = 0.

b1 = 0.
b2 = 0.
b3 = 0.

* Calculate area of integration:
area = DSQRT((y1*z2 + y2*z3 + y3*z1 - y1*z3 - y2*z1 - y3*z2)**2
. + (x1*y2 + x2*y3 + x3*y1 - x1*y3 - x2*y1 - x3*y2)**2
. + (z1*x2 + z2*x3 + z3*x1 - z1*x3 - z2*x1 - z3*x2)**2) / 2.

* Calculate volume between point and surface:
volume = ((x1-x)*(y2-y)*(z3-z) - (x1-x)*(y3-y)*(z2-z)
. + (x2-x)*(y3-y)*(z1-z) - (x2-x)*(y1-y)*(z3-z)
. + (x3-x)*(y1-y)*(z2-z) - (x3-x)*(y2-y)*(z1-z))

* Find Gaussian Integrals:
DO 10 i = 1, 13
rx = (g(1,i)*x1 + g(2,i)*x2 + g(3,i)*x3) - x
ry = (g(1,i)*y1 + g(2,i)*y2 + g(3,i)*y3) - y
rz = (g(1,i)*z1 + g(2,i)*z2 + g(3,i)*z3) - z
r = DSQRT(rx*rx + ry*ry + rz*rz)
a = w(i) * volume / r**3 / twopi
b = w(i) * area / r / pi
a1 = a1 + g(1,i) * a
a2 = a2 + g(2,i) * a
a3 = a3 + g(3,i) * a
b1 = b1 + g(1,i) * b
b2 = b2 + g(2,i) * b
10 b3 = b3 + g(3,i) * b

* Check for diagonal elements
IF (volume .EQ. 0.) THEN
IF (x1 .EQ. x .AND. y1 .EQ. y .AND. z1 .EQ. z) THEN
b1 = SFUNC(x1,y1,z1,x2,y2,z2,x3,y3,z3) / twopi
ELSEIF (x2 .EQ. x .AND. y2 .EQ. y .AND. z2 .EQ. z) THEN
b2 = SFUNC(x2,y2,z2,x3,y3,z3,x1,y1,z1) / twopi
ELSEIF (x3 .EQ. x .AND. y3 .EQ. y .AND. z3 .EQ. z) THEN
b3 = SFUNC(x3,y3,z3,x1,y1,z1,x2,y2,z2) / twopi
ENDIF
ENDIF
RETURN
END

```

```

*****
* Calculate Diagonal Coefficients *
*****
REAL*8 FUNCTION SFUNC(x1,y1,z1,x2,y2,z2,x3,y3,z3)
IMPLICIT REAL*8 (a-h,o-s)

rx = x3 - x2
ry = y3 - y2
rz = z3 - z2
r1 = DSQRT(rx*rx + ry*ry + rz*rz)

rx = x3 - x1
ry = y3 - y1
rz = z3 - z1
r2 = DSQRT(rx*rx + ry*ry + rz*rz)

rx = x2 - x1
ry = y2 - y1
rz = z2 - z1
r3 = DSQRT(rx*rx + ry*ry + rz*rz)

alpha1 = DACOS((r3*r3 + r2*r2 - r1*r1) / (2.*r2*r3))
alpha2 = DACOS((r3*r3 + r1*r1 - r2*r2) / (2.*r1*r3))
alpha3 = DACOS((r1*r1 + r2*r2 - r3*r3) / (2.*r2*r1))

SFUNC = -r2*DSIN(alpha3)*DLOG(DTAN(alpha3/2.)*DTAN(alpha2/2.))
RETURN
END

```

```

.....
*                               Create Global Matrix
*                               .....
SUBROUTINE pack
$INCLUDE j:common

```

```

* Combine R with Q to form vector of knowns
* and transfer from Q to QQ and G to GG

```

```

DO 5 i = 1, mx1
  qq(i) = 0.
  DO 5 j = 1, mx1
    gg(i,j) = 0.
  DO 50 i = 1, nd
    l = nj(i)
    DO 50 k = 1, n(i)
      m = kode(k,i)
      nj = MOD(-m,128)
      mk = -m/128
      ml = nj(mk)+mj
      IF (m .GT. 0) THEN
        DO 10 j = 1, n(i)
          qq(l+j) = qq(l+j) - g(j,k,i) * u(k,i)
10          gg(l+j,l+k) = - h(j,k,i)
        ELSEIF (m .EQ. 0) THEN
          DO 20 j = 1, n(i)
            qq(l+j) = qq(l+j) + h(j,k,i) * u(k,i)
20          gg(l+j,l+k) = g(j,k,i)
        ELSEIF (i .LT. mk) THEN
          DO 30 j = 1, n(i)
            gg(l+j,ml) = -h(j,k,i)
30          gg(l+j,l+k) = g(j,k,i)
        ELSE
          DO 40 j = 1, n(i)
            gg(l+j,ml) = -g(j,k,i)
40          gg(l+j,l+k) = -h(j,k,i)
        ENDIF
50 CONTINUE
      RETURN
    END

```

```

.....
*                               Plot matrix
*                               .....
SUBROUTINE plot
$INCLUDE j:common

```

```

CHARACTER c(200),d,s,b
DATA s,b / ' ', 'g' /
DATA k / 1 /

CALL cls(0)
DO 20 i = 1, nn
  DO 10 j = 1, nn
    IF (gg(i,j) .GT. 0.) THEN
      c(j)='+'
    ELSEIF (gg(i,j) .LT. 0.) THEN
      c(j)='- '
    ELSE
      c(j)='.'
    ENDIF
10 CONTINUE
    IF (qq(i) .NE. 0) THEN
      d = 'x'
    ELSE
      d = 'o'
    ENDIF
    WRITE (*,100) (c(j),j=1,nn),s,b,s,d
    IF (i .EQ. (mj(k) + n(k))) THEN
      CALL cls(1)
      k = k + 1
    ENDIF
20 CONTINUE
  RETURN
100 FORMAT(1x,200A1)
  END

```

```

*****
*           Gaussian elimination           *
*****
SUBROUTINE gauss
$INCLUDE j:common
DATA epsilon / 1.E-6 /

DO 20 l = 1, nn-1
  c = gg(l,l)
  IF (c .LE. epsilon) THEN
    WRITE (*,*) ' Singularity in row ', l
    STOP
  ENDIF
  qq(l) = qq(l) / c
  DO 10 k = l+1, nn
    gg(l,k) = gg(l,k) / c
  DO 20 j = l+1, nn
    c = gg(j,l)
    qq(j) = qq(j) - c * qq(l)
  DO 20 k = l+1, nn
    gg(j,k) = gg(j,k) - c * gg(l,k)

* Compute last unknown
IF (ABS(gg(nn,nn)) .LE. epsilon) THEN
  WRITE (*,*) ' *** Singularity *** In row', nn
  STOP
ELSE
  qq(nn) = qq(nn) / gg(nn,nn)
ENDIF

* Back substitute
DO 30 j = 1, nn-1
  l = nn-j
  DO 30 k = l+1, nn
    qq(l) = qq(l) - gg(l,k) * qq(k)
RETURN
END

```

```

*****
*           Reduce Global Equation       *
*****
SUBROUTINE unpack
$INCLUDE j:common

DO 10 i = 1, nd
DO 10 j = 1, n(i)

  m = kode(j,i)
  nj = MOD(-m,128)
  mk = -m/128
  k = nj(mk)+mj
  l = nj(i) + j

  IF (m .GT. 0) THEN
    q(j,i) = u(j,i)
    u(j,i) = qq(l)
  ELSEIF (m .EQ. 0) THEN
    q(j,i) = qq(l)
  ELSEIF (i .LT. mk) THEN
    q(j,i) = qq(l)
    u(j,i) = qq(k)
  ELSE
    q(j,i) = -qq(k)
    u(j,i) = qq(l)
  ENDIF

10 CONTINUE

RETURN
END

```

```

*****
*           Display Solution           *
*****
SUBROUTINE show
$INCLUDE j:common

OPEN (6,FILE='output',STATUS='new')
WRITE(6,601)

sum = 0.
DO 40 i = 1, nd

* Find flow across each face:
DO 20 j = 1, n0(i)
* Fix page break
IF (MOD(j-1,20) .EQ. 0) THEN
  IF (j .NE. 1) CALL cls(1)
  IF (j .EQ. 1) CALL cls(0)
  WRITE (*,603)
ENDIF
disch = 0.
DO 10 k = 1, m1(j,i)
  l1 = mm(k,j,i)
  IF (k .NE. m1(j,i)-1) THEN
    l2 = mm(k+1,j,i)
  ELSE
    l2 = m1(j,i)
  ENDIF
  l3 = mm(m1(j,i),j,i)
  10 disch = disch + area(k,j,i)*(q(l1,i)+q(l2,i)+q(l3,i))/3.
  20 WRITE(*,604) i, j, disch
  CALL cls(1)
* Display locations, potentials, discharges, and mass balance
DO 30 j = 1, n(i)
* Fix page break
IF (MOD(j-1,20) .EQ. 0) THEN
  IF (j .NE. 1) CALL cls(1)
  IF (j .EQ. 1) CALL cls(0)
  WRITE (*,601)
ENDIF
WRITE (*,602) i,j,x(j,i),y(j,i),z(j,i),u(j,i),q(j,i)
30 WRITE (6,602) i,j,x(j,i),y(j,i),z(j,i),u(j,i),q(j,i)
40 CALL cls(1)

RETURN
601 FORMAT (' Boundary Nodes:')
    //4x,'i',4x,'n',7x,'x',9x,'y',9x,'z',9x,'u',9x,'q'//
602 FORMAT (2i5,6f10.3)
603 FORMAT (' Boundary Faces:',8x,'i',4x,'s',8x,'discharge'//)
604 FORMAT (20x,2i5,5x,f10.3)
END

```

```

*****
*           Common Block File COMMON
*****
IMPLICIT REAL*8(A-H,O-Z)
PARAMETER (mx1=100,mx2=3,mx3=100)

COMMON nn,nd,n(3),nj(3),t(3),m0(3),mm(20,10,3),
.      m1(100,3),m2(100,3),m3(100,3),area(20,10,3),
.      x(100,3),y(100,3),z(100,3),
.      u(100,3),q(100,3),kode(100,3),g(100,100,3),h(100,100,3)

COMMON /a/ gg(mx1,mx1), qq(mx1)

```

GLOSSARY

- adsorption** - adherence of gas molecules, ions, or molecules in solution to the surface of solids.
- advection** - the process whereby solutes are transported by the bulk mass of flowing fluid.
- anisotropy** - the condition of having different properties in different directions.
- breakthrough curve** - a plot of relative concentration versus time, where relative concentration is defined as C/C_0 with C as the concentration at a point in the ground-water flow domain, and C_0 as the source concentration.
- contaminant** - an undesirable substance not normally present or an unusually high concentration of a naturally occurring substance in soil or water.
- diffusion** - process whereby ionic or molecular constituents move under the influence of their kinetic activity in the direction of their concentration gradient.
- distribution coefficient** - the quantity of solute, chemical or radionuclide sorbed by the solid per unit weight of solid divided by the quantity dissolved in the water per unit volume of water.
- Fickian diffusion** - spreading of solutes from regions of highest to regions of lower concentrations caused by the concentration gradient.
- flow path** - the course a water molecule or solute would follow in a given velocity field.
- fluid potential** - the mechanical energy per unit mass of a fluid at any given point in space and time with regard to an arbitrary state and datum.
- ground-water travel time** - the time required for ground water to travel between two locations along a flow path.
- head, capillary** - the difference in pressure head across the interface between two immiscible fluid phases jointly occupying the pores of a medium caused by interfacial tension between the two phases.
- head, gravitational** - the component of total hydraulic head related to the position of a given mass of water relative to an arbitrary datum.
- head, total** - the total head of a liquid at a given point is the sum of the gravitational, pressure, capillary, and osmotic heads.
- head, osmotic** - the difference in pressure head across a membrane between two liquids with different solute concentrations.

head, pressure - the height of a column of static water that can be supported by the static pressure at the point.

heterogeneity - a characteristic of a medium in which material properties vary from point to point.

homogeneity - a characteristic of a medium in which material properties are identical everywhere.

hydraulic conductivity, relative - equal to the specific discharge divided by the hydraulic gradient, a property of a porous medium, the liquid used during the test, and the relative saturation of the medium.

hydraulic gradient - the change in static head per unit of distance in a given direction.

hydrodynamic dispersion - the spreading at the macroscopic level of a solute front during transport resulting from mechanical dispersion and molecular diffusion.

interface - the contact zone between two materials of different chemical or physical composition.

isohead line - line along which the head is constant.

isotropy - the condition in which the property or properties of interest are the same in all directions

matric suction - the energy required to extract water from a porous medium to overcome the capillary and adsorptive forces per unit volume of porous medium.

matrix - the solid framework of a porous system.

mechanical dispersion - the process whereby solutes are mechanically mixed during advective transport caused by the velocity variations at small scales.

permeability, relative - the property of a porous medium to transmit fluids under a hydraulic gradient, a function of relative saturation.

porosity - the ration of the total volume of voids of a given medium to the total volume of the medium.

porosity, effective - the amount of interconnected pore space and fracture openings available for the transmission of fluids, expressed as the ratio of the volume of interconnected pores and openings to the volume of rock.

relative saturation - the ratio of the volume of water to the total volume of voids in a given porous medium.

retardation - the process or processes that cause the time required for a given solute to move between two locations to be greater than the ground water travel time, due to physical and chemical interactions between the solute and the geohydrologic unit through which the solute travels.

solute - the substance present in a solution in the smaller amount.

solute transport - the net flux of solute through a hydrogeologic unit controlled by the flow of subsurface water and transport mechanisms.

specific discharge - the rate of discharge of ground water per unit area of a porous medium, measured perpendicular to the direction of flow.

transmissivity - the rate at which water is transmitted through a two dimensional flow domain per unit length of flow domain perpendicular to the direction of flow divided by the hydraulic gradient.

unsaturated zone - the zone between the land surface and the regional water table. Generally, the water in this zone is under less than atmospheric pressure, although zones of positive pressure may occur locally.

LIST OF REFERENCES

- Abramowitz, M. and I.A. Stegun (eds.), 1972, Handbook of Mathematical Functions, Dover Publications, Inc., Ninth Dover Printing.
- Allen, M.B. and C.L. Murphy, 1986, "A Finite-Element Collocation Method for Variably Saturated Flow in Two Space Dimensions", Water Resour. Res., 22(11):1537-1542.
- Andersson, J. and B. Dverstorp, 1987, "Conditional Simulations of Fluid Flow in Three-Dimensional Networks of Discrete Fractures", Water Resour. Res., 23(10):1876-1886.
- Andersson, J. and A.M. Shapiro, 1983, "Stochastic Analysis of One-Dimensional Steady State Unsaturated Flow: A Comparison of Monte Carlo and Perturbation Methods", Water Resour. Res., 19(1):121-133.
- Andersson, J., A.M. Shapiro and J. Bear, 1984, "A Stochastic Model of a Fractured Rock Conditioned by Measured Information", Water Resour. Res., 20(1):79-88.
- Bachmat, Y., J. Bredehoeft, B. Andrews, D. Holtz, and S. Sebastian, 1980, Groundwater Management: The Use of Numerical Models, AGU, 127 pp.
- Baecher, G.B., N.A. Lanney and H.H. Einstein, 1977, "Statistical Description of Rock Properties and Sampling", Proc. 18th U.S. Symp. Rock Mech., 501:1-8.
- Barackman, M.L., 1986, "Diverging Flow Tracer Tests in Fractured Granite: Equipment Design and Data Collection", M.S. Thesis, University of Arizona, Tucson, 72 pp.
- Barton, C.M., 1977, "A Geotechnical Analysis of Rock Structure and Fabric in the CSA Mine, Cobor, New South Wales", CSIRO Aust. Div. Appl. Geomech. Tech. Paper, 24:30-37.
- Bear, J., 1979, Hydraulics of Groundwater, McGraw-Hill Book Co., 569 pp.
- Bear, J. and G. Dagan, 1964, "Moving Interface in Coastal Aquifers", J. Hydraulics Div. ASCE, 90:193-216.
- Bianchi, L. and D.T. Snow, 1969, "Permeability of Crystalline Rock Interpreted from Measured Orientations and Apertures of Fractures", Annals of Arid Zones, 8(2):231-235.
- Bibby, R., 1981, "Mass Transport of Solutes in Dual-Porosity Media", Water Resour. Res., 17(4):1075-1081.
- Billiaux, B., B. Feuga and S. Gentier, 1984, "Etude Theorique et en Laboratoire du Comportement d'une Fracture Rocheuse sous Contrainte Normale", Rev. Fr. Geotech..
- Bles, J.L. and B. Feuga, 1986, The Fracture of Rocks, North Oxford Academic, 131 pp.
- Brebbia, C.A., 1978, The Boundary Element Method for Engineers, Pentech Press, 189 pp.
- Brebbia, C.A., and A.J. Ferrante, 1979, Computational Methods for the Solution of Engineering Problems, Pentech Press, 2nd Edition, 354 pp.
- Brebbia, C.A. (Ed.), 1981a, Progress in Boundary Element Methods, Volume 1, Pentech Press, 315 pp.
- Brebbia, C.A. (Ed.), 1981b, Progress in Boundary Element Methods, Volume 2, Pentech Press, 217 pp.
- Brebbia, C.A. (Ed.), 1984, Boundary Element Techniques in Computer-Aided Engineering, Martinus Nijhoff Publishers, 440 pp.

- Brebbia, C.A., 1984, "Time Dependent Problems", in Brebbia, C.A. (Ed.), Boundary Element Techniques in Computer-Aided Engineering, Martinus Nijhoff Publishers, p. 261-291.
- Brebbia, C.A., C. Maksimovic, and M. Radojkovic, 1984a, Hydrossoft '84: Hydraulic Engineering Software, Proceedings of the International Conference, Portoroz, Yugoslavia, September 10-14, 1984, Elsevier.
- Brebbia, C.A., J.C.F. Telles and L.C. Wrobel, 1984b, Boundary Element Techniques, Theory and Applications in Engineering, Springer-Verlag, 464 pp.
- Brebbia, C.A. and G. Maier (Eds.), 1985, Boundary Elements VII - Volume 2, Proceedings of the 7th International Conference, Villa Olmo, Loake Como, Italy, September 1985, Springer-Verlag.
- Brebbia, C.A. and B.J. Noye (Eds.), 1985, BETECH 85, Proceedings of the 1st Boundary Element Technology Conference, South Australian Institute of Technology, Adelaide, Australia, November 1985, Springer-Verlag, 298 pp.
- Bresler, E. and G. Dagan, 1982a, "Unsaturated Flow in Spatially Variable Fields, 2, Application of Water Flow Models to Various Fields", Water Resour. Res., 19(2):421-428.
- Bresler, E. and G. Dagan, 1982b, "Unsaturated Flow in Spatially Variable Fields, 3, Solute Transport Models and Their Application to Two Fields", Water Resour. Res., 19(2):429-435.
- Bridges, M.C., 1975, "Presentation of Fracture Data for Rock Mechanics", 2nd Australian-New Zealand Conf. on Geomech., Brisbane, p. 144-148.
- Call, R.B., 1976, "Estimation of Joint Set Characteristics from Surface Mapping Data", 17th U.S. Symp. on Rock Mech., 2b2:1-9.
- Carslaw, H.S. and J.C. Jaeger, 1959, Conduction of Heat in Solids, Clarendon Press, 510 pp.
- Case, J.B. and P.C. Kelsall, 1987, "Modification of Rock Mass Permeability in the Zone Surrounding a Shaft in Fractured, Welded Tuff", SAND86-7001, 84 pp.
- Castellani, C., D. DiCastro, and L. Peliti (eds.), 1981, "Disordered Systems and Localization", Lecture Notes in Physics, Springer-Verlag, p. 10.
- Castillo, E., G.M. Karadi, and R.J. Krizeck, 1972, "Unconfined Flow Through Jointed Rock", Water Resour. Bull., 8(2):226-281.
- Chen, C.-S., 1986, "Solutions for Radionuclide Transport From an Injection Well into a Single Fracture in a Porous Formations", Water Resour. Res., 22(4):508-518.
- Cheng, A. H.-D., 1984, "Darcy's Flow With Variable Permeability: A Boundary Integral Solution", Water Resour. Res., 20(7):980-984.
- Churchill, R.V., 1974, Introduction to Complex Variables and Applications, McGraw-Hill.
- Cole, F.W., 1969, Reservoir Engineering Manual, Gulf Publishing Co., 385 pp.
- Cooley, R.L., 1983, "Some New Procedures for Numerical Solution of Variably Saturated Flow Problems", Water Resour. Res., 19(5):1271-1285.
- Cowper, G.R., 1973, "Gaussian Quadrature Formulas for Triangles", Int. J. Math. Engr., 7(3):405-408.
- Dagan, G. and E. Bresler, 1982, "Unsaturated Flow in Spatially Variable Fields, 1, Derivation of Models of Infiltration and Redistribution", Water Resour. Res., 19(2):413-420.
- Davidson, M.R., 1984, "A Green-Ampt Model of Infiltration in a Cracked Soil", Water Resour. Res., 20(11):1685-1690.

- Davidson, M.R., 1985a, "Numerical Calculation of Saturated-Unsaturated Infiltration in a Cracked Soil", *Water Resour. Res.*, 21(5):709-714.
- Davidson, M.R., 1985b, "Asymptotic Behavior of Infiltration in Soils Containing Cracks or Holes", *Water Resour. Res.*, 21(9):1345-1354.
- Davis, L.A. and S.P. Neuman, 1983, "UNSAT2 - Variably Saturated Flow Model", US Nuclear Regulatory Commission, NUREG CR-3390.
- Davis, S.N. and R.J.M. DeWiest, 1966, *Hydrogeology*, John Wiley and Sons, Inc., 463 pp.
- Deer, W.A., R.A. Howie, and J. Zussman, 1966, *An Introduction to the Rock-Forming Minerals*, Longman Group Limited, 528 pp.
- Duffy, C.J. and J. Harrison, 1987, "The Statistical Structure and Filter Characteristics of Tritium Fluctuations in Fractured Basalt", *Water Resour. Res.*, 23(5):894-902.
- El-Kadi, A.I. and W. Brutsaert, 1985, "Applicability of Effective Parameters for Unsteady Flow in Nonuniform Aquifers", *Water Resour. Res.*, 21(2):183-198.
- Elliott, G.M., E.T. Brown, P.I. Boodt, and J.A. Hudson, 1985, "Hydromechanical Behaviour of Joints in the Carnmenellis Granite, S.W. England", in Stephansson, O. (Ed.), 1985, *Fundamentals of Rock Joints*, Proceedings of the International Symposium on Fundamentals of Rock Joints, Bjorkliden, Sweden, September 15-20, 1985, Centek Publishers, Lulea, Sweden, p. 249-258.
- Elsworth, D. and R.E. Goodman, 1985, "Hydro-Mechanical Characterisation of Rock Fissures of Idealized Sawtooth or Sinusoidal Form", in Stephansson, O. (Ed.), 1985, *Fundamentals of Rock Joints*, Proceedings of the International Symposium on Fundamentals of Rock Joints, Bjorkliden, Sweden, September 15-20, 1985, Centek Publishers, Lulea, Sweden, p. 259-268.
- Elsworth, D., 1986, "A Model to Evaluate the Transient Hydraulic Response of Three-Dimensional Sparsely Fractured Rock Masses", *Water Resour. Res.*, 22(13):1809-1819.
- Elsworth, D., 1987, "A Boundary Element-Finite Element Procedure For Porous and Fractured Media Flow", *Water Resour. Res.*, 23(4):551-560.
- Endo, H.K., J.C.S. Long, C.R. Wilson, and P.A. Witherspoon, 1984, "A Model for Investigating Mechanical Transport in Fracture Networks", *Water Resour. Res.*, 20(10):1390-1400.
- Evans, D.D., 1983, "Unsaturated Flow and Transport Through Fractured Rock - Related to High-Level Waste Repositories", Progress Report, NRC-04-81-224, Third Quarter.
- Evans, D.D. and C.H. Huang, 1983, "Role of Desaturation on Transport through Fractured Rock", in J. W. Mercer, P. S. C. Rao, and I. W. Marine (eds.), *Role of the Unsaturated Zone in Radioactive and Hazardous Waste Disposal*, Ann Arbor Science, p. 165-178.
- Evans, D.D. and T.C. Rasmussen, 1988, "Unsaturated Fractured Rock Flow Model Calibration for a Tuff Site", presented at International Conference and Workshop on the Validation of Flow and Transport Models for the Unsaturated Zone, Ruidoso, NM, May 22-25, 1988.
- Farmer, I.W., J.J.K. Daemen, C.S. Desai, C.E. Glass and S.P. Neuman (Eds.), 1987, *Rock Mechanics: Proceedings of the 28th US Symposium*, University of Arizona, Tucson, June 29-July 1, A.A. Balkema, 1240 pp.
- Fatt, I., 1956a, "The Network Model of Porous Media: I. Capillary Pressure Characteristics", *Trans. AIME*, 207:144-159.

- Fatt, I., 1956b, "The Network Model of Porous Media: II. Dynamic Properties of a Single Size Tube Network", Trans. AIME, 207:160-163.
- Fatt, I., 1956c, "The Network Model of Porous Media: III. Dynamic Properties of Networks with Tube Radius Distribution", Trans. AIME, 207:164-181.
- Freeze, R.A. and J.A. Cherry, 1979, Groundwater, Prentice-Hall, Inc., 604 pp.
- Frind, E.O. and G.B. Matanga, 1985, "The Dual Formulation of Flow for Contaminant Transport Modeling, 1, Review of Theory and Accuracy Aspects", Water Resour. Res., 21(2):159-169.
- Frind, E.O., G.B. Matanga and J.A. Cherry, 1985, "The Dual Formulation of Flow for Contaminant Transport Modeling, 2, The Borden Aquifer", Water Resour. Res., 21(2):170-182.
- Fuentes, H.R., W.L. Polzer and E.P. Springer, 1987, "Effects from Influent Boundary Conditions on Tracer Migration and Spatial Variability Features in Intermediate-Scale Experiments", NUREG/CR-4901.
- Gelhar, L.W. and C.L. Axness, 1983, "Three-Dimensional Stochastic Analysis of Macrodispersion in Aquifers", Water Resour. Res., 19(1):161-180.
- Goldman, A.M., and S.A. Wolf, 1983, Percolation, Localization, and Superconductivity, Plenum Press.
- Goodman, R.E., 1976, Methods of Geological Engineering in Discontinuous Rocks, West Publishing, New York.
- Gordon, M.J., 1986, "Dependence of Effective Porosity on Fracture Continuity in Fractured Rock", Groundwater, 24(4):446-452.
- Green, R.T. and D.D. Evans, 1986, "Radionuclide Transport as Vapor Through Unsaturated, Fractured Rock", NUREG/CR-4654, U.S. Nuclear Regulatory Commission, 163 pp.
- Grisak, G.E., and J.F. Picketts, 1981, "Solute Transport Through Fractured Media, 1, The Effect of Matrix Diffusion", Water Resour. Res., 16(4):719-730.
- Hardin, E., N. Barton, D. Ling, M. Board and M. Voegelé, 1981, "A Heated Flatjack Test Series to Measure the Thermomechanical and Transport Properties of In Situ Rock Masses", Terra Tek Report TK81-88, Salt Lake, Utah.
- Hillel, D., 1971, Soil and Water, Academic Press, 288 pp.
- Hsieh, P.A. and S.P. Neuman, 1985, "Field Determination of the Three-Dimensional Hydraulic Conductivity Tensor of Anisotropic Media, 1, Theory", Water Resour. Res., 21(11):1655-1666.
- Hsieh, P.A., S.P. Neuman, G.K. Stiles and E.S. Simpson, 1985, "Field Determination of the Three-Dimensional Hydraulic Conductivity Tensor of Anisotropic Media, 2, Methodology and Application to Fractured Rocks", Water Resour. Res., 21(11):1667-1676.
- Huang, C. and D.D. Evans, 1985, "A 3-Dimensional Computer Model to Simulate Fluid Flow and Contaminant Transport Through a Rock Fracture System", NUREG/CR-4042, U.S. Nuclear Regulatory Commission, 108 pp.
- Hughes, B.D. and B.W. Ninham (eds.), 1983, "The Mathematics and Physics of Disordered Media", Lecture Notes in Mathematics, Springer-Verlag,
- Hull, L.C. and K.N. Koslow, 1986, "Streamline Routing Through Fracture Junctions", Water Resour. Res., 22(12):1731-1734.

- Huyakorn, P.S., B.H.Lester and C.R. Faust, 1983a, "Finite Element Techniques of Modeling Groundwater Flow in Fractured Aquifers", *Water Resour. Res.*, 19(4):1019-1035.
- Huyakorn, P.S., B.H.Lester and J.W. Mercer, 1983b, "An Efficient Finite Element Technique for Modeling Transport in Fractured Porous Media, 1, Single Species Transport", *Water Resour. Res.*, 19(3):841-854.
- Huyakorn, P.S., B.H.Lester and J.W. Mercer, 1983c, "An Efficient Finite Element Technique for Modeling Transport in Fractured Porous Media, 2, Nuclide Decay Chain Transport", *Water Resour. Res.*, 19(3):841-854.
- Huyakorn, P.S. and G.F. Pinder, 1983, *Computational Methods in Subsurface Flow*, Academic Press, 473 pp.
- Huyakorn, P.S., S.D. Thomas and B.M. Thompson, 1984, "Techniques for Making Finite Elements Competitive in Modeling Flow in Variably Saturated Porous Media", *Water Resour. Res.*, 20(8):1099-1115.
- Huyakorn, P.S., J.W. Mercer, and D.S. Ward, 1985, "Finite Element Matrix and Mass Balance Computational Schemes for Transport in Variably Saturated Porous Media", *Water Resour. Res.*, 21(3):346-358.
- Huyakorn, P.S., E.P. Springer, V. Guvanasen and T.D. Wadsworth, 1986, "A Three-Dimensional Finite-Element Model for Simulating Water Flow in Variably Saturated Porous Media", *Water Resour. Res.*, 22(13):1790-1808.
- Iwai, K., 1976, "Fundamental Studies of Fluid Flow Through a Single Fracture", Ph.D. Dissertation, Univ. of Calif., Berkeley.
- Jennings, A.A., 1987, "Critical Chemical Reaction Rates for Multicomponent Groundwater Contamination Models" *Water Resour. Res.*, 23(9):1775-1784.
- Jones, J.W., 1983, "Structural, Hydraulic and Geophysical Properties of Granite near Oracle, Arizona", Masters Thesis, University of Arizona, Tucson, 140 pp.
- Karadi, G.M., R.J. Krizek, and E. Castillo, 1972, "Hydrodynamic Dispersion in a Single Rock Joint", *J. Appl. Phys.*, 43(12):5013-5021.
- Karplus, W.J., 1958, *Analog Simulation*, McGraw-Hill, 427 pp.
- Kesten, H., 1982, *Percolation Theory for Mathematicians*, Birkhauser.
- Kilbury, R.K., T.C. Rasmussen, D.D. Evans, and A.W. Warrick, 1986, "Water and Air Intake of Surface-Exposed Rock Fractures in Situ", *Water Resour. Res.*, 22(10):1431-1443.
- Klavetter, E.A. and R.R. Peters, 1987, "An Evaluation of the Use of Mercury Porosimetry in Calculating Hydrologic Properties of Tuffs from Yucca Mountain, Nevada", SAND86-0286.
- Klute, A. (Ed.), 1986, *Methods of Soil Analysis, Part 1*, Am. Soc. Agron., Soil Sci. Soc. Am., 1188 pp.
- Lafe, O.E., J.A. Liggett and P.L-F. Liu, 1981, "BIEM Solutions to Combinations of Leaky, Layered, Confined, Unconfined, Nonisotropic Aquifers", *Water Resour. Res.*, 17(5):1431-1444.
- Lafe, O.E. and A. H.H-D. Cheng, 1987, "A Perturbation Boundary Element Code for Steady State Groundwater Flow in Heterogeneous Aquifers", *Water Resour. Res.*, 23(6):1079-1084.
- Law, A.M. and W.D. Kelton, 1982, *Simulation Modeling and Analysis*, McGraw-Hill Book Company, 400 pp.
- Lennon, G.P., P.L.-F. Liu and J.A. Liggett, 1979a, "Boundary Integral Equation Solution to Axisymmetric Potential Flows, 1, Basic Formulation", *Water Resour. Res.*, 15(5):1102-1106.

- Lennon, G.P., P.L.-F. Liu and J.A. Liggett, 1979b, "Boundary Integral Equation Solution to Axisymmetric Potential Flows, 2, Recharge and Well Problems in Porous Media", *Water Resour. Res.*, 15(5):1107-1115.
- Liggett, J.A. and P.L.-F. Liu, 1979a, "Unsteady Interzonal Free Surface Flow in Porous Media", *Water Resour. Res.*, 15(2):240-246.
- Liggett, J.A. and P.L.-F. Liu, 1979b, "Unsteady Flow in Confined Aquifers: A Comparison of Two Boundary Integral Methods", *Water Resour. Res.*, 15(4):861-866.
- Liggett, J.A. and P.L.-F. Liu, 1983, *The Boundary Integral Equation Method for Porous Media Flow*, George Allen and Unwin, 255 pp.
- Liu, P.L.-F., A.H.-D. Cheng, J.A. Liggett, and J.H. Lee, 1981, "Boundary Integral Equation Solutions to Moving Interface Between Two Fluids in Porous Media", *Water Resour. Res.*, 17(5):1445-1452.
- Long, J.C.S., J.S. Remer, C.R. Wilson, and P.A. Witherspoon, 1982, "Porous Media Equivalents for Networks of Discontinuous Fractures", *Water Resour. Res.*, 18(3):645-658.
- Long, J.C.S., P. Gilmour, and P.A. Witherspoon, 1985, "A Model for Steady Fluid Flow in Random Three-Dimensional Networks of Disc-Shaped Fractures", *Water Resour. Res.*, 21(8):1105-1115.
- MacMahon, B.K., 1974, "Design of Rock Slopes Against Sliding on Pre-existing Fractures", *Proc. 3rd Congr. Int. Soc. Rock Mechanics*, 2:803-808.
- Mantoglou, A. and L.W. Gelhar, 1985, "Large-Scale Models of Transient Unsaturated Flow and Contaminant Transport Using Stochastic Methods", Ralph M. Parsons Laboratory, MIT, Report No. 299, R85-2, 326 pp.
- Mantoglou, A. and L.W. Gelhar, 1987a, "Stochastic Modeling of Large-Scale Transient Unsaturated Flow Systems", *Water Resour. Res.*, 23(1):37-46.
- Mantoglou, A. and L.W. Gelhar, 1987b, "Capillary Tension Head Variance, Mean Soil Moisture Content, and Effective Specific Soil Moisture Capacity of Transient Unsaturated Flow in Stratified Soils", *Water Resour. Res.*, 23(1):47-56.
- Mantoglou, A. and L.W. Gelhar, 1987c, "Effective Hydraulic Conductivities of Transient Unsaturated Flow in Stratified Soils", *Water Resour. Res.*, 23(1):57-68.
- Marshall, T.J. and J.W. Holmes, 1979, *Soil Physics*, Cambridge University Press, 345 pp.
- McClellan, H., 1948, "Core Orientation by Graphical and Mathematical Methods", *AAPG Bull.*, 32(2):262-282.
- McKeon, T.J. and W.-S. Chu, 1987, "A Multigrid Model for Steady Flow in Partially Saturated Porous Media", *Water Resour. Res.*, 23(4):542-550.
- Meehan, D.N. and E.L. Vogel, 1982, *HP-41 Reservoir Engineering Manual*, PennWell Books, Tulsa, OK, 364 pp.
- Mercer, J.W., P.S.C. Rao, I.W. Marine (Eds.), 1983, *Role of the Unsaturated Zone in Radioactive and Hazardous Waste Disposal*, Ann Arbor Science, 339 pp.
- Messer, A.A., 1986, "Fracture Permeability Investigations Using a Heat-Pulse Flowmeter", *Masters Thesis*, Univ. of Arizona, Tucson, 72 pp.
- Moench, A.F., 1984, "Double-Porosity Models for a Fissured Groundwater Reservoir With Fracture Skin", *Water Resour. Res.*, 20(7):831-646.

- Montazer, P., 1982, "Permeability of Unsaturated Fractured Metamorphic Rocks Near an Underground Opening", Ph.D. Dissertation, Colorado School of Mines, Golden, 311 pp.
- Montazer, P. and W.E. Wilson, 1984, "Conceptual Hydrologic Model of Flow in the Unsaturated Zone, Yucca Mountain, Nevada", USGS Water Resources Investigations, Report 84-4345, p. 36-43.
- Moreno, L., I. Neretnieks and T. Eriksen, 1985, "Analysis of Some Laboratory Tracer Runs in Natural Fissures", Water Resour. Res., 21(7):951-958.
- Moreno, L. and A. Rasmuson, 1986, "Contaminant Transport Through a Fractured Porous Rock: Impact of the Inlet Boundary Condition on the Concentration Profile in the Rock Matrix", Water Resour. Res., 22(12):1728-1730.
- Muskat, M., 1946, The Flow of Homogeneous Fluid Through Porous Media, McGraw-Hill.
- Myer, L.R., L.J. Pyrak-Nolte, and N.G.W. Cook, 1986, "Experimental Determination of Fracture Void Geometry", LBL Progress Report.
- Narasimhan, T.N., 1982, "Multidimensional Numerical Simulation of Fluid Flow in Fractured Porous Media", Water Resour. Res., 18(4):1235-1247.
- Neretnieks, I., 1981, "Age Dating of Groundwater in Fissured Rock: Influence of Water Volume in Micropores", Water Resour. Res., 17(2):421-422.
- Neretnieks, I., 1983, "A Note on Fracture Flow Dispersion Mechanisms in the Ground", Water Resour. Res., 19(2):364-370.
- Neretnieks, I. and A. Rasmuson, 1984, "An Approach to Modelling Radionuclide Migration in a Medium with Strongly Varying Velocity and Block Sizes along the Flow Path", Water Resour. Res., 20(12):1823-1836.
- Neuman, S.P., C.L. Winter and C.M. Newman, 1987, "Stochastic Theory of Field-Scale Fickian Dispersion in Anisotropic Porous Media", Water Resour. Res., 23(3):453-466.
- Neuman, S.P. and P.A. Witherspoon, 1970, "Finite Element Method of Analyzing Steady Seepage with a Free Surface", Water Resour. Res., 6(3):889-897.
- Neuman, S.P., 1987, "Stochastic Continuum Representation of Fractured Rock Permeability as an Alternate to the REV and Fracture Network Concepts", 28th US Symposium on Rock Mechanics", Tucson, AZ, June 29 - July 1, 533-561.
- Neuzil, C.E. and J.V. Tracy, 1981, "Flow Through Fractures", Water Resour. Res., 17(1):191-199.
- Neuzil, C.E., 1986, "Groundwater Flow in Low-Permeability Environments", Water Resour. Res., 22(8):1163-1196.
- Nielsen, D.R., M.Th. van Genuchten and J.W. Biggar, 1986, "Water Flow and Solute Transport Processes in the Unsaturated Zone", Water Resour. Res., 22(9):89S-108S.
- Nimmo, J.R., J. Rubin and D.P. Hammermeister, 1987, "Unsaturated Flow in a Centrifugal Field: Measurement of Hydraulic Conductivity and Testing of Darcy's Law", Water Resour. Res., 23(1):124-134.
- Noorishad, J. and H. Mehran, 1982, "An Upstream Finite Element Method for Solution of Transient Transport Equation in Fractured Porous Media", Water Resour. Res., 18(3):588-596.
- Orbach, R., 1986, "Dynamics of Fractal Networks", Science, 231:814-819.

- Peters, R.R. and E.A. Klavetter, 1988, "A Continuum Model for Water Movement in an Unsaturated Fractured Rock Mass", *Water Resour. Res.*, 24(3):416-430.
- Peterson, D.W., 1961, "Dacitic Ash-Flow Sheet Near Superior and Globe, Arizona", Ph.D. Dissertation, Stanford University, Palo Alto, CA.
- Philip, J.R., 1985, "Scattering Functions and Infiltration", *Water Resour. Res.*, 21(12):1889-1894.
- Philip, J.R., 1988, "The Fluid Mechanics of Fracture and Other Junctions", *Water Resour. Res.*, 24(2):239-246.
- Pina, H.L.G., 1984, "Time Dependent Potential Problems", in Brebbia, C.A. (Ed.), *Boundary Element Techniques in Computer-Aided Engineering*, Martinus Nijhoff Publishers, p. 292-313.
- Pinder, G.F. and W.G. Gray, 1977, *Finite Element Simulation in Surface and Subsurface Hydrology*, Academic Press, 295 pp.
- Pollock, D.W., 1986, "Simulation of Fluid Flow and Energy Transport Processes Associated with High-Level Radioactive Waste Disposal in Unsaturated Alluvium", *Water Resour. Res.*, 22(5):765-775.
- Pratt, H.R., H.S. Swilfs, W.F. Brace, A.D. Black and J.W. Handin, 1977, "Elastic and Transport Properties of an In Situ Jointed Granite", *Int. J. Rock Mech. Min. Sci. and Geomech. Abstr.*, 14:35-45.
- Priest, S.D. and J.A. Hudson, 1976, "Discontinuity Spacings in Rock", *Int. J. Rock Mech. Min. Sci. and Geomech. Abstr.*, 13:135-148.
- Pruess, K., 1987, "TOUGH User's Guide", US Nuclear Regulatory Commission, NUREG/CR-4645, 78 pp.
- Rahi, K.A., 1986, "Hydraulic Conductivity Assessment for A Variably-Saturated Rock Matrix", Masters Thesis, University of Arizona, Tucson, 104 pp.
- Rasmuson, A., T.N. Narasimhan and I. Neretnieks, 1982, "Chemical Transport in a Fissured Rock: Verification of a Numerical Model", *Water Resour. Res.*, 18(5):1479-1492.
- Rasmuson, A., 1985, "Analysis of Hydrodynamic Dispersion in Discrete Fracture Networks Using the Method of Moments", *Water Resour. Res.*, 21(11):1677-1683.
- Rasmuson, A. and I. Neretnieks, 1986, "Radionuclide Transport in Fast Channels in Crystalline Rock", *Water Resour. Res.*, 22(8):1247-1256.
- Rasmussen, T.C., 1982, "Solute Transport in Saturated Fractured Media", Unpublished M.S. Thesis, University of Arizona, 61 pp.
- Rasmussen, T.C. and D.D. Evans, 1986, "Unsaturated Flow and Transport Through Fractured Rock - Related to High-Level Waste Repositories - Phase II", NUREG/CR-4655, U.S. Nuclear Regulatory Commission, 475 pp.
- Rasmussen, T.C., 1987, "Meso-Scale Estimates of Unsaturated Fractured Rock Fluid Flow Parameters", in Farmer, I.W., J.J.K. Daemen, C.S. Desai, C.E. Glass and S.P. Neuman (Eds.), 1987, *Rock Mechanics: Proceedings of the 28th US Symposium*, University of Arizona, Tucson, June 29-July 1, A.A. Balkema, p. 525-532.
- Rasmussen, T.C., 1987, "Computer Simulation of Steady Fluid Flow and Solute Transport Through Three-Dimensional Networks of Variably Saturated, Discrete Fractures", in Evans, D.D. and T.J. Nicholson (Eds.), *Flow and Transport Through Unsaturated Fractured Rock*, AGU Geophysical Monograph 39.
- Reeves, M., D.S. Ward, N.D. Johns, R.M. Cranwell, 1986, *Theory and Implementation for SWIFT II*, NUREG/CR-3328, U.S. Nuclear Regulatory Commission, 189 pp.

- Reiss, L.H., 1980, *The Reservoir Engineering Aspects of Fractured Formation*, Gulf Publishing Company, 108 pp.
- Richards, L.A., 1931, "Capillary Conduction of Liquids Through Porous Mediums", *Physics*, 1(November):318-333.
- Robertson, A., 1970, "The Interpretation of Geological Factors for Use in Slope Theory", in P.W.J. vanRensburg (ed.), *Planning Open Pit Mines*, Balkema, Cape Town, South Africa, p. 55-71.
- Rodrigues, A.E. and D. Tondeur, 1981, *Percolation Processes: Theory and Applications*, Sijthoff and Noordhoff.
- Ross, B., 1984, "A Conceptual Model of Deep Unsaturated Zones with Negligible Recharge", *Water Resour. Res.*, 20(11):1627-1629.
- Ross, B., 1986, "Dispersion in Fractal Fracture Networks", *Water Resour. Res.*, 22(5):823-828.
- Sagar, B. and A. Runchal, 1982, "Permeability of Fractured Rock: Effect of Fracture Size and Data Uncertainties", *Water Resour. Res.*, 18(2):266-274.
- Schrauf, T.W. and D.D. Evans, 1984, "Relationship Between the Gas Conductivity and Geometry of a Natural Fracture", NUREG/CR-3640, U.S. Nuclear Regulatory Commission, 131 pp.
- Schrauf, T.W. and D.D. Evans, 1986, "Laboratory Studies of Gas Flow Through a Single Natural Fracture", *Water Resour. Res.*, 22(7):1038-1050.
- Schwartz, F.W., L. Smith, and A.S. Crowe, 1983, "A Stochastic Analysis of Macroscopic Dispersion in Fractured Media", *Water Resour. Res.*, 19(5):1253-1265.
- Segerlind, L.J., 1984, *Applied Finite Element Analysis*, John Wiley and Sons, 427 pp.
- Selby, S.M. (ed.), 1968, *Standard Math Tables*, The Chemical Rubber Co., Sixteenth Edition, p. 527.
- Shanley, R.J. and M.A. Mahtab, 1974, "A Computer Program for Clustering Data Points on the Sphere", USBM, Circular 8624.
- Shapiro, A.M. and J. Andersson, 1983, "Steady State Fluid Response in Fractured Rock: A Boundary Element Solution for a Coupled, Discrete Fracture Continuum Model", *Water Resour. Res.*, 19(4):959-969.
- Sharp, J.C., 1970, "Fluid Flow Through Fissured Media", Ph.D. Dissertation, Imperial College, London, England.
- Sharp, J.C. and Y.N.T. Maini, 1972, "Fundamental Considerations on the Hydraulic Characteristics of Joints in Rock", *Int. Soc. Rock Mech., Proc. Symp. on Percolation Through Fissured Rock*, Stuttgart, No. TI-F.
- Silliman, S.E.J., 1986, "Stochastic Analysis of High-Permeability Paths in the Subsurface", Ph.D. Dissertation, University of Arizona, Tucson, 179 pp.
- Skibitzke, H.E., 1955, *Electronic Flowmeter*, U.S. Patent No. 2,728,225.
- Smith, L. and F.W. Schwartz, 1984, "An Analysis of the Influence of Fracture Geometry on Mass Transport in Fractured Media", *Water Resour. Res.*, 20(9):1241-1252.
- Smith, L., C.W. Mase, and F.W. Schwartz, 1987, "Estimation of Fracture Aperture Using Hydraulic and Tracer Tests", 28th US Symposium on Rock Mechanics, Tucson, AZ, June 29 - July 1, 453-463.
- Snow, D.T., 1965, "A Parallel Plate Model of Fracture Perveable Media", Ph.D. Dissertation, University of California, Berkeley.
- Snow, D.T., 1968, "Rock Fracture Spacings, Openings, and Porosities", *J. Soil Mech. and Found. Div. Proc. ASCE*, 94(SMI),73-91.

- Snow, D.T., 1969, "Anisotropic Permeability of Fractured Media", *Water Resour. Res.*, 5(6):1273-1289.
- Snow, D.T., 1970, "The Frequency and Apertures of Fractures in Rock", *Int. J. Rock Mech. Miner. Sci.*, 7:23-40.
- Sophocleous, M., 1979, "Analysis of Water and Heat Flow in Unsaturated-Saturated Porous Media", *Water Resour. Res.*, 15(5):1195-1206.
- Springer, E.P. and H.R. Fuentes, 1987, "Modeling Study of Solute Transport in the Unsaturated Zone", NUREG/CR-4615, Vol. 2, 241 pp.
- Stephansson, O. (Ed.), 1985, *Fundamentals of Rock Joints*, Proceedings of the International Symposium on Fundamentals of Rock Joints, Bjorkliden, Sweden, September 15-20, 1985, Centek Publishers, Lulea, Sweden, 577 pp.
- Stephens, D.B., 1979, "Analysis of Constant Head Borehole Infiltration Tests in the Vadose Zone", Ph.D. Dissertation, University of Arizona, 366 pp.
- Stephens, D.B., 1985, "A Field Method to Determine Unsaturated Hydraulic Conductivity Using Flow Nets", *Water Resour. Res.*, 21(1):45-50.
- Stetten, O., 1975, "Recent Developments in the Interpretation of Data From Joint Surveys in Rock Masses", 6th Reg. Conf. for Africa on Soil Mech. and Found., 9(2):7-26.
- Sudicky, E.A. and E.O. Frind, 1982, "Contaminant Transport in Fractured Porous Media: Analytical Solution for a System of Parallel Fractures", *Water Resour. Res.*, 18(6):1634-1642.
- Sudicky, E.A. and E.O. Frind, 1984, "Contaminant Transport in Fractured Porous Media: Analytical Solution for a Two-Member Decay Chain in a Single Fracture", *Water Resour. Res.*, 20(7):1021-1029.
- Tang, D.H., E.O. Frind, and E.A. Sudicky, 1981, "Contaminant Transport in Fracture Porous Media: Analytical Solution for a Single Fracture", *Water Resour. Res.*, 17(3):555-564.
- Tang, D.H., F.W. Schwartz and L. Smith, 1982, "Stochastic Modeling of Mass Transport in a Random Velocity Field", *Water Resour. Res.*, 18(2):231-244.
- Terzaghi, R.D., 1965, "Sources of Error in Joint Surveys", *Geotechnique*, 15:287-304.
- Tidwell, V.C., 1988, "Determination of the Equivalent Saturated Hydraulic Conductivity of Fractured Rock Located in the Vadose Zone", Unpublished M.S. Thesis, University of Arizona, 135 pp.
- Travis, B.J., 1984, "TRACER3D: A Model of Flow and Transport in Porous/Fractured Media", LANL, LA-9667-MS, 190 pp.
- Tang, D.H., E.O. Frind, and E.A. Sudicky, 1981, "Contaminant Transport in Fractured Porous Media: Analytical Solution for a Single Fracture", *Water Resour. Res.*, 17(3):555-564.
- Tsang, C.F. (Ed.), 1986, "Geotechnical Support and Topical Studies for Nuclear Waste Geologic Repositories: Annual Report, Fiscal Year 1986", LBL-23069, 224 pp.
- Tsang, Y.W., 1984, "The Effect of Tortuosity on Fluid Flow Through a Single Fracture", *Water Resour. Res.*, 20(9):1209-1215.
- Tsang, Y.W. and K. Pruess, 1987, "A Study of Thermally Induced Convection Near a High-Level Nuclear Waste Repository in Partially Saturated Fractured Tuff", *Water Resour. Res.*, 23(10):1958-1966.
- Tsang, Y.W. and C.F. Tsang, 1987, "Channel Model of Flow Through Fractured Media", *Water Resour. Res.*, 23(3):467-480.
- USNRC, 1987, "Ground-Water Protection Activities of the U.S. Nuclear Regulatory Commission", NUREG-1243.

- Valocchi, A., 1986, "Effect of Radial Flow on Deviations From Local Equilibrium During Sorbing Solute Transport Through Homogeneous Soils", *Water Resour. Res.*, 22(12):1693-1701.
- Van Genuchten, M.Th., 1978, "Calculating the Unsaturated Hydraulic Conductivity with a New Closed-Form Analytical Model", *Water Resource Program, Civil Engr. Dept., Princeton University, Research Report 78-WR-08*.
- Van Genuchten, M.Th., 1980, "A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils", *Soil Sci. Am. J.*, 44:892-898.
- Wächter, R.T. and J.R. Philip, 1985, "Steady Two- and Three-Dimensional Flows in Unsaturated Soil: The Scattering Analog", *Water Resour. Res.*, 21(12):1875-1888.
- Wang, J.S.Y. and T.N. Narasimhan, 1985, "Hydrologic Mechanisms Governing Fluid Flow in a Partially Saturated, Fractured, Porous Medium", *Water Resour. Res.*, 21(12):1861-1874.
- Wang, S.Y., C.V. Alonso, C.A. Brebbia, W.G. Gray and G.F. Pinder (Eds.), 1980, *Finite Elements in Water Resources*, Rose Printing Company, Inc., Tallahassee, FL.
- Wheatcraft, S.W. and F. Winterberg, 1985, "Steady State Flow Passing Through a Cylinder of Permeability Different From the Surrounding Medium", *Water Resour. Res.*, 21(12):1923-1929.
- Wierenga, P.J., L.W. Gelhar, C.S. Simmons, G.W. Gee, and T.J. Nicholson, 1986, "Validation of Stochastic Flow and Transport Models for Unsaturated Soils: A Comprehensive Field Study", NUREG/CR-4622, U.S. Nuclear Regulatory Commission.
- Witherspoon, P.A., C.H. Amick, J.E. Gale and K. Iwai, 1979, "Observation of a Potential Size Effect in Experimental Determination of the Hydraulic Properties of Fractures", *Water Resour. Res.*, 15(5):1142-1146.
- Witherspoon, P.A., J.S.Y. Wang, K. Iwai, and J.E. Gale, 1980, "Validity of Cubic Law for Fluid Flow in a Deformable Rock Fracture", *Water Resour. Res.*, 16(6):1016-1024.
- Wrobel, L.C. and C.A. Brebbia, 1981, "Time Dependent Potential Problems", in Brebbia, C.A. (Ed.), *Progress in Boundary Element Methods, Volume 1*, Pentech Press, p. 192-212.
- Yeh, T.-C.J., L.W. Gelhar and A.L. Gutjahr, 1985a, "Stochastic Analysis of Unsaturated Flow in Heterogeneous Soils, 1, Statistically Isotropic Media", *Water Resour. Res.*, 21(4):447-456.
- Yeh, T.-C.J., L.W. Gelhar and A.L. Gutjahr, 1985b, "Stochastic Analysis of Unsaturated Flow in Heterogeneous Soils, 2, Statistically Anisotropic Media with Variable Alpha", *Water Resour. Res.*, 21(4):457-464.
- Yeh, T.-C.J., L.W. Gelhar and A.L. Gutjahr, 1985c, "Stochastic Analysis of Unsaturated Flow in Heterogeneous Soils, 3, Observations and Applications", *Water Resour. Res.*, 21(4):465-472.
- Zanback, C., 1977, "Statistical Interpretation of Discontinuity Contour Diagrams", *Int. J. Rock Mech. Min. Sci. and Geomech. Abst.*, 14:111-120.
- Zimmer, P.W., 1963, "Orientation of Small Diameter Drill Core", *Econ. Geol.*, 58:1313-1325.

NRC FORM 335 (8-87) NRCM 1102, 3201, 3202	U.S. NUCLEAR REGULATORY COMMISSION	1. REPORT NUMBER (Assigned by PPMB: DPS, add Vol. No., if any)				
BIBLIOGRAPHIC DATA SHEET		NUREG/CR-5239				
SEE INSTRUCTIONS ON THE REVERSE		3. LEAVE BLANK				
2. TITLE AND SUBTITLE Fluid Flow and Solute Transport Modeling Through Three-Dimensional Networks of Variably Saturated Discrete Fractures		4. DATE REPORT COMPLETED <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; width: 50%;">MONTH</td> <td style="text-align: center; width: 50%;">YEAR</td> </tr> <tr> <td style="text-align: center;">November</td> <td style="text-align: center;">1988</td> </tr> </table>	MONTH	YEAR	November	1988
MONTH	YEAR					
November	1988					
5. AUTHOR(S) T.C. Rasmussen, D.D. Evans		6. DATE REPORT ISSUED <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; width: 50%;">MONTH</td> <td style="text-align: center; width: 50%;">YEAR</td> </tr> <tr> <td style="text-align: center;">January</td> <td style="text-align: center;">1989</td> </tr> </table>	MONTH	YEAR	January	1989
MONTH	YEAR					
January	1989					
7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code) Department of Hydrology and Water Resources University of Arizona Tucson, AZ 85721		8. PROJECT/TASK/WORK UNIT NUMBER				
10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code) Division of Engineering Office of Nuclear Regulatory Research U.S. Nuclear Regulatory Commission Washington, DC 20555		9. FIN OR GRANT NUMBER FIN D1662				
12. SUPPLEMENTARY NOTES		11a. TYPE OF REPORT Technical b. PERIOD COVERED (Inclusive dates) July 1986 - November 1988				
13. ABSTRACT (200 words or less) <p>The boundary integral method is used to estimate hydraulic and solute transport properties of unsaturated, fractured rock by solving the boundary value problem within intersecting fracture planes. Flow through both impermeable and permeable rock is determined using two and three dimensional formulations, respectively. Synthetic fracture networks are created to perform sensitivity studies, results of which show that: (1) The global hydraulic conductivity is linearly dependent on the product of fracture transmissivity and density for fractures of infinite length; (2) The effect of correlation between fracture length and transmissivity is to increase the global hydraulic conductivity; and (3) Simulated flow through a fractured permeable matrix compare favorably with analytic results.</p> <p>Flow through variably saturated fractures is modeled using a constant capillary head within individual fractures. A simulated free surface compares favorably with an approximate analytic solution and with laboratory results. Simulations indicate zones of water under both positive and negative pressure, as well as regions of air-filled voids. Travel times and breakthrough curves are determined by integrating the inverse velocity over a streamline, and then summing over all streamlines. Faster travel times are noted as fracture saturation decreases for the fracture network examined.</p>						
14. DOCUMENT ANALYSIS - a. KEYWORDS/DESCRIPTORS Unsaturated Zone, Fracture Flow, Boundary Integral Method, Fluid Flow Solute Transport		15. AVAILABILITY STATEMENT Unlimited				
b. IDENTIFIERS/OPEN-ENDED TERMS		16. SECURITY CLASSIFICATION (This page) Unclassified (This report) Unclassified				
		17. NUMBER OF PAGES				
		18. PRICE				

**UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555**

**OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, \$300**

**SPECIAL FOURTH-CLASS RATE
POSTAGE & FEES PAID
USNRC
PERMIT No. G-67**