## Title

Models for calculating fluid flow in randomly generated three-dimensional networks of disc-shaped fractures: THeory and design of FMG3D, DISCEL, and DIMES

## Authors

Gilmour, H.M.P.
Billaux, D.
Long, J.C.S.

Publication Date
1986-09-01

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

# EARTH SCIENCES DIVISION

# MODELS FOR CALCULATING FLUID FLOW IN RANDOMLY GENERATED THREE-DIMENSIONAL NETWORKS OF DISC-SHAPED FRACTURES: THEORY AND DESIGN OF FMG3D, DISCEL, AND DIMES

H.M.P. Gilmour, D. Billaux, and J.C.S. Long

September 1986

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

# DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-19515

# MODELS FOR CALCULATING FLUID FLOW IN RANDOMLY GENERATED THREE-DIMENSIONAL NETWORKS OF DISC-SHAPED FRACTURES

## Theory and Design of FMG3D, DISCEL, and DIMES

*H. M. Peggy Gilmour, Daniel Billaux, and Jane C. S. Long*

Earth Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

September 1986

This work was supported by the U.S. Department of Energy, Chicago Operations Office for the Crystalline Repository Project. Task No. E504-621AH under Contract No. DE-AC03-76SF00098.

# ABSTRACT

A model for steady fluid flow in three-dimensional, random networks of fractures has been developed. In this model, the fractures are disc-shaped discontinuities in an impermeable matrix. The fracture discs can be arbitrarily located within the rock volume and can have any desired distribution of aperture, radius orientation, and density. Thus, where the disc model is appropriate it is possible to calculate flow through fracture networks which are statistically similar to those that occur in nature. After the desired fractures are specified, the intersections (nodes) between these discs (elements) are identified and the boundary conditions are applied. Steady flow through the network is then calculated using a mixed analytical-numerical technique. In each fracture, analytic equations for flow into or out of each node as a function of the average head at each node are developed. The equations are based on image theory and the assumption that each node is a source (or sink) of uniform strength. A set of mass balance equations is constructed which equate flow into a node from one of its associated fractures to flow out of the node into the other associated fracture. These equations are solved for the average head at each node, and flux between fractures can then be calculated by substituting the average head values back into the analytical equations. The model has been successfully checked against analytical results for several cases of two and three intersecting fractures. The authors plan to use these techniques to measure the permeability of fracture networks.

## Table of Contents

LIST OF FIGURES vii

LIST OF TABLES xi

NOTATION xiii

ACKNOWLEDGMENTS xv

1.0 Introduction 1

2.0 Fluid Flow Model DISCEL 8

2.1 Flow in a Fracture Disc 11

2.2 Global Mass Balance and Flux 21

3.0 Fracture Generation Program FMG3D 30

3.1 Generation of Fracture System 32

3.2 Fracture System in the Flow Region 43

3.3 Fracture System to be Used in Flow Model 56

3.4 Fluid Flow Model Input 58

4.0 Plotting Program for FMG3D: DIMES 66

4.1 Program Characteristics 66

4.2 View Angles Definition and Rotation Matrix 70

4.3 Initial Calculations 70

4.4 Set up of the Segments Arrays and Reordering of the Fractures 72

4.5 Hiding 74

4.6 Drawing 77

5.0 Examples and Test Cases 78

5.1 Verification of Head and Flux Calculations 78

5.2 Testing Nodal Discretization Schemes 82

5.3 Examples of Disc Networks Plots 91

APPENDICES 91

A. Local Fracture System Coordinate Transformation and Other Computations 95

B. Method for Expressing the Potential in a Plane Due to a Node 97

C. Method for Approximating Shape Function $\Upsilon_{ij}^{k}$ 108

D. Method for Computing the Area of a Fracture Truncated at the Generation Region Boundary 111

E. Method of Determining the Position of a Circular Fracture Relative to Two Parallel Boundary Lines 116

F. Method for Computing the Area of a Fracture Truncated at the Flow Region Boundary 120

G. Method for Determining Fracture Intersections and Truncation of Intersections at Flow Region Boundaries 127

## LIST OF FIGURES

Figure 1-1. An example of a three-dimensional fracture network of disc shaped orthogonal fractures. ........ 4

Figure 1-2. Three-dimensional generation and flow regions. ........ 5

Figure 1-3. A cubic region isolated from the network shown in Figure 1-1. ........ 6

Figure 2-1. Flow conditions in a circular fracture. ........ 9

Figure 2-2. Image system for a point source in a circle. ........ 13

Figure 2-3. Construction of point, radial and nonradial nodal images. ........ 17

Figure 2-4. Boundary conditions used to produce a constant gradient. ........ 23

Figure 2-5. An example of a three-dimensional fracture system. ........ 25

Figure 3-1. Orientation of normal to fracture plane determined by angle of dip, $\phi'$ , and aximuth of dip, $\theta'$ . ........ 34

Figure 3-2. Locating circular fracture with center $(x_c, y_c, z_c)$ in a plane normal to unit vector N. The perpendicular distance from the origin to the plane is distance OA, where A is the point of intersection of the plane and the extension of vector N. Vector $N'$ passes through the point $(x_c, y_c, z_c)$ , is parallel to vector N and therefore, is normal to the fracture plane. ........ 39

Figure 3-3. Rotation to a corordinate system convenient to fracture plane ax + by + cz + d = 0. ........ 41

Figure 3-4. Flow region within spherical generation region. Rotation angles $\phi_R$ and $\theta_R$ equal zero. ........ 45

Figure 3-5. Rotation of the XY-plane through angle $\theta_R$ followed by rotation of the $ZY'$ -plane through angle $\phi_R$ . ........ 45

Figure 3-6. The fracture circle is inside all three pairs of parallel boundary lines, therefore, save the fracture, without truncation. ........ 47

Figure 3-7. Two examples of flow region boundary lines intersecting at 12 points ........ 49

Figure 3-8. Two examples of flow region boundary lines intersection at 8 points ........ 51

Figure 3-9. Two examples of determining boundary nodes. ........ 53

Figure 3-10. All fracture/boundary intersections are outside the flow region, therefore, the fracture is discarded. ........ 54

Figure 3-11. Example of a three-dimensional fracture system with five fracture and nine nodes. ........ 61

Figure 4-1. Example of subregion definition: projection of subregion boundaries, with iord = 5, ix1 = 3, ix2 = 4, iy1 = 2, iy2 = 4, iz1 = 4, iz2 = 5. ........ 68

Figure 4-2. Definition of view angles $\phi_v$ and $\theta_v$ . ........ 71

Figure 4-3. Relative position of intersecting ellipse and segment. ........ 76

Figure 5-1. Example of applications: (a) one fracture intersecting two parallel boundary planes;
(b) two equi-dimensional intersecting fractures, each intersecting one boundary;

(c) two equi-dimensional intersecting fractures, each intersecting two boundaries; (d) three equi-dimensional, orthogonal fractures of radius r which intersect symmetrically and are truncated by a boundary cube of dimensions r $\sqrt{2}$ by $\sqrt{2}$ by $r$ $\sqrt{2}$. .......... 79

Figure 5-2. Network generated by FMG3D for random case, seen from varying points of view .......... 86

Figure 5-3. Total flux results for small random case as a function of nodal discretization scheme. .......... 87

Figure 5-4. Difference in average head at internal nodes for a small random case as a function of nodal discretization scheme. .......... 88

Figure 5-5. Difference in total flux through each node for a small random case as a function of nodal discretization scheme. .......... 89

Figure 5-6. Difference in total flux thrrough each boundary plane for a small random case as a function of nodal discretization scheme. .......... 90

Figure 5-7. Regular fracture network. .......... 92

Figure 5-8. Random network, non rotated flow region. .......... 93

Figure 5-9. Random network, rotated flow region. .......... 94

Figure B-1. Construction for calculating potential due to a point node and its image. .......... 98

Figure B-2. (a) Construction for calculating potential due to a radial node. (b) Construction for calculating potential due to the image source of a radial node. .......... 100

Figure B-3. Construction for calculating potential due to a nonradial node. .......... 103

Figure B-4. Construction for calculating potential due to the image of a nonradial node. .......... 105

Figure D-1. Intersection of fracture plane and spherical generation region. .......... 112

Figure D-2. Possible relationships between fracture circle f and boundary circle S. .......... 114

Figure D-3. Computing area of fracture to be truncated at boundary of generation region. .......... 115

Figure E-1. Method of defining the position of circular fracture f relative to a pair of parallel boundary lines, $L_1$ and $L_2$, in the $z' = -d$ plane. .......... 117

Figure E-2. Examples of various possibilities of fracture position relative to parallel boundary lines. .......... 118

Figure F-1. General procedure for calculating area of truncation. .......... 121

Figure F-2. Examples of determining the areas of truncation. .......... 122

Figure F-3. Example of truncation of a fracture with multiple boundary nodes. .......... 125

Figure G-1. Circular intersection, $j'$ , of sphere j,

$$(x' - x_j')^2 + (y' - y_j')^2 + (z' - z_j')^2 = r_j^2$$

and fracture plane f, $z' = -$ d. .......... 129

Figure G-2. Intersection of circular fracture $j$ and circle $j'$ J in the plane $z' = -d$ . The intersection is the line segment $J_1J_2$. .......... 130

Figure G-3. Examples of possible relationships between the line of intersection of planes f and j and the circle $j'$ formed by sphere j and plane f. The diagram is in the plane of fracture f, $z' = -d$ . .......... 131

Figure G-4. Example of intersection of circular fracture f with line segment $J_1J_2$ which is fracture j in the $z' = -d$ plane. ........ 133

Figure G-5. Possible relationships between circle f and points $J_1$ and $J_2$ when the line through $J_1$ and $J_2$ intersects the circle at points $I_1$ and $I_2$. The points of intersection of fractures f and j, marked , will define the node, i.e., the line segment of fracture intersection. ........ 134

Figure G-6. Determining whether intersection segment $P_3P_4$ is inside the flow region when line $L_j$ is parallel to boundary lines $L_1$ and $L_2$ . ........ 136

Figure G-7. Determining whether intersection segment $P_3P_4$, or any portion of it, is inside side $L_1$ when line $L_j$ intersects boundary lines $L_1$ and $L_2$ . ........ 137

Figure G-8. Determining whether truncated endpoint of intersection segment $P_3P_4$ is on boundary node $Pn_1Pn_2$ . ........ 139

Figure G-9. Intersection of $P_3P_4$ is truncated twice and finally discarded for being outside the flow region. ........ 140

Figure G-10. Intersection $P_3P_4$ is truncated at both ends. ........ 141

## LIST OF TABLES

Table 2-1. Description of Nodes in Figure 2-5. .................... 26

Table 2-2. Nodes in Each Fracture of Example in Figure 2-5. .................... 27

Table 3-1. Coefficients of the Equations of the Six Boundary Planes. .................... 44

Table 3-2. Description of Nodes in Figure 3-12. .................... 60

Table 3-3. Nodal Arrays from Example in Figure 3-12. .................... 62

Table 3-4. Nodal Arrays from Example in Figure 3-12 when each Internal Node. is Divided into Two Segments and Each Boundary Node is Divided; into Three Segments. .................... 64

Table 5-1. Total Flux through the Fracture Network in Case 4, All Nodes are Divided into a Given Number of Equal Segments .................... 83

Table 5-2. Total Flux through the Fracture Network in Case 4, as a Function of Nodal Discretization Scheme. .................... 84

## NOTATION USED IN SECTION 2

| Symbol | Description | Units |
|---|---|---|
| a | radius of a fracture | [L] |
| $b^k$ | effective hydraulic aperture of fracture k | [L] |
| $C^k$ | datum head in fracture k | [L] |
| $f_{iN}^k$ | geometric function, $\phi_{iN}^k = Q_i f_{iN}^k$ | [L] |
| $f_{iI+}^k$ | geometric function, $\phi_{iI+}^k = Q_i\ f_{iI+}^k$ | [L] |
| $f_{iI-}^k$ | geometric function, $\phi_{iI-}^k = Q_i\ f_{iI-}^k$ | [L] |
| $F_i^k$ | geometric function, $F_i^k = f_{iN}^k + f_{iI}^{k+} + f_{iI-}^k$ | [L] |
| $\Upsilon_{ij}^k$ | matrix of shape functions for average head on node j due to the presence of node i, both in fracture k | [-] |
| $G_{ij}^k$ | matrix of shape factors for fracture k, derived from $\Gamma_{ij}^k$ such that<br>$Q_i = K^k b^k \sum_{j=1}^{n} G_{ij}^k \overline{h}_j$ | [-] |
| $\Gamma_{ij}^k$ | $\mid G_{ij}^k \mid^{-1}$ | [-] |
| h | hydraulic head | [L] |
| $\overline{h}_i$ | average head on node i | [L] |
| $H_i$ | prescribed head on boundary node i | [L] |
| I | subscript to reference the contribution of an image | [-] |
| K | permeability of a fracture $K = b^2\rho g/12\mu$ | [$L^2$/T] |
| k | superscript for fracture k | [-] |
| m | strength of a source | [$L^3$/T] |
| $m_\xi$ | strength per unit length on a node | [$L^2$/T] |
| $m_\eta$ | strength per unit length on an image source | [$L^2$/T] |
| n | total number of intersections on fracture k | [-] |
| N | subscript to reference the contribution of a node | [-] |
| i,j | subscripts to reference node numbers | [-] |

| | | |
|---|---|---|
| $r_p$ | distance from a point on a line source or image to an arbitrary point in the plane | [L] |
| Q | flux or strength | $[L^3/T]$ |
| $Q_T$ | total flux through the face of a flow region of fractured rock | $[L^3/T]$ |
| w | width of a node | [L] |
| $\epsilon$ | small distance | [L] |
| $\eta$ | variable of integration along the image of a node | [L] |
| $\xi$ | variable of integration along a node | [L] |
| $\phi$ | potential, $\phi = Kbh$ | $[L^3/T]$ |
| $\phi_{iN}^k$ | potential contribution of node i in fracture k | $[L^3/T]$ |
| $\phi_{iI+}^k$ | potential contribution of the image source of node i in fracture k | $[L^3/T]$ |
| $\phi_{iI-}^k$ | potential contribution of the image sink of node i in fracture k | $[L^3/T]$ |
| $\phi_i{}^k$ | potential due to node i and its images in fracture k | $[L^3/T]$ |
| $\Phi^k$ | total potential in fracture k | $[L^3/T]$ |
| $\overline{\Phi}_i{}^k$ | average of total potential on node i in fracture k | $[L^3/T]$ |

# ACKNOWLEDGMENTS

The authors would like to thank Professor F. Sisto for his help in identifying the use of image theory to solve this problem. We are also indebted to Iraj Javendel for his help and comments. Many thanks to Ellen Klahn for typing and retyping this manuscript. This work was supported by the U.S. Department of Energy, Chicago Operations Office for the Crystalline Repository Project. Task No. E504-621AH under Contract No. DE-AC03-76SF00098.

# 1.0 INTRODUCTION

Analysis of fluid flow through large regions of fractured rock is usually performed by assuming that the fracture network behaves like a continuum on some scale. In this way, a block of rock on that scale can be characterized by a permeability tensor and the analysis of fluid flow can proceed in the same way that flow through porous media is analyzed. Previous work with two-dimensional fracture networks (Long et al., 1982; Long, 1983; Long and Witherspoon, 1985; Long et al., 1985) has demonstrated that such networks do not necessarily behave like continua. Techniques have been developed to determine (1) when a fracture network behaves like an equivalent continuum, and (2) the components of a permeability tensor which minimize the error associated with performing an equivalent continuum analysis.

Such two-dimensional analysis has a limitation in that fractures which are not connected to the network in the plane of analysis may in fact be connected in the third dimension. Thus, two-dimensional analysis will tend to underestimate the permeability and the network will appear to behave less like a continuum than it actually does in three dimensions. To overcome this limitation, we have extended our techniques to three-dimensions and thus we developed a model to calculate steady fluid flow in random three-dimensional fracture networks.

The first step in developing a fracture flow model is to adopt a conceptual model for fracture networks which is compatible with the geometry observed in the field. Through the analysis of trace data and examination of fracture surfaces, several studies have reported that fractures are likely to be roughly elliptical or circular (Baecher and Lanney, 1978; Robertson, 1970; Pollard, 1976). In this three-dimensional model all fractures are assumed to be circular. That is, we have adopted a subset of the general elliptical case mainly because the circular shape simplifies the calculation of flow. The extension of the mathematics to elliptical fractures is possible but not trivial. We further assume that the two opposite surfaces of the fractures are parallel and that a parallel plate model for flow is appropriate. Thus, the model cannot account directly for channeling within the fracture planes. Head loss across the intersections between fractures and flow along the intersections are assumed to be negligible.

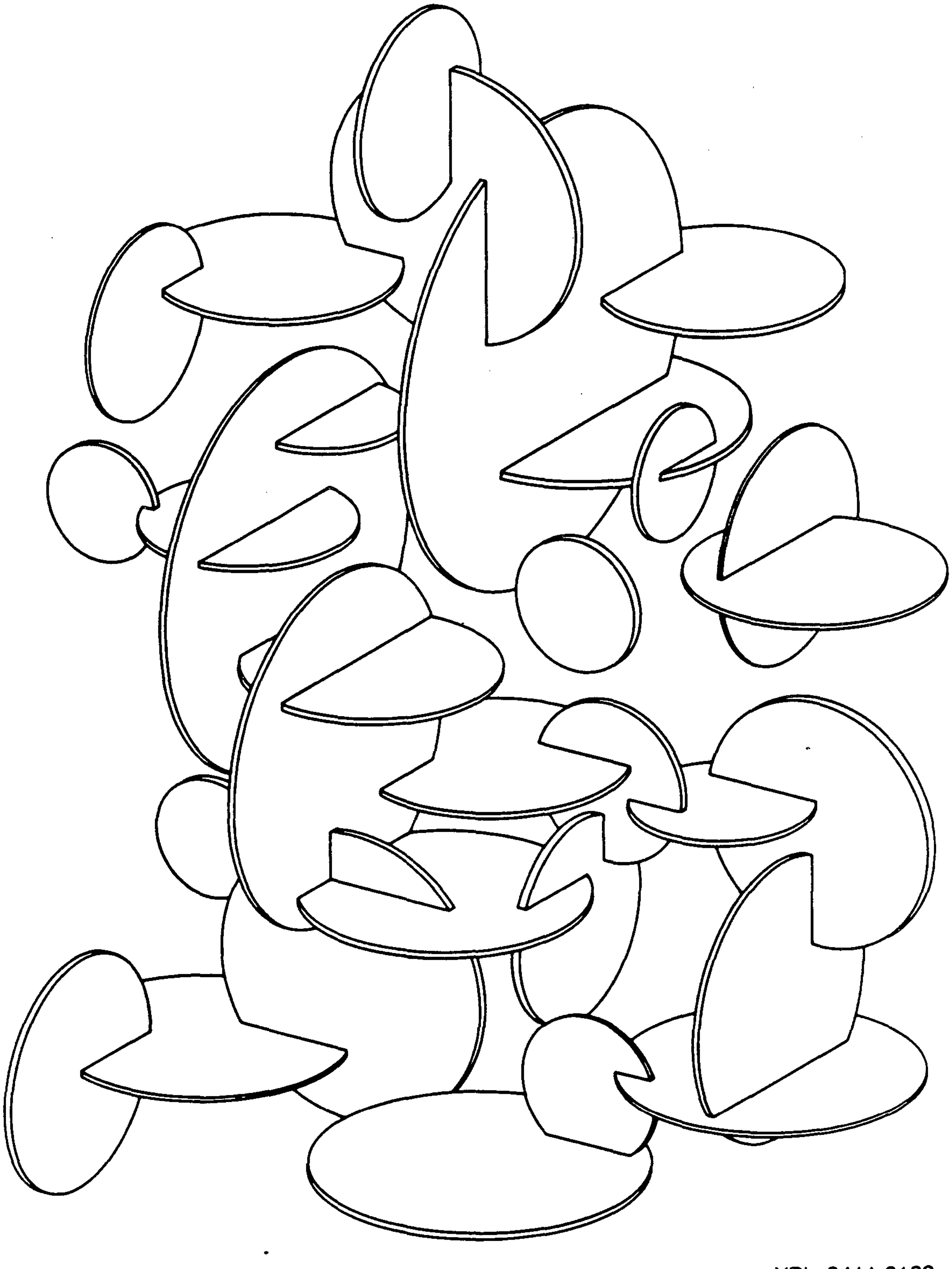
Although we are limited to circular fractures, the arrangement and size distribution of fractures in the model can be completely arbitrary. Several studies have contributed conceptual models for the arrangement of fractures in rock. Conrad and Jacquin (1973), LaPointe and Hudson (1981), Veneziano (1979), Baecher and Lanney (1978), and most recently Dershowitz (1984) have developed such conceptual models. For much of our work we use a model which is essentially the same as that developed by Baecher and Lanney in which circular fractures have lognormally distributed radii and are randomly located in space. Baecher and Lanney have supported this concept through analysis of trace data observed in outcrops, intersection data observed in boreholes and the examination of fracture surfaces (Baecher and Lanney, 1978; Robertson, 1970; Pollard, 1976; Hudson and Priest, 1979). A variety of fracture orientation distributions is possible and in general, these distributions can be determined from field data (Pincus, 1953; Mahtab and Yegulalp, 1982; Mahtab, et al., 1972). Arnold's hemispherical normal, Bingham's, Fisher's, and uniform distributions have all been used by various authors. Apertures can have lognormal distribution as was deduced by Snow (1969).

More complex statistical relationships between geometric parameters are also possible. Long (1983), Long and Witherspoon (1985) and Long et al. (l985) have pointed out that a correlation between aperture and fracture extent can account for the presence of "super-conductors" which are commonly observed in the field. Others (LaPointe and Hudson, 1981; Barton, 1984) have observed that in some cases fractures occur in zones or bands of sub-parallel features. In other words, fractures may be spatially correlated. The presence of one fracture may increase the likelihood of there being another of the same set close by. Such fracture zones or super conductors may define the boundaries of blocks of rock which in turn may contain smaller features (Conrad and Jacquin, 1973; Barton, 1984). Our flow model can accommodate any of the above features explicitly or stochastically. The only restriction at this time is that the fractures must be circular.

Our three-dimensional model, therefore, consists of randomly located fracture discs with distributed orientations, apertures, and radii that intersect to form the flow system. An example of such a random fracture system is shown in Figure 1-1. In this case, three orthogonal sets of fractures have been drawn to simplify the illustration; in general, any orientation distribution is possible. The fracture system is numerically generated in a spherical generation region. The region in which flow is calculated, the flow region, is a rectangular parallelepiped that lies entirely within the generation region, as shown in Figure 1-2. Figure 1-3 shows a cubic region isolated from the fracture system shown in Figure 1-1. A two-dimensional analysis of flow through this network would only include the traces of the fractures in a specified plane. Such a set of traces can be observed on any of the faces of the cube. Clearly, these traces do not interconnect to the same degree that the discs do. A two-dimensional analysis would greatly underestimate the permeability of this particular network. Much more realistic results can be obtained using a three-dimensional flow model.

In three dimensions the form of an intersection is a line segment, whereas in the two-dimensional model intersections are points. Thus, line segments will become the "nodes" of the three-dimensional model. Steady flow takes place in any given disc-shaped fracture from one node to another.

. The flow model is designed to measure permeability and observe hydraulic behavior of the fracture system. As in the two-dimensional model (Long et al., 1982; Long, 1983), permeability can be measured in the direction of gradient. This direction can be changed



XBL 8411-6163

Figure 1-1. An example of a three-dimensional fracture network of disc shaped orthogonal fractures

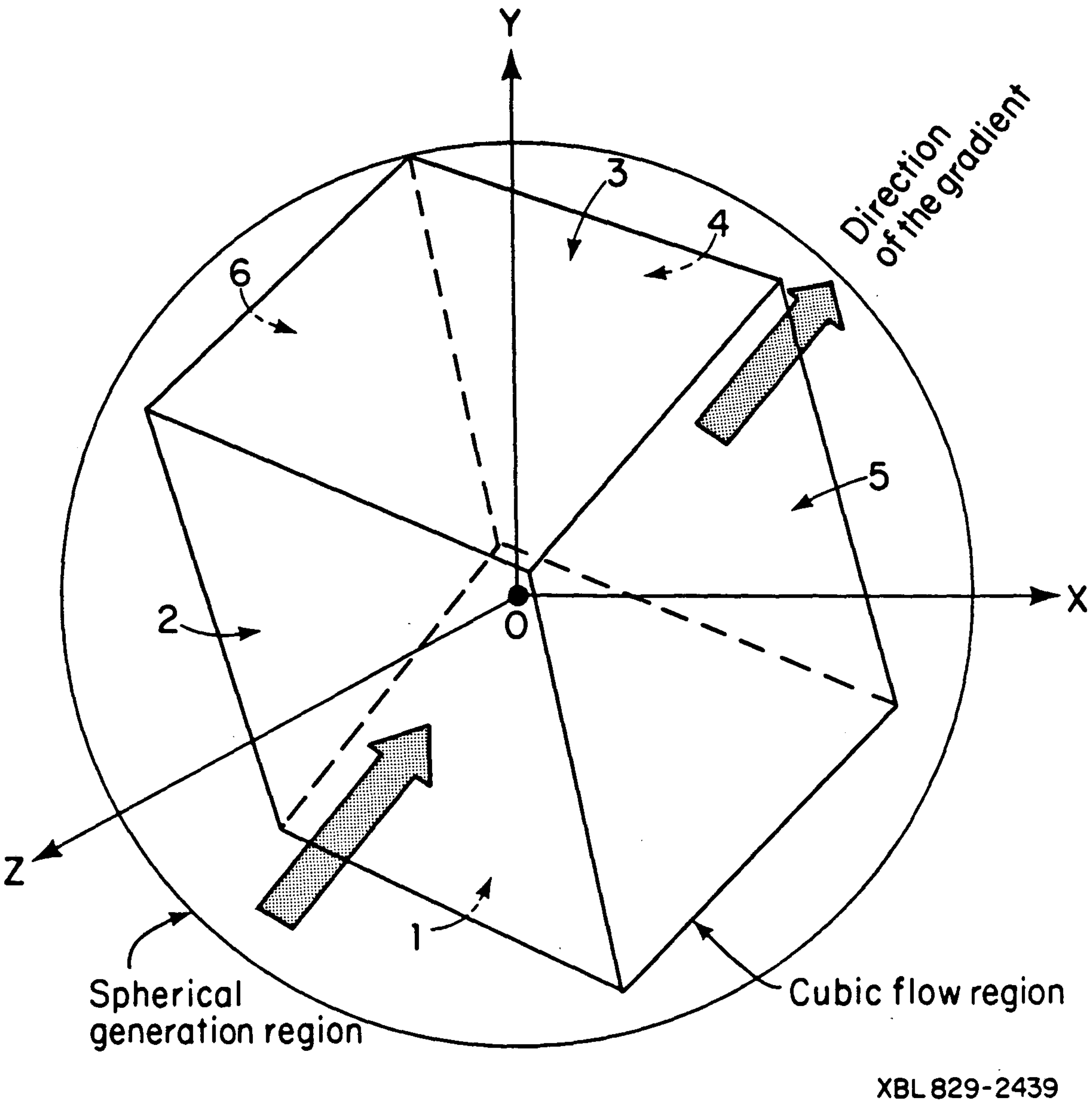
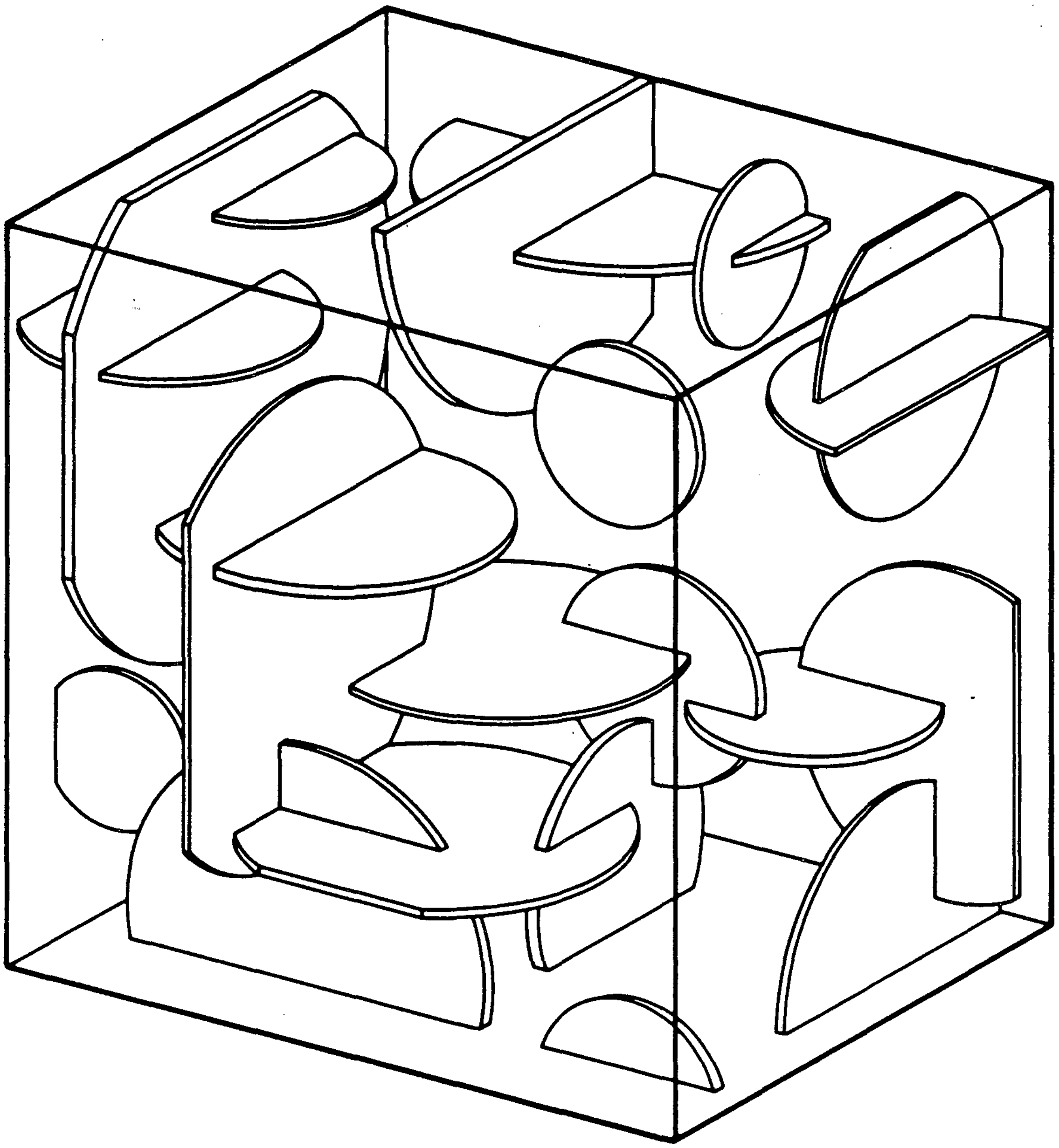


Figure 1-2. Three-dimensional generation and flow regions



XBL 8411-6162

Figure 1-3. A cubic region isolated from the network shown in Figure 1-1.

by mathematically defining flow regions that are oriented in different directions but centered at the same point. In the two-dimensional model the flow region was rotated in equal intervals from $0^{o}$ to $180^{o}$ to obtain the permeability ellipse. In this model, rotations should be made over an entire hemisphere in order to define the permeability ellipsoid.

This report provides documentation for the three-dimensional fluid flow model, DISCEL, the auxiliary fracture generation program, FMG3D, and its plotting program, DIMES. Section 2.0 presents the flow model theory including flow in individual fracture discs, formation and solution of global mass balance equations, designation of flow region boundary conditions, and computation of flux on the six flow region boundary planes. Section 3.0 describes the generation of a primary fracture system within a spherical generation region, the statistical basis of fracture generation, designation of a flow region within the generation region, selection of a subset of fractures lying within the flow region, determination of a conducting fracture network within the flow region, and preparation of data for analysis by DISCEL. Details of some of the calculations pertaining to Sections 2.0 and 3.0 are given in appendices. Section 4.0 describes the plotting program. Section 5.0 presents some test cases and examples used to verify the codes.

# 2.0 FLUID FLOW MODEL DISCEL

We have not been able to find a general analytical solution for flow in large, random, three-dimensional fracture systems. On the other hand, in three dimensions, a purely numerical solution scheme such as used in our two-dimensional model would require discretization within each fracture disc. While theoretically possible, this approach has two practical problems. The first is that the total number of unknowns would be equal to the number of fractures times the average number of elements in each fracture. Thus there would be severe limitations on the size of problem that could be analyzed. Secondly, the intersections between fractures are randomly located in the fracture disc. Thus the development of a numerical mesh generator to discretize every fracture would be difficult. The solution technique proposed here is a mixed numerical-analytical method similar to a boundary element technique. The flow relationships in each fracture plane are calculated analytically. The flux through the system is then calculated using a numerical solution based on (1) analytical solutions in each fracture and (2) preserving mass balance in the system.

The analytical solution for flow in each fracture plane is based on the assumption that each intersection acts as a source or sink with constant strength per unit line length and the fracture itself acts as a permeable disc with an impermeable boundary, (Figure 2-1). Solution of the Laplace equation for this case consists of an expression for the potential distribution in the plane, in terms of the total flux into or out of each of the intersections in any given fracture. From these equations we calculate the average head on each node in terms of the total fluxes. This results in a set of equations which can be inverted to produce an expression for the total flux into or out of each intersection in

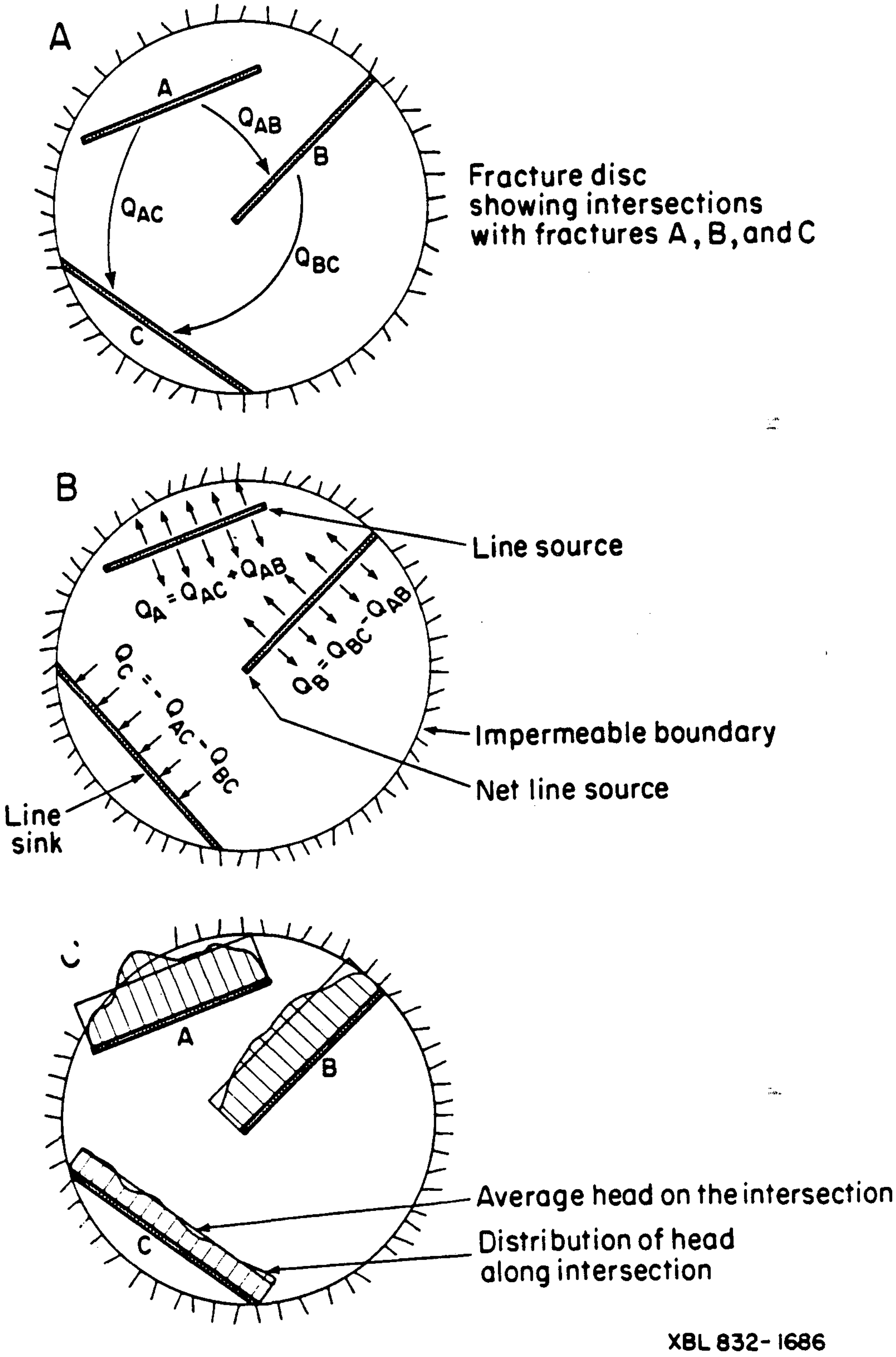


Figure 2-1. Flow conditions in a circular fracture.

terms of the average head at each of the nodes.

These equations are developed in each fracture disc, and therefore two different expressions for the total flux into or out of each intersection will have been found for each node, i.e., one expression is found for each of the two fractures that form the intersection. A global mass balance equation can then be written by equating the flux into a node from one of its associated fractures to the flux out of the intersection into the other fracture. There will be one equation for each node. Solution of these equations gives the average head at each intersection. Knowing the average heads, the flux through each intersection can then be calculated using the solution to the Laplace equation which was developed for each fracture.

The technique described here does not force the head distribution calculated on the intersection to be identical in both fractures, and we have assumed that the intersections are of constant strength per unit line length. In reality, the nodes will not have constant strength per unit line length and the head distribution along the node must be the same as measured in either of the two intersecting fractures. We assume that the real total flux into or out of an intersection is approximately equal to the flux predicted with a source of constant strength per unit length. Furthermore, we require only the average head along the fracture intersection to be the same in each of the fractures which form the intersection. This means that solution accuracy should be improved by dividing each node into segments and treating each segment independently. In this way we can allow piecewise matching of average heads in the calculation of total flux at the expense of problem size. This concept is examined in the example flow problems at the end of this paper. An alternative to this approach would be to allow head to vary along the intersection in a piece-wise linear fashion and solve for head at the endpoints of the linear segments. This modification may be made in the future.

This program can be used to measure directional permeability. For this use, flow region boundary conditions are set up to ensure a constant average gradient in the medium. Using the cubic flow region shown in Figure 1-2 as an example, the inflow face, side 1, is assigned a head of $\Delta \phi$ and the outflow face, side 3, is assigned a head of zero. Sides 2, 4, 5, and 6 have fixed linearly distributed heads. Plots of the head distribution over these boundary planes would look like wedges with a value of $\Delta\phi$ along the edges intersecting side 1 and decreasing to a value of zero along the edges intersecting side 3. The head at any point on these four sides can, therefore, be found by linear interpolation. Intersections between fractures and boundary planes, called boundary nodes, are assigned fixed average head values accordingly. That is, boundary nodes on side 1 are assigned a fixed head of $\Delta\phi$, those on side 3 are given a fixed head of zero, and, for those lying on sides 2, 4, 5, or 6, the head at each endpoint is computed from its position in the plane, and the two values are averaged. After solving the global mass balance equations for average heads on internal nodes, i.e., intersections between fractures, flux through each node is calculated. For each of the six flow region faces, fluxes through boundary nodes are summed. Thus, we have the gradient, $\Delta\phi$, and the total flux through the flow region in the direction of the gradient.

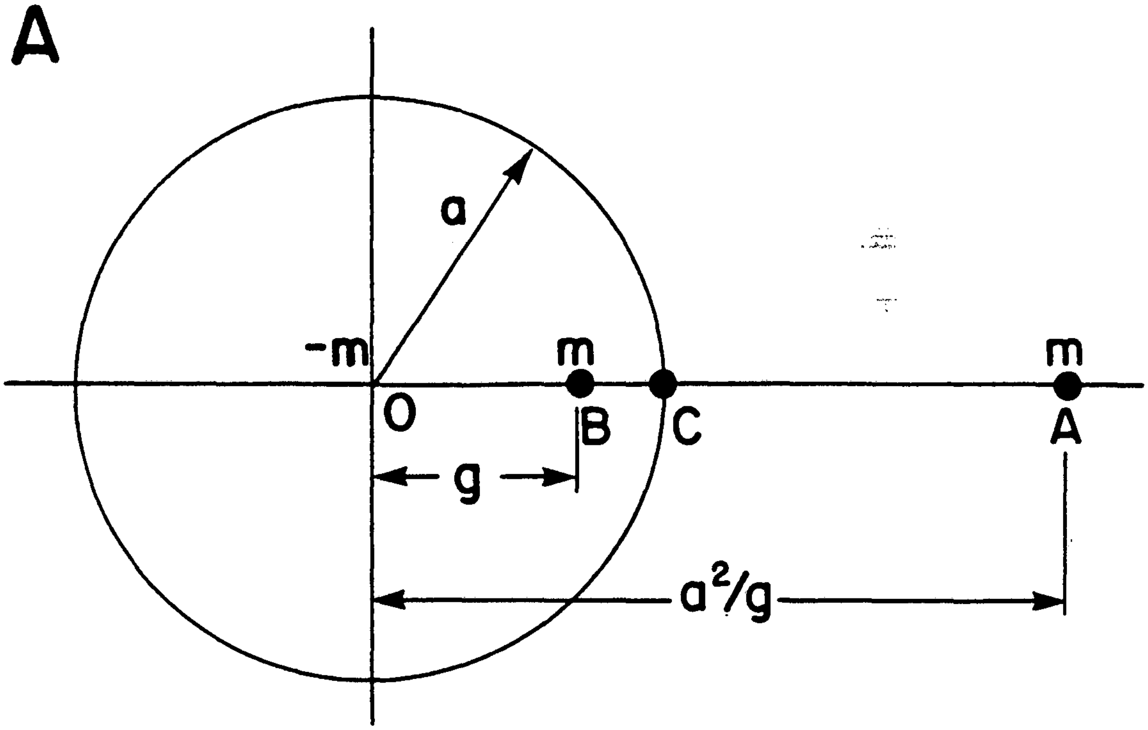
This section describes how program DISCEL solves for flow in each fracture, global mass balance in the fracture network, and flux through the flow region. Appendicies A, B, and C supply more detailed descriptions of some formulations and calculations. The user's guide for the program, as well as a listing of the code are provided in a separate report.

## 2.1. FLOW IN A FRACTURE DISC

Flow can only take place in a fracture if it is intersected by at least two other fractures. If a fracture is intersected by only one other fracture then it is a dead end which we assume does not conduct fluid in steady flow. If a fracture is intersected by two other fractures, then, within the fracture, one of the intersections is (on the average) a line source and the other is (on the average) a line sink. We say "on the average" because flow within a given fracture may enter an intersection on one side and leave the intersection on the other. The net flow into or out of the intersection determines whether it is a source or sink for potential calculations. If the fracture is intersected by more than two other fractures then at least one of the intersections acts as a source and at least one acts as a sink.

The solution of the problem of flow in a fracture disc uses image sources and sinks to account for the impermeable boundaries. In fact, the simplicity of the image system for a source or sink within a circle is a major advantage of assuming fractures are circular. The head distribution within a disc with impermeable boundaries containing an arbitrary number of line sources and sinks is based on the solution for a point source within a circular flow region. Consider a disc which contains a point source of strength m at B as shown in Figure 2-2. For steady state conditions Milne-Thomson (1968, p. 222) gives an image system which accounts for the impermeable boundary at $r = a$. If the source is located within the circle at B where $r = g$, then there is an image source of strength m on the same ray at A, where $r = a^2/g$, and an image sink of strength -m at $r = 0$. The head at any point in the fracture plane due to a point source within the fracture disc can be found by accumulating the head contributions of the source and the two images.

At least two fracture intersections in a fracture disc are necessary to have steady flow in the disc, and at least one of the intersections must act as a source and at least one must act as a sink. Furthermore, in order that the total flow into the fracture equals the total flow out of the fracture, the total strength of all sources must be equal and opposite to the total strength of all sinks. The total strength of all required images at $r = 0$ will then always be zero. Therefore, all the images at the center of the fracture could be discarded since they have no net contribution. However, getting rid of these terms destroys the symmetry of the resolution scheme: the end result is the same, but in the course of the computation, the influence of a unit head at intersection i on the flux at intersection j is not any more equal to the influence of a unit head at intersection j on the flux at intersection i. This symmetry property holds only if the "center of fracture"



XBL 832-1692

Figure 2-2. Image system for a point source in a circle.

images are taken into account. The amount of computation required for handling these images is very small, and having a symmetrical problem speeds up the matrix inversions and lowers the storage requirements of the inverting routine. Thus we chose to keep the center images.

Note that we use the term "source" for any real node, either source or sink. A net source will have a positive strength whereas a net sink will have a negative strength. The term "image source" will be used only for the image located outside the disc-shaped fracture, with the same strength as the real node, either positive or negative. The term "image sink" will be reserved for the image in the center of the fracture, with a strength equal to the opposite of the strength of the real node.

### 2.1.1 Solution of the Laplace Equation in the Fracture

Given a point source of strength Q = m in an infinite plane, the fundamental solution of the Laplace equation for the potential at any point in the plane due to the point source is given by

$$\phi = \mathrm{Kbh} = \frac{-\mathrm{Q}}{2\pi} \ln \mathrm{r} \qquad (2\text{-}1)$$

where r is the distance from the source to the point, K is the permeability of the fracture, b is the fracture aperture, and h is the hydraulic head.

Now, we assume such point sources are distributed along an arbitrary line segment in a disc, i.e. a node, such that the strength per unit length is constant. Milne-Thomson (1968) has shown that the potential at any point in the plane due to a line source of length $l$ from $\xi_1$ to $\xi_2$ is given by

$$\phi_{\mathrm{N}} = \mathrm{Kbh} = \frac{-\mathrm{Q}}{2\pi \mathrm{l}} \int_{\xi_1}^{\xi_2} \ln \mathrm{r_p}(\xi)\, d\xi, \qquad (2\text{-}2)$$

where Q is the total strength of the line source, $\mathrm{Q/l} = \mathrm{m}_\xi$ is the constant strength per unit line length, and $\mathrm{r_p}(\xi)$ is the distance from a point $\xi$ on the line source to any point in the plane. For an infinitesimal piece of the node, $d\xi$ , the total strength is

$$m_\xi \, d\xi = \frac{Q}{l} \, d\xi \tag{2-3}$$

A similar integration along the image source will give us its potential contribution. The total strength of sources distributed on the image source is also Q, however, the strength per unit line length on this image is not constant. For an infinitesimal piece of the image, $d\eta$ , corresponding to $d\xi$ on the node, the total strength is also $m_\xi \, d\xi$ , however, on the image this strength is distributed over the length $d\eta$ . Thus, the strength per unit length along the image, $m_\eta$ , is

$$m_\eta = m_\xi \, \frac{d\xi}{d\eta} = \frac{Q}{l} \, \frac{d\xi}{d\eta}. \tag{2-4}$$

Now, we integrate along the image where the potential due to sources on the image distributed according to $m_\eta$ from $\eta_1$ to $\eta_2$ is given by

$$\phi_{I+} = Kbh = \frac{-Q}{2\pi l} \int_{\eta_1}^{\eta_2} \frac{d\xi}{d\eta} \ln r_p(\eta) \, d\eta, \tag{2-5}$$

where $r_p(\eta)$ is the distance from a point $\eta$ on the image to any point in the plane. In order to avoid evaluating $d\xi/d\eta$ , we will find the distance $r_p(\eta)$ in terms of $\xi$ and integrate along the node from $\xi_1$ to $\xi_2$ , rather than integrating along the image. Then we can write

$$\phi_{I+} = Kbh = \frac{-Q}{2\pi l} \int_{\xi_1}^{\xi_2} \ln r_p(\xi) \, d\xi, \tag{2-6}$$

where $r_p(\eta)$ is written in terms of $\xi$ .

The potential contribution of the image sink of strength -Q at the fracture center is of a very simple shape

$$\phi_{I-} = Kbh = \frac{Q}{2\pi} \ln r \tag{2-7}$$

where r is the distance from the center of the fracture to any point.

In the following the notation for potential contributions will be as explained below.

In $\phi_{iN}^{k}$, $\phi_{iI+}^{k}$ and $\phi_{iI-}^{k}$,

$\phi$ stands for potential contribution k means the potential is computed in fracture k

i means the potential is due to intersection i

N stands for the contribution of the actual node

I+ stands for the contribution of the image source

I- stands for the contibution of the image sink.

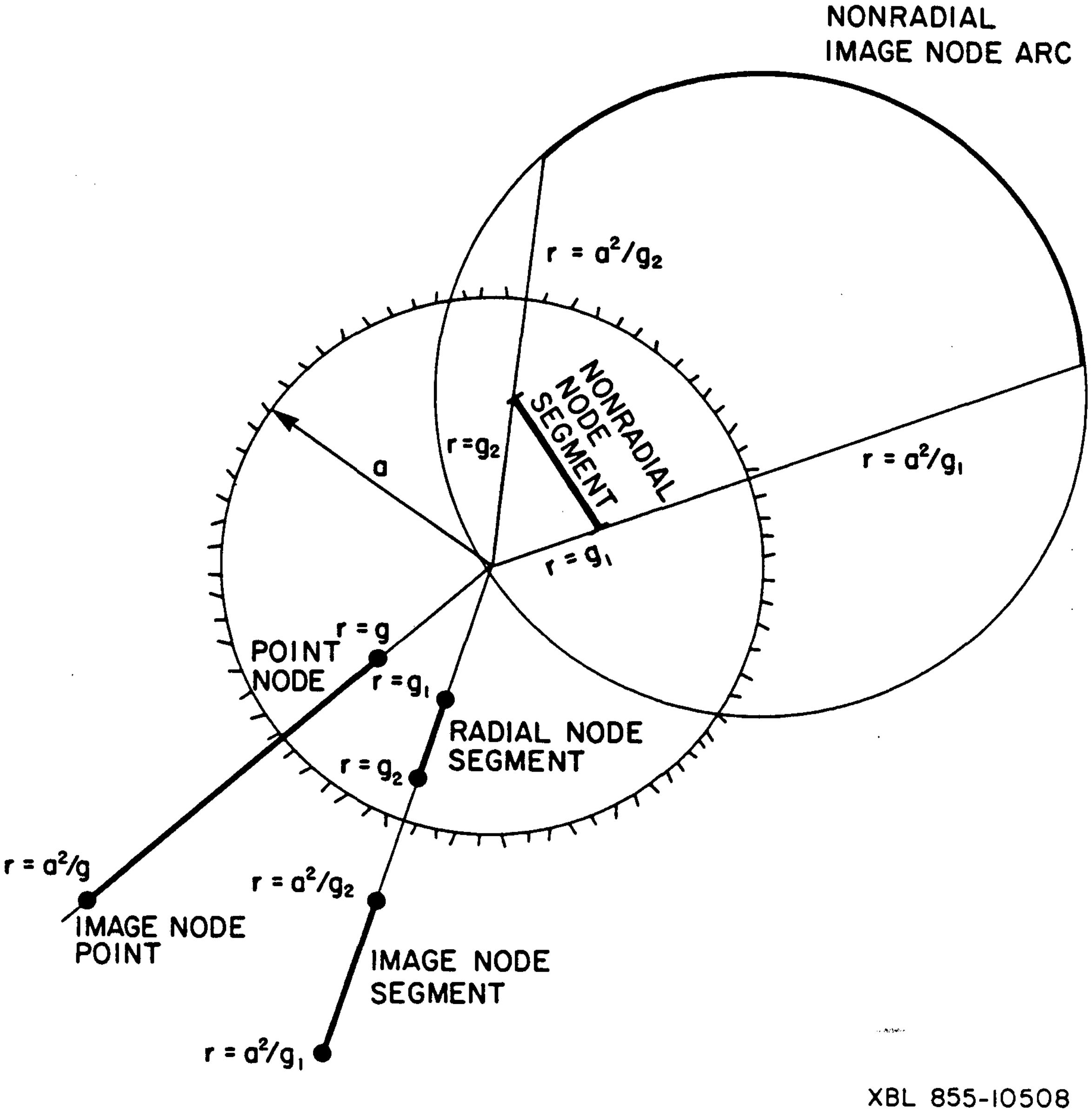
The potential due to the presence of intersection i in fracture k is ${\phi_i}^k = \phi_{iN}^k + \phi_{iI+}^k + \phi_{iI-}^k$. The total potential due to the n intersections in fracture k will be given by

$$\Phi^k = \sum_{i=1}^{n} {\phi_i}^k + C^k K^k b^k, \tag{2-8}$$

where $C^k$ is the datum potential, $K^k$ is the permeability, and $b^k$ is the aperture of fracture k. The average value of $\Phi^k$ on intersection i will be called ${\Phi_i}^k$ .

Evaluation of the potential distribution in a fracture disc due to both the intersection itself and the image source will depend on the geometry of the intersection. As shown in Figure 2-3, a point source will have an image point, a radial line source will have a radial segment image, and a nonradial line source will have an arc-shaped image. Using the coordinates of the nodal endpoints and the radius of the fracture disc, provided by the fracture generation model FMG3D, we can determine the locus of the image and an expression for potential due to the node and its images for each of the three node types. Since expressions for potential due to intersections in a particular fracture are developed in the plane of that fracture, the problem can be simplified by working in two dimensions rather than three. Appendix A describes the coordinate transformation used as well as other computations performed for each fracture before expressions for potential can be evaluated. Details of the methods used to determine the locus of the image and the potential equations for an arbitrary point node, radial node, and nonradial node are given in Appendix B.

When the appropriate integrals have been evaluated, for each node in the fracture we have an expression for potential of the form:



XBL 855-10508

Figure 2-3. Construction of point, radial, and nonradial nodal image sources

$$\phi_i{}^k = \phi_{iN}^k + \phi_{iI+}^k + \phi_{iI-}^k \tag{2-9}$$

which can be expressed in terms of the total strength of node i and shape functions $f_{iN}^k$, $f_{iI+}^k$, and $f_{iI-}^k$:

$$\phi_i{}^k = Q_i\, f_{iN}^k + Q_i\, f_{iI+}^k + Q_i\, f_{iI-}^k = Q_i\, F_i{}^k \tag{2-10}$$

where $F_i{}^k = f_{iN}^k + f_{iI+}^k + f_{iI-}^k$ .

### 2.1.2 Total Potential Field in the Fracture

The total potential in fracture k due to its n intersections is given by

$$\Phi^k = \sum_{i=1}^{n} \phi_i{}^k + C^kK^kb^k = \sum_{i=1}^{n} (\phi_{iN}^k + \phi_{iI+}^k + \phi_{iI-}^k) + C^kK^kb^k. \tag{2-11}$$

Equation 2-11 can be rewritten

$$\Phi^k = \sum_{i=1}^{n} Q_iF_i{}^k + C^kK^kb^k. \tag{2-12}$$

where the $F_i{}^k$ represent shape functions for the total head distribution over the entire fracture k due to the presence of node i, (Equation 2-10). The average potential at each node j in fracture k, (where j also equals 1 to n), due to the presence of nodes i = 1,n is

$$\overline{\Phi}_j{}^k = \frac{1}{l_j} \int_{l_j} \Phi^k\, dl_j = \frac{1}{l_j} \int_{l_j} \left[\sum_{i=1}^{n} Q_i\, F_i{}^k + C^kK^kb^k \right] dl_j,$$

$$\overline{\Phi}_j{}^k = \sum_{i=1}^{n} \frac{Q_i}{l_j} \int_{l_j} F_i{}^k\, dl_j + C^kK^kb^k = \sum_{i=1}^{n} Q_i\Upsilon_{ij}^k + C^kK^kb^k, \tag{2-13}$$

where $\Upsilon_{ij}^k$ is the shape function for the average head on node j in fracture k due to node i, also in fracture k. That is,

$$\Upsilon_{ij}^k = \frac{1}{l_j} \int_{l_j} F_i{}^k\, dl_j = \frac{1}{l_j} \left( \int_{l_j} f_{iN}^k\, dl_j + \int_{l_j} f_{iI+}^k\, dl_j + \int_{l_j} f_{iI-}^k\, dl_j \right). \tag{2-14}$$

The value of $\Upsilon_{ij}^k$ may be difficult to obtain analytically, but it can be easily approximated by numerical integration, that is, by evaluating $F_i{}^k$ at a finite number of points on node j and averaging the results. The method is given in Appendix C.

### 2.1.3 Solution of the Flow Equations in the Fracture

Equation 2-13 is the analytical solution for the average potential at intersection j in fracture k in terms of flux through the n intersections of fracture k. Thus, we have the potential distribution in each fracture as a function of the flux into or out of each node.

The average potential can also be written in terms of average head and fracture transmissivity:

$$\overline{\Phi}_j{}^k = K^k b^k \overline{h}_j. \tag{2-15}$$

Therefore, we can write an expression for average head at each node j in terms of flux through the n nodes in fracture k:

$$K^k b^k \overline{h}_j = \sum_{i=1}^{n} Q_i \Upsilon_{ij}^k + C^k K^k b^k \tag{2-16}$$

or

$$\overline{h}_j = \sum_{i=1}^{n} \frac{Q_i}{K^k b^k} \Upsilon_{ij}^k + C^k. \tag{2-17}$$

We solve for $\overline{h}_j$ rather than $\overline{\Phi}_j{}^k$ because the transmissivity, Kb, may not be the same in the two fractures forming the intersection. If the transmissivity is different, the average potential, $\overline{\Phi}_j{}^k$ , will have a different value in each of the two fractures. The average head at the node, $\overline{h}_j$ , however, will be the same in both fractures.

Next, we need to solve this system of equations for flux through node i in terms of average head on node j. In order to satisfy mass balance locally, we must ensure that the sum of the fluxes of the n nodes in fracture k is zero, i.e.

$$\sum_{i=1}^{n} Q_i = 0. \tag{2-18}$$

Thus we have n + 1 linear equations: the n equations 2-17 for j = 1 to n, and equation 2-18, matching the n+1 unknowns which are the n values of $Q_i$ and the datum potential $C_k$. We can form the (n+1) by (n+1) matrix of the system, $\Gamma^k$ and the matrix equation becomes:

$$\begin{bmatrix} \Upsilon^k_{11}\Upsilon^k_{12} & \Upsilon^k_{13} & \cdot & \cdot & \cdot & \Upsilon^k_{1n} & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \Upsilon^k_{n1} & \cdot & \cdot & \cdot & \cdot & \Upsilon^k_{nn} & 1 \\ 1 & \cdot & \cdot & \cdot & \cdot & 1 & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_n \\ K^k\, b^k\, C_k \end{bmatrix} = \begin{bmatrix} \overline{h}_1 \\ \overline{h}_2 \\ \overline{h}_3 \\ \vdots \\ \overline{h}_n \\ 0 \end{bmatrix} K^k\, b^k$$

or

$$[\Gamma^k]\ [Q] = [\overline{h}]\ K^k\, b^k \tag{2-19}$$

Inverting the symmetric $\Gamma^k$ matrix, we get a symmetric matrix $G^k$ such that:

$$[Q] = K^k\, b^k\ [G^k]\ [\overline{h}] \tag{2-20}$$

Equation 2-20 is the solution for flux through intersection i in fracture k in terms of average head at the n nodes in fracture k.

Subroutine FORMG derives the n+1 by n+1 matrix from the $\Upsilon^k$ matrix. Inversion of the $\Gamma^k$ matrix is performed by the International Mathematical and Statistical Library (IMSL) routine LINV2F. If the fracture intersects one or more flow region boundary planes, i.e. contains boundary nodes, the n first rows and columns of the G matrix are stored in a linear arrray [Gstore (maxn * maxfrc)] by subroutine STOREG. This array will be used to compute flux after the global mass balance equations are solved for average head values at each node.

### 2.1.4 Return to Global Node Numbering System

The arrays and matrix in Equation 2-20 have subscripts which take on the values of the node numbers in the local numbering system of fracture k (Appendix A). When filling the global mass balance equations, the original node numbers of the global numbering system are used as subscripts for Q and h. Therefore, i and j now assume the global node numbers corresponding to local node numbers 1 through n. The global node numbers of the nodes on fracture k are not necessarily sequential. Thus, for instance, if nodes 6, 7, and 3 lie on fracture 4 and correspond to local nodes 1, 2, and 3, the set of equations formed relative to the global numbering system is

$$\begin{aligned} Q_6 &= K^4b^4\,(G^4_{11}\,\bar{h}_6 + G^4_{12}\,\bar{h}_7 + G^4_{13}\,\bar{h}_3) \\ Q_7 &= K^4b^4\,(G^4_{21}\,\bar{h}_6 + G^4_{22}\,\bar{h}_7 + G^4_{23}\,\bar{h}_3) \\ Q_3 &= K^4b^4\,(G^4_{31}\,\bar{h}_6 + G^4_{32}\,\bar{h}_7 + G^4_{33}\,\bar{h}_3). \end{aligned} \tag{2-21}$$

## 2.2 GLOBAL MASS BALANCE AND FLUX

So far we have been looking at individual fractures, now we need to look at the entire fracture network and the boundary conditions imposed on the flow region in order to find the average head at each node and the flux through the system. Equation 2-20 gives us the total flux into or out of an intersection in terms of the average head at each of the nodes on the same fracture. For each internal node we have two expressions for flux, one for each of the fractures forming the intersection. The average head at each internal node is the same in either of the fractures. In order to maintain mass balance in the system, the sum of the fluxes into and out of an intersection must be zero. We can, therefore, establish a global relationship between flux and head for all nodes in the flow region fracture network and solve for average heads.

The global mass balance equations,

$$[A]*[\bar{h}]=[B], \tag{2-22}$$

are assembled from the G matrices of the individual fractures, Equation 2-20, and the constant average heads of the boundary nodes which are determined by the specified boundary conditions.

After solving for the average head at each internal node, we go back to the individual fracture to solve for flux. Substituting the solved and constant head values in Equation 2-20 yields the flux through each node. Then, for each of the six flow region faces, boundary node fluxes are summed to determine the directional flux through the system.

### 2.2.1 Boundary Conditions

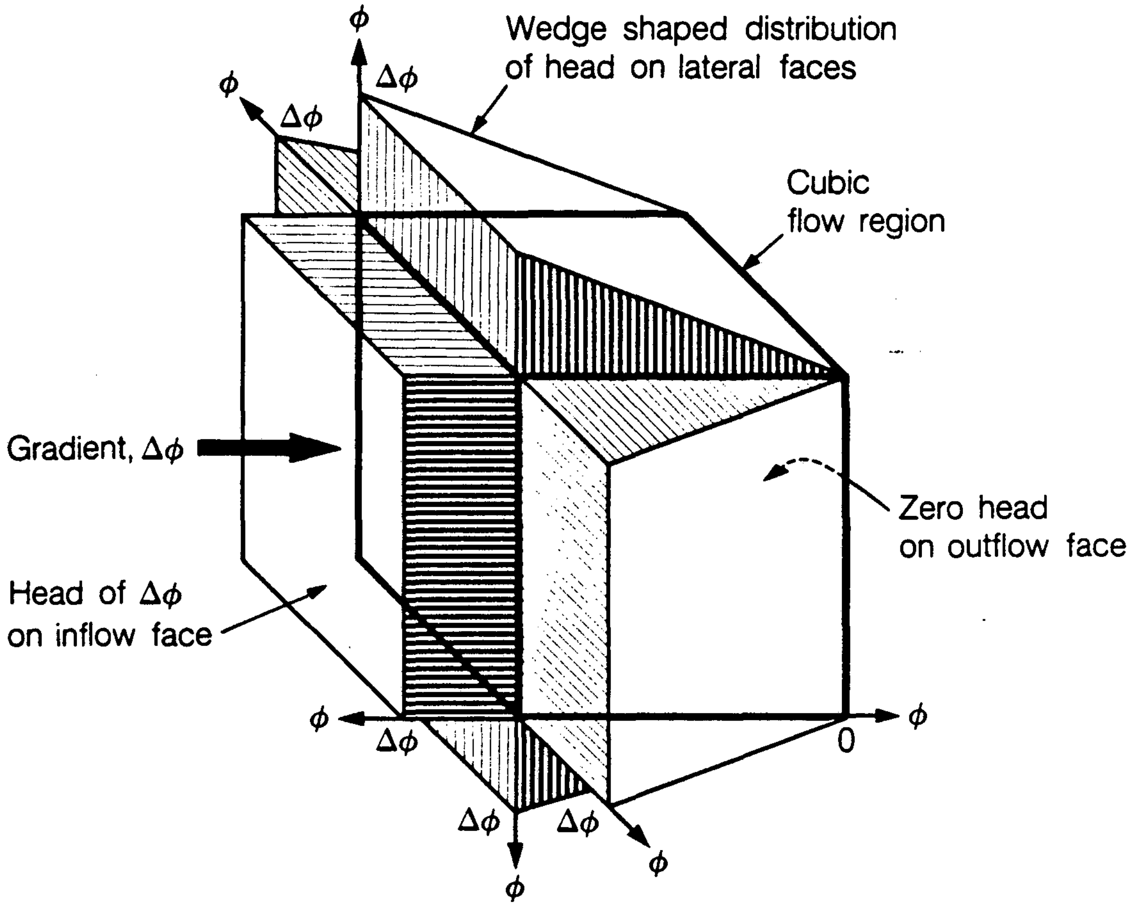
When using program DISCEL to measure directional permeability, flow region boundary conditions are set up to ensure a constant average gradient in the medium.

Using a cubic flow region as an example, Figure 2-4 diagrams the head distribution on the six boundary planes. The inflow face is assigned a head equal to the gradient, $\Delta\phi$, and the outflow face is assigned a head of zero. The other four sides have fixed linearly distributed or wedge shaped head distributions with a value of $\Delta\phi$ along edges intersecting the inflow face and the edges intersecting the outflow face. The head at any point on these four sides can, therefore, be found by linear interpolation. Boundary nodes are assigned constant average head values, $H_j$ accordingly. That is, boundary nodes on the inflow face are assigned a fixed head of $\Delta\phi$ , those on the outflow face are given a fixed head of zero, and for those lying on the other sides the head at each endpoint is computed from its position in the plane and the two values are averaged. When the global mass balance equations are formed if node j is a boundary node, the assigned constant head $H_j$ is used instead of solving for the average head at the node, $\bar{h}_j$. The direction of flow can be from side 1 to side 3, or from side 5 to side 6, or from side 2 to side 4 (refer to Figure 1-2 for side numbering convention). If flow is to go in the opposite direction, from side 3 to side 1 for example, then the gradient, $\Delta\phi$ , is negative.

For a given fracture network and flow region, up to three sets of boundary conditions can be used at one time. These can be three different flow directions, or three different gradients, or any desired combination. When the global mass balance equations, $[A]*[\bar{h}]=[B]$ (Equation 2-22), are set up, for each A matrix there can be one, two, or three B vectors which are solved for a corresponding number of head vectors.

Alternative types of boundary conditions may be used to solve problems other than directional permeability. For program validation we needed constant head values on all six flow region boundary planes. With this option, each boundary node is assigned a fixed head value according to the boundary plane in which it lies, regardless of the position of the node in that plane. This option was left in the code and others could easily be added as needed.

Subroutine BNDCON reads boundary condition parameters and computes and


ϕ
Wedge shaped distribution of head on lateral faces
Δϕ
ϕ
Δϕ
Cubic flow region
Gradient, Δϕ
Zero head on outflow face
Head of Δϕ on inflow face
ϕ
Δϕ
ϕ
0
Δϕ
Δϕ
ϕ
ϕ

XBL 8411-6161

Figure 2-4. Boundary conditions used to produce a constant gradient

stores head values. Up to three, [maxd], sets of boundary conditions are read in, either flow directions and gradients, [ibcc=1, ndir(maxd), delphi(maxd)] or fixed head values for the six flow region boundary planes, [ibcc=2, seth(6,maxd)]. Head values for each boundary node under each set of flow conditions are computed and stored in a linear array, [head(maxh)], which is keyed by global node number and the number of boundary conditions used, [keyh(maxnod)].

### 2.2.2 Formation and Solution of Global Mass Balance Equations

The relationship between flow in the individual fractures and flow in the global fracture network is based on three requirements:

1. Flow into an intersection must equal flow out of that intersection in order to maintain mass balance in the system.

2. The average head at an intersection is the same in each of the fractures forming the intersection.

3. Boundary nodes have known constant head values.

We will use an example to demonstrate formation of the global mass balance equations. A flow region and a network of five fractures is shown in Figure 2-5. These form nine nodes that are described in Table 2-1. Table 2-2 shows the local node numbering used to form shape factors, $G^k$, for each fracture, Equation 2-20, and the global node numbers used for the $Q_i$ and $\bar{h}_j$, as in Equation 2-21.

The resulting equations for flux in each fracture, in matrix form would be:

Fracture 1

$$\begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \end{pmatrix} = K^1 b^1 \begin{pmatrix} G^1_{11} & G^1_{12} & G^1_{13} \\ G^1_{21} & G^1_{22} & G^1_{23} \\ G^1_{31} & G^1_{32} & G^1_{33} \end{pmatrix} \begin{pmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \end{pmatrix} \qquad (2\text{-}23)$$

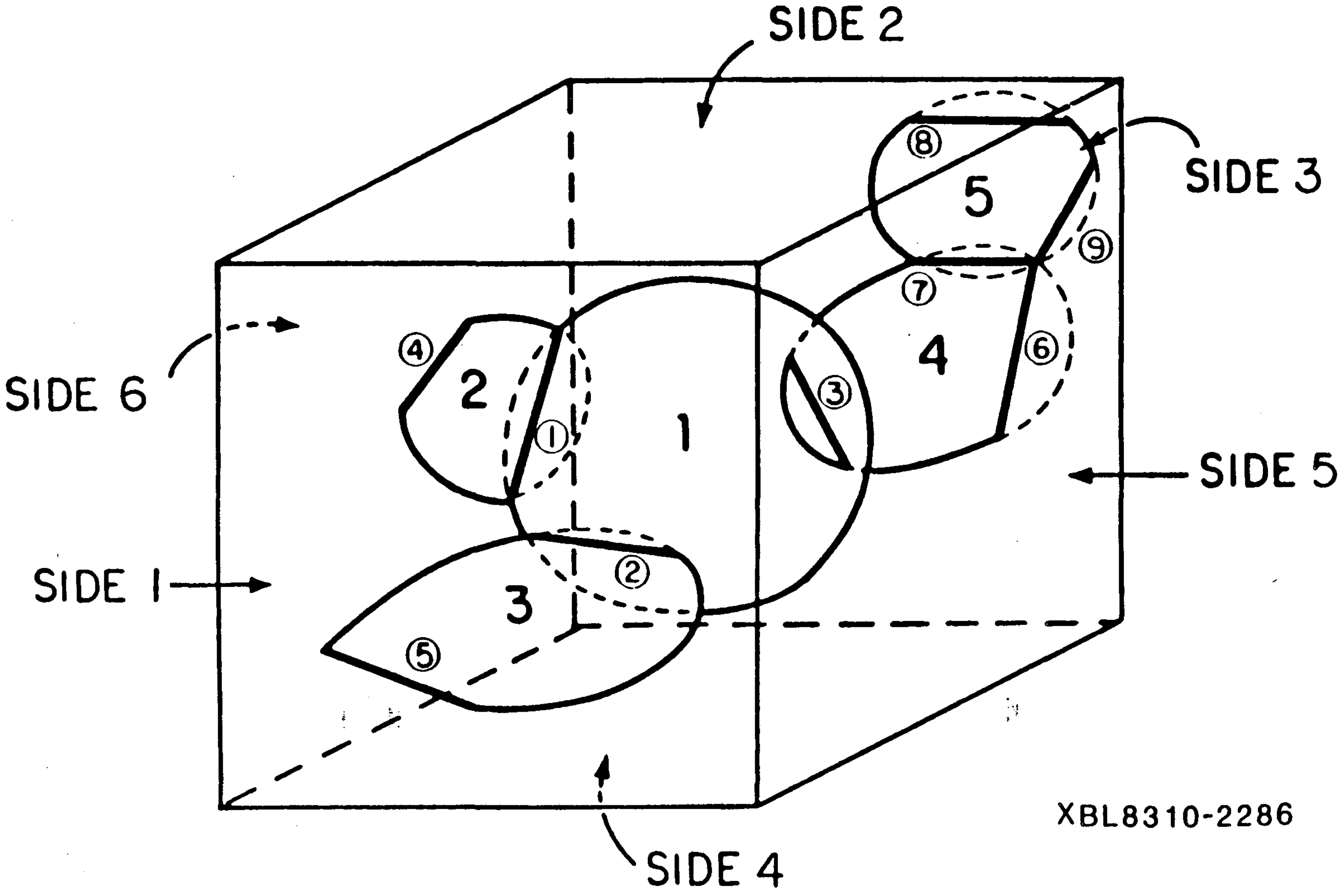


Figure 2-5. An example of a three-dimensional fracture system

Fracture 2

$$\begin{pmatrix} Q_4 \\ Q_1 \end{pmatrix} = K^2 b^2 \begin{pmatrix} G^2_{11} & G^2_{12} \\ G^2_{21} & G^2_{22} \end{pmatrix} \begin{pmatrix} \bar{h}_4 \\ \bar{h}_1 \end{pmatrix} \qquad (2\text{-}24)$$

Fracture 3

$$\begin{pmatrix} Q_5 \\ Q_2 \end{pmatrix} = K^3 b^3 \begin{pmatrix} G^3_{11} & G^3_{12} \\ G^3_{21} & G^3_{22} \end{pmatrix} \begin{pmatrix} \bar{h}_5 \\ \bar{h}_2 \end{pmatrix} \qquad (2\text{-}25)$$

Fracture 4

$$\begin{pmatrix} Q_6 \\ Q_7 \\ Q_3 \end{pmatrix} = K^4 b^4 \begin{pmatrix} G^4_{11} & G^4_{12} & G^4_{13} \\ G^4_{21} & G^4_{22} & G^4_{23} \\ G^4_{31} & G^4_{32} & G^4_{33} \end{pmatrix} \begin{pmatrix} \bar{h}_6 \\ \bar{h}_7 \\ \bar{h}_3 \end{pmatrix} \qquad (2\text{-}26)$$

Fracture 5

$$\begin{pmatrix} Q_8 \\ Q_9 \\ Q_7 \end{pmatrix} = K^5 b^5 \begin{pmatrix} G^5_{11} & G^5_{12} & G^5_{13} \\ G^5_{21} & G^5_{22} & G^5_{23} \\ G^5_{31} & G^5_{32} & G^5_{33} \end{pmatrix} \begin{pmatrix} \bar{h}_8 \\ \bar{h}_9 \\ \bar{h}_7 \end{pmatrix} \qquad (2\text{-}27)$$

**TABLE 2-1**
**Description of Nodes in Figure 2-5**

| Node No. | Description of Intersection |
|---|---|
| 1 | Internal intersection of fractures 1 and 2 |
| 2 | Internal intersection of fractures 1 and 3 |
| 3 | Internal intersection of fractures 1 and 4 |
| 4 | Boundary node, intersection of fracture 2 and side 6 |
| 5 | Boundary node, intersection of fracture 3 and side 1 |
| 6 | Boundary node, intersection of fracture 4 and side 5 |
| 7 | Internal intersection of fractures 4 and 5, truncated at side 5 |
| 8 | Boundary node, intersection of fracture 5 and side 3 |
| 9 | Boundary node, intersection of fracture 5 and side 5 |

**TABLE 2-2**
**Nodes in Each Fracture of Example in Figure 2-5**

| Fracture Number | Global Node Number | Local Node Number | Node Type |
|---|---|---|---|
| 1 | 1 | 1 | Internal |
| | 2 | 2 | Internal |
| | 3 | 3 | Internal |
| 2 | 4 | 1 | Boundary |
| | 1 | 2 | Internal |
| 3 | 5 | 1 | Boundary |
| | 2 | 2 | Internal |
| 4 | 6 | 1 | Boundary |
| | 7 | 2 | Internal |
| | 3 | 3 | Internal |
| 5 | 8 | 1 | Boundary |
| | 9 | 2 | Boundary |
| | 7 | 3 | Internal |

For each internal node, 1, 2, 3, and 7, there are two independent equations for flux and the sum of the fluxes must be zero. For each boundary node, 4, 5, 6, 8, and 9, there is only one equation for flux but the head is known, $\overline{h}_j = H_j$. Substituting known heads and setting the net flux for each node to zero gives us:

$$
\begin{aligned}
&0 = K^1b^1 (G^1_{11}\,\overline{h}_1 + G^1_{12}\,\overline{h}_2 + G^1_{13}\overline{h}_3) + K^2b^2 (G^2_{21}\,H_4 + G^2_{22}\overline{h}_1)\\
&0 = K^1b^1 (G^1_{21}\,\overline{h}_1 + G^1_{22}\,\overline{h}_2 + G^1_{23}\,\overline{h}_3) + K^3b^3 (G^3_{21}\,H_5 + G^3_{22}\,\overline{h}_2)\\
&O = K^1b^1 (G^1_{31}\,\overline{h}_1 + G^1_{32}\,\overline{h}_2 + G^1_{33}\,\overline{h}_3) + K^4b^4 (G^4_{31}\,H_6 + G^4_{32}\,\overline{h}_7 + G^4_{33}\,\overline{h}_3)\\
&\overline{h}_4 = H_4\\
&\overline{h}_5 = H_5\\
&\overline{h}_6 = H_6\\
&O = K^4b^4 (G^4_{21}\,H_6 + G^4_{22}\,\overline{h}_7 + G^4_{23}\,\overline{h}_3) + K^5b^5 (G^5_{31}\,H_8 + G^5_{32}\,H_9 + G^5_{33}\,\overline{h}_7)\\
&\overline{h}_8 = H_8\\
&\overline{h}_9 = H_9
\end{aligned}
\tag{2-28}
$$

These equations form the global mass balance equations, $[A][\overline{h}]=[B]$. Terms involving unknown heads, $\overline{h}_j$, fill the A matrix while those with known heads, $H_j$, fill the B vector.

$$
\begin{bmatrix}
\begin{pmatrix} K^1b^1G^1_{11} \\ + \\ K^2b^2G^2_{22} \end{pmatrix} & K^1b^1G^1_{12} & K^1b^1G^1_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\
K^1b^1G^1_{21} & \begin{pmatrix} K^1b^1G^1_{22} \\ + \\ K^3b^3G^3_{22} \end{pmatrix} & K^1b^1G^1_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\
K^1b^1G^1_{31} & K^1b^1G^1_{32} & \begin{pmatrix} K^1b^1G^1_{33} \\ + \\ K^4b^4G^4_{33} \end{pmatrix} & 0 & 0 & 0 & K^4b^4G^4_{32} & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & K^4b^4G^4_{23} & 0 & 0 & 0 & \begin{pmatrix} K^4b^4G^4_{22} \\ + \\ K^5b^5G^5_{33} \end{pmatrix} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\times
\begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \\ \bar{h}_4 \\ \bar{h}_5 \\ \bar{h}_6 \\ \bar{h}_7 \\ \bar{h}_8 \\ \bar{h}_9 \end{bmatrix}
=
\begin{bmatrix} -K^2b^2G^2_{21}H_4 \\ -K^3b^3G^3_{21}H_5 \\ -K^4b^4G^4_{31}H_6 \\ H_4 \\ H_5 \\ H_6 \\ \begin{pmatrix} -K^4b^4G^4_{21}H_6 \\ -K^5b^5(G^5_{31}H_8+G^5_{32}H_9) \end{pmatrix} \\ H_8 \\ H_9 \end{bmatrix}
$$

(2-29)

In general most of the elements of the B vector will be zero. However, in this simple example, each fracture is either connected to a boundary or to another fracture that is connected to a boundary. As a result all the elements in the B vector are non-zero.

We can now solve the global mass balance equations for the average head at each internal node, $\bar{h}_j$. Matrix A is sparse and symmetric, so only its upper triangle non zero elements are stored in a one-dimensional array [a(nonz)] by subroutine FULFIL. FULFIL also stores the nonzero elements of the B vector in [bmm(maxnod,3)]. The appropriate elements of both arrays are filled as each fracture's G matrix is computed. The matrix inversion is performed by subroutine CGSOLV, using a conjugate gradient iterative algorithm (Crochet et Al, 1984).

### 2.2.3 Solving for Flux

The last step is to determine the directional flux through the fracture system by summing the fluxes through each of the six flow region boundary planes. Solution of the global mass balance equations gave us the average head at each internal node and the boundary conditions gave us the average head at each boundary node. Now we go back to the solution for flux developed for each individual fracture and substitute the average head values in Equation 2-20. Only the G matrices for those fractures that intersect one or more boundary planes were stored since only boundary node fluxes are needed to compute directional flux. Each internal node has two independent equations for flux, one for each of the fractures that form the intersection.

Subroutine SFLUX computes flux on a fracture by fracture basis. First it sets up the local node numbering system for the fracture, as before. It then retrieves the appropriate stored G matrix from the linear array [Gstore(maxn*maxfrc)] and the average head values from the globally numbered array, [hd(maxnod,maxd)]. There may be up to three sets of head values depending upon the number of sets of boundary conditions being used (Section 2.2.1). The fluxes are then computed using Equation 2-20 and boundary node fluxes are summed for each face of the flow region [fluxb(6,maxd)].

## 3.0 FRACTURE GENERATION PROGRAM FMG3D

The purpose of program FMG3D is to create a fracture network input file for the fluid flow program, DISCEL. To do this, FMG3D can either start with a specified fracture system or generate a random system based on stochastic variables provided by the user. By either process, the location, orientation, radius, and aperture of all fractures in a generation region are determined. Next, FMG3D identifies only those fractures that are within a subregion called the flow region and are connected by at least one path to its boundaries. Flow through the fracture network in this region will be calculated by DISCEL. FMG3D determines the boundary nodes which are the intersections between fractures and flow region boundary planes and intersections between fractures, called internal nodes. Non-conducting fractures are eliminated from the fracture system. The geometric information needed to calculate flow is written to a file to be read by DISCEL.

The first step is to generate a primary fracture system within a spherical generation region. The fracture system consists of one or more fracture sets. The number of sets, the number of fractures per set, and the radius of the spherical generation region are determined by the user. The location, orientation, and size of each fracture may either be specified by the user or fracture characteristics may be stochastically generated on a set by set basis. In the stochastic mode, fracture centers are randomly located and uniformly distributed throughout the generation region. A spherical generation region is used to allow full rotation of any flow region.

Distribution functions that can be used to generate fracture orientation angles, radii, and apertures currently include normal, lognormal, exponential, uniform, and normal with an option to correlate fracture aperture to fracture radius. Fracture radii are usually allowed to vary lognormally, just as fracture lengths varied in the two-dimensional model of Long et al. (1982). Fracture apertures are usually generated from a lognormal distribution. Fracture orientation angles will probably require the addition of one or more distribution functions such as Arnold's hemispherical normal (Mahtab et al., 1972) or Bingham's (Mahtab and Yegulalp, 1982), depending on the case studied. In general, field data will dictate which distribution function is appropriate for each fracture characteristic.

For fractures extending beyond the boundary of the generation region, the fracture area is truncated. This is done for statistical purposes only, so that the mean and standard deviation of fracture areas within the generation region can be calculated and compared to the input parameters.

The cube enclosing the generation sphere is then divided into a number of cubic subregions, and the subregions which each fracture intersects are recorded. This will be used to speed up the search for intersections between fractures, which otherwise would be by far the most time consuming calculation. The number of divisions in the three directions is specified by the user. The subregion division will also be used by the plotting routine DIMES to select the plotting of a subset of the fracture mesh.

Next, a rectangular parallelepiped flow region is defined within the generation region. Input parameters that define the region include three side lengths and two orientation angles. The flow rgeion is centered in the generation sphere. For a given primary fracture system, any number of flow regions of varying size and orientation may be defined and corresponding flow meshes generated.

The system of fractures lying within the flow region is determined by comparing the position of each fracture with the position of the six flow region boundary planes. Fractures lying entirely inside the flow region are saved and fractures lying entirely outside the flow region are discarded. Fractures intersecting boundary planes are saved and line segments of intersection stored as boundary nodes. A fracture may intersect more than one boundary plane and, therefore, contain more than one boundary node. Fracture areas are truncated at boundary planes for statistical purposes only; the entire fracture disc is used to solve for flow in the fracture.

The system of fractures contributing to flow through the region is determined next. Intersections between fractures are found and stored as internal nodes. Flow in a fracture can only take place between nodes. This can be between two or more internal nodes, two or more boundary nodes or any combination of internal and boundary nodes. Any fracture containing less than two nodes can not conduct flow and is, therefore, discarded from the system of conducting fractures.

Finally, flow network information is prepared and written for input to the fluid flow model, DISCEL. Nodal arrays are assembled from stored boundary and internal nodes on conducting fractures in the flow region. The nodes are sorted and renumbered in order to reduce the bandwidth of the global mass balance matrix in DISCEL. All necessary run identification, fracture system, flow region, and nodal data is written to the file. This section describes program FMG3D, Appendices D, E, F, and G supply more detailed descriptions of some computations. The user's guide for the program as well as a listing of the code are provided in a separate report.

## 3.1 GENERATION OF FRACTURE SYSTEM

A primary fracture system, consisting of one or more sets of circular fractures randomly distributed within a spherical generation region, is created and fractures extending beyond the boundary are truncated.

### 3.1.1 Fracture System Characteristics

In the stochastic mode the primary fracture system is generated using parameters specified by the user. These include the radius of the spherical generation region, $r_g$ [rgene], the number of fracture sets [nsets], either the number of fractures per set [nfrac] or the fracture density (i.e., number of fractures per unit volume) per set [rlamb], and distribution parameters for generating fracture orientation angles, radii, and apertures. The fracture centers are randomly generated within the region. The main program, FMG3D, calls subroutine FRAGEN to read input parameters and coordinate generation of fracture characteristics. This section describes how FRAGEN works in the stochastic mode. Alternatively fracture centers, orientation angles, radii, and apertures can be specified for any or all fracture sets.

**Fracture Centers.** The fracture centers are generated by subroutine RANCEN to be randomly distributed throughout the spherical region. Coordinates of fracture centers ($x_c, y_c, z_c$) are computed by generating triples of random numbers, uniformly distributed between zero and one, then scaling them by multiplying by twice the generation region radius $r_g$, and subtracting $r_g$. This makes the center of the sphere the origin of the rectangular coordinate system. The resultant triples are rejected if they do not satisfy the condition $x_c^2 + y_c^2 + z_c^2 \leq r_g^2$; i.e., the point must be within the sphere.

**Orientation.** The orientation of each fracture plane is determined by the angular coordinates of the normal to the plane, i.e., the dip, $0 \leq \phi' \leq 90°$, and the azimuth of dip, $0 \leq \theta' \leq 360°$ (Figure 3-1). The direction cosines of the unit vector representing the fracture plane are derived from $\phi'$ and $\theta'$ :

$$\begin{aligned} \lambda &= \sin\phi' \cos\theta' \\ \mu &= -\sin\phi' \sin\theta' \\ \nu &= \cos\phi' \ . \end{aligned} \tag{3-1}$$

Fracture orientation angles are either input or generated by the program. If statistically generated, the mean orientation vector of each fracture set is required as well as other parameters, depending on the distribution chosen. Distribution functions which might be appropriate as program options are Arnolds hemispherical normal (Mahtab et al., 1972) and Binghams (Mahtab and Yegulalp, 1982). Any distributiion which fits field data can be incorporated into the model.

**Fracture Raddi, Areas, and Apertures.** The radius and aperture of each fracture are generated according to a normal, lognormal or exponential distribution. The generation procedure for the first two distributions requires the mean [ev] and standard deviation [sd] for each fracture set. The exponential distribution requires only the mean.

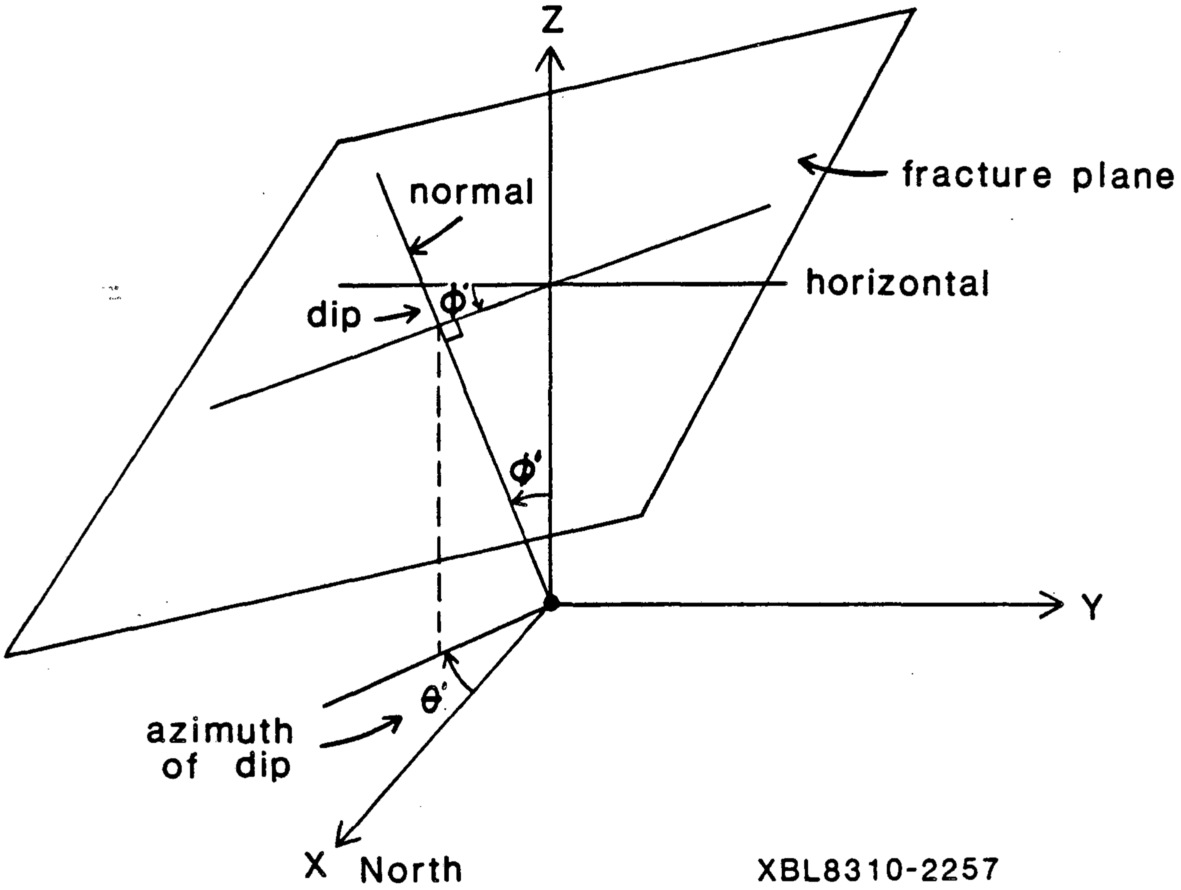


Figure 3-1. Orientation of normal to fracture place determined by angle of dip, $\phi'$ , and aximuth of dip $\theta'$

Apertures may also be correlated with fracture radius. Fracture areas and their original mean and standard deviation are computed for each set.

### Statistical Simulation

*Random number generator.* The statistical distribution subroutines and RANCEN use a random number generator called GGUBFS which is an International Mathematical and Statistical Library (IMSL) subroutine. GGUBFS returns random numbers uniformly distributed between zero and one and requires a double precision seed value [dseed]. GGUBFS returns a different random number each time it is called within a program, however, the same sequence of random numbers is produced each time the program is run with the same initial seed. This mode of operation is optionally overridden by generating an arbitrary seed,

$$\text{dseed} = \text{SECNDS}(0.0) * 100.0. \qquad (3\text{-}2)$$

SECNDS is a VAX-11 FORTRAN function subprogram which returns the system time of day in seconds less the value of its argument. An input flag [iranf] controls whether the seed is read or generated, and the initial seed is printed out. Since the seed defines the starting location for the random number generator, the user can reproduce a series of random numbers, i.e., reproduce a random fracture system by inputting the same initial seed in a later run.

*Random generation of fracture centers.* Subroutine RANCEN calls GGUBF once for each coordinate of the fracture center. The coordinates ( $x_c, y_c, z_c$) are computed from the random numbers by the equations

$$\begin{aligned} x_c &= \text{FLOAT } [\text{INT}(2r_g*10^n*\text{GGUBFS(d)})]/10^n - r_g, \\ y_c &= \text{FLOAT } [\text{INT}(2r_g*10^n*\text{GGUBFS(d)})]/10^n - r_g, \\ z_c &= \text{FLOAT}[\text{INT}(2r_g*10^n*\text{GGUBFS(d)})]/10^n - r_g, \end{aligned} \qquad (3\text{-}3)$$

where n is the number of decimal places in the coordinate [itole] $r_g$ is the radius of the generation sphere [rgene] d is a dummy variable initially equal to the input or generated seed [dseed] then reset by GGUBFS. INT and FLOAT are intrinsic library functions which convert a real number to an integer by truncation and an integer to a real number, respectively. Truncating coordinates to n decimal places limits the minimum distance between fracture centers.

## Random Generation of Fracture Characteristics

*Normal distribution.* In subroutine NORDIS, the sum $S_N$ is calculated by calling GGUBFS twenty-five times and accumulating the sum,

$$S_N=\sum_{n=1}^{25} r_n \qquad (3\text{-}4)$$

where $r_n$ equals the value returned by a call to GGUBFS, $r_n$ = GGUBFS(d). As shown by Hammersly and Hanscomb (1964), $S_N$ is distributed normally with an expected value of 25/2 and a variance of 25/12; therefore,

$$S_N^* = S_N - \frac{25}{2}\sqrt{\frac{25}{12}} \qquad (3\text{-}5)$$

is distributed normally with expected value 0 and variance 1. If $\mu$ and $\sigma$ are the expected value and standard deviation supplied by the user [ev and sd] then

$$x = \sigma S_N^* + \mu \qquad (3\text{-}6)$$

is distributed normally, N ( $\mu$ , $\sigma$ ) with the specified parameters. (Note that in this equation, x does not refer to a point coordinate.)

*Lognormal distribution.* If $S_N^*$ and x are defined as in the previous section, then $S_N^*$ is distributed N(0,1), and x is distributed N($\mu$,$\sigma$), and y = exp(x) is distributed lognormally. In terms of the parameters $\mu$ and $\sigma$ of the normal distribution for x, the mean $\alpha$ and variance $\beta^2$ of the lognormal distribution are

$$\alpha = \exp(\mu)\exp(\sigma^2/2)$$

and

$$\beta^2 = \exp(\sigma^2+2\mu)[\exp(\sigma^2) - 1]. \qquad (3\text{-}7)$$

Since the user will specify $\alpha$ and $\beta$, it is necessary to solve for $\mu$ and $\sigma$ in terms of these variables:

$$\mu = 2 \ln \alpha - \frac{1}{2} \ln (\beta^2 + \alpha^2), \qquad (3\text{-}8)$$

$$\sigma = \sqrt{\ln(\beta^2 + \alpha^2) - 2 \ln \alpha}.$$

Therefore, subroutine LOGDIS can calculate y from

$$y = \exp(\sigma S_N^* + \mu), \tag{3-9}$$

where $\mu$ and $\sigma$ are defined above and $\alpha$ and $\beta$ are specified by the user [ev and sd].

*Exponential distribution.* In subroutine EXPDIS, $\mu$ is the expected value given by the user [ev] and

$$x = \mu \ln(1 - r) \tag{3-10}$$

is distributed exponentially, where r = GGUBFS(d).

*Normal distribution correlating two variables.* Subroutine NORDIS1 generates random variables, x, distributed normally where the expected value, $\mu$, is proportional to the logarithm of another parameter, $x_1$. This correlation is used to compute fracture aperture as a function of radius. $S_N^*$ is defined as before (Equation 3-5) and the standard deviation, $\sigma$ , is supplied by the user [sd]. The user also supplies the y-intercept and the slope [ycept and slope] of a linear relationship between mean values of the variable, x, and given values of the logarithm of $x_1$. The expected value of x, $\mu$ is computed,

$$\mu = \text{ycept} + \text{slope}*\log_{10}(x_1) \tag{3-11}$$

and x is computed as before, (Equation 3-6). Since this can result in values of x that are less than or equal to zero, which is not reasonable when x is the aperture, a minimum value for x is set in the subroutine.

*Uniform distribution.* Subroutine UNIDIS generates random variables, $a_i$, distributed uniformly over a given range, amin to amax:

$$a_i = (\text{amax} - \text{amin})*\text{GGUBFS(dseed)} + \text{amin} \tag{3-12}$$

If the parameter being generated is $\phi'$ , the dip angle, the distribution is corrected to give effectively uniform orientations on the hemisphere: cos $\phi$ is generated uniformly instead of $\phi$ directly.

*Additional distributions.* The fracture generation code can easily be modified to include additional distribution functions that are found to be appropriate for any of the fracture characteristics.

### 3.1.2 Equations of Fracture Planes

The coefficients of the equation of the plane in which a fracture lies are computed by subroutine EQPLAN from the fracture's orientation angles, $\phi'$ and $\theta'$. The equation of a plane in space is

$$ax + by + cz + d = 0 \tag{3-13}$$

where a, b, and c are direction numbers of a normal to the plane. The equation is in normal form when a, b, and c are direction cosines of the normal vector and -d is then the perpendicular distance from the origin to the plane. Since direction cosines of the normal vector are $\lambda$, $\mu$, and $\nu$ (Equations 3-1) and the center of the fracture disc $(x_c,y_c,z_c)$ lies in the plane, then

$$\lambda x + \mu y + \nu z + d = 0,$$

and

$$\lambda x_c + \mu y_c + \nu z_c + d = 0. \tag{3-14}$$

Therefore,

$$a = \lambda, \quad b = \mu, \quad c = \nu,$$

and

$$d = -\lambda x_c - \mu y_c - \nu z_c \tag{3-15}$$

are the coefficients of the equation of that plane. Figure 3-2 diagrams determination of the location of a circular fracture in space.

### 3.1.3 Definition of cubic subregions

Depending on an input parameter [iord] given by the user, Subroutine SUBINT divides the cube enclosing the generation sphere into $[\text{iord}]^3$ equal subregions (or "cubes") and records the fractures intersecting each cube and the cubes intersecting each fracture. The number of the cube where the center of the fracture is located is first determined. Next, the intersections between the fracture and the six boundaries of the cube are

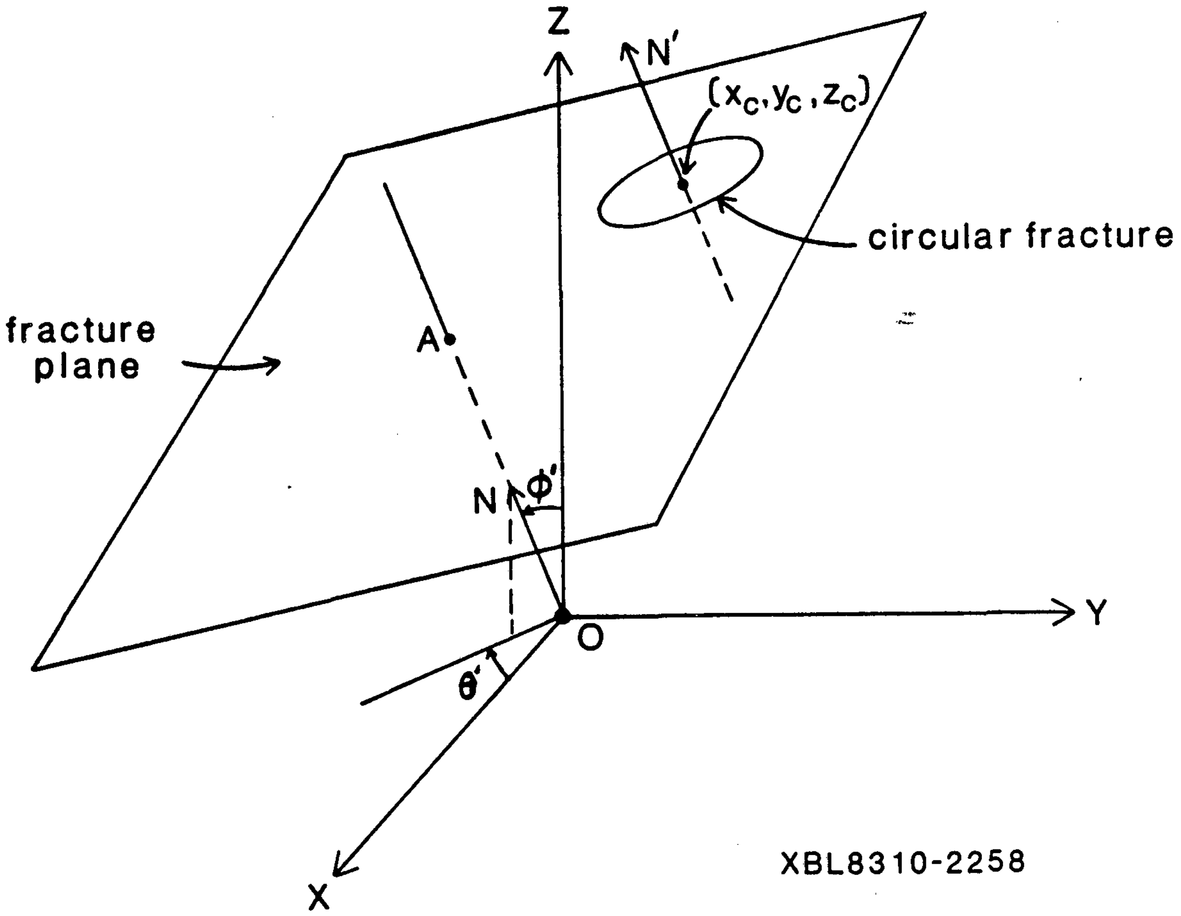


Figure 3-2. Locating circular fracture with center $(s_c,y_c,z_c)$ in a plane normal to unit vector N. The perpendicular distance from the origin to the plane is distance OA, where A is the point of intersection of the plane and the extension of vector N. Vector N′ passes through the point $(x_c,y_c,z_c)$, is parallel to vector N and therefore, is normal to the fracture plane

looked for. For each boundary, if there is an intersection, the fracture also intersects an adjacent cube of known number which is recorded. The search for boundary intersections is repeated for each newly found intersecting cube until no new cube is found.

To facilitate this, the axes are rotated such that the new X-Y plane is parallel to the plane of the fracture, thus reducing the problem to two dimensions.

*Rotation of axes.* Subroutine ROFRAC computes a rotation matrix for a given fracture, with center at $(x_c, y_c, z_c)$ lying in the plane ax + by + cz + d = 0, that will rotate the Z-axis such that it is normal to the plane. The new Z-axis, Z', has direction cosines a, b, and c relative to the XYZ system, intersects the plane at the point (-ad, -bd, -cd) and is at a distance -d from the origin (see Appendix D, part a). The X-axis is rotated such that it is parallel to a line through points (-ad,-bd,-cd) and $(x_c, y_c, z_c)$ as shown in Figure 3-3a. The direction of this line is $(x_c + ad, y_c + bd, z_c + cd)$ and dividing by the distance, k, between the two points yields direction cosines $l_1$, $l_2$, and $l_3$ where

$$k = (x_c + ad)^2 + (y_c + bd)^2 + (z_c + cd)^2$$
$$l_1 = \frac{x_c + ad}{k}, \quad l_2 = \frac{y_c + bd}{k}, \quad l_3 = \frac{z_c + cd}{k}. \qquad (3\text{-}16)$$

The orthogonal matrix of rotation [rotf] from the XYZ coordinate system to the X'Y'Z' system is, therefore,

$$\begin{pmatrix} l_1 & m_1 & a \\ l_2 & m_2 & b \\ l_3 & m_3 & c \end{pmatrix}$$

where

$$\begin{aligned} m_1 &= l_3 b - l_2 c \\ m_2 &= l_1 c - l_3 a \\ m_3 &= l_2 a - l_1 b. \end{aligned} \qquad (3\text{-}17)$$

The equation of the fracture plane reduces to z' + d = 0; the points (-ad, -bd, -cd) and $(x_c, y_c, z_c)$ become (0, 0, -d) and (k, 0, -d), respectively (see Figure 3-3b).

If the distance k equals zero, that is $(x_c, y_c, z_c)$ and (-ad, -bc, -cd) are coincident, then the rotation matrix must be generated in a different manner. The Z-axis is rotated

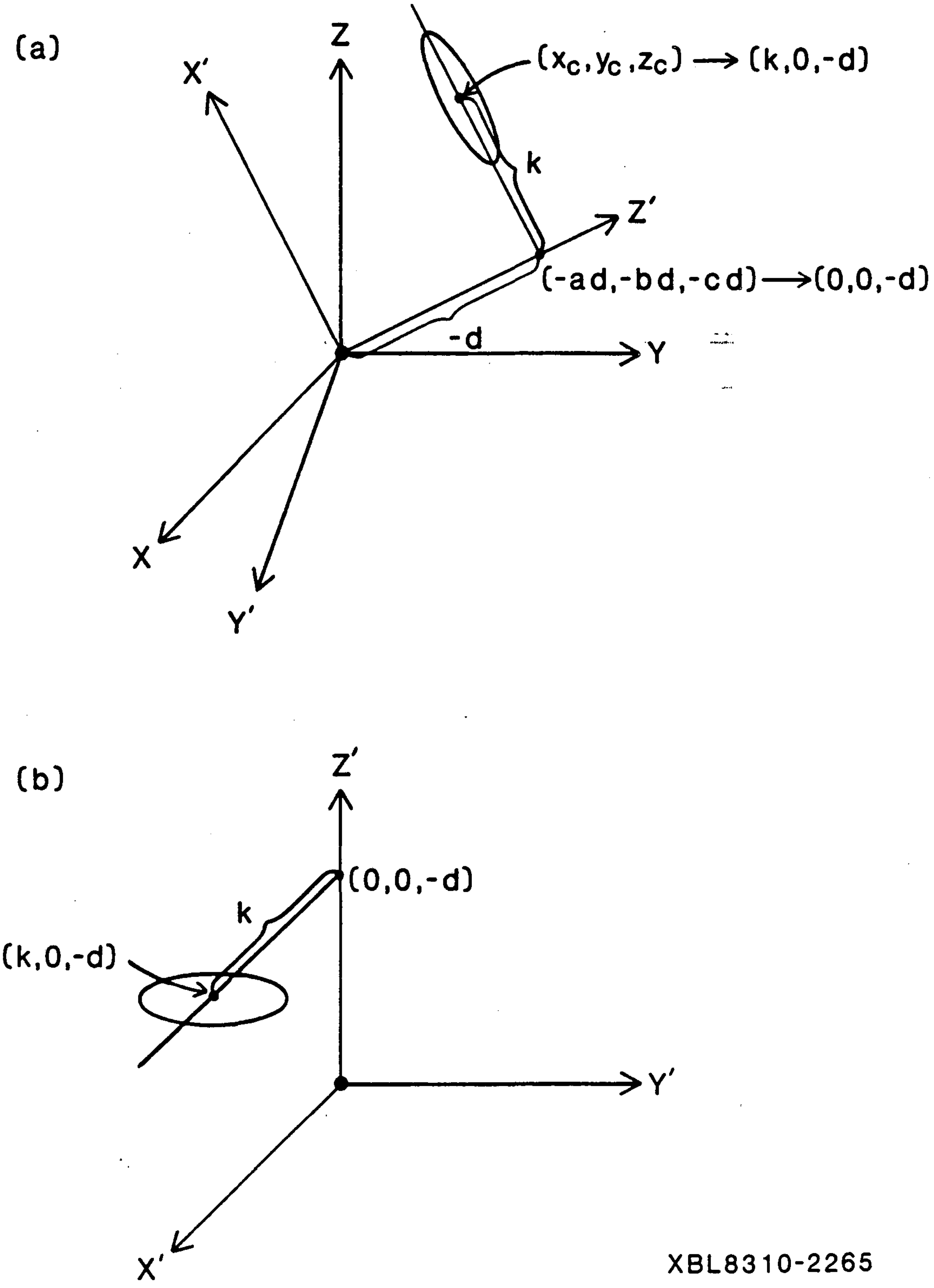


Figure 3-3. Rotation to a coordinate system convenient to fracture plane ax + by + cz + d = 0

to have direction cosines a, b, and c, as before, but any other rotation is arbitrary as long as the result is orthogonal. For convenience, the orientation angles of the fracture plane, $\phi'$ and $\theta'$ , are used since a, b, and c were computed from them (Section 3.1.2). By letting

$$l_1 = \cos\phi' \ \cos\theta'$$
$$l_2 = -\cos\phi' \ \sin\theta'$$
$$l_3 = -\sin\phi'$$

and

$$m_1 = \sin\theta' \qquad (3\text{-}18)$$
$$m_2 = \cos\theta'$$
$$m_3 = 0,$$

an orthogonal rotation is assured. The equation of the fracture plane reduces to z' + d = 0 and point ($x_{c,}$ $y_{c,}$ $z_c$) becomes (0, 0, -d).

Subroutine RFLOFRC transforms the cube boundary planes to the X'Y'Z' coordinate system through the orthogonal rotation

$$(a_i\ b_i\ c_i) \begin{bmatrix} l_1 & m_1 & a \\ l_2 & m_2 & b \\ l_3 & m_3 & c \end{bmatrix} = (a_i'\ \ b_i'\ \ c_i'\ ),\ i{=}1 \text{ to } 6. \qquad (3\text{-}19)$$

The perpendicular distance from the origin to the plane, $-d_i$, remains the same and the equations of the boundary planes become

$$a_i'\,x' \ + b_i'\,y' \ + c_i'\,z' \ + d_i = 0, \quad i = 1 \text{ to } 6. \qquad (3\text{-}20)$$

The six cube boundary planes intersect the fracture plane, z' = -d, along the lines

$$a_i'\ \ x' \ + b_i'\ \ y' \ - c_i'\ \ d + d_i = 0,\ i = 1 \text{ to } 6. \qquad (3\text{-}21)$$

SUBINT then determines if the circular fracture f, with center (k, 0) and radius $r_f$, intersects these lines. If an intersection is found, the number of the corresponding adjacent cube is checked against the previously found cubes, and if new, the cube is recorded. Array [icub(maxf2)] keyed by array [icubk(maxfrc)] records all cubes intersecting each fracture, and array [ifcu(maxf2)] keyed by array [ifcuk(maxcub)] records all fractures intersecting each cube.

### 3.1.4 Truncation at Boundary

Fractures extending beyond the spherical generation region are truncated at the boundary and the area of truncation is computed for each fracture by subroutine TRUNCG. The truncated area is used for statistical calculations only in order to determine mean and standard deviation of fracture areas entirely within the generation region. The method is given in Appendix D.

### 3.1.5 Statistical calculations

At the end of the generation stage, in order to compare the generated fracture system with the input parameters provided by the user, simple statistical calculations are performed. Subroutine PFS computes and prints

- the number of fractures, number of truncated fractures and volumic density for each fracture set and overall.
- the mean and angular deviation of orientation for each set, using subroutine STATOR.
- the mean and standard deviation of radius, total area, area after truncation, and aperture for each fracture set.

These statistics are computed two other times in the program, when the fracture network is altered.

## 3.2. FRACTURE SYSTEM IN THE FLOW REGION

The flow region, which is the volume to be analyzed, is included in the generation sphere and only fractures within this region are considered for input to the fluid flow model. Fractures extending beyond the flow region are truncated at the boundary and those lying entirely outside the region are discarded. Truncated fracture areas are used for statistical calculations only, the fluid flow model considers the entire fracture disc. Intersections between fracture discs and flow region boundary planes are identified at this time. These line segments, called boundary nodes, will be assigned fixed head values by the fluid flow program, DISCEL.

### 3.2.1 Flow Region

The flow region is a rectangular parallelepiped centered at the origin of the generation sphere. The dimensions, Xmesh, Ymesh, and Zmesh and orientation angles $\phi_R$ [rophi] and $\theta_R$ [rothe], are input parameters and must be such that the resulting region lies completely within the generation sphere. Figure 3-4 shows the flow region without rotation, i.e., $\phi_R = \theta_R = 0$. Figure 3-5 diagrams rotation of the XY-plane through angle $\theta_R$ followed by rotation of the ZX'-plane through angle $\phi_R$. Subroutine ROFLOW applies both rotations and computes the coefficients of the equations of the six boundary planes, listed in Table 3-1, where

$$\begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \\ \nu_1 & \nu_2 & \nu_3 \end{bmatrix} = \begin{bmatrix} \cos\phi_R \cos\theta_R & -\cos\phi_R \sin\theta_R & -\sin\phi_R \\ \sin\theta_R & \cos\theta_R & 0 \\ \sin\phi_R \cos\theta_R & -\sin\phi_R \sin\theta_R & \cos\phi_R \end{bmatrix}. \qquad (3\text{-}22)$$

**TABLE 3-1.**
**Coefficients of the equations of the six boundary planes.**

| i | $a_i$ | $b_i$ | $c_i$ | $d_i$ |
|---|---|---|---|---|
| side 1 | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | -Xmesh/2 |
| side 2 | $\nu_1$ | $\nu_2$ | $\nu_3$ | -Zmesh/2 |
| side 3 | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | Xmesh/2 |
| side 4 | $\nu_1$ | $\nu_2$ | $\nu_3$ | Zmesh/2 |
| side 5 | $\mu_1$ | $\mu_2$ | $\mu_3$ | -Ymesh/2 |
| side 6 | $\mu_1$ | $\mu_2$ | $\mu_3$ | Ymesh/2 |

Since sides 1 and 3 are parallel, $a_1 = a_3$, $b_1 = b_3$, $c_1 = c_3$, and $d_1 = -d_3$ . Similar equalities hold for sides 2 and 4 and sides 5 and 6.

### 3.2.2 Fractures in Flow Region

The next step is to determine which fractures in the primary fracture system lie within the flow region. Subroutine TRUNCF determines boundary nodes, truncates and

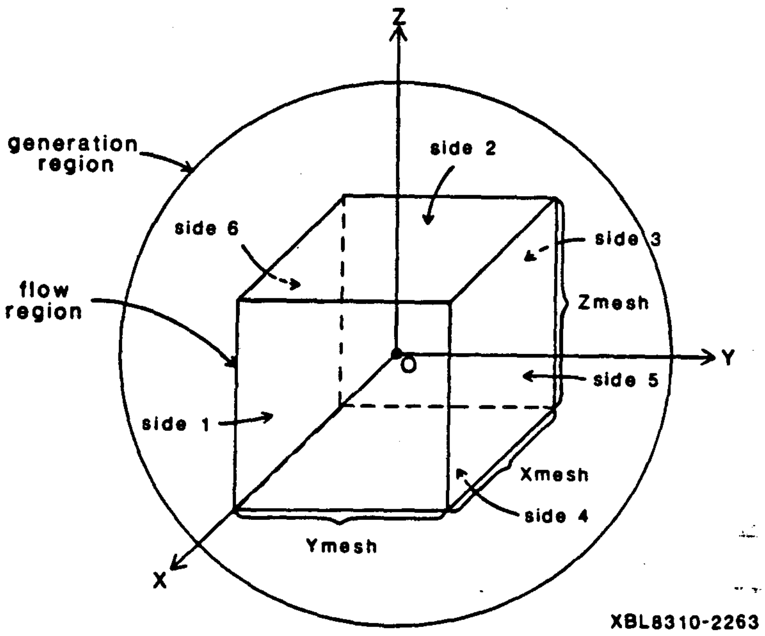


Figure 3-4. Flow region within spherical generation region. Rotation angles $\phi_R$ and $\theta_R$ equal zero.

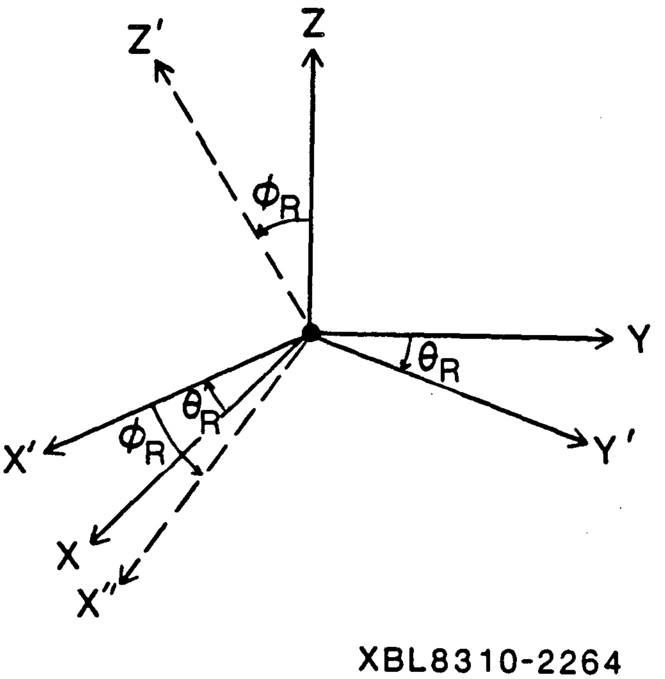


Figure 3-5. Rotation of the XY-plane through angle $\theta_R$ followed by rotation of the $ZY'$ - plane through angle $\phi_R$ .

renumbers fractures which lie partially outside the flow region, and flags those which lie completely outside the flow region for elimination. To facilitate this, the axes are rotated such that the new XY-plane is parallel to the plane of each fracture in turn.

*Rotation of axes* This is done in exactly the same way as for the cubic subregions treatment, section 3.1.3, using the same subroutines, ROFRAC to compute a rotation matrix and then RFLOFRC to transform the flow region boundary planes to the new coordinate system.

*Position of fracture relative to boundary planes* Keeping the same notations as in section 3.1.3, the six flow region boundary planes intersect the fracture plane, $z' = -d$, along the lines specified by Equation 3-21:

$$a_i' \ x' + b_i' \ y' + c_i' \ d + d_i = 0, \quad i = 1 \text{ to } 6$$

These six lines [flcoef(6,4)] form three pairs of parallel lines corresponding to parallel planes 1 and 3, 2 and 4, and 5 and 6. Subroutine FLAG determines the position of the circular fracture f, with center at (k,0) and radius $r_f$, relative to these boundary lines and sets flags accordingly. One flag [nflag(nf), nf=1, nfrac] is set to -1, 0, or +1 to indicate whether the fracture, [nf], is to be discarded, saved, or truncated, respectively. A second flag array [ntrunc(i), i=1, 7] is set with truncation codes for each of the six boundary lines and also stores the total number of lines intersected. This code indicates the number of points of intersection, zero, one, or two, on each line and the position of the fracture center relative to each intersected line. If a fracture circle is found to be outside any pair of parallel boundary lines, it is necessarily outside the flow region and, therefore, flagged for discarding. If it is entirely inside all three pairs of lines, it is flagged to be saved without truncation, as shown in Figure 3-6. The method for determining the position of the fracture circle relative to the boundary lines is given in Appendix E.

When the fracture position has been compared with all three pairs of boundary lines, the fracture is discarded, saved without truncation, or further screened for truncation as explained below.

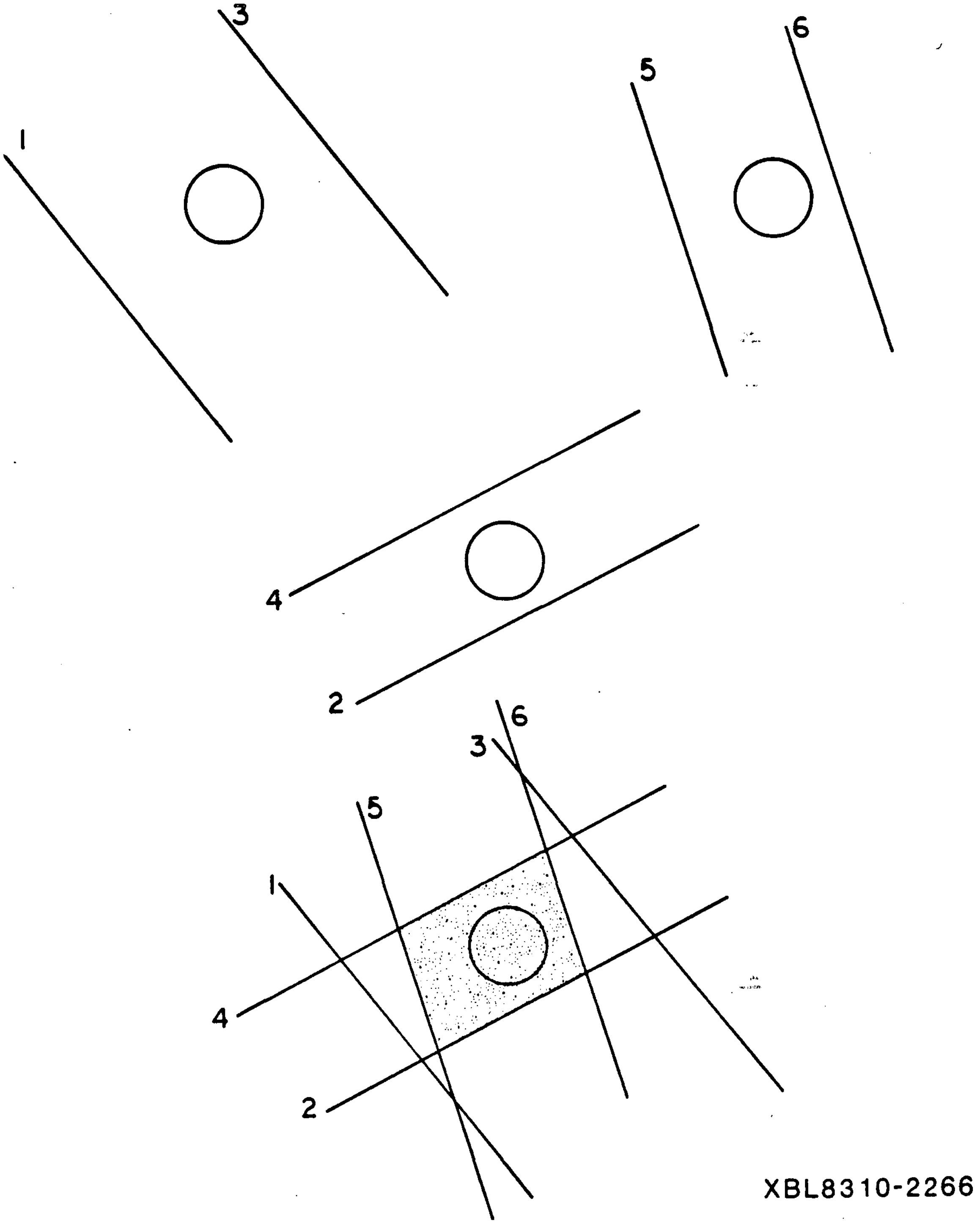


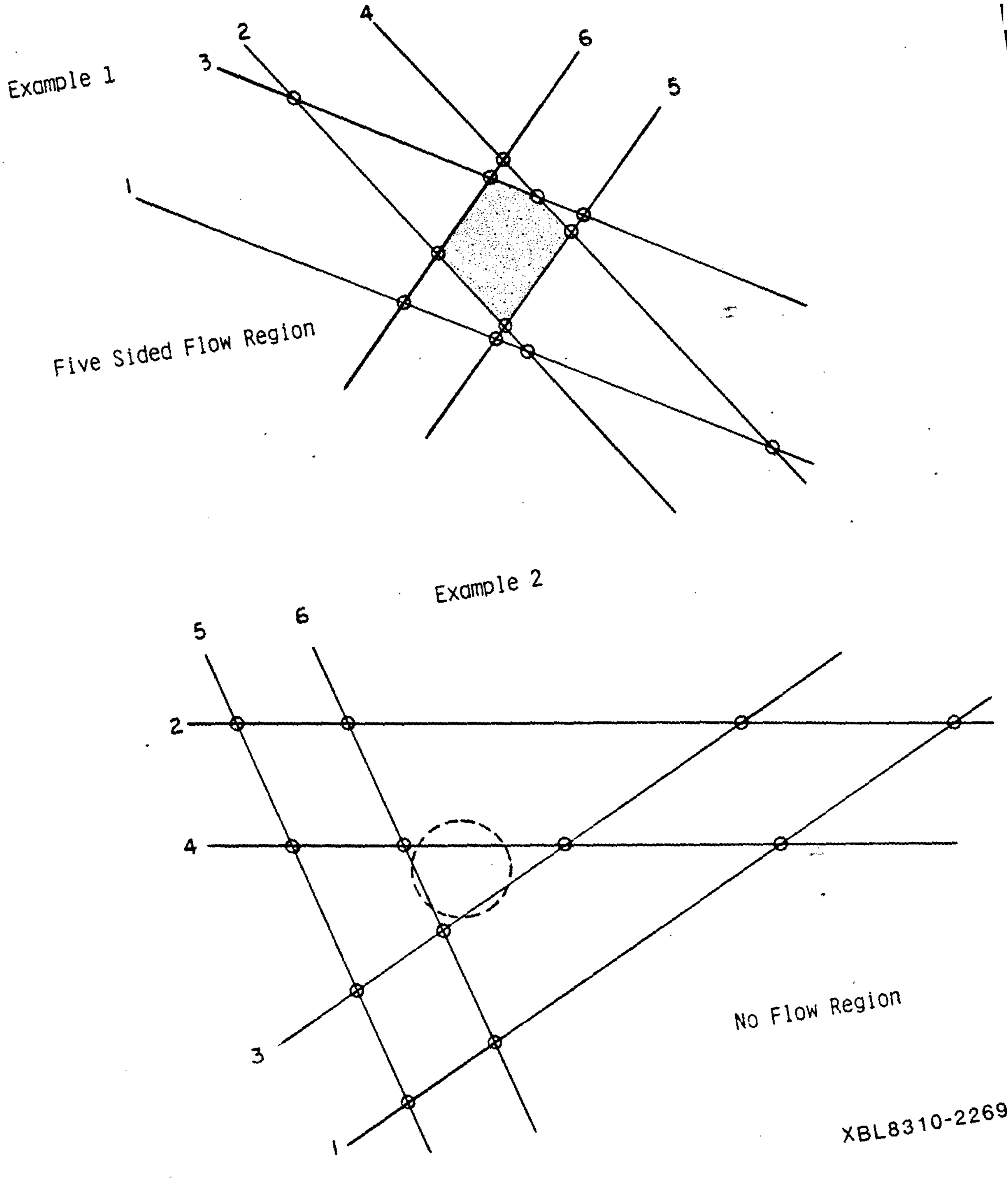
Figure 3-6. The fracture circle is inside all three pairs of parallel boundary lines, therefore, sav e the fracture, without truncation

*Flow region in fracture plane.* Although a fracture flagged for truncation intersects one or more boundary lines, it may not be inside the flow region. Therefore, the locus of the flow region in the plane of the fracture must be determined.

If the fracture plane intersects the three-dimensional flow region, the lines of the intersection will form a polygon with three, four, five, or six sides. This polygon, which in general is irregular, is the flow region in the fracture plane, and the endpoints of the line segments delimiting it are determined by subroutine BNDINT. If a fracture has been flagged for truncation, it will be discarded if the fracture plane does not intersect the flow region. If there is a flow region in the plane of the fracture, it must be determined whether or not the fracture circle is within that region.

If the fracture plane is parallel to, and in between, a pair of boundary planes, the flow region in that plane exists and is a rectangle. Using Equation 3-21, the equations of the boundary lines in the fracture plane, and simultaneously solving pairs of non-parallel sides, the endpoints of the sides of the rectangle are determined.

If there are six boundary lines in the fracture plane, they will, in general, intersect at 12 points; however, the fracture plane may intersect the flow region in such a way that two pairs of boundary lines are parallel, yielding only eight points on intersection. In either case, it must be determined whether there is an area which lies within all three pairs of boundary lines. If this common area does not exist, there is no flow region in the plane of this fracture and the fracture is flagged for discarding. If there are 12 points of intersection, there are four points on each of the six boundary lines. These four points delimit two line segments corresponding to intersections with two sets of parallel lines. If there is a common segment between these two segments, it forms a side of the flow region. If not, the boundary line in question is not part of the flow region. Figure 3-7 diagrams two such cases. In Example 1, on side 1 there is no common segment between the segment defined by intersections with sides 2 and 4 and the segment defined by intersections with sides 5 and 6. On side 2 the segment formed by intersections with



XBL8310-2269

Figure 3-7. Two examples of flow region boundary lines intersecting at 12 points

sides 5 and 6 is entirely within the segment formed by intersections with sides 1 and 3, thereby forming a side of the flow region. For side 3 the common segment is between the intersections with sides 6 and 4, and so on. In this case there are five sides to the flow region in the fracture plane. In Example 2, although the fracture was flagged for truncation because the circle is partially inside all three pairs of parallel sides, there is no area common to all three pairs of sides, i.e., no flow region. Therefore this fracture is flagged for discarding. Figure 3-8 diagrams two cases with eight points of intersection, four on each of the two non-parallel lines. Again, in Example 1 a flow region is formed and in Example 2 there is no region common to all three pairs of parallel sides.

Subroutine BNDINT determines whether there are 8 or 12 points of intersection between boundary lines and uses subroutine SIMULL to determine the coordinates of those points. The coordinates of each intersection are stored [bline(12,2)] and keyed [keybl(6,6)] to identify which intersections are on which lines. Subroutine BNDSEG is called to determine which points, if any, define a flow region. If there are four points of intersection on a boundary line, subroutine COMSEG is called to compare the relative position of the two line segments and determine whether there is a common segment. The key array [keybl] is adjusted to identify those lines that form sides of the flow region and to identify those points of intersection [bline] that are endpoints of the flow region sides.

*Boundary nodes.* A boundary node is formed wherever a fracture circle intersects the flow region boundary. Given a fracture that intersects one or more boundary lines and three or more line segments that define a flow region in the fracture plane, subroutine BNDNOD determines which intersections, or portions of intersections form boundary nodes. For each boundary line, j, intersected by fracture f, subroutine SIMULC simultaneously solves the equations of the fracture circle and the boundary line,

$$\begin{aligned} (x' - k)^2 + y'^{\,2} &= r_f^2 \\ a_j' x' + b_j' y' - c_j' d + d_j &= 0, \end{aligned} \tag{3-23}$$

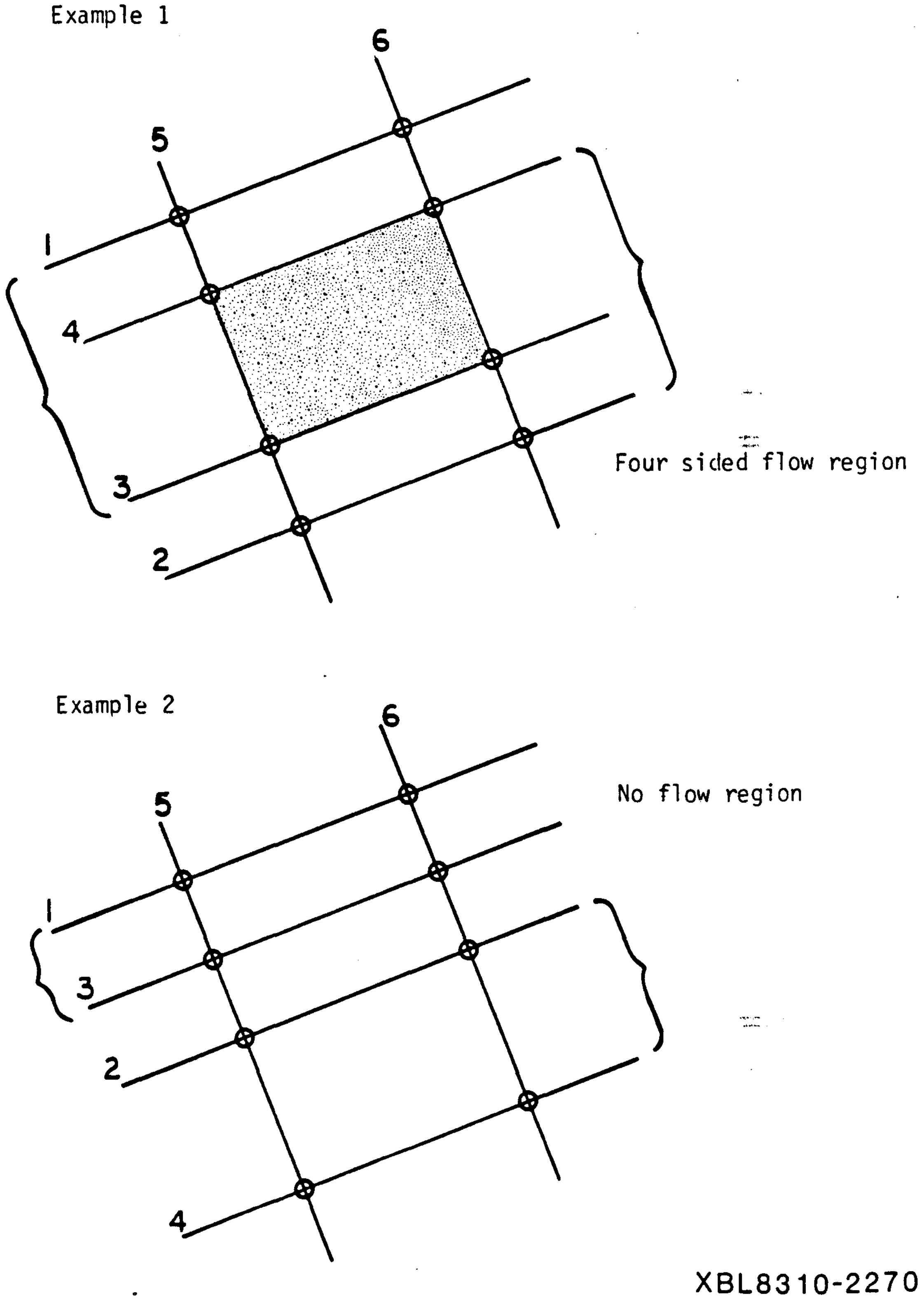


Figure 3-8. Two examples of flow region boundary lines intersecting at eight points.

to get the points or point of intersection. The line segment defined by these two points (or one point) is then compared with the line segment on line j that forms a side of the flow region. Subroutine COMSEG is again used to compare the two line segments and determine any common segment between them. The common segment is the boundary node. If there is no common segment, there is no boundary node on line j. If boundary line j does not form a side of the flow region, any intersection with line j is necessarily outside the flow region and can be ignored. All boundary node information is stored: coordinates of endpoints [binte(6,maxint,)] identification numbers of the fracture and the flow region side [ibinte(2,maxint)] side codes [ibside(2,maxint)] and a key to associate a given fracture with boundary nodes on that fracture, [ibkey(2,maxfrc)].

Once it has been established that a fracture has a boundary node, it is permuted with the first non boundary fracture in the list of fractures, so that after TRUNCF all the boundary fractures will be at the top of the list. In fact, the fracture characteristics array [frac(12,maxfrc)] is not modified, but instead a reference array [iref(maxfrc)] is set up so that whenever fracture number [ifrac] is studied, its characteristics are taken from row number [ifold] in [frac] with [ifold] = [iref(ifrac)].

Figure 3-9 illustrates the procedure for determining boundary nodes. In Example 1, the fracture intersects sides 3 and 6. The intersection with side 3 is outside the flow region so it isn't a boundary node. The intersection with side 6 is entirely within the flow region so this segment will be stored as a boundary node. In Example 2, the fracture intersects sides 1, 2 and 5. Only a portion of the intersection with side 1 is within the flow region boundary; this portion forms a boundary node. Similarly for side 2, a portion of the intersection forms a boundary node. The intersection with side 5 is ignored because side 5 is outside the flow region in this plane.

Figure 3-10 illustrates the third reason for discarding a fracture. The fracture was flagged for truncation by subroutine FLAG because it is at least partially inside all three pairs of parallel boundary lines. There is a flow region in the plane so the fracture wasn't

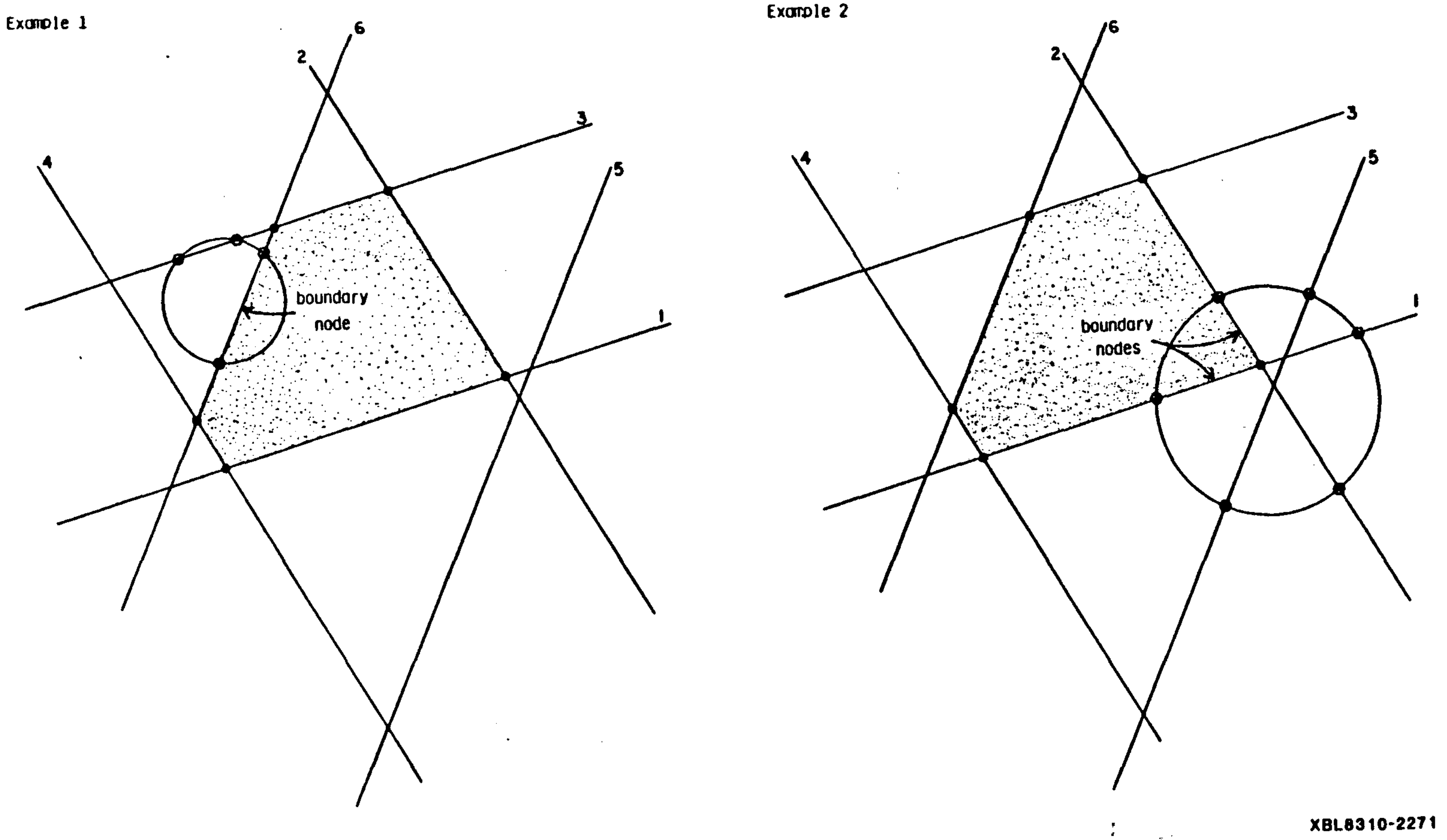


Figure 3-9. Two examples of determining boundary nodes

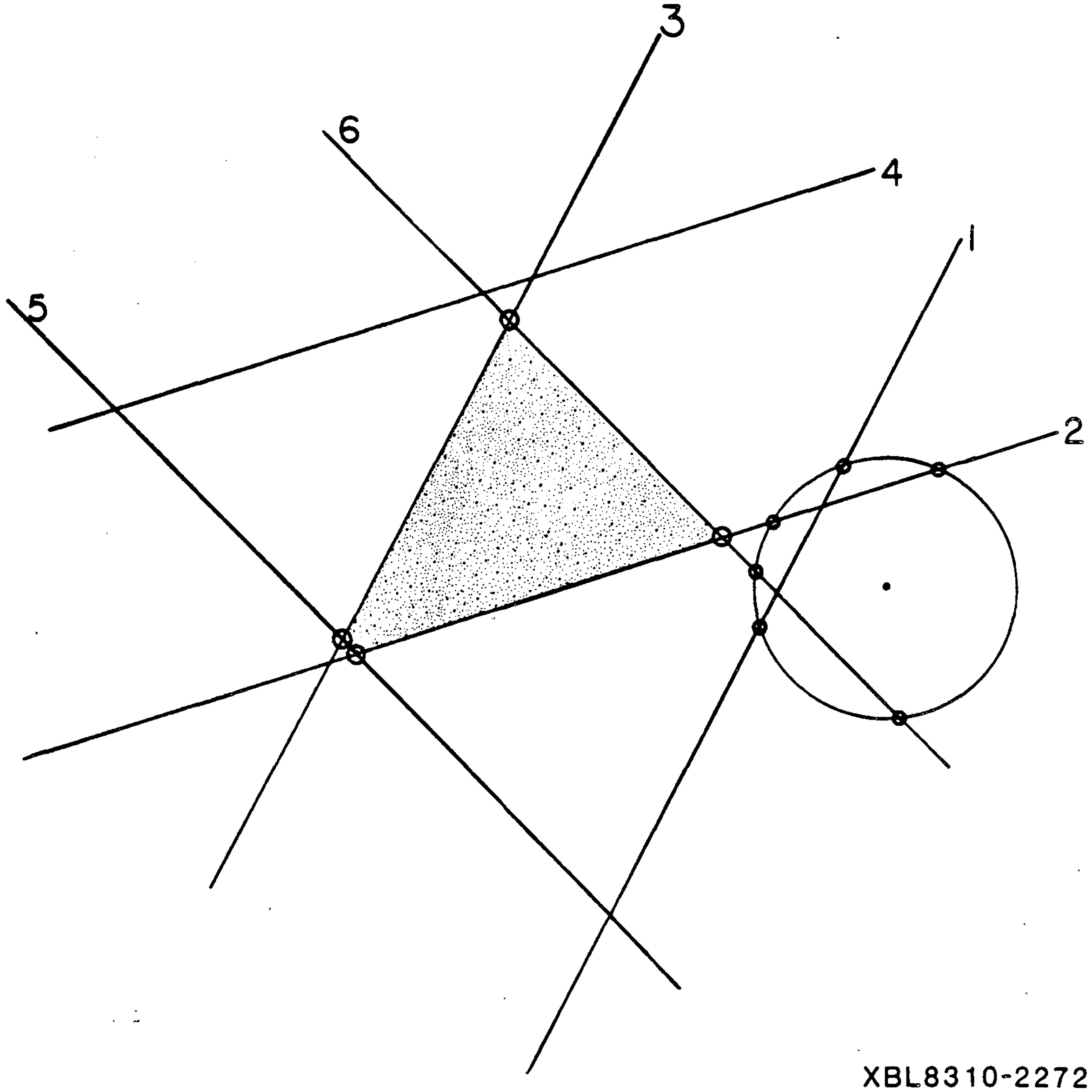


Figure 3-10. All fracture/boundary intersections are outside the flow region, therefore, the fracture is discarded.

deleted by subroutine BNDINT. However, all fracture/boundary intersections are outside the flow region; therefore the fracture is outside the flow region and is flagged for discarding.

*Truncation of fracture area at flow region boundary.* While the coordinate system is still rotated to the plane of fracture f, the area to be truncated is computed and subtracted from the total fracture area by subroutine AREAT. The area of truncation is computed separately for each boundary node and summed before subtracting from the original area. A truncation code was set for each of the six boundary lines in subroutine FLAG indicating either no intersection, [ntrunc(i) $<$ 0], intersection at only one point [ntrunc(i) $=$ 1]; or intersection at two points with the fracture center inside, outside or on the boundary line [ntrunc(i) $=$ 2, 3 or 4, respectively]. If a fracture/boundary intersection was found to be outside the flow region in subroutine BNDNOD, the code was changed to indicate no truncation on that side [ntrunc(i) $=$ 0]. If a boundary node has zero length, i.e., is a single point, then no truncation is necessary for that node [ntrunc(i) $=$ 1]. Each boundary line, i, (where i $=$ 1, 6), with a truncation code greater than 1, therefore, contains a boundary node and the area of truncation is computed for that node. Another code [nxy(6,2)], was set for each boundary node in subroutine BNDNOD to indicate whether each endpoint lies on the fracture circle or inside the circle. If an endpoint is inside the circle, it is necessarily an endpoint of another boundary node on the same fracture. Computation of the truncated fracture area is given in Appendix F.

### 3.2.3 Discarding Fractures Outside the Flow Region.

Fractures flagged for discarding [nflag(n) $=$ -1] are eliminated from the reference array by subroutine DISCAR. The number of fractures [nfrac] is changed to reflect deletions as is the number of fractures per set [iseti(1,2)].

As array [iref] is adjusted to delete references to discarded fractures, an array [inew(maxfrc)] is set up to provide back-reference.

If [ifold] $=$ [iref(ifrac)] then [ifrac] $=$ [inew(ifold)]

### 3.2.4 Statistical calculations

After eliminating all fractures outside the flow region, and truncating all those which intersect the flow region boundary, the same statistics as described in section 3-1-5 are computed and printed by subroutine PFS. Comparison of these values with those computed for the generation region indicates whether the flow region is statistically representative. When FMG3D is used to model a real site, the input parameters for the three-dimensional mesh generator are in most cases derived from one or two-dimensional data. A direct check of the validity of the diameter distribution and volumetric density derived from two-dimensional data, i.e. trace lengths and areal densities, is to compute the distribution of the trace lengths and the number of traces on each face of the cube, and to compare them with the original data. Subroutine STAT2D calculates and prints such statistics.

## 3.3. FRACTURE SYSTEM TO BE USED IN FLOW MODEL

In order to calculate flow through the system, all fracture intersections must be located. Intersections between fractures and boundary planes have already been determined. Therefore, the next step is to locate all intersections between fractures (internal nodes), and truncate those intersections that extend beyond flow region boundaries. This is done sequentially, starting with fractures intersecting the boundaries, then fractures intersecting these ones, and so on, until no new intersection is found. In this way, all the fractures from which there exists no path to any boundary, i.e. isolated clusters, are automatically discarded. Furthermore, since flow in a fracture can only occur between intersections, only those fractures containing two or more nodes, of either type or combination of types, conduct flow. Therefore, to simplify the flow problem, all fractures containing less than two nodes are identified as nonconducting and eliminated from the catalogue.

### 3.3.1 Calculation of Fracture Intersections.

The fracture mesh is built by subroutine INTERS from the boundaries to the inside of the flow region, level by level. Intersections of all the fractures intersecting the boundaries (fractures in level 1) with all the fractures, either in level 1 or not, are searched. All the fractures not in level one but which intersect a fracture in level one are put in level two, and are then screened for intersections, and so on. Fracture numbers are rearranged level by level during the process, by permuting elements of the [iref] and [inew] reference arrays. When a fracture [ifrac] is studied, only fractures with numbers bigger than [ifrac] are screened for intersections. To speed up the process, only the fractures intersecting the cubic subregions intersected by fracture [ifrac] are considered. When comparing fractures to the particular fracture [ifrac], the first step is to rotate the axes to a more convenient coordinate system, as discussed in section 3.1.3. Then, for each of the comparison fractures, it must be determined whether the fracture intersects the plane of fracture f and if so, whether that line of intersection intersects fracture f. If the fractures intersect, the point or line segment of intersection is determined. Intersections extending beyond flow region boundary lines are truncated, the truncated segments are recorded for plotting, and all intersection and boundary node endpoints are returned to the global XYZ coordinate system. Details of this procedure are given in Appendix G.

The following fracture intersection information is stored: the number of intersections [nint]; coordinates of intersection endpoints [intel (6,maxint)]; truncation side codes or zero [inside(2,maxint)]; identification numbers of intersecting fractures [inte2(2,maxint)]; an intersection/fracture catalog containing identification numbers of intersections on each fracture [inte3(maxint*2+1)]; and a key to the catalog by fracture number [inkey(maxfrc,2)]. The endpoints of the truncated intersection segments are recorded [endout(6,maxout)] as well as the numbers of the intersecting fractures [ifout(maxout,2)].

### 3.3.2 Elimination of Nonconducting Fractures.

In order for a fracture to conduct flow, it must contain at least two intersections, either with other fractures or with boundary planes. Subroutine INTFRA determines which fractures have less than two intersections, flags them for deletion [nflag(n) = -1], and eliminates them from intersection and boundary node arrays [intel, inte2, inte3, inside, inkey, binte, ibinte, ibside, ibkey]. The number of boundary nodes [ninb], and fracture intersections [nint], are also reset. This type of elimination is iterative since the removal of one fracture may result in another fracture having less than two intersections. Elimination of nonconducting fractures is hydrologically correct for flux problems and greatly reduces the size of the fluid flow analysis.

Among the intersection segments outside the flow region which were recorded by TRUNCN (see appendix G for details), the ones which lay on a fracture flagged for discarding must be deleted. Subroutine DISCOUT performs this task and also sets up the reference array [ifro(3*maxout)], keyed by [ifrok(maxcub +1)] which gives access to all the intersection segments outside the flow region which are part of fractures intersecting any given cubic subregion. Fractures flagged for discarding are eliminated by adjusting the fererence arrays [iref] and [inew] in subroutine DISCAR. The number of fractures [nfrac] is changed to reflect deletions as is the number of fractures per set [iseti(1,2)].

After elimination of nonconducting fractures, the same statistics as described in section 3.1.5 are computed and printed by subroutine PFS.

## 3.4 FLUID FLOW MODEL INPUT

The nodal arrays to be used by the fluid flow model are assembled from boundary node and fracture intersection arrays. Nodes are, optionally, divided into a given number of segments in order to verify averaging assumptions made in the flow model. The number of segments for boundary nodes [nsegb] and for fracture intersections [nseg] are read by the main program; default values are three and one, respectively.

### 3.4.1 Nodal Arrays

Subroutine NODELE takes each fracture in turn [nf=1,nfrac] and stores any boundary nodes on the fracture first, then stores any internal nodes on the fracture in the nodal arrays. The boundary node key [ibkey(nfrac,2)] is used to identify boundary nodes on the given fracture and the intersection key [inkey(nfrac,2)] is used to identify internal nodes on the fracture. Each internal intersection involves two fractures and will, therefore, be identified twice, once for each of the fractures. As in the intersection arrays, the endpoints, side codes, and fracture numbers of an internal node are only stored once, the first time it is sorted out, which is for the lower of the two fracture numbers.

Subroutine NODSEG is called for each node before it is stored. If the number of segments is greater then one, the node is divided and each segment stored as a separate node. The procedure follows:

- Let the endpoints of the node be P1 = $(x_{1,}y_{1,}z_{1})$ and P2 = $(x_{2,}y_{2,}z_{2})$ .

- To divide the line segment P1P2 into n equal segments, find n-1 points of division on the line between P1 and P2.

- Point P = (x,y,z) will divide segment P1P2 in the ratio $r_1/r_2$ if

$$x = \frac{x_1 r_2 + x_2 r_1}{r_1 + r_2},$$
$$y = \frac{y_1 r_2 + y_2 r_1}{r_1 + r_2},$$
$$z = \frac{z_1 r_2 + z_2 r_1}{r_1 + r_2}, \tag{3-24}$$

- Therefore, let $r_1/r_2$ equal 1/(n-1), 2/(n-2), 3/(n-3), ... (n-1)/1 to find the n-1 points of division, and let the resulting points become nodal endpoints. For example, if n=2 the node with endpoints P1 and P2 becomes two nodes, one with endpoints P1 and P and one with endpoints P and P2.

If the distance between the endpoints of a node is zero, i.e., the node is a point, then the node is not divided.

The nodal arrays contain: (1) coordinates of endpoints of each node in global coordinates [node1(maxint,6)]; (2) identification numbers of fractures or fracture and side that form the node [node2(maxint,2)]; (3) side codes for endpoints of boundary nodes and internal nodes truncated at boundary planes [inside(maxint,2)]; (4) a fracture/node catalog to identify all nodes on a given fracture [node3(maxint*2+1)]; and (5) the key to the catalog by fracture [nkey(nfrac,2)]. An example using five fractures is shown in Figure 3-11 and described in Table 3-2.

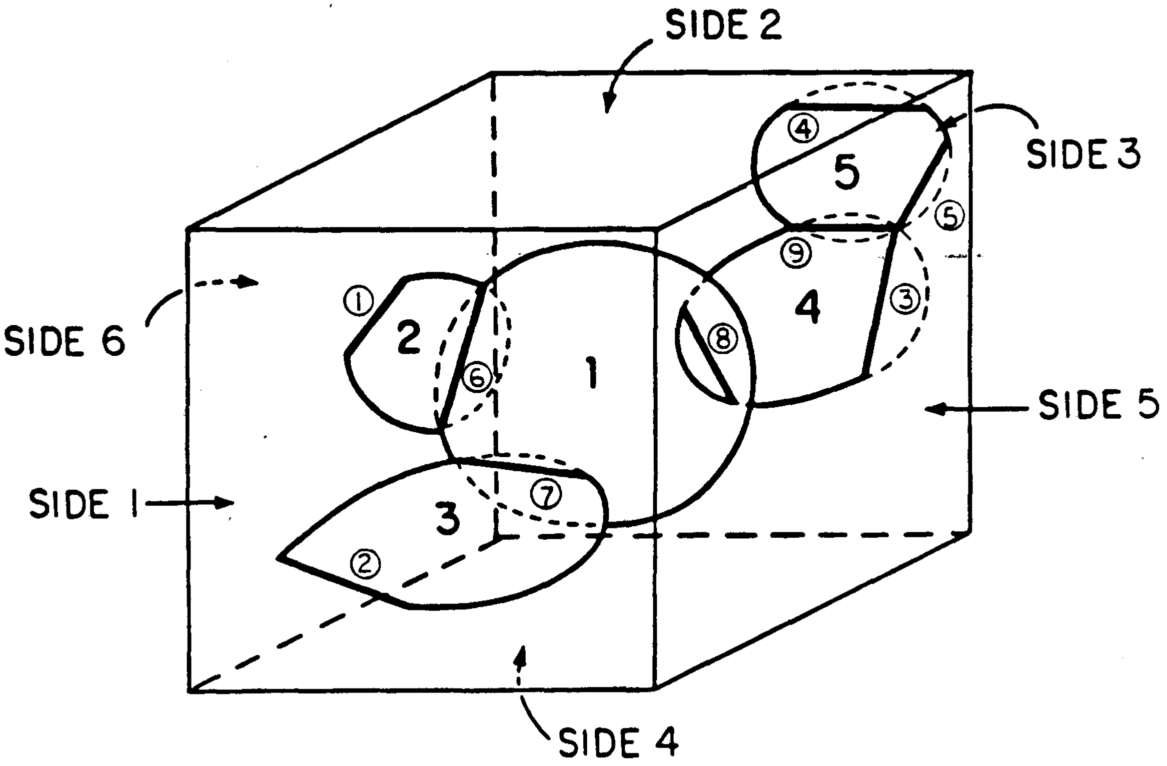
**Table 3-2**
**Description of Nodes in Figure 3-11**

| Node No. | Description of Intersection |
|---|---|
| 1 | Boundary node, intersection of fracture 2 and side 6 |
| 2 | Boundary node, intersection of fracture 3 and side 1 |
| 3 | Boundary node, intersection of fracture 4 and side 5 |
| 4 | Boundary node, intersection of fracture 5 and side 3 |
| 5 | Boundary node, intersection of fracture 5 and side 5 |
| 6 | Internal intersection of fractures 1 and 2 |
| 7 | Internal intersection of fractures 1 and 3 |
| 8 | Internal intersection of fractures 1 and 4 |
| 9 | Internal intersection of fractures 4 and 5, truncated at side 5 |

Using default values for segmenting nodes, nsegb = 3 and nseg = 1, the original nine nodes become 19 nodes and result in the nodal arrays given in Table 3-3. The array nkey is used to find the nodes on a particular fracture, e.g., fracture 4:

nkey(4,1) = 5 -- there are 5 nodes on fracture 4;

nkey(4,2) = 13 -- the identification numbers of the nodes on fracture 4 begin in location 13 of catalog array node3 and end in location



XBL 856-2712

Figure 3-11. Example of a three-dimensional fracture system with five fracture and nine nodes.

node1(19,6) Coordinates of endpoints of nodes:

$$\begin{bmatrix} x1,1 & y1,1 & z1,1 & x1,2 & y1,2 & z1,2 \\ x2,1 & y2,1 & z2,1 & x2,2 & y2,2 & z2,2 \\ x3,1 & y3,1 & z3,1 & x3,2 & y3,2 & z3,2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x19,1 & y19,1 & z19,1 & x19,2 & y19,2 & z19,2 \end{bmatrix}$$

node2(19,2)

Fracture numbers for internal nodes, fracture number and negative side number for boundary nodes:

$$\begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 2 & -6 \\ 2 & -6 \\ 2 & -6 \\ 3 & -1 \\ 3 & -1 \\ 3 & -1 \\ 4 & -5 \\ 4 & -5 \\ 4 & -5 \\ 4 & 5 \\ 5 & -3 \\ 5 & -3 \\ 5 & -3 \\ 5 & -5 \\ 5 & -5 \\ 5 & -5 \end{bmatrix}$$

nside(19,2)

Side codes for endpoints of nodes:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 6 & 6 \\ 6 & 6 \\ 6 & 6 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 5 & 5 \\ 5 & 5 \\ 5 & 5 \\ 0 & 5^{*} \\ 3 & 3 \\ 3 & 3 \\ 3 & 3 \\ 5 & 5 \\ 5 & 5 \\ 5 & 5 \end{bmatrix}$$

* Endpoint 2 of node 7 was truncated at side 5

node3(24) Fracture/node catalog:

[ 24* 1 2 3 4 5 6 1 7 8 9 2 10 11 12
3 13 14 15 16 17 18 19 13 ]

* The first location contains the number of locations actually used in array node3

nkey(5,2) Key to fracture/node catalog:

$$\begin{bmatrix} 3 & 2 \\ 4 & 5 \\ 4 & 9 \\ 5 & 13 \\ 7 & 18 \end{bmatrix}$$

Table 3-3. Nodal Arrays from Example in Figure 3-11

nkey(4,2) + nkey(4,1) - 1 = 13 + 5 - 1 = 17 ;

node3(i), i = 13 to 17, contain node numbers 10, 11, 12, 3, and 13 -- the nodes on fracture 4.

Using the same five fractures example but letting the number of segments for internal nodes [nseg] equal two, while the number of segments for boundary nodes [nsegb] remains three, will result in the generation of different nodal arrays, as given in Table 3-4.

### 3.4.2 Input to Fluid Flow Model

Subroutine WNODES writes a data file to be used as an input file by the fluid flow model. For each flow region WNODES writes the following data:

- Run identification [iray, idate, title(40)]
- Flow region parameters [xmesh, ymesh, zmesh, rophi, rothe] and coefficients of flow region boundary plane equations [frcoef(6,4)]
- The total number of fractures [nfrac], fracture characteristics including orientation angles, radius, aperture, center (xc,yc,zc), area, and coefficients of fracture plane equations [frac(12,nfrac)].
- The total number of nodes [nnode], nodal segmentation parameters [nseg, nsegb], and all nodal arrays (described in section 3.4.1) [nodel(nnode,6), node2(nnode,2), nside(nnode,2), node3(node3(1)), nkey(nfrac,2)].

### 3.4.3 Input to the plotting program

Subroutine WDIMES writes files to be used as input files by the plotting program. One file is written for each different flow region, and they are named DIMES01.DAT, DIMES02.DAT, etc... The following data is written on each file:

- Run title [title(40)]

node1(23,6) Coordinates of endpoints of nodes:

$$\begin{bmatrix} x1,1 & y1,1 & z1,1 & x1,2 & y1,2 & z1,2 \\ x2,1 & y2,1 & z2,1 & x2,2 & y2,2 & z2,2 \\ x3,1 & y3,1 & z3,1 & x3,2 & y3,2 & z3,2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x23,1 & y23,1 & z23,1 & x23,2 & y23,2 & z23,2 \end{bmatrix}$$

node2(23,2)

Fracture numbers for internal nodes, fracture number and negative side number for boundary nodes:

$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 1 & 4 \\ 1 & 4 \\ 2 & -6 \\ 2 & -6 \\ 2 & -6 \\ 3 & -1 \\ 3 & -1 \\ 3 & -1 \\ 4 & -5 \\ 4 & -5 \\ 4 & -5 \\ 4 & 5 \\ 4 & 5 \\ 5 & -3 \\ 5 & -3 \\ 5 & -3 \\ 5 & -5 \\ 5 & -5 \\ 5 & -5 \end{bmatrix}$$

nside(23,2)

Side codes for endpoints of nodes:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 6 & 6 \\ 6 & 6 \\ 6 & 6 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 5 & 5 \\ 5 & 5 \\ 5 & 5 \\ 0 & 0^* \\ 0 & 5^* \\ 3 & 3 \\ 3 & 3 \\ 3 & 3 \\ 5 & 5 \\ 5 & 5 \\ 5 & 5 \end{bmatrix}$$

* Endpoint 2 of original node 7 was truncated at side 5; node 7 is now nodes 16 and 17, therefore node 16 is not truncated, but endpoint 2 of node 17 is truncated at side 5.

node3(32) Fracture/node catalog:

```
[ 32   1   2   3   4   5   6   7   8   9   1   2  10  11  12
   3   4  13  14  15  16  17   5   6  18  19  20  21  22  23
   6   7 ]
```

nkey(5,2) Key to fracture/node catalog:

$$\begin{bmatrix} 6 & 2 \\ 5 & 8 \\ 5 & 13 \\ 7 & 18 \\ 8 & 25 \end{bmatrix}$$

Table 3-4. Nodal Arrays from Example in Figure 3-11 when each internal node is divided into two segments and each boundary node is divided into three segments

- Generation and flow region parameters [xgene, ygene, zgene, xmesh, ymesh, zmesh,rophi, rothe]
- Total number of fractures [nfrac], of intersections [inst+nint], of intersection segments outside the flow region [nout]
- Parameter [iord] for division into cubes (subregions)
- Maximum aperture and radius [apmax, radmax]
- Number of fractures, of intersections in each cube [numfrc (ncub), numinc (ncub)] and key [ifrok(ncub+1)] for array [ifro].

Then, for each cube:

- Number of the cube [ic]
- Number and characteristics of each fracture intersecting the cube [ifrac, frac (7, iref (ifrac))].
- Number, endpoints coordinates and fracture numbers of each intersection segment outside the flow region on a fracture intersecting cube [ic], [iout, endout [6, iout], inew (ifout (iout,2)).
- The same information for each boundary and internal intersection on a fracture intersecting cube [ic] : [k, intel(6,k), inew (inte2(2,k))].

Writing the information cube by cube makes the file longer, since the information about one fracture and its intersections is repeated for each cube the fracture intersects. But it saves a lot of time and memory space when plotting, since only the information on the cubes specified by the user has to be read.

## 4.0 PLOTTING PROGRAM FOR FMG3D: DIMES

In order to use the three dimensional fracture mesh generator FMG3D, able to produce and analyze complex fracture meshes made of disc-shaped fractures randomly distributed in a given volume, we also need a means to represent the meshes generated. Visual inspection is an incomparably more efficient way to check the output of the generation procedure than scrolling through very long numerical files. For example, checking that the program has found the right intersections between fractures is easy on a plot of both fractures and intersections, and quite difficult without a plot.

Existing software does not have the versatility needed to plot a number of fractures and perform the hiding procedures required to make the plot intelligible. Therefore, DIMES (DIscs MEShes) was designed.

The program, was designed to:

- give a reasonably accurate plot of the discs, allowing for any viewpoint;
- keep the computing time within manageable limits;
- provide enough versatility to be installed on any plotter;
- show the relative position of the fractures and the flow and generation regions;
- be able to select a subset of the fractures in the flow region, both for reducing computing time and for seeing 'inside' the fracture mesh without being blinded by big fractures in the fore-front.

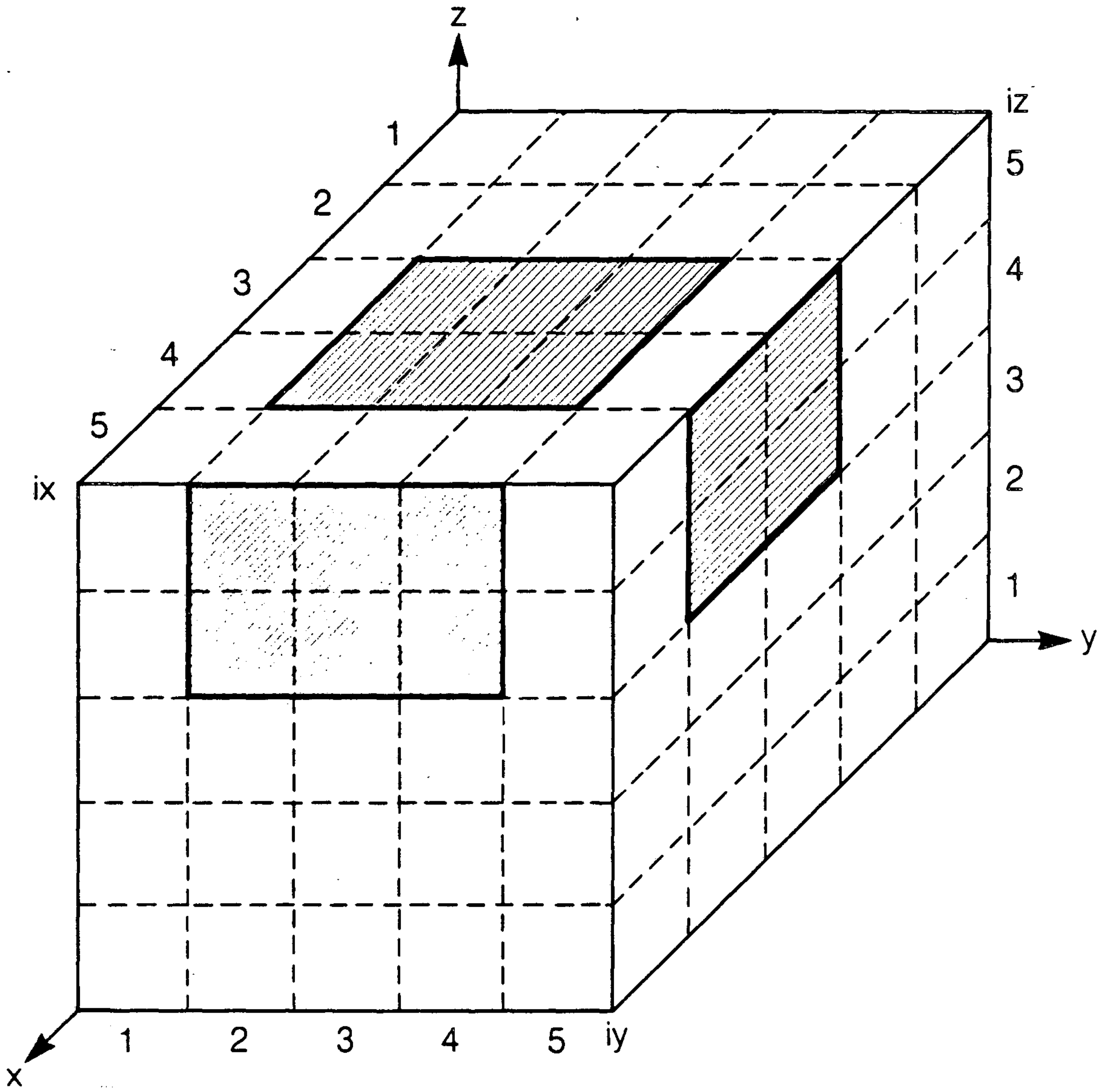
## 4.1 PROGRAM CHARACTERISTICS

The program is potentially able to handle data from any code generating circular fractures, by simply changing the input orders, apart from one feature which is specifically linked with FMG3D: the ability to peek into any part of the region under study and represent fractures otherwise hidden by the fore-front. This feature is directly derived from the way FMG3D performs its search for intersections between fractures. The generation region is divided into equal volume cubes, to which the fractures are assigned according to their position. A fracture will be stored in a given cube if the cube contains all or part of it. In this way, a fracture can be assigned to several different cubes. This enables FMG3D, each time it looks at a fracture, to screen only the fractures which are in the same cubes, thus greatly reducing the number of necessary checks. In the output of FMG3D used as input data for DIMES, the fractures are ordered cube by cube. The program then asks the user to specify a set of cubes (from one cube to the whole generation region, see Figure 4-1.) and plots all the fractures lying at least partially in this set of cubes, or subregion.

We chose the orthographic projection method, which is a projection normal to the plane of the plot, because it is very easy to implement. Once the viewpoint has been chosen, the coordinates of all the points in the plot are rotated accordingly so that the x-y plane corresponds to the plane of the plot. When plotting, the z coordinate is then simply dropped.

In order to give a good idea of orientation and aperture the fractures are plotted as two parallel discs separated by a distance proportional to the aperture. Each disc is approximated by a regular polygon, the number of segments in one polygon being left as a choice for the user. The lines which represent the intersections between fractures are then added. We must keep track of the fractures that form the intersection so that a fracture disc never hides one of its own intersections. An intersection is plotted as a plain line if it is inside the flow region, and as a dotted line if it is outside the flow region.

The core of this program is two nested do loops, the exterior one over the discs and the interior one over all the line segments associated with a disc which are either parts of an intersection or parts of the disc itself. Segments are eliminated from this loop as they are found to be hidden. DIMES checks to see if a segment must be hidden by the disc, and if yes, either deletes the old segment or calculates the coordinates of the endpoints of the new shorter segment. Creating a second new segment is necessary if the old seg-



XBL 869-11037

Figure 4-1. Example of subregion definition. Shaded areas are the projections of subregion boundaries, with iord = 5, ix1 = 3, ix2 = 4, iy1 = 2, iy2 = 4, iz1 = 4, iz2 = 5

ment is hidden only in its middle. This process takes the greatest proportion of the computer time needed by the program, because of the high number of segments involved in even comparably simple sets. For example, 50 fractures divided into two discs of 40 segments each and having 100 intersections make up a set of 4100 segments, at the first iteration of the discs loop.

In order to reduce the time involved in the hiding process outlined above, the discs are sorted with respect to the volume they hide behind them in descending order. Looping over the discs in this order results in hiding more segments in the early iterations, thus saving a number of later comparisons.

Because we did not want to bind ourselves to any plotting device or driver, we resolved to use only very simple orders from the plotting software, namely 'draw a segment' or 'write a line of text,' apart from the initialization and end-of-plot orders. Also, all the plotting orders are grouped into one subroutine, thus making easy an adaption to any other plotting interface. The plotting software currently in use is DISPLA, by ISSCO corporation.

From information on the size and orientation of the generation and flow regions, and given the viewpoint angles, DIMES draws the flow region into the main plot area, showing its edges only when they are not hidden behind a fracture. In addition, to visualize both the viewpoint and the orientation and size of the flow region, a small "orientation box" is drawn in the lower left corner of the plot, showing the relative size and orientation of the flow region and the cube encompassing the generation region from the same viewpoint as the main plot. Recall that the generation region is created in global coordinates and the flow region is selected inside the generation region. The flow region may be rotated with respect to the generation region. The plot created by DIMES has a viewpoint which amounts to rotating the generation region and flow region by the same amount. The coordinate axes shown in the "orientation box" pass through lowermost edges of the generation and flow regions respectively towards increasing coordinates.

The x,y, and z axes can then be differentiated by knowing first that they form a right hand coordinate system. Due to the order of rotations used to determine the point of view, the z-axis of the generation region is always vertical in the plot plane. Also, the y axis of the rotated flow region is always horizontal, due to the nature of the two rotations that can be performed by FMG3D.

## 4.2 VIEW ANGLES DEFINITION AND ROTATION MATRIX.

The view angles $\phi_v$ and $\theta_v$ [viewphi, viewthe] are input from the user and are defined as shown in Figure 4-2, $\phi_v$ being the angle between the x-y plane and the viewing position, and $\theta_v$ being the angle with the x-axis of its projection to the x-y plane. The rotation matrix [viewmat] is the combination of a $\theta_v$ rotation about the z axis and then a (90° - $\theta_v$) rotation about the $y'$ axis obtained from the first rotation. Setting $\phi_A$ [viewang] = 90° -$\phi_v$, the matrix is:

$$[\text{viewmat}] = \begin{bmatrix} \cos\phi_A \cos\theta_v & \cos\phi_A \sin\theta_v & -\sin\theta_A \\ -\sin\theta_v & \cos\theta_v & o \\ \sin\phi_A \cos\theta_v & \sin\phi_A \sin\theta_v & \cos\phi_A \end{bmatrix} \qquad (4\text{-}1)$$

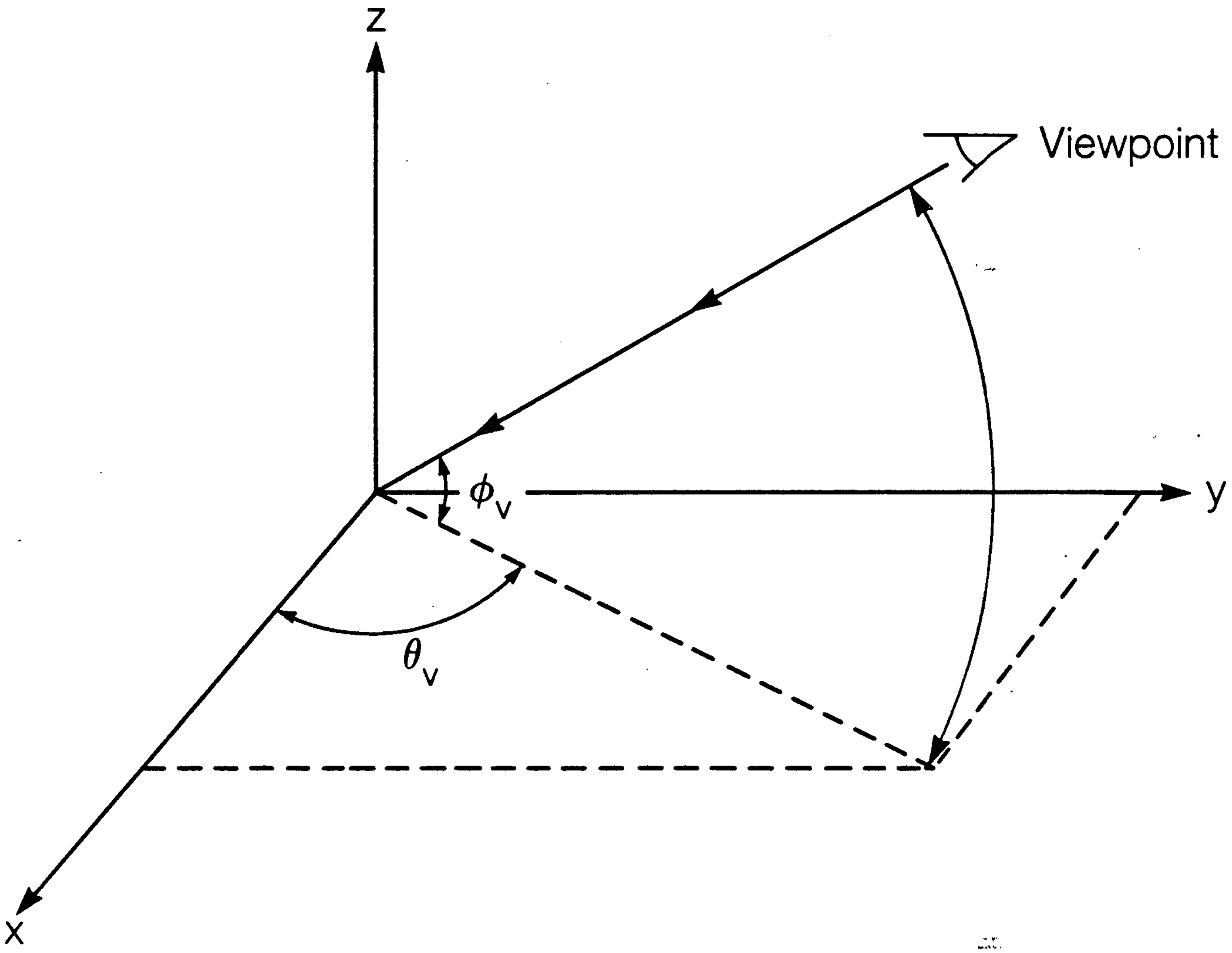
Subroutine ROTMAT computes this rotation matrix.

## 4.3 INITIAL CALCULATIONS

Scaling factors for the main plot and the "orientation box" plot are computed. Fracture apertures are rescaled to make them visible on the plot. The "orientation box" plot is computed. These operations are performed by subroutine INIPLOT. The maximum distance between a point on a fracture and the center of the plot is computed:

$$[\text{dmax}] = 1/2\sqrt{[\text{xmesh}]^2 + [\text{ymesh}]^2 + [\text{zmesh}]^2} + [\text{radmax}] \qquad (4\text{-}2)$$

where [xmesh, ymesh, zmesh] are the dimensions of the flow region and [radmax] is the maximum fracture radius (data). [dmax] will be used to chose the proper scale for the plot. Similarly, a scaling factor [dgene] is computed for the small "orientation box" plot (section 4-1):



XBL 869-11038

Figure 4-2 Definition of view angles $\phi_v$ and $\theta_v$

$$[\text{dgene}] = 1/2\ \sqrt{[\text{xgene}]^2 + [\text{ygene}]^2 + [\text{zgene}]^2} + [\text{radmax}] \qquad (4\text{-}3)$$

where [xgene, ygene, zgene] are the size of the generation region. The fracture apertures aper [frac(4,ifrac)] are rescaled linearly so that disc thickness is visible on the plot:

$$\text{aper} = \text{aper}^*[\text{thicmax–thicmin}]/[\text{apmax}] + [\text{thicmin}] \qquad (4\text{-}4)$$

where [thicmax, thicmin] are the maximum and minimum thickness on the plot, chosen for visibility and [apmax] is the maximum fracture aperture (read from data).

The flow and generation region for the "orientation box" as well as the flow region for the mainplot are then computed. The main plot flow region segments are first computed, with no rotation. A quantity [radmax] is added to three edges of the parallelepiped in order to visualize the frame of reference. Using the appropriate scaling factors, [xgene/xmesh], [ygene/ymesh] and [zgene/zmesh], the generation region is deduced from the flow region at the scale of the main plot. It is rotated according to the view angle, scaled down and translated to the "orientation box" in the lower left corner of the plot by subroutine BOXES. The resulting segments are stored in [xbox (24,2), ybox (24,2)]. The two flow region plots are computed. The appropriate rotation matrix [romat] is computed by subroutine ROTMAT from the flow region rotation angles $\phi_R$ and $\theta_R$ [rophi, rothe] read from the data file, and defined in the same manner as in FMG3D (section 3.2.1, figure 3.5). The flow region segments are then rotated according to [romat] and stored in the segments arrays to be processed for hiding [end (6, maxseg), infint (2, maxseg)]. Subroutine BOXES is finally called again to rotate the flow region segments according to the view angle, scale them down, translate them to the "orientation box" and store them in [xbox (24, 2), ybox (24,2)].

## 4.4 SET UP OF THE SEGMENTS ARRAYS AND REORDERING OF THE FRACTURES

Before the hiding processing can take place (i.e. compare any disc with any line segment of the main plot and erase the line segment if necessary), all the line segments describing the intersections, the intersections outside the flow region, the flow region, and the fracture discs must be properly rotated and incorporated in the segment arrays, [end (6, maxseg)] containing the endpoints of each segment and [ifint (2, maxseg)] containing either zero or if relevant the number(s) of the fracture(s) of which a segment is part.

The intersections, either inside or outside the flow region, are read directly into these arrays. The flow region segments are stored in them by subroutine INIPLOT (section 4-3). Subroutine ROTATE is then called to rotate all the segments already stored according to the view point.

Next each fracture is rotated according to the viewpoint: the new unit normal to the fracture $(\lambda,\mu,\upsilon)$ [xnorm, ynorm, znorm] and the sines and cosines of the new fracture orientation angles $\phi_n$ and $\theta_n$ are computed. These angles are defined as in FMG3D (section 3.1.1 and figure 3.1). This is performed by subroutine ROTFRAC. The new fracture center $(x_c, y_c, z_c)$ is computed. The fracture is divided into two discs of same radius R and orientation $(\phi_n, \theta_n)$, with different center coordinates:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \text{ and } \begin{pmatrix} x_c - \sigma \text{ aper } \lambda \\ y_c - \sigma \text{ aper } \mu \\ z_c - \sigma \text{ aper } \upsilon \end{pmatrix}$$

where $\sigma$ is chosen between 1 and -1 so that $\sigma$ aper $\upsilon$ is positive. The two discs are then divided into np [npol] segments each. The coordinates of the endpoints of these segments are computed:

$$\begin{aligned} x_i &= x_c + R \cos\phi_n \cos\theta_n \cos(\Delta i) + R \sin\theta_n \sin(\Delta i) \\ y_i &= y_c - R \cos\phi_n \sin\theta_n \cos(\Delta i) + R \cos\theta_n \sin(\Delta i) \\ z_i &= z_c - R \sin\phi_n \cos(\Delta i) \end{aligned} \qquad (4\text{-}4)$$

where $(x_c, y_c, z_c)$ take the values of both discs centers

$\Delta = 2\pi/\text{np}$

This gives 2np segments, which are recorded in the segments arrays.

Next, the fractures are sorted according to $z_c$, $|\cos \phi_n|$, and R. The chances of a fracture hiding segments increase when any of these parameters increases. By putting in first the fractures likely to hide a lot of segments, we decrease the time needed for later iterations. The parameter combination used for ordering is:

$$\alpha_{ord} = (z_c + \mathrm{dmax}) \; |\cos \theta_n| \, R^2 \tag{4-5}$$

$\alpha_{ord}$ is approximately proportional to the volume hidden by the fracture. During the sorting, it is stored in place of the aperture [frac (4, ifr)] which is not used any more. In fact, the fracture characteristics arrays [frac(7,maxfr), trig (4,maxfr)] are not changed, but an index array [index(maxfr)] is set up.

### 4.5 HIDING

Looping over the fracture discs d [ifrac], the program compares each of them with each line segment s [iseg] and deletes s or part of it if necessary.

If segment s is part of disc d or of an intersection on the fracture represented by disc d, it should not be checked for hiding because a disc could end up hiding part of itself or of its intersections due to the limitations of computer precision. This check is the first made by the program.

When a fracture disc is projected on the x-y plane, the resulting figure is an ellipse. From here on, the term "ellipse" will be used for the projection of the disc on the plotting plane, and the term "disc" will be used for the actual "3-D" disc. In the same manner, "plot segment" will be used for the projection on the x-y plane of a segment s defined in 3-D. The endpoints of segment s have coordinates $(x_{1,} \, y_{1,} \, z_{1},)$ and $(x_2, y_2, z_2)$.

Before computing the exact relative position of the ellipse and the plot segment, two more checks are performed: first on the distance between the center of the ellipse and the center of the plot segment, and second on the intersection between the segment and the disc plane. The distance between the center of the ellipse and the center of the plot segment is compared with the sum of the disc radius and half of the plot segment length. If the latter is bigger, no hiding can occur and the next segment is considered. This check is not strictly necessary but serves to eliminate most segments before beginning the more complex computations. The relative position of the segment and the disc plane is checked, using the equation of the fracture plane as obtained in Appendix A, Equations A-2 and A-3. If the segment is above the plane, the next segment is considered. If the segment is under the plane, the program goes to the next step in the hiding process. If the segment intersects the plane, it is divided in two segments: one above the plane, which is stored, and one under the plane, which becomes the current segment.

The segments considered at this stage are under the disc plane and the plot segments are close to the ellipse. The intersections betwen the plot-segment line and the ellipse are computed. If there is one or no intersection, nothing is hidden and the next segment is considered. If there are two intersections, their position on the line is compared with the plot segment endpoints. If the segment and the ellipse have a common part, four cases can occur (Figure 4-3):

- two endpoints hidden (Figure 4-3a), the segment is deleted;
- first endpoint hidden (Figure 4-3b), the first endpoint is adjusted to the second ellipse-line intersection;
- second endpoint hidden (Figure 4-3c), the second endpoint is adjusted to the first ellipse-line intersection;
- middle of segment hidden (Figure 4-3d) the second endpoint of the segment is adjusted to the first intersection and a new segment is created, with endpoints (second intersection, second endpoint of original segment).

During all the hiding process, track is kept of the position of deleted segments in the arrays, so that whenever a segment is created, it is stored in a void position of the arrays instead of being added at their end.

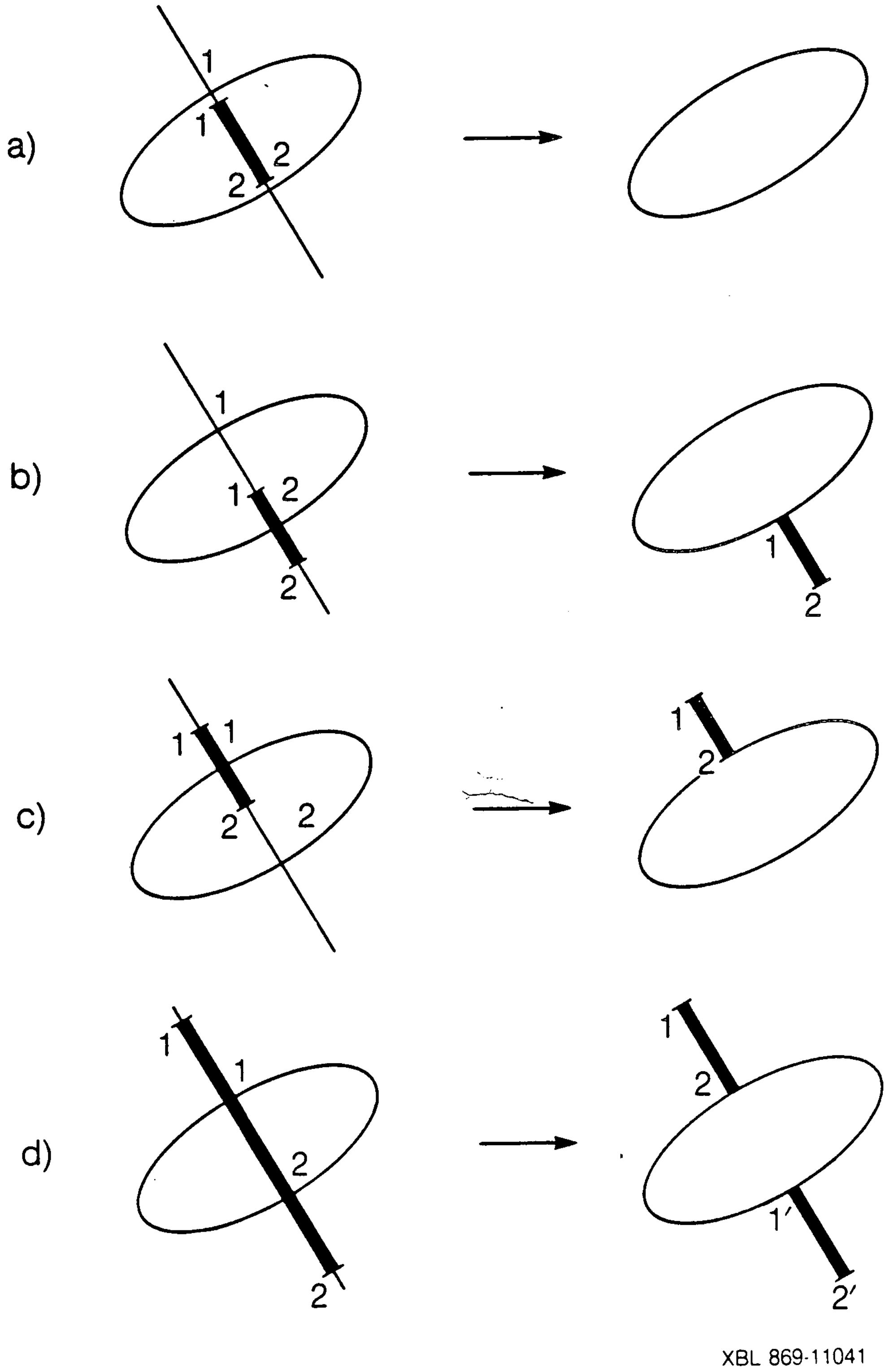


Figure 4-3. Relative position of intersecting ellipse and segment

## 4.6 DRAWING

The drawing is performed by subroutine DRAW, using the graphic software package DISPLA. To ensure portability of the program, only simple orders from DISPLA are used: initialization and end of plot, drawing of a frame around the plot, printing of a title, drawing of a line segment, either plain or dotted.

## 5.0 EXAMPLES AND TEST CASES

This section describes examples that were among those used to verify and validate the computations performed by the flow model DISCEL and the plotting program DIMES. The first four examples are illustrated in Figure 5-1. In each case, the left hand side illustrates the problem solved by the three-dimensional flow program. On the right are shown the corresponding analytical problems used for comparison. The fifth case is a small random case used to compare various nodal discretization schemes; no analytical solution is available. The results presented below are dimensionless for consistent units. Examples of plots are given at the end of this section.

## 5.1 VERIFICATION OF HEAD AND FLUX CALCULATIONS

### 5.1.1 Case 1: One Fracture Intersecting Two Parallel Boundary Planes

This is the simplest case possible, the fracture plane is intersected at right angles by two parallel boundary planes. If the boundary planes are centered and closely spaced relative to the radius of the fracture, the flux in the fracture is approximately equal to the linear flow in a rectangular slab with two constant head boundaries and thickness b, the fracture aperture. Figure 4-1a illustrates both the numerical and the analytical problems that were solved. In this case we were only checking the calculation of flux because both intersections are boundary nodes with assigned heads. The parameters used for this case were as follows:

$H_1$ = 1.0, (inflow face head),

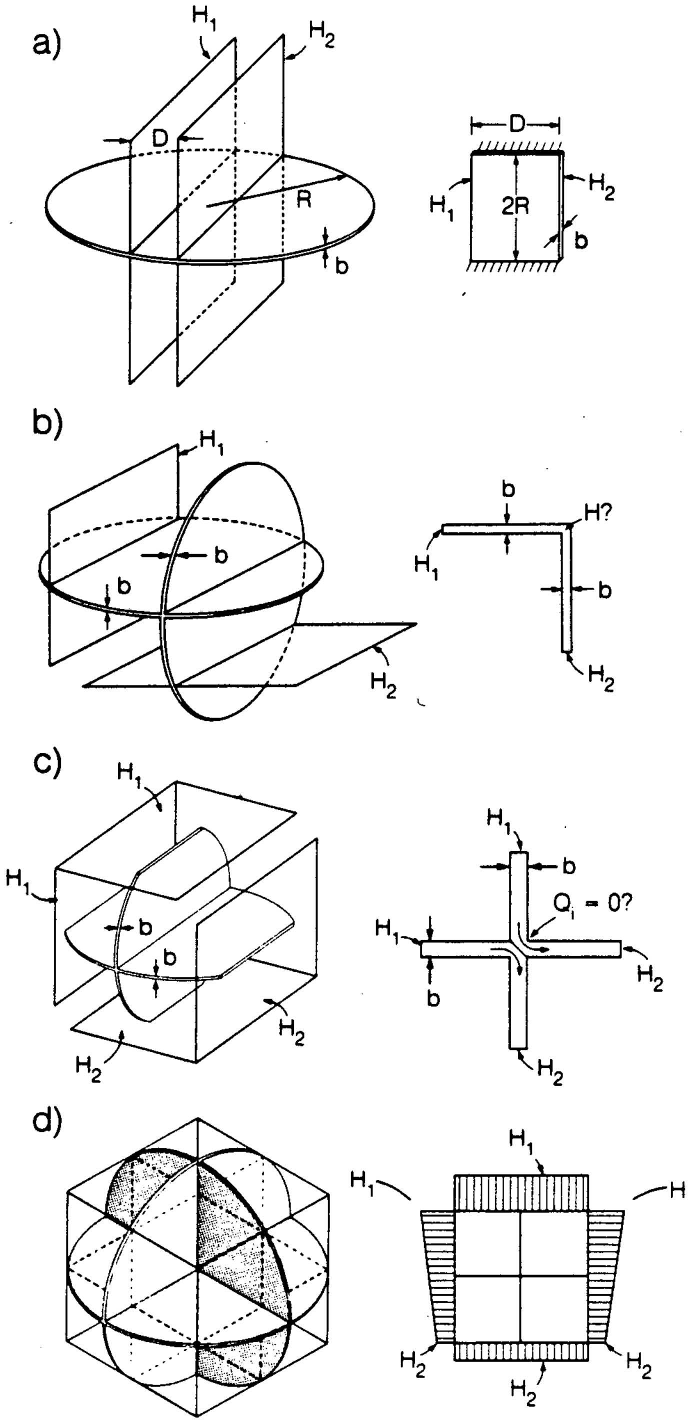
$H_2$ = 0.0, (outflow face head),

Kb = 1.0, (transmissivity),

r = 5.0, (fracture radius),

l = 9.798, (node length), and

D = 2.0, (the distance between the constant head boundaries).



XBL 853-8834

Figure 5-1. Example of applications:
(a) one fracture intersecting two parallel boundary planes;
(b) two equi-dimensional intersecting fractures, each intersecting one boundary;
(c) two equi-dimensional intersecting fractures, each intersecting two boundaries;
(d) three equi-dimensional, orthogonal fractures of radius r which intersect symmetrically and are truncated by a boundary cube of dimensions r $\sqrt{2}$ by r $\sqrt{2}$ by r $\sqrt{2}$

Solving for flux in rectangles with areas slightly larger and slightly smaller than the actual flow area gave us an upper and a lower bound on the correct answer. In the analytical case, for a slab of dimensions 10.0 x 2.0, where 10.0 equals the diameter of the fracture, the flow is 5.0. For a slab 9.798 x 2.0, where 9.798 equals the length of the nodes, the flow is 4.899. The flux between the nodes in the numerical case was 4.968, thus, the analytical results bound the numerical result, as expected.

### 5.1.2 Case 2: Two Equi-Dimensional Intersecting Fractures, Each Intersecting One Boundary

In this case, illustrated in Figure 5-1b, we used two perpendicular fracture discs of equal area and two perpendicular boundary planes. The intersection between the fractures is the same length as two boundary nodes formed by the intersections of the fractures and the boundary planes. We expected the average head on the intersection between the two fractures to be the average of the heads on the two boundary nodes and by symmetry we expected the total flux into or out of each node to be equal in magnitude. Parameters used for this case were:

$H_1$ = 1.0,

$H_2$ = 0.0,

r = 5.0,

l = 6.0,

Kb = 1.0, and

D = 8.0, (the distance between the intersections).

The numerical results were Q = +.4949 for each inflow node and Q = -.4949 for each outflow node and the average head at the intersection between the fractures was 0.5, as expected.

### 5.1.3 Case 3: Two Equi-Dimensional Intersecting Fractures, Each Intersecting Two Boundaries

For this case we used two equal area, perpendicular fractures which intersect through the fracture centers. Parallel to each fracture we have a pair of boundary planes intersecting the other fracture symmetrically, as shown in Figure 5-1c. This case was designed to check the calculation for net flux at an intersection when, in the plane of the fracture, flow is entering the intersection on one side and leaving the intersection on the other. For such a symmetric case, the head at the center intersection should be $(H_1 + H_2)/2$ and the total flux at the intersection should be zero because the same amount of flow is entering this node on one side as is leaving it on the other. In fact, for $H_1$=1.0 and $H_2$=0.0, the numerical results were Q = 0.0 and H = 0.5 at the center node, as expected.

### 5.1.4 Case 4: Three Equi-Dimensional Orthogonal Fractures

In this case, illustrated in Figure 5-1d, we used three orthogonal fractures of radius r that intersect symmetrically and are truncated at right angles by a cubic flow region of dimensions $r\sqrt{2} \times r\sqrt{2} \times r\sqrt{2}$ . The applied boundary conditions induce a constant gradient in the flow region (these conditions are fully described in Section 2.2.1 and illustrated in Figure 2-4). The two fractures which are perpendicular to the inflow face, therefore, have a constant gradient, as illustrated on the right hand side of Figure 5-1d. There is zero gradient in the fracture parallel to the inflow face; thus this fracture experiences no flow. An analytical solution provides us with an approximation of the flux into each of the boundary nodes on these fractures. It is an approximation because it assumes a square flow area in each fracture plane, while the numerical solution assumes a circular flow area, which in this case is slightly larger. Flux into the the nodes on the inflow face must be equal to flux out of the corresponding nodes on the outflow face. Also, from symmetry, we should have $H = (H_1 + H_2)/2$ and Q=0.0 on all of the internal nodes.

The parameters used in this case were:

$H_1$ $= 1.0$,

$H_2$ $= 0.0$,

r $= 71.0$,

l $= 100.0 \approx r\sqrt{2}$ ,

Kb $= 0.8172 \times 10^{-5}$.

The analytical solution for flux through the inflow or outflow face of the cubic flow region is:

$$\begin{aligned} Q_r &= n \left[ Kbl \left( \frac{H_1 - H_2}{l} \right) \right] \\ &= 2(0.8172 \times 10^{-5} \times 100 \times 1.0/100), \\ &= 0.1634 \times 10^{-4}, \end{aligned}$$

where n = 2 because two fractures intersect the inflow and outflow faces. The numerical result, when neither internal nodes nor boundary nodes were discretized, was $Q_r = 0.1982 \times 10^{-4}$. In an attempt to improve the correlation between the analytical and numerical values, we experimented with nodal discretization.

## 5.2 TESTING NODAL DISCRETIZATION SCHEMES

### 5.2.1 Discretizing Nodes in Case 4: Three Equi-Dimensional Orthogonal Fractures

Using Test Case 4, as presented above, we subdivided the nodes in various ways to determine the effect on accuracy. In all cases, the conditions on the internal nodes were satisfied, i.e., H = 0.5 and Q = 0.0. However, the different schemes for dividing the nodes produced slightly different values for flux into the boundary nodes on the inflow face (which must be, and were, equal to the flux out of the corresponding boundary nodes on the outflow face).

used. The first was to divide each node into a number of equal segments [nseg = nsegb = 2, 3, 4, and 10] and compare the results with the analytical solution. The numerical results, given in Table 5-1, show that dividing the nodes improves the accuracy. The greatest improvement was achieved by going from unsegmented nodes to two segments per node. Results with ten segments per node were even better but uneconomical since DISCEL sees each segment as a separate node. Therefore, other discretization schemes were tried in an attempt to get the best results with the least number of segments.

**TABLE 5-1.**
**Total Flux Through the Fracture Network in Case 4, All Nodes are Divided into a Given Number of Equal Segments**

| Number of Equal Segments | Total flux × $10^{-4}$ |
|---|---|
| unsegmented | 0.1982 |
| 2 | 0.1705 |
| 3 | 0.1679 |
| 4 | 0.1665 |
| 100 | .1647 |
| Analytical Solution | 0.1634 |

In this particular case, each boundary node endpoint is also the endpoint of a perpendicular boundary node (Figure 5-1d). We tried dividing the boundary nodes near the endpoints, as when using ten segments per node, by adjusting the nodal division points so that the first and last segments were one tenth the node length. This was done sequentially to all nodes, boundary nodes only, and then boundary nodes with internal nodes divided into a given number of equal segments. Results from the various runs indicated that only boundary node segmentation affected overall flux. In order to confirm this, boundary nodes were divided into a given number of equal segments and internal nodes were not divided.

Table 5-2 lists the various combinations of boundary node and internal node segmentations that were tried as well as the resulting flux. Dividing internal nodes appears to have no effect on the total flux through the fracture system; therefore, only boundary node discretization may be necessary to improve accuracy. If this were true in general it would be advantageous because we wish to minimize the number of nodes, i.e., the

**TABLE 5-2.**
**Total Flux through the Fracture Network in Case 4 as a Function of Nodal Discretization Scheme**

| Segmentation of boundary nodes | Segmentation of internal nodes | Total flux × $10^{-4}$ |
|---|---|---|
| unsegmented | unsegmented | 0.1982 |
| 2 equal | unsegmented | 0.1705 |
| | 2 equal | 0.1705 |
| 3 equal | unsegmented | 0.1679 |
| | 3 equal | 0.1679 |
| | 3 adjusted(a) | 0.1679 |
| 3 adjusted(a) | unsegmented | 0.1692 |
| | 2 equal | 0.1692 |
| | 3 equal | 0.1692 |
| | 3 adjusted(a) | 0.1692 |
| 4 equal | unsegmented | 0.1665 |
| | 4 equal | 0.1665 |
| | 4 adjusted(b) | 0.1665 |
| 4 adusted(b) | unsegmented | 0.1651 |
| | 2 equal | 0.1651 |
| | 4 equal | 0.1651 |
| | 4 adjusted(b) | 0.1651 |
| 5 adjusted(c) | unsegmented | 0.1648 |
| | 2 equal | 0.1648 |
| 10 equal | unsegmented | 0.1647 |
| | 10 equal | 0.1647 |

(a) 3 adjusted -- proportions 0.1 : 0.8 : 0.1
(b) 4 adjusted -- proportions 0.1 : 0.4 : 0.4 : 0.1
(c) 5 adjusted -- proportions 0.1 : 0.267 : 0.266 : 0.267 : 0.1

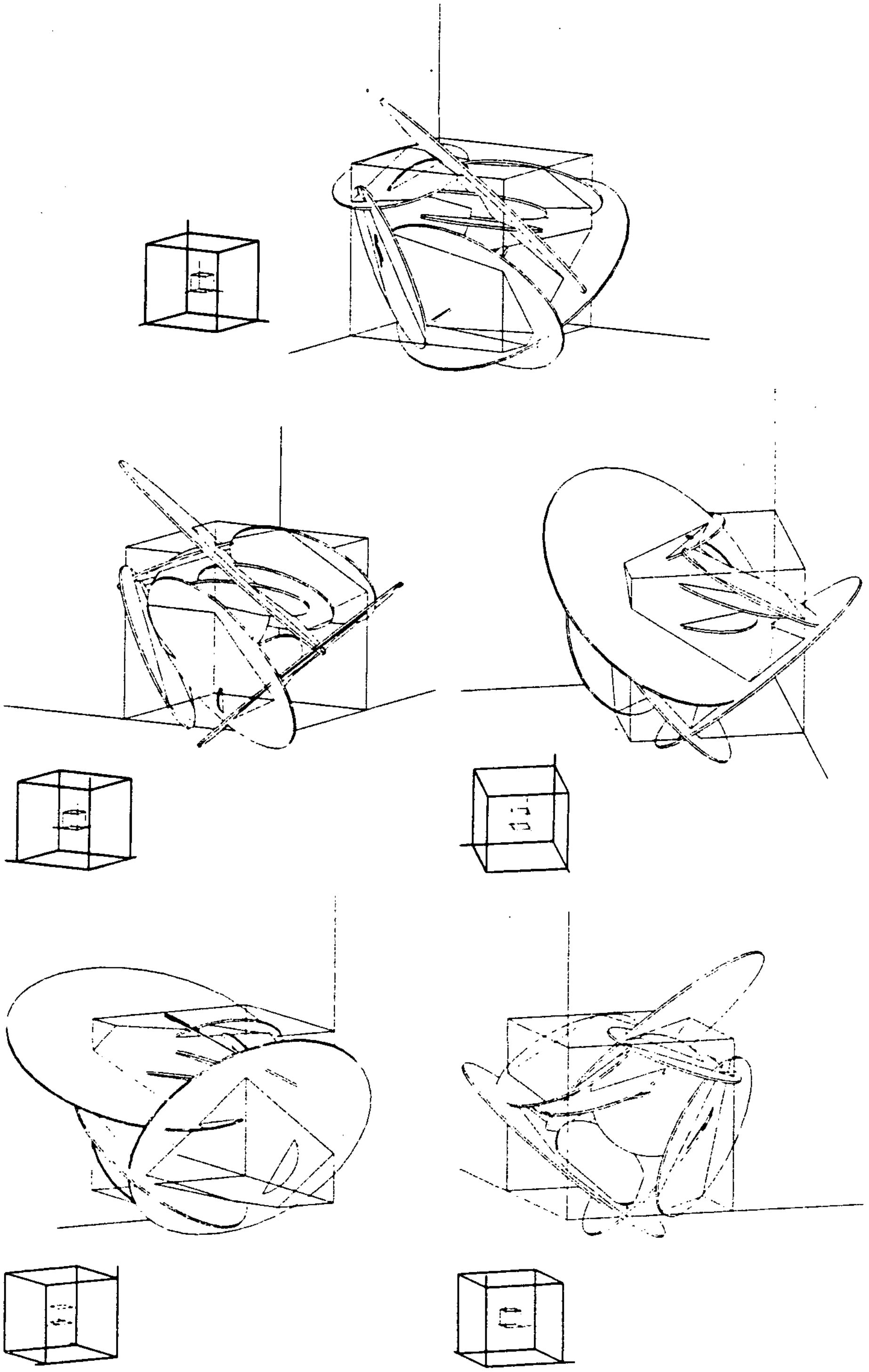
number of unknowns. This particular case is, however, symmetric and each internal node is truncated by boundary nodes at both ends. So, we looked at a more general case.

### 5.2.2 Ten Randomly Located, Random-Sized Fractures in a Cubic Flow Region

In this case, ten fractures were randomly generated and truncated by a cubic flow region with constant gradient boundary conditions. The fracture system was designed to be highly interconnected and formed 38 nodes, including 18 internal nodes and 20 boundary nodes (Figure 5-2). Because this is a random case, no analytical solution is available to check the results. Therefore, we made the assumption that the "true head and flux values" are closest to the values calculated when all the nodes are divided into ten equal segments. The results for less than ten segments on some or all nodes are then compared with these "true values". The fracture network and boundary conditions were such that there was flow entering or leaving all six faces of the boundary cube.

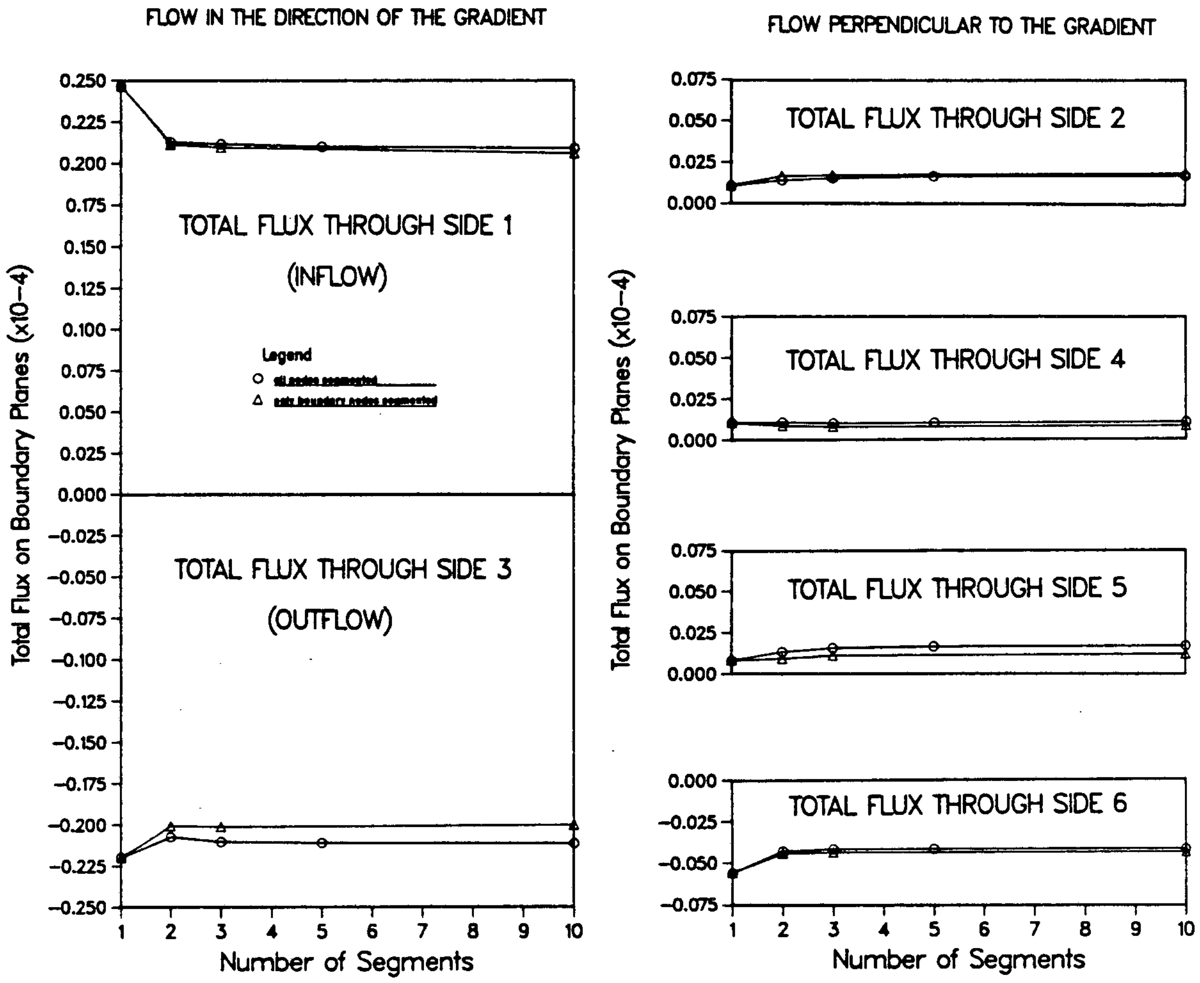
First, we comfirmed that very little accuracy is gained from discretizing all nodes over discretizing only boundary nodes. Although the boundary fluxes are not identical, as in Test Case 4, the results are only slightly different. The graphs presented in Figure 5-3 show total flux through the six boundary planes when all nodes were divided into two, three, five, and ten equal segments, as well as the unsegmented case. Also shown are the results of dividing only the boundary nodes into two, three, and ten equal segments. As can be seen, there is very little difference. Therefore, in order to minimize problem size and still maintain accuracy, it appears to be most efficient to discretize the boundary nodes and not discretize the internal nodes.

Next, we confirmed a general improvement in accuracy with increased discretization. Results of discretizing only boundary nodes into two, three, and ten equal segments and into three adjusted segments, (see footnote, Table 5-2), are given in Figures 5-4, 5-5, and 5-6. These results are presented as differences between each of the test case



XBL 869 11039

Figure 5-2. Network generated by FMG3D for random case, seen from varying points of view



XBL 853-1858

Figure 5-3. Total flux results for small random case as a function of nodal discretization scheme.

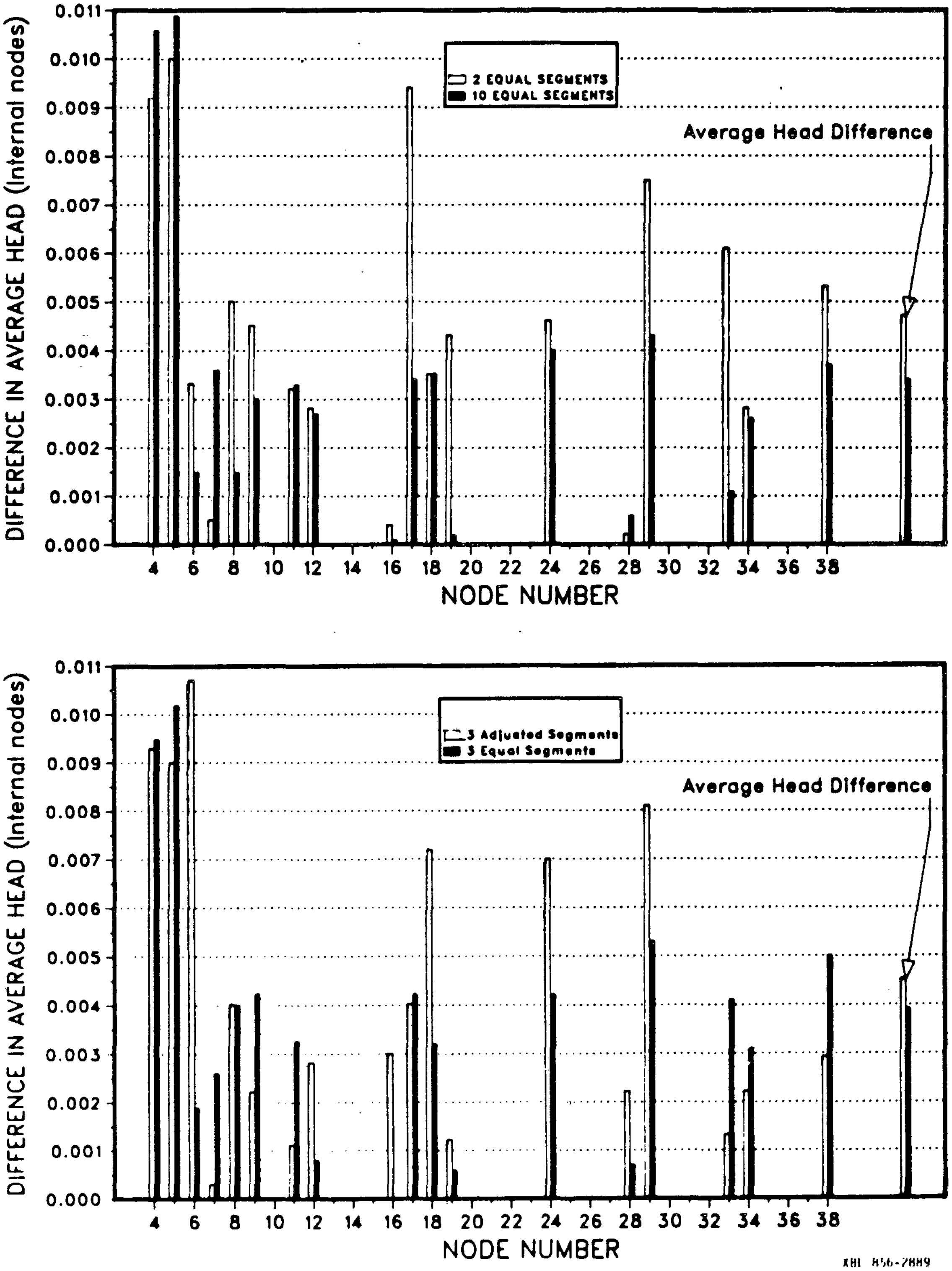
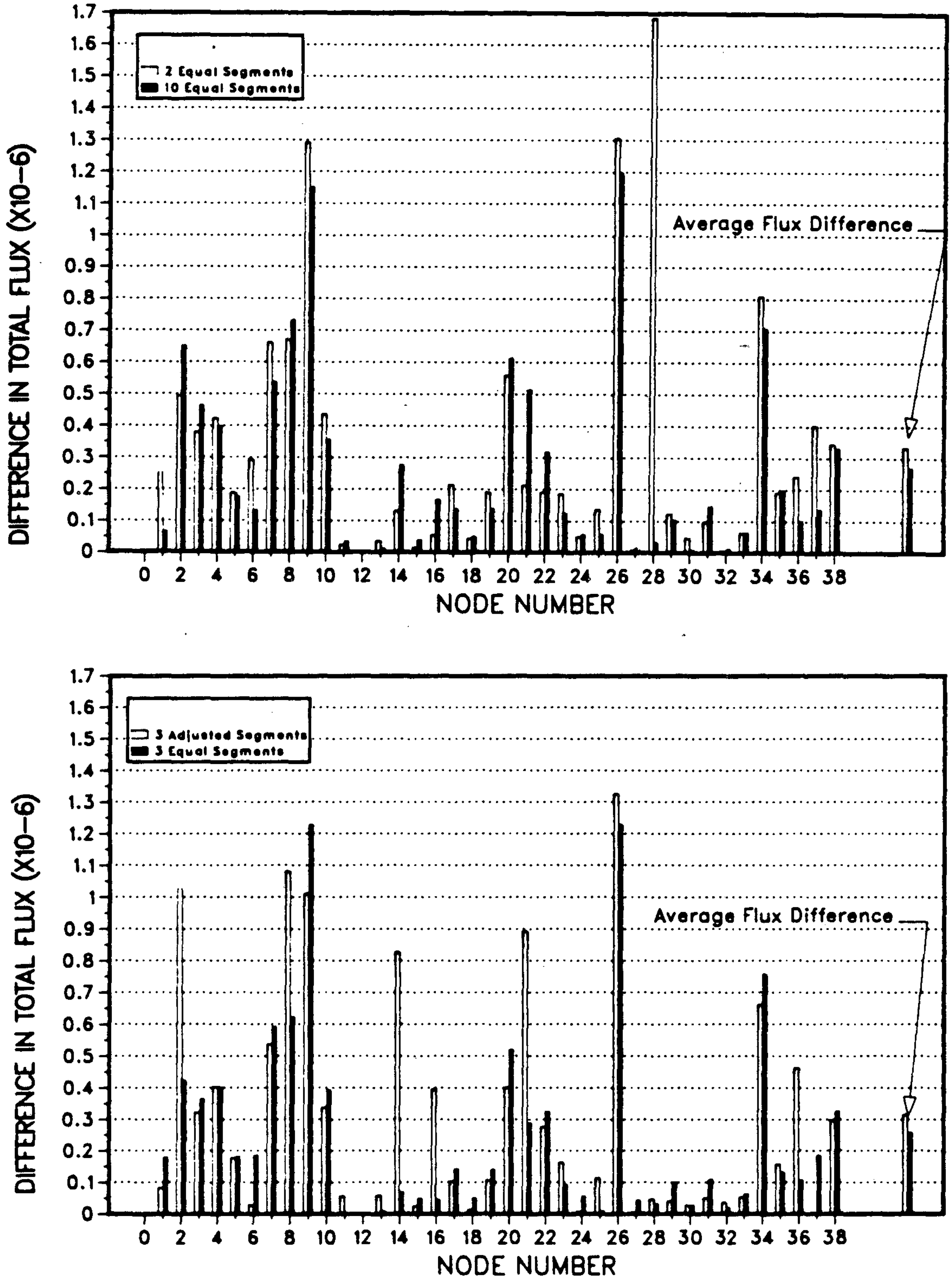
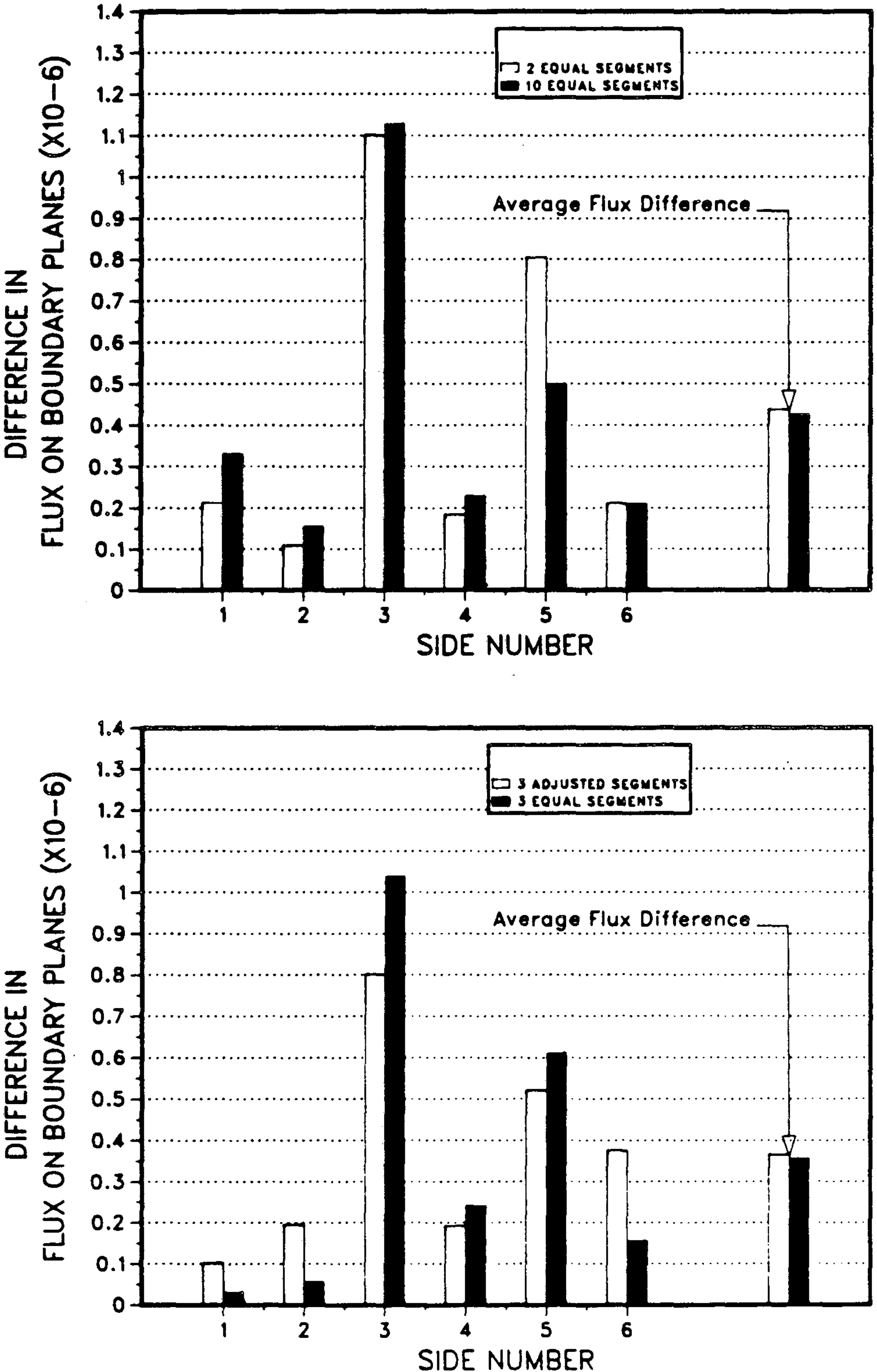


Figure 5-4. Difference in average head at internal nodes for a small random case as a function of nodal discretization scheme.



XBL 856-2891

Figure 5-5. Difference in total flux through each node for a small random case as a function of nodal discretization scheme.



XBL 856-2890

Figure 5-6. Difference in total flux through each boundary plane for a small random case as a function of nodal discretization scheme.

values and the "true values" obtained when all nodes were divided into ten equal segments. In Figure 5-4 we see the difference in average head at each internal node and in Figure 5-5 the difference in total flux through each node. Figure 5-6 shows the difference in total flux through each of the six boundary planes. At the right hand side of each graph is the average difference in head or flux. From these results, it appears that dividing boundary nodes into three equal segments and leaving internal nodes unsegmented [nseg = 1 and nsegb = 3] achieves the highest accuracy with the least increase in problem size. Further investigation is needed, however, to confirm that this is the best scheme under all circumstances and the user has the option to override these default values and input other segmentation parameter values.

## 5.3 EXAMPLES OF DISC NETWORKS PLOTS

Figure 5-7 shows a regular and symmetrical fracture network with the flow region totally included in all fractures. The size of the flow region and the gradient imposed where adjusted so that the fluxes entering or leaving each of the boundaries should be equal, according to the analytical solution for infinite fractures. This was well verified. Figures 5-8 and 5-9 show how the program can be used for plotting a subregion instead of plotting all the fractures. They show two different flow regions originated from the same network.

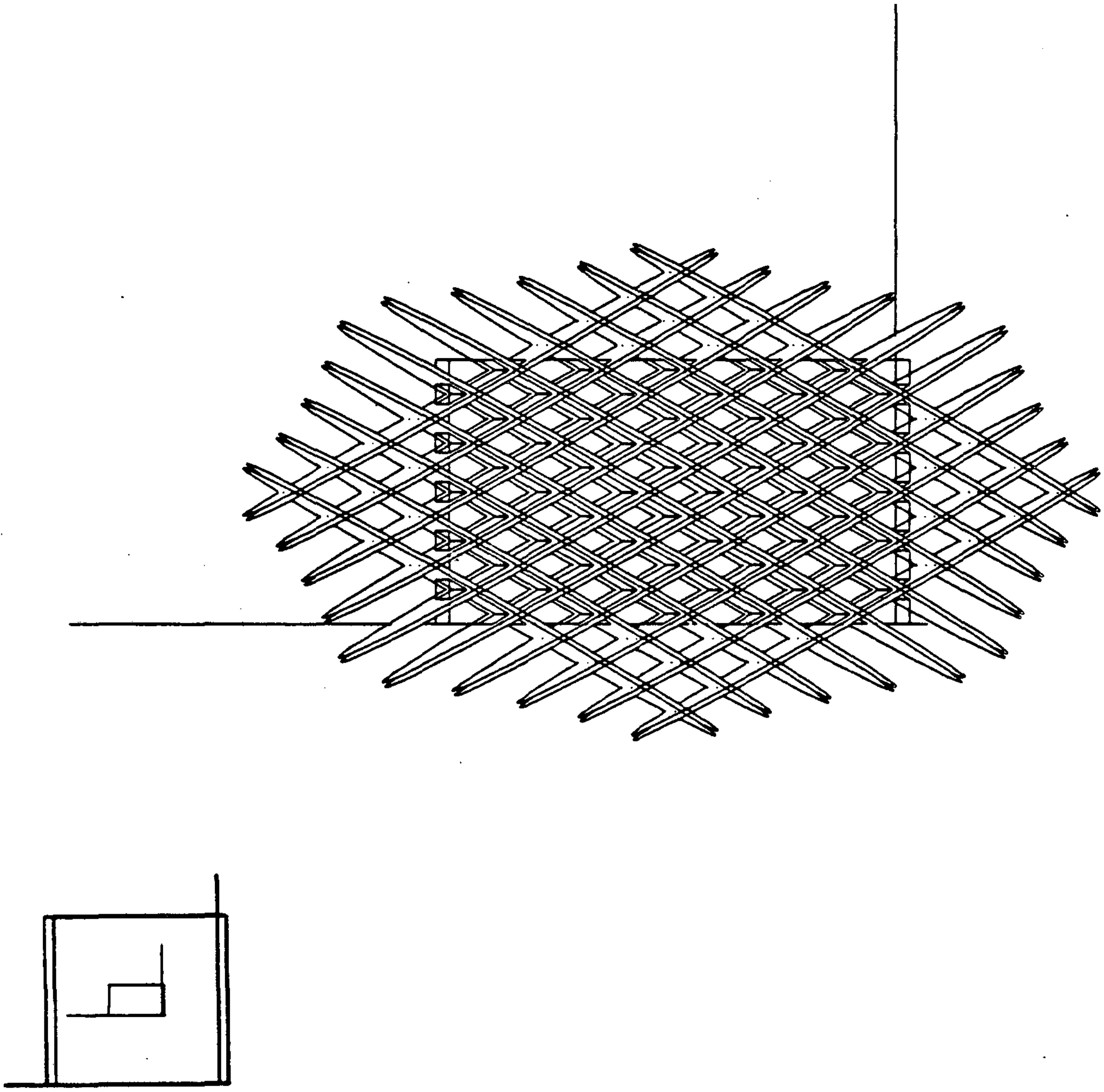
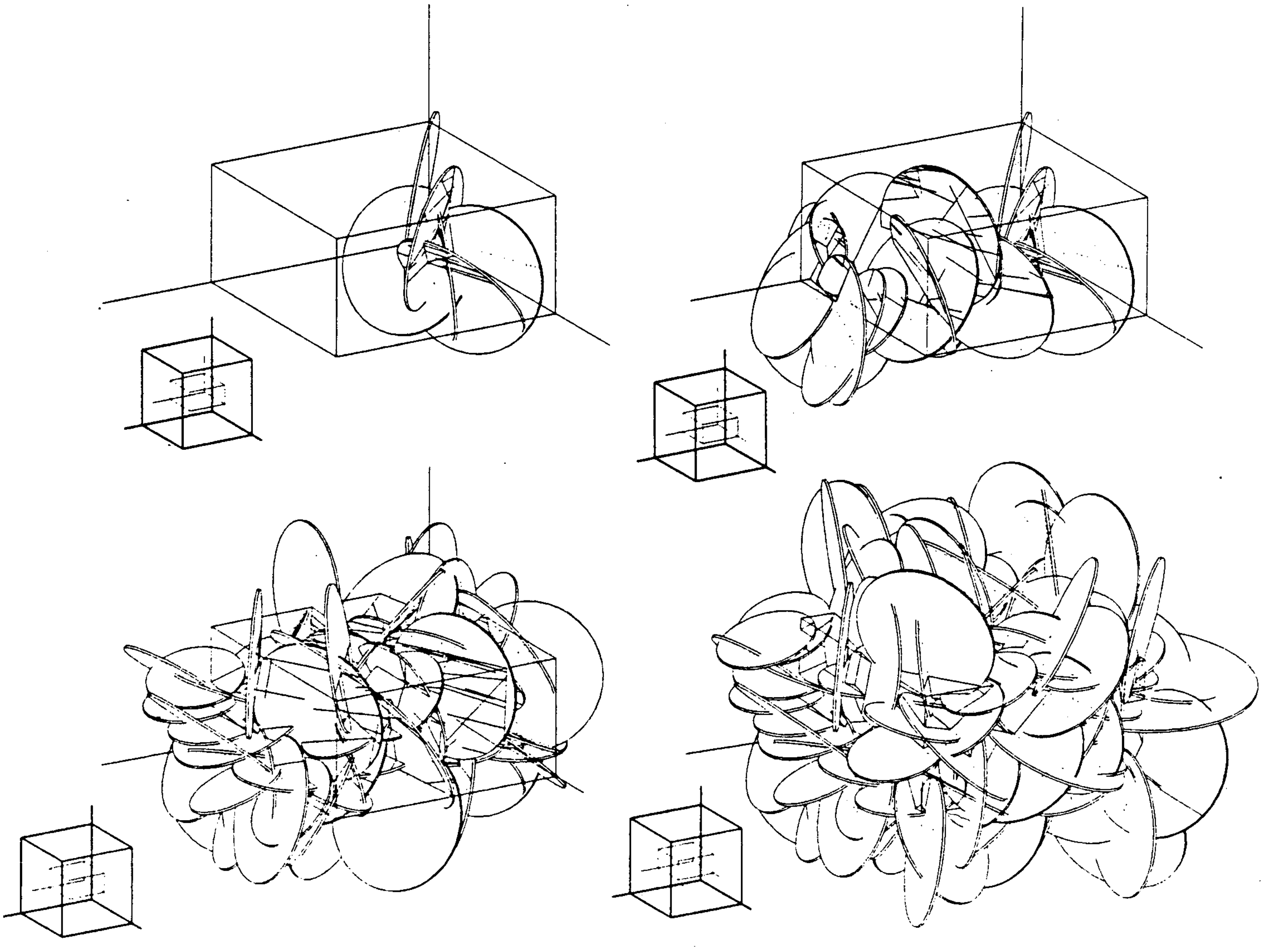
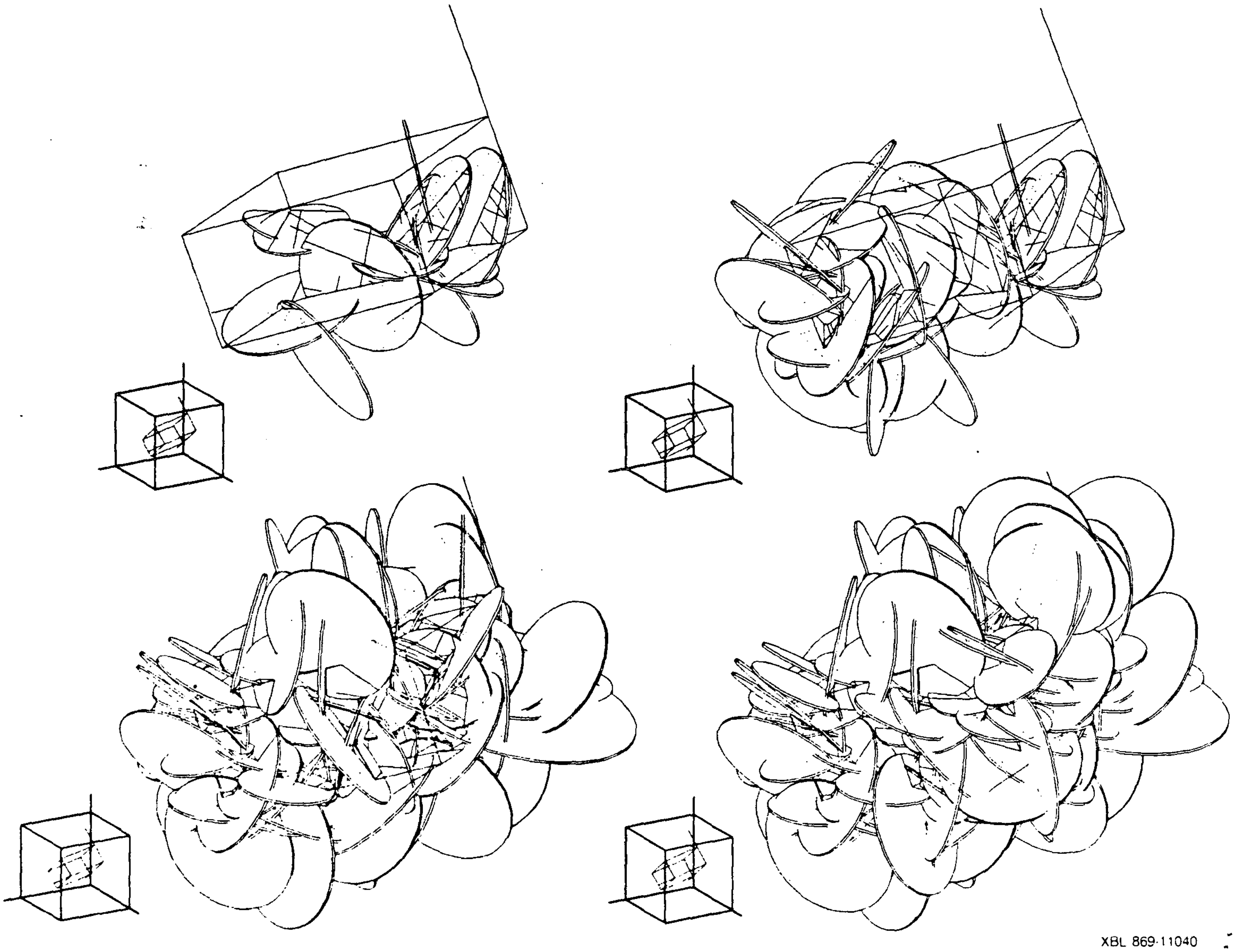


Figure 5-7. Regular fracture network



XBL 8610-3726

Figure 5-8. Random network, non rotated flow region



XBL 869-11040

Figure 5-9. Random network, rotated flow region

# APPENDIX A
# LOCAL FRACTURE SYSTEM COORDINATE TRANSFORMATION AND OTHER COMPUTATIONS

The solution of potential in Equations 2-2 and 2-6 is obtained in the plane of the fracture, therefore, for each fracture in turn, a coordinate transformation from the global XYZ system to a local xy system must be performed.

The global XYZ system is rotated, using the fracture orientation angles $\phi$ and $\theta$ (provided by FMG3D), such that the new positive z-axis is perpendicular to the fracture plane and the origin is translated to the fracture center $(X_{c,}Y_{c,}Z_c)$. The transformation,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\theta & -\cos\phi\sin\theta & -\sin\phi \\ \sin\theta & \cos\theta & 0 \\ \sin\phi\cos\theta & -\sin\phi\sin\theta & \cos\phi \end{bmatrix} \begin{bmatrix} X - X_c \\ Y - Y_c \\ Z - Z_c \end{bmatrix} \qquad \text{[A-1]}$$

is applied to the endpoints of all nodes on the fracture giving endpoints $(x_{1,}y_1)$ and $(x_{2,}y_2)$ for each node. The z-coordinate will always be zero since the equation of the fracture plane in normal form is

$$aX + bY + cZ + d = 0, \qquad \text{[A-2]}$$

where a, b, and c are direction cosines of the normal to the plane and

$$a = \sin\phi\cos\theta,$$
$$b = -\sin\phi\sin\theta,$$
$$c = \cos\phi,$$

and

$$-d = aX_c + bY_c + cZ_{c.} \qquad \text{[A-3]}$$

For the particular fracture, k, the nodes on the fracture are renumbered from 1 to n [nt]. This is the local node numbering system. The length of each node, $l_i$ [rl(i)], is computed,

$$l_i = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad . \qquad \text{[A-4]}$$

The width of each node is computed from the aperture and the coefficients of the equation of the intersecting plane, j,

$$w_i = \frac{\text{aperture}_j}{\sqrt{1-\cos^2\theta}} \quad , \qquad [A\text{-}5]$$

where

$$\cos\theta = \left| a_k a_j + b_k b_j + c_k c_j \right| \quad .$$

Half this width is stored [bm(i)].

The node type is determined to be point, radial, or nonradial. If the node length, $l_i$, is zero, the node is a point. If $l_i$ is greater than zero, the equation of the line containing the node, in normal form, is

$$\lambda x + \mu y - p = 0 \quad , \qquad [A\text{-}6]$$

where

$$\lambda = \frac{y_1 - y_2}{\pm l_i}, \quad \mu = \frac{x_2 - x_1}{\pm l_i} \quad ,$$

and

$$p = \frac{y_1\, x_2 - y_2\, x_1}{\pm l_i} \quad , \qquad [A\text{-}7]$$

the sign of $l_i$ is such that p is positive. In this form, $\lambda$ and $\mu$ are direction cosines of the normal to the line and p is the distance from the origin to the line along the normal. If p equals zero, the line passes through the origin; thus node i is radial. If p is greater than zero, node i is nonradial.

Fracture transmissivity [pkb] is computed from the aperture and permeability of fracture k in either cgs or mks units, depending on an input parameter [iunits]:

$$K^k\, b^k = \text{aperture}_k^3 \ \frac{\rho g}{12\mu} \quad . \qquad [A\text{-}8]$$

where $\rho$ is the density and $\mu$ is the viscosity of water and g is the gravitational constant.

Subroutine ROFRAC is called once for each fracture to determine and execute the coordinate transformation, set up a local node numbering system, compute the length, width, and type of each node on the fracture, and to calculate fracture transmissivity.

# APPENDIX B
# METHOD FOR EXPRESSING THE POTENTIAL IN A PLANE DUE TO A NODE

## B.1 Point Node

When node i is a point source in fracture disc k, the expressions for potential due to the node and its images, $\phi_{iN}^{k}$, $\phi_{iI+}^{k}$, and $\phi_{iI-}^{k}$ are derived from Equations 2-1 and 2-7. Figure B-1 diagrams such a node and its images. The coordinates of the node in the local fracture coordinate system (Appendix A), are $(x_1,y_1)$, the fracture radius is a, and P(x,y) is an arbitrary point in the plane. The coordinates of the image source are

$$x_I = \frac{a^2 x_1}{x_1^2 + y_1^2} \quad \text{and} \quad y_I = \frac{a^2 y_1}{x_1^2 + y_1^2} \quad . \qquad [B\text{-}1]$$

Substituting an expression for r, the distance from point P to node i, in Equation 2-1 gives an equation for potential due to a source of strength $Q_i$ at node i. And substituting an expression for $r_p$, the distance from point P to the image source of node i, gives an equation for potential due to a source of strength $Q_i$ at this image. Therefore

$$\phi_{iN}^{k} = K^k b^k \overline{h}_{iN} = \frac{-Q_i}{4\pi} \ln \left[(x - x_1^2) + (y - y_1)^2\right]$$

and

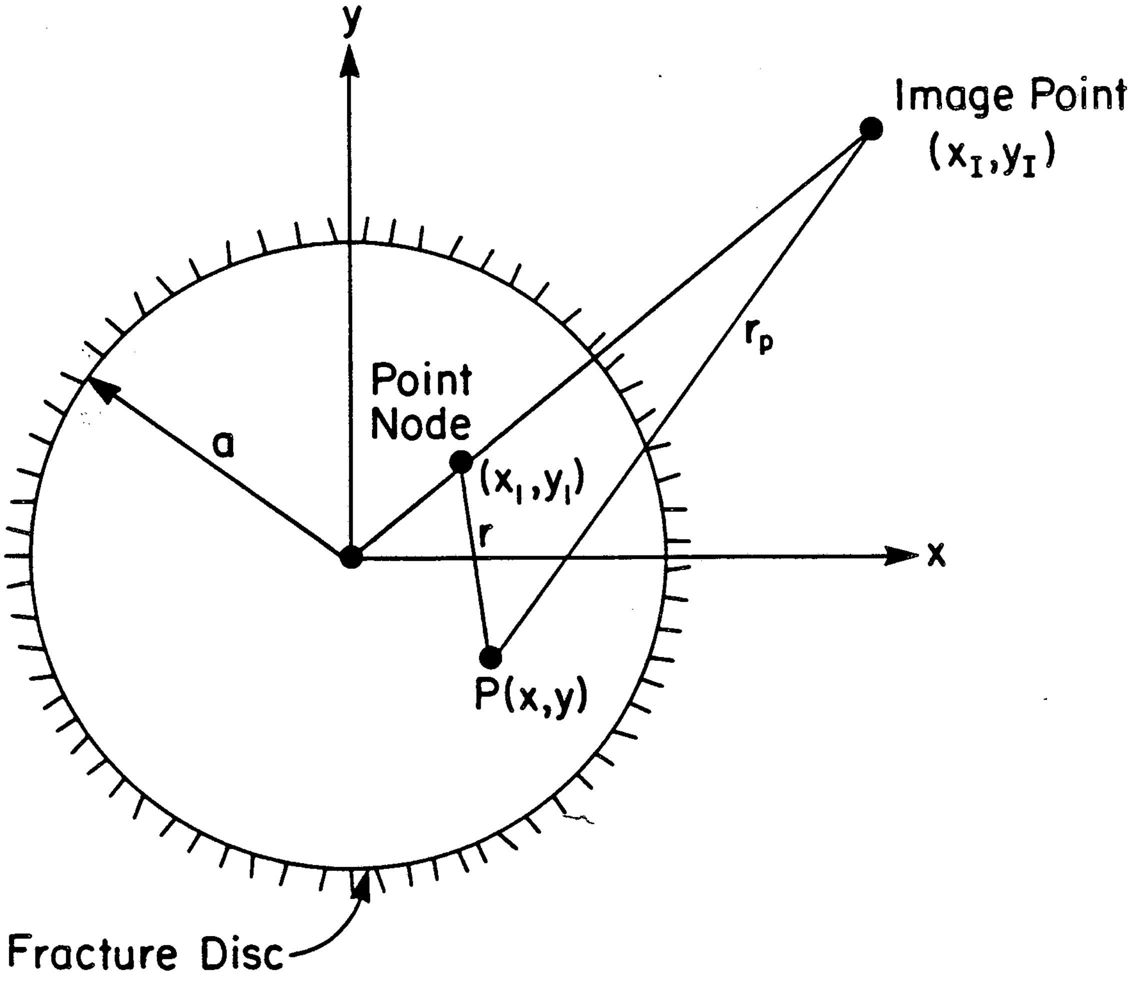
$$\phi_{iI}^{k} = K^k b^k \overline{h}_{iI} = \frac{-Q_i}{4\pi} \ln \left[(x - x_I)^2 + (y - y_I)^2\right] \qquad [B\text{-}2]$$

We also have:

$$\phi_{iI-}^{k} = K^k \, b^k \, \overline{h}_{iI-} = \frac{Q_i}{4\pi} \ln(x^2 + y^2) \qquad [B\text{-}3]$$

Note that the expression for $\phi_{iI-}^{k}$ will be exactly the same in the case of a radial or non radial node, depending only on the strength of the node and on the distance from the center to the point where it is computed. We now have an expression for potential due to the presence of point node i in fracture k in terms of $Q_i$ , x and y, and can write

$$\phi_i^k = \phi_{iN}^{k} + \phi_{iI+}^{k} + \phi_{iI-} = Q_i \, f_{iN}^{k} \, (x,y) + Q_i \, f_{iI+}^{k} \, (x,y) + Q_i \, f_{iI-}^{k} \, (x,y)$$



XBL 855-10509

Figure B-1. Construction for calculating potential due to a point node and its images

$$= Q_i\,[f^k_{iN}\,(x,y) + f^k_{iI+}\,(x,y) + f^k_{iI-}\,(x,y)]$$

$$= Q_i\,F_i^k\,(x,y) \qquad [B\text{-}4]$$

where $Q_i$ is the total strength of the point source and $F_i^k = f^k_{iN} + f^k_{iI+} + f^k_{iI-}$. Subroutine POINT computes $x_I$ and $y_I$ and calls INTEGFP to compute potentials.

## B.2 Radial Node

When node i is a radial line segment of length $l_i$ in fracture disc k, expression for potential due to sources distributed along the node and its image source, $\phi^k_{iN}$ and $\phi^k_{iI+}$, are derived from Equations 2-2 and 2-6. First, the local fracture x, y-coordinate system, (Appendix A), is rotated to an x'y' system such that the x'-axis is coincident with the node, as shown in Figure B-2a. The positive x'-axis is in the direction of the nodal endpoint farthest from the origin. If $g_1$ is the distance from (0,0) to $(x_1,y_1)$, and $g_2$ is the distance from (0,0) to $(x_2, y_2)$, then the greater distance, $g_m$, is used to compute coefficients $\lambda$ and $\mu$ for the rotation:

$$\lambda = \frac{x_m}{g_m} \text{ and } \mu = \frac{y_m}{g_m} \quad .$$

The rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \lambda & \mu \\ -\mu & \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad [B\text{-}5]$$

is applied giving nodal endpoints $(x_1'\ , 0)$ and $(x_2'\ , 0)$. The limits of integration, $\xi_1$ and $\xi_2$, are set to the minimum and maximum of $x_1'$ and $x_2'$ , respectively.

Equation 2-2 is used for the potential due to sources with a total strength of $Q_i$ distributed along the node. The strength per unit line length, $Q_i/l_i$, is constant. The distance $r_p$ from any point $P(x'\ ,-y'\ )$ in the plane to a source at point $(\xi,0)$ on the node is computed:

$$r_p(\xi) = [(x' - \xi)^2 + y'^2]^{1/2} \quad . \qquad [B\text{-}6]$$

Substituting $r_p(\xi)$ in Equation 2-2 gives

$$\phi^k_{iN} = K^k b^k \overline{h}_{iN} = \frac{-Q_i}{4\pi l_i} \int_{\xi_1}^{\xi_2} \frac{1}{2} \ln\,[(x' - \xi)^2 + y'^2]\, d\xi \quad , \qquad [B\text{-}7]$$

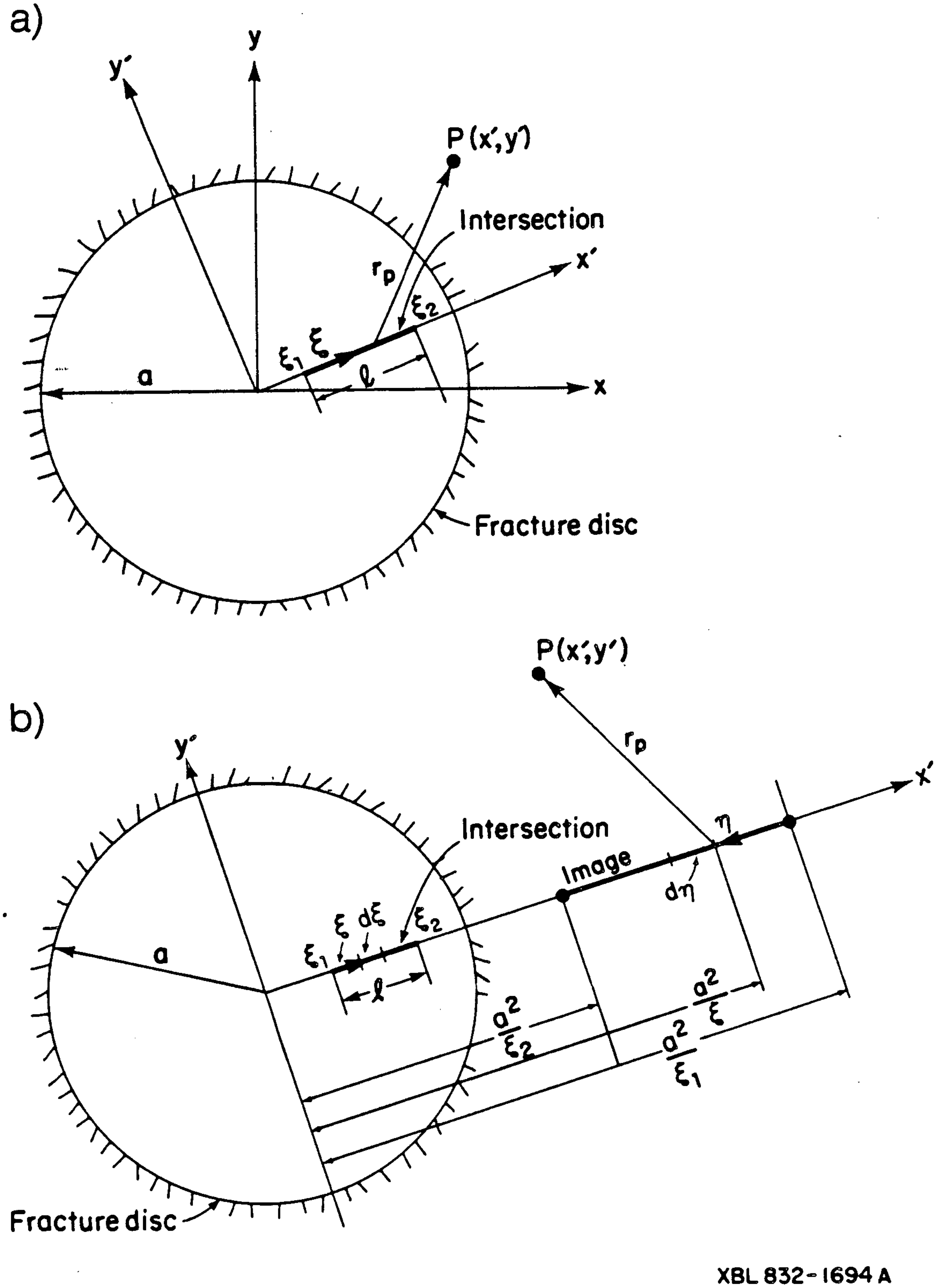


Figure B-2. (a) Construction for calculating potential due to a radial node.
(b) Construction for calculating potential due to the image source of a radial node

where $K^k$ is the permeability of fracture k, $b^k$ is the aperture and $\overline{h}_{iN}$ is the average head on node i in both fracture k and the other fracture forming the intersection. This integral as evaluated in Selby (1965, p.334, #380) is:

$$\phi_{iN}^{k} = \frac{-Qi}{4\pi l_i} \left[ (\xi - x') \ln [(x' - \xi)^2 + y'^{\,2}] - 2\xi + 2\,|\,y'\,|\,\tan^{-1}\left( \frac{\xi - x'}{|\,y'\,|} \right) \right]_{\xi_1}^{\xi_2} . \qquad [B\text{-}8]$$

The image source of a radial node is also a radial line segment, extending from $x' = a^2/\xi_1$, to $x' = a^2/\xi_2$, as shown in Figure B2-b. The distance from a source at point $\eta$ on the image, measured from image endpoint $a^2/\xi_1$, to an arbitrary point $P(x', y')$ in the plane is

$$r_p(\eta) = \left[ \left( x' - \left(\frac{a^2}{\xi_1} - \eta\right) \right)^2 + y'^{\,2} \right]^{1/2} . \qquad [B\text{-}9]$$

To get $r_p(\eta)$ in terms of $\xi$, we see that

$$\eta = \frac{a^2}{\xi_1} - \frac{a^2}{\xi} .$$

Therefore,

$$r_p(\xi) = [(x' - \frac{a^2}{\xi})^2 + y'^{\,2}]^{1/2}$$

and

$$r_p^2(\xi) = \frac{(x'^{\,2} + y'^{\,2})\xi^2 - 2a^2x'\,\xi + a^4}{\xi^2} \qquad [B\text{-}10]$$

Now Equation 2-6, the potential due to sources with a total strength of $Q_i$ distributed along the image of radial node i in fracture k, can be written:

$$\phi_{iI+}^{k} = K^k b^k \overline{h}_{iI} = \frac{-Q_i}{r\pi l_i} \left( \int_{\xi_1}^{\xi_2} \ln[(x'^{\,2} + y'^{\,2})\xi^2 - 2a^2x'\,\xi + a^4] d\xi - 2\int_{\xi_1}^{\xi_2} \ln\xi d\xi \right) . \qquad [B\text{-}11]$$

The first integral is of the same form as Equation B-7. Evaluating gives:

$$\phi_{iI+}^{k} = \frac{-Qi}{4\pi l_i} \Bigg\{ \left( \xi - \frac{a^2x'}{x'^{\,2} + y'^{\,2}} \right) \ln[(x'^{\,2} + y'^{\,2})\xi^2 - 2a^2x'\,\xi + a^4]$$
$$- 2\xi + \frac{2a^2\,|\,y'\,|}{x'^{\,2} + y'^{\,2}} \tan^{-1}\left( \frac{(x'^{\,2} + y'^{\,2})\xi - a^2x'}{a^2\,|\,y'\,|} \right) - \xi\ln\xi^2 + 2\xi \Bigg\}_{\xi_1}^{\xi_2} . \qquad [B\text{-}12]$$

Substituting the limits $\xi_1$ and $\xi_2$ in Equations B-8 and B-12 gives expressions for $\phi_{iN}^{k}$ and $\phi_{iI+}^{k}$ in terms of $Q_i$ , x' and y'. Therefore, for any radial node we can write an expression for potential due to the presence of that node, i, in fracture k, adding the expression for $\phi_{iI-}^{k}$ found previously (Equation B-3):

$$\begin{aligned}\phi_i{}^k = \phi_{iN}^{k} + \phi_{iI+}^{k} + \phi_{iI-}^{k} &= Q_i\, f_{iN}^{k}(x' ,y' ) + Q_i\, f_{iI+}^{k}(x' ,y' ) + Q_i\, f_{iI-}^{k}(x' ,y' ) \\ &= Q_i\, [F_{iN}^{k}(x' ,y' ) + f_{iI+}^{k}(x' ,y' ) + f_{iI+}^{k}(x' , y' )] \\ &= Q_i\, f_i{}^k (x' ,y' ) \qquad [B\text{-}13]\end{aligned}$$

where $Q_i$ is the total strength of the line source and $F_i{}^k = f_{iN}^{k} + f_{iI+}^{k} + f_{iI-}^{k}$.

Subroutine RADIAL determines $\lambda$, $\mu$, $\xi_1$ , and $\xi_2$ , and calls INTEGFR to compute potentials.

### B.3 Nonradial node

When node i is a nonradial line segment of length $l_i$ in fracture disc k, expressions for potential due to sources distributed along the node and its image source, $\phi_{iN}^{K}$ and $\phi_{iI+}^{K}$, are derived from Equations 2-2 and 2-6. A convenient coordinate system for nonradial node calculations is shown in Figure B-3. The local fracture xy system, (Appendix A), is rotated to an x'y' system in which the positive x'-axis is perpendicular to the node and the y'-axis is parallel to the node. The origin remains at the fracture center. The rotation uses the coefficients of the equation of the nodal line in normal form, $\lambda x + \mu y - p = 0$, (Equation A-6):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \lambda & \mu \\ -\mu & \lambda \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad [B\text{-}14]$$

The equation of the nodal line becomes x' = p, and the nodal endpoints rotate to $(p, y_1')$ and $(p, y_2')$. The limits of integration, $\xi_1$ and $\xi_2$, are set to the minimum and maximum of $y_1'$ and $y_2'$ , respectively.

When evaluating the expression for potential, the total strength of the line source is $Q_i$ and the strength per unit line length, $Q_i/l_i$, is constant. The distance from a source at $(p,\xi)$ on node i to an arbitrary point P(x',y') in the plane, (Figure B-3), is given by

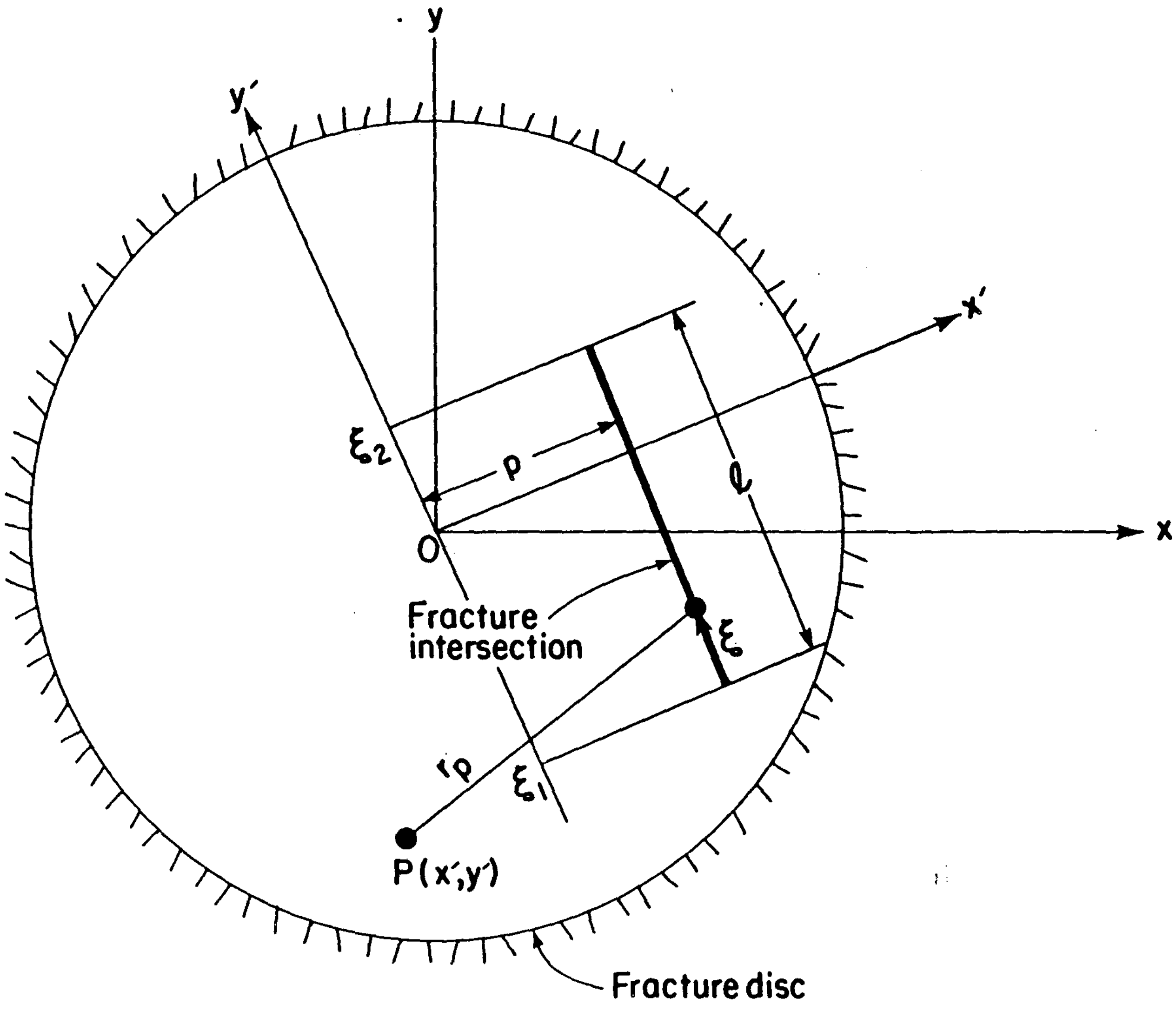


Figure B-3. Construction for calculating potential due to a nonradial node.

$$r_p(\xi) = [(x' - p)^2 + (y' - \xi)^2]^{1/2} \quad . \qquad [B\text{-}15]$$

When this is substituted in Equation B-14 we get:

$$\phi_{iN}^k = K^k b^k \bar{h}_{iN} = \frac{-Q_i}{2\pi l_i} \int_{\xi_1}^{\xi_2} \frac{1}{2} \ln[(x' - p)^2 + (y' - \xi)^2] d\xi, \qquad [B\text{-}16]$$

where $K^k$ is the permeability of fracture k,$b^k$ is the aperture, and $\bar{h}_{iN}$ is the average head on node i in both fracture k and the other fracture forming the intersection. This integral, which is of the same form as Equation B-7, is evaluated:

$$\phi_{iN}^k = \frac{-Q_i}{4\pi l_i} \left\{ (\xi - y' )^2 \ln[(x' - p)^2 + (y' - \xi)^2] - 2\xi + 2\,|\,x' - p\,|\ \tan^{-1}\left( \frac{\xi - y'}{|\,x' - p\,|} \right) \right\}_{\xi_1}^{\xi_2} \quad . \qquad [B\text{-}17]$$

The image source of a nonradial node is an arc. Construction of the image arc, in the same x'y' coordinate system, is shown in Figure B-4. Given any point $\xi$ on the node at a distance $r_N$ from the origin, the corresponding point, $\eta$, on the image arc is at a distance $r_I$ from the origin, where $r_I = a^2/r_N$. The two points can be expressed in polar coordinates as $(r_N,\theta)$ and $(r_I,\theta)$, where

$$\cos\theta = \frac{p}{r_{N.}}$$

The locus of points of the image arc can now be expressed as

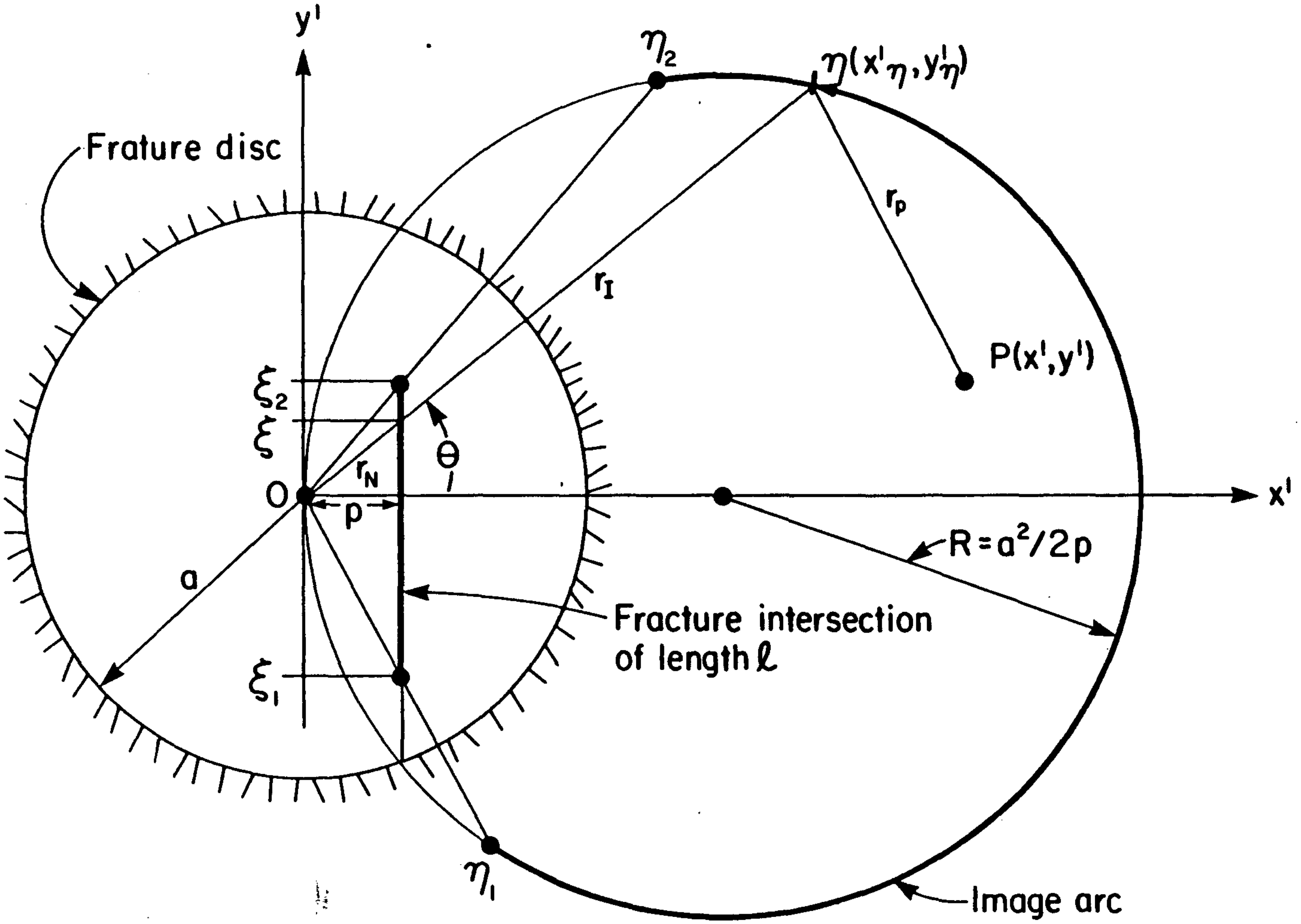
$$r_I = \frac{a^2}{r_N} = \frac{a^2}{p} \cos\theta \quad . \qquad [B\text{-}18]$$

This is the equation of a circle passing through (0,0), with radius $R = a^2/2p$ and center at $(a^2/2p,0)$. In rectangular coordinates, the equation of the circle containing the image arc is, therefore,

$$\left( x' - \frac{a^2}{2p} \right)^2 + y'^{\,2} = \frac{a^4}{4p^2} \quad . \qquad [B\text{-}19]$$

The potential due to sources with a total strength of $Q_i$ distributed along the image arc is given by Equation 2-5. The distance from a source at point $\eta$ $(x_\eta' ,y_\eta' )$ on the image arc to an arbitrary point P(x',y') in the plane, in terms of $\eta$ ,is

$$r_p(\eta) = \left[ (x' - x_\eta' )^2 + (y' - y_\eta' )^2 \right]^{1/2} \quad . \qquad [B\text{-}20]$$



XBL 855-10510

Figure B-4. Construction for calculating potential due to the image source of a nonradial node

However, to simplify the integration to the form given in Equation 2-6, we need $r_p$ in terms of $\xi$. From Figure B-4 it can be seen that

$$\frac{x_\eta'}{r_I} = \frac{p}{r_N} \quad \text{and} \quad \frac{y_\eta'}{r_I} = \frac{\xi}{r_N} \quad .$$

But, $r_I = a^2/r_N$ and $r_N = \sqrt{p^2 + \xi^2}$, therefore,

$$x_\eta' = \frac{a^2 p}{p^2 + \xi^2} \quad \text{and} \quad y_\eta' = \frac{a^2 \xi}{p^2 + \xi^2} \quad . \qquad [B\text{-}21]$$

Substituting these in Equation B-19 gives

$$r_p(\xi) = \left[\left(x' - \frac{a^2 p}{p^2 + \xi^2}\right)^2 + \left(y' - \frac{a^2 \xi}{p^2 + \xi^2}\right)\right]^{1/2} ,$$

that is,

$$r_p^2(\xi) = \frac{(x'^2 + y'^2)(p^2 + \xi^2) - 2a^2(x' p + y' \xi) + a^4}{p^2 + \xi^2} \quad . \qquad [B\text{-}22]$$

Now, we can use Equation 2-6 and write

$$\phi_{iI}^k = K^k b^k \bar{h}_{iI} = \frac{-Q_i}{4\pi l_i}\left\{\int_{\xi_1}^{\xi_2} \ln(\Omega) d\xi - \int_{\xi_1}^{\xi_2} \ln(\xi^2 + p^2) d\xi\right\} \qquad [B\text{-}23]$$

where

$$\begin{aligned} \Omega &= \gamma\xi^2 + \beta\xi + \alpha \\ \gamma &= x'^2 + y'^2 \\ \beta &= -2a^2 y' \\ \alpha &= (x'^2 + y'^2)p^2 - 2a^2 x' p + a^4 \end{aligned} \qquad [B\text{-}24]$$

These integrals are also of the same form as Equation B-7 and are evaluated:

$$\phi_{iI}^k = \frac{-Q_i}{4\pi l_i}\left\{[\xi + \frac{\beta}{2\gamma}]\ln(\Omega) - 2\xi + \frac{\sqrt{4\alpha\gamma - \beta^2}}{\gamma}\tan^{-1}\left(\frac{2\gamma\xi + \beta}{\sqrt{4\alpha\gamma - \beta^2}}\right)\right.$$
$$\left. - \left[\xi\ln(\xi^2 + p^2) - 2\xi + 2p\tan^{-1}\left(\frac{\xi}{p}\right)\right]\right\}_{\xi_2}^{\xi_2} \qquad [B\text{-}25]$$

Substituting the limits $\xi_1$ and $\xi_2$ in Equations B-16 and B-17 gives expressions for $\phi_{iN}^k$ and $\phi_{iI+}^k$ in terms of $Q_i$, x' and y'. Thus, for any nonradial node we can write an expression for potential due to the presence of that node, i, in fracture k, using $\phi_{iI-}^k$ from Equation

B-3:

$$
\begin{aligned}
\phi_i^k = \phi_{iN}^k + \phi_{iI+}^k + \phi_{iI-}^k &= Q_i\, f_{iN}^k\,(x' \ ,y' \ ) + Q_i\, f_{iI+}^k\,(x' \ ,y' \ ) + Q_i\, f_{iI-}^k\,(x' \ , y' \ ) \\
&= Q_i\,[f_{iN}^k\,(x' \ ,y' \ ) + f_{iI+}^k\,(x' \ ,y' \ ) + f_{iI-}^k\,(x' \ , y' \ )] \\
&= Q_i\, F_i^k\,(x' \ ,y' \ ) \qquad \text{[B-26]}
\end{aligned}
$$

where $Q_i$ is the total strength of the line source and $F_i^k = f_{iN}^k + f_{iI+}^k + f_{iI-}^k$. It should be noted that the x'y' coordinate system of the nonradial node is different from that of the radial node used in Equation B-13.

Subroutine NRADIAL determines $\lambda$, $\mu$, p, $\xi_{1,}$ and $\xi_{2,}$ and calls INTEGFN to compute potentials.

# APPENDIX C

## METHOD FOR APPROXIMATING SHAPE FUNCTION $\Upsilon_{ij}^k$

The shape function $\Upsilon_{ij}^k$ represents the average head on node j in fracture k due to node i, also in fracture k. For each node i, j takes on values from 1 to n, where n is the number of nodes on fracture k. Evaluation of $\Upsilon_{ij}^k$, as given in Equation 2-14, involves integrating shape functions $f_{iN}^k$, $f_{iI+}^k$ and $f_{iI-}^k$ along node j. This would be difficult analytically, but it can be easily approximated by evaluating $f_{iN}^k$, $f_{iI+}^k$ and $f_{iI-}^k$ at a finite number of points on node j and averaging the results. The number of points on node j that are used for the approximation, mp [ninteg] is determined by the user. The coordinate system used for evaluating $\Upsilon_{ij}^k$, for a given i and j = 1 to n, is the same as that used for obtaining expressions for $\phi_{iN}^k$ and $\phi_{iI}^k$, which depends on the node type of intersection i (Appendix B). Since the same coordinate system is used for each node j, and relative distances are being integrated, functions $f_{iN}^k$ and $f_{iI}^k$ need not be transformed back to the local fracture xy coordinate system before obtaining $\Upsilon_{ij}^k$. For discussion, we will assume an x′ y′ system of either a radial or nonradial node i. The endpoints of node j are rotated to that system, becoming $(x_1' , y_1' )$. To divide this line segment into mp pieces, a $\Delta x'$ and $\Delta y'$ are computed:

$$\Delta x' = \frac{x_2' - x_1'}{mp} \text{ and } \Delta y' = \frac{y_2' - y_2'}{mp} \quad . \qquad [C-1]$$

The points used to evaluate the potential functions appropriate to the type of node i, $f_{iN}^k(x' , y' )$, $f_{iI+}^k(x' , y' )$, and $f_{iI-}^k(x' , y' )$, are

$$x_m' = x_1' + (m - 0.5)\,\Delta x'$$

and

$$y_m' = y_1' + (m - 0.5)\Delta y' \ , \qquad [C-2]$$

for m = 1 mp.

The shape function $\Upsilon_{ij}^k$ is then approximated:

$$\Upsilon_{ij}^k = \frac{1}{l_j}\int_{l_j} F_i^k \, dl_j \approx \frac{1}{mp}\sum_{m=1}^{mp}\left[ f_{iN}^k(x_m' , y_m' ) + f_{iI+}^k(x_m' , y_m' ) + f_{iI-}^k(x_m' , y_m' ) \right] \qquad [C-3]$$

where $(x_m', y_m')$ is a point on node j and mp is the number of equally spaced points selected on node j. If node j is a point node, then mp = 1 and the node itself is used in Equation C-3.

In most of the preceding discussion and calculations, we have been considering nodes as line segments. However, a fracture intersection in a plane is two-dimensional, that is, it has width, as computed in Equation A-5. A more precise expression than that given in Equation 2-14 for $\Upsilon_{ij}^k$,

$$\Upsilon_{ij}^k = \frac{1}{P_j} \int_{P_j} F_i^k \, dP_j \qquad [C\text{-}4]$$

where $P_j = 2l_j w_j$, the perimeter of node j. The approximation method we are using could take mp points on each side of the node and average the resulting values of $f_{iN}^k$, $f_{iI+}^k$, and $f_{iI-}^k$. But, since the width of the node is relatively small, based on fracture aperture, mp points on the mid-line of the node are considered sufficient for approximating $\Upsilon_{ij}^k$.

There is an exception to the use of Equation C-3 that uses the intersection width. When j equals i the contribution of the intersection itself, $f_{iN}^k$, cannot be obtained in the same way because the potential due to the source is not defined on the source itself. In expressions for $f_{iN}^k$, we let points $(x', y')$ lie on the perimeter of the node and integrate directly. If we let $bm_i$ equal half the intersection width, $w_i$, then points $(x', y')$ are on line segments of length $l_i$ that are parallel to node i at distances + and - $bm_i$ from the midline. Integrations using $+bm_i$ and $-bm_i$ would be averaged, but since they yield the same value, the integration uses only $+bm_i$. Therefore, we have

$$\frac{1}{l_i} \int_{l_i} f_{iN}^k \, dl_i = \frac{-1}{4\pi l_i^2} \left\{ (l_i^2 - bm_i^2) \ln (l_i^2 + bm_i^2) + bm_i^2 \ln (bm_i^2) \right.$$
$$\left. - 3l_i^2 + 4l_i \, bm_i \tan^{-1} \left( \frac{l_i}{bm_i} \right) \right\}, \qquad [C\text{-}5]$$

and

$$\Upsilon_{ii}^k \approx \frac{1}{l_i} \int_{l_i} f_{iN}^k \, dl_i + \frac{1}{mp} \sum_{m=1}^{mp} [\, f_{iI+}^k (x_m', y_{m'}) + f_{iI-}^k (x_m', y_m') \,], \qquad [C\text{-}6]$$

for a radial or nonradial node i. If node i is a point node, its length is zero but its radius is $bm_i$, therefore, $\Upsilon_{ii}^k$ can be expressed:

$$\Upsilon_{ii}^k = \frac{-1}{4\pi}\left\{ \ln(bm_i^2) + \ln\left[(x_1-x_I)^2 + (y_1-y_I)^2\right] - \ln(x_1^2 + y_1^2) \right\}, \qquad [C-7]$$

where point $(x_{1,}\ y_1)$ is the node and $(x_{I,}\ y_I)$ is the image source point.

Thus, for each node i in fracture k, where i = 1 to n, the appropriate functions $f_{iN}^k$, $f_{iI+}^k$ and $f_{iI-}^k$ are determined, then, for each node j in fracture k, where j = 1 to n, the corresponding $\Upsilon_{ij}^k$ is approximated. The result is an n by n matrix of shape functions, each representing the average head on node j due to the presence of node i.

The subroutines that compute approximations of $\Upsilon_{ij}^k$ are INTEGFP, INTERFR, and INTERFN which are called by subroutines POINT, RADIAL, and NRADIAL, respectively. Each of these subroutines sets up a coordinate transformation and computes other required parameters as determined by the geometric type of node i before calling the specific integration subroutine.

# APPENDIX D.

# METHOD FOR COMPUTING THE AREA OF A FRACTURE TRUNCATED AT THE GENERATION REGION BOUNDARY

For fractures extending beyond the spherical generation region, the truncated area is computed as follows:

Given

- the radius of the generation sphere, $r_g$ ,
- the coefficients of the equation of the fracture plane, a, b, c, and d, (see Equation 3-15),
- the fracture radius, $r_f$ ,
- the coordinates of the fracture center, $(x_c, y_c, z_c)$ .

1. The generation sphere, $x^2 + y^2 + z^2 = r_g^2$, is intersected by the plane of fracture f, $ax + by + cz + d = 0$, forming a circle, S. The center and radius, $r_s$, of that circle can be determined since the perpendicular distance between the origin and the plane is d and that perpendicular must intersect the center of circle S. As shown in Figure D-1, $r_g$, $r_s$ , and d form a right triangle, therefore

$$r_s = (r_g^2 - d^2)^{1/2} \quad . \tag{D-1}$$

The center of circle S $(x_s, y_s, z_s)$ lies at a distance -d from the origin on a line with direction cosines a, b, c. Therefore,

$$a = \frac{x_s - 0}{-d}, \quad b = \frac{y_s - 0}{-d}, \quad c = \frac{z_s - 0}{-d}, \tag{D-2}$$

and the coordinates of the center of circle S are (-ad, -bd, -cd).

2. The problem is now reduced to the two-dimensional determination of what portion, if any, of fracture f, with center $(x_c, y_c, z_c)$ and radius $r_f$ , lies outside circle S.

The area to be truncated can be determined using the radii of the circles, $r_f$ and $r_s$, and the distance between the centers of the circles, $d_f$ , where

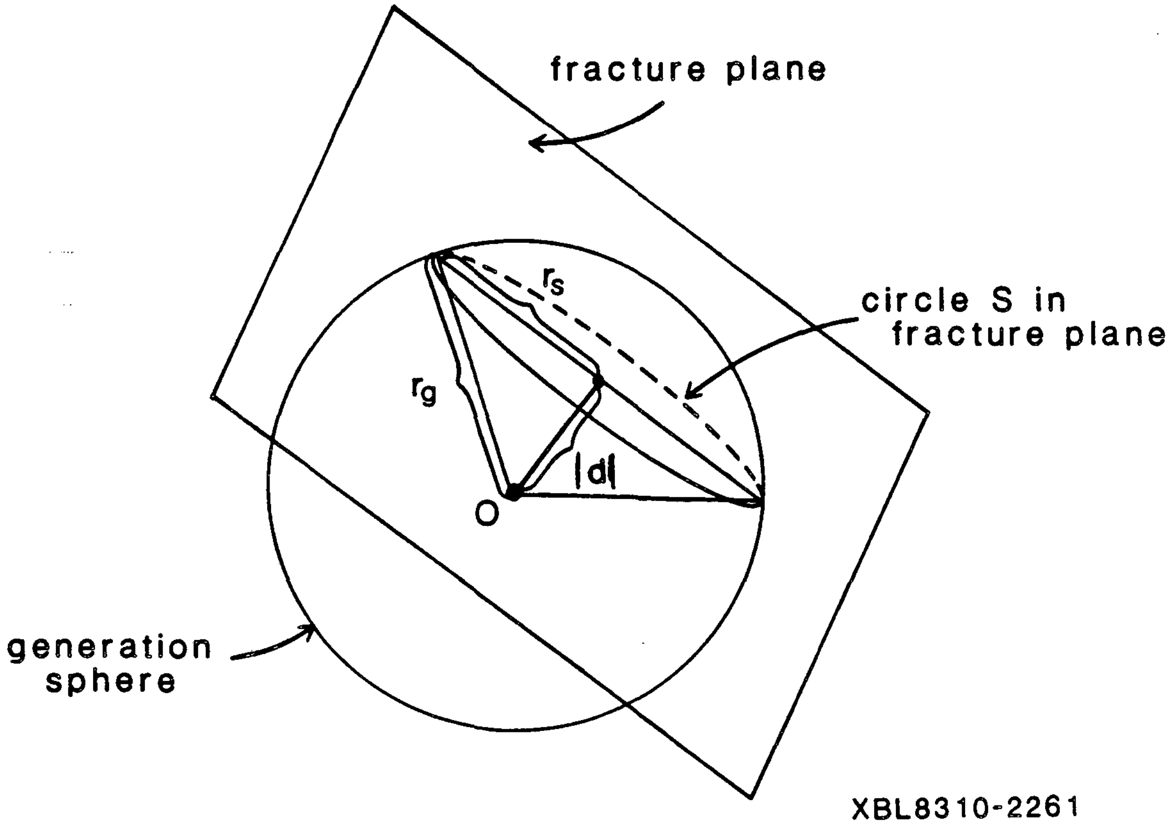


Figure D-1. Intersection of fracture plane and spherical generation region

$$d_f = AC = [(x_c + ad)^2 + (y_c + bd)^2 + (z_c + cd)^2]^{1/2} \ . \qquad \text{(D-3)}$$

Figure D-2 illustrates the possible relationships between the two circles and the conditions for truncation. If $(d_f + r_f) \leq r_s$, as in Example 1, then the entire fracture lies within the boundary. If $(d_f + r_f) > r_s$, then the fracture extends beyond the boundary and the area to be truncated is computed.

In Example 2, the fracture circle encompasses the boundary circle, there fore the truncated fracture area is equal to the area of circle S.

In Examples 3 and 4, the area to be truncated, $A_t$ , is computed as shown in Figure D-3. $A_t$ is the area of the sector of circle f of angle $\gamma$ minus the area of triangle BCD minus the area of the sector of circle S of angle $\alpha$ plus the area of triangle ABD. To find $\gamma$, consider triangle ACD in which, by the law of cosines,

$$r_s^2 = d_f^2 + r_f^2 - 2d_f r_f \cos\beta \ . \qquad \text{(D-4)}$$

Therefore,

$$\beta = \cos^{-1}\left(\frac{d_f^2 + r_f^2 - r_s^2}{2d_f r_f}\right) \qquad \text{(D-5)}$$

and

$$\gamma = 360 - 2\beta \ . \qquad \text{(D-6)}$$

The angle $\alpha$ is determined by the same triangle, ACD, in which

$$r_f^2 = d_f^2 + r_s^2 - 2d_f\, r_s \cos\frac{\alpha}{2} \ . \qquad \text{(D-7)}$$

Therefore

$$\alpha = 2\cos^{-1}\left(\frac{d_f^2 + r_s^2 - r_f^2}{2d_f\, r_s}\right) \ . \qquad \text{(D-8)}$$

Then the area to be truncated is

$$A_t = \left(\pi r_f^2 \frac{\gamma}{360} - \frac{1}{2} r_f^2 \sin\gamma\right) - \left(\pi r_s^2 \frac{\alpha}{360} - \frac{1}{2} r_s^2 \sin\alpha\right) \qquad \text{(D-9)}$$

which, by factoring and converting to radians, becomes

$$A_t = \frac{r_f^2}{2}(\gamma - \sin\gamma) - \frac{r_s^2}{2}(\alpha - \sin\alpha) \ . \qquad \text{(D-10)}$$

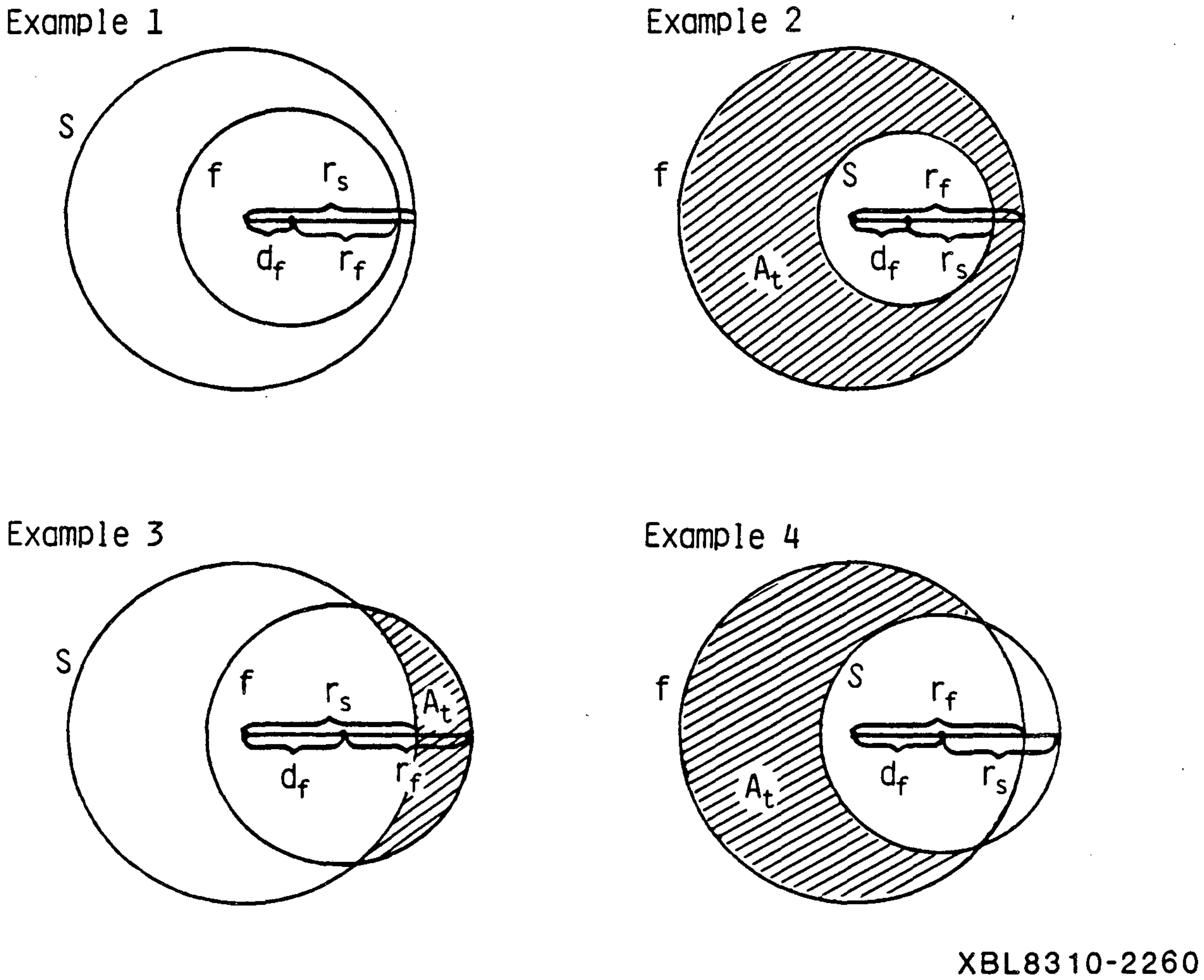


Figure D-2. Possible relationship between fracture circle f and boundary circle S

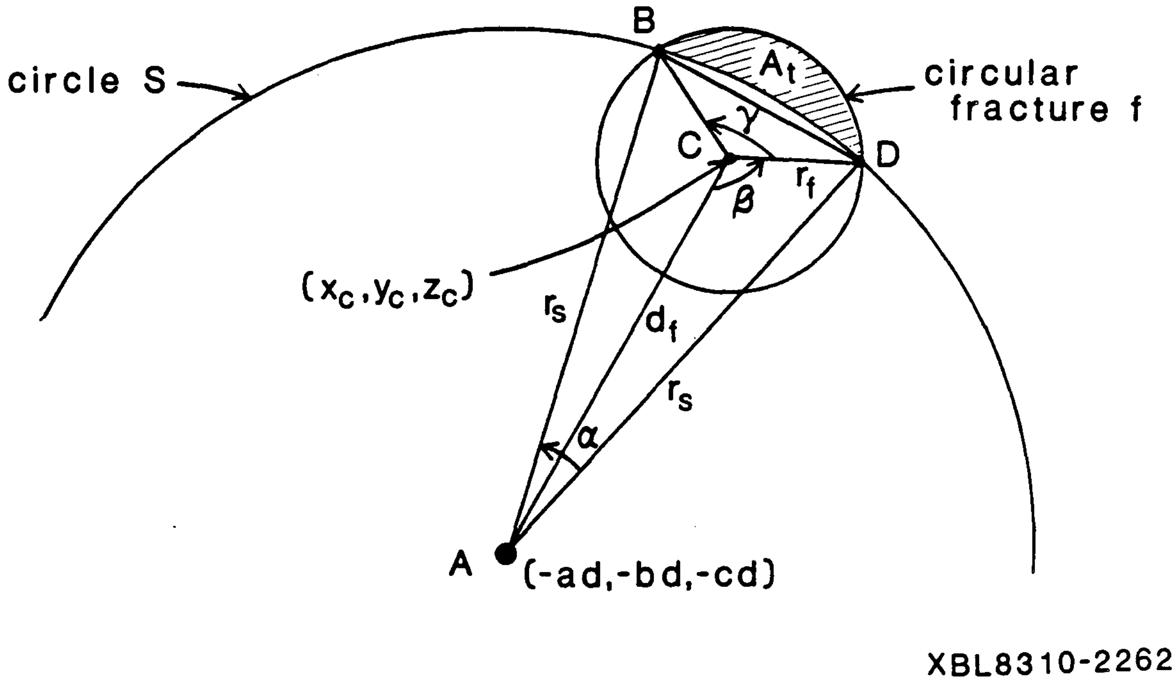


Figure D-3. Computing area of fracture to be truncated at boundary of generation region

# APPENDIX E.

# METHOD OF DETERMINING THE POSITION OF A CIRCULAR FRACTURE RELATIVE TO TWO PARALLEL BOUNDARY LINES

The plane of a particular fracture intersects the six flow region boundary planes along three pairs of parallel boundary lines (or two pairs if the fracture plane is parallel to a flow region face). The first step in determining whether or not a circular fracture is within the flow region is to determine the position of the disc relative to each pair of parallel boundary lines in the fracture plane.

Given, in the fracture plane z' = -d, a circular fracture with radius $r_f$, center (k,0), and two parallel boundary lines, $L_1$ and $L_2$ , with equations

$$a_i' x' + b_i' y' - c_i' d - d_i = 0$$

and

$$a_i' x' + b_i' y' - c_i' d + d_i = 0 \qquad \text{(E-1)}$$

respectively. Equations E-1 are derived from the rotated boundary plane equations given in Section 3.1.3, Equation 3-20,

$$a_i' x' + b_i' y' + c_i' z' + d_i = 0 \ , \ (\text{for } i = 1,6) \ . \qquad \text{(E-2)}$$

The method of determining the position of the fracture circle relative to a pair of parallel lines, $L_1$ and $L_2$ , is shown in Figures E-1 and E-2. The perpendicular distances, $D_1$ and $D_2$, from the center of the fracture circle, (k, 0), to the boundary lines $L_1$ and $L_2$ and the perpendicular distance D between $L_1$ and $L_2$ are computed:

$$\begin{aligned}
D &= \frac{|\,(-c_i' d + d_i) -(-c_i' d - d_i)\,|}{\sqrt{a_i'^{\,2} + b_i'^{\,2}}} = \frac{|\,2d_i\,|}{\sqrt{a_i'^{\,2} + b_i'^{\,2}}} \\
D_1 &= \frac{|\,(-c_i' d - d_i) - (a_i' k)\,|}{\sqrt{a_i'^{\,2} + b_i'^{\,2}}} \qquad \text{(E-3)} \\
D_2 &= \frac{|\,(-c_i' d + d_i) - (-a_i' k)\,|}{\sqrt{a_i'^{\,2} + b_i'^{\,2}}} \ .
\end{aligned}$$

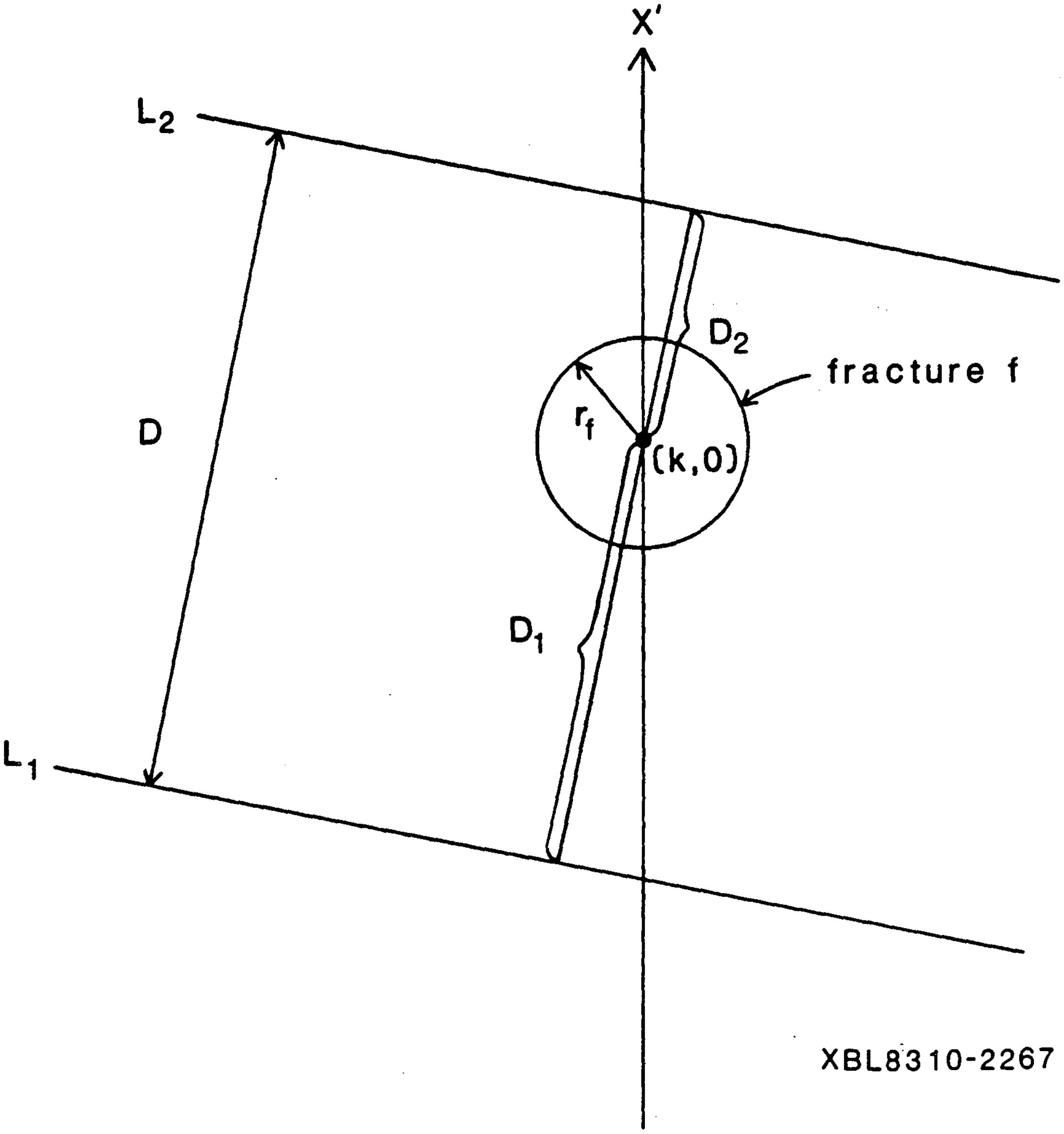


Figure E-1. Method of defining the position of circular fracture f relative to a pair of parallel boundary lines, $L_1$ and $L_2$, in the $z' = -d$ plane

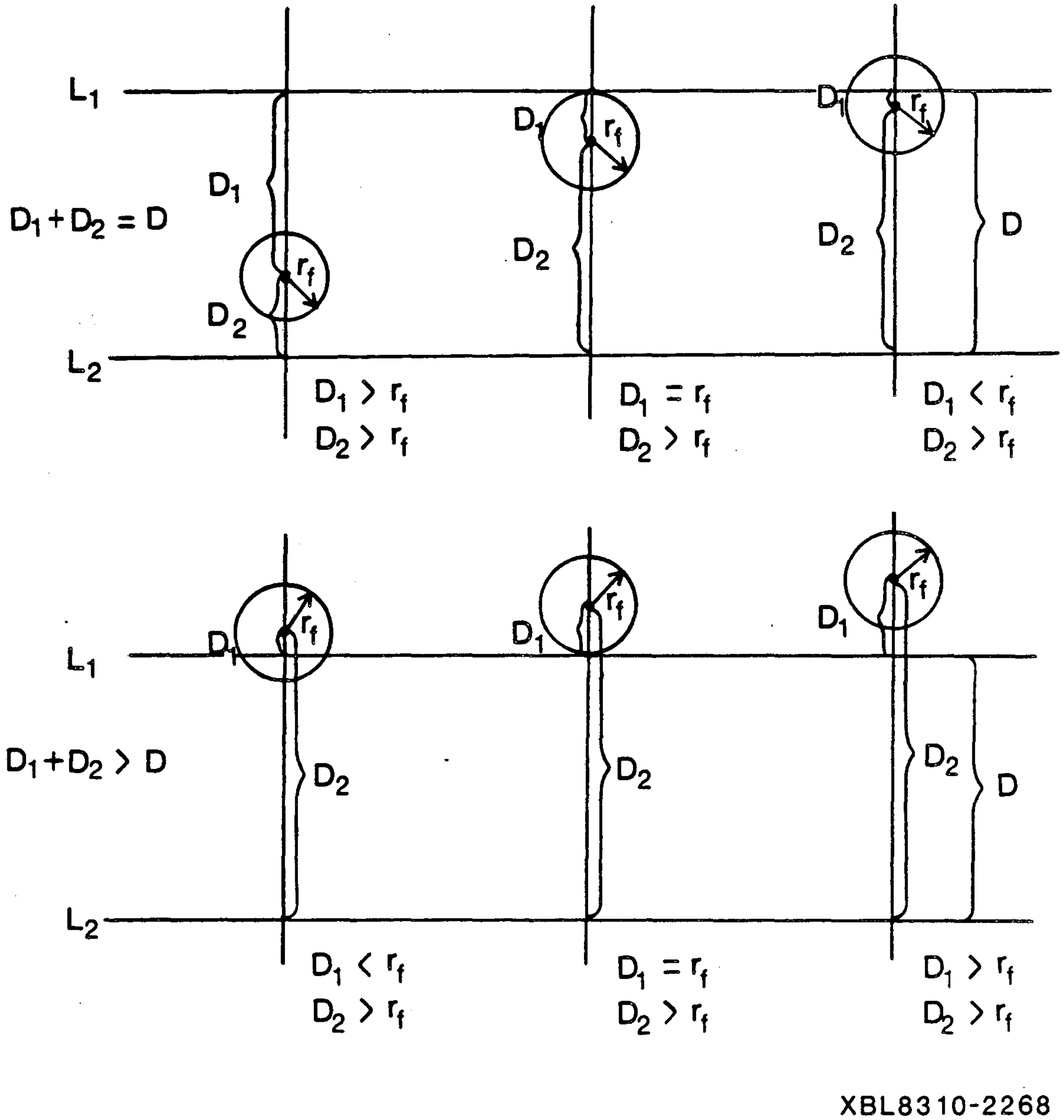


Figure E-2. Examples of various possibilities of fracture position relative to parallel boundary lines

As shown in Figure E-2, if $D_1 + D_2 = D$, the fracture center lies between the boundary lines. If in addition $D_1 > r_f$ and $D_2 > r_f$, the fracture circle does not intersect either boundary. If $D_1 = r_f$, (or $D_2 = r_2$) the circle intersects the boundary at only one point and is flagged to indicate this. If $D_1 < r_f$ (or $D_2 < r_f$) the fracture partially extends beyond the boundary and is flagged for truncation. (A fracture is not truncated until it has been compared with all three pairs of boundary lines; it may lie within the first two pairs but outside the third and therefore be discarded.) If, however, $D_1 + D_2 > D$ , the fracture center lies outside the boundary lines and the fracture will be discarded if $D_1 \geq r_f$ and $D_2 \geq r_f$. If $D_1 < r_{f,}$ (or $D_2 < r_f$) , the fracture lies partially inside the boundary and is flagged for truncation.

If $a_i' = b_i' = 0$ for any i in boundary plane Equations E-2, then that boundary plane and its parallel boundary plane, j, are parallel to the fracture plane z' = -d. The equations of planes i and j will be

$$z' = \frac{-d_i}{c_i'} \quad \text{and} \quad z' = \frac{+d_i}{c_i'} \ , \tag{E-4}$$

(since $a_i' = a_j'$ , $b_i' = b_j'$ , $c_i' - c_j'$ , and $d_i = -d_j$ for parallel planes 1 and 3, 2 and 4, and 5 and 6). The relative positions of the three planes are determined as follows:

$$\text{If} \quad -d = \frac{-d_i}{c_i'} \quad \text{or} \quad -d = \frac{+d_i}{c_i'} \ , \tag{E-5}$$

the fracture plane coincides with boundary plane i or j, respectively. The fracture is flagged for discarding since it behaves as part of the boundary.

$$\text{If} \quad \left| \frac{-d_i}{c_i'} + d \right| + \left| \frac{d_i}{c_i'} + d \right| = \left| \frac{-2d_i}{c_i'} \right| \ , \tag{E-6}$$

then the fracture plane lies between the boundary planes i and j.

$$\text{If} \left| \frac{-d_i}{c_i'} + d \right| + \left| \frac{d_i}{c_i'} + d \right| > \left| \frac{-2d_i}{c_i'} \right| \ , \tag{E-7}$$

then the fracture lies outside the pair of boundary planes and is therefore flagged for discarding.

# APPENDIX F
# METHOD FOR COMPUTING THE AREA OF A FRACTURE TRUNCATED AT THE FLOW REGION BOUNDARY

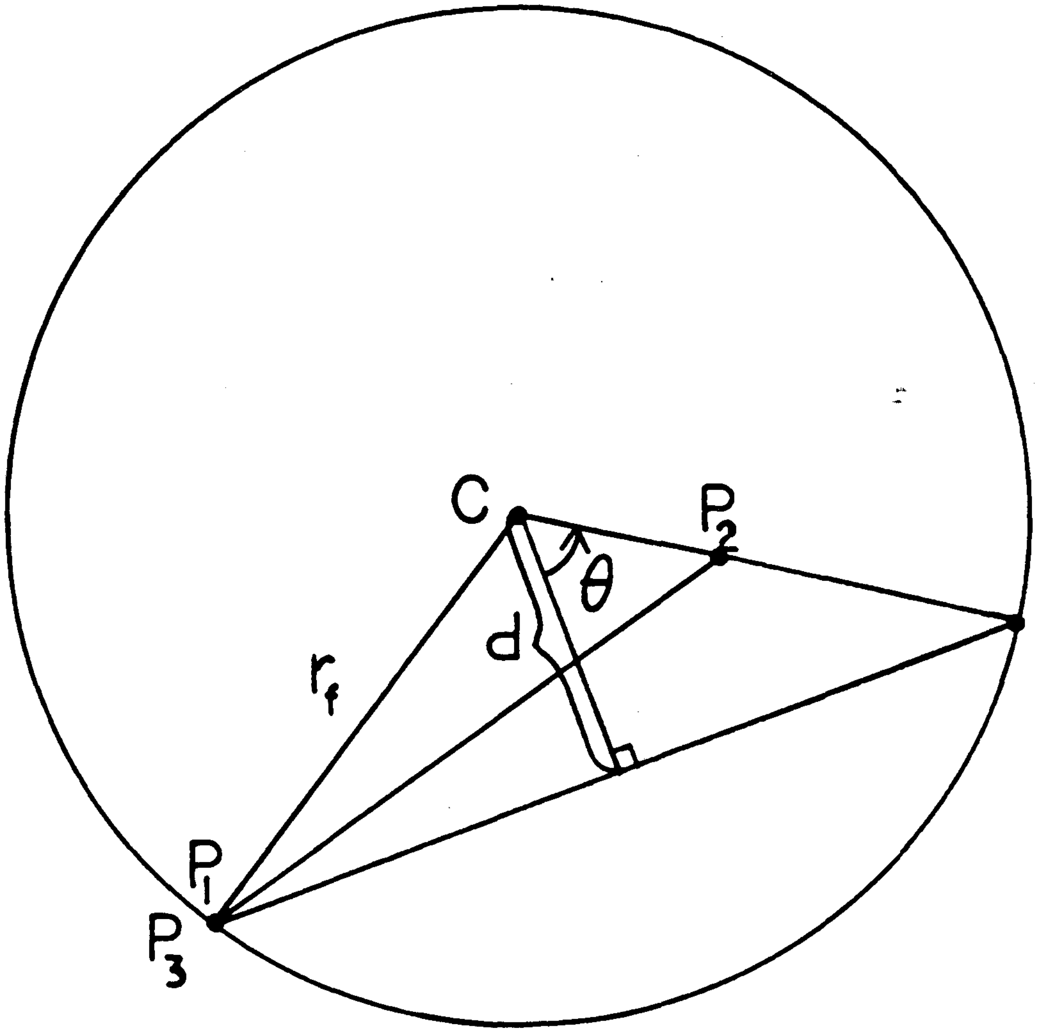
For any fracture containing boundary nodes, the area of the fracture lying outside the flow region is calculated and subtracted from the area of the fracture circle. The area to be truncated is computed separately for each boundary node and the method used depends on the position of the fracture center relative to the boundary line containing the node. The truncation flag [ntrunc(i), i=1,6] set for each boundary line by subroutine FLAG, is greater than one for a boundary line containing a boundary node with a length greater than zero and equals 2, 3, or 4 to indicate the center is inside, outside or on the boundary line, respectively.

Computation of the truncated area for each node follows the general procedure diagrammed in Figure F-1 with variations for flag = 2, 3 and 4 diagrammed in Figure F-2. The endpoints of the node are P1 and P2. If an endpoint is not on the circle, subroutine CIRCPT determines the point at which a line from the circle center, C, through the endpoint intersects the circle. This point is P3 for endpoint P1, and P4 for endpoint P2. If a nodal endpoint is on the circle, then P3 = P1, or P4 = P2. These four points [bxy(4,2)] are used to compute the area of truncation for a node on side i as follows:

Given:

- a circle with center C = (k,0) and radius $r_f$ , points P1 = $(x_1,y_1)$ and P2 = $(x_2,y_2)$ either inside or on the circle,
- points P3 = $(x_3,y_3)$ and P4 = $(x_4,y_4)$ on the circle.

Compute:



XBL8310-2273

Figure F-1. General procedure for calculating area of truncation

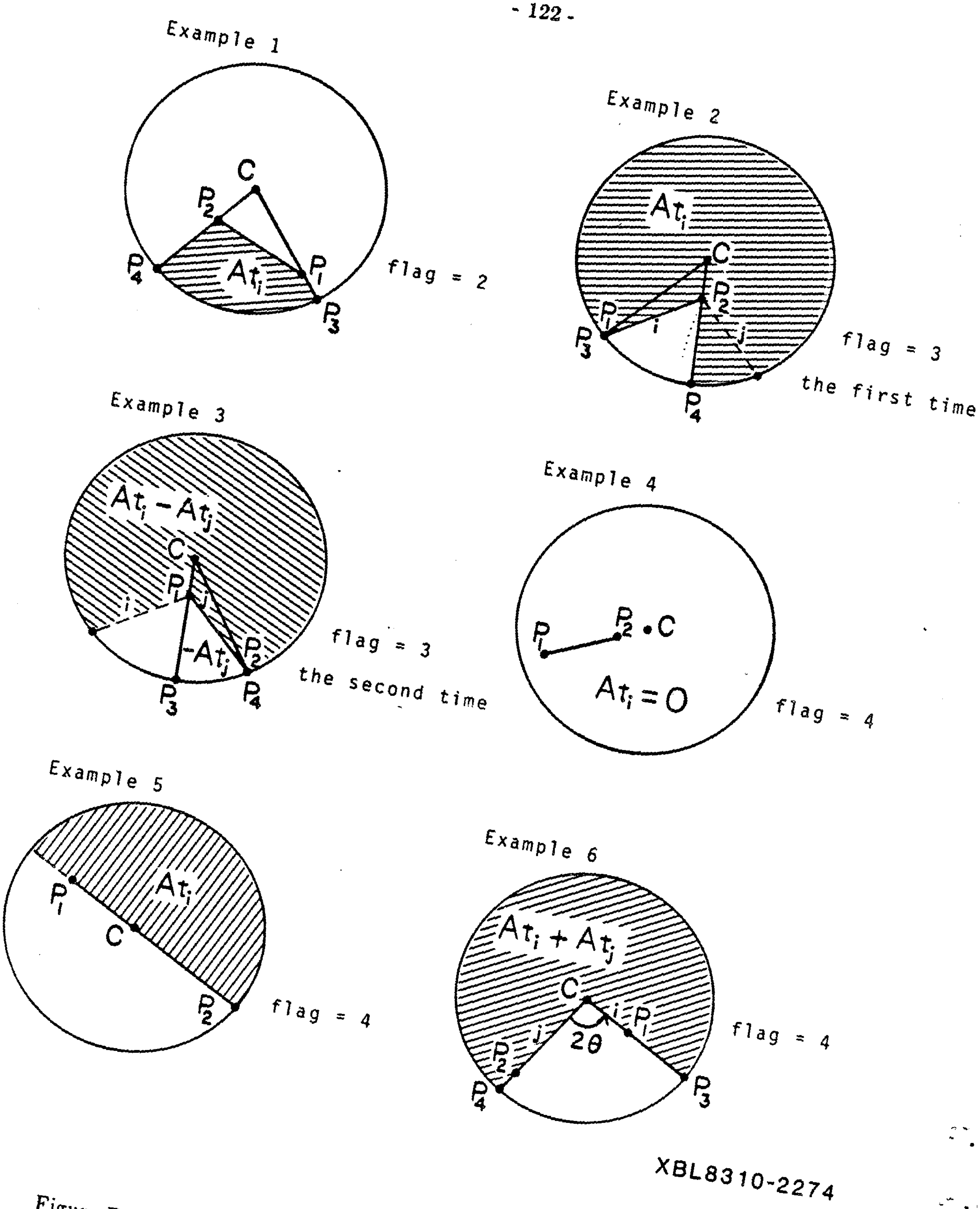


Figure F-2. Examples of determining the areas of truncation

- the area of sector P3-C-P4, Asect
- the area of triangle P1-C-P2, Atri
- the area of truncation for side i, $At_i$.

The coefficients of the equation of line P3-P4 are

$$\begin{aligned} a &= y_4 - y_3 \\ b &= x_3 - x_4 \\ c &= x_4 y_3 - x_3 y_4 \ . \end{aligned} \tag{F-1}$$

The perpendicular distance from line ax + by + c = 0 to point C is

$$d = \frac{ak + c}{\sqrt{a^2 + b^2}} \ . \tag{F-2}$$

The angle of sector P3-C-P4 is

$$2\theta = 2 \cos^{-1} (d/r_f) \ . \tag{F-3}$$

The area of sector P3-C-P4 is

$$\mathrm{Asect} = \frac{2\theta}{2\pi} \pi r_f^2 = \theta r_f^2 \ . \tag{F-4}$$

And the area of triangle P1-C-P2 is

$$\mathrm{Atri} = 1/2 \ | y_2(x_1 - k) + y_1(k - x_2) | \ . \tag{F-5}$$

1. If the center is inside the boundary line containing the node [ntrunc(i) = 2] the area to be truncated is

$$At_i = \mathrm{Asect} - \mathrm{Atri} \ , \tag{F-6}$$

as shown in Figure F-2, Example 1.

2. If the center is outside the boundary line containing the node [ntrunc(i) = 3] the area to be truncated is first

$$At_i = \pi r_f^2 - \mathrm{Asect} + \mathrm{Atri} \ , \tag{F-7}$$

(Figure F-2, Example 2) and

$$At_i = \mathrm{Atri} - \mathrm{Asect} \tag{F-8}$$

any succeeding times the code equals three for boundary nodes on the given fracture. In most cases, this actually adds back part of the area subtracted by Equation F-7 as shown in Figure F-2, Example 3.

3. If the fracture center is on the boundary line, i, containing the node [ntrunc(i) = 4] then the computation depends on whether the endpoints of the node are on the same side of the center, on opposite sides of the center or one endpoint is coincident with the center.

   If P1 and P2 are on the same side, Figure F-2, Example 4, then at least one of the endpoints is inside the circle and truncation will be taken care of by calculations for the node(s) sharing the endpoint(s),

$$At_i = 0 \ . \qquad \text{(F-9)}$$

If P1 and P2 are on opposite sides of center, Example 5, then half the circle is truncated,

$$At_i = 1/2 \ \pi r_f^2 \ . \qquad \text{(F-10)}$$

If one endpoint is on the center, Example 6, then another node, on side j, will also have an endpoint on the center. For the node on side i, the endpoint not on the center is P1 and for the node on side j, the endpoint not on the center is P2. P3 and P4 are found as in the general case and the area of the sector, P3-C-P4, is computed. The area to be truncated is the remainder of the circle,

$$At_i + At_j = \pi r_f^2 - \text{Asect} \ . \qquad \text{(F-11)}$$

When areas of truncation have been computed for all sides i with truncation codes greater than 1, (where i = 1,6), the areas are summed and subtracted from the original fracture area to get the truncated area of fracture f,

$$A_f = \pi r_f^2 - \sum_{i=1}^{6} At_i \ . \qquad \text{(F-12)}$$

Figure F-3 illustrates the steps in determining the truncated fracture area with an example of a fracture containing four bondary nodes on sides 1, 3, 4 and 5. The boundary lines are looked at in sequence, 1 through 6, and for those with truncation codes greater than one, the area of truncation is computed. All others, in this case sides 2 and 6, have areas of truncation equal to zero. Truncations on sides 1 and 4, Steps 1 and 3 in the

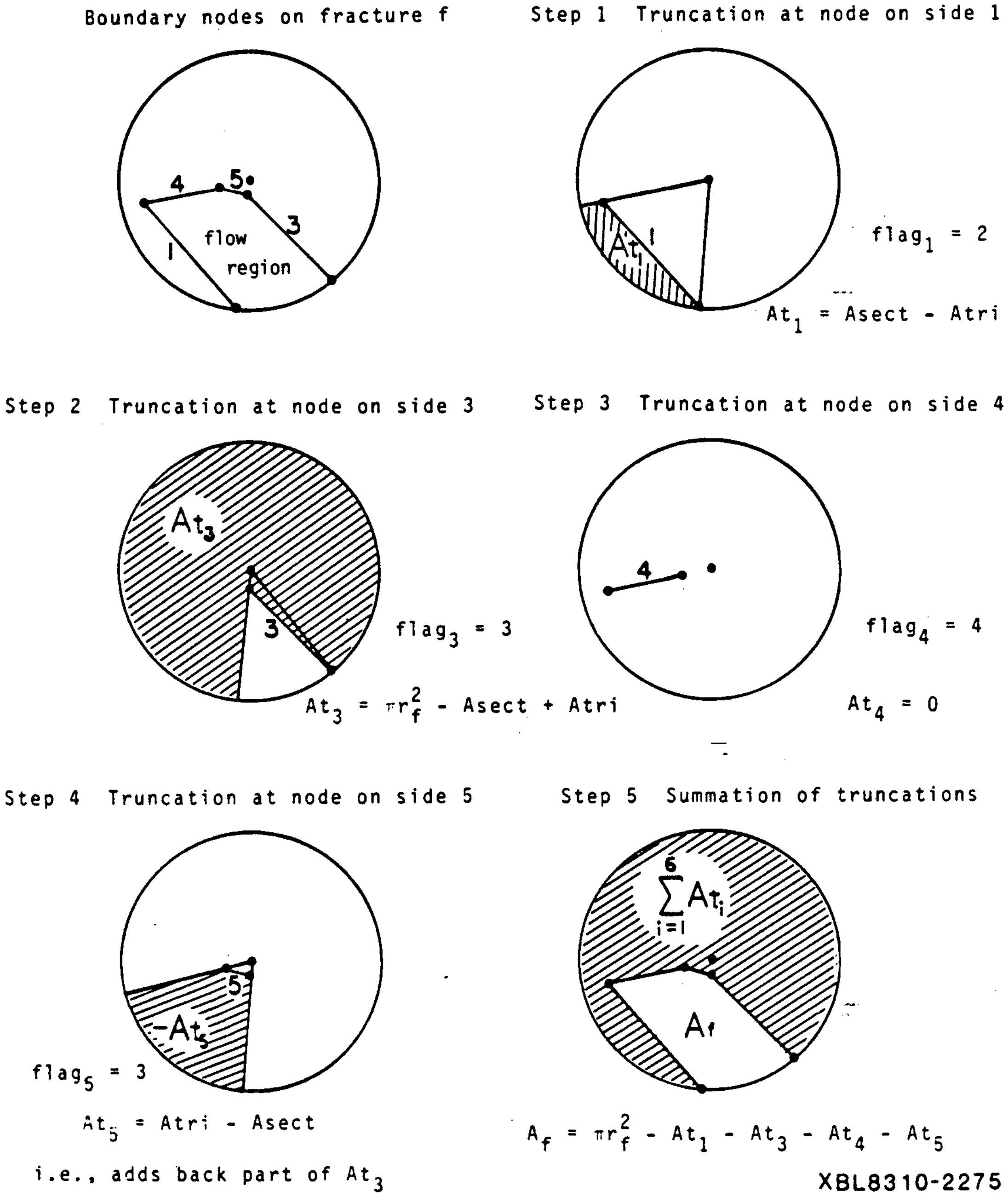

Boundary nodes on fracture f
4
5
3
1
flow region
Step 1 Truncation at node on side 1
$At_1$
1
$flag_1 = 2$
$At_1$ = Asect - Atri
Step 2 Truncation at node on side 3
$At_3$
3
$flag_3 = 3$
$At_3 = \pi r_f^2$ - Asect + Atri
Step 3 Truncation at node on side 4
4
$flag_4 = 4$
$At_4 = 0$
Step 4 Truncation at node on side 5
5
$-At_5$
$flag_5 = 3$
$At_5$ = Atri - Asect
i.e., adds back part of $At_3$
Step 5 Summation of truncations
$\sum_{i=1}^{6} At_i$
$A_f$
$A_f = \pi r_f^2 - At_1 - At_3 - At_4 - At_5$
XBL8310-2275

Figure F-3. Example of truncation of a fracture with multiple boundary nodes figure, are straightforward. In Step 2, the truncated area of side 3 overlaps the truncated area of side 1 and part of the flow region. However, in Step 4 the area of truncation, $At_5$ , is actually added back rather than subtracted, thus yielding the correct area of truncation when the $At_i$' s are summed in Step 5.

# APPENDIX G
# METHOD FOR DETERMINING FRACTURE INTERSECTIONS AND TRUNCATION OF INTERSECTIONS AT FLOW REGION BOUNDARY

When comparing fractures to a particular fracture, f, the first step is to rotate the axes to a convenient coordinate system. Then, it must be determined whether each fracture j intersects the plane of fracture f and if so, whether that line of intersection intersects circular fracture f. If the fractures intersect, the point or line segment of intersection is determined and intersections extending beyond flow region boundaries are truncated. Finally, all nodal endpoints are returned to the global coordinate system.

*Rotation of axes.* Subroutine ROFRAC is called for fracture f in the plane ax + by + cz + d = 0, with center ($x_c$ $y_{c,}$ $z_c$) and radius $r_f$, to generate the same rotation [rotf] as was used to truncate and discard fractures in the flow region (Sections 3.1.3 and 3.2.2). This rotation simplifies the equation of the plane of f to z' = -d and the coordinates of the center to (k, 0, -d). Subroutine RFLOFRC uses the rotation matrix, as in Section 3.1.3, to determine the flow region in the fracture plane yielding boundary line Equations 3-21. And subroutine ROJTOF applies the same rotation from the global XYZ system to the local X'Y'Z' system to any fracture, j, being tested for intersection with f. Fracture j now lies in plane $a_j'$ x + $b_j'$ y + $c_j'$ z + $d_j$ = 0, has center ($x_j'$ , $y_j'$ ,$z_j'$ ), and radius $r_j$ .

*Discarding of fractures too far away.* To speed up the process, before establishing the exact equations for intersection, the distance between the plane of fracture f and the center of fracture j is computed. If this distance is greater than the radius of fracture j, the program goes to the next fracture.

*Fracture intersections.* In order to determine the intersection, if any, between fractures f and j, first determine whether planes f and j intersect, then whether fracture j intersects plane f, and finally, whether fracture j intersects fracture f. This is done by subroutine INTFRC.

To determine the intersection of fracture j with plane f, the circular fracture j may be considered as the intersection of the plane of j with a sphere, j, which has center $(x_j', y_j', z_j')$ and radius $r_j$ :

$$(x' - x_j')^2 + (y' - y_j')^2 + (z' - z_j') = r_j^2 \ . \qquad \text{(G-1)}$$

If sphere j doesn't intersect plane f, i.e., z' = -d, then the circular fracture j doesn't intersect plane f. If sphere j intersects the plane z' = -d, that intersection is a circle, j' (or a point circle if the radius equals zero), as diagrammed in Figure G-1. The center of the circle of intersection is $(x_j', y_j', -d)$ and the radius $r_j'$ is computed:

$$r_j' = \sqrt{r_j^2 - (z_j' + d)^2} \ . \qquad \text{(G-2)}$$

The conditions of intersection are:

$$(z_j' + d)^2 \begin{cases} > r_j^2 & \text{no intersection,} \\ = r_j^2 & \text{intersection in a point circle,} \\ < r_j^2 & \text{intersection in a circle.} \end{cases} \qquad \text{(G-3)}$$

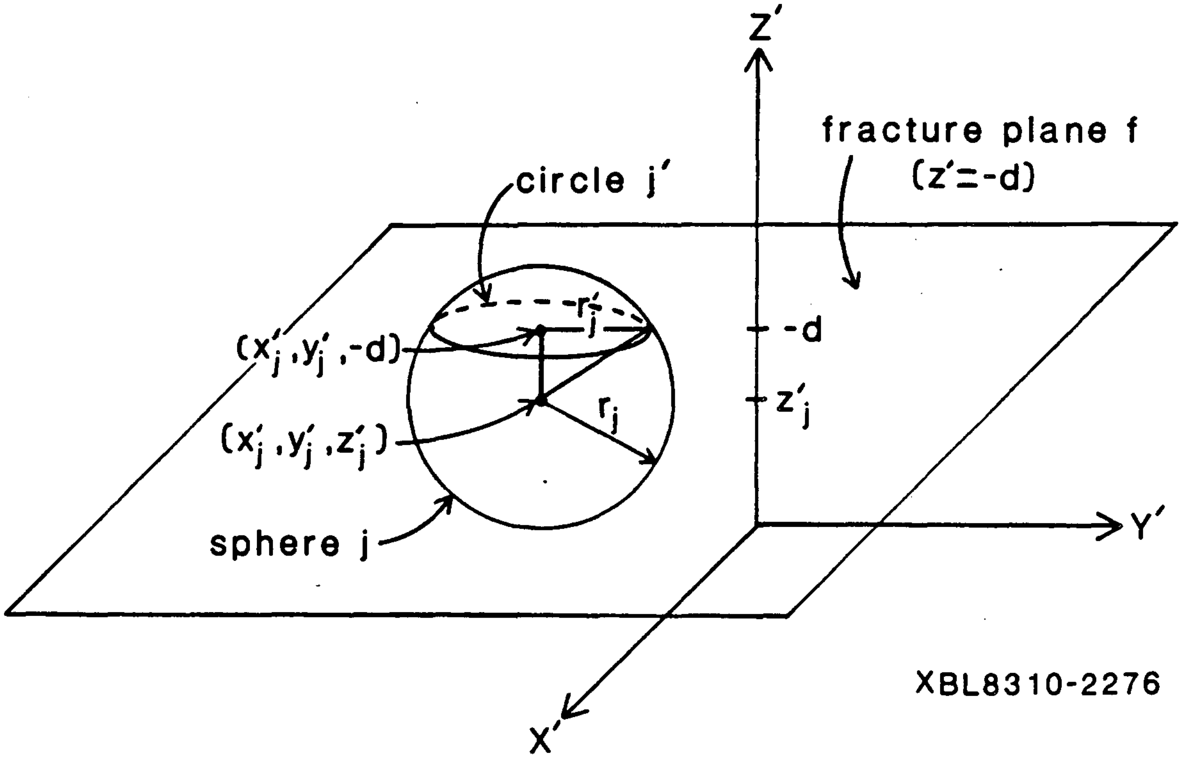
If sphere j intersects plane f and planes j and f are not parallel, then fracture j may intersect plane f. The intersection of planes j and f is the line

$$a_j' x' + b_j' y' - c_j' d + d_j = 0 \ . \qquad \text{(G-4)}$$

The circular fracture j intersects plane f if and only if that line intersects the circle j' formed by plane f and sphere j,

$$(x' - x_j')^2 + (y' - y_j')^2 = r_j^2 - (z_j' + d)^2 = r_j'^{\,2} \ . \qquad \text{(G-5)}$$

Figure G-2 diagrams such an intersection and Figure G-3 shows the effective two-dimensional problem in the z' = -d plane. The equations of the line and the circle are solved simultaneously by subroutine SIMULC to yield zero, one, or two real roots:



XBL8310-2276

Figure G-1. Circular intersection, $j'$, of sphere j, $(x' - x_j') + (y' - y_j')^2 + (z' - z_j')^2 = r_j^2$ and fracture plane f, $z' = -d$

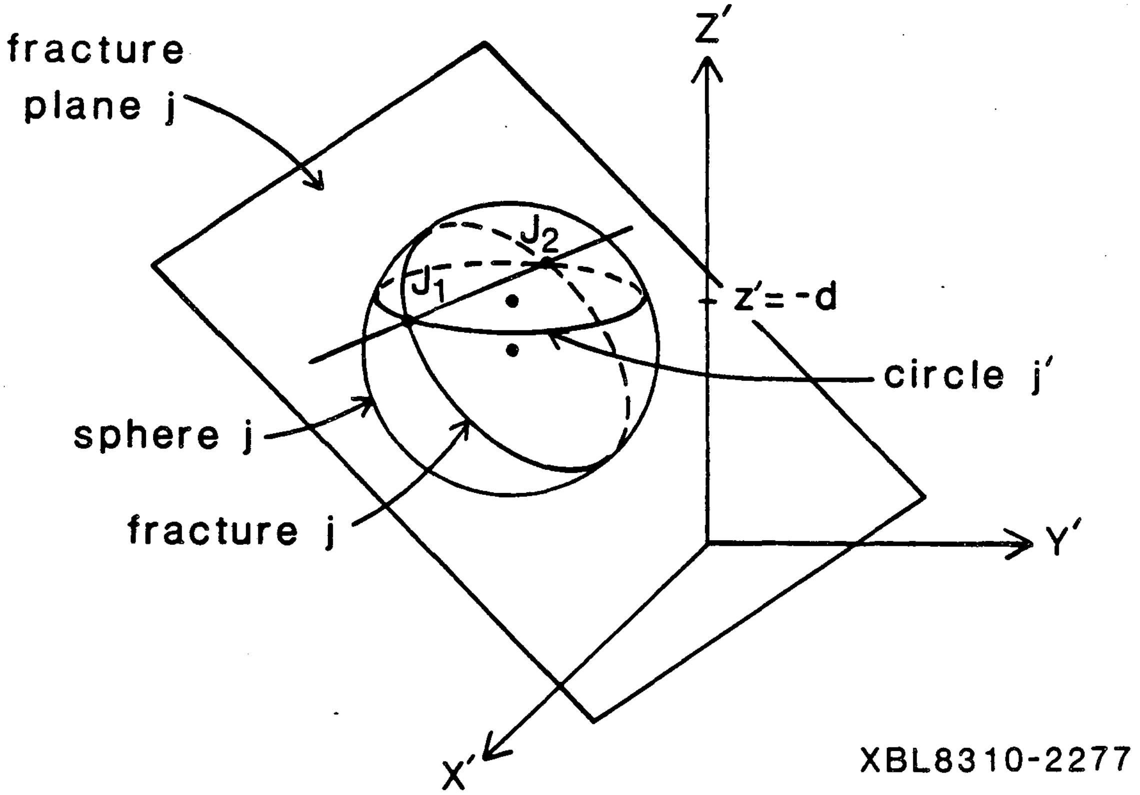


Figure G-2. Intersection of circular fracture j and circle j′ in the plane z′ = -d. The intersection is the line segment $J_1$ $J_2$

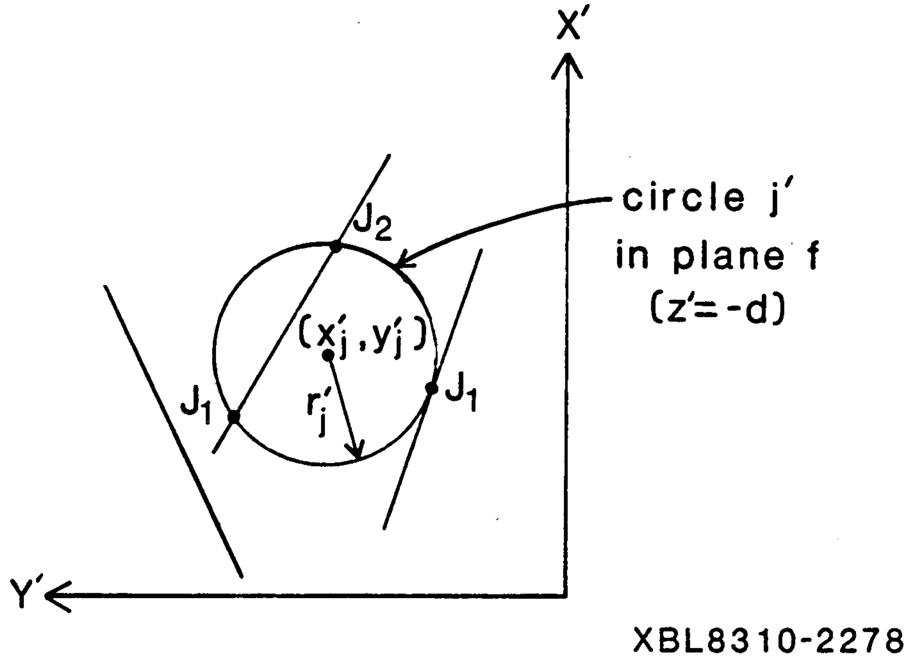


Figure G-3. Examples of possible relationships between the line of intersection of planes f and j and the circle $j'$ formed by sphere j and plane f.
The diagram is in the plane of fracture f, $z' = -d$

no root -- no intersection;
one root -- one point of intersection, $J_1$;
two roots -- two points of intersection, $J_2$ and $J_2$.

If the circle is a point circle, the point is $(x_j', y_j')$. This is a point of intersection, $J_1$, only if it satisfies the equation of the line. If fracture j intersects plane f at point $J_1$, or along the line segment $J_1J_2$, then it must be determined whether fractures f and j intersect or not.

In plane f, z' $=$ d, fracture f is a circle with center (k,0) and radius $r_f$ and fracture j is either a point, $J_1$, or a line segment connecting points $J_1$ and $J_2$. Fractures f and j intersect if, in the first case, the point $J_1$, or, in the second case, any part of the line segment $J_1J_2$ lies within the circle f. Figure G-4 is an example of such an intersection.

If fracture j is only the point $J_1$, then it intersects circle f if the distance between $J_1$ and (k,0) is less than or equal to the radius, $r_f$.

If fracture j is the line segment $J_1J_2$ on the line $a_j' x' + b_j' y' - c_j' d + d_j = 0$, then subroutine SIMULC is called to determine the simultaneous solution of that line and fracture circle f, $(x' - k)^2 + y'^2 = r_j^2$. If there are no roots, there is no intersection between fractures f and j. If there is one root, $I_1$, or two roots, $I_1$ and $I_2$, there may be an intersection, depending on the locations of $J_1$ and $J_2$. Figure G-5 diagrams the possibilities and shows that the I's and J's may coincide and there is no predetermined order of the points along the line. Subroutine COMPT is called in the first case to determine whether the point $I_1$ is between points $J_1$ and $J_2$. If it is, there is intersection at only one point, $I_1$. In the second case, subroutine COMSEG is called to determine the common segment, if any, between segments $I_1I_2$ and $J_1J_2$. The common segment (the length may be zero, i.e. one point) is the intersection of fractures f and j.

*Truncation of intersections at flow region boundary lines.* All intersections with fracture f lie entirely within that fracture, but if fracture f was truncated at one or more flow region boundary lines, then intersections with that fracture may extend beyond the

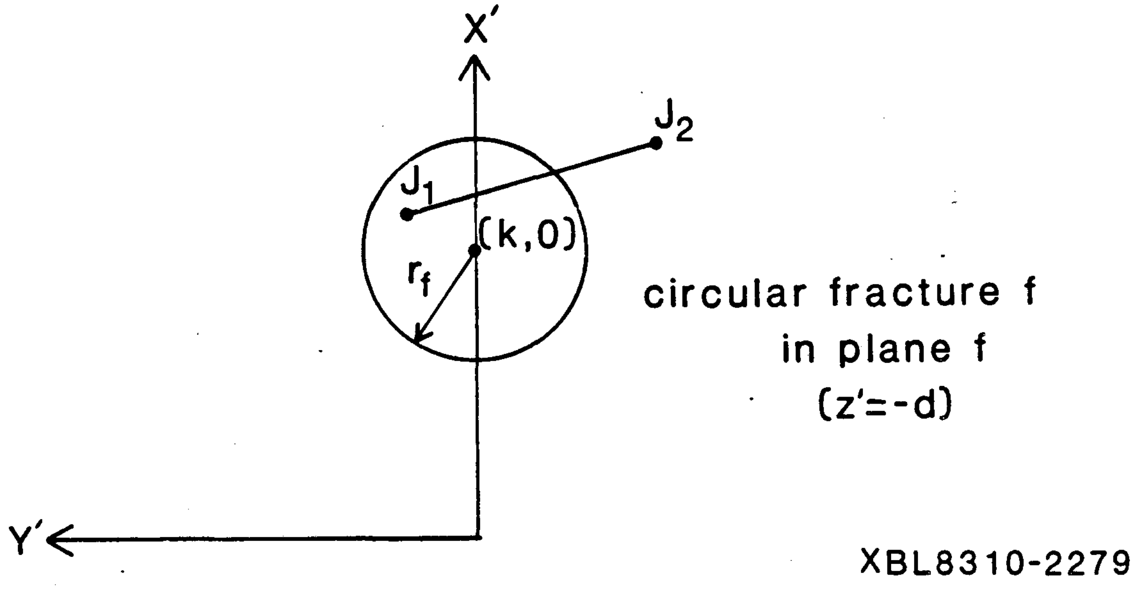
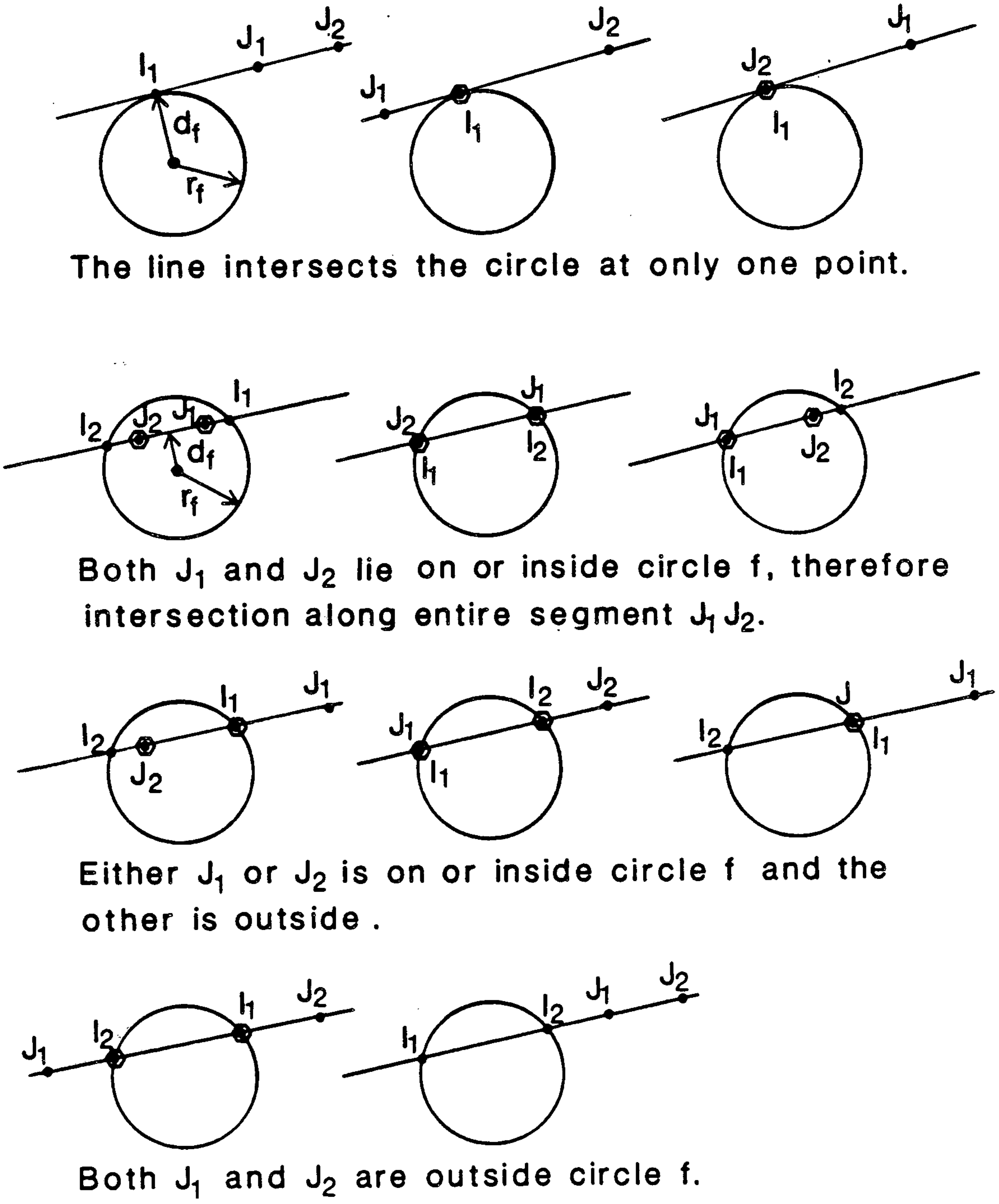


Figure G-4. Example of intersection of circular fracture f with line segment $j_1$ $J_2$ which is fracture j in the $z'$ = -d plane



The line intersects the circle at only one point.

Both $J_1$ and $J_2$ lie on or inside circle f, therefore intersection along entire segment $J_1 J_2$.

Either $J_1$ or $J_2$ is on or inside circle f and the other is outside .

Both $J_1$ and $J_2$ are outside circle f.

XBL8310-2280

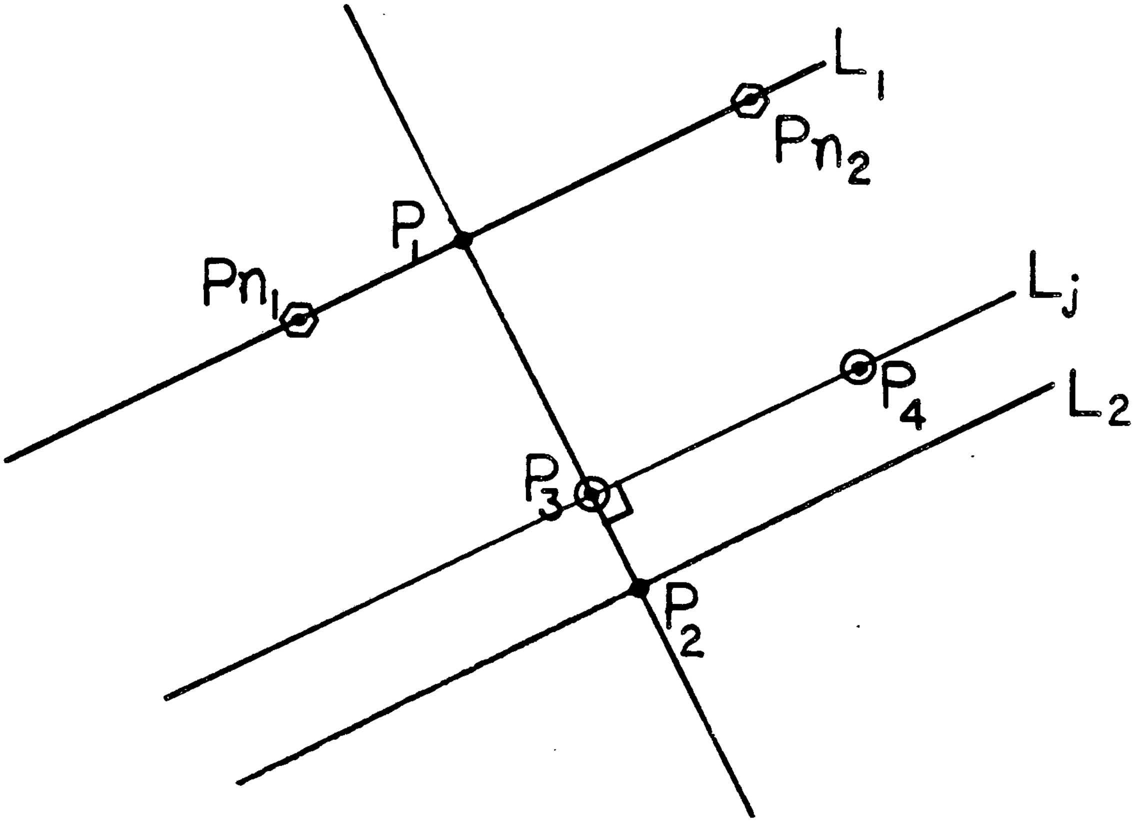
Figure G-5. Possible relationships between circle f and points $J_1$ and $J_2$ when the line through $J_1$ and$J_2$ intersects the circle at points $I_1$ and$I_2$. The points of intersection of fractures f and j, marked , will define the node, i.e., the line segment of fracture intersection.

flow region. Subroutine TRUNCN is called for each intersection found on a fracture containing boundary nodes to determine what portion of the intersection, if any, is inside the flow region. The following procedure is applied to each boundary node on fracture f.

Given: endpoints of the fracture intersection, $P_3$ and $P_4$, on line $L_j$, (Equation G-4); a boundary node on fracture f with endpoints $Pn_1$ and $Pn_2$; and the boundary line containing the node, $L_1$, and its parallel, $L_2$, (Equation 3-21).

1. If $L_j$ is parallel to $L_1$ and $L_2$, determine whether it is between them or not by calling subroutine SIMULL to find the points, $P_1$ and $P_2$, where a line perpendicular to $L_j$ and passing through intersection endpoint $P_3$ intersects lines $L_1$ and $L_2$, respectively. This is diagrammed in Figure G-6. Then call subroutine COMPT to determine the position of $P_3$ relative to $P_1$ and $P_2$. There are five possibilities: in between, coincident with either $P_1$ or $P_2$, or outside on the side of either $P_1$ or $P_2$. If $P_3$ is in between, i.e. line $L_j$ is between boundary lines $L_1$ and $L_2$ , the fracture intersection is inside the flow region relative to the boundary node in question. If $P_3$ is coincident with either $P_1$ or $P_2$, or is not between them, then the fracture intersection is discarded because it is outside the flow region.

2. If line $L_j$ intersects lines $L_1$ and $L_2$ , subroutine SIMULL is called twice to find the points of intersection, $P_1$ and $P_2$, respectively. Subroutine COMPT is called twice to determine positions of points $P_3$ and $P_4$ relative to $P_1$ and $P_2$. Some examples are shown in Figure G-7. If both $P_3$ and $P_4$ are outside but on the same side (Example 1) then the intersection is outside the flow region and is discarded. If either $P_3$ or $P_4$ is between $P_1$ and $P_2$ and the other is in between, coincident with $P_2$, or outside on the side of $P_2$ (Example 2) then the intersection remains as is for comparison with the next boundary node. If either $P_3$ or $P_4$ is outside on the side of $P_2$, then there is a boundary node on that line which will be examined in turn. If either $P_3$ or $P_4$ is outside or coincident with $P_1$ while the other is in between $P_1$ and $P_2$, coincident with $P_2$, or outside on the side of $P_2$, then that fracture intersection



XBL8310-2281

Figure G-6. Determining whether intersection segment $P_3$ $P_4$ is inside the flow region when line $L_j$ is parallel to boundary lines $L_1$ and $L_2$

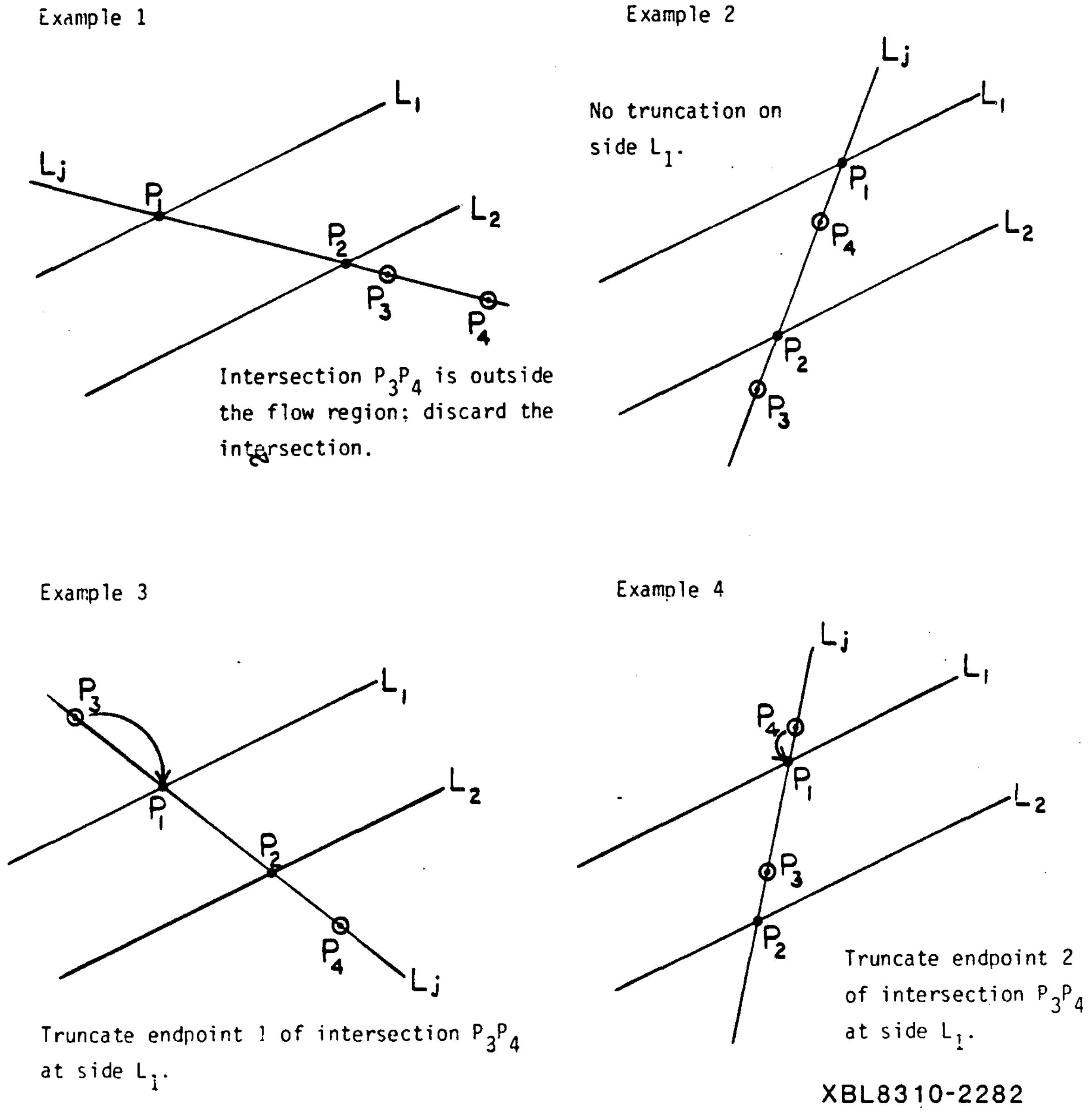

Example 1
$L_1$
$L_j$
$P_1$
$L_2$
$P_2$
$P_3$
$P_4$
Intersection $P_3P_4$ is outside the flow region; discard the intersection.
Example 2
$L_j$
No truncation on side $L_1$.
$L_1$
$P_1$
$P_4$
$L_2$
$P_2$
$P_3$
Example 3
$P_3$
$L_1$
$P_1$
$L_2$
$P_2$
$P_4$
$L_j$
Truncate endpoint 1 of intersection $P_3P_4$ at side $L_1$.
Example 4
$L_j$
$L_1$
$P_4$
$P_1$
$L_2$
$P_3$
$P_2$
Truncate endpoint 2 of intersection $P_3P_4$ at side $L_1$.
XBL8310-2282

Figure G-7. Determining whether intersection segment $P_3$ $P_4$, or any portion of it, is inside side $L_1$ when line $L_j$ intersects boundary lines $L_1$ and $L_2$

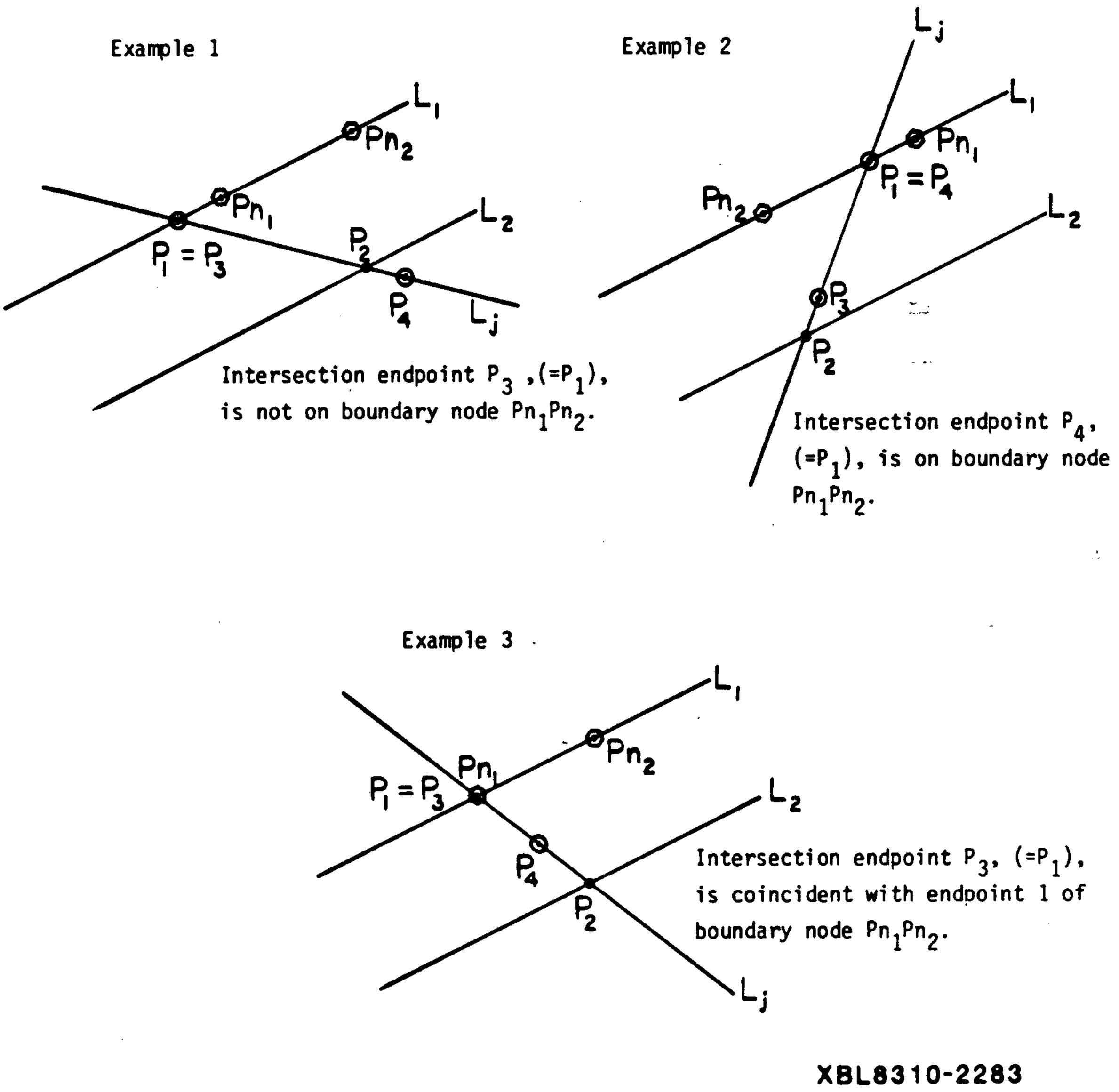
endpoint is moved to $P_1$ (Examples 3 and 4). Thus, $P_3$ and $P_4$ are both "inside" line $L_1$ but not necessarily "inside" line $L_2$.

3. If either intersection endpoint, $P_3$ or $P_4$, was coincident with or moved to $P_1$, it must be determined whether $P_1$ is in between the endpoints of the boundary node. Subroutine COMPT is called to determine the position of $P_1$ relative to the nodal endpoints $Pn_1$ and $Pn_2$. If $P_1$ is outside, Figure G-8, Example 1, then the intersection does not intersect the boundary node. If $P_1$ is in between or coincident with either $Pn_1$ or $Pn_2$, Examples 2 and 3, then the intersection is truncated at the boundary node and the side code is set for either $P_3$ (intersection endpoint 1), or $P_4$ (endpoint 2), depending on which end was truncated.

Figure G-9 illustrates a series of truncations in which the fracture intersection is discarded and Figure G-10 shows a series in which the intersection is truncated at both ends.

The segments of fracture intersections which are discarded because they lay outside the flow region are needed for the plotting program DIMES. Thus, they are recorded by subroutine TRUNCN in arrays [endout (6,maxout)] and [ifout (maxout,2)]. After all intersections with fracture f have been determined and truncated, the coordinates of the endpoints of intersections and boundary nodes on fracture f are transformed from the local X'Y'Z' coordinate system back to the global XYZ system by subroutine RFRCGLO:

$$(x' \ \ y' \ \ z') \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ a & b & c \end{pmatrix} = (x \ y \ z) \ . \qquad \text{(G-6)}$$

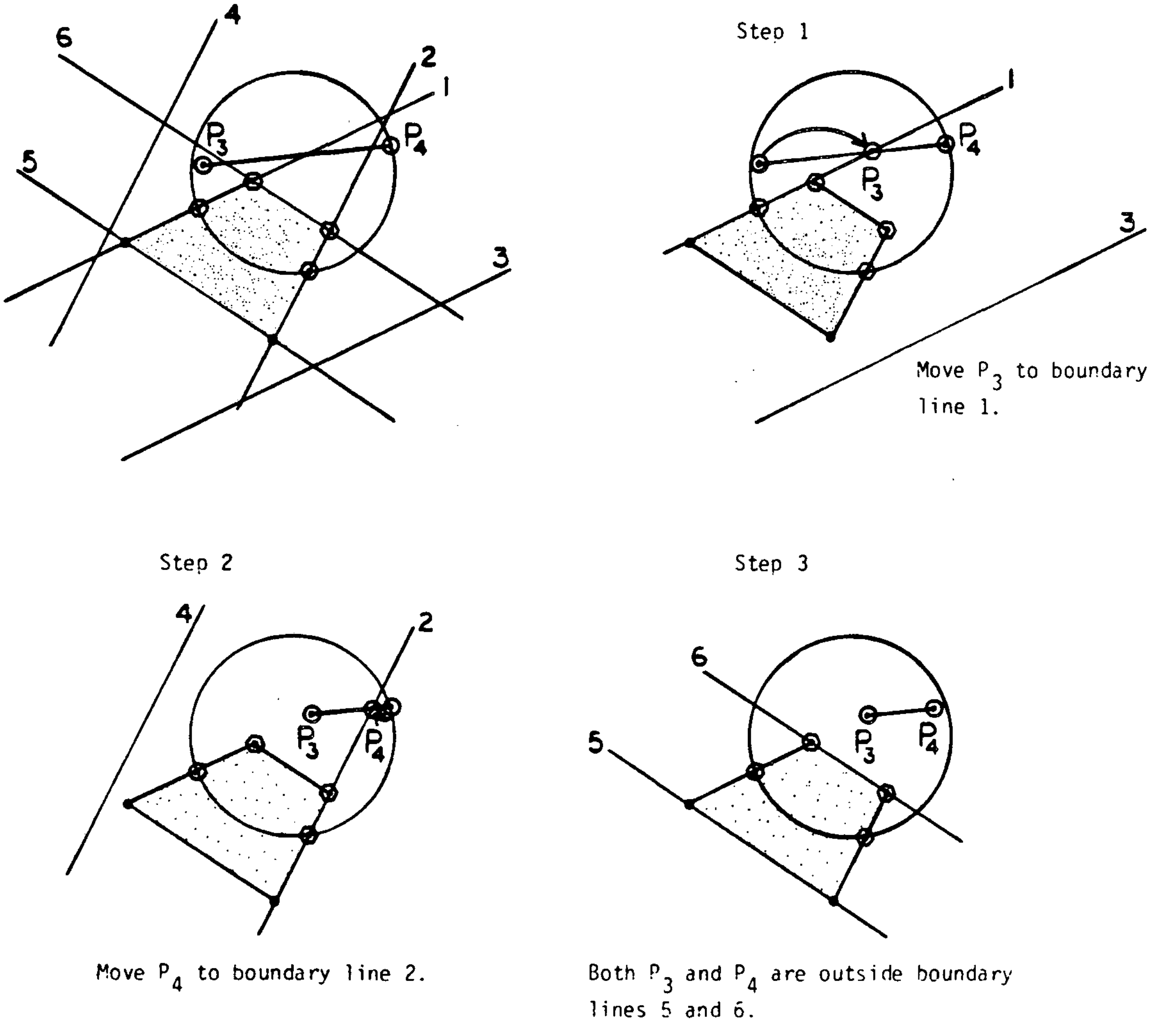


Intersection endpoint $P_3$ ,(=$P_1$), is not on boundary node $Pn_1Pn_2$.

Intersection endpoint $P_4$, (=$P_1$), is on boundary node $Pn_1Pn_2$.

Intersection endpoint $P_3$, (=$P_1$), is coincident with endpoint 1 of boundary node $Pn_1Pn_2$.

Figure G-8. Determining whether truncated endpoint of intersection segment $P_3$ $P_4$ is on boundary node $Pn_1$ $Pn_2$



Step 1

Move $P_3$ to boundary line 1.

Step 2

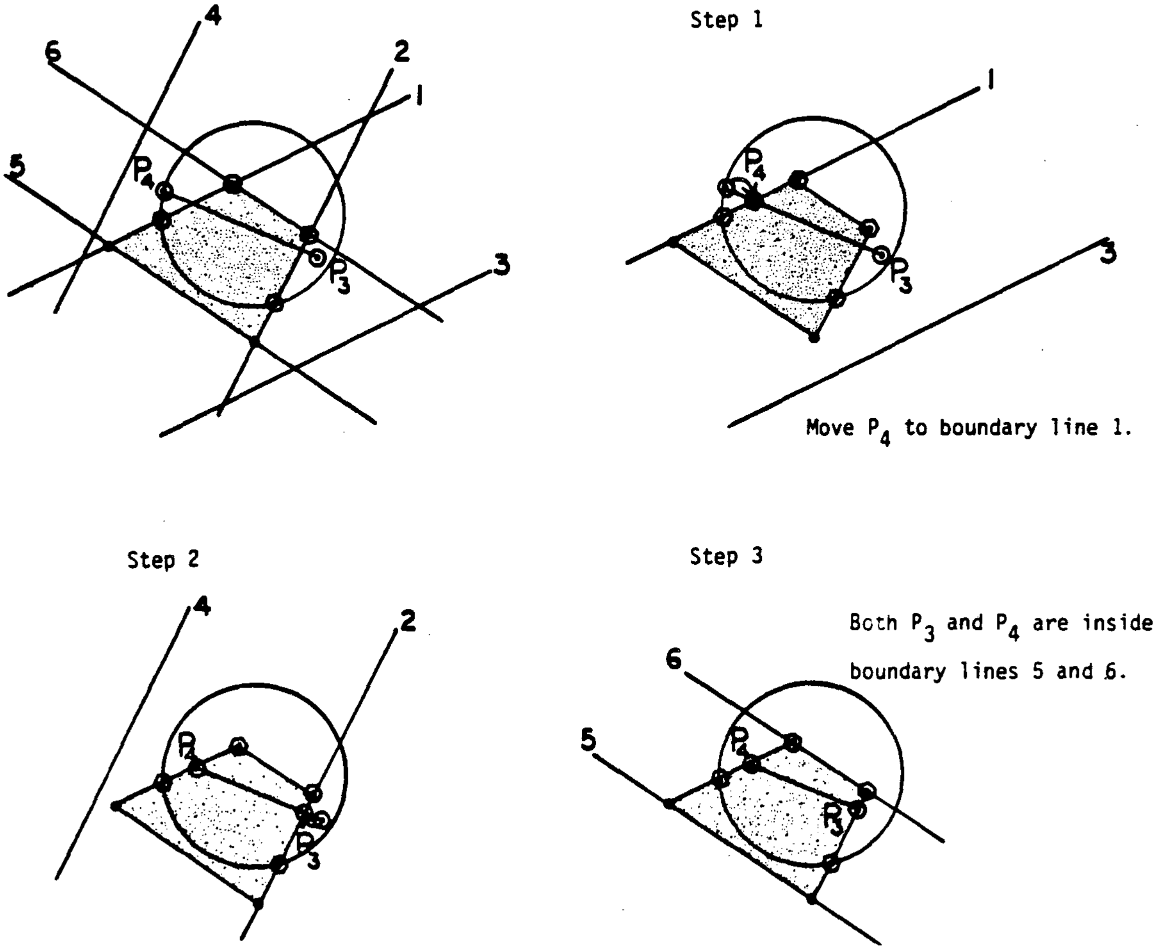
Move $P_4$ to boundary line 2.

Step 3

Both $P_3$ and $P_4$ are outside boundary lines 5 and 6.

XBL8310-2284

Figure G-9. Intersection of $P_3$ $P_4$ is truncated twice and finally discarded for being outside the flow region



Step 1

Move $P_4$ to boundary line 1.

Step 2

Move $P_3$ to boundary line 2.

Step 3

Both $P_3$ and $P_4$ are inside boundary lines 5 and 6.

XBL8310-2285

Figure G-10. Intersection $P_3$ $P_4$ is truncated at both ends

## REFERENCES

Baecher, G. B. and N. A. Lanney (1978). Trace length biases in joint surveys. *Proceedings of the 19th U.S. Symposium on Rock Mechanics,* American Institute of Mining Engineers, p. 56-65.

Barton, C. C. (1984). Tectonic significance of fractures in welded tuff, Yucca Mountain, Southwest Nevada, Abstracts with Programs, Vol. 16, Issue No. 6, GSA Annual Meeting, Nevada Test Site Symposium, Reno, Nevada, November.

Conrad, F. and C. Jacquin (1973). Representation of a two-dimensional fracture network by a probabilistic model: Application to calculation of the geometric magnitude of matrix blocks. University of California, Lawrence Livermore Laboratory Publication UCRL-Trans-10814 (1975), 75 pp.

Crochet, M. J., A. R. Davies and K. Walters (1984). *Numerical Simulation of Non-Newtonian Flows.* Elsevier, New York, 352 pp.

Dershowitz, W. S. (1984). Rock Joint Systems, Ph.D. Thesis, Dept. of Civil Engineering, Massachusetts Institute of Technology.

Gilmour H. M. P., D. Billaux and J.C.S. Long (1986) *Models for calculating fluid flow in randomly generated three-dimensional networks of disc-shaped fractures.* User Manuals and listings for FMG3D, DISCEL and DIMES. Lawrence Berkeley Laboratory report No. 19516.

Hammersly, J. M. and D. C. Handscomb (1964). *Monte Carlo Methods.* Methuen and Co., London, 178 pp.

Hudson, J. A., and S. D. Priest, (1979). Discontinuities and rock mass geometry, *Int. J. Rock Mech. Min. Sci. Geomech. Abstr.,* Vol. 16.

LaPointe, P. R., and J. A. Hudson (1981). Characterization and Interpretation of Rock Mass Jointing Patterns, Department of Metallurgical and Mineral Engineering, University of Wisconsin, Madison.

Long, J. C. S. (1983). Investigation of Equivalent Porous Medium Permeability in Networks of Discontinuous Fractures, Ph.D. Thesis, College of Engineering, University of California, Berkeley.

Long, J. C. S., J. S. Remer, C. R. Wilson, P. A. Witherspoon (1982). Porous media equivalents for networks of discontinuous fractures. *Water Resources Research,* 18(3), p. 645-658.

Long, J. C. S., P. A. Witherspoon, (1985). The relationship of the degree of interconnection and permeability in a fracture network, *Journal of Geophysical Research,* Vol. 90, No. B4, pp. 3087-3098, March 10, 1985.

Long, J. C. S., H. K. Endo, K. Karasaki, L. Pyrak, P. Maclean and P. A. Witherspoon (1985). Hydrologic behavior of fracture networks, Proc. International Association of Hydrogeologists 17th International Congress, Tucson, Arizona, January 7-10.

Mahtab, M. A., and F. M. Yegulalp (1982) A rejection criterion for definition of clusters in orientation data. *Proceedings, 23rd Symposium on Rock Mechanics,* University of California, Berkeley.

Mahtab, M. A., D. B. Bolstad, J. R. Alldredge, and R. J. Shanley (1972). *Analysis of Fractured Orientations for Input to Structural Models of Discontinuous Rock.* U.S. Department of the Interior, Bureau of Mines Report, Investigation No. 7669, 76 p.

Milne-Thomson, R. M. (1968). *Theoretical Hydrodynamics,* MacMillan Co., New York. N.Y.

Pincus, H. J. (1953). The analysis of aggregates of orientation data in the earth sciences, *Journal of Geology, 61,* pp. 482-509.

Pollard, D. D. (1976). On the form and stability of open hydraulic fractures in the earth's crust, *Geophysical Research Letters, 3,* (9), p. 513-516.

Robertson, A., The interpretation of geological factors for use in slope stability, in *Proceedings, Symposium on the Theoretical Backround to the Planning of Open Pit Mines with Special Reference to Slope Stability* pp. 55-71, South African Institute of Mining and Metallurgy, Johannesburg, 1970.

Selby, S. M., ed. (1965). *Standard Mathematical Tables,* 14th Edition. The Chemical Rubber Co., Cleveland, Ohio.

Snow, D. T. (1969). Anisotropic permeability of fractured media, *Water Resources Research, 5,* (6), p. 1273-1289.

Veneziano; D., (1979). Probabilistic model of joints in rock, Report 47 pp., Civil Engineering Department, Massachusetts Institute of Technology, Boston.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

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