

*PINDER*

**SCP  
5-1765**

**3384**

NNA-870806-0062

SAIC/T&MSS

AUG 06 1987

CCF RECEIVED

HYDROLOGY DOCUMENT NUMBER 382



# Water Resources Program

DEPARTMENT OF CIVIL ENGINEERING  
DEPARTMENT OF GEOLOGICAL &  
GEOPHYSICAL SCIENCES

**GALERKIN-FINITE ELEMENT MODELS  
FOR AQUIFER SIMULATION**

**Prepared by G. F. Pinder**

**January 1974**

**Revised August 1974  
Revised August 1976**

**76-WR-5**

PART 1

SIMULATION OF GROUNDWATER FLOW

Galerkin-Finite Element Models for Aquifer Simulation,  
76-WR-5, 1976

## 1. INTRODUCTION

### 1.1 Purpose of Manual

This manual is prepared with a dual purpose. The first section will introduce you to the finite element-Galerkin method for solving partial differential equations and the second part consists of a documentation that will permit you to use the accompanying computer program in solving problems of areal groundwater flow. The manual is designed for the practicing hydrologist rather than the theoretician. Accordingly we will omit detailed discussions of equation development and error analysis, and refer the interested reader to recent publications in the numerical methods, structural engineering and water resources literature.

### 1.2 What is the Finite Element-Galerkin Method?

As the name would imply the finite element-Galerkin method evolved from the marriage of the finite element method of integration and the Galerkin method of generating approximate integral equations. The finite element method was introduced about two decades ago in structural engineering, more specifically in the aerospace industry. At first it was fundamentally intuitive, but gradually came to be associated with the calculus of variations and the Ritz method of generating approximate integral equations. There were, however, limitations to the Ritz scheme which precluded its use in important areas of hydrology such as contaminant transport.

Meanwhile, in the oil industry, a group of numerical analysts were experimenting with the use of the Galerkin method of generating approximate integral equations. In contrast to the structural engineers, they were not concerned with the accurate representation of domain geometry but focused their interest primarily on solution accuracy and numerical efficiency. As a result, the integration schemes they utilized were restricted to rectangular sub-domains very similar to those encountered in standard finite-difference

methods. It is important to point out, however, that the Galerkin method is totally general and, in contrast to the Ritz procedure, could be used to approximate all classes of partial differential equations.

Only recently have the finite-element and Galerkin methods been combined. This combination provides the mathematical flexibility of the Galerkin scheme with the inherent ability of the finite-element method to accurately represent irregular geometry.

## 2. THEORETICAL DEVELOPMENT

### 2.1 Galerkin's Method

Our objective in this section is to introduce the method we use to generate the approximate integral equations that we later solve using the finite-element method. This scheme, commonly known as Galerkin's method, assumes there exists an infinite series which will exactly represent the solution we seek to our partial differential equation. Because, in the case of groundwater flow, this solution will be the areal hydraulic head distribution at specified times, the approximating series is of the form

$$h = \hat{h} = \sum_{i=1}^n H_i(t) w_i(x,y) \quad (2-1)$$

where

$h$  is hydraulic head [L],

$\hat{h}$  is the series approximation to  $h$  [L],

$H_i$  is an undetermined coefficient [L], and

$w_i$  is a basis function.

The series approximation (2-1) will provide an exact representation as  $n$  approaches infinity ( $\hat{h}$  will approach  $h$ ). By a careful selection of the basis functions  $w_i$ , the undetermined coefficients  $H_i$  become the head values at selected points (or nodes). The choice of basis functions to fulfill this condition should be considered the key step in making the Galerkin formulation

a computer oriented solution scheme. Basis functions will be considered in more detail later in our theoretical development.

The equation we will consider here describes transient, two dimensional groundwater flow, including the effects of vertical transient leakage from a confining layer. To minimize algebra during the ensuing development we will define this equation as the operator L,

$$L(h) \equiv \frac{\partial}{\partial x} \left( T_{xx} \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( T_{yy} \frac{\partial h}{\partial y} \right) - s \frac{\partial h}{\partial t} - Q - \frac{K}{l'} (h-h_w) - c(h-h_1) \quad (2-2)$$

where

$c$  is the transient leakage coefficient [ $T^{-1}$ ]

$T$  is the transmissivity [ $L^2 T^{-1}$ ],

$s$  is the storage coefficient [ $L^0$ ],

$Q$  is the strength of a sink function [ $LT^{-1}$ ],

$K$  is the vertical hydraulic conductivity of a confining layer [ $LT^{-1}$ ]

$l'$  is the thickness of the confining layer [ $L$ ],

$h_w$  is the hydraulic head in the adjacent aquifer [ $L$ ], and

$h_1$  is the initial hydraulic head in the aquifer [ $L$ ]

The sink function  $Q$  is used to introduce well discharge. To allow us to easily incorporate this term later in the analysis we introduce the Dirac delta function  $\delta$  (note this is not the Kroneker delta commonly encountered in applied mathematics) and define  $Q$  as

$$Q = \sum_{j=1}^n Q_w(x_j, y_j) \delta(x-x_j) \delta(y-y_j) \quad (2-3)$$

The Dirac delta function has the property that, when integrated, the function  $Q$  will equal  $Q_w$  at  $(x_j, y_j)$ . In other words it enables us to treat a point source or sink such as a well.

The transient leakage coefficient  $c$  was introduced by Bredehoeft

and Pinder, 1970<sup>1</sup> to account, albeit approximately, for transient leakage from a confining bed. This coefficient is developed from a one-dimensional transient flow solution and is given by

$$C = K/[L'(\pi w)^{1/2}] \cdot [1 + 2 \sum_{n=1}^{\infty} \exp(-n^2/w)]$$

where

$$w = 2K t / (2L'^2 S_g),$$

and  $S_g$  is the specific storage of the confining layer.

Since the approximating function  $\hat{h}$  becomes the exact solution  $h$  only as  $n$  approaches infinity, for a finite series equation 2-2 will not be exactly satisfied and there will be a residual  $R$ . Using the notation of (2-2) this is expressed as

$$L(\hat{h}) = R \tag{2-4}$$

To solve (2-2) using Galerkin's method we attempt, in a sense, to minimize the residual  $R$ . We accomplish this by first considering a complete set of functions  $w_i$  (note that in the Galerkin method the basis functions turn out to be the same set). Now imagine generating a function that would be orthogonal to each of the  $n$  functions  $w_i$ . The only function which fulfills this condition is zero. If we now force the residual  $R$  to be orthogonal to all possible values of  $w_i$  we are, in fact, forcing  $R$  to zero and thereby obtaining a solution to 2-2. Expressing this another way

$$L(\hat{h}) = R = 0 = L(h) \tag{2-5}$$

Unfortunately the condition expressed by (2-5) can be achieved only as  $n$  approaches infinity and computers can deal only with finite sets of numbers. We are forced, therefore, to consider a finite subset of values  $w_i$ ,  $i = 1, 2, \dots, N$  which generally makes our solution approximate rather than exact.

<sup>1</sup>Bredehoeft, J. D. and G. F. Pinder, Digital analysis of areal flow in multiquifer groundwater systems: a quasi three-dimensional model, Water Resource. Res. 6(3), p. 883-888.

Thus we require  $N$  conditions of orthogonality of the function  $R$  and the basis functions  $w_i$ ,  $i = 1, 2, \dots, N$ . Recalling the definition of orthogonal functions these  $N$  conditions can be expressed as

$$\iint L(\hat{h}) w_i \, dx dy = 0 \quad i = 1, 2, \dots, N \quad (2-6)$$

Introducing the definition of (2-2) we obtain

$$\iint \left[ \frac{\partial}{\partial x} \left( T_{xx} \frac{\partial \hat{h}}{\partial x} \right) + \frac{\partial}{\partial y} \left( T_{yy} \frac{\partial \hat{h}}{\partial y} \right) - S \frac{\partial \hat{h}}{\partial t} - Q - \frac{K}{L} (\hat{h} - h_w) - C(\hat{h} - h_1) \right] w_i \, dx dy = 0 \quad i = 1, 2, \dots, N \quad (2-7)$$

Equation (2-7) can be integrated by parts to obtain

$$\begin{aligned} & \iint \left( T_{xx} \frac{\partial \hat{h}}{\partial x} \frac{\partial w_i}{\partial x} + T_{yy} \frac{\partial \hat{h}}{\partial y} \frac{\partial w_i}{\partial y} \right) dx dy - \int \left( T_{xx} \frac{\partial \hat{h}}{\partial x} l_x + T_{yy} \frac{\partial \hat{h}}{\partial y} l_y \right) w_i \, dS \\ & + \iint \left[ S \frac{\partial \hat{h}}{\partial t} + Q + \frac{K}{L} (\hat{h} - h_w) + C(\hat{h} - h_1) \right] w_i \, dx dy = 0 \quad i = 1, 2, \dots, N \end{aligned} \quad (2-8)$$

where  $l_x$  and  $l_y$  are the direction cosines between the normal to the surface  $S$  and the  $x$  and  $y$  co-ordinate axes respectively. It is interesting to note that derivatives in transmissivity that implicitly appear in (2-7) are not in (2-8). Substituting (2-1) into (2-8) gives us the final form of the approximating equations.

$$\begin{aligned} & \iint \left[ T_{xx} \left( \sum_{j=1}^N H_j \frac{\partial w_j}{\partial x} \right) \frac{\partial w_i}{\partial x} + T_{yy} \left( \sum_{j=1}^N H_j \frac{\partial w_j}{\partial y} \right) \frac{\partial w_i}{\partial y} + S \frac{\partial}{\partial t} \sum_{j=1}^N H_j w_j w_i \right. \\ & \left. + Q w_i + \left( \frac{K}{L} + C \right) \sum_{j=1}^N H_j w_j w_i - \left( \frac{K}{L} h_w + C h_1 \right) w_i \right] dx dy \\ & - \int \left[ T_{xx} \frac{\partial \hat{h}}{\partial x} l_x + T_{yy} \frac{\partial \hat{h}}{\partial y} l_y \right] w_i \, dS = 0 \quad i = 1, 2, \dots, N \end{aligned} \quad (2-9)$$

A formal substitution for the last term in (2-9) was not made because the quantity in brackets is, in fact, the flux across the boundary of the region

5. This term thus represents a flux boundary condition and is referred to in the calculus of variations as a *natural boundary condition*. The other type of boundary condition encountered in groundwater flow is the constant head condition which is designated as an *essential boundary condition*. The task remains to determine the form of the basis functions  $w_j$ .

## 2.2 Basis Functions

Examination of equation 2-9 reveals that, in order to generate a set of algebraic equations to solve for the  $n$  undetermined coefficients  $H_j$ , it is necessary to perform  $N^2$  integrations of the form

$$\iint \frac{\partial w_i}{\partial x} \frac{\partial w_j}{\partial x} dx dy \text{ and } \iint w_i w_j dx dy$$

Fortunately the required number of integrations can be reduced substantially through a judicious choice of the functions  $w_j$ .

The original Galerkin formulation assumed that each basis function  $w_j$  was defined over the entire region of interest. The idea of defining these functions as piecewise continuous functions defined as non-zero only over a small subarea of the total region, was key in generating an efficient numerical scheme. There are two basic philosophies on how the basis functions and subregions or elements should be defined. In the petroleum industry emphasis has been placed on higher-order functions defined over simple elements, generally squares or rectangles. Structural engineering, on the other hand, has utilized relatively simple basis functions defined over irregular elements, principally triangles or rectangles which may, in some cases, have curved sides. The element that is used in this program is fundamentally a rectangle which may be deformed in a specific way. This element is referred to in the structures literature as a deformed isoparametric quadrilateral element. Because of its irregular geometry it is necessary to use numerical

Integration (more specifically gaussian quadrature) to evaluate the integrals appearing in (2-9). To facilitate this integration the irregular element defined in  $(x, y)$  space (global co-ordinates) is transformed into a local  $(\xi, \eta)$  co-ordinate system (local co-ordinates). A typical element in  $(x, y)$  space and its transformed equivalent in  $(\xi, \eta)$  space is shown in Figure 1.

A basis function is defined for each node such that it is non-zero *only* over the element on which the node is located. Since a node is nearly always located on more than one element, each node will have associated with it several sub-basis functions (commonly called shape functions in the structures literature), one for each of these elements. Each of the sub-basis functions is integrated only over the element on which it is defined as non-zero, thereby reducing the computational effort considerably.

A typical set of sub-basis functions is illustrated in Figure 2. There are two important characteristics common to each of these three functions: they are *unity* at the node for which they are defined and they are *zero* at every other node, including nodes in the element over which they are defined. Consider the impact these two characteristics of basis functions have on evaluating the series given by equation 2-1, which we introduced at the beginning of our discussion. Assume you wish to determine the approximate value of  $h$ , which we designated  $\hat{h}$ , at some node  $k$ . Because the basis functions are zero at all nodes  $i \neq k$  the series evaluated at  $k$  would reduce to

$$\hat{h}(\text{at } k) = H_k w_k \quad (2-10)$$

Moreover, because  $w_k$  is defined as unity at node  $k$  the desired head value is, in fact, the undetermined coefficient  $H_k$ . In practical problems, where one is generally satisfied with a solution at specified points, the numerical problem reduces to one of solving for  $H_k$ .

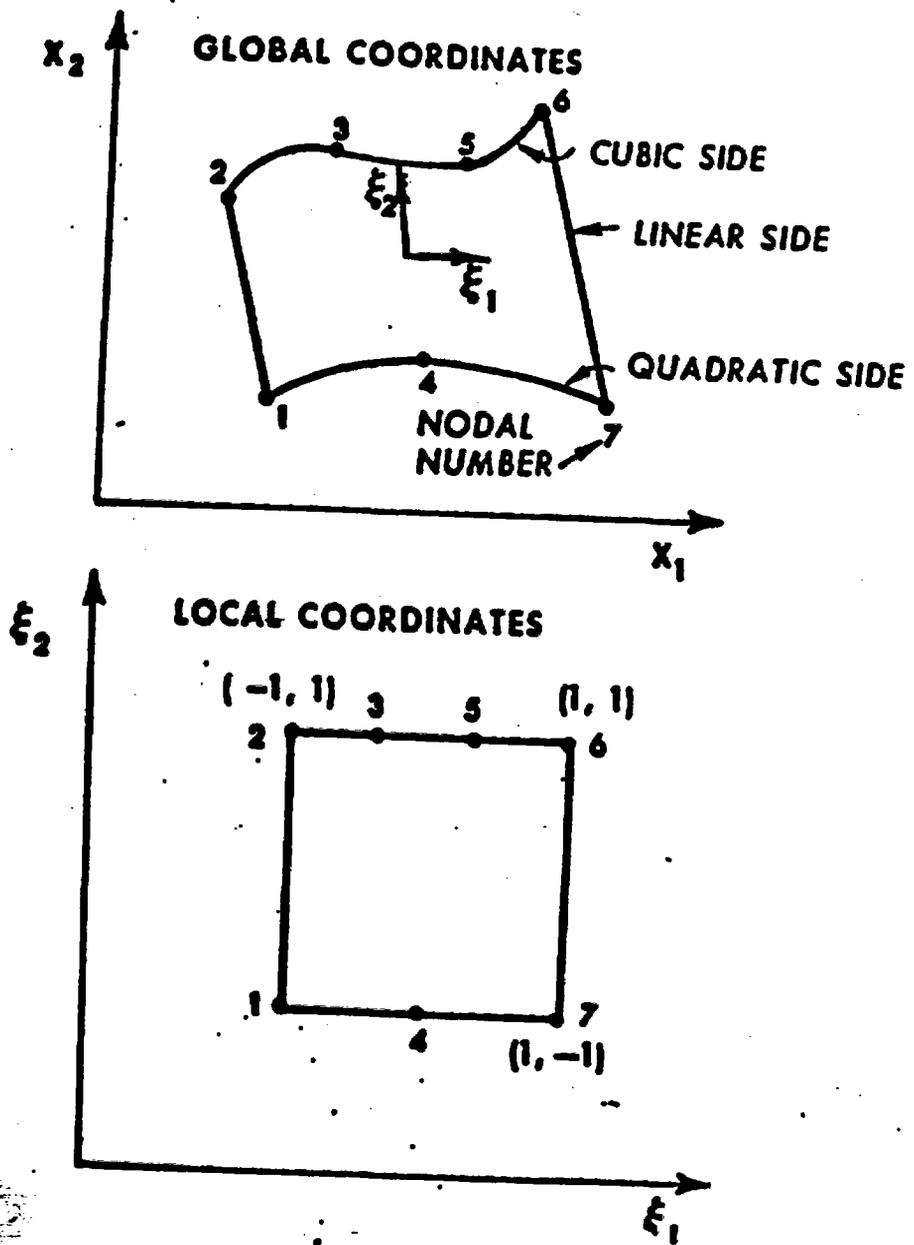


Fig. 1. Deformed isoparametric quadrilateral element in global  $(x,y)$  and local  $(\xi,\eta)$  co-ordinates.

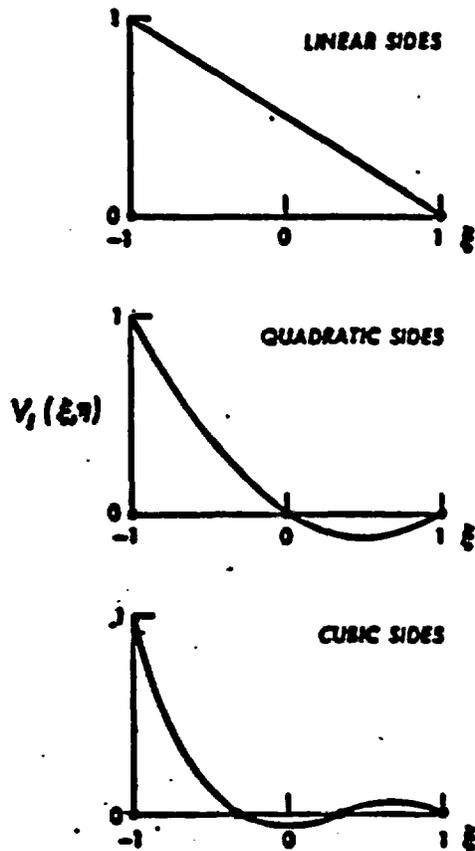


Fig. 2. Basis function along  $\xi = -1$  for corner node  $(-1, -1)$  with two adjacent sides of indicated order.

### 2.3 Matrix Equations

The  $N$  equations of 2-9 can be written in matrix form as

$$[A] \{H\} + [B] \left\{ \frac{dH}{dt} \right\} + \{F\} = 0 \quad (2-11)$$

where  $[A]$  and  $[B]$  are  $N \times N$  matrices and  $\{H\}$ ,  $\left\{ \frac{dH}{dt} \right\}$ , and  $\{F\}$  are column vectors. Typical elements of  $[A]$ ,  $[B]$ , and  $\{F\}$  are

$$A_{ij} = \iint \left( T_{xx} \frac{\partial w_i}{\partial x} \frac{\partial w_j}{\partial x} + T_{yy} \frac{\partial w_i}{\partial y} \frac{\partial w_j}{\partial y} + \left( \frac{K}{L} + C \right) w_i w_j \right) dx dy \quad (2-12a)$$

$$B_{ij} = \iint S w_i w_j dx dy \quad (2-12b)$$

$$F_i = - \iint w_i \left( \frac{K}{L} h_w - Q + Ch_i \right) dx dy - \int_S w_i \sum_{j=1}^N \left( T_{xx} \frac{\partial w_j}{\partial x} l_x + T_{yy} \frac{\partial w_j}{\partial y} l_y \right) H_j dS \quad (2-12c)$$

where, as discussed earlier, the last term in  $F$  is considered as a known boundary condition.

In the sequence of steps leading up to equation 2-9 we have tacitly assumed that the time derivative would be treated using a finite-difference approximation. Although the time derivative may also be incorporated into the Galerkin scheme by simply making  $w_i$  a function of time, our experience indicates that, in general, it is advantageous to use a finite difference approach. There are several finite difference schemes that can be used, each having inherent advantages and disadvantages. Our experience indicates that an implicit backward difference scheme provides the most accurate solutions to groundwater-flow problems at minimum cost. In the backward difference representation a first order correct scheme is used to approximate the time derivative and the spatial derivatives are written

at the new time level. The appropriate matrix equation is

$$[A] (H)_{t+\Delta t} + [B] ((H)_{t+\Delta t} - (H)_t) / \Delta t + \{F\} = 0 \quad (2-13)$$

or rearranging for ease of computation

$$([A] + [B]/\Delta t) (H)_{t+\Delta t} = [B]/\Delta t (H)_t - \{F\} \quad (2-14)$$

where the unknown values appear on the left hand side of the equation and the known values on the right. It is important to note that the matrices  $[A]$  and  $[B]$  need be generated only once unless the geometry of the element configuration is changed. A second important observation is that the matrix  $[A] + [B]/\Delta t$  is *symmetric*. This is important not only because only half of the matrix need be stored, but also because of the decreased computational effort required to generate a solution to the  $N$  simultaneous equations. When the "two-step" upper triangularization-back substitution scheme<sup>1</sup> is used the major effort involved in triangularization is required only when the time step  $\Delta t$  is changed. It may prove advantageous to back substitute several times before again changing the time step, since back substitution requires relatively few arithmetic calculations.

Although it may not be obvious from equation 2-12 the coefficient matrices are not only sparse (contain a majority of zero elements) but are also banded. The bandwidth of non-zero elements in the coefficient matrices plays a significant role in the amount of computational effort required to solve (2-14) and it is important to minimize this parameter. The bandwidth is a function of the maximum difference between nodal numbers occurring on the same element. The difference occurring in the element of Figure 1, for example, is 6 (7-1) and, if this was the maximum difference encountered after examining all elements, the *half-bandwidth* would be 7 (the maximum difference plus one). You will soon learn from experience that the minimum

<sup>1</sup>Weaver, William Jr., Computer Programs in Structural Analysis, Van Nostrand Reinhold Co., New York, 1967.

bandwidth is generally obtained by numbering sequentially in the direction of the smallest dimension of the model. To illustrate this point reference is made to Figure 3 where a model consisting of five elements and 27 nodes is numbered first using the procedure described above (case A) and then along the length of the model (case B). In case A the half-bandwidth of nine is dictated by element 3 where the maximum nodal difference is eight (18-10). The half-bandwidth of 19 in example B is determined by element 1 where the difference is 18(19-1). Example A further demonstrates that when the minimum dimension convention is adhered to, the use of higher order elements is generally at the expense of an increased bandwidth.

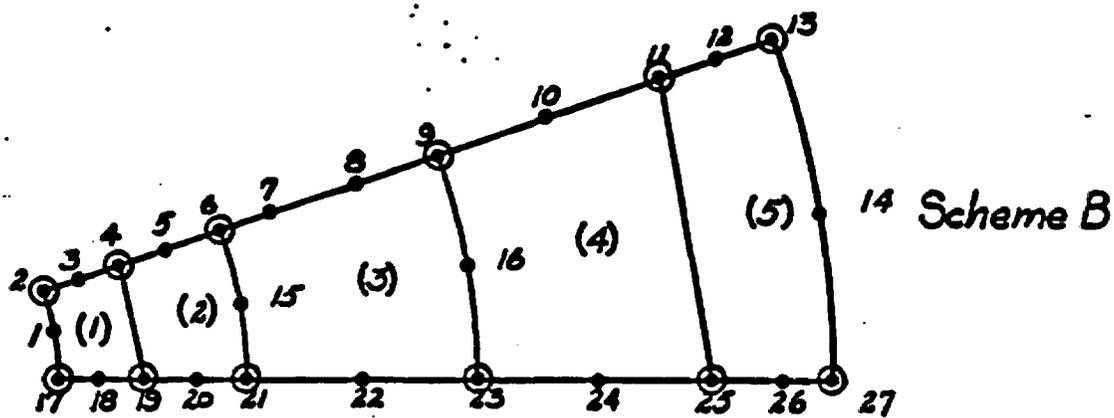
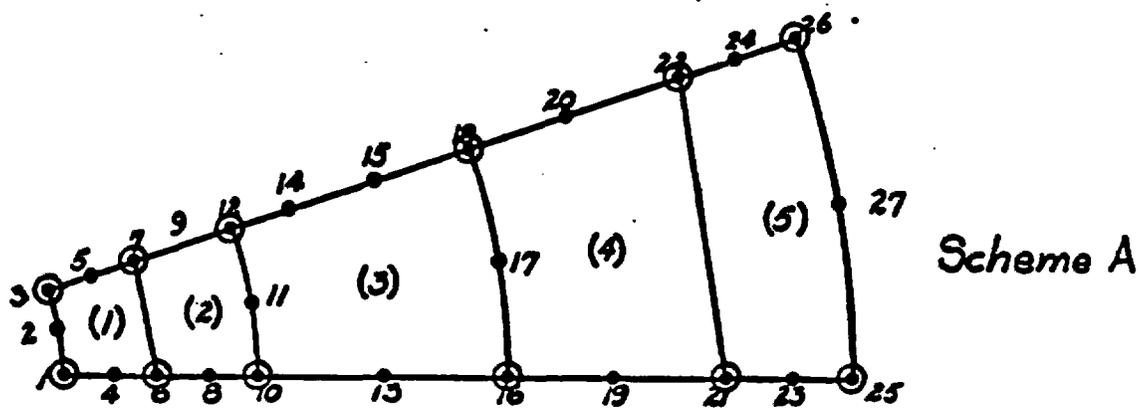
#### 2.4 Boundary Conditions

Constant head and constant flux are the two boundary conditions generally encountered in groundwater flow problems. The constant flux condition was mentioned earlier, and is easily incorporated by simply specifying a value for the last term in equation 2-9, which may be written

$$- \int_S \left( T_{xx} \frac{\partial \hat{h}}{\partial x} L_x + T_{yy} \frac{\partial \hat{h}}{\partial y} L_y \right) w_i dS = - \int_S q_n w_i dS \quad (2-15)$$

where  $q_n$  is the normal flux along  $S$ . When the flux  $q_n$  is assumed constant along an element face of length  $L$  the integration of (2-15) will give the nodal allocations indicated in Figure 4. When the total flux crossing the boundary is known, such as at a well bore, the allocations are made in the same way. In contrast to finite difference methods, a point sink can be represented using the Galerkin-finite element scheme by simply allocating the appropriate discharge to a specific node.

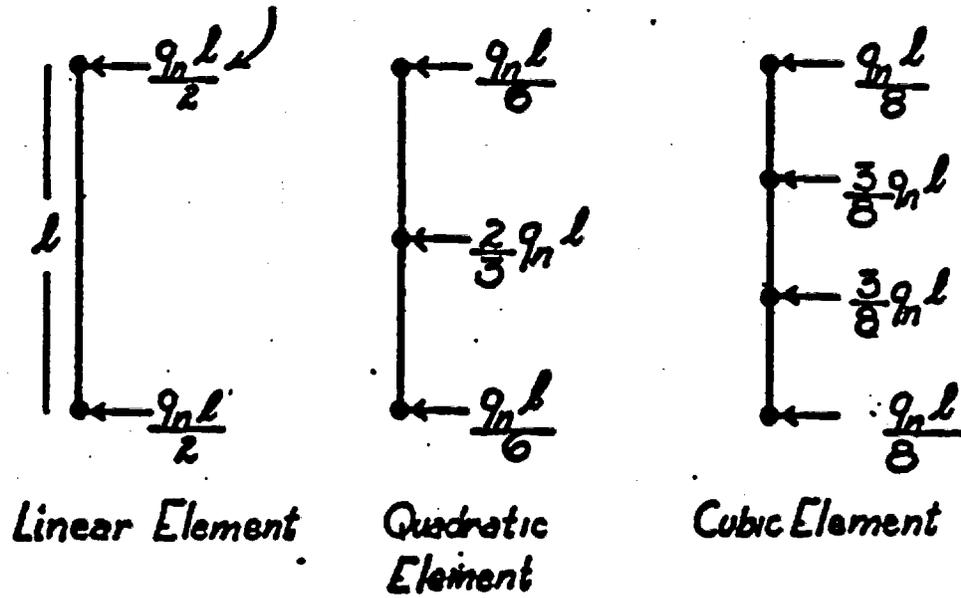
Constant-head boundaries are handled by specifying the initial head conditions at the appropriate boundary nodes and factoring out those rows and columns in the coefficient matrix associated with those nodes.



- Side node
- ⊙ Corner node
- ( ) Element number

**Fig. 3** Two identical element configurations with markedly different bandwidths dictated by the choice of nodal numbering scheme. Bandwidth of A is nine, of B is 19.

*Nodal Flux Allocations*



**Fig. 4** Allocation to boundary nodes of a constant normal flux  $q_n$  acting along an element of side length  $l$ .

Inasmuch as constant-head nodes are thus effectively eliminated from our matrix equation, we will be left with  $(N-M)$  equations in  $(N-M)$  unknowns with  $M$  being the number of constant-head boundary nodes.

## 2.5 Parameter Definition

Examination of equation 2-2 reveals that there are four aquifer parameters to be specified in order to simulate transient, two dimensional groundwater flow. These parameters can, in theory, be specified either for each node or for each element. Because the numerical solution requires slightly less computational effort when parameters are assumed constant over an element the storage coefficient  $S$ , the head in the adjacent aquifer  $h_w$ , and the ratio of vertical hydraulic conductivity to aquitard thickness  $\frac{K}{L}$ , are specified for each element. The transmissivity may be specified either by element or by node and this choice is left to the discretion of the user. A modest improvement in solution accuracy can be anticipated when nodal transmissivity is used.

## 2.6 Numerical Integration

Although the integration scheme used in the program is invisible to the user it is discussed here for the benefit of those readers who wish to examine and understand the computer code. As indicated earlier the basis functions  $w_i$  are defined, and the numerical integrations performed, in the local  $(\xi, \eta)$  coordinate system. Equation 2-9 is, unfortunately written in global  $(x, y)$  coordinates and must be transformed to the  $(\xi, \eta)$  system. To perform the transformation of the basis function derivatives  $\frac{\partial w_i}{\partial x}$  and  $\frac{\partial w_i}{\partial y}$  we use the relationship

$$\begin{Bmatrix} \frac{\partial w_i}{\partial x} \\ \frac{\partial w_i}{\partial y} \end{Bmatrix} = [J^{-1}] \begin{Bmatrix} \frac{\partial w_i}{\partial \xi} \\ \frac{\partial w_i}{\partial \eta} \end{Bmatrix} \quad (2-16)$$

where  $J$  is defined as

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (2-17)$$

Fortunately the matrix  $J$  can easily be evaluated numerically from the relationship

$$J = \begin{bmatrix} \frac{\partial w_1}{\partial \xi} & \frac{\partial w_2}{\partial \xi} & \dots & \frac{\partial w_n}{\partial \xi} \\ \frac{\partial w_1}{\partial \eta} & \frac{\partial w_2}{\partial \eta} & \dots & \frac{\partial w_n}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \quad (2-18)$$

where  $x_1, x_2, \dots$  and  $y_1, y_2, \dots$  are the nodal coordinates. This relationship (2-18) may be obtained by developing a series similar to (2-1) for the spatial variables  $x$  and  $y$ , and substituting this information into (2-17).

The only other transformation required is the replacement of the element of area  $dx dy$  by the relationship

$$dx dy = \det [J] d\eta d\xi. \quad (2-19)$$

The limits of integration in the local coordinate system become  $-1$  and  $+1$  in both integrals and we obtain for a typical element

$$A_{IJ} = \int_{-1}^1 \int_{-1}^1 \left( T_{xx} \frac{\partial w_i}{\partial x} \frac{\partial w_j}{\partial x} + T_{yy} \frac{\partial w_i}{\partial y} \frac{\partial w_j}{\partial y} + \left( \frac{K}{2} + C \right) w_i w_j \right) \det [J] d\eta d\xi \quad (2-20)$$

Similar expressions are developed for the remaining terms in 2-12 without the introduction of any additional transformations. Terms of the form of (2-20) are appropriate for integration by Gauss points. In this method of integration particular values of the function to be integrated are calculated at selected points in the domain of integration (In this case within the 2

by 2 square of the local coordinate system). These values are weighted in a prescribed manner and summed to give the value of the desired integral. The numbers of points required to provide an exact solution depends upon the degree of the polynomial being integrated. A polynomial  $f(x,y)$  of degree  $2N-1$  requires  $N^2$  points distributed evenly over the area of integration. A 4 by 4 rule is used in the accompanying groundwater flow program. For a more detailed discussion of Gauss points, their location and their weighting factors see Zienkiewicz and Cheung, 1967, p. 263.<sup>1</sup>

### 3. PROGRAM DOCUMENTATION

#### 3.1 Introduction

The accompanying computer code was developed by G. F. Pinder and E. O. Frind for solving transient, areal, groundwater flow. The theoretical foundation of the code and a comparison with an alternative finite-difference scheme is presented in Pinder and Frind, 1972.<sup>2</sup> Other modifications were recently added by P. C. Trescott of the U.S. Geological Survey, Reston, Virginia.

#### 3.2 Program Structure

The computer code is written in FORTRAN IV for use on the IBM 360-370 series computers. It consists of a main program and five subroutines, SHAPEI, CLAY, PLOT, DBAIDI, and SBANDI. SHAPEI generates the particular values of the shape functions, their derivatives and the determinant in  $(\xi, \eta)$  space for integration using Gauss points. CLAY provides the transient leakage coefficients necessary to simulate vertical flow into or out of the aquifer due to adjacent confining beds. PLOT presents information pertaining to nodes on an  $(x,y)$  plot generated on the line printer. DBAIDI places the

<sup>1</sup>Zienkiewicz, O. C. and Y. K. Cheung, The Finite Element Method in Structural and Continuum Mechanics, p. 263, McGraw-Hill, New York, 1967.

<sup>2</sup>Pinder, G. F. and E. O. Frind, Application of Galerkin's procedure to aquifer analysis, Water Resour. Res., 8(1), 1972.

coefficient matrix in upper triangular form. SBANDI uses the results obtained from DBANDI in conjunction with the known vector to back substitute for the desired solution at a particular time step.

Because the coding of the Galerkin-finite element method is somewhat difficult to follow care has been taken to provide adequate comment cards in the program.

### 3.3 Program Logic

Input to the program is straight-forward with the exception of the element incidence, which is a description of the nodal arrangement around each element. Because this information is read in with a "free" format, some effort must be expended by the computer in translating the card image. In addition the use of mixed elements requires the condensation of the unused nodes around each element. Twelve nodes must be initially assumed for each element and, depending upon the order of basis function assumed for each side, from four to twelve of these nodes may be actually used.

All operations involving numerical integration are performed element by element. The first step in the computational algorithm, therefore, is the calculation of information required for numerical integration. This information is obtained for all nodes on a given element. The integrations indicated in (2-12) are now performed and element coefficient matrices generated. These matrices contain all the coefficient information for the element under consideration and, because these matrices are symmetric, only half of the elements need be calculated. The information from the element coefficient matrices are allocated to the global coefficient matrices, which are used in solving the  $(N-N)$  simultaneous equations. When boundary conditions or wells are encountered this information is placed in the known vector. The matrix equation 2-14 is solved by upper-triangularization and back-substitution. Triangularization is necessary

only once provided the coefficient matrices are not modified. Unfortunately the matrix B depends upon  $\Delta t$  and a tradeoff results between the computational advantages of changing the time step and the additional effort of upper triangularization. Back substitution requires relatively little computational effort.

The printer-plotting capability is provided because the numerical results are, in general, difficult to interpret due to the arbitrary arrangement of the nodes. To permit the individual representation of nodes in close proximity, only three significant figures can be used to describe the plotted parameters. Although this constraint limits the usefulness of the plot for detailed analysis, it nevertheless provides a convenient mechanism for quickly evaluating the numerical results.

#### 4.1 Input Data

#### 4.2 Data Set 1: Identification

<u>Variable</u>	<u>Format</u>	<u>Variable Description</u>
CARD	(20A4)	Data set identification; punch ISOQUAD in the columns 1-7.

#### 4.3 Data Set 2: Title

TITLE	(20A4)	Title information; information typed on this card is reproduced exactly as a heading on printout.
-------	--------	---

#### 4.4 Data Set 3: Element Identification

NN	15	Number of nodes
NE	15	Number of elements
NS	15	Number of constant head nodes
NB*	15	Global coefficient matrix half-band width
NF	15	Number of source or sink nodes
NL	15	Number of leaky elements

#### 4.5 Data Set 4: Time Identification

TIME	F10.0	Duration of simulation period in hours
DELT	F10.0	Initial time step $\Delta t$ in hours
CHNG**	F10.0	Multiplier used to change size of time step according to relationship $\Delta t_i = \text{CHNG} * \Delta t_{i-1}$
ITMAX	I10	Maximum permitted number of time steps
ITCHNG	I10	Number of time steps between changes in $\Delta t$ .

\* See text for method of approximating this parameter.

\*\* A suggested value for this parameter is 1.5.

#### 4.6 Data Set 5: Control Codes \*

K0D1	15	Consider as watertable problem
K0D2	15	Printout of element coefficient matrices
K0D3	15	Printout of global coefficient matrices
K0D4	15	Punchout of calculated head values at normal termination of simulation
K0D5	15	Require steady-state solution only: matrix equation is solved only once
K0D6	15	Printer plot of computer drawdown at each time step
K0D7	15	Printer plot of computed head at each time step
K0D8	15	Printout of known vector in matrix equation
K0D9	15	Read nodal transmissivity**

#### 4.7 Data Set 6: Nodal Co-ordinates

FACTSR	F10.0	Multiplier for nodal coordinates: often used as a scaling factor for changing units i.e. changing feet to meters.
--------	-------	---

*new card*

J	15	Nodal number
X(J)	G10.0***	X coordinates of node J
Y(J)	G10.0	Y coordinates of node J

Note 1: For each node there is one card containing the above three variables (there should be NM cards)

#### 4.8 Data Set 7: Discharge at Source and Sink Nodes

IQ	15	Node where discharge is specified
FQ(IQ)	G10.0	Specified discharge (this may be specified along boundaries as well as at point sources and sinks)

Note 2: The above two variables are specified for each discharge node, one card for each node (there should be NF cards). Discharging wells are negative.

\*

A value of 1 initiates procedure, a value of zero suppresses procedure.

\*\*

When transmissivity is provided by element only, K0D9 should read zero so that data set 9 may be omitted.

\*\*\*

A G format is equivalent to an j, F, or E format specification. See Section 4.17.

#### 4.9 Data Set 8: Initial Head

UI (8F10.0) Initial head values.

Note 3: There will be 8 initial head values per card and NH/8 cards.

#### 4.10 Data Set 9: Nodal Transmissivity

PTRANS (8F10.0) Transmissivity values defined at nodal locations.

Note 4: There will be 8 transmissivities per card and NH/8 cards. When transmissivity is defined by element, and KOD9 is specified as zero this data set may be omitted. When a watertable aquifer is assumed, this data set is interpreted as hydraulic conductivity.

#### 4.11 Data Set 10: Base Elevation

BASE (8F10.0) The elevation of the base of a water table aquifer

Note 5: There will be 8 values per card and NH/8 cards. When transmissivity is defined by element, and KOD9 is specified as zero, this data set is omitted. When a watertable aquifer is not assumed, and KOD1 is zero, this data set is omitted.

#### 4.12 Data Set 11: Element Transmissivity and Storage

L	I3	Element number
TRANS (L)	F6.0	Transmissivity defined as a constant over the element L
STORAG(L)	F6.0	Storage coefficient defined as a constant over the element L

Note 6: The above information is recorded for each of the NE elements with one card required for each element. Recording a zero value for TRANS (L) indicates to the program that nodal transmissivity is to be used. When a watertable aquifer is assumed STORAG is the specific storage.

#### 4.13 Data Set 12: Element Incidences

L	I3	Element Number
CHAR	.free format	Corner and side nodes located on element L.

Note 7: The node numbers are recorded beginning at a corner node and moving counterclockwise around the perimeter of the element. Each corner node is followed immediately by a dollar sign and the first node is repeated. One or more blanks must appear between each nodal number or nodal number plus dollar sign. As an example the element appearing in Figure 1 would be described as 1\$ 4 7\$ 6\$ 5 3 2\$ 1\$. Thus the order of each element side is completely defined by the element incidence.

#### 4.14 Data Set 13: Constant Head Boundary Nodes

LRT	I4	Constant head node
-----	----	--------------------

Note 8: A maximum of twenty values are recorded per card. This data set thus requires NS/20 cards. When there are no constant head boundary nodes this data set may be omitted. An error message is generated when the number of non-zero values is not equal to NS.

#### 4.15 Data Set 14: Leaky Element Parameters

HYCOND	F15.0	Hydraulic conductivity of confining bed
SS	F15.0	Specific storage of confin- ing bed

new card

I	15	Leaky element number
THK (I)	G10.0	Thickness of adjacent confining bed
HZERØ (I)	G10.0	Hydraulic head in adjacent aquifer

Note 9: The above information is recorded for each "leaky element," one element per card, resulting in NL cards in this data set. When no leaky elements occur this data set is omitted.

#### 4.16 Data Set 15: Information for X-Y Plot

<b>HPP</b>	<b>I5</b>	HPP is one for head plot only, two for drawdown plot only, three for both head and drawdown plots.
<i>new card</i>		
<b>XLABEL</b>	<b>2A8</b>	Up to 16 alphameric characters for the X-axis label
<b>YLABEL</b>	<b>3A8</b>	Up to 24 alphameric characters for the Y-axis label
<b>TITLE</b>	<b>5A8</b>	Up to 40 alphameric characters for the plot title

**Note 10:** For plots of both head and drawdown, two cards with the above information are required, with the information for the head plot on the first card and for the drawdown plot on the second. Data set 15 may be omitted if KOD6 and KOD7 are zero.

<i>new card</i>		
<b>XMAX</b>	<b>G10.0</b>	Maximum value on X axis (usually slightly greater than the maximum X-coordinate value so that nodal values appear inside plot border)
<b>XMIN</b>	<b>G10.0</b>	Minimum value on X-axis (slightly less than minimum X-coordinate value)
<b>NXS</b>	<b>G10.0</b>	Number of segments on X-axis (if NXS is one, only the border lines will be plotted)
<b>NINX</b>	<b>G10.0</b>	Number of inches per X segment (if NXS is one, NINX is the number of inches between borders)
<b>YMAX</b>	<b>G10.0</b>	Maximum value on Y axis (slightly greater than maximum Y-coordinate value)
<b>YMIN</b>	<b>G10.0</b>	Minimum value on Y axis (slightly less than minimum Y-coordinate value)

Data Set 15: (continued)

NYS	G10.0	Number of segments on Y-axis
NINY	G10.0	Number of inches per Y segment (if NYS is one, NINY is the number of inches between the borders note the requirement $NYS * NINY \leq 12$ )
<i>new card</i>		
KK(1)	G10.0	Number of values to be plotted (i.e. number of nodes in model, NN)
CMAx	G10.0	Controls the smallest significant figure plotted for head
ADDN	G10.0	Controls smallest significant figure plotted for drawdown (analogous to CMAx)

Note 11: As an example for the use of CMAx consider the head value 1.23456. The plotted number versus CMAx value selected is presented in Table 1.

<u>CMAx</u>	<u>Plotted Number</u>
0.01	1
0.1	12
1.0	123
10.0	234*

\* note that a maximum of three figures are plotted.

Table 1. Tabulated values of CMAx and plotted values resulting from the number 1.23456.

4.17 Data Set 18: End of Data Identification

CARD	(20A4)	End of data is indicated by punching XXXX on this card which appears at the end of the last data set. If more than one problem is being run this card does not appear between problems.
------	--------	---

#### 4.18 Miscellaneous Comments on Data Input

Although the G format allows considerable flexibility in data input, it is important to recognize that when an E or I format specification is implied the rules governing these specifications must be followed. Note particularly the numbers read using E or I specification must be right justified in their field.

Although a satisfactory element can be generated out of very irregular geometry, there are occasions when a particular element geometry will lead to an ill-conditioned matrix. When this situation arises the upper triangularization algorithm will again fail. Unfortunately those elements with the most extreme shapes are not, necessarily, responsible for the ill-conditioning. One recognized source of difficulty involves the location of side nodes. While we have not discussed specifically the location of side nodes, it has been tacitly assumed that they would be placed such as to bisect a side in the case of a quadratic element or to trisect a side in the case of a cubic element. In practice these restrictions are unnecessary and one has considerable freedom in locating nodes along a side.

An ill-conditioned matrix may be generated, however, by locating a side node too near a corner node.

Figures 6 and 7 are provided for those users totally unfamiliar with card input to a digital computer. Figure 6 is a diagrammatic representation of a card deck and a sample coding form is presented in Figure 7. Figure 8 is a printout of the data presented in Figure 7.

To facilitate modification of the program to accommodate problems of varying size, the accompanying program uses object time dimension statements. As a result only one dimension statement (the first in the main program) is dependent upon problem size. Table 2 is provided to assist the user in making changes to minimize the storage requirements of the program where NN is the number of nodes, NB is the half-band width, N is the degrees of freedom and NE is the number of elements.

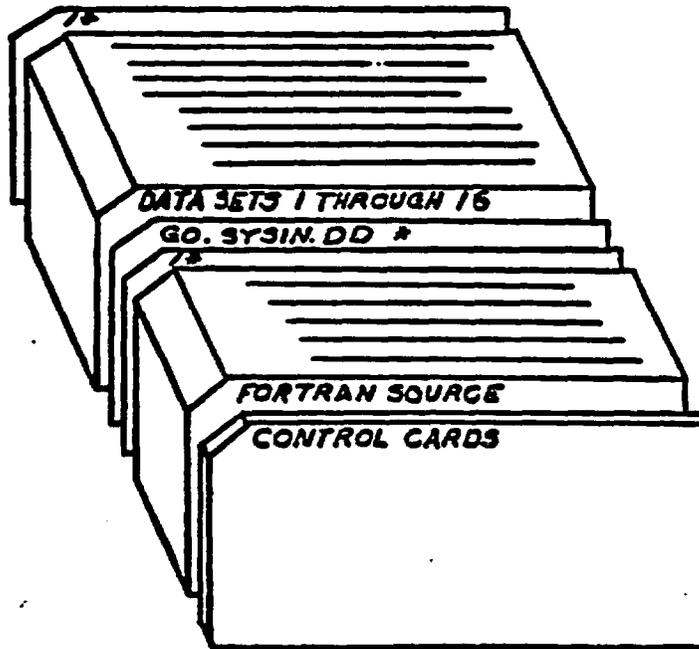
S(N,NB)	X(NN)	DDN(NN)
P(N,NB)	UI(NN)	STORAG(NE)
H(N,NB)	FM(NN)	XX(1,NN)
IN(13,NE)	QLD(NN)	YY(1,NN)
TRANS(NE)	PQ(NN)	CC(NN)
THK(NE)	RT(NN)	INDEX (NN)
LR(NN)	INPLQ(NE,5)	U(NN)
LRC(NN)	HZERQ(NE)	UQ(NN)
Y(NN)	PTRANS(NN)	QCQP(NN)

Table 2. Major arrays with dimensions indicated symbolically by variables from data set 3.

#### 4.19 Printed Output

While the printed output is basically self-explanatory a few additional comments may be warranted. The nodal coordinates printed are identical to those read and have not been multiplied by the scaling factor. The degrees of freedom printed under the heading "Finite Element Data" refer to the number of equations to be solved and differ from the total number of nodes by the number of constant-head nodes where, as indicated earlier, no equation is generated. The global coefficient half-bandwidth presented

immediately preceding the first computed head solution is the actual half-band required as contrasted to the estimated value provided as input to the program.



**Fig. 6** Diagrammatic representation of a computer card deck for input to ISOQUAD 2.



```

C *****
C *                               I S O Q U A D 2                               *
C *                               JANUARY 1974                               *
C *                               REVISED AUGUST, 1974                       *
C *                               REVISED AUGUST 1976                       *
C *  PURPOSE: TO SIMULATE TRANSIENT, AREAL, GROUNDWATER FLOW USING        *
C *             MIXED ISOPARAMETRIC ELEMENTS.                               *
C *
C *  PROGRAMMED BY:
C *             G.P. PINDER, DEPT. OF CIVIL ENGINEERING, PRINCETON        *
C *             UNIVERSITY, PRINCETON N.J. 08540 AND U.S.G.S.             *
C *             RESTON, VA.
C *             E.O. FRIND, DEPT. OF EARTH SCIENCES, UNIVERSITY OF        *
C *             WATERLOO, WATERLOO, ONTARIO, CANADA
C *             WITH RECENT MODIFICATIONS BY
C *             P.C. TRESCOTT, U.S.G.S., RESTON, VA.
C *****
C
C *****
C *  ADDN.....PLOTTING CONTROL FOR DRAWDOWN                               *
C *  AG.....ARRAY OF INTEGRATION POINT LOCATIONS                          *
C *  BASE.....ELEVATION OF BASE OF AQUIFER FOR WATER TABLE PROBLEM       *
C *  CHAR.....CHARACTER ARRAY FOR INCIDENCES                              *
C *  CHNG.....MULTIPLIER FOR INCREASING TIME STEP (DELT=CHNG*DELT)        *
C *  CMAX.....PLOTTING CONTROL FOR HEAD                                    *
C *  DDDN.....PROJECTED HEAD FOR CALCULATION OF TRANSMISSIVITY            *
C *  DELT.....INITIAL TIME STEP IN HOURS                                  *
C *  DET.....DETERMINANT FOR POINT XI,YI                                  *
C *  DIFF.....COMPUTED RESIDUAL OF MASS BALANCE                           *
C *  DGI.....X DERIVATIVE OF BASIS FUNCTION FOR ALL ACTIVE NODES IN      *
C *             ELEMENT EVALUATED AT XI,YI                                  *
C *  DGY.....Y DERIVATIVE EQUIVALENT OF DGI                               *
C *  DLEAKS...STEADY LEAKAGE FOR ONE TIME STEP                             *
C *  DLEAKT...TRANSIENT LEAKAGE FOR ONE TIME STEP                         *
C *  DSTOR...CHANGE IN STORAGE FOR ONE TIME STEP                          *
C *  FACTOR...MULTIPLIER FOR X AND Y COORDINATES                           *
C *  PP.....BASIS FUNCTION FOR ALL ACTIVE NODES IN ELEMENT, EVALUATED*
C *             AT XI,YI
C *  PQ.....SOURCE AND SINK (DISCHARGE IS NEGATIVE)                       *
C *  H11.....WEIGHTING COEFFICIENT FOR NUMERICAL INTEGRATION              *
C *  H12.....WEIGHTING COEFFICIENT FOR NUMERICAL INTEGRATION              *
C *  H22.....WEIGHTING COEFFICIENT FOR NUMERICAL INTEGRATION              *
C *  HYCOND...HYDRAULIC CONDUCTIVITY OF CONFINING BED                     *
C *  HZERO...HEAD IN ADJACENT AQUIFER                                       *
C *  IPLO...LOCATION OF CONSTANT HEAD NODES                                  *
C *  IN.....ELEMENT INCIDENCE
C *  ISTEADY...TEST FOR STEADY-STATE LEAKAGE                               *
C *  IT.....NUMBER OF TIME STEPS CALCULATED                               *
C *  ITCNG...NUMBER OF TIME STEPS BETWEEN CHANGES IN DELT               *
C *  ITMAX...MAXIMUM PERMITTED NUMBER OF TIME STEPS                       *
C *  JBNH...HALFBANDWIDTH (INCLUDES DIAGONAL)                             *
C *  JD.....ARRAY OF ACTIVE NODES IN ELEMENT                              *
C *  JNDP...DEGREES OF FREEDOM (NODES LESS CONSTANT HEAD NODES)          *

```

```

C * JNZL.....NUMBER OF ELEMENTS * 0
C * JNND.....NUMBER OF NODES * 0
C * KK.....NUMBER OF VALUES TO BE PLOTTED * 0
C * KODI.....CONTROL CODES: 1 INITIATES PROCEDURE, * 0
C * 0 SUPPRESSES PROCEDURE * 0
C * KOD1...CONSIDER WATERTABLE PROBLEM * 0
C * KOD2...PRINT ELEMENT COEFFICIENT MATRICES * 0
C * KOD3...PRINT GLOBAL COEFFICIENT MATRIX * 0
C * KOD4...PUNCH CALCULATED HEAD AT END OF RUN * 0
C * KOD5...ASSUME STEADY STATE PROBLEM * 0
C * KOD6...PLOT COMPUTED DRAWDOWN * 0
C * KOD7...PLOT COMPUTED HEAD * 0
C * KOD8...PRINT KNOWN VECTOR (IE. RIGHT HAND SIDE OF EQUATION) * 0
C * KOD9...READ NODAL TRANSMISSIVITY * 0
C * L.....INDEX INDICATING ELEMENT NUMBER * 0
C * LRC.....CUMULATIVE SUMMATION FOR CONSTANT HEAD VALUES * 0
C * LR.....INDICATOR ARRAY FOR CONSTANT HEAD VALUES * 0
C * M.....NUMBER OF ACTIVE NODES ON ELEMENT * 0
C * NBI.....CALCULATED HALF-BAND WIDTH * 0
C * NINX.....NUMBER OF INCHES PER X SEGMENT OF PLOT * 0
C * NINY.....NUMBER OF INCHES PER Y SEGMENT OF PLOT * 0
C * N.....NUMBER OF DEGREES OF FREEDOM (NN-NE) * 0
C * NB.....GLOBAL MATRIX HALFBAND WIDTH (ESTIMATED) * 0
C * NE.....NUMBER OF ELEMENTS * 0
C * NF.....NUMBER OF SOURCE OR SINK NODES * 0
C * NL.....NUMBER OF LEAKY ELEMENTS * 0
C * NN.....NUMBER OF NODES * 0
C * NP.....NUMBER OF INTEGRATION POINTS PER COORDINATE AXIS * 0
C * NPP.....CONTROL FOR PLOT; 1 FOR HEAD, 2 FOR DRAWDOWN, 3 FOR BOTH * 0
C * NR.....NUMBER OF CONSTANT HEAD NODES (ACTUALLY READ) * 0
C * NS.....NUMBER OF CONSTANT HEAD NODES (REQUESTED) * 0
C * NXS.....NUMBER OF SEGMENTS ON X AXIS OF PLOT * 0
C * NYS.....NUMBER OF SEGMENTS ON Y AXIS OF PLOT * 0
C * OLD.....COMPUTED HEAD FROM PREVIOUS TIME STEP (CONDENSED) * 0
C * P.....GLOBAL COEFFICIENT MATRIX ASSOCIATED WITH TIME DERIVATIVE * 0
C * PE.....ELEMENT MATRIX ASSOCIATED WITH TIME DERIVATIVE * 0
C * PTRANS...NODAL TRANSMISSIVITY * 0
C * PUMP.....SUM OF SOURCES AND SINKS * 0
C * QCOEF...TRANSIENT LEAKAGE COEFFICIENT * 0
C * QCOF...ARRAY OF TRANSIENT LEAKAGE COEFFICIENTS * 0
C * RT.....VECTOR CONTAINING SOURCES, SINKS, AND BOUNDARY CONDITIONS * 0
C * S.....GLOBAL COEFFICIENT MATRIX ASSOCIATED WITH SPACE DERIVATIVE * 0
C * SE.....ELEMENT MATRIX ASSOCIATED WITH SPACE DERIVATIVE * 0
C * SHIN.....ELAPSED TIME IN MINUTES * 0
C * SRCR.....CONTRIBUTION OF ELEMENT TO KNOWN VECTOR * 0
C * SS.....SPECIFIC STORAGE OF CONFINING BED * 0
C * SSEC.....ELAPSED TIME IN SECONDS * 0
C * SSEC1...ELAPSED TIME IN SECONDS * 0

```

```

C * STINE....ELAPSED TIME IN HOURS * 1
C * STORAG...STORAGE COEFFICIENT * 1
C * THK.....THICKNESS OF CONFINING LAYER * 1
C * TIME.....SIMULATION PERIOD IN HOURS * 1
C * TK.....TRANSMISSIVITY EVALUATED AT INTEGRATION POINT * 1
C * TRANS....ELEMENT TRANSMISSIVITY * 1
C * U.....CALCULATED HEAD (EXPANDED) * 1
C * UI.....INITIAL HEAD (EXPANDED) * 1
C * UIK.....INITIAL HEAD EVALUATED AT INTEGRATION POINT * 1
C * UK.....HEAD EVALUATED AT INTEGRATION POINT * 1
C * UKO.....HEAD FROM PREVIOUS TIME STEP EVALUATED AT INTEGRATION PT.* 1
C * UO.....COMPUTED HEAD FROM PREVIOUS TIME STEP (EXPANDED) * 1
C * X .....X COORDINATE OF NODE * 1
C * XI.....X COORDINATE OF INTEGRATION POINT * 1
C * XLABEL...LABEL FOR X AXIS OF PLOT * 1
C * XMAX.....MAXIMUM VALUE ON X AXIS OF PLOT * 1
C * XMIN.....MAXIMUM VALUE ON Y AXIS OF PLOT * 1
C * Y .....Y COORDINATE OF NODE * 1
C * YI.....Y COORDINATE OF INTEGRATION POINT * 1
C * YLABEL...LABEL FOR Y AXIS OF PLOT * 1
C * YMAX.....MAXIMUM VALUE ON Y AXIS OF PLOT * 1
C * YMIN.....MINIMUM VALUE ON Y AXIS OF PLOT * 1
C ***** * 1
C * 1
C....MAIN PROGRAM FOR INITIATING OBJECT TIME DIMENSION STATEMENTS * 1
  DIMENSION DGX(12),DGY(12),P(12) * 1
  INTEGER*2 IN * 1
C * 1
C +-----THE ENCLOSED CARDS ARE PROBLEM DEPENDENT-----+ * 1
  DIMENSION S(50,25),P(50,25),H(50,25),IN(13,30),XX(1,50),YY(1,50), * 1
  1IPLO(30,5),LRC(50),Y(50),X(50),UI(50),PH(50),OLD(50),FO(50),RT(50) * 1
  2,PTRANS(50),DDDN(50),CC(50),INDEX(50),U(50),UO(50),QCOF(50),TRANS( * 1
  330),FK(30),HZERO(30),STORAG(30),LR(50),BASE(50) * 1
  JNDF=50 * 1
  JNEL=30 * 1
  JHND=50 * 1
  JBWH=25 * 1
C +-----+ * 1
C * 1
  CALL SHAPI (JNEL,JHND,JBWH,JNDF,S,PM,IN,U,X,Y,DGX,DGY,P) * 1
  CALL DBANDI (JNEL,JHND,JBWH,JNDF,S,PM,IN,U,X,Y) * 1
  CALL SBANDI (JNEL,JHND,JBWH,JNDF,S,PM,IN,U,X,Y) * 1
  CALL MPRCG (JNEL,JHND,JBWH,S,P,H,IN,TRANS,THK,LRC,Y,X,UI,PH,U,OLD, * 1
  1 PQ,RT,IPLO,HZERO,PTRANS,DDDN,STORAG,II,YY,CC,INDEX, * 1
  2 UO,QCOF,DGX,DGY,P,LR,JNDF,BASE) * 1
  STOP * 1
  END * 1
  SUBROUTINE MNPORG (JNEL,JHND,JBWH,S,P,H,IN,TRANS,THK,LRC,Y,X,UI,PH, * 1

```

1 U,OLD,FQ,RT,IFLO,HZERO,PTRANS,DDDN,STORAG,IX,YY,  
2 CC,INDEX,UO,QCOF,DGX,DGY,FF,LR,JNDF,BASE)

C  
C

```

=====
DIMENSION S (JNDF,JBWH), P (JNDF,JBWH), H (JNDF,JBWH), IN (13, JNEL),
1 TRANS (JNEL), THK (JNEL), LRC (JNND), Y (JNND), X (JNND),
2 UI (JNND), FM (JNND), U (JNND), OLD (JNND), FO (JNND), RT (JNND),
3 HZERO (JNEL), PTRANS (JNND), DDDN (JNND), STORAG (JNEL),
4 IX (1, JNND), YY (1, JNND), CC (JNND), INDEX (JNND), UO (JNND),
5 QCOF (JNND), IFLO (JNND, 5), LR (JNND), BASE (JNND)
DIMENSION TITLE (20), CHAR (65), DIGIT (10), AG (6), COR (15), DGX (12),
1 DGY (12), DX (12, 16), DY (12, 16), DETJ (16), Q (16), SE (12, 12),
2 R (12), KR (50), JD (12), LRT (50), P (12, 16), PE (12, 12), FF (12),
3 QP (16), SRCRT (16), SRCLT (16), SPCR (12), TK (16), UIK (16),
4 UK (16), UKO (16), ZL (16), TS (16), TT (16)
COMMON/AREA1/ XLABEL (3, 2), YLABEL (3, 3), TATLE (3, 5), NPP, CMAX, XMAX,
1 XMIN, YMAX, YMIN, NXS, NINX, NYS, NINY, NDS, KK (10)
COMMON /WEIGHT/ SUM1, SUM2, SUM3, SUM4
INTEGER*2 IN
REAL*8 XLABEL, YLABEL, TATLE, AG, H1, H2, H11, H12, H22

```

C

```

DATA BLANK/1H /, STAR/1HS/, DIGIT/1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7,
1 1H8, 1H9/, START/4HIS00/, EXIT/4HXXX/, AG/-.577350269189626,
2 .577350269189626, -.861136311594053, -.339981043584856,
3 .339981043584856, .861136311594053/, H1/.347854845137454/,
4 H2/.652145154862546/, COR/-1., 0., 1., 0., -1., -1., -1., -.3333333,
5 .3333333, 1., 1., .3333333, -.3333333, -1., -1./, NST/0/, KR (1)/0/,
6 QCDEF/0./
=====

```

C  
C

```

C.....READ CONTROL DATA
10 READ (5,790) CARD
IF (CARD.EQ.EXIT) STOP
IF (CARD.NE.START) GO TO 10
READ (5,780) TITLE

```

C

```

READ (5,790) NW,NE,NS,NB,NF,NL
WRITE (6,800)
WRITE (6,810)
WRITE (6,820) TITLE
WRITE (6,1060)
WRITE (6,830) NW,NE,NS,NF,NB,NL

```

C

```

READ (5,840) TIME,DELTA,CHNG,ITMAX,ITCHNG
WRITE (6,850) TIME,DELTA,CHNG,ITMAX,ITCHNG
DELTA=DELTA*3600.

```

C

C.....INITIALIZE CONSTANTS



```

1520 WRITE (6,1521)
1531 CONTINUE
C
C.....READ AND NORMALIZE NODE COORDINATES
      READ(5,920) FACTOR
      READ (5,860) (J,X(J),Y(J),K=1,NN)
      WRITE (6,870)
      WRITE (6,871) FACTOR
      WRITE (6,880)
      WRITE (6,890) (J,X(J),Y(J),J=1,NN)
      DO 19 J=1,NN
        X(J)=X(J)*FACTOR
        Y(J)=Y(J)*FACTOR
C      INITIALIZE X AND Y VALUES FOR PLOT
        XX(1,J)=X(J)
        YY(1,J)=Y(J)
        INDEX(J)=J
      19 CONTINUE
C
C.....READ SOURCE AND SINK NODES
C.....SOURCES ARE POSITIVE
      WRITE (6,900)
      PUMP=0.
      DO 20 I=1,NN
        QCOF(I)=0.
      20 FQ(I)=0.
        IF (NP.22.0) GO TO 40
        DO 30 I=1,NP
          READ (5,860) IQ,FQ(IQ)
          PUMP=PUMP-FQ(IQ)
        30 WRITE (6,910) IQ,FQ(IQ)
      40 CONTINUE
C
C.....INITIAL HEAD VALUES
      READ (5,920) (UI(I),I=1,NN)
      WRITE (6,930)
      WRITE (6,940) (I,UI(I),I=1,NN)
C
C.....READ JOUAL TRANSMISSIVITY
      IF (KDD1.20.0) GO TO 1622
      READ (5,920) (PTRANS(I),I=1,NN)
      WRITE (6,931)
      WRITE (6,940) (I,PTRANS(I),I=1,NN)
      IF (KDD1.30.0) GO TO 1622
      READ (5,920) (BASE(I),I=1,NN)
      WRITE (6,1601)
      WRITE (6,940) (I,BASE(I),I=1,NN)
1601 FORMAT(////,11X,15HBASE OF AQUIFER/11X,15(1H-)//11X,6(4HNODE,5X,5H

```





```

WRITE (6,1070) NN,NR,N
C
C
C.....READ LEAKY ELEMENT COEFFICIENTS
DO 243 I=1,NE
HZERO(I)=0.
243 THK(I)=J.
IF (NL.EQ.0) GO TO 241
C
C.....READ CONFINED AQUIFER PARAMETERS
READ (5,921) HYCOND,SS
WRITE (6,7005) HYCOND,SS
DO 244 J=1,NL
244 READ (5,860) I,THK(I),HZERO(I)
WRITE (6,245) (I,THK(I),HZERO(I),I=1,NE)
241 CONTINUE
C
C.....READ DATA FOR X-Y PLOT OF RESULTS
IF(KOD7.NE.1.AND.KOD6.NE.1) GO TO 367
READ (5,730) NPP
GO TO (362,363,362),NPP
362 READ (5,365) (XLABEL(1,J),J=1,2), (YLABEL(1,J),J=1,3), (TATL(1,J),J=
11,5)
IF(NPP.EQ.1) GO TO 364
363 READ (5,365) (XLABEL(2,J),J=1,2), (YLABEL(2,J),J=1,3), (TATL(2,J),J=
11,5)
364 READ (5,366) XMAX,XMIN,NYS,NINY,YMAX,YMIN,NYS,NINY,RR(1),CMAX,ADDN
CALL PLOT(JNND,XX,YY,CC,INDEX)
C
C.....CALCULATE OLD FROM UI, THE INITIAL HEAD VALUES
367 II=0
DO 260 I=1,NN
U(I)=UI(I)
UO(I)=UI(I)
DDD(I)=UI(I)
IF (LR(I).GT.0) GO TO 260
II=II+1
OLD(II)=UI(I)
260 CONTINUE
C
C
C.....BEGIN CALCULATIONS FOR GLOBAL COEFFICIENT MATRICES
C.....CLEAR ARRAYS
999 DO 250 I=1,NN
DO 250 J=1,NB
P(I,J)=0.
250 S(I,J)=0.
C

```

C.....INITIALIZE KNOWN VECTOR TO SPECIFIED DISCHARGE

```

II=0
DO 992 I=1,NN
IF(LR(I).GT.0) GO TO 992
II=II+1
RT(II)=-PQ(I)
992 CONTINUE

```

C.....INITIALIZE CONSTANTS

```

NB1=0
LEKCHK=0
THKZD = 0.
KEX=0
254 TSTOJ=0.
TLEAKS=0.
TLEAKT=0.
L=0

```

C  
C

C.....BEGIN CALCULATION OF ELEMENT COEFFICIENT MATRICES

```

270 L=L+1
M=IN(13,L)
K=0
DO 430 I=1,M
420 K=K+1
IF (IN(K,L).EQ.0) GO TO 420
JD(I)=IS(K,L)
430 CONTINUE

```

C.....THE ARRAY JD NOW CONTAINS THE INCIDENCES OF THE ACTIVE NODES IN  
C ELEMENT L.

C.....INTEGRATION BY GAUSSIAN QUADRATURE

C 2 X 2 RULE FOR FULLY LINEAR ELEMENTS  
C 4X4 RULE FOR ALL OTHER ELEMENTS

C.....SELECT NUMBER OF GAUSS POINTS

```

NP=2
IF (4.3E.4) NP=4
NP2=NP*NP

```

C.....COMPUTE VALUE OF SHAPE FUNCTIONS AT INTEGRATION POINTS

```

DO 310 I=1,NP
DO 310 J=1,NP
K=(I-1)*NP+J
IF (NP.EQ.4) GO TO 280
XI=AG(J)
YI=AJ(I)
GO TO 230
280 XI=AJ(J+2)
YI=AG(I+2)

```



```

320 Q(K) = (Z1*(DX(I,K)*DY(J,K) + DY(I,K)*DX(J,K))) *DETJ(K)
      IF (NP.EQ.4) GO TO 330
      SE(I,J) = J(1)+Q(2)+Q(3)+Q(4)
      PE(I,J) = JP(1)+QP(2)+QP(3)+QP(4)
      SRCL=SRCLT(1)+SRCLT(2)+SRCLT(3)+SRCLT(4)
      GO TO 340
330 SE(I,J) = H11*(Q(1)+Q(4)+Q(13)+Q(16)) + H12*(Q(2)+Q(3)+Q(5)+Q(8)+Q(9) +
      1Q(12)+Q(14)+Q(15)) + H22*(Q(6)+Q(7)+Q(10)+Q(11))
      PE(I,J) = H11*(QP(1)+QP(4)+QP(13)+QP(16)) + H12*(QP(2)+QP(3)+QP(5)+QP(
      18)+QP(7)+QP(12)+QP(14)+QP(15)) + H22*(QP(6)+QP(7)+QP(10)+QP(11))
      SRCL=H11*(SRCLT(1)+SRCLT(4)+SRCLT(13)+SRCLT(16)) + H12*(SRCLT(2)
      2+SRCLT(3)+SRCLT(5)+SRCLT(8)+SRCLT(9)+SRCLT(12)+SRCLT(15)+SRCLT(14)
      3) + H22*(SRCLT(6)+SRCLT(7)+SRCLT(10)+SRCLT(11))
340 CONTINUE
      SE(I,J) = SE(I,J) + SRCL
350 CONTINUE
      DO 341 I=1,M
      IF (THK(L).NE.0.) GO TO 351
      SRCR(I) = 0.
      GO TO 341
351 DO 323 K=1,NP2
323 SRCRT(K) = (ZCOEF*UIK(K) + HCOND/THK(L) * (HZERO(L) - UIK(K))
      1) * F(I,K) * DETJ(K)
      IF (NP.EQ.4) GO TO 331
      SRCR(I) = SRCRT(1)+SRCRT(2)+SRCRT(3)+SRCRT(4)
      GO TO 341
331 SRCR(I) = H11*(SRCRT(1)+SRCRT(4)+SRCRT(13)+SRCRT(16)) + H12*(SRCRT(2)
      2+SRCRT(3)+SRCRT(5)+SRCRT(8)+SRCRT(9)+SRCRT(12)+SRCRT(15)+SRCRT(14)
      3) + H22*(SRCRT(6)+SRCRT(7)+SRCRT(10)+SRCRT(11))
341 CONTINUE
C
C.....FILL LOWER HALF OF SE AND PE ARRAYS
      DO 360 I=2,M
      I1=I-1
      DO 360 J=1,I1
      PE(I,J) = PE(J,I)
360 SE(I,J) = SE(J,I)
C
C.....PRINT ELEMENT COEFFICIENT MATRICES
      IF (MOD2.NE.1) GO TO 410
      WRITE (6,1100) L
      MM2=(M+7)/8*8-7
      DO 380 K1=1,MM2,8
      K2=K1+7
      IF (K1.EQ.MM2) K2=M
      DO 370 I=1,M
370 WRITE (6,1110) I, (SE(I,J), J=K1,K2)
380 WRITE (6,1170)

```







```

      NPP=2
C      *****
      CALL GRAPH
C      *****
614 IF (KJ07.NE.1) GO TO 615
C
C.....PRINT MAP OF COMPUTED HEAD VALUES
      DO 616 I=1,NN
616 CC(I)=J(INDEX(I))*CHX
      NPP=1
C      *****
      CALL GRAPH
C      *****
615 CONTINUE
C
C
C.....ELAPSED TIME CHECK
      DELIP=DELPT
      IF (STIME.LT.TIME) GO TO 740
      WRITE (6,1240) STIME
      IFINAL=1
      GO TO 770
740 IF (IT.LT.ITMAX) GO TO 750
      WRITE (6,1250) IT
      IFINAL=1
      GO TO 770
750 IF (MOD(IT,ITCHNG).GT.0) GO TO 752
      DELT=CHNG*DELT
C
752 WRITE (6,753) IT
754 IF (KJ01.NE.0) GO TO 254
      IF (MOD(IT,ITCHNG).GT.0) GO TO 560
      GO TO 540
770 IF (KJ04.NE.1) GO TO 771
      WRITE (7,1051) (U(I),I=1,NN)
771 IF (KJ01.EQ.0) GO TO 10
      GO TO 254
C
245 FORMAT (////11X,39HTHICKNESS AND HEAD IN OVERLYING AQUIFER/
      111X,39(1H-)/11X,3(4HNODE,3X,9HTHICKNESS,4X,4HHEAD,4X)//(11X,
      23(13H,1PE9.2,2X,1PE9.2)))
365 FORMAT(1JA8)
366 FORMAT(9J10.0)
481 FORMAT (1H0,10X,37HCUMULATIVE MASS BALANCE FOR TIME STEP ,I4,1H:,
      1/11X,42(1H-))
482 FORMAT (1H0,10X,8HSTORAGE=,G25.5/11X,15HSTEADY LEAKAGE=,G18.5/11X,
      118HT3ANISIENT LEAKAGE=,G15.5/11X,14HCONSTANT HEAD=,G19.5/11X,
      215HSUM OF SOURCES=,G18.5/11X,13HPUMPING RATE=,320.5/11X

```

```

3,11HDIFFERENCE=G22.5)
582 FORMAT (1H0,10X,15HKNOWN VECTOR FN/11X,15(1H-)/)
753 FORMAT (1H0,10X,36H***** STEADY LEAKAGE AT TIME STEP ,I4,8H****
1****)
780 FORMAT (20A4)
790 FORMAT (10I5)
800 FORMAT (1H1,36X,15HS O Q U A D 2//33X,26HGROUND-WATER FLOW ANALY
1SIS//25X,41HWITH ISOPARAMETRIC QUADRILATERAL ELEMENTS//11X,70(1H*)
2//)
810 FORMAT (1H ,30X,30HDIFFUSION EQUATION, MARCH 1971)
820 FORMAT (11X,70(1H*)//11X,20A4//11X,70(1H*)//)
830 FORMAT (1H ,10X,11HNUMBER OF -,2X,5HNODES,I24/21X,1H-,2X,8HELEMENT
1S,I21/21X,1H-,2X,24HCONSTANT HEAD BOUNDARIES,I5/21X,1H-,2X,20HSOUR
2CE OR SINK NODES,I9/21X,1H-,2X,20HELEMENTS IN HALFBAND,I9/21X,1H-,
22X,24HNUMBER OF LEAKY ELEMENTS,I5//)
840 FORMAT (3F10.0,2I10)
850 FORMAT (//////11X,15HTIME PARAMETERS/11X,15(1H-)//11X,26HSIMULATION
1 PERIOD IN HOURS,P30.2/11X,26HINITIAL TIME STEP IN HOURS P30.6/11X,
235HMULTIPLIER FOR INCREASING TIME STEPP21.3/11X,38HMAXIMUM PERMITT
3ED NUMBER OF TIME STEPS I18/11X,44HNUMBER OF TIME STEPS BETWEEN CHA
4NGES IN DELTI12//)
860 FORMAT (I5,2G10.0)
870 FORMAT (1H0,10X,16HNODE COORDINATES/11X,16(1H-)/)
871 FORMAT (11X,33HMULTIPLICATION FACTOR FOR X AND Y,G10.0/)
880 FORMAT (1H ,11X,4HNODE,12X,1HX,15X,1HY/)
890 FORMAT (1H ,10X,I4,2F16.2)
900 FORMAT (//////11X,27HSOURCE(+) AND SINK(-) NODES/11X,27(1H-)/11X,4H
1NODE,7X,14HDISCHARGE(CFS)/)
910 FORMAT (11X,I4,P16.3)
920 FORMAT (3F10.0)
921 FORMAT (5F15.0)
930 FORMAT (//////,11X,12HINITIAL HEAD/11X,12(1H-)//11X,6(4HNODE,5X,5HV
1ALUE,5A))
931 FORMAT (//////,11X,20HNODAL TRANSMISSIVITY/11X,20(1H-)//11X,6(4HNOD
1E,5X,5IVALUE,5X))
940 FORMAT (/ (11X,6(I4,2X,1PE10.3,3X)))
950 FORMAT (I3,2F6.0)
951 FORMAT (I3,65A1)
952 FORMAT (I3,P6.0,6X,P6.0)
960 FORMAT (1H0,10X,17HINVALID CHARACTER,2X,A1,5X,7HELEMENT,I5)
970 FORMAT (1H0,10X,25HINCOMPLETE DATA - ELEMENT,I5)
980 FORMAT (//////11X,53HELEMENT INCIDENCES, STORAGE COEFFICIENT AND TRA
1NSMISSIVITY/11X,58(1H-)/)
990 FORMAT (1H ,10X,7HELEMENT,5X,1H/,3(1H-),9H CORNERS ,4(1H-),1H/,7X,
11H/,14(1H-),7H SIDES ,15(1H-),1H/,2X,13HSTORAGE COEF.,2X,14HTRANSM
2ISSIVITY,/)
1000 FORMAT (1H ,4X,I10,5X,4I5,5X,8I5,3X,1PE9.2,6X,1PE10.3)
1010 FORMAT (//////11X,19HCONSTANT HEAD NODES/11X,19(1H-))

```

1020 FORMAT (20I4)  
 1030 FORMAT (11X,10(1H\*),19HCONSTANT HEAD NODEI4,37HDOES NOT EXIST - EX  
 ECUTION TERMINATEDI10(1H\*))  
 1040 FORMAT (11X,20I5)  
 1050 FORMAT (1H0,10X,10(1H\*),34HNUMBER OF CONSTANT HEAD NODES READI6,33  
 1HDISAGREES WITH NUMBER ANTICIPATEDI6,10(1H\*))  
 1051 FORMAT (3F10.5)  
 1060 FORMAT (///11X,19HFINITE ELEMENT DATA/11X,19(1H-)/)  
 1070 FORMAT (1I,10X,21HTOTAL NUMBER OF NODES,I9/11X,19HCONSTANT HEAD N  
 ODES,I4,I10/11X,18HDEGREES OF FREEDOM,I12//)  
 1100 FORMAT (////,10X,7HELEMENT,I4,5X,16HSTIFFNESS MATRIX/)  
 1110 FORMAT (I5,3E15.6)  
 1120 FORMAT (1I,9X,7HELEMENT,I4,5X,14HSTORAGE MATRIX/)  
 1130 FORMAT (1H0,10X,7HELEMENT,I4,5X,38HINSUFFICIENT HALF-BAND WIDTH -  
 1REQUIRE,I5,2X,10HINSTEAD OF,I5)  
 1140 FORMAT (1H,10X,35HA COEFFICIENT MATRIX HALFBAND WIDTH,I6)  
 1150 FORMAT (1H1,10X,37HA COEFFICIENT MATRIX - UPPER HALFBAND/11X,37(1H  
 1-)//)  
 1160 FORMAT (I5,10E12.4)  
 1170 FORMAT (//)  
 1180 FORMAT (1H1,10X,37HP COEFFICIENT MATRIX - UPPER HALFBAND/11X,37(1H  
 1-)//)  
 1190 FORMAT (////,11X,20HP COEFFICIENT MATRIX/11X,20(1H-)//)  
 1200 FORMAT (11X,10E12.3)  
 1210 FORMAT (1H0,10X,17HCOMPUTED HEAD /11X,13(1H-)//11X,12HELAPSED  
 1TIME 1PE13.2,6H HOURS/23X,1PE13.2,8H MINUTES/23X,1PE13.2,8H SECOND  
 2S//11X,9HTIME STEP I16,//11X,4HDELT 1PE21.2// 11X,6(4HNODE,5X,5HVA  
 3LUE,5X))  
 1240 FORMAT (////11X,10(1H\*),41HEXECUTION TERMINATED ON TIME-ELAPSED T  
 1IME1PE12.4,6H HOURS10(1H\*))  
 1250 FORMAT (////11X,10(1H\*),42HEXECUTION TERMINATED ON TIME STEPS AT S  
 1TEPI10,10(1H\*))  
 1260 FORMAT (///11X,15HKOD1 PARAMETERS/11X,15(1H-)/)  
 1281 FORMAT (11X,27HKOD1-MASS BALANCE REQUESTED)  
 1301 FORMAT (11X,44HKOD2-DO NOT PRINT ELEMENT STIFFNESS MATRICES)  
 1311 FORMAT (11X,37HKOD2-PRINT ELEMENT STIFFNESS MATRICES)  
 1331 FORMAT (11X,43HKOD3-DO NOT PRINT GLOBAL STIFFNESS MATRICES)  
 1341 FORMAT (11X,36HKOD3-PRINT GLOBAL STIFFNESS MATRICES)  
 1361 FORMAT (11X,29HKOD4-DO NOT PUNCH HEAD VALUES)  
 1371 FORMAT (11X,22HKOD4-PUNCH HEAD VALUES)  
 1391 FORMAT (11X,31HKOD5-DO NOT ASSUME STEADY STATE)  
 1401 FORMAT (11X,24HKOD5-ASSUME STEADY STATE)  
 1421 FORMAT (11X,34HKOD6-DO NOT PLOT COMPUTED DRAWDOWN)  
 1431 FORMAT (11X,27HKOD6-PLOT COMPUTED DRAWDOWN)  
 1451 FORMAT (11X,30HKOD7-DO NOT PLOT COMPUTED HEAD)  
 1461 FORMAT (11X,23HKOD7-PLOT COMPUTED HEAD)  
 1481 FORMAT (11X,30HKOD8-DO NOT PRINT KNOWN VECTOR)  
 1491 FORMAT (11X,23HKOD8-PRINT KNOWN VECTOR)

```

1511 FORMAT (11X,37HKOD9-DO NOT READ NODAL TRANSMISSIVITY)
1512 FORMAT (11X,23HKOD1-NOT WATERTABLE PROBLEM)
1521 FORMAT (11X,30HKOD9-READ NODAL TRANSMISSIVITY)
1522 FORMAT (11X,24HKOD1-WATERTABLE PROBLEM)
7005 FORMAT (//11X,39HHYDRAULIC CONDUCTIVITY OF CONFINING BED ,E15.4 /
111X,33HSPECIFIC STORAGE OF CONFINING BED, E15.4)
RETURN
END
SUBROUTINE SHAPE (JNRL,JNND,JBWH,JNDP,S,P,IX,J,X,Y,DGX,DGY,P)
DIMENSION IN(13,JNRL),S(JNDP,JBWH),P(JNND),X(JNND),Y(JNND),U(JNND)
1,ALP(4),DAX(4),DAY(4),BTX(4),BTY(4),DBX(4),DPX(12),DPY(12),D
2GX(12),DGY(12),P(12),DBY(4)
COMMON /WEIGHT/ SUM1,SUM2,SUM3,SUM4
INTEGER*2 IN
RETURN
ENTRY      SHAPE (L,N,XI,YI,DET)
C
XI1=1.-XI
XI2=1.+XI
YI1=1.-YI
YI2=1.+YI
C
C
CORNER NODE SHAPE FUNCTIONS, BASIC PART
ALP(1) =.25*XI1*YI1
ALP(2) =.25*XI2*YI1
ALP(3) =.25*XI2*YI2
ALP(4) =.25*XI1*YI2
DAX(1) =-.25*YI1
DAX(2) =.25*YI1
DAX(3) =.25*YI2
DAX(4) =-.25*YI2
DAY(1) =-.25*XI1
DAY(2) =-.25*XI2
DAY(3) =.25*XI2
DAY(4) =.25*XI1
C
C
CORNER NODE SHAPE FUNCTIONS, SIDE-DEPENDENT PART
IQ1=XI-.5
IQ2=-XI-.5
YQ1=YI-.5
YQ2=-YI-.5
XC1=1.125*XI*XI-.625
XC2=2.25*XI
YC1=1.125*YI*YI-.625
YC2=2.25*YI
J1=1
J2=2
J3=5

```

```
DO 50 J=1,2
IF (IX(J3,L).EQ.0) GO TO 10
IF (IX(J3+1,L).EQ.0) GO TO 20
GO TO 30
10 CONTINUE
BTX(J1) =.5
BTX(J2) =.5
DBX(J1) =0.
DBX(J2) =0.
GO TO 40
20 CONTINUE
BTX(J1) =XQ2
BTX(J2) =XQ1
DBX(J1) =-1.
DBX(J2) =1.
GO TO 40
30 CONTINUE
BTX(J1) =XC1
BTX(J2) =XC1
DBX(J1) =XC2
DBX(J2) =XC2
40 CONTINUE
J1=4
J2=3
J3=9
50 CONTINUE
J1=2
J2=3
J3=7
DO 100 J=1,2
IF (IX(J3,L).EQ.0) GO TO 60
IF (IX(J3+1,L).EQ.0) GO TO 70
GO TO 80
60 CONTINUE
BTY(J1) =.5
BTY(J2) =.5
DBY(J1) =0.
DBY(J2) =0.
GO TO 30
70 CONTINUE
BTY(J1) =YQ2
BTY(J2) =YQ1
DBY(J1) =-1.
DBY(J2) =1.
GO TO 90
80 CONTINUE
BTY(J1) =YC1
BTY(J2) =YC1
```





```

K=0
DO 260 I=1,N
250 K=K+1
   IF (IJ(K,L).EQ.0) GO TO 250
   KI=IN(K,L)
   SUM1=SUJ1+DPX(I)*X(KI)
   SUM2=SUJ2+DPX(I)*Y(KI)
   SUM3=SUJ3+DPY(I)*X(KI)
   SUM4=SUJ4+DPY(I)*Y(KI)
260 CONTINUE
   DET=SUJ1*SUM4-SUM2*SUM3
   DET1=1./DET
   C11=DET1*SUM4
   C12=-DET1*SUM2
   C21=-DET1*SUM3
   C22=DET1*SUM1
C
C  SHAPE FUNCTION DERIVATIVES - GLOBAL
DO 270 J=1,N
DGX(J)=C11*DPX(J)+C12*DPY(J)
DGY(J)=C21*DPX(J)+C22*DPY(J)
270 CONTINUE
RETURN
END
SUBROUTINE DBANDI (JNEL,JNND,JBWH,JNDP,S,P,IN,I,X,Y)
DIMENSION IN(13,JNEL),S(JNDP,JBWH),P(JNND),X(JNND),Y(JNND),U(JNND)
INTEGER*2 IN
RETURN
ENTRY      DBAND (N,NB,IBX)
IBX=0
DO 50 I=1,N
IP=N-I+1
IF (NB.LT.IP) IP=NB
DO 50 J=1,IP
IQ=NB-J
IF ((I-1).LT.IQ) IQ=I-1
SUM=S(I,J)
IF (IQ.LT.1) GO TO 20
DO 10 K=1,IQ
II=I-K
JZ=J+K
10 SUM=SUM+3(II,K+1)*S(II,JZ)
20 IF (J.LE.1) GO TO 40
IF (SUJ.LE.0.) GO TO 30
TEMP=1./SQRT(SUM)
S(I,J)=TEMP
GO TO 50
30 WRITE (6,60) I

```

```

WRITE (6,70) N,NB,IP,IQ,I,J,SUM
IEX=1
STOP
40 S(I,J)=SUM*TEMP
50 CONTINUE
RETURN

C
60 FORMAT (1H1,10X,19HDBAND FAILS AT ROW ,I4)
70 FORMAT (1H0,6I5,E20.8)
END
SUBROUTINE SBANDI (JNEL,JNND,JBWH,JNDF,S,P,IN,U,X,Y)
DIMENSION IN(13,JNEL),S(JNDF,JBWH),P(JNND),X(JNND),Y(JNND),U(JNND)
INTEGER*2 IN
RETURN
ENTRY SBAND (N,NB)
DO 30 I=1,N
J=I-NB+1
IF ((I+1).LE.NB) J=1
SUM=P(I)
K1=I-1
IF (J.GT.K1) GO TO 20
DO 10 K=J,K1
II=I-K+1
10 SUM=SUM-S(K,II)*U(K)
20 U(I)=SUM*S(I,1)
30 CONTINUE
DO 60 I1=1,N
I=N-I1+1
J=I-NB-1
IF (J.LE.N) J=N
SUM=U(I)
K2=I+1
IF (K2.GT.J) GO TO 50
DO 40 K=K2,J
KK=K-I+1
40 SUM=SUM+S(I,KK)*U(K)
50 U(I)=SUM*S(I,1)
60 CONTINUE
RETURN
END
SUBROUTINE PLOT (JNND,X,Y,CC,INDEX)
-----
C
C ---SPECIFICATIONS---
DIMENSION X(1,JNND),Y(1,JNND),CC(JNND),INDEX(JNND)
COMMON /AREA1/ XLABEL(3,2),YLABEL(3,3),TITLE(3,5),
1NP ,CMAX, IMAX,XMIN,YMAX,YMIN,NXS,NINX,NYS,NINY,NDS, K(10)
REAL*8 XLABEL,YLABEL,TITLE,Z,XN1
REAL P&NT(122),SYM(17),NX(50),NY(14)

```

```

INTEG23 DIGIT(82),VP1(6),VP2(6),VP3(7)
INTEG23      N(10),BLANK(60)
DATA SYD/'1','2','3','4','5','6','7','8','9','10',' ',' ','Y','*',
1'1','-','+','/','PRNT/122*' '/,N1,N2,N3,XN1/6,10,133,.0833333333D0/,
1BLANK/60*' '/
DATA DIGIT/'1','2','3','4','5','6','7','8','9','10','11','12','13'
1,'14','15','16','17','18','19','20','21','22','23','24','25','26'
1,2,'27','28','29','30','31','32','33','34','35','36','37','38','39',
1,2'40',      '41','82','83','84','85','86','87','88','89','90','91',
1'92','93','94','95','96','97','98','99','100','101','102','103',
1,2104','105','106','107','108','109','110','111','112','113','114',
1,3115','116','117','118','119','120','121','122'/'
DATA VP1/'(1H ',' ',' ',' ',' ','A1,P','10.2',') '/
DATA VP2/'(1H ',' ',' ',' ',' ','A1,P','X,18',') '/
DATA VP3/'(1H0',' ',' ',' ',' ','A1,P','3.0',' ','12P1','0.2) '/

```

C  
C

---INITIALIZE VARIABLES---

```

NXD=NXS*NINX
NYD=YS3*NINY
XSP=(X1AX-XMEN)/NXD
YSP=(Y1AY-YMEN)/NYD
N4=NXD*N1+1
N5=NYD+1
N6=NYD+1
N7=N1*NINX
N8=N2*NID+1
N9=N2*NINY
NR=N3-1
NA=N4/2-2
NB=N4/2+4
NC=(N3-N3-10)/2
ND=NC+N3
NE=MAX(N5,N6)
VP1(3)=DIGIT(ND-40)
VP2(3)=DIGIT(ND-40)
VP3(3)=DIGIT(NC)

```

C  
C

---ARRANGE EACH DATA SET IN DESCENDING VALUES OF X---

```

DO 90 L=1,NDS
  NN=K(L)
  DO 30 I=1,NN
    BIG=X(L,I)
    KK=I
    DO 20 J=1,NN
      IF(X(L,J).GT.BIG) GO TO 15
    GO TO 20
15  BIG=X(L,J)
    KK=J

```



```

      IP ((J-1)/N2*N2.EQ.J-1) PRNT(J)=SYM(14)
120 IP ((J-1)/N2*N2.NE.J-1) PRNT(J)=SYM(16)
C
C      ---COMPUTE LOCATION OF POINTS---
130 DO 150 J=1,NDS
135 IP (N(J).EQ.K(J)+1) GO TO 150
      IP (I.GT.1) GO TO 137
      IP (X(J,N(J)).LE.Z+XN1*YSP) GO TO 137
      N(J)=N(J)+1
      GO TO 135
137 IP (X(J,N(J)).LE.Z+XN1*YSP.AND.X(J,N(J)).GE.Z-XN1*YSP) GO TO 140
      GO TO 150
140 M=NR+0.5- ((Y(J,N(J))-YMIN)*N2)/YSP
      IP (M.LT.0.OR.N.GT.NR) GO TO 145
      IP (CC(N(J)) 1+2,146,147)
142 IP (M.NE.0) PRNT(M)=SYM(16)
      NUM=(-CC(N(J))+.005)*10.
      GO TO 141
147 NUM=(CC(N(J))+0.005)*100.
      IP (NUM.GT.999) NUM=MCD(NUM,1000)
141 IP (NUM.LT.100) GO TO 143
      INDX3=NUM/100
      IP (M.NE.0.AND.CC(N(J)).GT.0.) PRNT(M)=SYM(INDX3)
      NUM=NUM-INDX3*100
143 INDX1=MCD(NUM,10)
      IP (INDX1.EQ.0) INDX1=10
      INDX2=NUM/10
      IP (INDX2.EQ.0) INDX2=10
      GO TO 144
146 INDX1=14
      INDX2=14
144 PRNT(J+1)=SYM(INDX2)
      PRNT(J+2)=SYM(INDX1)
145 N(J)=N(J)+1
      IP (N(J).EQ.K(J)+1) GO TO 150
      IP (X(J,N(J)).LE.Z+XN1*YSP.AND.X(J,N(J)).GE.Z-XN1*YSP) GO TO 140
150 CONTINUE

```

```

C
C      ---PRINT AXES, LABELS, AND POINTS---
      IP (I-NA.EQ.0) GO TO 170
      IP (I-NB.EQ.0) GO TO 180
      IP ((I-1)/N1*N1-(I-1)) 190,160,190
160 WRITE (5,VP1) (BLANK(J),J=1,NC), (PRNT(J),J=1,N8), NY(1+(I-1)/6)
      GO TO 200
170 WRITE (5,VP2) (BLANK(J),J=1,NC), (PRNT(J),J=1,N8), XLABEL(NP,1)
      GO TO 200
180 WRITE (5,VP2) (BLANK(J),J=1,NC), (PRNT(J),J=1,N8), XLABEL(NP,2)
      GO TO 200

```

```

190 WRITE (6,VP2) (BLANK(J),J=1,NC), (PRNT(J),J=1,N8)
C
C   ---COMPUTE NEW VALUE FOR Z AND INITIALIZE PRNT---
200 Z=Z-2.*XN1*XS
    DO 210 J=1,N8
210 PRNT(J)=SYM(11)
C
C   ---NUMBER AND LABEL Y AXIS AND PRINT TITLE---
    WRITE (6,VP3) (BLANK(J),J=1,NC), (NY(I),I=1,N6)
    WRITE (6,70) (YLABEL(NP,I),I=1,3)
    WRITE (6,80) (TITLE(NP,I),I=1,5)
C
    RETURN
C
C   ---FORMATS---
40 FORMAT ('1')
70 FORMAT ('0',49X,3A8)
80 FORMAT ('0',41X,5A8)
98 FORMAT(' ',2G20.10)
97 FORMAT('0DATA SET NO.',I4, '//10X','Y',19X,'Y')
    END
    SUBROUTINE CLAY (QCOEF,THKL,SSEC,HYCOND,SS)
C
C   PURPOSE: THIS PROGRAM COMPUTES THE TRANSIENT LEAKAGE COEFFICIENT,
C   QCOEF.
C
C   DINT-----DIMENSIONLESS TIME
C   SS-----SPECIFIC STORAGE
C   HYCOND----HYDRAULIC CONDUCTIVITY
C   THK-----THICKNESS OF CONFINING LAYER
C
    DATA PIE/3.141593/
    TT=0.
    SUMN=J.0
    IF(SS.NE.0.) GO TO 5
    DENOM=1.0
    GO TO 31
5 DINT=HYCOND*SSEC/(THKL*THKL*SS*2.)
    IF (DINT.GT.TT) TT=DINT
    PPT = PIE*PIE*DINT
    IF (DINT.LT.1.0E-03) PPT = 1./DINT
    CK = (2.3-PPT)/(2.*PPT)
    DO 20 LL=1,200
    POWER = LL*LL*PPT
    IF (POWER.LE.174.) GO TO 10
    POWER = 150.

```

```
10 PEX = EXP(-POWER)
    SUMN = SUMN + PEX
    IF (PEX.GT.0.00009) GO TO 20
    IF (LL.GT.CK) GO TO 30
20 CONTINUE
30 DENOM = 1.0
    IF (DINT.LT.1.0E-03) DENOM = SQRT (PIE*DINT)
31 Q1 = HYCOND / (THKL * DENOM)
    QCORP = Q1 + 2.0 * Q1 * SUMN
    RETURN
END
```

3  
3  
3  
3  
3  
3  
3  
3  
3  
3

**PART 2**

**COMBINED GROUNDWATER FLOW AND MASS TRANSPORT**

## 1. INTRODUCTION

### 1.1 Purpose and Scope

In this second section of the manual we will extend the concepts presented in part 1 to include the movement of conservative solutes. To simplify the presentation it is assumed that the contaminant does not significantly change the density of the fluid and that a two dimensional areal representation, as indicated in part 1, is appropriate. Because the groundwater flow section of the transport code is virtually the same as that presented earlier let us now examine the species transport equation.

## 2. THEORETICAL DEVELOPMENT

### 2.1 Governing Equation

Areal, two dimensional transport of a conservative solute through a groundwater reservoir can be written

$$\begin{aligned} L(c) = & \frac{\partial}{\partial x} (D_{xx} \frac{\partial c}{\partial x}) + \frac{\partial}{\partial x} (D_{xy} \frac{\partial c}{\partial y}) + \frac{\partial q}{\partial y} (D_{yy} \frac{\partial c}{\partial y}) + \frac{\partial q'}{\partial y} (D_{yx} \frac{\partial c}{\partial x}) \\ & - \frac{\partial}{\partial x} (cq_x) - \frac{\partial}{\partial y} (cq_y) - \frac{\partial}{\partial t} (\theta bc) + Qc' + \frac{K'}{l'} (h-h_w)c'' \end{aligned} \quad (2-1)$$

where  $b$  is the saturated thickness of the aquifer [L],

$c$  is the concentration [ $ML^{-3}$ ],

$c'$  is the concentration of the discharging (recharging) fluid [ $ML^{-3}$ ],

$c''$  is the concentration of the fluid discharging (recharging) the aquifer through leakage [ $ML^{-3}$ ]

$D$  is the dispersion coefficient [ $L^2T^{-1}$ ],

$K'$  is the hydraulic conductivity of the confining bed [ $LT^{-1}$ ],

$l'$  is the thickness of the confining bed [L]

$q$  is the mass average flux vector [ $L^2T^{-1}$ ],

$Q$  is the rate of fluid withdrawal [ $LT^{-1}$ ],

$\theta$  is the porosity.

It is apparent from the units of  $D$  and  $q$  that they represent vertically integrated parameters, i.e.,

$$D_{xx}(x,y) = \int_b D'_{xx}(x,y,z) dz \quad (2-2)$$

$$q_x(x,y) = \int_b q'_x(x,y,z) dz \quad (2-3)$$

where  $D'$  and  $q'$  are point values.

In the most general case time derivatives of head would appear in the transport equation and concentration time derivatives in the flow equation. These terms are generally small, however, and have been neglected in this development.

## 2.2 Galerkin Formulation

The approximating integral equations required for the finite-element formulation are obtained for the flow and transport equations using Galerkin's method. Following a development analogous to that presented in part I we approximate the unknown concentration and certain of the spatially dependent parameters in terms of the basis functions  $w_j(x,y)$ :

$$c = \hat{c} = \sum_{j=1}^N c_j w_j(x,y) \quad (2-4)$$

$$D_{xx} = \hat{D}_{xx} = \sum_{j=1}^N D_{xxj} w_j(x,y) \quad (\text{same with } D_{xy}, D_{yx}, D_{yy}) \quad (2-5)$$

$$q_x = \hat{q}_x = \sum_{j=1}^N q_{xj} w_j(x,y) \quad (2-6)$$

The flux  $q_x$  is obtained through differentiation of the head solution as follows

$$\hat{q}_x = \sum_{j=1}^N T_{xx} H_j \frac{\partial w_j}{\partial x}(x,y) \quad (2-7)$$

where, in this code, the transmissivity is assumed constant over an element. Note that the derivatives of the basis functions (and consequently the fluxes)

used in this analysis are discontinuous at element boundaries and care must be exercised in choosing elements so that continuity of mass is satisfactorily maintained.

The approximating integral equations are obtained from Galerkin's scheme by making the residual, generated by substituting (2-4) into (2-1), orthogonal to each of the  $N$  basis functions  $w_j$ :

$$\int_A L(\hat{c})w_i dA = 0, \quad i = 1, 2, \dots, N \quad (2-8)$$

This may be expanded and reformulated in matrix form as

$$[N][C] + [M] \left\{ \frac{dC}{dt} \right\} + \{F\} = 0 \quad (2-9)$$

where typical elements of  $N$ ,  $M$ , and  $F$  are:

$$N_{kl} = \int_A \hat{D}_{xx} \frac{\partial w_l}{\partial x} \frac{\partial w_k}{\partial x} + 2\hat{D}_{xy} \frac{\partial w_l}{\partial y} \frac{\partial w_k}{\partial x} + \hat{D}_{yy} \frac{\partial w_l}{\partial y} \frac{\partial w_k}{\partial y} + \hat{q}_x w_k \frac{\partial w_l}{\partial x} + \hat{q}_y w_k \frac{\partial w_l}{\partial y} + w_k w_l \frac{\partial \hat{q}_x}{\partial x} + w_k w_l \frac{\partial \hat{q}_y}{\partial y} dA \quad (2-10a)$$

$$M_{kl} = \int_A b \theta w_k w_l dA \quad (2-10b)$$

$$F_k = \int_A w_k Q c^i dA - \int_S (w_k \underline{D} \cdot \nabla \hat{c}) \cdot \underline{n} dS \quad (2-10c)$$

where  $\underline{n}$  is the unit normal vector. The time derivative is approximated using finite differences as in part 1.

In formulating the equations in this way we have assumed that the dispersion tensor is symmetric (i.e.,  $D_{xy} = D_{yx}$ ). This is indeed the case for the dispersion-velocity relationship we are using, namely,

$$D_{xx} = D_L \frac{q_x q_x}{q^2} + D_T \frac{q_y q_y}{q^2} + D_d \tau$$

$$D_{yy} = D_L \frac{q_y q_y}{q^2} + D_T \frac{q_x q_x}{q^2} + D_d \tau$$

$$D_{xy} = D_{yx} = (D_L - D_T) \frac{q_x q_y}{q^2}$$

$$D_L = \alpha_L |q| \quad , \quad D_T = \alpha_T |q|$$

where  $\alpha_L$  is the longitudinal dispersivity [L]

$\alpha_T$  is the transverse dispersivity [L]

$D_d$  is the molecular diffusion coefficient [ $L^2 T^{-1}$ ], and

$\tau$  is the tortuosity.

We have also assumed porosity time independent in the transport equation (this is a reasonable assumption for most field situations). While equation (2-9) can be solved as presented it is often modified to facilitate computer coding. The continuity of fluid flow is generally used to eliminate the spatial derivatives of the flux terms (terms six and seven in 2-10a). Occasionally it is advantageous to apply Green's theorem to terms five and six in equation (2-1). This approach will generate a natural mixed-type boundary condition instead of the second type indicated in the surface integral of equation (2-10c).

## PROGRAM DOCUMENTATION

### 3.1 Data Set 1: Identification

<u>Variable</u>	<u>Format</u>	<u>Variable Description</u>
CARD	20A4	Data set identification; punch ISOQUAD in columns 1-7.

### 3.2 Data Set 2: Title

TITLE	20A4	Title information; information typed on this card is reproduced exactly as a heading on printout.
-------	------	---

### 3.3 Data Set 3: Element Identification

NN	15	Number of nodes
NE	15	Number of elements
NS	15	Number of constant head nodes
NB	15	Global flow matrix halfband width*
NF	15	Number of source or sink nodes
NL	15	Number of leaky elements
KNS	15	Number of constant concentration nodes
KNB	15	Global concentration matrix halfbandwidth
NLH	15	Number of elements with vertical leakage.

### 3.4 Data Set 4: Time Identification

TIME	F10-0	Duration of simulation period in hours
DELT	F10-0	Initial time step $\Delta t$ in hours
CHNG	F10-0	Multiplier used to change size of time step according to relationship $\Delta t_i = \text{CHNG} * \Delta t_{i-1}$
ITMAX	110	Maximum permitted number of time steps
ITCHNG	110	Number of time steps between changes in $\Delta t$
TCHK	F10-0	Spatial criterion for time step selection $\Delta t = \text{TCHK} / (\text{maximum calculated velocity})$
TYPE	A4	Time derivative indicator; type FORW for backward difference (fully implicit) and CENT for centered difference (Crank-Nicolson)

#### NEW CARD

DIFUSN	F10-0	Molecular diffusion constant
NPRINT	110	Number of time steps between printouts
IGO	110	Number of concentration solutions between head calculations, $\Delta t(\text{head}) = \Delta t(\text{concentration}) * \text{IGO}$
UPVAL	F10-0	Upstream weighting coefficient (normally use 1.) <sup>+</sup>

### 3.5 Data Set 5: Constant Multipliers

AFMOBX	F10-0	Multiplier for X hydraulic conductivity, i.e., $\text{FMOBX}(\text{used}) = \text{FMOBX}(\text{read}) * \text{AFMOBX}$
AFMOBY	F10-0	Multiplier for Y mobility
AST	F10-0	Multiplier for storage coefficient
APOR	F10-0	Multiplier for porosity
AELONG	F10-0	Multiplier for longitudinal dispersivity
AETRAN	F10-0	Multiplier for transverse dispersivity
ATHICK	F10-0	Multiplier for aquifer thickness
APHII	F10-0	Multiplier for initial hydraulic head

#### NEW CARD

ACONCI	F10-0	Multiplier for initial concentration
FACTOR	F10-0	Multiplier for nodal co-ordinates
ALEAK	F10-0	Multiplier for leaky element coefficient (COEF)

\* Note that the halfband, as used herein, includes the diagonal.

<sup>+</sup>A value of zero is more accurate if no discontinuities exist in concentration field.

### 3.6 Data Set 6: Input-Output Codes\*

KOD1	15	Print element flow matrices (1=output; 0=suppress output)
KOD2	15	Print global flow matrix (1=output; 0=suppress output)
KOD3	15	Print element concentration matrices (1=output; 0=suppress output)
KOD4	15	Print global concentration matrix (1=output; 0=suppress output)
KOD7	15	Print point velocities (value of KOD7 is the number of time steps between each printout)
KOD8	15	Print vector of known values on the right-hand side of the matrix equation (1=output; 0=suppress output).
KOD9	15	Print hydraulic head solution (1=output; 0=suppress output)
KOD10	15	Print concentration solution (1=output; 0=suppress output)
KOD12	15	Number of time steps between punched output
KOD14	15	Punched output selector: 0=punch head and concentration -1=punch head values only 1=punch concentration values only

### 3.7 Data Set 7: Nodal Co-ordinates

J	15	Node number
X(J)	F10.0	X co-ordinate value for node J
Y(J)	F10.0	Y co-ordinate value for node J

### 3.8 Data Set 8: Source and Sink Nodes

IQ	15	Node number where a source (+) or sink (-) is located
FQ(IQ)	F10.0	Discharge (or recharge) at node IQ (volume per unit time)

### 3.9 Data Set 9: Concentration for Sources and Sinks

IQ	15	Nodal number of source or sink node
CFQ(IQ)	F10.0	Concentration of recharging fluid (positive number). A negative value indicates a discharging node with current reservoir concentration.

### 3.10 Data Set 10: Initial Head Values

PHI1(I)	5G15.7	Initial hydraulic head values at each nodal location
---------	--------	--

\*Printout of global or element coefficient matrices is voluminous.

3.11 Data Set 11: Initial Concentration Values

CONCI(I)            5G15-7            Initial concentration values at each nodal location

3.12 Data Set 12: Element Incidences

L                    13                    Element Number  
 CHAR                free format          Corner and side nodes located on element L

The node numbers are recorded beginning at a corner node and moving counterclockwise around the perimeter of the element. Each corner node is followed immediately by a dollar sign and the first node is repeated. One or more blanks must appear between each nodal number or nodal number plus dollar sign. As an example the element appearing in Figure 1 would be described as 1\$ 4 7\$ 6\$ 5 3 2\$ 1\$. Thus the order of each element side is completely defined by the element incidence.

3.13 Data Set 13: Element Parameters

L                    13                    Element number  
 FMOBX(L)            F10-3                Hydraulic conductivity of element L (principal component along x, i.e.,  $K_{xx}$ )  
 FMOBY(L)            F10-3                Hydraulic conductivity of element L (principal component along y, i.e.,  $K_{yy}$ )  
 ST(L)                F10-3                Storage coefficient of element L  
 POR(L)               F10-3                Porosity of element L  
 ELONG(L)            F10-3                Longitudinal dispersivity of element L  
 ETRANS(L)           F10-3                Transverse dispersivity of element L  
 THICK(L)            F10-3                Aquifer thickness in element L

3.14 Data Set 14: Leaky Element Coefficients

I                    15                    Leaky element number  
 COEF(I)            F10.0                Hydraulic conductivity of the confining bed divided by its thickness for element I  
 CCOEF(I)            F10.0                Concentration of leakage entering the aquifer from the confining bed. When leakage is out of the aquifer a negative value should be specified and the current aquifer concentration will be used.

If there are no leaky elements (NL=0) this data set is omitted.

3.15 Data Set 15: Dirichlet Nodes for Flow

LRT(ITT)            20(I4)                The number of those nodes where constant head values are specified.

*Data set 14b leaky element Node N  
 12520 3.10.0 Constant*

3.16 Data Set 16: Dirichlet Nodes for Concentration

LRT(ITT)\*            20(I4)                The number of those nodes where constant concentration conditions are specified.

3.17 Data Set 17: End of Data Identification

CARD                (20A4)                End of data is indicated by punching XXXX on this card which appears at the end of the last data set. If more than one problem is being run this card does not appear between problems.

\*The use of LRT in data sets 15 and 16 is simply a computational convenience.

C		I S O Q U A D	00001
C		C O M B I N E D F L O W A N D M A S S T R A N S P O R T	00002
C			00003
C		PROGRAMMED BY GEORGE F. PINDER, DEPT. OF CIVIL ENGINEERING,	00004
C		PRINCETON UNIVERSITY, PRINCETON N. J. 08540	00005
C			00006
C		D E F I N I T I O N O F V A R I A B L E S	00007
C		ACCNCI-----MULTIPLIER FOR INITIAL CONCENTRATION	00008
C		AEICNG-----MULTIPLIER FOR ELONG, LONGITUDINAL DISPERSIVITY	00009
C		AETRAN-----MULTIPLIER FOR STRANS, TRANSVERSE DISPERSIVITY	00010
C		AFPCBX-----MULTIPLIER FOR MOBXY, MOBILITY IN X-DIRECTION	00011
C		AFPCBY-----MULTIPLIER FOR MOBY, MOBILITY IN Y-DIRECTION	00012
C		AG(I)-----CONSTANTS FOR NUMERICAL INTEGRATION	00013
C		AFHII-----MULTIPLIER FOR INITIAL HEAD	00014
C		APCF-----MULTIPLIER FOR POR, POROSITY	00015
C		AST-----MULTIPLIER FOR S, STORAGE COEFFICIENT	00016
C		ATHICK-----MULTIPLIER FOR THICK, THICKNESS OF AQUIFER	00017
C		CAVC-----AVERAGE CONCENTRATION OVER AN ELEMENT	00018
C		CCCF(L)----CONCENTRATION OF LEAKAGE	00019
C		CPC(I)-----CONCENTRATION OF SOURCE OR SINK - IF NEGATIVE,	00020
C		CURRENT CONCENTRATION OF THE NODE IS ASSUMED	00021
C		CHAF(I)-----ALPHAMERIC CHARACTER STRING FOR INCIDENCES	00022
C		CHNG-----MULTIPLIER FOR INCREASING TIME STEP (DELT=CHNG*DELT)	00023
C		CH(I,J)-----TIME INDEPENDENT GLOBAL CONC. MATRIX	00024
C		COEF-----HYDRAULIC CONDUCTIVITY OF CONFINING BED DIVIDED BY ITS	00025
C		THICKNESS	00026
C		COLI-----STORAGE VECTOR FOR CONCENTRATION	00027
C		CONC(I)-----CONCENTRATION (M/L**3)	00028
C		COB(I)-----LOCATION OF PARTICULAR SOLUTIONS	00029
C		CP(I,J)-----TIME DEPENDENT GLOBAL CONC. MATRIX	00030
C		DELT-----TIME STEP (T)	00031
C		DELIGO-----TIME STEP FOR PRESSURE - DELTGO=DELT*IGO	00032
C		DETIJ(I)-----DETERMINANT OF J AT EACH INTEGRATION POINT	00033
C		DEX(I)-----TEMPORARY STORAGE FOR THE POINT X VELOCITY	00034
C		DEY(I)-----TEMPORARY STORAGE FOR THE POINT Y VELOCITY	00035
C		DIFUSN-----MOLECULAR DIFFUSION CONSTANT	00036
C		DGX(I)-----DF/DX FOR NODE BEING PROCESSED	00037
C		DGY(I)-----DF/DY FOR NODE BEING PROCESSED	00038
C		DIGIT(I)-----ALPHAMERIC CHARACTER STRING FOR INCIDENCES	00039
C		DPX(I)-----TEMPORARY STORAGE FOR PRESSURE DERIVATIVE	00040
C		DPI(I)-----TEMPORARY STORAGE FOR PRESSURE DERIVATIVE	00041
C		DPEX(I)-----AVERAGE X-DERIVATIVE OF PRESSURE FOR ELEMENT L	00042
C		DPEY(I)-----AVERAGE Y-DERIVATIVE OF PRESSURE FOR ELEMENT L	00043
C		DX(I,J)-----DF/DX FOR I NODES OF ELEMENT L	00044
C		DY(I,J)-----DF/DY FOR I NODES OF ELEMENT L	00045
C		ELGEG-----LONGITUDINAL DISPERSIVITY	00046
C		ETRANS-----TRANSVERSE DISPERSIVITY	00047
C		F(I,J)-----BASIS FUNCTION VALUES FOR ALL NODES IN ELEMENT	00048
C		FACTOR-----MULTIPLIER FOR CO-ORDINATES. VALUE = VALUE(READ)*FACTO	00049
C		FE(I)-----BASIS FUNCTION AT A POINT FOR ALL NODES IN ELEMENT	00050
C		FN(I)-----VECTOR ON RHS OF MATRIX EQUATION	00051
C		PHOEX(I)-----X-COMPONENT OF MOBILITY TENSOR, K/MU (L**3T/M)	00052
C		PHOXY(I)-----Y-COMPONENT OF MOBILITY TENSOR, K/MU (L**3T/M)	00053
C		FQ(I)-----RECHARGE TO AQUIFER (POS) (L**3/T)	00054
C		H(I,J)-----TIME DEPENDENT PART OF GLOBAL FLOW MATRIX	00055
C		HZEFO(I)-----HEAD IN ADJACENT AQUIFER I	00056
C		IN(I,J)-----INCIDENCES	00057
C		IT-----NUMBER OF TIME STEPS COMPLETED	00058
C		ITCENG-----NUMBER OF TIME STEPS BETWEEN CHANGES IN DELT	00059

C	IGC-----	NUMBER OF CONCENTRATION CALCULATIONS PER PRESSURE	000600
C		CALCULATION	000610
C	ITMAX-----	MAXIMUM PERMITTED NUMBER OF TIME STEPS	00
C	IVTST-----	EQUALS 1 IF VELOCITY HAS BEEN COMPUTED, ZERO	00
C		OTHERWISE	000640
C	JD(I)-----	CONTAINS THE INCIDENCES OF THE ACTIVE NODE IN ELEMENT	000650
C	JTST1-----	EQUALS 0 IF CONC. MATRIX TO BE GENERATED,	000660
C		EQUALS 1 IF FLOW MATRIX TO BE GENERATED	000670
C		EQUALS 2 IF BOTH MATRICES TO BE GENERATED	000680
C	KKB(I)-----	ROW INDICES OF CONSTANT CONCENTRATION NODES	000690
C	KLB(I)-----	EQUALS ONE FOR CONSTANT CONCENTRATION NODE, ZERO	000700
C		OTHERWISE	000710
C	KLBC(I)-----	RUNNING SUM OF CONSTANT CONCENTRATION NODES	000720
C	KN-----	DEGREES OF FREEDOM FOR CONCENTRATION	000730
C	KNB-----	CS MATRIX HALFBAND WIDTH (ESTIMATED)	000740
C	KNBST-----	ORIGINAL ESTIMATE OF CS HALFBANDWIDTH	000750
C	KNB2-----	CS BANDWIDTH	000760
C	KNB2ST-----	ORIGINAL ESTIMATE OF CS BANDWIDTH	000770
C	KNS-----	NUMBER OF CONSTANT CONCENTRATION NODES	000780
C	KOE1-----	PRINT ELEMENT FLOW MATRICES (1=OUTPUT)	000790
C	KOE2-----	PRINT GLOBAL FLOW MATRIX (1=OUTPUT)	000800
C	KOE3-----	PRINT ELEMENT CONCENTRATION MATRICES (1=OUTPUT)	000810
C	KOE4-----	PRINT GLOBAL CONCENTRATION MATRIX (1=OUTPUT)	000820
C	KOE5-----	NOT CURRENTLY USED	000830
C	KOE6-----	NOT CURRENTLY USED	000840
C	KOE7-----	PRINT POINT VELOCITIES (VALUE OF KOD7 IS THE NUMBER	000850
C		OF TIME STEPS BETWEEN EACH CALCULATION)	000860
C	KOE8-----	PRINT VECTOR OF KNOWN VALUES (1=OUTPUT)	000870
C	KOE9-----	PRINT PRESSURE SOLN. (1=OUTPUT)	000880
C	KOE10-----	PRINT CONCENTRATION SOLUTION (1=OUTPUT)	000890
C	KOE11-----	NOT CURRENTLY USED	00
C	KOE12-----	NUMBER OF TIME STEPS BETWEEN PUNCHED OUTPUT	000910
C	KOE13-----	NOT CURRENTLY USED	000920
C	KOE14-----	TYPE OF PUNCHED OUTPUT	000930
C		0 INDICATES PRESSURES AND CONCENTRATION	000940
C		-1 INDICATES PRESSURES ONLY	000950
C		1 INDICATES CONCENTRATION ONLY	000960
C	KR(I)-----	ROW INDICES OF CONSTANT PRESSURE NODES	000970
C	I-----	ELEMENT NUMBER	000980
C	LR(I)-----	EQUALS ONE FOR CONSTANT PRESSURE NODE, ZERO OTHERWISE	000990
C	LBC(I)-----	RUNNING SUM OF CONSTANT PRESSURE NODES	001000
C	LBI(I)-----	VALUES OF CONSTANT PRESSURE NODES (NUMBER)	001010
C	N-----	NUMBER OF ACTIVE NODES IN ELEMENT	001020
C	N-----	NUMBER OF DEGREES OF FREEDOM FOR FLOW	001030
C	NB-----	S MATRIX HALFBAND WIDTH (ESTIMATED)	001040
C	NB1-----	MODIFIED HALFBAND WIDTH	001050
C	NE-----	NUMBER OF ELEMENTS	001060
C	NF-----	NUMBER OF SOURCE OR SINK NODES	001070
C	NL-----	NUMBER OF LEAKY ELEMENTS	001080
C	NH-----	NUMBER OF NODES	001090
C	NPRINT-----	NO. OF TIMES EQUATIONS ARE SOLVED BETWEEN PRINTOUTS	001100
C	NB-----	NUMBER OF DIRICHLET BOUNDARY NODES FOR FLOW	001110
C	NS-----	NUMBER OF CONSTANT PRESSURE NODES	001120
C	OLE(I)-----	STORAGE VECTOR FOR U	001130
C	P(I,J)-----	TIME DEPENDENT GLOBAL FLOW MATRIX	001140
C	PE(I,J)-----	TIME DEPENDENT ELEMENT FLOW MATRIX	001150
C	PHI(I)-----	PRESSURE (M/LT**2) OR HEAD IF AREAL PROBLEM (L)	001160
C	PHI AVG-----	AVERAGE HEAD OVER AN ELEMENT	00
C	PHI I(I)-----	INITIAL PRESSURE (M/LT**2) OR HEAD (L)	00
C	POF(I)-----	POROSITY	00



```

DIMENSION TITLE(20),CHAR(77),DIGIT(10),KR(40),LRT(50),00180
1KKB(4C)00181
DATA BLANK/1H/,STAR/1H3/,DIGIT/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H
18,1H9/,START/4HISQ/,EXIT/4HXXX/,KR(1)/0/,00184
2PWF/4HFORW/,CNTX/4HCENT/,YES/0./,NO/1/00185
-----00185
C00186
C00187
C.....READ AND WRITE DATA00187
1C READ (5,840) CARD00188
IF (CARD.EQ.EXIT) STOP00189
IF (CARD.NE.START) GO TO 1000190
READ (5,840) TITLE00191
WRITE (6,850)00192
WRITE (6,860) TITLE00193
C00194
REAL (5,910) NN,NE,NS,NB,NP,NL,KNS,KNB,NLH00195
WRITE (6,920)00196
WRITE (6,930) NN,NE,NS,NP,NB,KNB,NL,KNS,NLH00197
C00198
REAL (5,940) TIME,DELT,CHNG,ITMAX,ITCHNG,TCHK,TYPE,DIPUSN,00199
1NFRINT,IGC00200
WRITE (6,950) TIME,DELT,CHNG,ITMAX,ITCHNG,TCHK,DIPUSN,00201
1NFRINT,IGC00202
IF (TYPE.EQ.PWD) WRITE (6,960)00203
IF (TYPE.EQ.CNTR) WRITE (6,970)00204
C00205
REAL (5,1440) AFMOBX,AFMOBY,AST,APOR,ABELONG,AETRAN,ATHICK,00206
1AFPHI,ACCNCI,FACTOR,ALEAK00207
WRITE (6,1450) AFMOBX,AFMOBY,AST,APOR,ABELONG,AETRAN,ATHICK,00208
1AFPHI,ACCNCI,FACTOR,ALEAK00209
C00210
REAL (5,910) KOD1,KOD2,KOD3,KOD4,KOD7,KOD8,KOD9,KOD10,KOD12,KOD1400211
C00212
C.....WRITE OUT CODE INTERPRETATION00213
WRITE (6,25)00214
IF (KOD1 .EQ.0) WRITE (6,1)00215
IF (KOD1 .NE.0) WRITE (6,2)00216
IF (KOD2 .EQ.0) WRITE (6,3)00217
IF (KOD2 .NE.0) WRITE (6,4)00218
IF (KOD3 .EQ.0) WRITE (6,5)00219
IF (KOD3 .NE.0) WRITE (6,6)00220
IF (KOD4 .EQ.0) WRITE (6,7)00221
IF (KOD4 .NE.0) WRITE (6,8)00222
IF (KOD7 .NE.0) WRITE (6,9) KOD700223
IF (KOD8 .EQ.0) WRITE (6,12)00224
IF (KOD8 .NE.0) WRITE (6,13)00225
IF (KOD9 .EQ.0) WRITE (6,14)00226
IF (KOD9 .NE.0) WRITE (6,15)00227
IF (KOD10 .EQ.0) WRITE (6,16)00228
IF (KOD10 .NE.0) WRITE (6,17)00229
IF (KOD12 .NE.0) WRITE (6,18) KOD1200230
IF (KOD14 .EQ.0) WRITE (6,22)00231
IF (KOD14 .EQ.-1) WRITE (6,23)00232
IF (KOD14 .EQ.1) WRITE (6,24)00233
C00234
C.....NODE COORDINATES00235
REAL (5,980) (J,X(J),Y(J),K=1,NN)00236
EC 19 J=1,NN00237
X(J)=X(J)*FACTOR00238
19 Y(J)=Y(J)*FACTOR00239

```

WRITE (6,990)	002400
WRITE (6,1000)	002410
WRITE (6,1010) (J,X(J),Y(J),J=1,NN)	002420
C	002430
C.....SOURCE AND SINK NODES, FQ.	002440
C.....SOURCES ARE POSITIVE	002450
WRITE (6,1020)	002460
DC 20 I=1,NN	002470
20 FQ(I)=0.	002480
IF (NF.EQ.0) GO TO 40	002490
DC 30 I=1,NF	002500
READ (5,980) IQ,FQ(IQ)	002510
30 WRITE (6,1030) IQ,FQ(IQ)	002520
40 CONTINUE	002530
C	002540
C.....SOURCE AND SINK CONCENTRATIONS, CFQ	002550
WRITE (6,1040)	002560
DC 50 I=1,NN	002570
50 CFQ(I)=0.	002580
IF (NF.EQ.0) GO TO 70	002590
DC 60 I=1,NF	002600
READ (5,980) IQ,CFQ(IQ)	002610
60 WRITE (6,1030) IQ,CFQ(IQ)	002620
70 CONTINUE	002630
C	002640
C.....READ INITIAL PRESSURE OR HEAD VALUES	002650
READ (5,1051) (PHII(I),I=1,NN)	002660
DC 1470 I=1,NN	002670
1470 PHII(I)=PHII(I)*APHII	002680
WRITE (6,1070)	002690
90 WRITE (6,1080)	002700
WRITE (6,1090) (I,PHII(I),I=1,NN)	002710
C	002720
C.....READ INITIAL CONCENTRATION VALUES	002730
READ (5,1051) (CONCI(I),I=1,NN)	002740
DC 1460 I=1,NN	002750
1460 CONCI(I)=CONCI(I)*ACONCI	002760
WRITE (6,1100) (I,CONCI(I),I=1,NN)	002770
C	002780
C.....ELEMENT INCIDENCES	002790
C.....FREE FORMAT FOR INPUT	002800
DC 100 I=1,13	002810
DC 100 L=1,NE	002820
100 IN(I,L)=0	002830
DO 200 LL=1,NE	002840
READ (5,1110) L,CHAR	002850
KC=C	002860
KE=4	002870
KS=2	002880
J=0	002890
110 NUN=0	002900
120 J=J+1	002910
IF (CHAR(J).EQ.BLANK) GO TO 120	002920
C	002930
C.....DETERMINE INCIDENCE FROM CHARACTER LIST AND ADJUST FOR POWER OF	002940
C.....TEN.	002950
130 I=0	002960
140 I=I+1	002970
IF (I.LE.10) GO TO 150	002980
WRITE (6,1120) CHAR(J),L	002990

GO TO 10	003000
150 IF (CHAR(J).NE.DIGIT(I)) GO TO 140	003010
NUM=10*NUM+I-1	003020
J=J+1	003030
C	
C.....CHECK NEXT LOCATION-IF IT IS AN \$ THIS IS A CORNER NODE. IF IT IS	003040
BLANK THIS IS LAST DIGIT OF A SIDE NODE.	003060
IF (CHAR(J).EQ.STAR) GO TO 160	003070
IF (CHAR(J).EQ.BLANK) GO TO 170	003080
GO TO 130	003090
160 KC=KC+1	003100
C	003110
C.....WHEN KC, THE CORNER COUNTER, IS FIVE THE INCIDENCE LIST FOR	003120
ELEMENT L IS COMPLETE.	003130
C.....WHEN A NODE IS ABSENT FROM A POSSIBLE LOCATION,ZERO IS RECORDED IN	003140
THE IN ARRAY.	003150
IF (KC.EQ.5) GO TO 180	003160
IN(KC,L)=NUM	003170
IF (KS.EQ.0) KE=KE+2	003180
IF (KS.IQ.1) KE=KE+1	003190
KS=C	003200
GO TO 110	003210
170 KE=KE+1	003220
KS=KS+1	003230
IN(KE,L)=NUM	003240
IF (J.LT.65) GO TO 110	003250
WRITE (6,1130) L	003260
GC TO 10	003270
180 N=4	003280
C	003290
C.....COUNT ACTIVE NODES. IF THOSE NOT EQUAL TO ZERO.	003300
DC 19C I=5,12	
IF (IN(I,L).EQ.0) GO TO 190	
N=N+1	003330
190 CONTINUE	003340
IN(13,L)=N	003350
200 CONTINUE	003360
WRITE (6,1140)	003370
WRITE (6,1150)	003380
DO 210 I=1,NE	003390
210 WRITE (6,1160) L,(IN(I,L),I=1,12)	003400
C	003410
C.....REAL ELEMENT PARAMETERS	003420
WRITE (6,1170)	003430
DO 220 LL=1,NE	003440
REAL (5,1180) L,PNOBX(L),PNOBY(L),ST(L),POR(L),ELONG(L),ETRANS(L),	003450
1,THICK(L)	003460
PNCBX(L)=PNOBX(L)*ATHICK	003470
PNCBY(L)=PNOBY(L)*ATHICK	003480
ST(I)=ST(I)*AST	003490
POR(L)=POR(L)*APOR	003500
ELONG(L)=ELONG(L)*AELONG	003510
ETRANS(L)=ETRANS(L)*AETRAM	003520
THICK(L)=THICK(L)*ATHICK	003530
WRITE (6,1190) L,PNOBX(L),PNOBY(L),ST(L),POR(L),ELONG(L),ETRANS(L)	003540
1,THICK(L)	003550
C.....NOTE THAT PNOBX AND PNOBY ARE NOW REDEFINED AS TRANSMISSIVITY	003560
PNCBY(L)=PNOBY(L)*THICK(L)	003570
PNCBX(L)=PNOBX(L)*THICK(L)	003580
220 CONTINUE	003590

C		003600
C.....	LEAKY ELEMENT COEFFICIENTS	003610
	DO 230 I=1,NE	003620
	CCOEF(I)=0.	003630
230	CCEF(I)=0.	003640
	IF (NL.EQ.0) GO TO 250	003650
	DO 240 J=1,NL	003660
	READ(5,983) I,COEF(I),CCOEF(I)	003670
240	CCEF(I)=COEF(I)*ALEAK	003680
	WRITE(6,1200) (I,COEF(I),CCOEF(I),I=1,NE)	003690
	DO 241 I=1,NN	003700
241	HZERO(I)=C.	003710
	READ(5,242) (I,HZERO(I),J=1,NLH)	003720
242	FORMAT(5(I5,G10.0))	003730
	WRITE(6,243)	003740
243	FORMAT(///11X,32HCONSTANT HEAD FOR LEAKY ELEMENTS/11X,32(1H-))	003750
	WRITE(6,1080)	003760
	WRITE(6,1090) (I,HZERO(I),I=1,NN)	003770
250	CONTINUE	003780
C		003790
C.....	DIRICHLET BOUNDARY NODES FOR FLOW (INDICATED BY LR=1)	003800
	WRITE(6,1230)	003810
	DO 270 I=1,NN	003820
270	LR(I)=0.	003830
	IF (NS.EQ.0) GO TO 310	003840
	NST=0.	003850
280	READ(5,1240) (LRT(ITI),ITI=1,20)	003860
	IA=C	003870
	DO 290 I=1,20	003880
	IF (LRT(I).EQ.0) GO TO 300	003890
	IA=I	003900
	J=LRT(I)	003910
	NST=NST+1	003920
	IF (J.LE.NN) GO TO 290	003930
	WRITE(6,1250) J	003940
	GO TO 10	003950
290	LR(J)=1	003960
300	WRITE(6,1260) (LRT(IIT),IIT=1,IA)	003970
	IF (IA.EQ.20) GO TO 280	003980
	IF (NST.NE.NS) WRITE(6,1270) NST,NS	003990
C		004000
C.....	DIRICHLET NODES FOR CONCENTRATION	004010
310	WRITE(6,1280)	004020
	DO 320 I=1,NN	004030
320	KLF(I)=0	004040
	IF (KNS.EQ.0) GO TO 360	004050
	NST=0	004060
330	READ(5,1240) (LRT(ITI),ITI=1,20)	004070
	IA=C	004080
	DO 340 I=1,20	004090
	IF (LRT(I).EQ.0) GO TO 350	004100
	IA=I	004110
	J=LRT(I)	004120
	NST=NST+1	004130
	IF (J.LE.NN) GO TO 340	004140
	WRITE(6,1290) J	004150
	GO TO 10	004160
340	KLF(J)=1	004170
350	WRITE(6,1260) (LRT(IIT),IIT=1,IA)	004180
	IF (IA.EQ.20) GO TO 330	004190

```

      IF (NST.NE.KNS) WRITE (6,1300) NST,KNS
360  CONTINUE
C.....END DATA INPUT
C
C.....INITIALIZATION
C.....SUN DIRICHLET BOUNDARY MODES
      IRC(1)=LR(1)
      DC 370 J=2,NN
370  IRC(J)=LRC(J-1)+LR(J)
      NR=IRC(NN)
      N=NN-NR
      KIRC(1)=KLR(1)
      DC 380 J=2,NN
380  KIRC(J)=KIRC(J-1)+KLR(J)
      KNR=KIRC(NN)
      KN=NN-KNR
      WRITE (6,1310) NN,NR,N,KNR,KN
C
C.....CONSTANT BOUNDARY MATRIX ROW INDICES
      K=0
      DO 390 J=1,NN
      IF (LR(J).LT.1) GO TO 390
      K=K+1
      KR(K)=J
390  CONTINUE
      K=0
      DO 400 J=1,NN
      IF (KIR(J).LT.1) GO TO 400
      K=K+1
      KR(K)=J
400  CONTINUE
C
C.....CONSTANTS
      ADVANC=YES
      STIME=0.
      RTII=0
      IT=C
      DELT=CELT*3600.
      SSIC=0.
      KNB1=0
      NNB1=0
      KNB2ST=KNB
      KNB2=2*KNB-1
      KNB2ST=KNB2
      L=0
C
C.....CLEAR ARRAYS
      DO 410 I=1,N
      DO 410 J=1,NB
      F(I,J)=0.
410  H(I,J)=0.
      DO 420 I=1,KN
      DC 420 J=1,KNB2
      CF(I,J)=0.
420  CH(I,J)=0.
C
      II=C
      RTII=0
      DC 470 I=1,NN
      CPH(I)=CONCI(I)

```

```

004200
004210
0
0
004240
004250
004260
004270
004280
004290
004300
004310
004320
004330
004340
004350
004360
004370
004380
004390
004400
004410
004420
004430
004440
004450
004460
004470
004480
004490
0
0
004520
004530
004540
004550
004560
004570
004580
004590
004600
004610
004620
004630
004640
004650
004660
004670
004680
004690
004700
004710
004720
004730
004740
004750
0047
004
( 10
0 90

```

PH(I)=PHII(I)	00480C
IF (LR(I).GT.0) GO TO 430	00481C
II=II+1	00482C
PHI(II)=PHII(I)	00483C
OLD(II)=PHI(II)	00484C
430 IF (KIR(I).GT.0) GO TO 470	00485C
KII=KII+1	00486C
IF (CFQ(I)) 450,440,440	00487C
440 CRT(KII)=-FQ(I)*CFQ(I)	00488C
GO TO 460	00489C
450 CRT(KII)=0	00490C
460 CCNC(KII)=CONCI(I)	00491C
CCLI(KII)=CONC(KII)	00492C
470 CCNTINUE	00493C
C	00494C
C.....END DATA INITIALIZATION	00495C
C	00496C
C	00497C
C T R A N S I E N T S O L U T I O N	00498C
C.....GENERATE COEFFICIENT MATRICES FOR FLOW AND TRANSPORT	00499C
C *****	00500C
582 CALL COGEN (2,B,KH,IT)	00501C
C *****	00502C
C	00503C
C.....TIME STEP MODIFICATION?	00504C
IF (MCD(IT,ITCHNG).NE.0) GO TO 580	00505C
C	00506C
C.....COMPUTE TIME STEP	00507C
IF (IT.LT.1) GO TO 580	00508C
DELT=TCHK/VMAX	00509C
580 CCNTINUE	00510C
C	00511C
C	00512C
C.....SOLVE FOR HYDRAULIC HEAD	00513C
DELTGC=DELT*IGO	00514C
DO 600 I=1,N	00515C
600 CLL(I)=PHI(I)	00516C
C	00517C
C.....SELECT APPROXIMATION FOR TIME DERIVATIVE	00518C
IF (TYPE.EQ.PWE) A3=1./DELTGC	00519C
IF (TYPE.EQ.CNTR) A3=2./DELTGC	00520C
C.....COMPUTE S FROM H AND P	00521C
DO 660 I=1,N	00522C
DO 660 J=1,NB1	00523C
660 S(I,J)=H(I,J)+P(I,J)*A3	00524C
C	00525C
C	00526C
C.....DECMPOSE	00527C
C *****	00528C
CALL DBAND (N,IX)	00529C
C *****	00530C
C	00531C
C.....COMPUTE VECTOR OF KNOWN VALUES	00532C
DO 670 I=1,N	00533C
670 PH(I)=-RT(I)	00534C
N1=N+1	00535C
N2=N-1	00536C
PH(N)=PH(N)+P(N,1)*A3*OLD(N)	00537C
DO 680 I=1,N2	00538C
IAFGI=NB	00539C

LI=MINO(IARG1,M1-I)	005430
FM(I)=FM(I)+P(I,1)*A3*OLD(I)	005440
DC 680 J=2,I1	005450
LM=I+J-1	005460
FM(I)=FM(I)+P(I,J)*A3*OLD(LM)	005470
680 FM(LM)=FM(LM)+P(I,J)*A3*OLD(L)	005480
IF (KOD8.NE.1) GO TO 690	005490
WRITE (6,1350)	005500
WRITE (6,1080)	005510
WRITE (6,1090) (I,FM(I),I=1,N)	005520
C	005530
C.....BACK SUBSTITUTE	005540
C *****	005550
690 CALL SBAND (N)	005560
C *****	005570
C	005580
C.....CALCULATE NEW VALUES	005590
IF (TYPE.EQ.PWD) GO TO 721	005600
DC 700 I=1,N	005610
700 PHI(I)=2.*PHI(I)-OLD(I)	005620
721 CCNTINUE	005630
C	005640
C.....ADJUST FOR CORRECT NODAL SEQUENCE - USE FM FOR TEMPORARY STORAGE	005650
IF=C	005660
DC 770 I=1,NH	005670
IF (LR(I).GT.0) GO TO 760	005680
IF=IF+1	005690
FM(I)=PHI(IF)	005700
GO TO 770	005710
760 FM(I)=PHI(I)	005720
770 CCNTINUE	005730
C.....END CALCULATIONS FOR HEAD	005740
C	005750
C.....CALCULATE CONCENTRATION	005760
C.....SELECT APPROXIMATION FOR TIA& DERIVATIVE	005770
IF (TYPE.EQ.PWE) A3=1./DELT	005780
IF (TYPE.EQ.CHT&) A3=2./DELT	005790
DC 421 I=1,KN	005800
DC 421 J=1,KNB2	005810
421 S(I,J)=0.	005820
C	005830
C.....POINT OF RETURN FOR INNER CONCENTRATION CALCULATION	005840
C.....COMPUTE S FROM CP AND CH AND KNOWN VECTOR, CPM	005850
C.....OFFLINE COLD	005860
581 DC 590 I=1,KN	005870
590 CCL(I)=CONC(I)	005880
KSTART=1	005890
KDIFF=KN-KNB+1	005900
JSTART=KNB+1+KSTART	005910
JSTOP=KNB1	005920
DO 640 I=1,KN	005930
CPM(I)=-CHT(I)	005940
IF (I.LE.KNB) JSTART=JSTART-1	005950
IF (I.GT.KDIFF) JSTOP=JSTOP-1	005960
IF (I.GT.KNB) KSTART=KSTART+1	005970
K=KSTART-1	005980
DO 640 J=JSTART,JSTOP	005990
K=K+1	006000
JACT=J-KSTART	006010

	IF (ADVANC.EQ.NO) GO TO 640	006000
	S(I,JACT)=CH(I,J)+CF(I,J)*A	006010
640	CFE(I)=CFH(I)+CP(I,J)*COLD(K)*A3	006020
	IF (ADVANC.NE.YES) GO TO 6405	006030
	IF=C	006040
	DC 6402 I=1,NH	006050
	IF (KIR(I).GT.0) GO TO 6402	006060
	IF=IP+1	006070
	IF (FQ(I).LT.0) GO TO 6402	006080
	S(IF,KNB)=S(IP,KNE)+FQ(I)	006090
6402	CONTINUE	006100
6405	CONTINUE	006110
	IF (KOD8.NE.1) GO TO 650	006120
	WRITE (6,1320)	006130
	WRITE (6,1080)	006140
	WRITE (6,1090) (I,CFH(I),I=1,KH)	006150
650	CONTINUE	006160
C		006170
C.....	SCIVE CONCENTRATION EQUATIONS	006180
	DC 651 I=1,KH	006190
651	ART(I)=CFH(I)	006200
	IARG1=KNE-1	006210
C	*****	006220
	IF (ADVANC.EQ.NO) GO TO 653	006230
	CALL SOLVE(1,S,ART,KH,IARG1,125,66)	006240
653	CALL SOLVE(2,S,ART,KH,IARG1,125,66)	006250
C	*****	006260
C		006270
	DC 652 I=1,KH	006280
652	CFE(I)=ART(I)	006290
C		006300
C.....	CALCULATE NEW VALUES	006310
	IF (TYPE.EQ.FWD) GO TO 720	006320
	DO 710 I=1,KH	006330
710	CCEC(I)=2.*CFH(I)-COLD(I)	006340
	GO TO 741	006350
720	DC 730 I=1,KH	006360
730	CONC(I)=CFH(I)	006370
C		006380
C.....	ADJUST FOR CORRECT NODAL SEQUENCE - USE CFH FOR TEMPORARY STORAGE	006390
741	KIF=0	006400
	DO 791 I=1,NH	006410
	IF (KIR(I).GT.0) GO TO 780	006420
	KIF=KIP+1	006430
	CFE(I)=CONC(KIP)	006440
	GO TO 791	006450
780	CFE(I)=CONC(I)	006460
791	CONTINUE	006470
755	CONTINUE	006480
	SSEC=SSEC+DELT	006490
	SHIN=SSEC/60.	006500
	SIPE=SHIN/60.	006510
	IT=IT+1	006520
C.....	END TRANSIENT SOLUTION CALCULATIONS	006530
C		006540
C		006550
C.....	C O M P U T	006560
751	IF (MOD(IT,NPRINT).NE.0) GO TO 820	006570
	IF (KOD9.NE.1.AND.KOD10.NE.1) GO TO 820	006580
	IF (IT.EQ.1) GO TO 800	006590

	DELT1=DELT/3600.	006600
	WRITE (6,1360) IT,DELT1,STIME,SMIN,SSEC	006610
800	IF (KOD9.NE.1) GO TO 810	006620
	WRITE (6,1370)	006630
	WRITE (6,1080)	006640
	WRITE (6,1090) (I,FM(I),I=1,NN)	006650
810	IF (KOD10.NE.1) GO TO 820	006660
	WRITE (6,1390)	006670
	WRITE (6,1080)	006680
	WRITE (6,1090) (I,CFM(I),I=1,NN)	006690
C	*****	006700
	IARG1=KOD7	006710
	IF (MOD(IT,IARG1).EQ.0) CALL VCALL	006720
C	*****	006730
	WRITE (6,1400)	006740
	WRITE (6,1410)	006750
820	CONTINUE	006760
	IARG1=KOD12	006770
	IF (MOD(IT,IARG1).NE.0) GO TO 829	006780
	IF (KOD14.LT.0) GO TO 828	006790
	WRITE (7,1053) IT,STIME	006800
	WRITE (7,1051) (CFM(I),I=1,NN)	006810
828	IF (KOD14.GT.0) GO TO 829	006820
	WRITE (7,1052) IT,STIME	006830
	WRITE (7,1051) (FM(I),I=1,NN)	006840
829	IF (STIME.LT.TIME) GO TO 830	006850
	WRITE (6,1420) STIME	006860
	GO TO 836	006870
830	IF (IT.LT.ITMAX) GO TO 520	006880
	WRITE (6,1430) IT	006890
836	IF (KOD14.LT.0) GO TO 827	006900
	WRITE (7,1053) IT,STIME	006910
	WRITE (7,1051) (CFM(I),I=1,NN)	006920
827	IF (KOD14.GT.0) GO TO 821	006930
	WRITE (7,1052) IT,STIME	006940
	WRITE (7,1051) (FM(I),I=1,NN)	006950
	GO TO 821	006960
C.....	END OUTPUT	006970
C		006980
C.....	SELECT INNER OR OUTER CONCENTRATION CALCULATION	006990
	520 ADVANC=NO	007000
	IF (MOD(IT,IGO).EQ.0) ADVANC=YES	007010
	IF (ADVANC.EQ.NO) GO TO 581	007020
	GO TO 582	007030
C		007040
C.....	SELECT NEW DATA SET	007050
	821 GO TO 10	007060
C		007070
C.....	P C B M A T S	007080
1	FORMAT (11X,'SUPPRESS PRINTOUT OF ELEMENT FLOW MATRICES')	007090
2	FORMAT (11X,'PRINT ELEMENT FLOW MATRICES')	007100
3	FORMAT (11X,'SUPPRESS PRINTOUT OF GLOBAL FLOW MATRIX')	007110
4	FORMAT (11X,'PRINT GLOBAL FLOW MATRIX')	007120
5	FORMAT (11X,'SUPPRESS PRINTOUT OF ELEMENT CONCENTRATION MATRICES')	007130
6	FORMAT (11X,'PRINT ELEMENT CONCENTRATION MATRICES')	007140
7	FORMAT (11X,'SUPPRESS PRINTOUT OF GLOBAL CONCENTRATION MATRIX')	007150
8	FORMAT (11X,'PRINT GLOBAL CONCENTRATION MATRIX')	007160
9	FORMAT (11X,'PRINT VELOCITIES EVERY' I4,' TIME STEP(S)')	007170
12	FORMAT (11X,'SUPPRESS PRINTOUT OF RIGHT HAND SIDE VECTORS')	007180
13	FORMAT (11X,'PRINT RIGHT HAND SIDE VECTORS')	007190

14	FORMAT (11X, 'SUPPRESS PRINTOUT OF COMPUTED HEAD')	007200
15	FORMAT (11X, 'PRINT COMPUTED HEAD')	007210
16	FORMAT (11X, 'SUPPRESS PRINTOUT OF CONCENTRATION SOLUTION')	007220
17	FORMAT (11X, 'PRINT CONCENTRATION SOLUTION')	007230
18	FORMAT (11X, 'PUNCH OUTPUT EVERY', I4, ' TIME STEP(S)')	007240
22	FORMAT (11X, 'PUNCH HEAD AND CONCENTRATION')	007250
23	FORMAT (11X, 'PUNCH HEAD ONLY')	007260
24	FORMAT (11X, 'PUNCH CONCENTRATION ONLY')	007270
25	FORMAT (////11X, 'CODE OPTIOAS'/11X, 12(1H-)//)	007280
840	FCREAT (20A4)	007290
850	FORMAT (1H1, 36X, 15HI S O Q U A D 4//28X, 35HGROUNDWATER FLOW AND MA07300 1SS TRANSPORT//25X, 41HWITH ISOPARAMETRIC QUADRILATERAL ELEMENTS//38007310 2X, 14HCCTOBER 1 1971//)	007320
860	FORMAT (11X, 70(1H*)//11X, 20A4//11X, 70(1H*)//)	007330
910	FCREAT (15I5)	007340
920	FORMAT (///11X, 19HFINITE ELEMENT DATA/11X, 19(1H-)/)	007350
930	FORMAT (1H , 10X, 11HNUMBER OF -, 2X, 5HNODES, I24/21X, 1H-, 2X, 8HELEMENT007360 1S, I21/21X, 1H-, 2X, 24HDIRICHLET BOUNDARY NODES, I5/21X, 1H-, 2X, 20HSOUR007370 2CE CE SINK NODES, I9/21X, 1H-, 2X, 20HELEMENTS IN HALFBAND, /21X, 1H , 2X007380 3, 8HFOR FLOW, I21/21X, 1H , 2X, 17HFOR CONCENTRATION, I12/21X, 1H-, 2X, 14H007390 4LEAKY ELEMENTS, I15/21X, 1H-, 2X, 25HCONSTANT CONC. BOUNDARIES, I4, / 521X, 1H-, 2X, 19HLEAKY ELEMENT NODES, I10) 007410	007420
940	FCREAT (3F10.0, 2I10, F10.0, A4/1F10.0, 2I10)	007420
950	FORMAT (////11X, 15HTIME PARAMETERS/11X, 15(1H-)//11X, 26HSIMULATION007430 1 PERIOD IN HOURS, F30.2/11X, 20HINITIAL TIME STEP IN HOURS, F30.6/11X007440 2, 35HMULTIPLIER FOR INCREASING TIME STEP, F21.3/11X, 38HMAXIMUM PERMIO07450 3TTED NUMBER OF TIME STEPS, I13/11X, 44HNUMBER OF TIME STEPS BETWEEN 007460 4CHANGES IN DELT, I12/11X, 31HSACIAL CRITERION FOR TIME STEP, F25.3/1007470 51X, 28HMOLECULAR DIFFUSION CONSTANT, 0FF28.3/ 6 11X, 45HNUMBER OF CALCULATED VALUES BETWEEN 007490 7FBIFIGUTS, I11/11X, 44HNUMBER OF CONCENTRATION STEPS PER PRES. STEP, 007500 6I12) 007510	007520
1052	FORMAT (5X, 4HHEAD, I4, G15.5)	007520
960	FORMAT (///11X, 24HIMPLICIT TIME DERIVATIVE//)	007530
970	FORMAT (///11X, 24HCENTERED TIME DERIVATIVE//)	007540
980	FORMAT (I5, 2F10.0)	007550
983	FCREAT (I5, 3F10.0)	007560
990	FORMAT (///11X, 16HNODE COORDINATES/11X, 16(1H-)/)	007570
1000	FCREAT (1H , 2(11X, 4HNODE, 12X, 1HX, 15X, 1HY, 2X))	007580
1010	FCREAT (1H , 2(10X, I4, 2F16.2)/(11X, I4, 2F16.2, 10X, I4, 2F16.2))	007590
1020	FCREAT (////11X, 27HSOURCE(+) AND SINK(-) NODES/11X, 27(1H-)/11X, 4H00760 1NODE, 10X, 9HDISCHARGE/) 00761	00762
1030	FCREAT (11X, I4, F16.8)	00762
1040	FORMAT (////11X, 30HSOURCE AND SINK CONCENTRATIONS/11X, 30(1H-)/11X00763 1, 4HNODE, 7X, 13HCONCENTRATION/) 00764	00765
1051	FORMAT (5G15.7)	00765
1053	FORMAT (5X, 13HCONCENTRATION, I4, G15.5)	00766
1070	FORMAT (////, 11X, 12HINITIAL HEAD/11X, 12(1H-))	007670
1080	FCREAT (//11X, 6(4HNGD2, 5X, 5HVALUE, 5X))	007680
1090	FCREAT (/ (11X, 6(I4, 2X, 1PE10.3, 3X))	007690
1100	FORMAT (///11X, 21HINITIAL CONCENTRATION/11X, 21(1H-)/11X, 6(4HNODE, 007700 15X, 5HVALUE, 5X)/(11X, 6(I4, 2X, 1PE10.3, 3X))) 007710	007720
1110	FCREAT (I3, 77A1)	007720
1120	FORMAT (1H0, 10X, 17HINVALID CHARACTER, 2X, A1, 5X, 7HELEMENT, I5)	007730
1130	FCREAT (1H0, 10X, 25HINCOMPLETE DATA - ELEMENT, I5)	007740
1140	FORMAT (////11X, 18HELEMENT INCIDENCES/11X, 18(1H-)/)	007750
1150	FORMAT (1H , 10X, 7HELEMENT, 5X, 1H/, 3(1H-), 9R CORNERS , 4(1H-), 1H/, 7X, 007760 11H/, 14(1H-), 7H SIDES , 15(1H-), 1H/) 00777	00778
1160	FORMAT (1H , 4X, I10, 5X, 4I5, 5A, 8I5)	00778
1170	FCREAT (////11X, 18HELEMENT PARAMETERS/11X, 18(1H-)//69X, 12HDISPERS00779	

11VITY/11X,88HELEMENT X-MOBILITY Y-MOBILITY STORAGE POROSITY L00780C  
 2CNGITUDINAL TRANSVERSE THACKNESS//) 00781C  
 1180 FCFEAT (I3,7P10.3) 00782C  
 1190 FCFEAT (13X,I3,4X,1PE10.3,2I,1P2E10.3,3X,0PP4.2,5X,1PE10.3,3X,E10. 00783C  
 13,2X,E10.3) 00784C  
 120C FCFEAT (////11X,20HLEAKAGE COEFFICIENTS/11X,20(1H-)/11X,2(4HNODE, 00785C  
 12X,11HCOEFFICIENT,3X,13HCONCENTRATION,3X)//(11X,2(I3,4X,1PE9.2,3X 00786C  
 2,1FI9.2,7X)) 00787C  
 1230 FCFEAT (/////11X,33HDIRICHLET BOUNDARY NODES FOR FLOW/11X,33(1H-)) 00788C  
 124C FCFEAT (20I4) 00789C  
 1250 FCFEAT (11X,10(1H\*),33H DIRICHLET BOUNDARY NODE FOR FLOW,I4,38HDOE 00790C  
 1S NOT EXIST - EXECUTION TERMINATED ,10(1H\*)) 00791C  
 1260 FCFEAT (11X,20I5) 00792C  
 127C FCFEAT (1H0,10(1H\*),49H NUMBER OF DIRICHLET BOUNDARY NODES FOR FLO 00793C  
 1W READ,I10,35H DISAGREES WITH NUMBER ANTICIPATED ,I10,10(1H\*)) 00794C  
 1280 FCFEAT (/////11X,42HDIRICHLET BOUNDARY NODES FOR CONCENTRATION/11X 00795C  
 1,42(1H-)) 00796C  
 1290 FCFEAT (11X,10(1H\*),42H DIRICHLET BOUNDARY NODE FOR CONCENTRATION, 00797C  
 1I4,38HDOES NOT EXIST - EXECUTION TERMINATED ,10(1H\*)) 00798C  
 130C FCFEAT (1H0,5(1H\*),58H NUMBER OF DIRICHLET BOUNDARY NODES FOR CONC 00799C  
 1ENITATION READ,I10,35H DISAGREES WITH NUMBER ANTICIPATED ,I10,5(100800 00800C  
 2H\*)) 00801C  
 131C FCFEAT (///11X,19HFINITE ELEMENT DATA/11X,19(1E-)/11X,21HTOTAL NUM 00802C  
 1BER OF NODES,I19/11X,33HDIRICHLET BOUNDARY NODES FOR FLOW,I7/11X,200803C  
 27HDISAGREES OF FREEDOM FOR FLOW,I13/11X,34HDIRICHLET BOUNDARY NODES 00804C  
 3FCB CONC.,I6/11X,28HDISAGREES OF FREEDOM FOR CONC.,I12//) 00805C  
 132C FCFEAT (1H1,11X,40HVECTOR OF KNOWN VALUES FOR CONCENTRATION/11X,40 00806C  
 10(1E-)) 00807C  
 1350 FCFEAT (/////11X,31HVECTOR OF KNOWN VALUES FOR HEAD/11X,31(1H-)) 00808C  
 1360 FCFEAT (1H0,/////11X,16HTIME STEP NUMBER,I20/ 00809C  
 1 11X,17HTIME STEP (HOURS),1PE19.3/11X,12HRELAPSED TIME,5X,1P00 00810C  
 2E19.3,6H HOURS/28X,1PE19.3,8H MINUTES/28X,1PE19.3,8H SECONDS) 00811C  
 1370 FCFEAT (//11X,14HHYDRAULIC HEAD/11X,14(1H-)) 00812C  
 1390 FCFEAT (//11X,13HCONCENTRATION/11X,13(1H-)) 00813C  
 140C FCFEAT (/11X,114(1H\*)) 00814C  
 1410 FCFEAT (1H1) 00815C  
 1420 FCFEAT (/////11X,10(1H\*),41HEXECUTION TERMINATED ON TIME-ELAPSED T 00816C  
 1INE1PE12.4,6H HOURS10(1H\*)) 00817C  
 1430 FCFEAT (/////11X,10(1H\*),42HEXECUTION TERMINATED ON TIME STEPS AT S 00818C  
 1IEFI1C,10(1H\*)) 00819C  
 1440 FCFEAT (8F10.0) 00820C  
 1450 FCFEAT (1H0,///11X,21HPARAMETER MULTIPLIERS /11X,21(1H-)/11X, 00821C  
 16HAPMOBY,F15.5/11X,6HAPMOBY,F15.5/11X,3HAST,F18.5/11X,4HAPOR, 00822C  
 2F17.5/11X,6HABLONG,F15.5/11X,6HABTRAN,F15.5/11X,6HATHICK,F15.5/11X 00823C  
 3,5HAPHIX,F16.5/11X,6HACONCI,F15.5/11X,6HFACTOR,F15.5/11X,5HAEAK, 00824C  
 1F16.5) 00825C  
 END 00826C

SUBROUTINE VCAL

PURPOSE - TO CALCULATE VELOCITIES AT ALL NODES AND AN AVERAGE VELOCITY FOR EACH ELEMENT. PRESSURE DERIVATIVES ARE ALSO CALCULATED.

C  
C  
C  
C  
C  
C

008270  
008280  
008290  
006300  
008310  
008320  
008330  
008340  
008350  
008360  
008370  
008380  
008390  
008400  
008410  
008420  
008430  
008440  
008450  
008460  
008470  
008480  
008490  
008500  
008510  
008520  
008530  
008540  
008550  
008560  
008570  
008580  
008590  
008600  
008610  
008620  
008630  
008640  
008650  
008660  
008670  
008680  
008690  
008700  
008710  
008720  
008730  
008740  
008750  
008760  
008770  
008780  
008790  
008800  
008810  
008820  
008830  
008840  
008850  
008860

-----  
IMPLICIT INTEGER\*2 (I-N)

COMMON /SCALAR/

1KOD1, KOD2, KOD3, KOD4, KOD5, KOD6, KOD7, KOD11, KOD12,  
2KOD13, KOD14, KOD15, NH, NE, NP, NB, NB1, KNB,  
3KNE1, KNE2ST, KSTRT, KNEBST, KNEB2

COMMON /DSCALR/

1CF, LIFUSN, ETRANS, ELONG, TCHK, SRCL, DELT, VMAX

COMMON /ARRAY/

1LR, LRC, IN, KLR, KLRC

COMMON /DARRAY/

1S, CP, CH, CPH, CRT, CONC, CONCI, COLD, VX,  
1VY, VLY, VLY, PHOBY, PHOBY, POR, ST, QP, SE, PE, DISPX, DISPY,  
2EISXY, PHI, P, FM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,  
3RT, COEF, HZEEC, COR, DGX, DGY, Q, CPQ, THICK, ART, CCOEF  
DIMENSION S(125,66), P(125,33), H(125,33), CP(125,66), CH(125,66),  
1PHI(150), OLD(150), CONC(150), COLD(150), CPH(150), VX(12), VY(12),  
2CCE(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),  
3CP(16), SRCR(12), SRCRT(16), FQ(150), FM(150), CRT(150),  
4PHII(150), PHOBY(91), PHOBY(91), LR(150), LRC(150), Y(150),  
5X(150), IN(13,91), RT(150), COEF(91), HZEEC(150), POR(91), ST(91),  
6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CPQ(150),  
ATHICK(91), ART(125), CCOEF(91)  
DIMENSION INDX(12), FF(12)

C  
C

WRITE(6,210)

I=C

20 I=I+1

WRITE(6,200) L

DC 10 I=1,12

VX(I)=0.

10 VY(I)=0.

H=IN(13,I)

XCEND=PHOBY(L)/POR(L)

YCEND=PHOBY(L)/POR(L)

NCCURT=0

I=C

30 I=I+1

IF (I.GT.4) GO TO 40

XI=1.

YI=1.

IF (I.EQ.1.OR.I.EQ.4) XI=-1.

IF (I.EQ.1.OR.I.EQ.2) YI=-1.

GC TO 70

40 IF (IN(I,L).EQ.0) GO TO 110

IF (I/2\*2.EQ.I) GO TO 50

IF (IN(I+1,L).EQ.0) GO TO 60

50 CCNTINUE

XI=COE(I+3)

YI=COE(I+1)

GC TO 70

60 CCNTINUE

I2=I/2

```

      XI=COB(I2)
      YI=COB(I2-1)
70  CONTINUE
      *****
      CALL SHAPE (I, M, XI, YI, JF, Dsf)
      *****
      NCCUNT=NCCUNT+1
      EX=C.
      EY=C.
      J=C
      DO 50 K=1, M
80  J=J+1
      IF (IN(J, I).EQ.0) GO TO 80
      J1=IN(J, I)

      COMPUTE PRESSURE DERIVATIVES
      EX=EX+DGI(K)*PM(J1)
      EY=EY+DGY(K)*PM(J1)
90  CONTINUE
      INDI(NCCUNT)=IN(I, I)
      VY(NCCUNT)=-YCOND*EY
      VX(NCCUNT)=-XCOND*EX
110 CONTINUE
      IF (M.EQ.4.AND.I.EQ.4) GO TO 130
      IF (I.LT.12) GO TO 30
130 CONTINUE
      WRITE (6, 220) (INDI(I), VX(I), VY(I), I=1, M)
      IF (L.LT.NE) GO TO 20
170 RETURN

C
210 FORMAT (//11X, 16HPOINT VELOCITIES/11X, 16(1H-))
200 FORMAT ( /11X, 7HELEMENT, I8/11X, 3(4HNODE, 2X, 11HX-COMPONENT, 2X, 11HY-COMPONENT, 6X) )
220 FORMAT ((11X, 3(I4, 2X, 1PE10.3, 3X, 1PE10.3, 7X)) )
      ENI

```

```

00887
00888
00889
00890
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00910
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921

```

SUBROUTINE MATGEN (I,J,L,M)

PURPOSE - TO GENERATE ELEMENT MATRICES

-----  
 IMPLICIT INTEGER\*2 (I-N)

COMMON /SCALAR/

1KOD1, KOD2, KOD3, KOD4, AOD5, KOD6, KOD7, KOD11, KOD12,  
 2KOD13, KOD14, KOD15, NN, ME, NP, NB, NB1, KNB,  
 3KNE1, KNE2ST, KSTRT, KNBEST, KNB2

COMMON /DSCALR/

1CF, LIPUSN, ETRANS, ELONG, TCHK, SRCL, DELT, VMAX

COMMON /ARRAY/

1LS, LRC, IN, KLR, KLRC

COMMON /DARRAY/

1S, CP, CH, CFM, CRT, CONC, CONCI, COLD, VY,  
 1VY, VLX, VLY, FMOBY, FMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,  
 2DISFY, PHI, P, FM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,  
 3RT, CCEF, HZERO, COR, DGX, DGY, Q, CPQ, THICK, ART, CCOEF  
 DIMENSION S(125,66), P(125,33), H(125,33), CP(125,66), CH(125,66),  
 1PHI(150), OLD(150), CJNC(150), COLD(150), CFM(150), VY(12), VY(12),  
 2CCE(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),  
 3QP(16), SRCR(12), SRCRT(16), FQ(150), PH(150), CET(150),  
 4PHII(150), FMOBY(91), FMOBY(91), LR(150), LRC(150), Y(150),  
 5X(150), IN(13,91), RT(150), CCEF(91), HZERO(150), POR(91), ST(91),  
 6KLB(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CPQ(150),  
 ATHICK(91), ART(125), CCOEF(91)

DATA H1/.347854845137454/,H2/.652145154862546/

H11=H1\*H1

H12=H1\*H2

H22=H2\*H2

IF (NF.EQ.4) GO TO 30

10 SRCI=SRCLT(1)+SRCLT(2)+SRCLT(3)+SRCLT(4)

20 SE(I,J)=Q(1)+Q(2)+Q(3)+Q(4)

PE(I,J)=QP(1)+QP(2)+QP(3)+QP(4)

GO TO 50

30 SE(I,J)=H11\*(Q(1)+Q(4)+Q(13)+Q(16))+H12\*(Q(2)+Q(3)+Q(5)+Q(8)+Q(9)+  
 1Q(12)+Q(14)+Q(15))+H22\*(Q(6)+Q(7)+Q(10)+Q(11))

PE(I,J)=H11\*(QP(1)+QP(4)+QP(13)+QP(16))+H12\*(QP(2)+QP(3)+QP(5)+QP(8)+  
 18)+QP(9)+QP(12)+QP(14)+QP(15))+H22\*(QP(6)+QP(7)+QP(10)+QP(11))

40 SRCI=H11\*(SRCLT(1)+SRCLT(4)+SRCLT(13)+SRCLT(16))+H12\*(SRCLT(2)+SRCLT(5)+  
 1LT(3)+SRCLT(5)+SRCLT(8)+SRCLT(9)+SRCLT(12)+SRCLT(15)+SRCLT(14))+H20096

22\*(SRCLT(6)+SRCLT(7)+SRCLT(10)+SRCLT(11))

50 CONTINUE

RETURN

ENTRY SECOND(I)

IF (NF.EQ.4) GO TO 60

SRCF(I)=SRCRT(1)+SRCRT(2)+SRCRT(3)+SRCRT(4)

GO TO 70

60 SRCF(I)=H11\*(SRCRT(1)+SRCRT(4)+SRCRT(13)+SRCRT(16))+H12\*(SRCRT(2)+  
 1SRCRT(3)+SRCRT(5)+SRCRT(8)+SRCRT(9)+SRCRT(12)+SRCRT(15)+SRCRT(14))

2+H22\*(SRCRT(6)+SRCRT(7)+SRCRT(10)+SRCRT(11))

70 CONTINUE

RETURN

C  
C

C  
C

C  
C

00922  
 00923  
 00924  
 00925  
 00926  
 00927  
 00928  
 00929  
 00930  
 00931  
 00932  
 00933  
 00934  
 00935  
 00936  
 00937  
 00938  
 00939  
 00940  
 00941  
 00942  
 00943  
 00944  
 00945  
 00946  
 00947  
 00948  
 00949  
 00950  
 00951  
 00952  
 00953  
 00954  
 00955  
 00956  
 00957  
 00958  
 00959  
 00960  
 00961  
 00962  
 00963  
 00964  
 00965  
 00966  
 00967  
 00968  
 00969  
 00970  
 00971  
 00972  
 00973  
 00974  
 00975  
 00976  
 00977  
 00978  
 00979  
 00980  
 00981  
 00982  
 00983  
 00984  
 00985  
 00986  
 00987  
 00988  
 00989  
 00990

ENTRY THIRD	009820
WRITE (6,130) I	009830
MM2=(M+7)/8*8-7	009840
DC SO K1=1,MM2,8	009850
K2=K1+7	009860
IF (K1.EQ.MM2) K2=M	009870
DO SO I=1,M	009880
80 WRITE (6,140) I, (SE(I,J),J=K1,K2)	009890
90 WRITE (6,120)	009900
WRITE (6,150) I	009910
MM2=(M+7)/8*8-7	009920
DC 110 K1=1,MM2,8	009930
K2=K1+7	009940
IF (K1.EQ.MM2) K2=M	009950
DO 100 I=1,M	009960
100 WRITE (6,140) I, (PE(I,J),J=K1,K2)	009970
110 WRITE (6,120)	009980
RETURN	009990
C	010000
120 PCBPAT (//)	010010
130 PCBPAT (////,10X,7HELEMENT,I+,5X,16HSTIFFNESS MATRIX/)	010020
140 PCBPAT (15,8E15.6)	010030
150 PCBPAT (1H ,9X,7HELEMENT,I4,>X,14HSTORAGE MATRIX/)	010040
END	010050

	<u>SUBROUTINE SBAND (N)</u>	01006
C	-----	01007
	IMPLICIT INTEGER*2 (I-N)	01008
	INTEGER*4 N	01009
	COMMON /SCALAR/	01010
	1KOE1, KOE2, KOD3, KOD4, KOD5, KOD6, KOD7, KOD11, KOD12,	01011
	2KCI13, KCI14, KOD15, NW, NZ, NP, NB, NB1, KNB,	01012
	3KNE1, KNB2ST, KSTRT, KNBEST, KNB2	01013
	COMMON /DSCALR/	01014
	1CF, LIFUSN, ETRANS, ELONG, TCHK, SECL, DELT, VMAX	01015
	COMMON /ARRAY/	01016
	1LE, LRC, IN, KLR, KLRC	01017
	COMMON /DARRAY/	01018
	1S, CP, CH, CPH, CRT, CONC, CONCI, COLD, VX,	01019
	1VY, VLX, VLY, FNOBX, FNOBY, POR, ST, QP, SE, PE, DISPX, DISPY,	01020
	2DISFXI, PHI, P, FM, OLD, H, SRCIT, SRCR, SRCRT, PHII, Y, X, FO,	01021
	3RT, CCEF, HZERO, COR, DGX, DGY, Q, CFQ, THICK, ART, CCOEF	01022
	DIMENSION S(125,66), P(125,33), H(125,33), CP(125,66), CH(125,66),	01023
	1PHI(150), OLD(150), CONC(150), COLD(150), CPH(150), VX(12), VY(12),	01024
	2COR(15), DGX(12), DGY(12), SRCIT(16), SE(12,12), Q(16), PE(12,12),	01025
	3QP(16), SRCR(12), SRCRT(16), PQ(150), FM(150), CRT(150),	01026
	4PHII(150), FNOBX(91), FNOBY(91), LR(150), LRC(150), Y(150),	01027
	5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),	01028
	6KIE(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),	01029
	7THICK(91), ART(125), CCOEF(91)	01030
C	-----	01031
	DC 30 I=1, N	01032
	J=I-NB+1	01033
	IF ((I+1).LE.NB) J=1	01034
	SUM=FM(I)	01035
	K1=I-1	01036
	IF (J.GT.K1) GO TO 20	01037
	DO 10 K=J, K1	01038
	II=I-K+1	01039
	10 SUM=SUM-S(K,II)*PHI(K)	01040
	20 PHI(I)=SUM*S(I,1)	01041
	30 CONTINUE	01042
	DO 60 I1=1, N	01043
	I=N-I1+1	01044
	J=I+NB-1	01045
	IF (J.GT.N) J=N	01046
	SUM=PHI(I)	01047
	K2=I+1	01048
	IF (K2.GT.J) GO TO 50	01049
	DO 40 K=K2, J	01050
	KK=K-I+1	01051
	40 SUM=SUM-S(I,KK)*PHI(K)	01052
	50 PHI(I)=SUM*S(I,1)	01053
	60 CONTINUE	01054
	RETURN	01055
	END	01056

	<u>SUBROUTINE BAND (N, IEX)</u>	01057
C	-----	01058
	IMPLICIT INTEGER*2 (I-N)	01059
	INTEGER*4 N	
	COMMON /SCALAR/	
	1KOD1, KOD2, KOD3, KOD4, KOD5, KOD6, KOD7, KOD11, KOD12,	01062
	2KOD13, KOD14, KOD15, NN, NE, NP, NB, NB1, KNB,	01063
	3KNE1, KNE2ST, KSTRT, KNEBST, KNE2	01064
	COMMON /DSCALE/	01065
	1CF, IIFUSH, ITRANS, ELONG, TCHK, SRCL, DELT, VMAX	01066
	COMMON /ARRAY/	01067
	1LB, LRC, IN, KLR, KLRC	01068
	COMMON /DARRAY/	01069
	1S, CP, CH, CPH, CRT, CONC, CONCI, COLD, VX,	01070
	1VY, VLY, VLY, PMOBX, PMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,	01071
	2DISFX, PHI, P, PM, OLD, H, SRCLT, SECR, SECR1, PHII, Y, X, PQ,	01072
	3RT, CCEP, HZERO, COR, DGX, DGY, Q, CPQ, THICK, ART, CCOEP	01073
	DIMENSION S(125,66), P(125,33), H(125,33), CP(125,66), CH(125,66),	01074
	1PHI(150), OLD(150), CONC(150), COLD(150), CPH(150), VX(12), VY(12),	01075
	2CCE(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),	01076
	3CF(16), SECR(12), SECR1(16), PQ(150), PM(150), CRT(150),	01077
	4PHII(150), PMOBY(91), PMOBY(91), LB(150), LRC(150), Y(150),	01078
	5X(150), IN(13,91), RT(150), CCEP(91), HZERO(150), POR(91), ST(91),	01079
	6RIB(150), KLRC(150), CONCI(150), ELONG(91), ITRANS(91), CPQ(150),	01080
	7THICK(91), ART(125), CCOEP(91)	01081
C	-----	01082
	IEX=0	01083
	DO 50 I=1, N	01084
	IF=5-I+1	01085
	IF (NB.LT.IP) IP=NB	01086
	DO 50 J=1, IP	01087
	IQ=NB-J	
	IF ((I-1).LT.IQ) IQ=I-1	9
	SUM=S(I,J)	01090
	IF (IQ.LT.1) GO TO 20	01091
	DO 10 K=1, IQ	01092
	II=I-K	01093
	JZ=J+K	01094
	10 SUM=SUM-S(II,K+1)*S(II,JZ)	01095
	20 IF (J.NE.1) GO TO 40	01096
	IF (SUM.LE.0.) GO TO 30	01097
	TEMP=1./SQRT(SUM)	01098
	S(I,J)=TEMP	01099
	GO TO 50	01100
	30 WRITE (6,60) I	01101
	WRITE (6,70) N, NB, IP, IQ, I, J, SUM	01102
	IEX=I	01103
	RETURN	01104
	40 S(I,J)=SUM*TEMP	01105
	50 CONTINUE	01106
	RETURN	01107
C		01108
	60 FORMAT (1H1,10X,19HDBAND FAILS AT ROW ,I4)	01109
	70 FORMAT (1H0,6I5,E20.8)	01110
	END	01111

	<u>SUBROUTINE SHAPE (L, S, XI, YI, P, DET)</u>	011120
C	-----	011130
	IMPLICIT INTEGER*2 (I-N)	011140
	COMMON /SCALAR/	011150
	1KOD1, KOD2, KOD3, KOD4, KOD5, KOD6, KOD7, KOD11, KOD12,	011160
	2KOD13, KOD14, KOD15, NN, NE, NP, NB, NB1, KNB,	011170
	3KNE1, KNB2ST, KSTRT, KNBZST, KNB2	011180
	COMMON /DSCALE/	011190
	1CF, IIFUSN, ETRANS, ELONG, TCHK, SRCL, DELT, VMAX	011200
	COMMON /ARRAY/	01210
	1LR, LRC, IN, KLR, KLRC	011220
	COMMON /DARRAY/	011230
	1S, CP, CH, CFH, CRT, CONC, CONCI, COLD, VX,	011240
	1VY, VLX, VLY, PNOBY, PNOBY, POR, ST, QP, SE, PE, DISPX, DISPY,	011250
	2DISFY, PHI, P, PH, OLD, H, SRCLT, SRCR, SECT, PHII, Y, X, FQ,	011260
	3RT, CCEP, HZERO, COR, DGX, DGY, Q, CFQ, THICK, ART, CCJEP	011270
	DIMENSION S(125,66), P(125,33), H(125,33), CP(125,66), CH(125,66),	011280
	1PHI(150), OLD(150), CONC(150), COLD(150), CFH(150), VX(12), VY(12),	011290
	2CCR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),	011300
	3CF(16), SECR(12), SECT(16), FQ(150), PH(150), CRT(150),	011310
	4PHII(150), PNOBX(91), PNOBY(91), LR(150), LRC(150), Y(150),	011320
	5X(150), IN(13,91), RT(150), CCJEP(91), HZERO(150), POR(91), ST(91),	011330
	6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),	011340
	ATHICK(91), ART(125), CCJEP(91)	011350
	DIMENSION ALP(4), DAX(4), DAY(4), BTX(4), BTY(4),	011360
	1DBX(4), DFX(12), DZY(12), F(12), DBY(4)	011370
	-----	011380
		011390
	XI1=1.-XI	011400
	XI2=1.+XI	011410
	YI1=1.-YI	011420
	YI2=1.+YI	011430
		011440
C		011450
C	CORNER HOLE SHAPE FUNCTIONS, BASIC PART	011460
	ALP(1)=.25*XI1*YI1	011470
	ALP(2)=.25*XI2*YI1	011480
	ALP(3)=.25*XI2*YI2	011490
	ALP(4)=.25*XI1*YI2	011500
	DAX(1)=-.25*YI1	011510
	DAX(2)=-.25*YI1	011520
	DAX(3)=-.25*YI2	011530
	DAX(4)=-.25*YI2	011540
	DAY(1)=-.25*XI1	011550
	DAY(2)=-.25*XI2	011560
	DAY(3)=.25*XI2	011570
	DAY(4)=.25*XI1	011580
C		011590
C	CORNER HOLE SHAPE FUNCTIONS, SIDE-DEPENDENT PART	011600
	XQ1=XI-.5	011610
	XQ2=-XI-.5	011620
	YQ1=YI-.5	011630
	YQ2=-YI-.5	011640
	XC1=1.125*XI*XI-.625	011650
	XC2=2.25*XI	011660
	YC1=1.125*YI*YI-.625	011670
	YC2=2.25*YI	011680
	J1=1	011690
	J2=2	011700
	J3=5	011710
	DO 50 J=1,2	011720

	IF (IN(J3,L).EQ.0) GO TO 10	011720C
	IF (IN(J3+1,L).EQ.0) GO TO 20	011730C
	GO TO 30	011740C
1C	CONTINUE	011750C
	BTX(J1)=.5	011760C
	BTX(J2)=.5	011770C
	DBX(J1)=0.	011780C
	DBX(J2)=0.	011790C
	GO TO 40	011800C
2C	CONTINUE	011810C
	BTX(J1)=XQ2	011820C
	BTX(J2)=XC1	011830C
	DBX(J1)=-1.	011840C
	DBX(J2)=1.	011850C
	GO TO 40	011860C
3C	CONTINUE	011870C
	BTX(J1)=XC1	011880C
	BTX(J2)=XC1	011890C
	DBX(J1)=XC2	011900C
	DBX(J2)=XC2	011910C
4C	CONTINUE	011920C
	J1=4	011930C
	J2=3	011940C
	J3=9	011950C
5C	CONTINUE	011960C
	J1=2	011970C
	J2=3	011980C
	J3=7	011990C
	DO 100 J=1,2	012000C
	IF (IN(J3,L).EQ.0) GO TO 60	012010C
	IF (IN(J3+1,L).EQ.0) GO TO 70	012020C
	GO TO 80	012030C
6C	CONTINUE	012040C
	BTX(J1)=.5	012050C
	BTX(J2)=.5	012060C
	DBX(J1)=0.	012070C
	DBX(J2)=0.	012080C
	GO TO 90	012090C
7C	CONTINUE	012100C
	BTX(J1)=YQ2	012110C
	BTX(J2)=YQ1	012120C
	DBX(J1)=-1.	012130C
	DBX(J2)=1.	012140C
	GO TO 90	012150C
8C	CONTINUE	012160C
	BTX(J1)=YC1	012170C
	BTX(J2)=YC1	012180C
	DBX(J1)=YC2	012190C
	DBX(J2)=YC2	012200C
9C	CONTINUE	012210C
	J1=1	012220C
	J2=4	012230C
	J3=11	012240C
100	CONTINUE	012250C
C		012260C
C	SHAPE FUNCTION DERIVATIVE MATRIX - CORNER NODES	012270C
	DO 110 J=1,4	012280C
	DPX(J)=DAX(J)*(BTX(J)+BTY(J))+DBX(J)*ALP(J)	012290C
	DPY(J)=DAY(J)*(BTX(J)+BTY(J))+DBY(J)*ALP(J)	012300C
	P(J)=ALP(J)*(BTX(J)+DPY(J))	012310C

110 CONTINUE

C  
C

```
SHAPE FUNCTION DERIVATIVE MATRIX - EDGE NODES
IF (N.EQ.4) GO TO 240
J=4
YEQ=1.-XI*YI
YEQ=1.-YI*YI
XE1=1.-3.*XI
XE2=1.+3.*XI
YE1=1.-3.*YI
YE2=1.+3.*YI
IF (IN(5,L).EQ.0) GO TO 140
IF (IN(6,L).EQ.0) GO TO 120
GO TO 130
120 J=J+1
DPX(J)=-XI*YI1
DPY(J)=-.5*YEQ
F(J)=.5*YEQ*YI1
GO TO 140
130 J=J+1
DPX(J)=-.28125*YI1*(3.*YEQ+2.*XI*XE1)
DPY(J)=-.28125*YEQ*XE1
F(J)=.28125*YEQ*XE1*YI1
J=J+1
DPX(J)=.28125*YI1*(3.*YEQ-2.*XI*XE2)
DPY(J)=-.28125*YEQ*XE2
F(J)=.28125*YEQ*XE2*YI1
140 IF (IN(7,L).EQ.0) GO TO 170
IF (IN(8,L).EQ.0) GO TO 150
GO TO 160
150 J=J+1
DPX(J)=.5*YEQ
DPY(J)=-YI*YI2
F(J)=.5*YI2*YEQ
GO TO 170
160 J=J+1
DPX(J)=.28125*YEQ*YE1
DPY(J)=-.28125*YI2*(3.*YEQ+2.*YI*YE1)
F(J)=.28125*YI2*YEQ*YE1
J=J+1
DPX(J)=.28125*YEQ*YE2
DPY(J)=.28125*YI2*(3.*YEQ-2.*YI*YE2)
F(J)=.28125*YI2*YEQ*YE2
170 IF (IN(9,L).EQ.0) GO TO 200
IF (IN(10,L).EQ.0) GO TO 180
GO TO 190
180 J=J+1
DPX(J)=-XI*YI2
DPY(J)=.5*YEQ
F(J)=.5*YEQ*YI2
GO TO 200
190 J=J+1
DPX(J)=.28125*YI2*(3.*YEQ-2.*XI*XE2)
DPY(J)=.28125*YEQ*XE2
F(J)=.28125*YEQ*XE2*YI2
J=J+1
DPX(J)=-.28125*YI2*(3.*YEQ+2.*XI*XE1)
DPY(J)=.28125*YEQ*XE1
F(J)=.28125*YEQ*XE1*YI2
200 IF (IN(11,L).EQ.0) GO TO 230
```

0123200  
0123300  
0123400  
0123500  
0123600  
0123700  
0123800  
0123900  
0124000  
0124100  
0124200  
0124300  
0124400  
0124500  
0124600  
0124700  
0124800  
0124900  
0125000  
0125100  
0125200  
0125300  
0125400  
0125500  
0125600  
0125700  
0125800  
0125900  
0126000  
0126100  
0126200  
0126300  
0126400  
0126500  
0126600  
0126700  
0126800  
0126900  
0127000  
0127100  
0127200  
0127300  
0127400  
0127500  
0127600  
0127700  
0127800  
0127900  
0128000  
0128100  
0128200  
0128300  
0128400  
0128500  
0128600  
0128700  
0128800  
0128900  
0129000  
0129100

	IF (IN(12,L).EQ.0) GO TO 210	0129200
	GO TO 220	0129300
210	J=J+1	0129400
	DFX (J)=-.5*YEQ	0129700
	DFY (J)=-YI*XI1	0129800
	F (J)=.5*XI1*YEQ	0129900
	GO TO 230	0130000
220	J=J+1	0130100
	DFX (J)=-.28125*YEQ*YE2	0130200
	DFY (J)=.28125*XI1*(3.*YEQ-2.*YI*YE2)	0130300
	F (J)=.28125*XI1*YEQ*YE2	0130400
	J=J+1	0130500
	DFX (J)=-.28125*YEQ*YE1	0130600
	DFY (J)=-.28125*XI1*(3.*YEQ+2.*YI*YE1)	0130700
	F (J)=.28125*YEQ*YE1*XI1	0130800
230	CONTINUE	0130900
240	CONTINUE	0131000
C	JACCBIAN	0131100
C	SUM1=0.	0131200
	SUM2=0.	0131300
	SUM3=0.	0131400
	SUM4=0.	0131500
	R=0	0131600
	DO 260 I=1,N	0131700
250	K=K+1	0131800
	IF (IN(K,L).EQ.0) GO TO 250	0131900
	KI=IN (K,L)	0132000
	SUM1=SUM1+DFX (I)*X (KI)	0132100
	SUM2=SUM2+DFY (I)*Y (KI)	0132200
	SUM3=SUM3+DFX (I)*X (KI)	0132300
	SUM4=SUM4+DFY (I)*Y (KI)	0132400
260	CONTINUE	0132500
	DET=SUM1*SUM4-SUM2*SUM3	0132600
	DET1=1./DET	0132700
	C11=DET1*SUM4	0132800
	C12=-DET1*SUM2	0132900
	C21=-DET1*SUM3	0133000
	C22=DET1*SUM1	0133100
C	SHAPE FUNCTION DERIVATIVES - GLOBAL	0133200
C	DO 270 J=1,N	0133300
	DGX (J)=C11*DFX (J)+C12*DFY (J)	0133400
	DGY (J)=C21*DFX (J)+C22*DFY (J)	0133500
270	CONTINUE	0133600
	RETURN	0133700
	END	0133800

240 LIVE MARIY - EDGE NODES

I\*XI  
\*Y  
\*Y  
\*Y  
\*Y  
\*Y  
\*Y  
1.EQ.0) GO TO 140  
1.EQ.0) GO TO 120

YI1  
XEQ  
\*YI1

25\*YI1\*(3.\*XEQ+2.\*YI\*XE1)  
25\*XEQ\*XE1  
:EC\*XE1\*YI1

\*YI1\*(3.\*XEQ-2.\*YI\*XE2)  
5\*XEQ\*XE2  
:C\*YE2\*YI1  
-0) GO TO 170  
-0) GO TO 150

Q\*YE1  
E2\*(3.\*YEQ+2.\*YI\*YE1)  
:EQ\*YE1

\*YE2  
\*(3.\*YEQ-2.\*YI\*YE2)  
:Q\*YE2  
GO TO 200  
GO TO 180

EQ-2.\*YI\*XE2)

Q+2.\*YI\*XE1)

01232000  
01233000  
01234000  
01235000  
01236000  
01237000  
01238000  
01239000  
01240000  
01241000  
01242000 CI, COLD, VX,  
01243000 PE, DISPX, DISPY,  
01244000 IT, PHII, Y, X, PQ,  
01245000 K, ART, CCOEF  
01246000 25,66), CH(125,66),  
01247000 (150), VX(12), VY(12),  
01248000 (12), Q(16), PE(12,12),  
01249000 0), CRT(150),  
01250000 RC(150), Y(150),  
01251000 (150), POR(91), ST(91),  
01252000 , ETRANS(91), CPQ(150),  
01253000  
01254000  
01255000 33333333, .33333333333, 1.  
01256000 -1./  
01257000  
01258000  
01259000  
01260000  
01261000  
01262000  
01263000  
01264000  
01265000  
01266000  
01267000  
01268000  
01269000  
01270000  
01271000  
01272000  
01273000  
01274000  
01275000  
01276000  
01277000  
01278000  
01279000  
01280000  
01281000  
01282000  
01283000  
01284000  
01285000  
01286000  
01287000  
01288000  
01289000  
01290000  
01291000

KOD11, KOD12,  
NB1, KNB,

0  
0  
01  
01  
01  
013  
013  
0135  
0135  
01361  
01362  
01363  
01364  
01365  
01366  
01367  
01368

	SUBROUTINE SOLVE (KKK,C,R,NMF,IHALFB,MAXNP,MAXBW)	0136900
	DIMENSION C(MAXNP,MAXBW),R(MAXNP)	0137000
	IHEI=IHALFB+1	0137100
	IF (KKK.EQ.2) GO TO 50	0137200
	NU=NMF-IHALFB	0137300
	DC 20 NI=1,NU	0137400
	PIVCTI=1./C(NI,IHBP)	0137500
	NJ=NI+1	0137600
	IE=IHBP	0137700
	NK=NI+IHALFB	0137800
	DC 10 NL=NJ,NK	0137900
	IE=IB-1	0138000
	A=-C(NL,IE)*PIVCTI	0138100
	C(NI,IB)=A	0138200
	JE=IB+1	0138300
	KE=IB+IHALFB	0138400
	IE=IHBP-IE	0138500
	DO 1C MB=JE,KB	0138600
	NB=IB+MB	0138700
10	C(NI,NB)=C(NL,NB)+A*C(NI,NB)	0138800
2C	CONTINUE	0138900
	NR=NU+1	0139000
	NU=NMF-1	0139100
	NK=NMF	0139200
	DC 40 NI=NR,NU	0139300
	PIVCTI=1./(C(NI,IHBP))	0139400
	NJ=NI+1	0139500
	IE=IHBP	0139600
	DC 30 NL=NJ,NK	0139700
	IE=IB-1	0139800
	A=-C(NL,IE)*PIVCTI	0139900
	C(NI,IB)=A	0140000
	JB=IB+1	0140100
	KB=IB+IHALFB	0140200
	IE=IHBP-IE	0140300
	DO 30 MB=JB,KB	0140400
	NB=IB+MB	0140500
30	C(NI,NB)=C(NL,NB)+A*C(NI,NB)	0140600
4C	CONTINUE	0140700
	RETURN	0140800
C		0140900
50	NU=NMF+1	0141000
	IBAND=2*IHALFB+1	0141100
	DC 70 NI=2,IHBF	0141200
	IE=IHBP-NI+1	0141300
	NJ=1	0141400
	SUM=0.0	0141500
	DO 60 JB=IB,IHALFB	0141600
	SUM=SUM+C(NI,JB)*R(NJ)	0141700
60	NJ=NJ+1	0141800
70	R(NI)=R(NI)+SUM	0141900
	IE=1	0142000
	NI=IHBP+1	0142100
	DC 50 NI=NL,NMF	0142200
	NJ=NI-IHBP+1	0142300
	SUM=0.	0142400
	DC 80 JB=IB,IHALFB	0142500
	SUM=SUM+C(NI,JB)*R(NJ)	0142600
80	NJ=NJ+1	0142700
90	R(NI)=R(NI)+SUM	0142800

```

C  EACK SCLVE
  R(NBP)=R(NNP)/C(NNP,IHBP)
  CC 110 IB=2,IHBP
  NI=KU-IB
  NJ=KI
  ME=IHALPB+IB
  SUM=0.
  DO 100 JB=NL,ME
  NJ=EJ+1
100  SUP=SUM+C(NI,JE)*R(NJ)
110  R(NI)=(R(NI)-SUM)/C(NI,IHBP)
  ME=IBAND
  DO 130 IB=NL,NNP
  NI=KU-IB
  NJ=KI
  SUM=0.
  DO 120 JE=NL,ME
  NJ=EJ+1
120  SUP=SUM+C(NI,JE)*R(NJ)
130  R(NI)=(R(NI)-SUM)/C(NI,IHBP)
  RETURN
  ENC

```

```

0142900
0143000
0143100
0143200
0143300
0143400
0143500
0143600
0143700
0143800
0143900
0144000
0144100
0144200
0144300
0144400
0144500
0144600
0144700
0144800
0144900
0145000

```

C  
C  
C  
C

SUBROUTINE COGEN (JTEST,N,KN,IT)

PURPOSE - TO GENERATE GLOBAL COEFFICIENT MATRICES FOR FLOW  
AND CONCENTRATION

```

-----
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 N,KN,IT,JTEST
COMMON /SCALAR/
1KOD1, KOD2, KOD3, KOD4, KOD5, KOD6, KOD7, KOD11, KOD12,
2KOD13, KOD14, KOD15, JN, JZ, NP, NB, NB1, KNB,
3KNB1, KNB2ST,KSTRT, KNB2ST,KNB2
COMMON /DSCALE/
1CF,EIFUSN,ETRANS,ELONG,TCHK,SECL,BZLT,VMAX
COMMON /ARRAY/
1LR, LRC, IN, KLR, KLRC
COMMON /DARRAY/
1S, CP, CH, CPH, CRT, CONC, CONCI, COLD, VX,
1VY, VLX, VLY, PHOBY, PHOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
2DISIXY, PHI, P, PH, OLD, H, SCLT, SRCR, SRCRT, PHII, Y, X, PO,
3RT, COEF, HZERO, COR, DGX, DGY, Q, CPQ, THICK, ART, CCOEF
DIMENSION S(125,66),P(125,33),H(125,33),CP(125,66),CH(125,66),
1PHI(150), OLD(150), CONC(150),COLD(150),CPH(150), VX(12), VY(12),
2COB(15), DGX(12), DGY(12), SCLT(16), SE(12,12), Q(16), PE(12,12),
3CF(16), SRCR(12), SRCRT(16), PQ(150), PH(150), CRT(150),
4PHII(150), PHOBY(91), PHOBY(91), LR(150), LRC(150), Y(150),
5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
6KLB(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CPQ(150),
ATHICK(91),ART(125),CCOEF(91)
DIMENSION AG(6), P(12,16), PF(12), DX(12,16),DY(12,16),
1BETJ(16),JD(24),VKY(16),VKX(16),CPHI(16),CK(16)
DIMENSION HK(16)
DATA AG/-.577350269189626, .577350269189626, -.861136311594053,
1-.339981043584856, .339981043584856, .861136311594053/
-----
KII=0
II=C
IF (JTEST.EQ.0) GO TO 50
DO 10 I=1,N
DO 10 J=1,NB
P(I,J)=0.
10 H(I,J)=0.
DO 40 I=1,NN
IF (LR(I).GT.0) GO TO 40
II=II+1
RT(II)=-PQ(I)
CONTINUE
20 IF (JTEST.EQ.1) GO TO 100
DO 60 I=1,KN
DO 60 J=1,KNB2ST
CF(I,J)=0.
60 CH(I,J)=0.
DO 50 I=1,NN
IF (KLR(I).GT.0) GO TO 90
KII=KII+1
IF (CFQ(I)) 80,70,70
70 CRT(KII)=-PQ(I)*CFQ(I)
GO TO 90
30 CRT(KII)=0

```

01451000  
01452000  
01453000  
01454000  
01455000  
01456000  
01457000  
01458000  
01459000  
01460000  
01461000  
01462000  
01463000  
01464000  
01465000  
01466000  
01467000  
01468000  
01469000  
01470000  
01471000  
01472000  
01473000  
01474000  
01475000  
01476000  
01477000  
01478000  
01479000  
01480000  
01481000  
01482000  
01483000  
01484000  
01485000  
01486000  
01487000  
01488000  
01489000  
01490000  
01491000  
01492000  
01493000  
01494000  
01495000  
01496000  
01497000  
01498000  
01499000  
01500000  
01501000  
01502000  
01503000  
01504000  
01505000  
01506000  
01507000  
01508000  
01509000  
01510000

90	CONTINUE	0151100
100	CCONTINUE	0151200
	VHAX=0.	0151300
	I=0	0151400
110	I=I+1	0151500
	H=IN(13,I)	0151600
	NP=2	0151700
	IF (H.GT.4) NP=4	0151800
	NP2=NP*NP	0151900
C		0152000
C	SHAPE FUNCTIONS FOR INTEGRATION POINTS	0152100
C	INTEGRATION BY GAUSSIAN QUADRATURE	0152200
C	2 X 2 RULE FOR FULLY LINEAR ELEMENTS	0152300
C	4X4 RULE FOR ALL OTHER ELEMENTS	0152400
	DC 150 I=1, NP	0152500
	DO 150 J=1, NP	0152600
	K= (I-1)*NP+J	0152700
	IF (NP.EQ.4) GO TO 120	0152800
	XI=AG(J)	0152900
	YI=2G(I)	0153000
	GO TO 130	0153100
120	XI=AG(J+2)	0153200
	YI=AG(I+2)	0153300
130	CCONTINUE	0153400
C	*****	0153500
C	CALL SHAPE (L,H,XI,YP,PP,DET)	0153600
	*****	0153700
	DC 140 JJ=1,H	0153800
	F(JJ,K)=PP(JJ)	0153900
	DX(JJ,K)=EGX(JJ)	0154000
140	DY(JJ,K)=DGY(JJ)	0154100
	DETJ(K)=DET	0154200
150	CCONTINUE	0154300
		0154400
C	COMPUTE FLOW COEFFICIENT MATRICES	0154500
	R=C	0154600
	DC 170 I=1,H	0154700
160	K=R+1	0154800
	IF (IN(K,L).EQ.0) GO TO 160	0154900
	JD(I)=IN(K,L)	0155000
170	CCONTINUE	0155100
		0155200
C	THE ARRAY JD NOW CONTAINS THE INCIDENCES OF THE ACTIVE NODES IN	0155300
C	ELEMENT L.	0155400
	IF (JTEST.EQ.0) GO TO 290	0155500
	IF (COEF(L).EQ.0.) GO TO 793	0155600
	DO 792 K=1, NP2	0155700
	HK(K)=0.	0155800
	DO 792 J=1,H	0155900
	JEJ=JE(J)	0156000
792	HK(K)=HK(K)+F(J,K)*HZERO(JDJ)	0156100
793	CCONTINUE	0156200
	DO 190 I=1,H	0156300
	DO 190 J=I,H	0156400
	DO 180 K=1, NP2	0156500
	SRCIT(K)=COEF(L)*F(I,K)*F(J,K)*DETJ(K)	0156600
	QP(K)=ST(I)*F(I,K)*F(J,K)*DETJ(K)	0156700
	Q(K)=(PHOBX(L)*DX(I,K)*DX(J,K)+PHOBY(L)*DY(I,K)*DY(J,K))*DETJ(K)	0156800
180	CCONTINUE	0156900
C	*****	0157000



	DC 292 K=1, NP2	016310
	VKY (K)=0.	016320
	VKY (K)=0.	016330
	CK (K)=0.	016340
	CPHI (K)=0.	016350
	DC 291 J=1, H	016360
	JEJ=JL (J)	016370
	CK (K)=CK (K)+F (J, K)*CFX (JDJ)	016380
	CPHI (K)=CPHI (K)+F (J, K)*FX (JDJ)	016390
	VKY (K)=VKY (K)+ICOND*DL (J, K)*FH (JDJ)	016400
291	VKY (K)=VKY (K)+YCOND*DY (J, K)*FH (JDJ)	016410
295	VLX=VLX+VKX (K)	016420
	VLY=VLY+VKY (K)	016430
292	CONTINUE	016440
	VLY=VLY/(NP2*THICK (L))	016450
	VLX=VLX/(NP2*THICK (L))	016460
	VMAX=AMAX1 (VMAX, ABS (VLX), ABS (VLY))	016470
	DO 310 I=1, H	016480
	DO 310 J=1, H	016490
	DO 300 K=1, NP2	016500
	VXSQRD=VKX (K)*VKX (K)	016510
	VYSQRD=VKY (K)*VKY (K)	016520
	VSQRD=VXSQRD+VYSQRD	016530
	IF (VSQRD.NE.0.) GO TO 723	016540
	DISEX=DIFUSH	016550
	DISEY=DIFUSH	016560
	DISEXY=0.	016570
	GO TO 724	016580
723	V=SQRT (VSQRD)	016590
	DI=ELONG (I)*V	016600
	DT=ITEANS (L)*V	016610
	DVX=VXSQRD/VSQRD	016620
	DVY=VYSQRD/VSQRD	016630
	DISEX=DI*DVX+DT*DVI+DIFUSH	016640
	DISEY=DT*DVX+DI*DVY+DIFUSH	016650
	DISEXY=(DI-DT)*(VKX (K)*VKY (K)/VSQRD)	016660
724	SECT (K)=C.	016670
	IF (CCOEF (L).GT.0.) SECLT (K)=COEF (L)*(CPHI (K)-HK (K))*F (I, K)*F (J, K)	016680
	1*DETJ (K)*0.5	016690
	QP (K)=POR (L)*THICK (L)*F (I, K)*F (J, K)*DETJ (K)	016700
	Q (K)=((DISPX*DX (I, K)*DX (J, K)+DISPY*DY (I, K)*DY (J, K)+DISPXY	016710
	1*(DY (I, K)*DX (J, K)+DX (I, K)*DY (J, K))+ (VKX (K)*DX (J, K)+VKY (K)*DY (J, K))	016720
	2*F (I, K)*FOR (L))*DETJ (K)	016730
300	CONTINUE	016740
C	*****	016750
	CALL SECT (I, J, K)	016760
C	*****	016770
310	SE (I, J)=SE (I, J)+SECL	016780
	DC 336 I=1, H	016790
	IF (CCOEF (L).LE.0.) GO TO 337	016800
	DO 312 K=1, NP2	016810
312	SECT (K)=-COEF (L)*(HK (K)-CPHI (K))*CCOEF (L)*F (I, K)*DETJ (K)	016820
C	*****	016830
311	CALL SECOND (I)	016840
C	*****	016850
	GO TO 336	016860
337	SECT (I)=0.	016870
336	CONTINUE	016880
	IF (KOD3.NE.1) GO TO 320	016890
C		016900

C	PRINT ELEMENT MATRICES FOR CONCENTRATION	016910
	IF (KCD3.NE.1) GO TO 320	016920
	WRITE (6,500)	016930
C	*****	0
	CALL THIRD	01
C	*****	016960
	320 CONTINUE	016970
C		016980
C	ASSEMBLY OF GLOBAL COEFFICIENT MATRIX	016990
	DC 370 I=1,N	017000
	JEI=JE(I)	017010
	IF (KIR(JEI).GT.0) GO TO 350	017020
	IB=JDI-KLRC(JDI)	017030
	CHI(IB)=CBI(IR)-SACR(I)	017040
	DC 340 J=1,N	017050
	JEJ=JE(J)	017060
	IF (KIR(JEJ).GT.0) GO TO 340	017070
	JC=JDJ-KLRC(JDJ)-IR+KNBEST	017080
	IF (JC.LE.KNB2ST) GO TO 330	017090
	WRITE (6,510) L,JC,KNB2	017100
	STOP	017110
330	CH(IR,JC)=CH(IR,JC)+SZ(I,J)	017120
	CF(IR,JC)=CF(IR,JC)+PZ(I,J)	017130
	IF (JC.GT.KNB1) KNB1=JC	017140
340	CONTINUE	017150
	GO TO 370	017160
350	CONTINUE	017170
	DC 360 J=1,N	017180
	JEJ=JE(J)	017190
	IF (KIR(JEJ).GT.0) GO TO 360	017200
	JC=JEJ-KLRC(JDJ)	017210
	CBI(JC)=CBI(JC)+SZ(J,I)*CONCI(JDI)	017220
360	CONTINUE	017230
370	CONTINUE	017240
380	IF (L.LT.NE) GO TO 110	017250
	NE=NB1	017260
	KNBCL=KNE	017270
	KNE=KNB1-KNBEST+1	017280
	KSTST=KNB2ST-KNB1	017290
	KNB2=2*KNE-1	017300
	IF (IT.LE.1) WRITE (6,520) NB,KNB	017310
	IF (JTEST.EQ.0) GO TO 430	017320
	IF (KOD2.NE.1) GO TO 430	017330
	WRITE (6,530)	017340
	WRITE (6,540)	017350
	NB2=(NB+9)/10*10-9	017360
	DO 400 K1=1,NB2,10	017370
	K2=K1+9	017380
	IF (K1.EQ.NB2) K2=NB	017390
	DC 390 I=1,N	017400
390	WRITE (6,570) I, (H(I,J),J=K1,K2)	017410
400	WRITE (6,550)	017420
	WRITE (6,580)	017430
	NB2=(NB+9)/10*10-9	017440
	DO 420 K1=1,NB2,10	017450
	K2=K1+9	017460
	IF (K1.EQ.NB2) K2=NB	017470
	DC 410 I=1,N	01748
410	WRITE (6,570) I, (P(I,J),J=K1,K2)	017
420	WRITE (6,550)	017

	WRITE (6,590)	017510
	WRITE (6,600) (RT(I),I=1,N)	017520
430	IF (JTEST.EQ.1) GO TO 480	017530
	IF (KCD4.NE.1) GO TO 480	017540
	WRITE (6,610)	017550
	WRITE (6,620)	017560
	NB2=(KNB2+9)/10*10-9	017570
	DC 450 K1=1,NB2,10	017580
	K2=K1+9	017590
	IF (K1.EQ.NB2) K2=KNB2	017600
	DO 440 I=1,KN	017610
440	WRITE (6,570) I, (CH(I,J+KST-1),J=K1,K2)	017620
450	WRITE (6,550)	017630
	WRITE (6,630)	017640
	DO 470 K1=1,NB2,10	017650
	K2=K1+9	017660
	IF (K1.EQ.NB2) K2=KNB2	017670
	DC 460 I=1,KN	017680
460	WRITE (6,570) I, (CP(I,J+KST-1),J=K1,K2)	017690
470	WRITE (6,550)	017700
	WRITE (6,590)	017710
	WRITE (6,600) (CRT(I),I=1,KN)	017720
480	CONTINUE	017730
	RETURN	017740
C		017750
490	FORMAT (1H1,////11X,25HELEMENT MATRICES FOR FLOW/11X,25(1H-))	017760
500	FORMAT (1H1,////11X,34HELEMENT MATRICES FOR CONCENTRATION/11X,34(1H-))	017770
510	FORMAT (1H0,10X,7HELEMENT,14,5X,33HINSUFFICIENT BAND WIDTH - REQUIRE	017780
	1RE,15,2X,10HINSTEAD OF,15)	017790
520	FORMAT (1H0,10X,41HGLOBAL COEFFICIENT MATRIX HALF BAND WIDTH/11X,80	017800
	1HFOR FLOW,115/11X,17HFOR CONCENTRATION,16)	017810
530	FORMAT (1H1,10X,36HGLOBAL COEFFICIENT MATRICES FOR FLOW/11X,36(1H-	017820
	1))	017830
540	FORMAT (1H0,10X,37HA COEFFICIENT MATRIX - UPPER HALFBAND/11X,37(1H-	017840
	1-)//)	017850
550	FORMAT (//)	017860
560	FORMAT (1H0,10X,7HELEMENT,14,5X,38HINSUFFICIENT HALF-BAND WIDTH -	017870
	1REQUIRE,15,2X,10HINSTEAD OF,15)	017880
570	FORMAT (15,10E12.4)	017890
580	FORMAT (1H0,10X,37HP COEFFICIENT MATRIX - UPPER HALFBAND/11X,37(1H-	017900
	1-)//)	017910
590	FORMAT (////,11X,20HP COEFFICIENT MATRIX/11X,20(1H-)//)	017920
600	FORMAT (11X,10E12.3)	017930
610	FORMAT (1H1,10X,37HGLOBAL COEFFICIENT MATRICES FOR CONC./11X,37(1H-	017940
	1-))	017950
620	FORMAT (1H0,10X,20HA COEFFICIENT MATRIX/11X,20(1H-)//)	017960
630	FORMAT (1H0,10X,20HP COEFFICIENT MATRIX/11X,20(1H-)//)	017970
	END	017980
		017990