PINDER

SCP
5-1765

3384

NNA.870806.0062

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

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

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,

$$L(h) \equiv \frac{\partial}{\partial x}\left(T_{xx}\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(T_{yy}\frac{\partial h}{\partial y}\right) - S\frac{\partial h}{\partial t} - Q - \frac{K}{\ell'}(h-h_w) - C(h-h_i) \tag{2-2}$$

where

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

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

S is the storage coefficient $[L^\circ]$,

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_i 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 δ (note this is not the Kroneker delta commonly encountered in applied mathematics) and define Q as

$$Q = \sum_{i=1}^{n} Q_w(x_i,y_i)\ \delta(x-x_i)\ (y-y_i) \tag{2-3}$$

The Dirac delta function has the property that, when integrated, the function Q will equal Q_w at (x_i, y_i). 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[1] 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

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

where

$$\omega = 2K\, t/(2\ell'^2\, S_s),$$

and S_s 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

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

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 othogonal 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

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

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, $i = 1, 2, \ldots N$ which generally makes our solution approximate rather than exact.

[1]Bredehoeft, J. D. and G. F. Pinder, Digital analysis of areal flow in multiaquifer 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, $i = 2,\ldots N$. Recalling the definition of orthogonal functions these N conditions can be expressed as

$$\iint L(\hat{h}) w_i \, dxdy = 0 \qquad i = 1, 2, \ldots N \tag{2-6}$$

Introducing the definition of (2-2) we obtain

$$\iint [\, \frac{\partial}{\partial x} (T_{xx} \frac{\partial \hat{h}}{\partial x}) + \frac{\partial}{\partial y} (T_{yy} \frac{\partial \hat{h}}{\partial y}) - S \frac{\partial \hat{h}}{\partial t} - Q - \frac{K}{\ell'} (\hat{h} - h_w) - C(\hat{h} - h_i)] \; w_i \, dxdy = 0 \quad i = 1, 2, \ldots N \tag{2-7}$$

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

$$\iint (T_{xx} \frac{\partial \hat{h}}{\partial x} \frac{\partial w_i}{\partial x} + T_{yy} \frac{\partial \hat{h}}{\partial y} \frac{\partial w_i}{\partial y}) \, dxdy - \int (T_{xx} \frac{\partial \hat{h}}{\partial x} \ell_x + T_{yy} \frac{\partial \hat{h}}{\partial y} \ell_y) \, w_i dS$$
$$+ \iint [S \frac{\partial \hat{h}}{\partial t} + Q + \frac{K}{\ell'} (\hat{h} - h_w) + C(\hat{h} - h_i)] \, w_i \, dxdy = 0 \quad i = 1, 2, \ldots N \tag{2-8}$$

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.

$$\iint [T_{xx} (\sum_{j=1}^{N} H_j \frac{\partial w_j}{\partial x}) \frac{\partial w_i}{\partial x} + T_{yy} (\sum_{j=1}^{N} H_j \frac{\partial w_j}{\partial y}) \frac{\partial w_i}{\partial y} + S \frac{\partial}{\partial t} \sum_{j=1}^{N} H_j w_j w_i$$
$$+ Q w_i + (\frac{K}{\ell'} + C) \sum_{j=1}^{N} H_j w_j w_i - (\frac{K}{\ell'} h_w + C h_i) w_i] \, dxdy \tag{2-9}$$
$$- \int [T_{xx} \frac{\partial \hat{h}}{\partial x} \ell_x + T_{yy} \frac{\partial \hat{h}}{\partial y} \ell_y] \, w_i \, dS = 0 \quad i = 1, 2, \ldots N$$

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

S. 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_i.

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

$$\iint \frac{\partial w_i}{\partial x} \frac{\partial w_j}{\partial x} \, dxdy \text{ and } \iint w_i \, w_j \, dxdy$$

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

The original Galerkin formulation assumed that each basis function w_i 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 (ξ, η) co-ordinate system (local co-ordinates). A typical element in (x,y) space and its transformed equivalent in (ξ,η) 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

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

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.

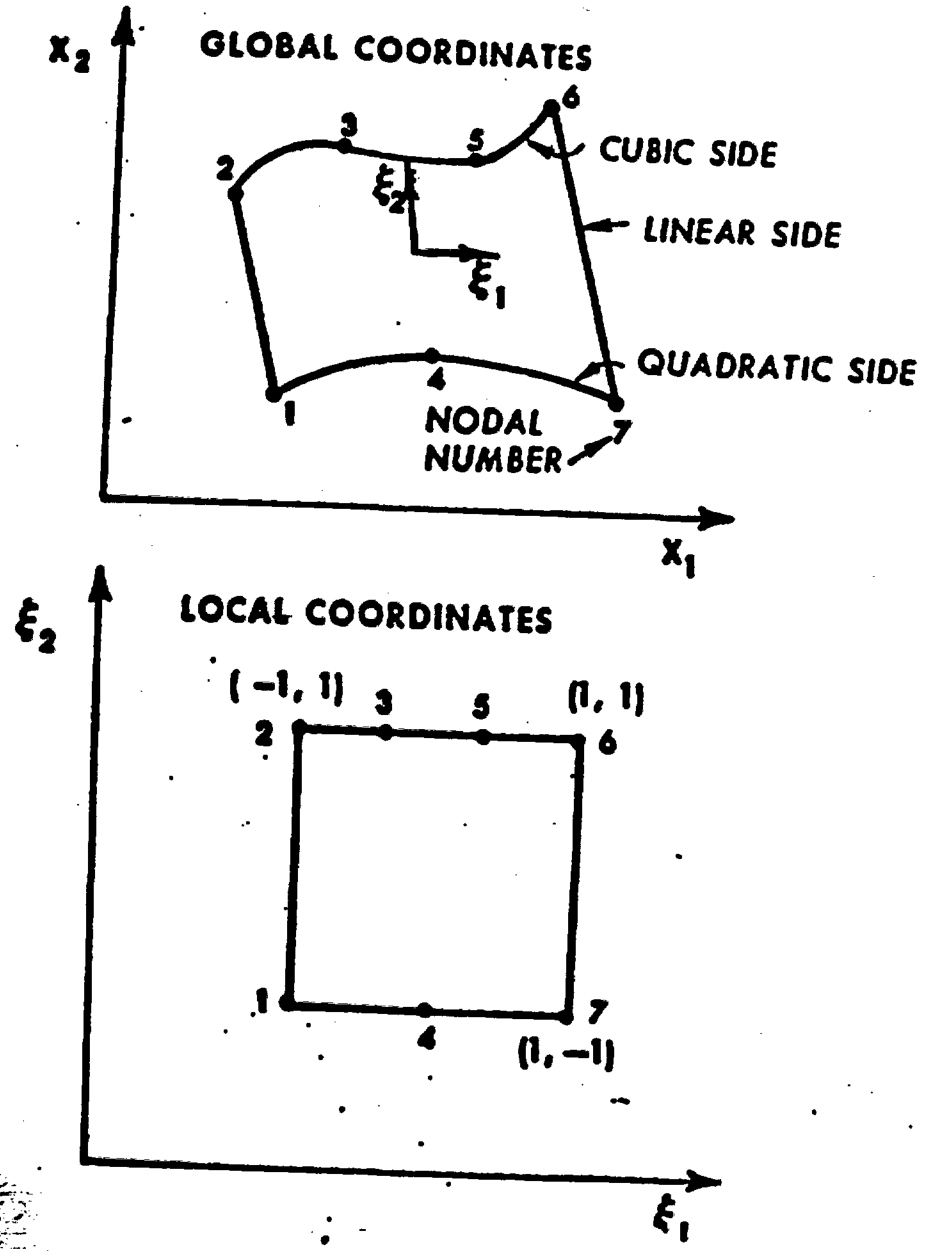


Fig. 1. Deformed isoparametric quadrilateral element in global (x,y) and local (ξ,η) co-ordinates.

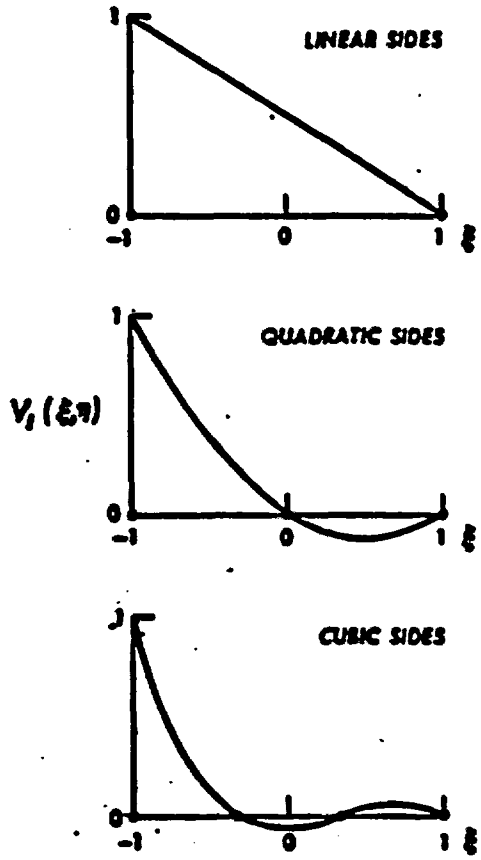


Fig. 2. Basis function along $\xi = -1$ for corner node (-1,-1) with two adjacent sides of indicated order.

2.3 Matrix Equations

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

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

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

$$A_{ij} = \iint \left(T_{xx} \frac{\partial w_i}{\partial x} \frac{\partial w_j}{\partial x} + T_{yy} \frac{\partial w_i}{\partial y} \frac{\partial w_j}{\partial y} + \left(\frac{K}{\ell'} + C\right) w_i w_j\right) dxdy \qquad (2\text{-}12a)$$

$$B_{ij} = \iint S\ w_i w_j\ dxdy \qquad (2\text{-}12b)$$

$$F_i = -\iint w_i \left(\frac{K}{\ell'}\ h_w - Q + Ch_i\right) dxdy - \int_S w_i \sum_{j=1}^{N} \left(T_{xx} \frac{\partial w_j}{\partial x} \ell_x + T_{yy} \frac{\partial w_j}{\partial y} \ell_y\right) H_j dS \qquad (2\text{-}12c)$$

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

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

or rearranging for ease of computation

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

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 $[A] + [C]/\Delta t$ 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[1] is used the major effort involved in triangularization is required only when the time step Δt 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 occuring on the same element. The difference occuring 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

[1]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

$$-\int_S \left(T_{xx} \frac{\partial \hat{h}}{\partial x} \ell_x + T_{yy} \frac{\partial \hat{h}}{\partial y} \ell_y\right) w_i \, dS = -\int_S q_n w_i \, dS \qquad (2\text{-}15)$$

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.

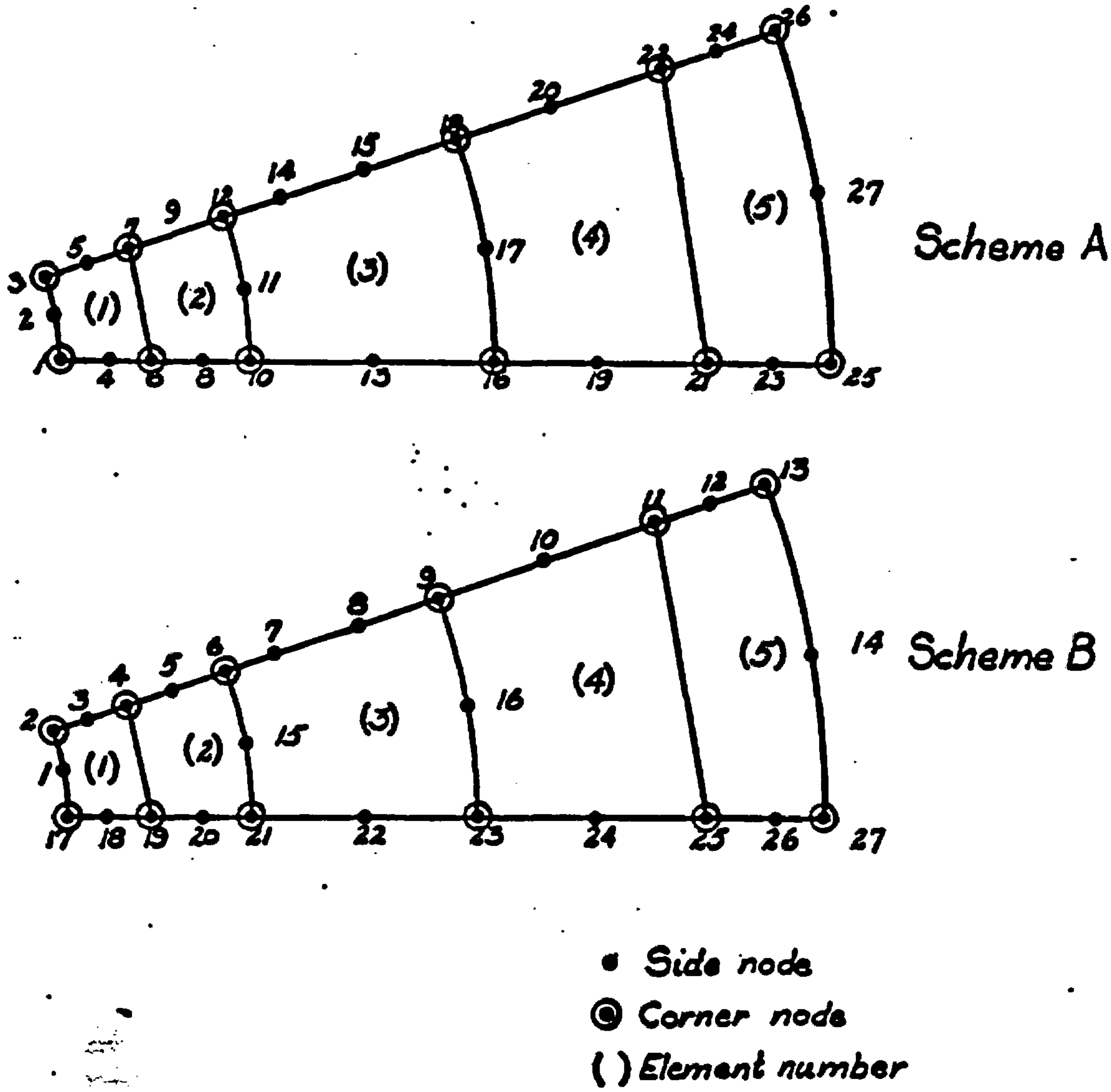


Fig. 3 Two identical element configurations with markedly different bandwidths dictated by the choice of nodal numbering scheme. Bandwidth of A is nine, of B is 19.

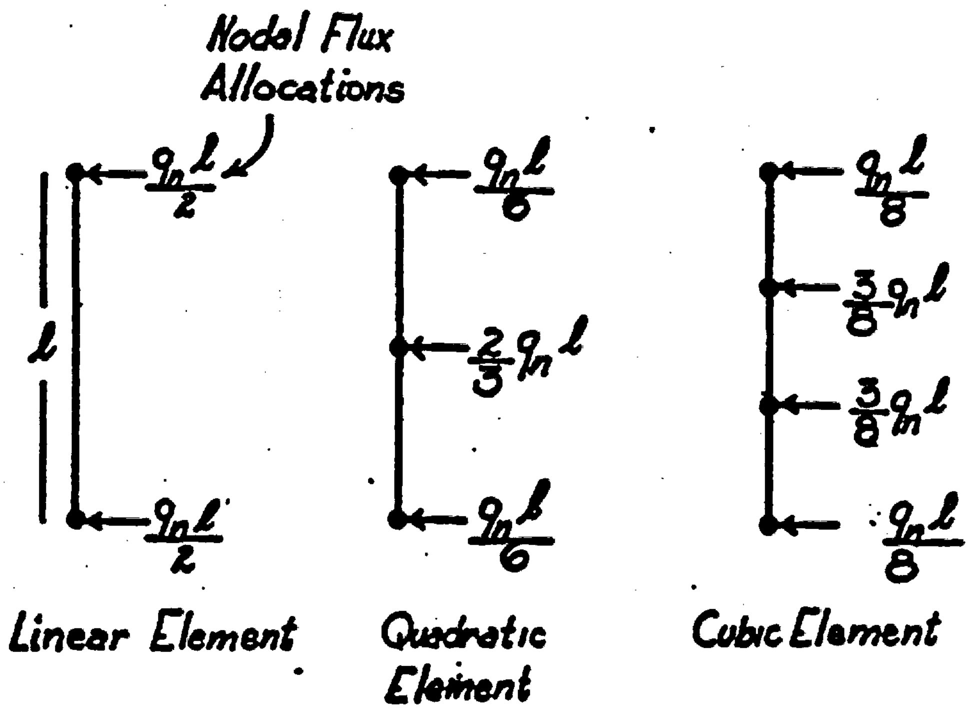


Fig. 4 Allocation to boundary nodes of a constant normal flux q_n acting along an element of side length L.

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 $\frac{K}{\ell}$, 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 (ξ,η) coordinate system. Equation 2-9 is, unfortunately written in global (x,y) coordinates and must be transformed to the (ξ,η) 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

$$\begin{Bmatrix} \dfrac{\partial w_i}{\partial x} \\ \\ \dfrac{\partial w_i}{\partial y} \end{Bmatrix} = [J^{-1}] \begin{Bmatrix} \dfrac{\partial w_i}{\partial \xi} \\ \\ \dfrac{\partial w_i}{\partial \eta} \end{Bmatrix} \tag{2-16}$$

where J is defined as

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

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

$$J = \begin{bmatrix} \frac{\partial w_1}{\partial \xi} & \frac{\partial w_2}{\partial \xi} \cdots \frac{\partial w_n}{\partial \xi} \\ \frac{\partial w_1}{\partial \eta} & \frac{\partial w_2}{\partial \eta} \cdots \frac{\partial w_n}{\partial \eta} \end{bmatrix} \qquad \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ x_n & y_n \end{bmatrix} \tag{2-18}$$

where $x_1, x_2, \ldots$ and $y_1, y_2, \ldots$ 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

$$dxdy = \det\,[J]\; d\eta d\xi. \tag{2-19}$$

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

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

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.[1]

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.[2] 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, DBANDI, and SBANDI. SHAPEI generates the particular values of the shape functions, their derivatives and the determinant in (ξ,η) 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. DBANDI places the

---

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

[2] Pinder, G. F. and E. O. Frind, Application of Galerkin's procedure to aquifer analysis, Water Resourc. 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-M) 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. Trangularization is necessary only once provided the coefficient matrices are not modified. Unfortunately the matrix B depends upon Δt and a tradeoff results between the computational advantages of changing the time step and the additional effort of upper trangularization. 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*

Variable	Format	Variable Description
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	I5	Number of nodes
NE	I5	Number of elements
NS	I5	Number of constant head nodes
NB*	I5	Global coefficient matrix half-band widtl
NF	I5	Number of source or sink nodes
NL	I5	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 Δt in hours
CHNG**	F10.0	Multiplier used to change size of time step according to relationship $\Delta t_i = CHNG * \Delta t_{i-1}$
ITMAX	I10	Maximum permitted number of time steps
ITCHNG	I10	Number of time steps between changes in Δt.

*See text for method of approximating this parameter.

**A suggested value for this parameter is 1.5.

4.6 *Data Set 5: Control Codes* *

KØD1	I5	Consider as watertable problem
KØD2	I5	Printout of element coefficient matrices
KØD3	I5	Printout of global coefficient matrices
KØD4	I5	Punchout of calculated head values at normal termination of simulation
KØD5	I5	Require steady-state solution only: matrix equation is solved only once
KØD6	I5	Printer plot of computer drawdown at each time step
KØD7	I5	Printer plot of computed head at each time step
KØD8	I5	Printout of known vector in matrix equation
KØD9	I5	Read nodal transmissivity**

4.7 *Data Set 6: Nodal Co-ordinates*

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

new card

J	I5	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 NW cards)

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

IQ	I5	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, KØD9 should read zero so that data set 9 may be omitted.

*** A G format is equivalent to an I, 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 NN/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 NN/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 NN/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
STØRAG(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 STØRAG 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*

HYCØND	F15.0	Hydraulic conductivity of confining bed
SS	F15.0	Specific storage of confining bed

new card

I	I5	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*

NPP	I5	NPP is one for head plot only, two for drawdown plot only, three for both head and drawdown plots.

new card

XLABEL	2A8	Up to 16 alphameric characters for the X-axis label
YLABEL	3A8	Up to 24 alphameric characters for the Y-axis label
TITLE	5A8	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.

new card

XMAX	G10.0	Maximum value on X axis (usually slightly greater than the maximum X-coordinate value so that nodal values appear inside plot border
XMIN	G10.0	Minimum value on X-axis (slightly less than minimum X-coordinate value)
NXS	G10.0	Number of segments on X-axis (if NXS is one, only the border lines will be plotted)
NINX	G10.0	Number of inches per X segment (if NXS is one, NINX is the number of inches between borders)
YMAX	G10.0	Maximum value on Y axis (slightly greater than maximum Y-coordinate value)
YMIN	G10.0	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 ≤12)

new card

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 significan figure plotted for drawdown (analagous 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.

CMAX	Plotted Number
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 16: 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 accomodate 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)	DDDN(NN)
P(N,NB)	UI(NN)	STORAG(NE)
H(N,NB)	FM(NN)	XX(1,NN)
IN(13,NE)	ØLD(NN)	YY(1,NN)
TRANS(NE)	FQ(NN)	CC(NN)
THK(NE)	RT(NN)	INDEX (NN)
LR(NN)	INFLØ(NE,5)	U(NN)
LRC(NN)	HZERØ(NE)	UØ(NN)
Y(NN)	PTRANS(NN)	QCØP(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 preceeding the first computed head solution is the actual half-band required as contrasted to the estimated value provided as input to the program.

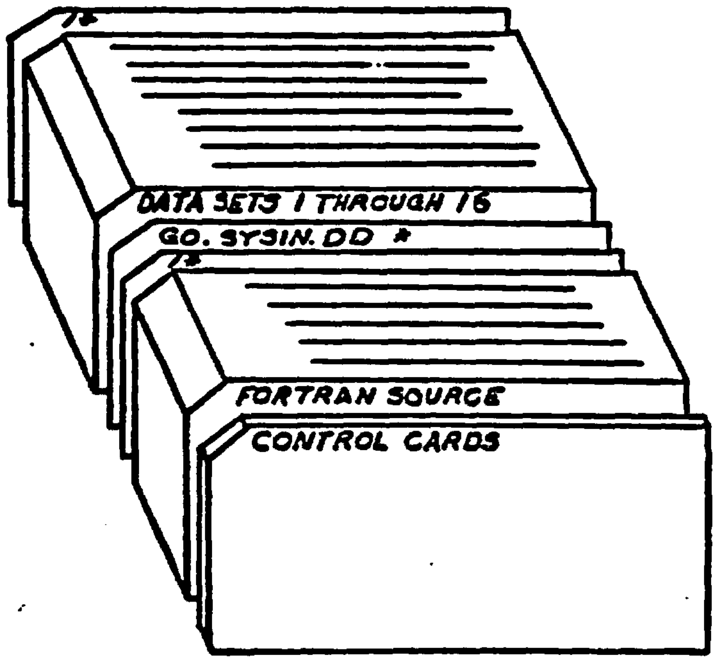


Fig. 6 Diagrammatic representation of a computer card deck for input to ISOQUAD 2.

PROGRAMMER PINDER DIVISION WRD
LOCATION PHONE

U S DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY
COMPUTER CODING FORM

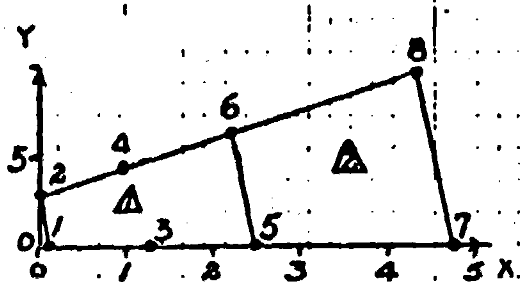
PROGRAM NO.
PROJECT

SHEET OF

PROGRAM IDENTIFICATION: ISOQUAD2
PROGRAM INFORMATION: SAMPLE CODING FORM FOR PROBLEM AS INSET

STATEMENT IDENTIFICATION

```
ISØQUAD
SAMPLE PRØBLEM FØR DØCUMENTATIØN: JANUARY 1974
    8    2    2    6    2    0
         1.        .0001     1.5       10        1
    0    1.   1.   0.   0    1    1    1    1    1
         1.
    1         0.5       0.
    2         0.0       3.0
    3         6.5       0.0
    4         4.5       4.5
    5        12.5       0.0
    6        11.0       6.5
    7        24.0       0.0
    8        21.5      10.0
    1         0.1
    2         0.1
         1.        1.        1.        1.        1.        1.        1.        1.
      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
  1      0.   .001
  2      0.   .001
  1  1$  3  5$  6$  4  2$  1$
  2  5$  7$  8$  6$  5$
  7      8
    3
   X-AXIS              Y-AXIS              SAMPLE PRØBLEM FØR DOCUMENTATIØN HEAD
   X-AXIS              Y-AXIS              SAMPLE PRØBLEM FØR DOCUMENTATIØN DDN
       25.        0.        10        1       11.       0.        5         1
         8        10.       10.
XXXX
```



FORTRAN CODING FORM

```
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.F. 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 ***********************************************************************
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 *   DGX......X DERIVATIVE OF BASIS FUNCTION FOR ALL ACTIVE NODES IN   *
C *            ELEMENT EVALUATED AT XI,YI                               *
C *   DGY......Y DERIVATIVE EQUIVALENT OF DGX                           *
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 *   FF.......BASIS FUNCTION FOR ALL ACTIVE NODES IN ELEMENT, EVALUATED*
C *            AT XI,YI                                                 *
C *   FQ.......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 *   IFLO.....LOCATION OF CONSTANT HEAD NODES                          *
C *   IN.......ELEMENT INCIDENCE                                        *
C *   ISTEDY...TEST FOR STEADY-STATE LEAKAGE                            *
C *   IT.......NUMBER OF TIME STEPS CALCULATED                          *
C *   ITCHNG...NUMBER OF TIME STEPS BETWEEN CHANGES IN DELT             *
C *   ITMAX....MAXIMUM PERMITTED NUMBER OF TIME STEPS                   *
C *   JBWH.....HALFBANDWIDTH (INCLUDES DIAGONAL)                        *
C *   JD.......ARRAY OF ACTIVE NODES IN ELEMENT                         *
C *   JNDF.....DEGREES OF FREEDOM (NODES LESS CONSTANT HEAD NODES)      *
```

```
C *   JNEL.....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......CUMMULATIVE 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 *   MBI......CALCULATED HALF-BAND WIDTH                                    *    0
C *   MINX.....NUMBER OF INCHES PER X SEGMENT OF PLOT                        *    0
C *   MINY.....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*
C *   PE.......ELEMENT MATRIX ASSOCIATED WITH TIME DERIVATIVE                *
C *   PTRANS...NODAL TRANSMISSIVITY                                          *
C *   PUMP.....SUM OF SOURCES AND SINKS                                      *
C *   QCOEF....TRANSIENT LEAKAGE COEFFICIENT                                 *
C *   QCOF.....ARRAY OF TRANSIENT LEAKAGE COEFFICIENTS                       *
C *   RT.......VECTOR CONTAINING SOURCES, SINKS, AND BOUNDARY CONDITIONS*
C *   S........GLOBAL COEFFICIENT MATRIX ASSOCIATED WITH SPACE DERIVATIV*
C *   SE.......ELEMENT MATRIX ASSOCIATED WITH SPACE DERIVATIVE               *
C *   SMIN.....ELAPSED TIME IN MINUTES                                       *
C *   SRCR.....CONTRIBUTION OF ELEMENT TO KNOWN VECTOR                       *
C *   SS.......SPECIFIC STORAGE OF CONFINING BED                             *
C *   SSEC.....ELAPSED TIME IN SECONDS                                       *
C *   SSEC1....ELAPSED TIME IN SECONDS                                       *
```

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

```
     1                    U,OLD,FQ,RT,IFLO,HZERO,PTRANS,DDDN,STORAG,XX,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),FQ(JNND),RT(JNND),
     3           HZERO(JNEL),PTRANS(JNND),DDDN(JNND),STORAG(JNEL),
     4           XX(1,JNND),YY(1,JNND),CC(JNND),INDEX(JNND),UO(JNND),
     5           QCOF(JNND),IFLO(JNEL,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),F(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),SL(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/1H*/, DIGIT/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,
     1     1H8,1H9/, START/4HISOQ/, EXIT/4HXXXX/, 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     QCOEF/0./
C     =================================================================
C
C.....READ CONTROL DATA
   10 READ (5,780) CARD
      IF (CARD.EQ.EXIT) STOP
      IF (CARD.NE.START) GO TO 10
      READ (5,780) TITLE
C
      READ (5,790) NN,NE,NS,NB,NF,NL
      WRITE (6,800)
      WRITE (6,810)
      WRITE (6,820) TITLE
      WRITE (6,1060)
      WRITE (6,830) NN,NE,NS,NF,NB,NL
C
      READ (5,840) TIME,DELT,CHNG,ITMAX,ITCHNG
      WRITE (6,850) TIME,DELT,CHNG,ITMAX,ITCHNG
      DELT=DELT*3600.
C
C.....INITIALIZE CONSTANTS
```

```
      NDS=1
      STIME=0.
      IT=0
      SSEC=0.
      IFINAL=0
      ISTEDY=0
      H11=H1*H1
      H12=H1*H2
      H22=H2*H2
C
C.....READ OPTION CODES AND PRINT OPTIONS
      READ (5,790) KOD1,KOD2,KOD3,KOD4,KOD5,KOD6,KOD7,KOD8,KOD9
      WRITE (6,1260)
      IF (KOD1) 1270,1270,1280
 1270 WRITE (6,1512)
      GO TO 1290
 1280 WRITE (6,1522)
 1290 IF (KOD2) 1300,1300,1310
 1300 WRITE (6,1301)
      GO TO 1320
 1310 WRITE (6,1311)
 1320 IF (KOD3) 1330,1330,1340
 1330 WRITE (6,1331)
      GO TO 1350
 1340 WRITE (6,1341)
 1350 IF (KOD4) 1360,1360,1370
 1360 WRITE (6,1361)
      GO TO 1380
 1370 WRITE (6,1371)
 1380 IF (KOD5) 1390,1390,1400
 1390 WRITE (6,1391)
      GO TO 1410
 1400 WRITE (6,1401)
 1410 IF (KOD6) 1420,1420,1430
 1420 WRITE (6,1421)
      GO TO 1440
 1430 WRITE (6,1431)
 1440 IF (KOD7) 1450,1450,1460
 1450 WRITE (6,1451)
      GO TO 1470
 1460 WRITE (6,1461)
 1470 IF (KOD8) 1480,1480,1490
 1480 WRITE (6,1481)
      GO TO 1500
 1490 WRITE (6,1491)
 1500 IF (KOD9) 1510,1510,1520
 1510 WRITE (6,1511)
      GO TO 1531
```

```
 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 (NF.EQ.0) GO TO 40
      DO 30 I=1,NF
      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 NODAL TRANSMISSIVITY
      IF (KOD3.EQ.0) GO TO 1422
      READ (5,920) (PTRANS(I),I=1,NN)
      WRITE (6,931)
      WRITE (6,940) (I,PTRANS(I),I=1,NN)
      IF (KOD1.EQ.0) GO TO 1422
      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
```

```
     2VALUE,5X))
 1422 IF(KODS.NE.1) GO TO 47
C
C.....READ ELEMENT TRANSMISSIVITY AND STORAGE
      READ (5,952) (L,TRANS(L),STORAG(L),K=1,NE)
      GO TO 48
   47 READ (5,950) (L,TRANS(L),STORAG(L),K=1,NE)
C
C.....READ AND ADJUST ELEMENT INCIDENCES
C.....FREE FORMAT FOR INPUT
   48 DO 50 I=1,13
      DO 50 L=1,NE
   50 IN(I,L)=0
      DO 150 LL=1,NE
      READ (5,951) L, CHAR
      KC=0
      KE=4
      KS=2
      J=0
   60 NUM=0
   70 J=J+1
      IF (CHAR(J).EQ.BLANK) GO TO 70
C.....DETERMINE INCIDENCE FROM CHARACTER LIST AND ADJUST FOR POWER OF
C     TEN.
   80 I=0
   90 I=I+1
      IF (I.LE.10) GO TO 100
      WRITE (6,960) CHAR(J),L
      GO TO 10
  100 IF (CHAR(J).NE.DIGIT(I)) GO TO 90
      NUM=10*NUM+I-1
      J=J+1
C
C.....CHECK NEXT LOCATION - IF IT IS A DOLLAR SIGN THIS IS A CORNER NODE
C     IF IT IS A BLANK THIS IS LAST DIGIT OF A SIDE NODE .....
      IF (CHAR(J).EQ.STAR) GO TO 110
      IF (CHAR(J).EQ.BLANK) GO TO 120
      GO TO 80
  110 KC=KC+1
C.....WHEN KC, THE CORNER COUNTER,IS FIVE THE INCIDENCE LIST FOR
C     ELEMENT L IS COMPLETE.
C.....WHEN A NODE IS ABSENT FROM A POSSIBLE LOCATION,ZERO IS RECORDED IN
C     THE IN ARRAY.
      IF (KC.EQ.5) GO TO 130
      IN(KC,L)=NUM
      IF (KS.EQ.0) KE=KE+2
      IF (KS.EQ.1) KE=KE+1
      KS=0
```

```
      GO TO 60
  120 KE=KE+1
      KS=KS+1
      IN(KE,L)=NUM
      IF (J.LE.65) GO TO 60
      WRITE (6,970) L
      GO TO 10
  130 M=4
C
C.....COUNT ACTIVE NODES.  IE THOSE NOT EQUAL TO ZERO.
      DO 140 I=5,12
      IF (IN(I,L).EQ.0) GO TO 140
      M=M+1
  140 CONTINUE
      IN(13,L)=M
  150 CONTINUE
      WRITE (6,980)
      WRITE (6,990)
      DO 160 L=1,NE
  160 WRITE (6,1000) L,(IN(I,L),I=1,12),STORAG(L),TRANS(L)
C
C.....READ CONSTANT HEAD BOUNDARY NODES (INDICATED BY LR=1)
      WRITE (6,1010)
      DO 170 I=1,NN
  170 LR(I)=0.
      IF (NS.EQ.0) GO TO 210
  180 READ (5,1020)  (LRT(ITT),ITT=1,20)
      IA=0
      DO 190 I=1,20
      IF (LRT(I).EQ.0) GO TO 200
      IA=I
      J=LRT(I)
      NST=NST+1
      IF (J.LE.NN) GO TO 190
      WRITE (6,1030) J
      GO TO 770
  190 LR(J)=1
  200 WRITE (6,1040)  (LRT(IIT),IIT=1,IA)
      IF (IA.EQ.20) GO TO 180
      IF (NST.NE.NS) WRITE (6,1050) NST,NS
C
C.....SUM CONSTANT BOUNDARY NODES
  210 LRC(1)=LR(1)
      DO 220 J=2,NN
  220 LRC(J)=LRC(J-1)+LR(J)
      NR=LRC(NN)
      N=NN-NR
      WRITE (6,1060)
```

```
      WRITE (6,1070) NN,NE,N
C
C
C.....READ LEAKY ELEMENT COEFFICIENTS
      DO 243 I=1,NE
      HZERO(I)=0.
  243 THK(I)=0.
      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),(TATLE(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),(TATLE(2,J),J=
     11,5)
  364 READ(5,366)XMAX,XMIN,NXS,NINX,YMAX,YMIN,NYS,NINY,KK(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)
      DDDN(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)=-FQ(I)
  992 CONTINUE
C
C.....INITIALIZE CONSTANTS
      NB1=0
      LEKCHK=0
      THKZRO = 0.
      KEY=0
  254 TSTOR=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)=IN(K,L)
  430 CONTINUE
C.....THE ARRAY JD NOW CONTAINS THE INCIDENCES OF THE ACTIVE NODES IN
C     ELEMENT L.
C
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 (M.GT.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=AG(I)
      GO TO 290
  280 XI=AG(J+2)
      YI=AG(I+2)
```

```
  290 CONTINUE
C      ****************************************
       CALL SHAPE (L,M,XI,YI,DET)
C      ****************************************
       DO 300 JJ=1,M
       P(JJ,K)=FF(JJ)
       DX(JJ,K)=DGX(JJ)
  300  DY(JJ,K)=DGY(JJ)
       DETJ(K)=DET
  310  CONTINUE
C.....SELECT PARAMETER VALUES
       TR=TRANS(L)
       QCOEF2=QCOF(L)
       DO 2392 K=1,NP2
       UIK(K)=0.
       UK(K)=0.
        UKO(K)=0.
       TK(K)=0.
       DO 2392 J=1,M
       JDJ=JD(J)
       UIK(K)=UIK(K)+UI(JDJ)*P(J,K)
       UK(K)=UK(K)+U(JDJ)*P(J,K)
       UKO(K)=UKO(K)+UO(JDJ)*P(J,K)
     — IF (KOD1.EQ.0) GO TO 292
       TK(K)=TK(K)+PTRANS(JDJ)*(DDDN(JDJ)-BASE(JDJ))*P(J,K)
       GO TO 2392
  292  IF(TRANS(L).EQ.0.) TK(K)=TK(K)+PTRANS(JDJ)*P(J,K)
 2392  CONTINUE
C
C.....CALCULATE COEFFICIENTS FOR ELEMENT MATRICES
       IF (THK(L).EQ.THKZRO) GO TO 7002
       QCOEF = 0.
       SSEC1=SSEC+DELT
       IF(THK(L).LE.0.) GO TO 7002
C      ******************************************
       CALL CLAY (QCOEF,THK(L),SSEC1,HYCOND,SS)
C      ******************************************
       CHKLEK=HYCOND/(THK(L)*QCOEF)
       IF(CHKLEK.LT..99) LEKCHK=1
 7002  THKZRO = THK(L)
       QCOF(L)=QCOEF
       ST=STORAG(L)
       DO 350 I=1,M
       DO 350 J=I,M
       DO 320 K=1,NP2
       SRCLT(K) = QCOEF*P(I,K)*DETJ(K)*P(J,K)
       QP(K)=P(I,K)*ST*P(J,K)*DETJ(K)
       IF (TRANS(L).EQ.0.) TR=TK(K)
```

```
  320 Q(K)=(Z2*(DX(I,K)*DX(J,K)+DY(I,K)*DY(J,K)))*DETJ(K)
      IF (NP.EQ.4) GO TO 330
      SE(I,J)=Q(1)+Q(2)+Q(3)+Q(4)
      PE(I,J)=QP(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(9)+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
      SRCE(I)=0.
      GO TO 341
  351 DO 323 K=1,NP2
  323 SRCRT(K)=(QCOEF*UIK(K)    +HYCOND/THK(L)*(HZERO(L)-UIK(K))
     1)*F(I,K)*DETJ(K)
      IF(NP.EQ.4) GO TO 331
      SRCE(I)=SRCRT(1)+SRCRT(2)+SRCRT(3)+SRCRT(4)
      GO TO 341
  331 SRCE(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 (KOD2.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)
```

```
      WRITE (6,1120) L
      MM2=(M+7)/8*8-7
      DO 400 K1=1,MM2,8
      K2=K1+7
      IF (K1.EQ.MM2) K2=M
      DO 390 I=1,M
  390 WRITE (6,1110) I,(PE(I,J),J=K1,K2)
  400 WRITE (6,1170)
  410 CONTINUE
C
C.....THIS COMPLETES THE ELEMENT STIFFNESS MATRIX
C     ----------------------------------------------------------------
C
C.....ASSEMBLY OF GLOBAL COEFFICIENT MATRICES
      DO 480 I=1,M
      JDI=JD(I)
      IF (LR(JDI).GT.0) GO TO 460
      IR=JDI-LRC(JDI)
      RT(IR)=RT(IR)-SRCR(I)
      DO 450 J=1,M
      JDJ=JD(J)
      IF (LR(JDJ).GT.0) GO TO 450
      JC=JDJ-LRC(JDJ)-IR+1
      IF (JC.LE.NB) GO TO 440
      KBX=1
      WRITE (6,1130) L,JC,NB
      GO TO 450
  440 IF (JC.LT.1) GO TO 450
      P(IR,JC)=P(IR,JC)+PE(I,J)
      S(IR,JC)=S(IR,JC)+SE(I,J)
      IF (JC.GT.NB1) NB1=JC
  450 CONTINUE
      GO TO 480
  460 CONTINUE
      DO 470 J=1,M
      JDJ=JD(J)
      IF (LR(JDJ).GT.0) GO TO 470
      JC=JDJ-LRC(JDJ)
      RT(JC)=RT(JC)+SE(I,J)*UI(JDI)
  470 CONTINUE
  480 CONTINUE
  411 IF (L.LT.NE) GO TO 270
C.....CALCULATIONS FOR GLOBAL COEFFICIENT MATRIX COMPLETE
C     ================================================================
C
C
      DO 412 I=1,NN
  412 UO(I)=U(I)
```

```
      NB=NB1
      WRITE (6,1140) NB
      IF(KODE.EQ.2) GO TO 525
      IF (KODE.NE.1) GO TO 760
      WRITE (6,1150)
      NB2=(NB+9)/10*10-9
      DO 500 K1=1,NB2,10
      K2=K1+9
      IF (K1.EQ.NB2) K2=NB
      DO 490 I=1,N
  490 WRITE (6,1160) I,(S(I,J),J=K1,K2)
  500 WRITE (6,1170)
      WRITE (6,1180)
      NB2=(NB+9)/10*10-9
      DO 520 K1=1,NB2,10
      K2=K1+9
      IF (K1.EQ.NB2) K2=NB
      DO 510 I=1,N
  510 WRITE (6,1160) I,(P(I,J),J=K1,K2)
  520 WRITE (6,1170)
  525 WRITE (6,1190)
      WRITE (6,1200) (RT(I),I=1,N)
C
C.....SAVE MATRIX A
  760 DO 530 I=1,N
      DO 530 J=1,NB1
  530 H(I,J)=S(I,J)
C
C.....BEGIN TIME LOOP
  540 A3=1./DELT
C
C.....COMPUTE COMPLETE COEFFICIENT MATRIX
      DO 550 I=1,N
      DO 550 J=1,NB1
  550 S(I,J)=H(I,J)+P(I,J)*A3
C
C.....DECOMPOSE
C     **********************
      CA[illegible]D (N,NB,IEX)
C     **********************
C
C.....CALCULATE VECTOR OF KNOWN VALUES
  560 DO 570 I=1,N
  570 FM(I)=-RT(I)
      N1=N+1
      N2=N-1
      FM(N)=FM(N)+P(N,1)*A3*OLD(N)
      DO 580 I=1,N2
```

```
      L1=MIN0(NB,M1-I)
      FM(I)=FM(I)+P(I,1)*A3*OLD(I)
      DO 580 J=2,L1
      LM=I+J-1
      FM(I)=FM(I)+P(I,J)*A3*OLD(LM)
  580 FM(LM)=FM(LM)+P(I,J)*A3*OLD(I)
      IF (KOD4.EQ.0) GO TO 581
      WRITE (6,582)
      WRITE (6,1200)  (FM(J),J=1,N)
  581 CONTINUE
C
C.....BACK SUBSTITUTE
C     *****************
      CALL SBAND (N,NB)
C     *****************
      IT=IT+1
      STIME=STIME+DELT/3600.
      SSEC=SSEC+DELT
      SMIN=SSEC/60.
C
C.....CALCULATE NEW HEADS
      DO 590 I=1,N
  590 OLD(I)=U(I)
      IP=0
      DO 610 I=1,NN
      IF (LE(I).GT.0) GO TO 600
      IP=IP+1
      U(I)=OLD(IP)
      GO TO 610
  600 U(I)=UI(I)
  610 CONTINUE
      DELT1=DELT/3600.
      WRITE (6,1170)
      WRITE (6,1170)
      WRITE (6,1170)
      WRITE (6,1210)  STIME,SMIN,SSEC,IT,DELT1
      WRITE (6,940)  (I,U(I),I=1,NN)
C.....ADJUST FOR WATERTABLE TRANSMISSIVITY
      IF (KOD1.EQ.0) GO TO 1594
      DO 1592 I=1,NN
      DDDN(I)=2.*U(I)-UO(I)
 1592 IF (DDDN(I).LT.0.) DDDN(I)=0.
 1594 CONTINUE
      IF(KOD5.NE.1) GO TO 614
C
C.....PRINT DRAWDOWN MAP
      DO 612 I=1,NN
  612 CC(I)=(UI(INDEX(I))-U(INDEX(I)))*ADDN
```

```
      NPP=2
C     **********
      CALL GRAPH
C     **********
  614 IF(KOD7.NE.1) GO TO 615
C
C.....PRINT MAP OF COMPUTED HEAD VALUES
      DO 616 I=1,NN
  616 CC(I)=U(INDEX(I))*CMAX
      NPP=1
C     **********
      CALL GRAPH
C     **********
  615 CONTINUE
C
C
C.....ELAPSED TIME CHECK
      DELTP=DELT
      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(KOD1.NE.0) GO TO 254
      IF(MOD(IT,ITCHNG).GT.0) GO TO 560
      GO TO 540
  770 IF (KOD4.NE.1) GO TO 771
      WRITE (7,1051) (U(I),I=1,NN)
  771 IF(KOD1.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(I3,4X,1PE9.2,2X,1PE9.2)))
  365 FORMAT(10A8)
  366 FORMAT(8G10.0)
  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,
     118HTRANSIENT LEAKAGE=,G15.5/11X,14HCONSTANT HEAD=,G19.5/11X,
     215HSUM OF SOURCES=,G18.5/11X,13HPUMPING RATE=,G20.5/11X
```

```
   3,11HDIFFERENCE=,G22.5)
 582 FORMAT (1H0,10X,15HKNOWN VECTOR FB/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,15HI S 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,F30.2/11X,26HINITIAL TIME STEP IN HOURSF30.6/11X,
   235HMULTIPLIER FOR INCREASING TIME STEPF21.3/11X,38HMAXIMUM PERMITT
   3ED NUMBER OF TIME STEPSI18/11X,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
   1NODE7X,14HDISCHARGE(CFS)/)
 910 FORMAT (11X,I4,F16.3)
 920 FORMAT (3F10.0)
 921 FORMAT (5F15.0)
 930 FORMAT (/////,11X,12HINITIAL HEAD/11X,12(1H-)//11X,6(4HNODE,5X,5HV
   1ALUE,5X))
 931 FORMAT (/////,11X,20HNODAL TRANSMISSIVITY/11X,20(1H-)//11X,6(4HNOD
   1E,5X,5HVALUE,5X))
 940 FORMAT (/(11X,6(I4,2X,1PE10.3,3X)))
 950 FORMAT (I3,2F6.0)
 951 FORMAT (I3,65A1)
 952 FORMAT (I3,F6.0,6X,F6.0)
 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*),18HCONSTANT HEAD NODEI4,37HDOES NOT EXIST - EX
    1ECUTION TERMINATED10(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 (1H ,10X,21HTOTAL NUMBER OF NODES,I9/11X,19HCONSTANT HEAD N
    1ODES,1X,I10/11X,18HDEGREES OF FREEDOM,I12///)
1100 FORMAT (////,10X,7HELEMENT,I4,5X,16HSTIFFNESS MATRIX/)
1110 FORMAT (I5,8E15.6)
1120 FORMAT (1H ,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,15HKODI 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 SHAPEI (JNEL,JNND,JBWH,JNDF,S,P,IN,U,X,Y,DGX,DGY,F)
      DIMENSION IN(13,JNEL),S(JNDF,JBWH),P(JNND),X(JNND),Y(JNND),U(JNND)
     1,ALP(4),DAX(4),DAY(4), BTX(4), BTY(4), DBX(4), DFX(12), DFY(12), D
     2GX(12), DGY(12), F(12), DBY(4)
      COMMON /WEIGHT/ SUM1,SUM2,SUM3,SUM4
      INTEGER*2 IN
      RETURN
      ENTRY      SHAPE (L,M,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
      XQ1=XI-.5
      XQ2=-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 (IN(J3,L).EQ.0) GO TO 10
      IF (IN(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 (IN(J3,L).EQ.0) GO TO 60
      IF (IN(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 90
   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
```

```
      DBY(J1)=YC2
      DBY(J4)=YC2
   90 CONTINUE
      J1=1
      J2=4
      J3=11
  100 CONTINUE
C
C     SHAPE FUNCTION DERIVATIVE MATRIX - CORNER NODES
      DO 110 J=1,4
      DFX(J)=DAX(J)*(BTX(J)+BTY(J))+DBX(J)*ALF(J)
      DFY(J)=DAY(J)*(BTX(J)+BTY(J))+DBY(J)*ALF(J)
      F(J)=ALF(J)*(BTX(J)+BTY(J))
  110 CONTINUE
C
C     SHAPE FUNCTION DERIVATIVE MATRIX - EDGE NODES
      IF (M.EQ.4) GO TO 240
      J=4
      XEQ=1.-XI*XI
      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
      DFX(J)=-XI*YI1
      DFY(J)=-.5*XEQ
      F(J)=.5*XEQ*YI1
      GO TO 140
  130 J=J+1
      DFX(J)=-.28125*YI1*(3.*XEQ+2.*XI*XE1)
      DFY(J)=-.28125*XEQ*XE1
      F(J)=.28125*XEQ*XE1*YI1
      J=J+1
      DFX(J)=.28125*YI1*(3.*XEQ-2.*XI*XE2)
      DFY(J)=-.28125*XEQ*XE2
      F(J)=.28125*XEQ*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
      DFX(J)=.5*YEQ
      DFY(J)=-YI*XI2
      F(J)=.5*XI2*YEQ
      GO TO 170
```

```
  160 J=J+1
      DFX(J)=.28125*YEQ*YE1
      DFY(J)=-.28125*XI2*(3.*YEQ+2.*YI*YE1)
      F(J)=.28125*XI2*YEQ*YE1
      J=J+1
      DFX(J)=.28125*YEQ*YE2
      DFY(J)=.28125*XI2*(3.*YEQ-2.*YI*YE2)
      F(J)=.28125*XI2*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
      DFX(J)=-XI*YI2
      DFY(J)=.5*XEQ
      F(J)=.5*XEQ*YI2
      GO TO 200
  190 J=J+1
      DFX(J)=.28125*YI2*(3.*XEQ-2.*XI*XE2)
      DFY(J)=.28125*XEQ*XE2
      F(J)=.28125*XEQ*XE2*YI2
      J=J+1
      DFX(J)=-.28125*YI2*(3.*XEQ+2.*XI*XE1)
      F(J)=.28125*XEQ*XE1*YI2
      DFY(J)=.28125*XEQ*XE1
  200 IF (IN(11,L).EQ.0) GO TO 230
      IF (IN(12,L).EQ.0) GO TO 210
      GO TO 220
  210 J=J+1
      DFX(J)=-.5*YEQ
      DFY(J)=-YI*XI1
      F(J)=.5*XI1*YEQ
      GO TO 230
  220 J=J+1
      DFX(J)=-.28125*YEQ*YE2
      DFY(J)=.28125*XI1*(3.*YEQ-2.*YI*YE2)
      F(J)=.28125*XI1*YEQ*YE2
      J=J+1
      DFX(J)=-.28125*YEQ*YE1
      DFY(J)=-.28125*XI1*(3.*YEQ+2.*YI*YE1)
      F(J)=.28125*YEQ*YE1*XI1
  230 CONTINUE
  240 CONTINUE
C
C     JACOBIAN
      SUM1=0.
      SUM2=0.
      SUM3=0.
      SUM4=0.
```

```
      K=0
      DO 260 I=1,N
  250 K=K+1
      IF (IN(K,L).EQ.0) GO TO 250
      KI=IN(K,L)
      SUM1=SUM1+DPX(I)*X(KI)
      SUM2=SUM2+DPX(I)*Y(KI)
      SUM3=SUM3+DPY(I)*X(KI)
      SUM4=SUM4+DPY(I)*Y(KI)
  260 CONTINUE
      DET=SUM1*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,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     DBAND (N,NB,IEX)
      IEX=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-S(II,K+1)*S(II,JZ)
   20 IF (J.NE.1) GO TO 40
      IF (SUM.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.GT.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,          XMAX,XMIN,YMAX,YMIN,NXS,NINX,NYS,NINY,NDS, K(10)
      REAL*8 XLABEL,YLABEL,TITLE,Z,XN1
      REAL PANT(122),SYM(17),NX(50),NY(14)
```

```
      INTEGER DIGIT(82),VP1(6),VP2(6),VP3(7)
      INTEGER        N(10),BLANK(60)
      DATA SYM/'1','2','3','4','5','6','7','8','9','0',' ','.','Y','*',
     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'
     2,'27','28','29','30','31','32','33','34','35','36','37','38','39',
     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','
     2104','105','106','107','108','109','110','111','112','113','114','
     3115','116','117','118','119','120','121','122'/
      DATA VP1/'(1H ',',',','     ','A1,F','10.2',')'/
      DATA VP2/'(1H ',',',','     ','A1,1','X,A8',')'/
      DATA VP3/'(1H0',',',','     ','A1,F','3.0,','12F1','0.2)'/
C
C     ---INITIALIZE VARIABLES---
      NXD=NXS*NINX
      NYD=NYS*NINY
      XSP=(XMAX-XMIN)/NXD
      YSP=(YMAX-YMIN)/NYD
      N4=NXD*N1+1
      N5=NXD+1
      N6=NYD+1
      N7=N1*NINX
      N8=N2*NYD+1
      N9=N2*NINY
      NR=N3-1
      NA=N4/2-2
      NB=N4/2+4
      NC=(N3-N8-10)/2
      ND=NC+N8
      NE=MAX0(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=I,NN
      IF(X(L,J).GT.BIG) GO TO 15
      GO TO 20
   15 BIG=X(L,J)
      KK=J
```

```
   20 CONTINUE
      TEMPI=Y(L,I)
      TEMPII=X(L,I)
      TEMP3=INDEX(I)
      Y(L,I)=Y(L,KK)
      X(L,I)=X(L,KK)
      INDEX(I)=INDEX(KK)
      INDEX(KK)=TEMP3
      Y(L,KK)=TEMPI
   30 X(L,KK)=TEMPII
   90 CONTINUE
C
C     ---COMPUTE NUMBERS FOR X AND Y AXES---
      DO 100 I=1,NE
      NNX=N5-I
      NNY=N6-I
      IF(NNY.LT.0) GO TO 95
      NY(I)=YSP*NNY+YMIN
   95 IF(NNX.LT.0) GO TO 100
      NX(I)=XSP*NNX+XMIN
  100 CONTINUE
      RETURN
C
C     -----------
      ENTRY GRAPH
C     -----------
C     ---INITIALIZE VARIABLES---
      Z=XMAX
      WRITE (6,40)
      DO 10 I=1,NDS
   10 N(I)=1
      DO 210 I=1,N4
C
C     ---LOCATE X AXES---
      IF (I.EQ.1.OR.I.EQ.N4) GO TO 110
      DO 114 J=1,N8,N9
  114 PRNT(J)=SYM(15)
C
C     ---LOCATE Y AXES---
      IF ((I-1)/N1*N1.NE.I-1) GO TO 117
  115 PRNT(1)=SYM(14)
      PRNT(N8)=SYM(14)
  117 IF((I-1)/N7*N7.NE.I-1) GO TO 130
      DO 118 J=2,NR
      IF((J-1)/N9*N9.EQ.J-1)PRNT(J)=SYM(17)
  118 IF((J-1)/N9*N9.NE.J-1)PRNT(J)=SYM(16)
      GO TO 130
  110 DO 120 J=1,N8
```

```
      IF ((J-1)/N2*N2.EQ.J-1) PRNT(J)=SYM(14)
  120 IF ((J-1)/N2*N2.NE.J-1) PRNT(J)=SYM(16)
C
C     ---COMPUTE LOCATION OF POINTS---
  130 DO 150 J=1,NDS
  135 IF (N(J).EQ.K(J)+1) GO TO 150
      IF(I.GT.1) GO TO 137
      IF(X(J,N(J)).LE.Z+XN1*XSP) GO TO 137
      N(J)=N(J)+1
      GO TO 135
  137 IF (X(J,N(J)).LE.Z+XN1*XSP.AND.X(J,N(J)).GE.Z-XN1*XSP) GO TO 140
      GO TO 150
  140 M=NR+0.5-    ((Y(J,N(J))-YMIN)*N2)/YSP
      IF(M.LT.0.OR.M.GT.NR) GO TO 145
      IF(CC(N(J)))142,146,147
  142 IF(M.NE.0) PRNT(M)=SYM(16)
      NUM=(-CC(N(J))+.005)*10.
      GO TO 141
  147 NUM=(CC(N(J))+0.005)*100.
      IF (NUM.GT.999) NUM=MCD(NUM,1000)
  141 IF(NUM.LT.100) GO TO 143
      INDX3=NUM/100
      IF (M.NE.0.AND.CC(N(J)).GT.0.) PRNT(M)=SYM(INDX3)
      NUM=NUM-INDX3*100
  143 INDX1=MOD(NUM,10)
      IF(INDX1.EQ.0) INDX1=10
      INDX2=NUM/10
      IF(INDX2.EQ.0) INDX2=10
      GO TO 144
  146 INDX1=14
      INDX2=14
  144 PRNT(M+1)=SYM(INDX2)
      PRNT(M+2)=SYM(INDX1)
  145 N(J)=N(J)+1
      IF (N(J).EQ.K(J)+1) GO TO 150
      IF (X(J,N(J)).LE.Z+XN1*XSP.AND.X(J,N(J)).GE.Z-XN1*XSP) GO TO 140
  150 CONTINUE
C
C     ---PRINT AXES,LABELS, AND POINTS---
      IF (I-NA.EQ.0) GO TO 170
      IF (I-NB.EQ.0) GO TO 180
      IF ((I-1)/N1*N1-(I-1)) 190,160,190
  160 WRITE (5,VF1) (BLANK(J),J=1,NC), (PRNT(J),J=1,N8),NY(1+(I-1)/6)
      GO TO 200
  170 WRITE (5,VF2) (BLANK(J),J=1,NC), (PRNT(J),J=1,N8),XLABEL(NP,1)
      GO TO 200
  180 WRITE (5,VF2) (BLANK(J),J=1,NC), (PRNT(J),J=1,N8),XLABEL(NP,2)
      GO TO 200
```

```
  190 WRITE (6,VF2) (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*XSF
      DO 210 J=1,N8
  210 PRNT(J)=SYM(11)
C
C     ---NUMBER AND LABEL Y AXIS AND PRINT TITLE---
      WRITE (6,VF3) (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,'X',19X,'Y'/)
      END
      SUBROUTINE CLAY (QCOEF,THKL,SSEC,HYCOND,SS)
C
C     PURPOSE:  THIS PROGRAM COMPUTES THE TRANSIENT LEAKAGE COEFFICIENT,
C               QCOEF.
C
C
C     DIMT------DIMENSIONLESS TIME
C     SS--------SPECIFIC STORAGE
C     HYCOND----HYDRAULIC CONDUCTIVITY
C     THK-------THICKNESS OF CONFINING LAYER
C
C
      DATA PIE/3.141593/
      TT=0.
      SUMN=0.0
      IF(SS.NE.0.) GO TO 5
      DENOM=1.0
      GO TO 31
    5 DIMT=HYCOND*SSEC/(THKL*THKL*SS*2.)
      IF (DIMT.GT.TT) TT=DIMT
      PPT = PIE*PIE*DIMT
      IF (DIMT.LT.1.0E-03) PPT = 1./DIMT
      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)                                        3
   SUMN = SUMN + PEX                                        3
   IF (PEX.GT.0.00009) GO TO 20                             3
   IF (LL.GT.CK) GO TO 30                                   3
20 CONTINUE                                                 3
30 DENOM = 1.0                                              3
   IF (DIMT.LT.1.0E-03) DENOM = SQRT (PIE*DIMT)             3
31 Q1 = HYCOND / (THKL * DENOM)                             3
   QCOEF = Q1 + 2.0 * Q1 * SUMN                             3
   RETURN                                                   3
   END                                                      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 I 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 I, 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

$$L(c) = \frac{\partial}{\partial x}\left(D_{xx}\frac{\partial c}{\partial x}\right) + \frac{\partial}{\partial x}\left(D_{xy}\frac{\partial c}{\partial y}\right) + \frac{\partial}{\partial y}\left(D_{yy}\frac{\partial c}{\partial y}\right) + \frac{\partial}{\partial y}\left(D_{yx}\frac{\partial c}{\partial x}\right)$$

$$- \frac{\partial}{\partial x}(cq_x) - \frac{\partial}{\partial y}(cq_y) - \frac{\partial}{\partial t}(\theta bc) + Qc' + \frac{K'}{\ell'}(h-h_w)c'' \qquad (2\text{-}1)$$

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 $[L^3T^{-1}]$,

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 $[LT^{-1}]$,

θ is the porosity.

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

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

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

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 1 we approximate the unknown concentration and certain of the spatially dependent parameters in terms of the basis functions $w_i(x,y)$:

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

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

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

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

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

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:

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

This may be expanded and reformulated in matrix form as

$$[N]\{C\} + [M]\{\frac{dC}{dt}\} + \{F\} = 0 \tag{2-9}$$

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

$$N_{k\ell} = \int_A \hat{D}_{xx} \frac{\partial w_\ell}{\partial x}\frac{\partial w_k}{\partial x} + 2\hat{D}_{xy}\frac{\partial w_\ell}{\partial y}\frac{\partial w_k}{\partial x} + \hat{D}_{yy}\frac{\partial w_\ell}{\partial y}\frac{\partial w_k}{\partial y}$$

$$+ \hat{q}_x w_k \frac{\partial w_\ell}{\partial x} + \hat{q}_y w_k \frac{\partial w_\ell}{\partial y} + w_k w_\ell \frac{\partial \hat{q}_x}{\partial x} + w_k w_\ell \frac{\partial \hat{q}_y}{\partial y}\, dA \tag{2-10a}$$

$$M_{k\ell} = \int_A b\theta w_k w_\ell dA \tag{2-10b}$$

$$F_k = \int_A w_k Q c' dA - \int_S (w_k \underset{\approx}{D} \cdot \underset{\sim}{\nabla}\hat{c}) \cdot \underset{\sim}{n}\, dS \tag{2-10c}$$

where $\underset{\sim}{n}$ 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 α_L is the longitudinal dispersivity [L]

α_T is the transverse dispersivity [L]

D_d is the molecular diffusion coefficient $[L^3T^{-1}]$, and

τ 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*

Variable	Format	Variable Description
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	I5	Number of nodes
NE	I5	Number of elements
NS	I5	Number of constant head nodes
NB	I5	Global flow matrix halfband width*
NF	I5	Number of source or sink nodes
NL	I5	Number of leaky elements
KNS	I5	Number of constant concentration nodes
KNB	I5	Global concentration matrix halfbandwidth
NLH	I5	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 Δt in hours
CHNG	F10•0	Multiplier used to change size of time step according to relationship $\Delta t_i = CHNG*\Delta t_{i-1}$
ITMAX	I10	Maximum permitted number of time steps
ITCHNG	I10	Number of time steps between changes in Δt
TCHK	F10•0	Spatial criterion for time step selection Δt=TCHK/(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	I10	Number of time steps between printouts
IGO	I10	Number of concentration solutions between head calculations, Δt(head)=Δt(concentration) *IGO
UPVAL	F10.0	Upstream weighting coefficient (normally use 1.)+

3.5 *Data Set 5: Constant Multipliers*

AFMOBX	F10•0	Multiplier for X hydraulic conductivity, i.e., FMOBX (used) = FMOBX (read)*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.

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

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

KOD1	I5	Print element flow matrices (1=output; 0=suppress output)
KOD2	I5	Print global flow matrix (1=output; 0=suppress output)
KOD3	I5	Print element concentration matrices (1=output; 0=suppress output)
KOD4	I5	Print global concentration matrix (1=output; 0=suppress output)
KOD7	I5	Print point velocities (value of KOD7 is the number of time steps between each printout)
KOD8	I5	Print vector of known values on the right-hand side of the matrix equation (1=output; 0=suppress output).
KOD9	I5	Print hydraulic head solution (1=output; 0=suppress output)
KOD10	I5	Print concentration solution (1=output; 0=suppress output)
KOD12	I5	Number of time steps between punched output
KOD14	I5	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	I5	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	I5	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	I5	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*

PHII(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	I3	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	I3	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	I5	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*

Data set 14b leaky [illegible] Node N
I I5
HZSRD G10.0 Content.

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

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
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
C
C      PROGRAMMED BY GEORGE F. PINDER, DEPT. OF CIVIL ENGINEERING,
C      PRINCETON UNIVERSITY, PRINCETON N. J. 08540
C
C      D E F I N I T I O N   O F   V A R I A B L E S
C      ACCNCI------MULTIPLIER FOR INITIAL CONCENTRATION
C      AELONG------MULTIPLIER FOR ELONG, LONGITUDINAL DISPERSIVITY
C      AETRAN------MULTIPLIER FOR ETRANS, TRANSVERSE DISPERSIVITY
C      AFMOBX------MULTIPLIER FOR FMOBX, MOBILITY IN X-DIRECTION
C      AFMOBY------MULTIPLIER FOR FMOBY, MOBILITY IN Y-DIRECTION
C      AG(I)-------CONSTANTS FOR NUMERICAL INTEGRATION
C      AHHII-------MULTIPLIER FOR INITIAL HEAD
C      APOR--------MULTIPLIER FOR POR, POROSITY
C      AST---------MULTIPLIER FOR ST, STORAGE COEFFICIENT
C      ATHICK------MULTIPLIER FOR THICK, THICKNESS OF AQUIFER
C      CAVG--------AVERAGE CONCENTRATION OVER AN ELEMENT
C      CCCEF(L)----CONCENTRATION OF LEAKAGE
C      CPC(I)------CONCENTRATION OF SOURCE OR SINK - IF NEGATIVE,
C                  CURRENT CONCENTRATION OF THE NODE IS ASSUMED
C      CHAR(I)-----ALPHAMERIC CHARACTER STRING FOR INCIDENCES
C      CHNG--------MULTIPLIER FOR INCREASING TIME STEP (DELT=CHNG*DELT)
C      CH(I,J)-----TIME INDEPENDENT GLOBAL CONC. MATRIX
C      COEF--------HYDRAULIC CONDUCTIVITY OF CONFINING BED DIVIDED BY ITS
C                  THICKNESS
C      COLI--------STORAGE VECTOR FOR CONCENTRATION
C      CONC(I)-----CONCENTRATION (M/L**3)
C      COR(I)------LOCATION OF PARTICULAR SOLUTIONS
C      CP(I,J)-----TIME DEPENDENT GLOBAL CONC. MATRIX
C      DELT--------TIME STEP (T)
C      DELTGO------TIME STEP FOR PRESSURE - DELTGO=DELT*IGO
C      DETJ(I)-----DETERMINANT OF J AT EACH INTEGRATION POINT
C      DEX(I)------TEMPORARY STORAGE FOR THE POINT X VELOCITY
C      DEY(I)------TEMPORARY STORAGE FOR THE POINT Y VELOCITY
C      DIFUSN------MOLECULAR DIFFUSION CONSTANT
C      DGX(I)------DF/DX FOR NODE BEING PROCESSED
C      DGY(I)------DF/DY FOR NODE BEING PROCESSED
C      DIGIT(I)----ALPHAMERIC CHARACTER STRING FOR INCIDENCES
C      DPX(I)------TEMPORARY STORAGE FOR PRESSURE DERIVATIVE
C      DPY(I)------TEMPORARY STORAGE FOR PRESSURE DERIVATIVE
C      DPDX(I)-----AVERAGE X-DERIVATIVE OF PRESSURE FOR ELEMENT L
C      DPDY(I)-----AVERAGE Y-DERIVATIVE OF PRESSURE FOR ELEMENT L
C      DX(I,J)-----DF/DX FOR I NODES OF ELEMENT L
C      DY(I,J)-----DF/DY FOR I NODES OF ELEMENT L
C      ELONG-------LONGITUDINAL DISPERSIVITY
C      ETRANS------TRANSVERSE DISPERSIVITY
C      F(I,J)------BASIS FUNCTION VALUES FOR ALL NODES IN ELEMENT
C      FACTOR------MULTIPLIER FOR CO-ORDINATES. VALUE = VALUE(READ)*FACTO
C      FE(I)-------BASIS FUNCTION AT A POINT FOR ALL NODES IN ELEMENT
C      FM(I)-------VECTOR ON RHS OF MATRIX EQUATION
C      FMOBX(I)----X-COMPONENT OF MOBILITY TENSOR, K/MU (L**3T/M)
C      FMOBY(I)----Y-COMPONENT OF MOBILITY TENSOR, K/MU (L**3T/M)
C      FQ(I)-------RECHARGE TO AQUIFER (POS) (L**3/T)
C      H(I,J)------TIME DEPENDENT PART OF GLOBAL FLOW MATRIX
C      HZERO(I)----HEAD IN ADJACENT AQUIFER I
C      IN(I,J)-----INCIDENCES
C      IT----------NUMBER OF TIME STEPS COMPLETED
C      ITCHNG------NUMBER OF TIME STEPS BETWEEN CHANGES IN DELT
```

```
C     IGC---------NUMBER OF CONCENTRATION CALCULATIONS PER PRESSURE
C                 CALCULATION
C     ITMAX-------MAXIMUM PERMITTED NUMBER OF TIME STEPS
C     IVTEST------EQUALS 1 IF VELOCITY HAS BEEN COMPUTED, ZERO
C                 OTHERWISE
C     JD(I)-------CONTAINS THE INCIDENCES OF THE ACTIVE NODE IN ELEMENT
C     JTEST-------EQUALS 0 IF CONC. MATRIX TO BE GENERATED,
C                 EQUALS 1 IF FLOW MATRIX TO BE GENERATED
C                 EQUALS 2 IF BOTH MATRICES TO BE GENERATED
C     KKB(I)------ROW INDICES OF CONSTANT CONCENTRATION NODES
C     KLB(I)------EQUALS ONE FOR CONSTANT CONCENTRATION NODE, ZERO
C                 OTHERWISE
C     KLBC(I)-----RUNNING SUM OF CONSTANT CONCENTRATION NODES
C     KN----------DEGREES OF FREEDOM FOR CONCENTRATION
C     KNB---------CS MATRIX HALFBAND WIDTH (ESTIMATED)
C     KNBEST------ORIGINAL ESTIMATE OF CS HALFBANDWIDTH
C     KNB2--------CS BANDWIDTH
C     KNB2ST------ORIGINAL ESTIMATE OF CS BANDWIDTH
C     KNS---------NUMBER OF CONSTANT CONCENTRATION NODES
C     KOD1--------PRINT ELEMENT FLOW MATRICES (1=OUTPUT)
C     KOD2--------PRINT GLOBAL FLOW MATRIX (1=OUTPUT)
C     KOD3--------PRINT ELEMENT CONCENTRATION MATRICES (1=OUTPUT)
C     KOD4--------PRINT GLOBAL CONCENTRATION MATRIX (1=OUTPUT)
C     KOD5--------NOT CURRENTLY USED
C     KOD6--------NOT CURRENTLY USED
C     KOD7--------PRINT POINT VELOCITIES (VALUE OF KOD7 IS THE NUMBER
C                 OF TIME STEPS BETWEEN EACH CALCULATION)
C     KOD8--------PRINT VECTOR OF KNOWN VALUES (1=OUTPUT)
C     KOD9--------PRINT PRESSURE SOLN. (1=OUTPUT)
C     KOD10-------PRINT CONCENTRATION SOLUTION (1=OUTPUT)
C     KOD11-------NOT CURRENTLY USED
C     KOD12-------NUMBER OF TIME STEPS BETWEEN PUNCHED OUTPUT
C     KOD13-------NOT CURRENTLY USED
C     KOD14-------TYPE OF PUNCHED OUTPUT
C                 0 INDICATES PRESSURES AND CONCENTRATION
C                 -1 INDICATES PRESSURES ONLY
C                 1 INDICATES CONCENTRATION ONLY
C     KR(I)-------ROW INDICES OF CONSTANT PRESSURE NODES
C     L-----------ELEMENT NUMBER
C     LB(I)-------EQUALS ONE FOR CONSTANT PRESSURE NODE, ZERO OTHERWISE
C     LBC(I)------RUNNING SUM OF CONSTANT PRESSURE NODES
C     LBT(I)------VALUES OF CONSTANT PRESSURE NODES (NUMBER)
C     M-----------NUMBER OF ACTIVE NODES IN ELEMENT
C     N-----------NUMBER OF DEGREES OF FREEDOM FOR FLOW
C     NB----------S MATRIX HALFBAND WIDTH (ESTIMATED)
C     NB1---------MODIFIED HALFBAND WIDTH
C     NE----------NUMBER OF ELEMENTS
C     NF----------NUMBER OF SOURCE OR SINK NODES
C     NL------------NUMBER OF LEAKY ELEMENTS
C     NN----------NUMBER OF NODES
C     NPRINT------NO. OF TIMES EQUATIONS ARE SOLVED BETWEEN PRINTOUTS
C     NR----------NUMBER OF DIRICHLET BOUNDARY NODES FOR FLOW
C     NS----------NUMBER OF CONSTANT PRESSURE NODES
C     OLD(I)------STORAGE VECTOR FOR U
C     P(I,J)------TIME DEPENDENT GLOBAL FLOW MATRIX
C     PE(I,J)-----TIME DEPENDENT ELEMENT FLOW MATRIX
C     PHI(I)------PRESSURE  (M/LT**2) OR HEAD IF AREAL PROBLEM (L)
C     PHIAVG------AVERAGE HEAD OVER AN ELEMENT
C     PHII(I)-----INITIAL PRESSURE  (M/LT**2) OR HEAD (L)
C     POR(I)------POROSITY
```

```
C      PZERO(I)----COEFFICIENT FOR TIME DERIVATIVE OF CONCENTRATION IN
C                  FLOW EQUATION
C      RSO(I)------KNOWN VALUES FOR R.H.S OF CONC. EQN. (CURRENTLY ZERO)
C      RT(I)-------VECTOR OF KNOWN VALUES
C      S(I,J)------STORAGE MATRIX FOR PRESSURE
C      SE(I,J)-----ELEMENT FLOW MATRIX
C      SLOPE-------ANGLE IN DEGREES BETWEEN X-AXIS AND HORIZONTAL
C      SMIN--------ELAPSED SIMULATION TIME (MINUTES)
C      SRCLT(I)----UNKNOWN PART OF LEAKAGE AT POINT I
C      SRCR(I)-----KNOWN PART OF LEAKAGE FOR NODE I
C      SRCRT(I)----KNOWN PART OF LEAKAGE AT POINT I
C      SSEC--------ELAPSED SIMULATION TIME (SECONDS)
C      STIME-------ELAPSED SIMULATION TIME (HOURS)
C      TCHK--------SPACIAL CRITERION FOR TIME STEP SELECTION
C      THICK-------THICKNESS OF AQUIFER IN AREAL PROBLEM
C      TIME--------SIMULATION PERIOD IN HOURS (T)
C      TITLE(I)----PROGRAM TITLE
C      TYPE--------TIME DERIVATIVE INDICATOR - FORWARD AND CENTRE
C      U(I)--------SOLUTION VECTOR FOR PRESSURE
C      VLX---------AVERAGE VELOCITY IN THE X DIRECTION
C      VLY---------AVERAGE VELOCITY IN THE Y DIRECTION
C      VX(I)-------X-COMPONENT OF VELOCITY (L/T)
C      VY(I)-------Y-COMPONENT OF VELOCITY (L/T)
C      X(I)--------X-COORDINATE OF NODE (L)
C      Y(I)--------Y-COORDINATE OF NODE (L)
C      XI----------X CO-ORDINATE OF INTEGRATION POINT
C      YI----------Y CO-ORDINATE OF INTEGRATION POINT
C
C.....NOTES:
C            1. THE HALFBANDWIDTH INCLUDES THE DIAGONAL ELEMENTS
C            2. THE VARIABLES FMOBX AND FMOBY CHANGE THEIR DEFINITION
C            3. ELONG,ETRANS, AND DIFUSN ARE VERTICALLY INTEGRATED VALUES
C            4. THE PROGRAM IS DESIGNED AS A TEACHING TOOL AND AND
C               THEREFORE HAS A MINIMUM OF FRILLS
C            5. LAST MODIFICATION OF THIS PROGRAM WAS AUGUST 1976
C      -----------------------------------------------------------------
       IMPLICIT INTEGER*2 (I-N)
       INTEGER*4 IT,IGO,KTCAL,ITERS,ITCHNG,IROW,NCOL,M1,I,NPRINT,IARG1,
      1KN,IER,N
       COMMON /SCALAR/
      1KOD1,  KOD2,  KOD3,  KOD4,  KOD5,  KOD6,  KOD7,  KOD11, KOD12,
      2KOD13, KOD14, KOD15, NN,    NE,    NP,    NB,    NB1,   KNB,
      3KNB1,  KNB2ST,KSTRT, KNBEST,KNB2
       COMMON /DSCALR/
      1CF,DIFUSN,ETRANS,ELONG,TCHK,SRCL,DELT,VMAX
       COMMON /ARRAY/
      1LR,    LRC,   IN,    KLR,   KLRC
       COMMON /DARRAY/
      1S,     CP,    CH,    CPM,   CRT,   CONC,  CONCI, COLD,  VX,
      1VY, VLX, VLY, FMOBX, FMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
      2DISPXY, PHI, P, FM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,
      3RT, COEF, HZERO, COR, DGX, DGY, Q, CFQ, 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),CPM(150), VX(12), VY(12),
      2COR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),
      3QP(16), SRCR(12), SRCRT(16), FQ(150), FM(150), CRT(150),
      4PHII(150), FMOBX(91), FMOBY(91), LR(150), LRC(150), Y(150),
      5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
      6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),
      ATHICK(91),ART(125),CCOEF(91)
```

```
      DIMENSION TITLE(20),CHAR(77), DIGIT(10),KR(40),   LRT(50),
     1KRB(40)
      DATA BLANK/1H /,STAR/1H*/,DIGIT/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H
     18,1H9/,START/4HISOQ/,EXIT/4HXXXX/,KR(1)/0/,
     2FWD/4HFORW/,CNTR/4HCENT/,YES/0./,NO/1/
C     ------------------------------------------------------------------
C
C.....R E A D   A N D   W R I T E   D A T A
   10 READ (5,840) CARD
      IF (CARD.EQ.EXIT) STOP
      IF (CARD.NE.START) GO TO 10
      READ (5,840) TITLE
      WRITE (6,850)
      WRITE (6,860) TITLE
C
      READ (5,910) NN,NE,NS,NB,NP,NL,KNS,KNB,NLH
      WRITE (6,920)
      WRITE (6,930) NN,NE,NS,NP,NB,KNB,NL,KNS,NLH
C
      READ (5,940) TIME,DELT,CHNG,ITMAX,ITCHNG,TCHK,TYPE,DIFUSN,
     1NPRINT,IGC
      WRITE(6,950) TIME,DELT,CHNG,ITMAX,ITCHNG,TCHK,DIFUSN,
     1NPRINT,IGC
      IF (TYPE.EQ.FWD) WRITE (6,960)
      IF (TYPE.EQ.CNTR) WRITE (6,970)
C
      READ (5,1440) APMOBX,APMOBY, AST, APOR, AELONG, AETRAN , ATHICK,
     1APHII, ACCNCI, FACTOR,ALEAK
      WRITE (6,1450) APMOBX, APMOBY, AST, APOR, AELONG, AETRAN, ATHICK,
     1APHII, ACONCI, FACTOR,ALEAK
C
      READ (5,910) KOD1,KOD2,KOD3,KOD4,KOD7,KOD8,KOD9,KOD10,KOD12,KOD14
C
C.....WRITE OUT CODE INTERPRETATION
      WRITE (6,25)
      IF (KOD1   .EQ.0) WRITE (6,1)
      IF (KOD1   .NE.0) WRITE(6,2)
      IF (KOD2   .EQ.0) WRITE (6,3)
      IF (KOD2   .NE.0) WRITE(6,4)
      IF (KOD3   .EQ.0) WRITE (6,5)
      IF (KOD3   .NE.0) WRITE(6,6)
      IF (KOD4   .EQ.0) WRITE (6,7)
      IF (KOD4   .NE.0) WRITE (6,8)
      IF (KOD7   .NE.0) WRITE (6,9)   KOD7
      IF (KOD8   .EQ.0) WRITE (6,12)
      IF (KOD8   .NE.0) WRITE (6,13)
      IF (KOD9   .EQ.0) WRITE (6,14)
      IF (KOD9   .NE.0) WRITE (6,15)
      IF (KOD10  .EQ.0) WRITE (6,16)
      IF (KOD10  .NE.0) WRITE (6,17)
      IF (KOD12  .NE.0) WRITE (6,18)  KOD12
      IF (KOD14  .EQ.0) WRITE(6,22)
      IF (KOD14  .EQ.-1) WRITE (6,23)
      IF (KOD14  .EQ.1) WRITE(6,24)
C
C.....NODE COORDINATES
      READ (5,980) (J,X(J),Y(J),K=1,NN)
      DO 19 J=1,NN
      X(J)=X(J)*FACTOR
   19 Y(J)=Y(J)*FACTOR
```

```
      WRITE (6,990)
      WRITE (6,1000)
      WRITE (6,1010)  (J,X(J),Y(J),J=1,NN)
C
C.....SOURCE AND SINK NODES, FQ.
C.....SOURCES ARE POSITIVE
      WRITE (6,1020)
      DO 20 I=1,NN
   20 FQ(I)=0.
      IF (NF.EQ.0) GO TO 40
      DO 30 I=1,NF
      READ (5,980) IQ,FQ(IQ)
   30 WRITE (6,1030) IQ,FQ(IQ)
   40 CONTINUE
C
C.....SOURCE AND SINK CONCENTRATIONS, CFQ
      WRITE (6,1040)
      DO 50 I=1,NN
   50 CFQ(I)=0.
      IF (NF.EQ.0) GO TO 70
      DO 60 I=1,NF
      READ (5,980) IQ,CFQ(IQ)
   60 WRITE (6,1030) IQ,CFQ(IQ)
   70 CONTINUE
C
C.....READ INITIAL PRESSURE OR HEAD VALUES
      READ (5,1051)  (PHII(I),I=1,NN)
      DO 1470 I=1,NN
 1470 PHII(I)=PHII(I)*APHII
      WRITE (6,1070)
   90 WRITE (6,1080)
      WRITE (6,1090)  (I,PHII(I),I=1,NN)
C
C.....READ INITIAL CONCENTRATION VALUES
      READ (5,1051)  (CONCI(I),I=1,NN)
      DO 1460 I=1,NN
 1460 CONCI(I)=CONCI(I)*ACONCI
      WRITE (6,1100)  (I,CONCI(I),I=1,NN)
C
C.....ELEMENT INCIDENCES
C.....FREE FORMAT FOR INPUT
      DO 100 I=1,13
      DO 100 L=1,NE
  100 IN(I,L)=0
      DO 200 LL=1,NE
      READ (5,1110) L,CHAR
      KC=0
      KE=4
      KS=2
      J=0
  110 NUM=0
  120 J=J+1
      IF (CHAR(J).EQ.BLANK) GO TO 120
C
C.....DETERMINE INCIDENCE FROM CHARACTER LIST AND ADJUST FOR POWER OF
C.....TEN.
  130 I=0
  140 I=I+1
      IF (I.LE.10) GO TO 150
      WRITE (6,1120) CHAR(J),L
```

```
      GO TO 10
  150 IF (CHAR(J).NE.DIGIT(I)) GO TO 140
      NUM=10*NUM+I-1
      J=J+1
C
C.....CHECK NEXT LOCATION-IF IT IS AN $ THIS IS A CORNER NODE. IF IT IS
C     BLANK THIS IS LAST DIGIT OF A SIDE NODE.
      IF (CHAR(J).EQ.STAR) GO TO 160
      IF (CHAR(J).EQ.BLANK) GO TO 170
      GO TO 130
  160 KC=KC+1
C
C.....WHEN KC, THE CORNER COUNTER, IS FIVE THE INCIDENCE LIST FOR
C.....ELEMENT L IS COMPLETE.
C.....WHEN A NODE IS ABSENT FROM A POSSIBLE LOCATION,ZERO IS RECORDED IN
C.....THE IN ARRAY.
      IF (KC.EQ.5) GO TO 180
      IN(KC,L)=NUM
      IF (KS.EQ.0) KE=KE+2
      IF (KS.EQ.1) KE=KE+1
      KS=0
      GO TO 110
  170 KE=KE+1
      KS=KS+1
      IN(KE,L)=NUM
      IF (J.LT.65) GO TO 110
      WRITE (6,1130) L
      GO TO 10
  180 N=4
C
C.....COUNT ACTIVE NODES.  IE THOSE NOT EQUAL TO ZERO.
      DO 190 I=5,12
      IF (IN(I,L).EQ.0) GO TO 190
      N=N+1
  190 CONTINUE
      IN(13,L)=N
  200 CONTINUE
      WRITE (6,1140)
      WRITE (6,1150)
      DO 210 L=1,NE
  210 WRITE (6,1160) L,(IN(I,L),I=1,12)
C
C.....READ ELEMENT PARAMETERS
      WRITE (6,1170)
      DO 220 LL=1,NE
      READ (5,1180) L,PMOBX(L),PMOBY(L),ST(L),POR(L),ELONG(L),ETRANS(L),
     1THICK(L)
      PMOBX(L)=PMOBX(L)*APMOBX
      PMOBY(L)=PMOBY(L)*APMOBY
      ST(L)=ST(L)*AST
      POR(L)=POR(L)*APOR
      ELONG(L)=ELONG(L)*AELONG
      ETRANS(L)=ETRANS(L)*AETRAN
      THICK(L)=THICK(L)*ATHICK
      WRITE (6,1190) L,PMOBX(L),PMOBY(L),ST(L),POR(L),ELONG(L),ETRANS(L)
     1,THICK(L)
C.....NOTE THAT PMOBX AND PMOBY ARE NOW REDEFINED AS TRANSMISSIVITY
      PMOBY(L)=PMOBY(L)*THICK(L)
      PMOBX(L)=PMOBX(L)*THICK(L)
  220 CONTINUE
```

```
C
C.....LEAKY ELEMENT COEFFICIENTS
      DO 230 I=1,NE
      CCCEF(I)=0.
  230 COEF(I)=0.
      IF (NL.EQ.0) GO TO 250
      DO 240 J=1,NL
      READ(5,983) I,COEF(I),CCOEF(I)
  240 COEF(I)=COEF(I)*ALEAK
      WRITE(6,1200) (I,COEF(I),CCOEF(I),I=1,NE)
      DO 241 I=1,NN
  241 HZERO(I)=0.
      READ (5,242) (I,HZERO(I),J=1,NLH)
  242 FORMAT(5(I5,G10.0))
      WRITE (6,243)
  243 FORMAT (///11X,32HCONSTANT HEAD FOR LEAKY ELEMENTS/11X,32(1H-))
      WRITE (6,1080)
      WRITE (6,1090) (I,HZERO(I),I=1,NN)
  250 CONTINUE
C
C.....DIRICHLET BOUNDARY NODES FOR FLOW (INDICATED BY LR=1)
      WRITE (6,1230)
      DO 270 I=1,NN
  270 LR(I)=0.
      IF (NS.EQ.0) GO TO 310
      NST=0.
  280 READ (5,1240) (LRT(ITT),ITT=1,20)
      IA=0
      DO 290 I=1,20
      IF (LRT(I).EQ.0) GO TO 300
      IA=I
      J=LRT(I)
      NST=NST+1
      IF (J.LE.NN) GO TO 290
      WRITE (6,1250) J
      GO TO 10
  290 LR(J)=1
  300 WRITE (6,1260) (LRT(IIT),IIT=1,IA)
      IF (IA.EQ.20) GO TO 280
      IF (NST.NE.NS) WRITE (6,1270) NST,NS
C
C.....DIRICHLET NODES FOR CONCENTRATION
  310 WRITE (6,1280)
      DO 320 I=1,NN
  320 KLR(I)=0
      IF (KNS.EQ.0) GO TO 360
      NST=0
  330 READ (5,1240) (LRT(ITT),ITT=1,20)
      IA=0
      DO 340 I=1,20
      IF (LRT(I).EQ.0) GO TO 350
      IA=I
      J=LRT(I)
      NST=NST+1
      IF (J.LE.NN) GO TO 340
      WRITE (6,1290) J
      GO TO 10
  340 KLR(J)=1
  350 WRITE (6,1260) (LRT(IIT),IIT=1,IA)
      IF (IA.EQ.20) GO TO 330
```

```
      IF (NST.NE.KNS) WRITE (6,1300) NST,KNS
  360 CONTINUE
C.....END DATA INPUT
C
C.....I N I T I A L I Z A T I O N
C.....SUM DIRICHLET BOUNDARY NODES
      LRC(1)=LR(1)
      DO 370 J=2,NN
  370 LRC(J)=LRC(J-1)+LR(J)
      NB=LRC(NN)
      N=NN-NB
      KLRC(1)=KLR(1)
      DO 380 J=2,NN
  380 KLRC(J)=KLRC(J-1)+KLR(J)
      KNB=KLRC(NN)
      KN=NN-KNB
      WRITE (6,1310) NN,NB,N,KNB,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
      KB(K)=J
  390 CONTINUE
      K=0
      DO 400 J=1,NN
      IF (KLR(J).LT.1) GO TO 400
      K=K+1
      KKB(K)=J
  400 CONTINUE
C
C.....CONSTANTS
      ADVANC=YES
      STIME=0.
      KTIT=0
      IT=0
      DELT=DELT*3600.
      SSIC=0.
      KNB1=0
      NB1=0
      KNBST=KNB
      KNB2=2*KNB-1
      KNB2ST=KNB2
      L=0
C
C.....CLEAR ARRAYS
      DO 410 I=1,N
      DO 410 J=1,NB
      P(I,J)=0.
  410 H(I,J)=0.
      DO 420 I=1,KN
      DO 420 J=1,KNB2
      CP(I,J)=0.
  420 CH(I,J)=0.
C
      II=0
      KII=0
      DO 470 I=1,NN
      CPH(I)=CONCI(I)
```

```
      PM(I)=PHII(I)
      IF (LR(I).GT.0) GO TO 430
      II=II+1
      PHI(II)=PHII(I)
      OLD(II)=PHI(II)
  430 IF (KIR(I).GT.0) GO TO 470
      KII=KII+1
      IF (CFQ(I)) 450,440,440
  440 CRT(KII)=-FQ(I)*CFQ(I)
      GO TO 460
450   CRT(KII)=0
  460 CONC(KII)=CONCI(I)
      COLD(KII)=CONC(KII)
  470 CONTINUE
C
C.....END DATA INITIALIZATION
C
C
C     T R A N S I E N T   S O L U T I O N
C.....GENERATE COEFFICIENT MATRICES FOR FLOW AND TRANSPORT
C     ***********************
582   CALL COGEN (2,N,KN,IT)
C     ***********************
C
C.....TIME STEP MODIFICATION?
      IF (MOD(IT,ITCHNG).NE.0) GO TO 580
C
C.....COMPUTE TIME STEP
      IF (IT.LT.1) GO TO 580
      DELT=TCHK/VMAX
580   CONTINUE
C
C
C.....SOLVE FOR HYDRAULIC HEAD
      DELTGO=DELT*IGO
      DO 600 I=1,N
  600 OLD(I)=PHI(I)
C
C.....SELECT APPROXIMATION FOR TIME DERIVATIVE
      IF (TYPE.EQ.FWD) A3=1./DELTGO
      IF (TYPE.EQ.CNTR) A3=2./DELTGO
C.....COMPUTE S FROM H AND P
      DO 660 I=1,N
      DO 660 J=1,NB1
  660 S(I,J)=H(I,J)+P(I,J)*A3
C
C
C.....DECOMPOSE
C     ******************
      CALL DBAND (N,IER)
C     ******************
C
C.....COMPUTE VECTOR OF KNOWN VALUES
      DO 670 I=1,N
  670 FM(I)=-RT(I)
      N1=N+1
      N2=N-1
      FM(N)=FM(N)+P(N,1)*A3*OLD(N)
      DO 680 I=1,N2
      IARG1=NB
```

```
      L1=MINO(IARG1,M1-I)
      FM(I)=FM(I)+P(I,1)*A3*OLD(I)
      DO 680 J=2,L1
      LM=I+J-1
      FM(I)=FM(I)+P(I,J)*A3*OLD(LM)
  680 FM(LM)=FM(LM)+P(I,J)*A3*OLD(I)
      IF (KOD8.NE.1) GO TO 690
      WRITE (6,1350)
      WRITE (6,1080)
      WRITE (6,1090) (I,FM(I),I=1,N)
C
C.....BACK SUBSTITUTE
C     **************
  690 CALL SBAND (N)
C     **************
C
C.....CALCULATE NEW VALUES
      IF (TYPE.EQ.FWD) GO TO 721
      DO 700 I=1,N
  700 PHI(I)=2.*PHI(I)-OLD(I)
  721 CONTINUE
C
C.....ADJUST FOR CORRECT NODAL SEQUENCE - USE FM FOR TEMPORARY STORAGE
      IP=0
      DO 770 I=1,NN
      IF (LR(I).GT.0) GO TO 760
      IP=IP+1
      FM(I)=PHI(IP)
      GO TO 770
  760 FM(I)=PHII(I)
  770 CONTINUE
C.....END CALCULATIONS FOR HEAD
C
C
C.....CALCULATE CONCENTRATION
C.....SELECT APPROXIMATION FOR TIME DERIVATIVE
      IF (TYPE.EQ.FWD) A3=1./DELT
      IF (TYPE.EQ.CNTR) A3=2./DELT
      DO 421 I=1,KN
      DO 421 J=1,KNB2
421   S(I,J)=0.
C
C.....POINT OF RETURN FOR INNER CONCENTRATION CALCULATION
C.....COMPUTE S FROM CP AND CM AND KNOWN VECTOR, CFM
C.....UPDATE COLD
581   DO 590 I=1,KN
  590 COLD(I)=CONC(I)
      KSTART=1
      KDIFF=KN-KNB+1
      JSTART=KNB+1+KSTRT
      JSTOP=KNB1
      DO 640 I=1,KN
      CFM(I)=-CRT(I)
      IF (I.LE.KNB) JSTART=JSTART-1
      IF (I.GT.KDIFF) JSTOP=JSTOP-1
      IF (I.GT.KNB) KSTART=KSTART+1
      K=KSTART-1
      DO 640 J=JSTART,JSTOP
      K=K+1
      JACT=J-KSTRT
```

```
      IF (ADVANC.EQ.NO) GO TO 640
      S(I,JACT)=CH(I,J)+CP(I,J)*A3
  640 CFM(I)=CFM(I)+CP(I,J)*COLD(K)*A3
      IF(ADVANC.NE.YES) GO TO 6405
      IP=0
      DO 6402 I=1,NN
      IF (KIR(I).GT.0) GO TO 6402
      IP=IP+1
      IF (FQ(I).LT.0) GO TO 6402
      S(IP,KNB)=S(IP,KNB)+FQ(I)
6402  CONTINUE
6405  CONTINUE
      IF (KOD8.NE.1) GO TO 650
      WRITE (6,1320)
      WRITE (6,1080)
      WRITE (6,1090) (I,CFM(I),I=1,KN)
  650 CONTINUE
C
C.....SOLVE CONCENTRATION EQUATIONS
      DO 651 I=1,KN
  651 ART(I)=CFM(I)
      IARG1=KNB-1
C     ************************************************
      IF(ADVANC.EQ.NO) GO TO 653
      CALL SOLVE (1,S,ART,KN,IARG1,125,66)
653   CALL SOLVE (2,S,ART,KN,IARG1,125,66)
C     ************************************************
C
      DO 652 I=1,KN
  652 CFM(I)=ART(I)
C
C.....CALCULATE NEW VALUES
      IF (TYPE.EQ.FWD) GO TO 720
      DO 710 I=1,KN
  710 CONC(I)=2.*CFM(I)-COLD(I)
      GO TO 741
  720 DO 730 I=1,KN
  730 CONC(I)=CFM(I)
C
C.....ADJUST FOR CORRECT NODAL SEQUENCE - USE CFM FOR TEMPORARY STORAGE
  741 KIP=0
      DO 791 I=1,NN
      IF (KIR(I).GT.0) GO TO 780
      KIP=KIP+1
      CFM(I)=CONC(KIP)
      GO TO 791
  780 CFM(I)=CONCI(I)
  791 CONTINUE
  755 CONTINUE
      SSEC=SSEC+DELT
      SMIN=SSEC/60.
      STIME=SMIN/60.
      IT=IT+1
C.....END TRANSIENT SOLUTION CALCULATIONS
C
C
C..... O U T P U T
751   IF (MOD(IT,NPRINT).NE.0) GO TO 820
      IF (KOD9.NE.1.AND.KOD10.NE.1) GO TO 820
      IF (IT.EQ.1) GO TO 800
```

```
      DELT1=DELT/3600.
      WRITE (6,1360) IT,DELT1,STIME,SMIN,SSEC
  800 IF (KOD9.NE.1) GO TO 810
      WRITE (6,1370)
      WRITE (6,1080)
      WRITE (6,1090) (I,FM(I),I=1,NN)
  810 IF (KOD10.NE.1) GO TO 820
      WRITE (6,1390)
      WRITE (6,1080)
      WRITE (6,1090) (I,CFM(I),I=1,NN)
C     ***********************
      IARG1=KOD7
      IF (MOD(IT,IARG1).EQ.0) CALL VCAL
C     ***********************
      WRITE (6,1400)
      WRITE (6,1410)
820   CONTINUE
      IARG1=KOD12
      IF (MOD(IT,IARG1) .NE. 0) GO TO 829
      IF (KOD14.LT. 0) GO TO 828
      WRITE (7,1053) IT,STIME
      WRITE (7,1051) (CFM(I),I=1,NN)
  828 IF (KOD14.GT.0) GO TO 829
      WRITE (7,1052) IT,STIME
      WRITE (7,1051) (FM(I),I=1,NN)
  829 IF (STIME.LT.TIME) GO TO 830
      WRITE (6,1420) STIME
      GO TO 836
  830 IF (IT.LT.ITMAX) GO TO 520
      WRITE (6,1430) IT
  836 IF (KOD14.LT.0) GO TO 827
      WRITE (7,1053) IT,STIME
      WRITE (7,1051) (CFM(I),I=1,NN)
  827 IF (KOD14.GT.0) GO TO 821
      WRITE (7,1052) IT,STIME
      WRITE (7,1051) (FM(I),I=1,NN)
      GO TO 821
C.....END OUTPUT
C
C.....SELECT INNER OR OUTER CONCENTRATION CALCULATION
  520 ADVANC=NO
      IF (MOD(IT,IGO).EQ.0) ADVANC=YES
      IF (ADVANC.EQ.NO) GO TO 581
      GO TO 582
C
C.....SELECT NEW DATA SET
  821 GO TO 10
C
C..... F O R M A T S
1     FORMAT (11X,'SUPPRESS PRINTOUT OF ELEMENT FLOW MATRICES')
2     FORMAT (11X, 'PRINT ELEMENT FLOW MATRICES')
3     FORMAT (11X,'SUPPRESS PRINTOUT OF GLOBAL FLOW MATRIX')
4     FORMAT (11X, 'PRINT GLOBAL FLOW MATRIX')
5     FORMAT(11X,'SUPPRESS PRINTOUT OF ELEMENT CONCENTRATION MATRICES')
6     FORMAT (11X,'PRINT ELEMENT CONCENTRATION MATRICES')
7     FORMAT(11X,'SUPPRESS PRINTOUT OF GLOBAL CONCENTRATION MATRIX')
8     FORMAT(11X,'PRINT GLOBAL CONCENTRATION MATRIX')
9     FORMAT(11X,'PRINT VELOCITIES EVERY' I4,' TIME STEP(S)')
12    FORMAT(11X, 'SUPPRESS PRINTOUT OF RIGHT HAND SIDE VECTORS')
13    FORMAT(11X, 'PRINT RIGHT HAND SIDE VECTORS')
```

```
14     FORMAT (11X,'SUPPRESS PRINTOUT OF COMPUTED HEAD')
15     FORMAT(11X,'PRINT COMPUTED HEAD')
16     FORMAT(11X, 'SUPPRESS PRINTOUT OF CONCENTRATION SOLUTION')
17     FORMAT(11X, 'PRINT CONCENTRATION SOLUTION')
18     FORMAT (11X,'PUNCH OUTPUT EVERY',I4,' TIME STEP(S)')
22     FORMAT (11X,'PUNCH HEAD AND CONCENTRATION')
23     FORMAT(11X,'PUNCH HEAD ONLY')
24     FORMAT(11X,'PUNCH CONCENTRATION ONLY')
25     FORMAT (/////11X,'CODE OPTIONS'/11X,12(1H-)//)
  840 FORMAT (20A4)
  850 FORMAT (1H1,36X,15HI S O Q U A D 4//28X,35HGROUNDWATER FLOW AND MA
     1SS TRANSPORT//25X,41HWITH ISOPARAMETRIC QUADRILATERAL ELEMENTS//38
     2X,14HOCTOBER 1 1971//)
  860 FORMAT (11X,70(1H*)//11X,20A4//11X,70(1H*)///)
  910 FORMAT (15I5)
  920 FORMAT (///11X,19HFINITE ELEMENT DATA/11X,19(1H-)/)
  930 FORMAT (1H ,10X,11HNUMBER OF -,2X,5HNODES,I24/21X,1H-,2X,8HELEMENT
     1S,I21/21X,1H-,2X,24HDIRICHLET BOUNDARY NODES,I5/21X,1H-,2X,20HSOUR
     2CE OR SINK NODES,I9/21X,1H-,2X,20HELEMENTS IN HALFBAND,/21X,1H ,2X
     3,8HFOR FLOW,I21/21X,1H ,2X,17HFOR CONCENTRATION,I12/21X,1H-,2X,14H
     4LEAKY ELEMENTS,I15/21X,1H-,2X,25HCONSTANT CONC. BOUNDARIES,I4,/
     521X,1H-,2X,19HLEAKY ELEMENT NODES,I10)
 940   FORMAT (3F10.0,2I10,F10.0,A4/1F10.0,2I10)
  950 FORMAT (/////11X,15HTIME PARAMETERS/11X,15(1H-)//11X,26HSIMULATION
     1 PERIOD IN HOURS,F30.2/11X,26HINITIAL TIME STEP IN HOURS,F30.6/11X
     2,35HMULTIPLIER FOR INCREASING TIME STEP,F21.3/11X,38HMAXIMUM PERMI
     3TTED NUMBER OF TIME STEPS,I18/11X,44HNUMBER OF TIME STEPS BETWEEN
     4CHANGES IN DELT,I12/11X,31HSPACIAL CRITERION FOR TIME STEP,F25.3/1
     51X,28HMOLECULAR DIFFUSION CONSTANT,0PF28.3/
     6                           11X,45HNUMBER OF CALCULATED VALUES BETWEEN
     7PRINTOUTS,I11/11X,44HNUMBER OF CONCENTRATION STEPS PER PRES. STEP,
     8I12)
 1052  FORMAT(5X,4HHEAD,I4,G15.5)
  960 FORMAT (///11X,24HIMPLICIT TIME DERIVATIVE///)
  970 FORMAT (///11X,24HCENTERED TIME DERIVATIVE///)
  980 FORMAT (I5,2F10.0)
  983 FORMAT(I5,3F10.0)
  990 FORMAT (////11X,16HNODE COORDINATES/11X,16(1H-)/)
 1000 FORMAT (1H ,2(11X,4HNODE,12X,1HX,15X,1HY,2X))
 1010 FORMAT (1H ,2(10X,I4,2F16.2)/(11X,I4,2F16.2,10X,I4,2F16.2))
 1020 FORMAT (/////11X,27HSOURCE(+) AND SINK(-) NODES/11X,27(1H-)/11X,4H
     1NODE,10X,9HDISCHARGE/)
 1030 FORMAT (11X,I4,F16.8)
 1040 FORMAT (/////11X,30HSOURCE AND SINK CONCENTRATIONS/11X,30(1H-)/11X
     1,4HNODE,7X,13HCONCENTRATION/)
 1051 FORMAT (5G15.7)
 1053 FORMAT(5X,13HCONCENTRATION,I4,G15.5)
 1070 FORMAT (/////,11X,12HINITIAL HEAD/11X,12(1H-))
 1080 FORMAT (/11X,6(4HNODE,5X,5HVALUE,5X))
 1090 FORMAT (/(11X,6(I4,2X,1PE10.3,3X)))
 1100 FORMAT (////11X,21HINITIAL CONCENTRATION/11X,21(1H-)/11X,6(4HNODE,
     15X,5HVALUE,5X)/(11X,6(I4,2X,1PE10.3,3X)))
 1110 FORMAT (I3,77A1)
 1120 FORMAT (1H0,10X,17HINVALID CHARACTER,2X,A1,5X,7HELEMENT,I5)
 1130 FORMAT (1H0,10X,25HINCOMPLETE DATA - ELEMENT,I5)
 1140 FORMAT (/////11X,18HELEMENT INCIDENCES/11X,18(1H-)/)
 1150 FORMAT (1H ,10X,7HELEMENT,5X,1H/,3(1H-),9H CORNERS ,4(1H-),1H/,7X,
     11H/,14(1H-),7H SIDES ,15(1H-),1H/)
 1160 FORMAT (1H ,4X,I10,5X,4I5,5X,8I5)
 1170 FORMAT (/////11X,18HELEMENT PARAMETERS/11X,18(1H-)//69X,12HDISPERS
```

```
     1IVITY/11X,88HELEMENT   X-MOBILITY   Y-MOBILITY   STORAGE   POROSITY   L
     2ONGITUDINAL   TRANSVERSE   THICKNESS//)
 1180 FORMAT (I3,7F10.3)
 1190 FORMAT (13X,I3,4X,1PE10.3,2X,1P2E10.3,3X,0PF4.2,5X,1PE10.3,3X,E10.
     13,2X,E10.3)
 1200 FORMAT (////11X,20HLEAKAGE COEFFICIENTS/11X,20(1H-)/11X,2(4HNODE,
     12X,11HCOEFFICIENT,3X,13HCONCENTRATION,3X)//(11X,2(I3,4X,1PE9.2,3X
     2,1PE9.2,7X)))
 1230 FORMAT (/////11X,33HDIRICHLET BOUNDARY NODES FOR FLOW/11X,33(1H-))
 1240 FORMAT (20I4)
 1250 FORMAT (11X,10(1H*),33H DIRICHLET BOUNDARY NODE FOR FLOW,I4,38HDOE
     1S NOT EXIST - EXECUTION TERMINATED ,10(1H*))
 1260 FORMAT (11X,20I5)
 1270 FORMAT (1H0,10(1H*),49H NUMBER OF DIRICHLET BOUNDARY NODES FOR FLO
     1W READ,I10,35H DISAGREES WITH NUMBER ANTICIPATED ,I10,10(1H*))
 1280 FORMAT (/////11X,42HDIRICHLET BOUNDARY NODES FOR CONCENTRATION/11X
     1,42(1H-))
 1290 FORMAT (11X,10(1H*),42H DIRICHLET BOUNDARY NODE FOR CONCENTRATION,
     1I4,38HDOES NOT EXIST - EXECUTION TERMINATED ,10(1H*))
 1300 FORMAT (1H0,5(1H*),58H NUMBER OF DIRICHLET BOUNDARY NODES FOR CONC
     1ENTRATION READ,I10, 35H DISAGREES WITH NUMBER ANTICIPATED ,I10,5(1
     2H*))
 1310 FORMAT (///11X,19HFINITE ELEMENT DATA/11X,19(1H-)/11X,21HTOTAL NUM
     1BER OF NODES,I19/11X,33HDIRICHLET BOUNDARY NODES FOR FLOW,I7/11X,2
     27HDEGREES OF FREEDOM FOR FLOW,I13/11X,34HDIRICHLET BOUNDARY NODES
     3FOR CONC.,I6/11X,28HDEGREES OF FREEDOM FOR CONC.,I12///)
 1320 FORMAT (1H1,/11X,40HVECTOR OF KNOWN VALUES FOR CONCENTRATION/11X,4
     10(1H-))
1350  FORMAT (/////11X,31HVECTOR OF KNOWN VALUES FOR HEAD/11X,31(1H-))
 1360 FORMAT (1H0,//////11X,16HTIME STEP NUMBER,I20/
     1 .      11X,17HTIME STEP (HOURS),1PE19.3/11X,12HELAPSED TIME,5X,1P
     2E19.3,6H HOURS/28X,1PE19.3,8H MINUTES/28X,1PE19.3,8H SECONDS)
 1370 FORMAT (//11X,14HHYDRAULIC HEAD/11X,14(1H-))
 1390 FORMAT (//11X,13HCONCENTRATION/11X,13(1H-))
 1400 FORMAT (/11X,114(1H*))
 1410 FORMAT (1H1)
 1420 FORMAT (/////11X,10(1H*),41HEXECUTION TERMINATED ON TIME-ELAPSED T
     1IME1PE12.4,6H HOURS10(1H*))
 1430 FORMAT (////11X,10(1H*),42HEXECUTION TERMINATED ON TIME STEPS AT S
     1TEPI10,10(1H*))
 1440 FORMAT (8F10.0)
 1450 FORMAT(1H0,///11X,21HPARAMETER MULTIPLIERS /11X,21(1H-)/11X,
     16HAFMOBX,F15.5/11X,6HAFMOBY,F15.5/11X,3HAST,F18.5/11X,4HAPOR,
     2F17.5/11X,6HABLONG,F15.5/11X,6HAETRAN,F15.5/11X,6HATHICK,F15.5/11X
     3,5HAPHII,F16.5/11X,6HACONCI,F15.5/11X,6HFACTOR,F15.5/11X,5HALEAK,
     1F16.5)
      END
```

```
      SUBROUTINE VCAL
C
C     PURPOSE - TO CALCULATE VELOCITIES AT ALL NODES AND AN AVERAGE
C               VELOCITY FOR EACH ELEMENT.   PRESSURE DERIVATIVES ARE
C               ALSO CALCULATED.
C
C     -----------------------------------------------------------------
      IMPLICIT INTEGER*2 (I-N)
      COMMON /SCALAR/
     1KOD1,  KOD2,  KOD3,  KOD4,  KOD5,  KOD6,  KOD7,  KOD11, KOD12,
     2KOD13, KOD14, KOD15, NN,    NE,    NP,    NB,    NB1,   KNB,
     3KNE1,  KNE2ST,KSTRT, KNBEST,KNB2
      COMMON /DSCALR/
     1CF,DIFUSN,ETRANS,ELONG,TCHK,SRCL,DELT,VMAX
      COMMON /ARRAY/
     1LR,    LRC,   IN,    KLR,   KLRC
      COMMON /DARRAY/
     1S,     CP,    CH,    CPM,   CRT,   CONC,  CONCI, COLD,  VX,
     1VY, VLX, VLY, PMOBX, PMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
     2DISPXY, PHI, P, PM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,
     3RT, COEF, HZERO, COR, DGX, DGY, Q, CFQ, 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),CPM(150), VX(12), VY(12),
     2COR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),
     3QP(16), SRCR(12), SRCRT(16), FQ(150), PM(150), CRT(150),
     4PHII(150), PMOBX(91), PMOBY(91), LR(150), LRC(150), Y(150),
     5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
     6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),
     ATHICK(91),ART(125),CCOEF(91)
      DIMENSION INDX(12),PP(12)
C     -----------------------------------------------------------------
C
      WRITE(6,210)
      L=0
   20 L=L+1
      WRITE(6,200) L
      DO 10 I=1,12
      VX(I)=0.
   10 VY(I)=0.
      M=IN(13,L)
      XCOND=PMOBX(L)/POR(L)
      YCOND=PMOBY(L)/POR(L)
      NCOUNT=0
      I=0
   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.
      GO 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 CONTINUE
      XI=COR(I+3)
      YI=COR(I+1)
      GO TO 70
   60 CONTINUE
      I2=I/2
```

```
      XI=COR(I2)
      YI=COR(I2-1)
   70 CONTINUE
C     ******************************
      CALL SHAPE (L,N,XI,YI,FP,DSF)
C     ******************************
      NCCUNT=NCCUNT+1
      EX=C.
      EY=C.
      J=C
      DO 90 K=1,N
   8C J=J+1
      IF (IN(J,L).EQ.0) GO TO 80
      J1=IN(J,L)
C
C     COMPUTE PRESSURE DERIVATIVES
      EX=EX+DGX(K)*FN(J1)
      EY=EY+DGY(K)*FN(J1)
   90 CONTINUE
      INDX(NCCUNT)=IN(I,L)
      VY(NCCUNT)=-YCOND*EY
      VX(NCCUNT)=-XCCND*EX
  11C CCNTINUE
      IF (N.EQ.4.AND.I.EQ.4) GO TO 130
      IF (I.LT.12) GO TO 30
  13C CONTINUE
      WRITE (6,220) (INDX(I),VX(I),VY(I),I=1,N)
      IF (L.LT.NE) GO TO 20
  170 RETURN
C
  21C FORMAT(///11X,16HPOINT VELOCITIES/11X,16(1H-))
  200 FORMAT( /11X,7HELEMENT,I8/11X,3(4HNODE,2X,11HX-COMPONENT,2X,11HY-C
     1CMPCNENT,6X) )
  22C FORMAT ((11X,3(I4,2X,1PE10.3,3X,1PE10.3,7X)))
      END
```

```
      SUBROUTINE MATGEN (I,J,L,M)

      PURPOSE - TO GENERATE ELEMENT MATRICES

      ----------------------------------------------------------------------
      IMPLICIT INTEGER*2 (I-N)
      COMMON /SCALAR/
     1KOD1,  KOD2,  KOD3,  KOD4,  KOD5,  KOD6,  KOD7,  KOD11, KOD12,
     2KOD13, KOD14, KOD15, NN,    NE,    NP,    NB,    NB1,   KNB,
     3KNB1,  KNB2ST,KSTRT, KNBEST,KNB2
      COMMON /DSCALR/
     1CF,DIFUSN,ETRANS,ELONG,TCHK,SRCL,DELT,VMAX
      COMMON /ARRAY/
     1LR,    LRC,   IN,    KLR,   KLRC
      COMMON /DARRAY/
     1S,     CP,    CH,    CPM,   CRT,   CONC,  CONCI, COLD,  VX,
     1VY, VLX, VLY, FMOBX, FMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
     2DISPXY, PHI, P, PM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,
     3RT, CCEF, HZERO, COR, DGX, DGY, Q, CFQ, 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),CPM(150), VX(12), VY(12),
     2CCR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),
     3QP(16), SRCR(12), SRCRT(16), FQ(150), PM(150), CRT(150),
     4PHII(150), FMOBX(91), FMOBY(91), LR(150), LRC(150), Y(150),
     5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
     6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),
     ATHICK(91),ART(125),CCOEF(91)
      DATA H1/.347854845137454/,H2/.652145154862546/
C     ----------------------------------------------------------------------
C
      H11=H1*H1
      H12=H1*H2
      H22=H2*H2
      IF (NE.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(
     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)+SRC
     1LT(3)+SRCLT(5)+SRCLT(8)+SRCLT(9)+SRCLT(12)+SRCLT(15)+SRCLT(14))+H2
     22*(SRCLT(6)+SRCLT(7)+SRCLT(10)+SRCLT(11))
   50 CONTINUE
      RETURN
C
C
      ENTRY SECOND(I)
      IF (NE.EQ.4) GO TO 60
      SRCR(I)=SRCRT(1)+SRCRT(2)+SRCRT(3)+SRCRT(4)
      GO TO 70
   60 SRCR(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
```

```
      ENTRY THIRD
      WRITE (6,130) L
      MM2=(M+7)/8*8-7
      DO 90 K1=1,MM2,8
      K2=K1+7
      IF (K1.EQ.MM2) K2=M
      DO 80 I=1,M
   80 WRITE (6,140) I,(SE(I,J),J=K1,K2)
   90 WRITE (6,120)
      WRITE (6,150) L
      MM2=(M+7)/8*8-7
      DO 110 K1=1,MM2,8
      K2=K1+7
      IF (K1.EQ.MM2) K2=M
      DO 100 I=1,M
  100 WRITE (6,140) I,(PE(I,J),J=K1,K2)
  110 WRITE (6,120)
      RETURN
C
  120 FORMAT (//)
  130 FORMAT (////,10X,7HELEMENT,I4,5X,16HSTIFFNESS MATRIX/)
  140 FORMAT (I5,8E15.6)
  150 FORMAT (1H ,9X,7HELEMENT,I4,5X,14HSTORAGE MATRIX/)
      END
```

```
      SUBROUTINE SBAND (N)
C     ------------------------------------------------------------------
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 N
      COMMON /SCALAR/
     1KOD1,  KOD2,  KOD3,  KOD4,  KOD5,  KOD6,  KOD7,  KOD11, KOD12,
     2KOD13, KOD14, KOD15, NN,    NE,    NP,    NB,    NB1,   KNB,
     3KNB1,  KNB2ST,KSTRT, KNBEST,KNB2
      COMMON /DSCALR/
     1CF,DIFUSN,ETRANS,ELONG,TCHK,SRCL,DELT,VMAX
      COMMON /ARRAY/
     1LR,    LRC,   IN,    KLR,   KLRC
      COMMON /DARRAY/
     1S,     CP,    CH,    CPM,   CRT,   CONC,  CONCI, COLD,  VX,
     1VY, VLX, VLY, PMOBX, PMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
     2DISPXY, PHI, P, FM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,
     3RT, COEF, HZERO, COR, DGX, DGY, Q, CFQ, 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),CPM(150), VX(12), VY(12),
     2COR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),
     3QP(16), SRCR(12), SRCRT(16), FQ(150), FM(150), CRT(150),
     4PHII(150), PMOBX(91), PMOBY(91), LR(150), LRC(150), Y(150),
     5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
     6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),
     &THICK(91),ART(125),CCOEF(91)
C     ------------------------------------------------------------------
      DO 30 I=1,N
      J=I-NB+1
      IF ((I+1).LE.NB) J=1
      SUM=FM(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)*PHI(K)
   20 PHI(I)=SUM*S(I,1)
   30 CONTINUE
      DO 60 I1=1,N
      I=N-I1+1
      J=I+NB-1
      IF (J.GT.N) J=N
      SUM=PHI(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)*PHI(K)
   50 PHI(I)=SUM*S(I,1)
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DBAND (N,IEX)
C     ----------------------------------------------------------------
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 N
      COMMON /SCALAR/
     1KOD1,  KOD2,  KOD3,  KOD4,  KOD5,  KOD6,  KOD7,  KOD11, KOD12,
     2KOD13, KOD14, KOD15, NN,    NE,    NP,    NB,    NB1,   KNB,
     3KNB1,  KNB2ST,KSTRT, KNBEST,KNB2
      COMMON /DSCALR/
     1CF,DIFUSN,ETRANS,ELONG,TCHK,SRCL,DELT,VMAX
      COMMON /ARRAY/
     1LR,    LRC,   IN,    KLR,   KLRC
      COMMON /DARRAY/
     1S,     CP,    CH,    CPM,   CRT,   CONC,  CONCI, COLD,  VX,
     1VY, VLX, VLY, PMOBX, PMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
     2DISPXY, PHI, P, PM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,
     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),CPM(150), VX(12), VY(12),
     2COR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),
     3CF(16), SRCR(12), SRCRT(16), FQ(150), PM(150), CRT(150),
     4PHII(150), PMOBX(91), PMOBY(91), LR(150), LRC(150), Y(150),
     5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
     6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CPQ(150),
     ATHICK(91),ART(125),CCOEF(91)
C     ----------------------------------------------------------------
      IEX=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-S(II,K+1)*S(II,JZ)
   20 IF (J.NE.1) GO TO 40
      IF (SUM.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
      RETURN
   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 SHAPE (L, M, XI, YI,  F, DET)
C     ----------------------------------------------------------------
      IMPLICIT INTEGER*2 (I-N)
      COMMON /SCALAR/
     1KOD1,  KOD2,  KOD3,  KOD4,  KOD5,  KOD6,  KOD7,  KOD11, KOD12,
     2KOD13, KOD14, KOD15, NN,    NE,    NP,    NB,    NB1,   KNB,
     3KNB1,  KNB2ST,KSTRT, KNBEST,KNB2
      COMMON /DSCAL2/
     1CE,DIFUSN,ETRANS,ELONG,TCHK,SRCL,DELT,VMAX
      COMMON /ARRAY/
     1LR,    LRC,   IN,    KLR,   KLRC
      COMMON /DARRAY/
     1S,     CP,    CH,    CPM,   CRT,   CONC,  CONCI, COLD,  VX,
     1VY, VLX, VLY, FMOBX, FMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
     2DISPXY, PHI, P, FM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,
     3RT, CCEF, HZERO, COR, DGX, DGY, Q, CFQ, 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),CPM(150), VX(12), VY(12),
     2COR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),
     3QP(16), SRCR(12), SRCRT(16), FQ(150), FM(150), CRT(150),
     4PHII(150), FMOBX(91), FMOBY(91), LR(150), LRC(150), Y(150),
     5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
     6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),
     ATHICK(91),ART(125),CCOEF(91)
      DIMENSION ALF(4),   DAX(4),   DAY(4),   BTX(4),   BTY(4),
     1DBX(4),   DFX(12),  DFY(12),  F(12),    DBY(4)
C     ----------------------------------------------------------------
C
      XI1=1.-XI
      XI2=1.+XI
      YI1=1.-YI
      YI2=1.+YI
C
C     CORNER NODE SHAPE FUNCTIONS, BASIC PART
      ALF(1)=.25*XI1*YI1
      ALF(2)=.25*XI2*YI1
      ALF(3)=.25*XI2*YI2
      ALF(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
      XQ1=XI-.5
      XQ2=-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 (IN(J3,L).EQ.0) GO TO 10
      IF (IN(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)=XC2
      BTX(J2)=XC1
      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 (IN(J3,L).EQ.0) GO TO 60
      IF (IN(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 90
   70 CONTINUE
      BTY(J1)=YC2
      BTY(J2)=YC1
      DBY(J1)=-1.
      DBY(J2)=1.
      GO TO 90
   80 CONTINUE
      BTY(J1)=YC1
      BTY(J2)=YC1
      DBY(J1)=YC2
      DBY(J2)=YC2
   90 CONTINUE
      J1=1
      J2=4
      J3=11
  100 CONTINUE
C
C     SHAPE FUNCTION DERIVATIVE MATRIX - CORNER NODES
      DO 110 J=1,4
      DFX(J)=DAX(J)*(BTX(J)+BTY(J))+DBX(J)*ALP(J)
      DFY(J)=DAY(J)*(BTX(J)+BTY(J))+DBY(J)*ALP(J)
      F(J)=ALP(J)*(BTX(J)+BTY(J))
```

```
  110 CONTINUE
C
C     SHAPE FUNCTION DERIVATIVE MATRIX - EDGE NODES
      IF (M.EQ.4) GO TO 240
      J=4
      XEQ=1.-XI*XI
      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
      DFX(J)=-XI*YI1
      DFY(J)=-.5*XEQ
      F(J)=.5*XEQ*YI1
      GO TO 140
  130 J=J+1
      DFX(J)=-.28125*YI1*(3.*XEQ+2.*XI*XE1)
      DFY(J)=-.28125*XEQ*XE1
      F(J)=.28125*XEQ*XE1*YI1
      J=J+1
      DFX(J)=.28125*YI1*(3.*XEQ-2.*XI*XE2)
      DFY(J)=-.28125*XEQ*XE2
      F(J)=.28125*XEQ*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
      DFX(J)=.5*YEQ
      DFY(J)=-YI*XI2
      F(J)=.5*XI2*YEQ
      GO TO 170
  160 J=J+1
      DFX(J)=.28125*YEQ*YE1
      DFY(J)=-.28125*XI2*(3.*YEQ+2.*YI*YE1)
      F(J)=.28125*XI2*YEQ*YE1
      J=J+1
      DFX(J)=.28125*YEQ*YE2
      DFY(J)=.28125*XI2*(3.*YEQ-2.*YI*YE2)
      F(J)=.28125*XI2*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
      DFX(J)=-XI*YI2
      DFY(J)=.5*XEQ
      F(J)=.5*XEQ*YI2
      GO TO 200
  190 J=J+1
      DFX(J)=.28125*YI2*(3.*XEQ-2.*XI*XE2)
      DFY(J)=.28125*XEQ*XE2
      F(J)=.28125*XEQ*XE2*YI2
      J=J+1
      DFX(J)=-.28125*YI2*(3.*XEQ+2.*XI*XE1)
      F(J)=.28125*XEQ*XE1*YI2
      DFY(J)=.28125*XEQ*XE1
  200 IF (IN(11,L).EQ.0) GO TO 230
```

```
      IF (IN(12,L).EQ.0) GO TO 210
      GO TO 220
  210 J=J+1
      DFX(J)=-.5*YEQ
      DFY(J)=-YI*XI1
      F(J)=.5*XI1*YEQ
      GO TO 230
  220 J=J+1
      DFX(J)=-.28125*YEQ*YE2
      DFY(J)=.28125*XI1*(3.*YEQ-2.*YI*YE2)
      F(J)=.28125*XI1*YEQ*YE2
      J=J+1
      DFX(J)=-.28125*YEQ*YE1
      DFY(J)=-.28125*XI1*(3.*YEQ+2.*YI*YE1)
      F(J)=.28125*YEQ*YE1*XI1
  230 CONTINUE
  240 CONTINUE
C
C     JACOBIAN
      SUM1=0.
      SUM2=0.
      SUM3=0.
      SUM4=0.
      K=0
      DO 260 I=1,N
  250 K=K+1
      IF (IN(K,L).EQ.0) GO TO 250
      KI=IN(K,L)
      SUM1=SUM1+DFX(I)*X(KI)
      SUM2=SUM2+DFX(I)*Y(KI)
      SUM3=SUM3+DFY(I)*X(KI)
      SUM4=SUM4+DFY(I)*Y(KI)
  260 CONTINUE
      DET=SUM1*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*DFX(J)+C12*DFY(J)
      DGY(J)=C21*DFX(J)+C22*DFY(J)
  270 CONTINUE
      RETURN
      END
```

```
240 ...IVE MATRIX - EDGE NODES

I*XI
*Y
*X
*XI
*YI
*YI
).EQ.0) GO TO 140
).EQ.0) GO TO 120

YI1
XEQ
*YI1

25*YI1*(3.*XEQ+2.*XI*XE1)
25*XEQ*XE1
XEQ*XE1*YI1

*YI1*(3.*XEQ-2.*XI*XE2)
5*XEQ*XE2
EQ*XE2*YI1
.0) GO TO 170
.0) GO TO 150

Q*YE1
I2*(3.*YEQ+2.*YI*YE1)
YEQ*YE1

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

EQ-2.*XI*XE2)

Q+2.*XI*XE1)
```

```
01232000  ----------------
01233000
01234000  KOD11, KOD12,
01235000  NB1,   KNB,
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),0135
01249000 0), CRT(150),
01250000 RC(150), Y(150),           0135
01251000 )(150), POR(91), ST(91),01360
01252000 , ETRANS(91), CPQ(150),  01361
01253000                           01362
01254000 --------------------------01363
01255000 333333333,.33333333333,1.013640
01256000 -1./                      013650
01257000                           013660
01258000                           013670
01259000                           013680
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
```

```
      SUBROUTINE SOLVE (KKK,C,R,NNP,IHALFB,MAXNP,MAXBW)
      DIMENSION C(MAXNP,MAXBW),R(MAXNP)
      IHBP=IHALFB+1
      IF(KKK.EQ.2 ) GO TO 50
      NU=NNP-IHALFB
      DO 20 NI=1,NU
      PIVOTI=1./C(NI,IHBP)
      NJ=NI+1
      IB=IHBP
      NK=NI+IHALFB
      DO 10 NL=NJ,NK
      IB=IB-1
      A=-C(NL,IB)*PIVOTI
      C(NI,IB)=A
      JB=IB+1
      KB=IB+IHALFB
      LB=IHBP-IB
      DO10 MB=JB,KB
      NB=IB+MB
   10 C(NI,MB)=C(NL,MB)+A*C(NI,NB)
   20 CONTINUE
      NB=NU+1
      NU=NNP-1
      NK=NNP
      DO 40 NI=NB,NU
      PIVOTI=1./(C(NI,IHBP))
      NJ=NI+1
      IB=IHBP
      DO 30 NL=NJ,NK
      IB=IB-1
      A=-C(NL,IB)*PIVOTI
      C(NI,IB)=A
      JB=IB+1
      KB=IB+IHALFB
      LB=IHBP-IB
      DO 30 MB=JB,KB
      NB=IB+MB
   30 C(NI,MB)=C(NL,MB)+A*C(NI,NB)
   40 CONTINUE
      RETURN
C
   50 NU=NNP+1
      IBAND=2*IHALFB+1
      DO 70 NI=2,IHBP
      IB=IHBP-NI+1
      NJ=1
      SUM=0.0
      DO 60 JB=IB,IHALFB
      SUM=SUM+C(NI,JB)*R(NJ)
   60 NJ=NJ+1
   70 R(NI)=R(NI)+SUM
      IB=1
      NL=IHBP+1
      DO 90 NI=NL,NNP
      NJ=NI-IHBP+1
      SUM=0.
      DO 80 JB=IB,IHALFB
      SUM=SUM+C(NI,JB)*R(NJ)
   80 NJ=NJ+1
   90 R(NI)=R(NI)+SUM
```

```
C   BACK SOLVE
      R(NNP)=R(NNP)/C(NNP,IHBP)
      DO 110 IB=2,IHBP
      NI=NU-IB
      NJ=NI
      ME=IHALFB+IB
      SUM=0.
      DO 100 JB=NL,ME
      NJ=NJ+1
100   SUM=SUM+C(NI,JB)*R(NJ)
110   R(NI)=(R(NI)-SUM)/C(NI,IHBP)
      ME=IBAND
      DO 130 IB=NL,NNP
      NI=NU-IB
      NJ=NI
      SUM=0.
      DO 120 JB=NL,ME
      NJ=NJ+1
 120  SUM=SUM+C(NI,JB)*R(NJ)
130   R(NI)=(R(NI)-SUM)/C(NI,IHBP)
      RETURN
      END
```

```
      SUBROUTINE COGEN (JTEST,N,KN,IT)
C
C     PURPOSE - TO GENERATE GLOBAL COEFFICIENT MATRICES FOR FLOW
C              AND CONCENTRATION
C
C     ------------------------------------------------------------------
      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, NN,    NE,    NP,    NB,    NB1,   KNB,
     3KNB1,  KNB2ST,KSTRT, KNB2ST,KNB2
      COMMON /DSCALR/
     1CF,DIFUSN,ETRANS,ELONG,TCHK,SRCL,DELT,VMAX
      COMMON /ARRAY/
     1LR,    LRC,    IN,     KLR,    KLRC
      COMMON /DARRAY/
     1S,     CP,     CH,     CPM,    CRT,    CONC,   CONCI,  COLD,  VX,
     1VY, VLX, VLY, FMOBX, FMOBY, POR, ST, QP, SE, PE, DISPX, DISPY,
     2DISPXY, PHI, P, FM, OLD, H, SRCLT, SRCR, SRCRT, PHII, Y, X, FQ,
     3RT, COEF, HZERO, COR, DGX, DGY, Q, CFQ, 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),CPM(150), VX(12), VY(12),
     2COR(15), DGX(12), DGY(12), SRCLT(16), SE(12,12), Q(16), PE(12,12),
     3CF(16), SRCR(12), SRCRT(16), FQ(150), FM(150), CRT(150),
     4PHII(150), FMOBX(91), FMOBY(91), LR(150), LRC(150), Y(150),
     5X(150), IN(13,91), RT(150), COEF(91), HZERO(150), POR(91), ST(91),
     6KLR(150), KLRC(150), CONCI(150), ELONG(91), ETRANS(91), CFQ(150),
     ATHICK(91),ART(125),CCOEF(91)
      DIMENSION AG(6),      F(12,16), FF(12),    DX(12,16),DY(12,16),
     1DETJ(16),JD(24),VKY(16),VKX(16),CPHI(16),CK(16)
      DIMENSION HK(16)
      DATA AG/-.577350269189626, .577350269189626, -.861136311594053,
     1-.339981043584856, .339981043584856, .861136311594053/
C     ------------------------------------------------------------------
C
      KII=0
      II=0
      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)=-FQ(I)
   40 CONTINUE
   50 IF (JTEST.EQ.1) GO TO 100
      DO 60 I=1,KN
      DO 60 J=1,KNB2ST
      CP(I,J)=0.
   60 CH(I,J)=0.
      DO 90 I=1,NN
      IF (KLR(I).GT.0) GO TO 90
      KII=KII+1
      IF (CFQ(I)) 80,70,70
   70 CRT(KII)=-FQ(I)*CFQ(I)
      GO TO 90
   80 CRT(KII)=0
```

```
   90 CONTINUE
  100 CONTINUE
      VMAX=0.
      L=0
  110 L=L+1
      M=IN(13,L)
      NP=2
      IF (M.GT.4) NP=4
      NP2=NP*NP
C
C     SHAPE FUNCTIONS FOR INTEGRATION POINTS
C     INTEGRATION BY GAUSSIAN QUADRATURE
C     2 X 2 RULE FOR FULLY LINEAR ELEMENTS
C     4X4 RULE FOR ALL OTHER ELEMENTS
      DO 150 I=1,NP
      DO 150 J=1,NP
      K=(I-1)*NP+J
      IF (NP.EQ.4) GO TO 120
      XI=AG(J)
      YI=AG(I)
      GO TO 130
  120 XI=AG(J+2)
      YI=AG(I+2)
  130 CONTINUE
C     ******************************
      CALL SHAPE (L,M,XI,YI,FF,DET)
C     ******************************
      DO 140 JJ=1,M
      F(JJ,K)=FF(JJ)
      DX(JJ,K)=DGX(JJ)
  140 DY(JJ,K)=DGY(JJ)
      DETJ(K)=DET
  150 CONTINUE
C
C     COMPUTE FLOW COEFFICIENT MATRICES
      K=0
      DO 170 I=1,M
  160 K=K+1
      IF (IN(K,L).EQ.0) GO TO 160
      JD(I)=IN(K,L)
  170 CONTINUE
C
C     THE ARRAY JD NOW CONTAINS THE INCIDENCES OF THE ACTIVE NODES IN
C     ELEMENT L.
      IF (JTEST.EQ.0) GO TO 290
      IF (COEF(L).EQ.0.) GO TO 793
      DO 792 K=1,NP2
      HK(K)=0.
      DO 792 J=1,M
      JDJ=JD(J)
  792 HK(K)=HK(K)+F(J,K)*HZERO(JDJ)
  793 CONTINUE
      DO 190 I=1,M
      DO 190 J=I,M
      DO 180 K=1,NP2
      SRCIT(K)=COEF(L)*F(I,K)*F(J,K)*DETJ(K)
      QP(K)=ST(L)*F(I,K)*F(J,K)*DETJ(K)
      Q(K)=(FMOBX(L)*DX(I,K)*DX(J,K)+FMOBY(L)*DY(I,K)*DY(J,K))*DETJ(K)
  180 CONTINUE
C     *******************
```

```
      CALL MATGEN (I,J,L,M)
C     ********************
  190 SE(I,J)=SE(I,J)+SRCL
      DO 210 I=1,M
      DO 200 K=1,NP2
200   SRCFT(K)=COEF(I)*HK(K)*F(I,K)*DETJ(K)
C     ***************
  210 CALL SECOND (I)
C     ***************
C
C     FILL LOWER HALF OF SE AND PE ARRAYS
      DO 220 I=2,M
      I1=I-1
      DO 220 J=1,I1
      PE(I,J)=PE(J,I)
  220 SE(I,J)=SE(J,I)
C
C     PRINT ELEMENT MATRICES FOR FLOW
      IF (KOD1.NE.1) GO TO 230
      WRITE (6,490)
C     **********
      CALL THIRD
C     **********
  230 CONTINUE
C
C     ASSEMBLY OF GLOBAL COEFFICIENT MATRIX
      DO 280 I=1,M
      JDI=JD(I)
      IF (LB(JDI).GT.0) GO TO 260
      IR=JDI-LRC(JDI)
      RT(IR)=RT(IR)-SRCE(I)
      DO 250 J=1,M
      JDJ=JD(J)
      IF (LB(JDJ).GT.0) GO TO 250
      JC=JDJ-LRC(JDJ)-IR+1
      IF (JC.LE.NB) GO TO 240
      WRITE (6,560) L,JC,NB
      GO TO 250
  240 IF (JC.LT.1) GO TO 250
      P(IR,JC)=P(IR,JC)+PE(I,J)
      H(IR,JC)=H(IR,JC)+SE(I,J)
      IF (JC.GT.NB1) NB1=JC
  250 CONTINUE
      GO TO 280
  260 CONTINUE
      DO 270 J=1,M
      JDJ=JD(J)
      IF (LB(JDJ).GT.0) GO TO 270
      JC=JDJ-LRC(JDJ)
      RT(JC)=RT(JC)+SE(J,I)*PHII(JDI)
  270 CONTINUE
  280 CONTINUE
  290 CONTINUE
      IF (JTEST.EQ.1) GO TO 380
C
C     COMPUTE CONCENTRATION COEFFICIENT MATRICES
      VLX=0.
      VLY=0.
      XCOND=-FMOBX(L)/POR(L)
      YCOND=-FMOBY(L)/POR(L)
```

```
      DO 292 K=1,NP2
      VKY(K)=0.
      VKX(K)=0.
      CK(K)=0.
      CPHI(K)=0.
      DO 291 J=1,M
      JDJ=JD(J)
      CK(K)=CK(K)+F(J,K)*CPM(JDJ)
      CPHI(K)=CPHI(K)+F(J,K)*FM(JDJ)
      VKX(K)=VKX(K)+XCOND*DX(J,K)*FM(JDJ)
  291 VKY(K)=VKY(K)+YCOND*DY(J,K)*FM(JDJ)
  295 VLX=VLX+VKX(K)
      VLY=VLY+VKY(K)
  292 CONTINUE
      VLY=VLY/(NP2*THICK(L))
      VLX=VLX/(NP2*THICK(L))
      VMAX=AMAX1(VMAX,ABS(VLX),ABS(VLY))
      DO 310 I=1,M
      DO 310 J=1,M
      DO 300 K=1,NP2
      VXSQRD=VKX(K)*VKX(K)
      VYSQRD=VKY(K)*VKY(K)
      VSQRD=VXSQRD+VYSQRD
      IF (VSQRD.NE.0.) GO TO 723
      DISPX=DIFUSN
      DISPY=DIFUSN
      DISPXY=0.
      GO TO 724
  723 V=SQRT(VSQRD)
      DL=ELONG(L)*V
      DT=ETRANS(L)*V
      DVX=VXSQRD/VSQRD
      DVY=VYSQRD/VSQRD
      DISPX=DL*DVX+DT*DVY+DIFUSN
      DISPY=DT*DVX+DL*DVY+DIFUSN
      DISPXY=(DL-DT)*(VKX(K)*VKY(K)/VSQRD)
  724 SECLT(K)=0.
      IF (CCOEF(L).GT.0.) SECLT(K)=COEF(L)*(CPHI(K)-HK(K))*F(I,K)*F(J,K)
     1*DETJ(K)*0.5
      QP(K)=POR(L)*THICK(L)*F(I,K)*F(J,K)*DETJ(K)
      Q(K)=((DISPX*DX(I,K)*DX(J,K)+DISPY*DY(I,K)*DY(J,K)+DISPXY
     1*(DY(I,K)*DX(J,K)+DX(I,K)*DY(J,K))+(VKX(K)*DX(J,K)+VKY(K)*DY(J,K))
     2*F(I,K))*POR(L))*DETJ(K)
  300 CONTINUE
C     ********************
      CALL [redacted])
C     ********************
  310 SE(I,J)=SE(I,J)+SECL
      DO 336 I=1,M
      IF (CCOEF(L).LE.0.) GO TO 337
      DO 312 K=1,NP2
  312 SECST(K)=-COEF(L)*(HK(K)-CPHI(K))*CCOEF(L)*F(I,K)*DETJ(K)
C     **************
  311 CALL SECOND(I)
C     **************
      GO TO 336
  337 SECS(I)=0.
  336 CONTINUE
      IF (KOD3.NE.1) GO TO 320
C
```

```
C      PRINT ELEMENT MATRICES FOR CONCENTRATION
       IF (KOD3.NE.1) GO TO 320
       WRITE (6,500)
C      **********
       CALL THIRD
C      **********
  320  CONTINUE
C
C      ASSEMBLY OF GLOBAL COEFFICIENT MATRIX
       DO 370 I=1,N
       JDI=JD(I)
       IF (KIR(JDI).GT.0) GO TO 350
       IR=JDI-KLRC(JDI)
       CRT(IR)=CRT(IR)-SRCR(I)
       DO 340 J=1,N
       JDJ=JD(J)
       IF (KIR(JDJ).GT.0) GO TO 340
       JC=JDJ-KLRC(JDJ)-IR+KNBEST
       IF (JC.LE.KNB2ST) GO TO 330
       WRITE (6,510) L,JC,KNB2
       STOP
  330  CH(IR,JC)=CH(IR,JC)+SE(I,J)
       CP(IR,JC)=CP(IR,JC)+PE(I,J)
       IF (JC.GT.KNB1) KNB1=JC
  340  CONTINUE
       GO TO 370
  350  CONTINUE
       DO 360 J=1,N
       JDJ=JD(J)
       IF (KIR(JDJ).GT.0) GO TO 360
       JC=JDJ-KLRC(JDJ)
       CRT(JC)=CRT(JC)+SE(J,I)*CONCI(JDI)
  360  CONTINUE
  370  CONTINUE
  380  IF (L.LT.NE) GO TO 110
       NB=NB1
       KNBOLD=KNB
       KNB=KNB1-KNBEST+1
       KSIFT=KNB2ST-KNB1
       KNB2=2*KNB-1
       IF (IT.LE.1) WRITE (6,520) NB,KNB
       IF (JTEST.EQ.0) GO TO 430
       IF (KOD2.NE.1) GO TO 430
       WRITE (6,530)
       WRITE (6,540)
       NB2=(NB+9)/10*10-9
       DO 400 K1=1,NB2,10
       K2=K1+9
       IF (K1.EQ.NB2) K2=NB
       DO 390 I=1,N
  390  WRITE (6,570) I,(H(I,J),J=K1,K2)
  400  WRITE (6,550)
       WRITE (6,580)
       NB2=(NB+9)/10*10-9
       DO 420 K1=1,NB2,10
       K2=K1+9
       IF (K1.EQ.NB2) K2=NB
       DO 410 I=1,N
  410  WRITE (6,570) I,(P(I,J),J=K1,K2)
  420  WRITE (6,550)
```

```
      WRITE (6,590)
      WRITE (6,600) (RT(I),I=1,N)
  430 IF (JTEST.EQ.1) GO TO 480
      IF (KCD4.NE.1) GO TO 480
      WRITE (6,610)
      WRITE (6,620)
      NB2=(KNB2+9)/10*10-9
      DO 450 K1=1,NB2,10
      K2=K1+9
      IF (K1.EQ.NB2) K2=KNB2
      DO 440 I=1,KN
  440 WRITE (6,570) I,(CH(I,J+KSTAR),J=K1,K2)
  450 WRITE (6,550)
      WRITE (6,630)
      DO 470 K1=1,NB2,10
      K2=K1+9
      IF (K1.EQ.NB2) K2=KNB2
      DO 460 I=1,KN
  460 WRITE (6,570) I,(CP(I,J+KSTAR),J=K1,K2)
  470 WRITE (6,550)
      WRITE (6,590)
      WRITE (6,600) (CRT(I),I=1,KN)
  480 CONTINUE
      RETURN
C
  490 FORMAT (1H1,////11X,25HELEMENT MATRICES FOR FLOW/11X,25(1H-))
  500 FORMAT (1H1,////11X,34HELEMENT MATRICES FOR CONCENTRATION/11X,34(1
     1H-))
  510 FORMAT (1H0,10X,7HELEMENT,I4,5X,33HINSUFFICIENT BAND WIDTH - REQUI
     1RE,I5,2X,10HINSTEAD OF,I5)
  520 FORMAT (1H0,10X,41HGLOBAL COEFFICIENT MATRIX HALF BAND WIDTH/11X,8
     1HFOR FLOW,I15/11X,17HFOR CONCENTRATION,I6)
  530 FORMAT (1H1,10X,36HGLOBAL COEFFICIENT MATRICES FOR FLOW/11X,36(1H-
     1))
  540 FORMAT (1H0,10X,37HA COEFFICIENT MATRIX - UPPER HALFBAND/11X,37(1H
     1-)//)
  550 FORMAT (//)
  560 FORMAT (1H0,10X,7HELEMENT,I4,5X,38HINSUFFICIENT HALF-BAND WIDTH - 
     1REQUIRE,I5,2X,10HINSTEAD OF,I5)
  570 FORMAT (I5,10E12.4)
  580 FORMAT (1H0,10X,37HP COEFFICIENT MATRIX - UPPER HALFBAND/11X,37(1H
     1-)//)
  590 FORMAT (/////,11X,20HP COEFFICIENT MATRIX/11X,20(1H-)//)
  600 FORMAT (11X,10E12.3)
  610 FORMAT (1H1,10X,37HGLOBAL COEFFICIENT MATRICES FOR CONC./11X,37(1H
     1-))
  620 FORMAT (1H0,10X,20HA COEFFICIENT MATRIX/11X,20(1H-)//)
  630 FORMAT (1H0,10X,20HP COEFFICIENT MATRIX/11X,20(1H-)//)
      END
```