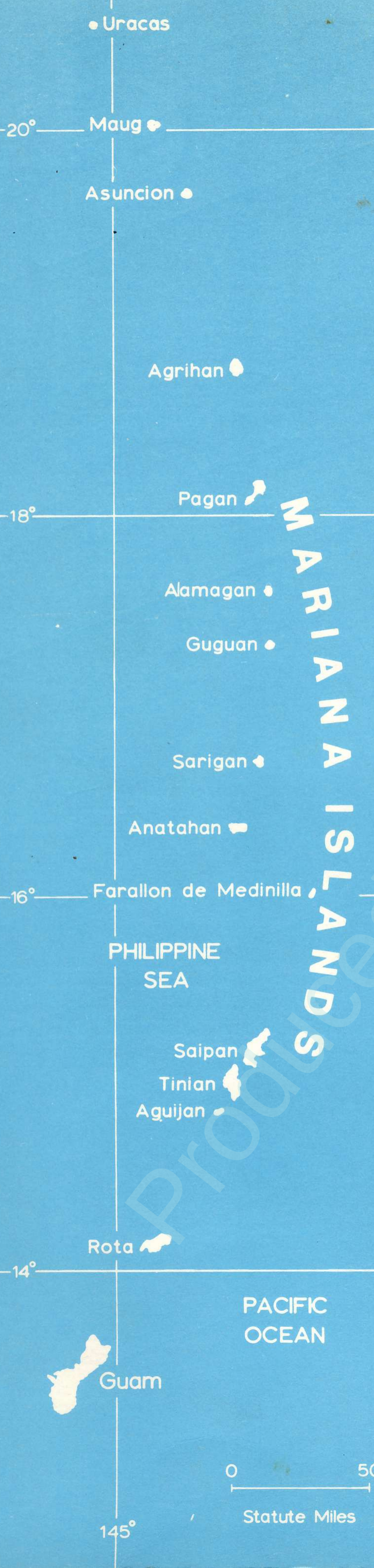

A TWO-DIMENSIONAL, FINITE ELEMENT MODEL OF SALT WATER INTRUSION IN GROUNDWATER SYSTEMS

By
Dinshaw N. Contractor

UNIVERSITY OF GUAM

Water and Energy Research Institute of the Western Pacific

Technical Report No. 26

October, 1981

A TWO-DIMENSIONAL, FINITE ELEMENT MODEL

OF

SALT WATER INTRUSION IN GROUNDWATER SYSTEMS

By

Dinshaw N. Contractor

UNIVERSITY OF GUAM

Water and Energy Research Institute
of the
Western Pacific

Technical Report No. 26

October, 1981

The work on which this report is based was supported in part by funds provided by the Guam Environmental Protection Agency in accordance with the Mathematical Modeling Sector of the Northern Guam Lens Study.

ABSTRACT

A two-dimensional, finite element model has been developed to simulate saltwater intrusion into coastal and insular groundwater aquifers. The depth-averaged equations of continuity for fresh water and salt water are solved simultaneously with the assumption of a sharp interface separating the two liquids. The aquifer is discretized into linear, triangular elements. The Galerkin weighted residual method was used to develop element equations. Boundary conditions of the Dirichlet and Neumann type can be applied. A third type (Cauchy) boundary condition is used where freshwater exits along the coastline.

A computer program has been written in FORTRAN to solve for the freshwater and saltwater heads at each node of the finite element grid. The program can simulate steady or unsteady conditions in phreatic or confined aquifers. Wells can be specified at any node. The pump rate, aquifer recharge rate and boundary conditions can be specified as functions of time. The program can track the location of the saltwater toes (where the interface touches the impervious basement) and the freshwater toe (where the phreatic surface touches the impervious basement). The program can provide the magnitude and direction of the fresh and salt water in each element of the grid. The program can also be used to simulate aquifers that have only fresh water, by specifying the saltwater head at all nodes to be much lower than the expected fresh water head.

The solution has been checked for accuracy against the following analytic solutions:

1) *Steady, one dimensional saltwater intrusion in confined and unconfined aquifers.*

2) *Unsteady drawdown due to pumping in a confined aquifer. (Theis solution). Saltwater heads are assumed to be zero everywhere.*

3) *One-dimensional gravitational, segregation problem.*

The model has also been applied to natural aquifers for management and future planning purposes.

TABLE OF CONTENTS

www.scantopdf.eu

LIST OF FIGURES

INTRODUCTION

Careful management and planning of groundwater aquifers is essential to assure users of an adequate and sustained yield of water. This process helps to avoid the 'mining' of water and excessive drawdown of the water table and the dangers inherent therein. When the aquifer has a coastal boundary, an additional danger is the intrusion of the seawater into the aquifer. Excessive pumping could cause an increase in the salinity concentration of water, rendering it unfit for drinking and other purposes. Management of the aquifer includes knowing the location of the saltwater under present and future pumping conditions. A numerical model is one method by which the location of the saltwater can be estimated for a given set of hydrologic conditions.

Two types of models have been developed to study saltwater intrusion into aquifers. The first type of model is one in which a sharp interface separates the freshwater and saltwater (Bear and Dagan, 1964; Fetter, 1972; Glover, 1959; Kashef, 1975; Pinder and Page, 1976; Rumer and Harleman, 1963; Shamir and Dagan, 1971; Van der Veer, 1976). The second type assumes that the saltwater has gradually mixed (diffused) with the freshwater and salt concentration varies continuously from zero in the freshwater to the concentration that exists in the sea (Bear and Dagan, 1963; Desai and Contractor, 1977; Gelhar et al., 1972; Henry, 1964; Segol et al., 1975). In general, there is always some degree of saltwater diffusion in coastal aquifers. However, when the diffusion is confined to a narrow band in the aquifer, it may still be appropriate to use a sharp interface model. This report describes the development and use of a sharp interface model.

Several numerical methods have been used in sharp interface models, including the method of characteristics, finite difference and finite element methods. The concepts and techniques described by Sã da Costa and Wilson (1979) have some unique features and were used extensively in this study. The model (SWIM) described by Sã da Costa and Wilson (1979) is very general, versatile and accurate and would have been used in this study if the necessary computer hardware facilities were available in Guam. For this reason, a simpler model (SWIGS2D) was developed to utilize the locally-available facilities. The model utilizes linear triangular elements and the Galerkin weighted residual method for deriving the element equations. The model can simulate steady or unsteady conditions in confined or unconfined aquifers. Several types of boundary conditions can be imposed; freshwater and saltwater heads can be specified as a function of time at any number of nodes; freshwater and saltwater flows can be specified between any pair of boundary nodes and a mixed-type boundary condition can be applied along coastal boundaries. By specifying the saltwater head at all nodes to be very low in comparison with expected freshwater heads, the thickness of the salt-water everywhere can be held to an arbitrarily small value, thus simulating an aquifer with only freshwater flows. By specifying the saltwater head to be equal to zero above mean sea leavel at all nodes, the Ghyben-Herzberg condition is fulfilled and the location of the interface can be calculated. Under these conditions, there is no saltwater movement with hydrostatic pressures existing everywhere in the salt water.

Application of the model to a variety of flow situations for which analytical solutions are available shows the accuracy of the model. The model has been applied to several natural aquifers, with complex geometries, basement topography and boundary conditions. These applications will be described in future technical reports of this Institute. Availability of such a model enables the manager of an aquifer to explore the advantages and disadvantages of a number of options relating to development of the resource.

FINITE ELEMENT EQUATIONS

Two partial differential equations were used to describe the flow of freshwater and saltwater. These equations were derived by Sã da Costa and Wilson (1979) and Bear (1972) and obtained by integrating the three-dimensional equations in the vertical 'z' direction, resulting in depth-averaged quantities (e.g. K_x^f, ϕ^f, n)

$$\partial/\partial x(K_x^f b^f(\partial\phi^f/\partial x))+\partial/\partial y(K_y^f b^f(\partial\phi^f/\partial y))+ N + q_p^f + K_o(\phi^f-\phi_o) =$$

$$(n\gamma^f/\Delta\gamma)\partial\phi^f/\partial t - (n\gamma^s/\Delta\gamma)\partial\phi^s/\partial t \qquad (1)$$

$$\partial/\partial x(K_x^s b^s(\partial\phi^s/\partial x))+\partial/\partial y(K_y^s b^s(\partial\phi^s/\partial y)) = (n\gamma^s/\Delta\gamma)\partial\phi^s/\partial t -$$

$$(n\gamma^f/\Delta\gamma)\partial\phi^f/\partial t \qquad (2)$$

To assure continuity of pressures across the interface in the fresh and salt water, the following equation must be satisfied

$$\zeta = \gamma^s/\Delta\gamma\phi^s - \gamma^f/\Delta\gamma\phi^f \qquad (3)$$

Linear triangular elements have been used in the solution of the above equations. Figure 1 shows triangular elements with Cartesian coordinates (x,y) and triangular coordinates (L_1,L_2,L_3). The relationship between the Cartesian and triangular coordinates for any point in the element is given by

$$\begin{Bmatrix}1\\x\\y\end{Bmatrix} = \begin{bmatrix}1 & 1 & 1\\x_1 & x_2 & x_3\\y_1 & y_2 & y_3\end{bmatrix}\begin{Bmatrix}L_1\\L_2\\L_3\end{Bmatrix} \qquad (4)$$

It can be shown that:

$$L_1 = A_1/A,\ L_2 = A_2/A,\ \text{and}\ L_3 = A_3/A \qquad (5)$$

Also

$$\begin{Bmatrix}L_1\\L_2\\L_3\end{Bmatrix} = \frac{1}{2A}\begin{bmatrix}2A_{23} & b_1 & a_1\\2A_{31} & b_2 & a_2\\2A_{12} & b_3 & a_3\end{bmatrix}\begin{Bmatrix}1\\x\\y\end{Bmatrix} \qquad (6)$$

Where

$$\begin{array}{lll} a_1 = x_3 - x_2 & b_1 = y_2 - y_3 & 2A_{12} = x_1y_2 - x_2y_1 \\ a_2 = x_1 - x_3 & b_2 = y_3 - y_1 & 2A_{23} = x_2y_3 - x_3y_2 \\ a_3 = x_2 - x_1 & b_3 = y_1 - y_2 & 2A_{31} = x_3y_1 - x_1y_3 \end{array} \qquad (7)$$

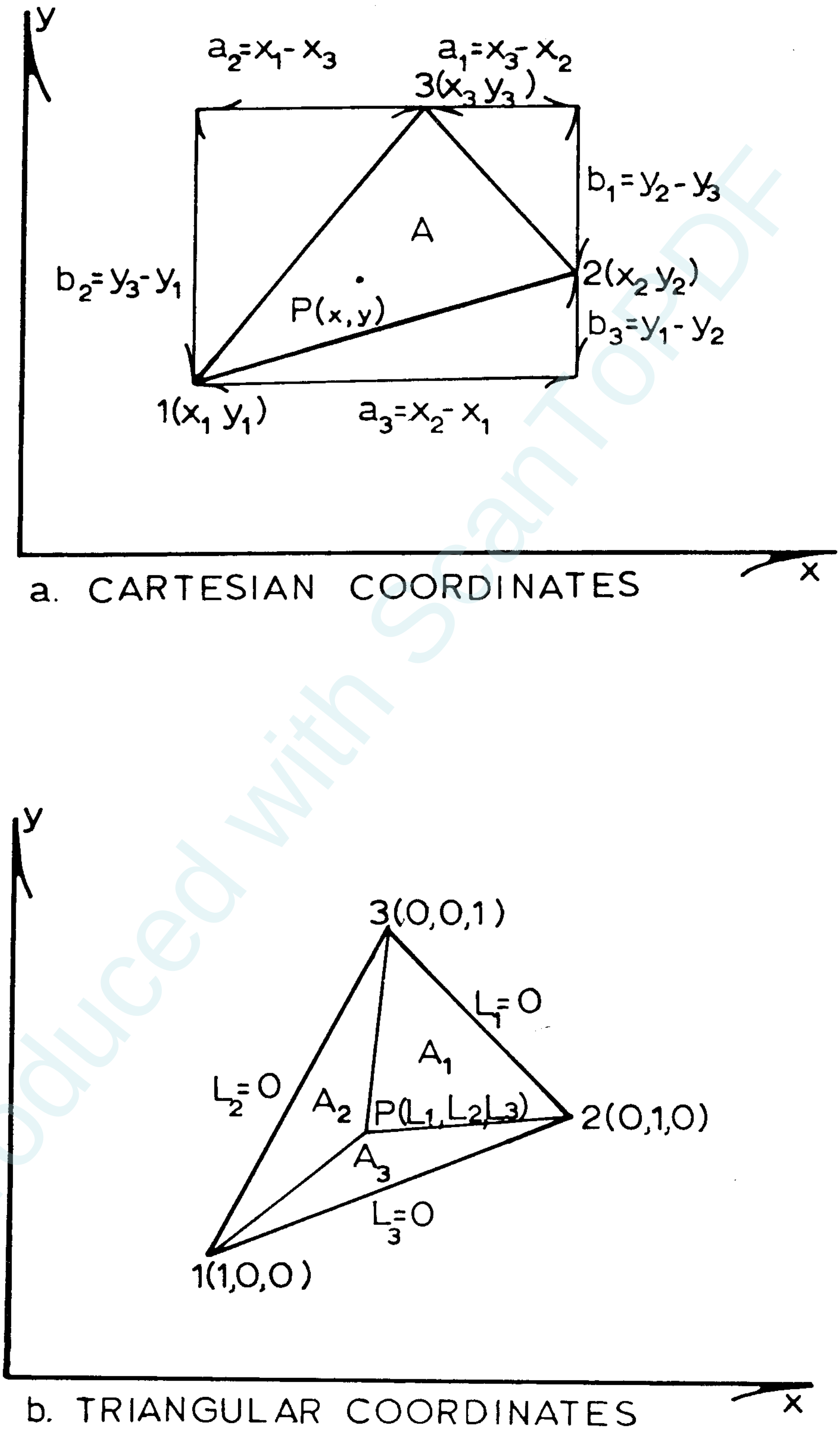

Figure 1. Cartesian and triangular co-ordinate systems.

and A = area of the element =

$$\tfrac{1}{2}(a_3b_2 - a_2b_3) = \tfrac{1}{2}(a_1b_3 - a_3b_1) = \tfrac{1}{2}(a_2b_1 - a_1b_2)$$

The dependent variables (ϕ^f, ϕ^s, b^f, b^s) are supposed to vary within an element in the following manner.

$$\phi^f = [L]\ \{\phi^f\}^e = [L_1\ L_2\ L_3] \begin{Bmatrix} \phi_1^f \\ \phi_2^f \\ \phi_3^f \end{Bmatrix} \tag{8}$$

Also, $\phi^s=[L]\{\phi^s\}^e$, $b^f=[L]\{b^f\}^e$, $b^s=[L]\{b^s\}^e$

Substituting the above equations in equations 1 and 2, we get

$$\partial/\partial x(K_x^f[L]\{b^f\}^e\partial/\partial x([L]\{\phi^f\}^e)) + \partial/\partial y(K_y^f[L]\{b^f\}^e\partial/\partial y([L]\{\phi^f\}^e)) +$$

$$N + q_p^f + K_o([L]\{\phi^f\}^e-\phi_o)-(n\gamma^f/\Delta\gamma)\partial\phi^f/\partial t+(n\gamma^s/\Delta\gamma)\partial\phi^s/\partial t = 0 \tag{9}$$

$$\partial/\partial x(K_x^s[L]\{b^s\}^e\partial/\partial x([L]\{\phi^s\}^e))+\partial/\partial y(K_y^s[L]\{b^s\}^e\partial/\partial y([L]\{\phi^s\}^e)) -$$

$$(n\gamma^s/\Delta\gamma)\partial\phi^s/\partial t + (n\gamma^f/\Delta\gamma)\partial\phi^f/\partial t = 0 \tag{10}$$

When the exact values of ϕ^f and ϕ^s at all the nodes are input into equations 9 and 10, the right-hand sides will be equal to zero. If, however, approximate values of ϕ^f and ϕ^s are used, the right-hand sides will be equal to residues R^f and R^s other than zero. The correct values of ϕ^f and ϕ^s are obtained by Galerkin's weighted residual method. In this method, the following equations must be satisfied

$$\int L_i R^f dA = 0 \qquad \text{for } i = 1,2,3 \tag{11}$$

and

$$\int L_i R^s dA = 0 \qquad \text{for } i = 1,2,3 \tag{12}$$

Equations 9 and 10 can be substituted into equations 11 and 12. Each term is integrated separately below.

first term: $\int L_i\partial/\partial x[K_x^f[L]\{b^f\}^e\partial/\partial x([L]\{\phi^f\}^e)]dA$

Use is made of the following identity in integrating the first term

$$\int L_i\ \partial/\partial x\ (B)\ dA = -\int (B)\ \partial L_i/\partial x\ dA + \int_s L_i B n_x ds \quad (13)$$

Thus the first term becomes

$$-\int (K_x^f [L]\{b^f\}^e \partial/\partial x([L]\{\phi^f\}^e))\partial L_i/\partial x\ dA + \int L_i (K_x^f [L]\{b^f\}^e \partial/\partial x([L]\{\phi^f\}^e))$$

$$n_x\ ds = -K_x^f b_i/2A[b/2A]\{\phi^f\}^e \int [L]\{b^f\}^e dA + \int_s L_i(-qn_x)ds = -K_x^f b_i/2A[b/2A]$$

$$\{\phi^f\}^e A/3\{b^f\}^e + \int_s L_i(-qn_x)ds = -K_x^f b_i/4A[b]\{\phi^f\}^e 1/3\{b^f\}^e + \int_s L_i(-qn_x)ds$$

Similarly the second term will become

$$-K_y^f a_i/4A[a]\{\phi^f\}^e + \int_s L_i(-qn_y)ds$$

Adding the first two terms, we get

$$-K_x^f b_i/4A[b]\{\phi^f\}^e 1/3\{b^f\}^e - K_y^f a_i/4A[a]\{\phi^f\}^e 1/3\{b^f\}^e + \int_s L_i(-qn_x - qn_y)ds$$

The last term has to be evaluated around the boundary of the element. For a flow boundary, this term becomes

$$\int_s L_i(-qn_x - qn_y)ds = \int L_i q ds = -q\int_0^{\ell_{ij}} (1-s/\ell_{ij})ds = -q\ell_{ij}/2 \quad (14)$$

For a coastal boundary, we get

$$\int L_i(-qn_x - qn_y)ds = -\int L_i q ds = -\int L_i K^c(\phi^f - \phi^s)ds = -\int (L_i)K^c \phi^f ds + \int (L_i)K^c \phi^s ds$$

$$= -\int L_i K^c [L]\{\phi^f\}^e ds + \int L_i K^c [L]\{\phi^s\}^e ds = -\int (1-s/\ell_{ij})K^c[(1-s/\ell_{ij})(s/\ell_{ij})]$$

$$\{\phi^f\}^e ds + \int (1-s/\ell_{ij})K^c[(1-s/\ell_{ij})(s/\ell_{ij})]\{\phi^s\}^e ds = -K^c \ell_{ij}[(1/3)\quad(1/6)]$$

$$\{\phi^f\}^e + K^c \ell_{ij}[(1/3)\quad(1/6)]\{\phi^s\}^e \quad (15)$$

The third term in equation 11 (after substituting equation 9) is

$\int L_i N dA = N\int L_i dA$ for constant N over element $= N(A/3)$

The fourth term is

$$\int L_i q_p^f \, dA = Lt_{\Delta A \to 0} \Sigma L_i q_p^f \, \Delta A = q_p^f \, \Delta A = Q_p/n_e$$

Where Q_p is the total pump flow and n_e is the number of elements at well node i, and $L_i = 0$ outside the well and equal to 1 within the well.

The fifth term is

$$\int L_i K_o([L]\{\phi^f\}^e - \phi_o)dA = \int K_o L_i [L]\{\phi^f\}^e dA - \int L_i K_o \phi_o \, dA = K_o A[(1/6) \quad (1/12) \quad (1/12)]\{\phi^f\}^e - K_o \phi_o A/3$$

The sixth term is

$$-\int L_i n\gamma^f/\Delta\gamma (d\phi^f/dt)dA = -n\gamma^f/\Delta\gamma \int L_i [L]\{d\phi^f/dt\}^e dA = (-n\gamma^f/\Delta\gamma)A[(1/6) \quad (1/12) \quad (1/12)]\{d\phi^f/dt\}^e$$

Similarly, the last term becomes

$$(n\gamma^s/\Delta\gamma)A[(1/6) \quad (1/12) \quad (1/12)]\{d\phi^s/dt\}^e$$

The three equations in equation 11 are given below in matrix form, without the boundary terms.

$$1/3\{b^f\}^e[K_{ij}^f]\{\phi^f\}^e + NA/3 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + Q_p/n_{Li} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + K_o(A/12) \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \{\phi^f\}^e - (K_o A\phi_o/3) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - n\gamma^f/\Delta\gamma(A/12) \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \{d\phi^e/dt\}^e + n\gamma^s/\Delta\gamma \; (A/12) \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \{d\phi^s/dt\}^e = 0 \qquad (16)$$

$$\text{where } K_{ij}^f = (-1/4A) \begin{bmatrix} K_x^f b_1 b_1 + K_x^s a_1 a_1 & K_x^f b_1 b_2 + K_x^s a_1 a_2 & K_x^f b_1 b_3 + K_x^s a_1 a_3 \\ K_x^f b_2 b_1 + K_x^s a_2 a_1 & K_x^f b_2 b_2 + K_x^s a_2 a_2 & K_x^f b_2 b_3 + K_x^s a_2 a_3 \\ K_x^f b_3 b_1 + K_y^s a_3 a_1 & K_x^f b_3 b_2 + K_y^s a_3 a_2 & K_x^f b_3 b_3 + K_y^s a_3 a_3 \end{bmatrix} \qquad (17)$$

Since ϕ^f and ϕ^s are functions of time, the following substitutions are made in equation 16.

$$\{\phi^f\}^e=(1-\Theta)\{\phi^f_t\}^e+\Theta\{\phi^f_{t+\Delta t}\}^e \text{ and } \{\phi^s\}^e=(1-\Theta)\{\phi^s_t\}^e+\Theta\{\phi^s_{t+\Delta t}\}^e \qquad (18)$$

$$\{b^f\}^e=\tfrac{1}{2}[\{b^f_t\}^e+\{b^f_{t+\Delta t}\}^e] \text{ and } \{b^s\}^e=\tfrac{1}{2}[\{b^s_t\}^e+\{b^s_{t+\Delta t}\}^e] \qquad (19)$$

$$\{d\phi^f/dt\}^e=1/\Delta t[\{\phi^f_{t+\Delta t}\}^e-\{\phi^f_t\}^e] \text{ and } \{d\phi^s/dt\}^e=1/\Delta t[\{\phi^s_{t+\Delta t}\}^e-\{\phi^s_t\}^e] \qquad (20)$$

The final element equation for freshwater flow is given below.

$$1/3\{\tfrac{1}{2}(b^f_t+b^f_{t+\Delta t})\}[K^f_{ij}]\Theta\{\phi^f_{t+\Delta t}\}^e-n\gamma^f/\Delta\gamma(A/\Delta t)(1/12)\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^f_{t+\Delta t}\}^e$$

$$+\ n\gamma^s/\Delta\gamma(A/\Delta t)(1/12)\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^s_{t+\Delta t}\}^e+K_oA\Theta/12\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^f_{t+\Delta t}\}^e=\ -1/3$$

$$\{\tfrac{1}{2}(b^f_t+b^f_{t+\Delta t})\}[K^f_{ij}](1-\Theta)\{\phi^f_t\}^e-NA/3\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}-Q_p/n_e\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}-n\gamma^f/\Delta\gamma\ (A/\Delta t)$$

$$(1/12)\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^f_t\}^e+n\gamma^s/\Delta\gamma(A/\Delta t)(1/12)\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^s_t\}^e+K_o\phi_oA/3$$

$$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}-(1-\Theta)K_oA/12\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^f_t\}^e \qquad (21)$$

Similarly, the final element equation for saltwater becomes

$$1/3\{\tfrac{1}{2}(b^s_t+b^s_{t+\Delta t})\}[K^s_{ij}]\Theta\{\phi^s_{t+\Delta t}\}^e-n\gamma^s/\Delta\gamma(A/\Delta t)(1/12)\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^s_{t+\Delta t}\}^e$$

$$+\ n\gamma^f/\Delta\gamma(A/\Delta t)(1/12)\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\{\phi^f_{t+\Delta t}\}^e=-1/3\{\tfrac{1}{2}(b^s_t+b^s_{t+\Delta t})\}[K^s_{ij}](1-\Theta)\{\phi^s_t\}^e$$

$$- n\gamma^{s}/\Delta\gamma(A/\Delta t)(1/12)\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}\{\phi_t^s\}^e + n\gamma^{f}/\Delta\gamma(A/\Delta t)(1/12)\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}\{\phi_t^f\}^e \quad (22)$$

where $[K_{ij}^s]$ is defined in a manner similar to the definition of $[K_{ij}^f]$ in equation 17.

DESCRIPTION OF COMPUTER PROGRAM

The computer program has the capacity to handle steady and unsteady flows, and analyze both confined and unconfined flows. If the aquifer is confined, the program can consider leaky and non-leaky conditions. Additionally, any number of pumps can be accommodated at the nodes of the network. Recharge is constant in an element but can be varied from element to element. Specified head or flow conditions can be applied at the boundaries. At a coastal boundary, a mixed or third-type boundary condition can be specified. For steady flow conditions, the saltwater head can be specified to be zero along the boundary or at every node in the network. This procedure assures that the Ghyben-Herzberg condition is satisfied. The program can also be run in the unsteady mode with the Ghyben-Herzberg condition. Constant head conditions are taken into account by eliminating that variable from the matrix. The right-hand side of each equation is reduced by the product of the constant head times its coefficient in the matrix. The row and column of that variable are eliminated and the size of the matrix reduced.

If the saltwater head at every node in the network is specified to be much less than the anticipated freshwater head, the results of the computer program will show that the thickness of the saltwater layer is equal to an arbitrarily small value (BTOE). Under these conditions, the program can simulate flow in a freshwater aquifer. Use of the program to simulate freshwater aquifers will be limited only by the number of nodes usable, since the element matrix still contains saltwater heads. Apart from this limitation, the solution procedure should still be efficient.

Since the heads are assumed to vary linearly across the triangular element, the velocity in each element will be constant. By specifying NVEL = 1, the program will print out the velocities in the x and y directions in each element in both the fresh water and salt water layers. These velocities can then be used to calculate the flow rates across any line or boundary. By specifying NTOE = 1, the program will determine where in the network a freshwater or saltwater toe occurs. A saltwater toe occurs where the interface intersects the lower impervious boundary. The output provides the element number, the node numbers, and the fractional distance between the two nodes where the saltwater toe occurs. The same kind of information is also provided about the freshwater toe. A freshwater toe occurs where the phreatic surface intersects the lower impervious boundary or where the interface intersects the upper impervious boundary in confined aquifers.

The program assumes that there are two independent variables at each node: the freshwater head and saltwater head. After solving for the heads, equation 3 is used to determine the depth (ζ) of the interface. The location of the interface determines the thickness of the freshwater and the saltwater layers. If, however, the interface is calculated to be below the lower impervious boundary, then the entire aquifer thickness should contain only freshwater and the thickness of the saltwater layer should theoretically be zero. However, the program makes the saltwater layer equal to BTOE. The permeability in the saltwater thickness BTOE is made much smaller than that specified in the region where the saltwater thickness is greater than BTOE. Similarly, when the phreatic surface intersects the lower impervious boundary, the freshwater layer beyond the toe is made equal to BTOE instead of zero. In the course of the program, as the interface and the phreatic surface move with respect to time, the thickness and permeability of the saltwater and freshwater toes are altered accordingly.

Equation 18 introduces a weighting factor, θ, which varies between zero and one. This factor is useful in regulating the stability and accuracy of the solution. When $\theta = 0$, the problem formulation is referred to as explicit. In this formulation, the spatial derivatives are evaluated at the known time-step, t. The time-step, Δt, necessary for stable results is very small. This results in very long execution times for the program. When $\theta = 0.5$, the problem formulation is referred to as the Crank-Nicolson approximation. This approach provides high-accuracy with large values of Δt, even though the results may show some numerical instability. When $\theta = 1.0$, the formulation is known as fully implicit. This formulation provides the maximum stability at a sacrifice of some accuracy. Values of θ between 0.5 and 1.0 (e.g., 0.6, 2/3, 3/4) have been used to provide the proper balance between accuracy and stability. All of the examples presented in this report were run with $\theta = 1$. When steady state results are desired, the program should be run with $\theta = 1.0$ and Δt equal to a very large number (e.g. 1.0E20). Since Δt is large, the time derivative terms in the element equations 21 and 22 become very small. By making $\theta = 1.0$, initial values of the thickness of the freshwater and saltwater layers are required because of the non-linear nature of the problem. The program should be run for several time steps until the error is less than the specified tolerance in one or two iterations.

The program can be run in any set of consistent units. Thus, if feet and seconds are the length and time units, the permeability and recharge must be input in ft/sec and the pump rate in cu.ft/sec. If meters and days are the length and time units, then the permeability and recharge must be input in m/day and the pump rate in cu.m/day.

When subdividing an aquifer into triangular elements, only one side of a triangle may form a boundary. The program cannot handle triangles with boundary conditions on two of its sides. The versatility of using triangles of different sizes and orientation should be taken advantage of. Place a node wherever a pump exists or is projected to be in the future. It is, however, advisable not to let the ratio of the largest triangle to the smallest triangle become too large. It will generally be the size of the smallest triangle that determines the time step Δt that can be used for stable results.

APPLICATIONS OF THE PROGRAM

The computer program described in the previous section is very general and can be applied to a wide variety of complex aquifers. Before this program can be used with confidence, it is advisable to check the results of the program against results that are analytically known. Even this does not guarantee that the program will behave properly when applied to complex aquifers. However, if it does not compare favorably with analytic results, it most certainly would not give valid results with complex aquifers. This program was checked out against three different groundwater situations. The first situation deals with the steady state, one dimensional saltwater intrusion into confined and unconfined aquifers. The second case is the unsteady drawdown of head due to pumping of a well (Theis solution) in a confined aquifer. Finally, the program is checked against the analytic solution of the one-dimensional gravitational segregation problem.

Figure 2 shows the 2-D network of elements used to simulate 1-D flow towards the sea coast. The recharge within the three impervious boundaries flows towards the sea in an unconfined aquifer. Van der Veer (1976) gives the analytic solution for this flow situation. Figure 3 shows a comparison between this analytic solution and the 2-D model results. Better results could have been obtained had a finer element mesh been used. The velocity in almost all of the elements had an angle of zero degrees to the x axis. The elements close to the sea coast did show velocity directions up to 10 degrees away from the x axis. The numerical results were symmetrical about a line joining node 3 and node 30.

The network shown in Figure 2 was also used to simulate 1-D flow towards the sea coast in a confined aquifer. The analytic solution to this flow situation is given by Glover (1959). A constant freshwater flow enters the aquifer between node 1 and node 2. A saltwater toe occurs in this case, with only freshwater occurring to the left of the toe and both fresh and saltwater occurring to the right. The program output provides the element numbers in which the toe occurs and provides data to plot the location of the toe in those elements. Both of these flow cases were simulated in the steady-state mode of the program, that is, with $\theta = 1.0$ and Δt equal to a very large number. Figure 4 presents a comparison between the analytic solution and the results of the 2-D model. Once again, the velocity direction in nearly all elements is in the x direction. However, the directions in the elements close to the sea coast were up to 10 degrees away from the x axis.

The next flow simulation dealt with the unsteady drawdown of the piezometric head due to pumping from a well in a confined aquifer. Because of radial symmetry, the finite element network (Figure 5) simulated only a 15 degree pie-shaped wedge. The pump is located at the tip of the wedge (node 1), and the pumping rate is kept constant. The analytic solution to the drawdown as a function of radius and time is presented by Theis (1935). The boundary conditions at the outer nodes were obtained from the Theis solution. In order to keep the thickness of the saltwater layer to an arbitrarily small value (BTOE), the saltwater heads at all nodes was specified to be much smaller than the anticipated freshwater heads. The program was run in the unsteady mode with $\theta = 1.0$ and $\Delta t = 0.1$ days for the 0.5 days and again with $\Delta t = 0.5$ days for four days. Figure 6 shows a comparison of the 2-D numerical results with the Theis solution at node 3. Because of the

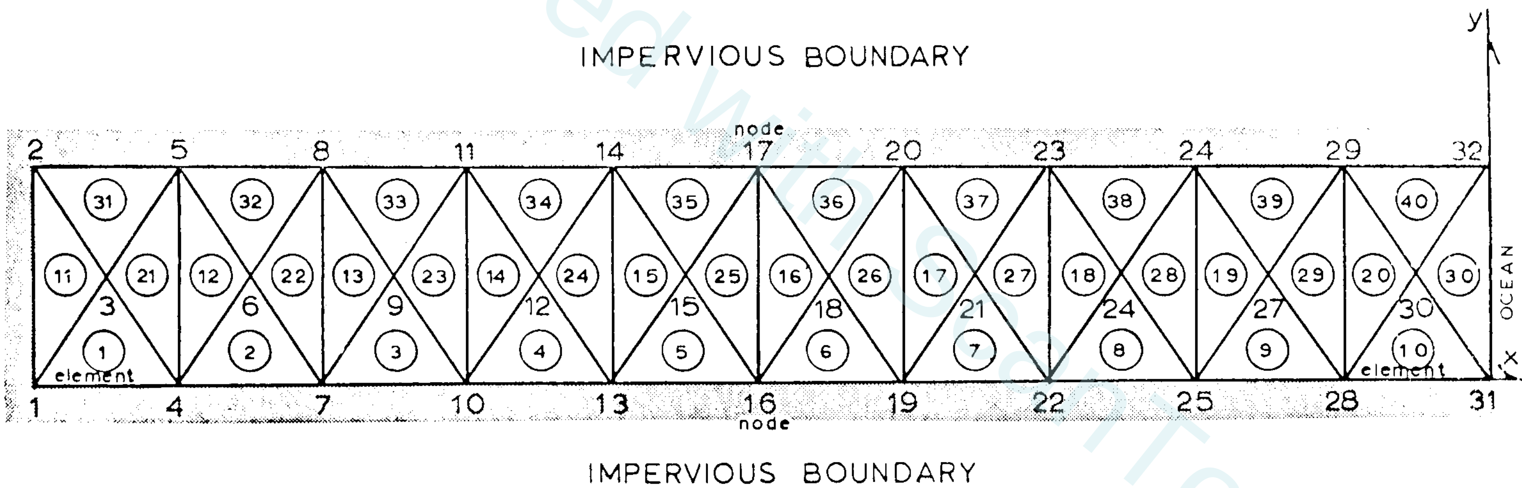

Figure 2. Element network used to simulate 1-D salt water intrusion.

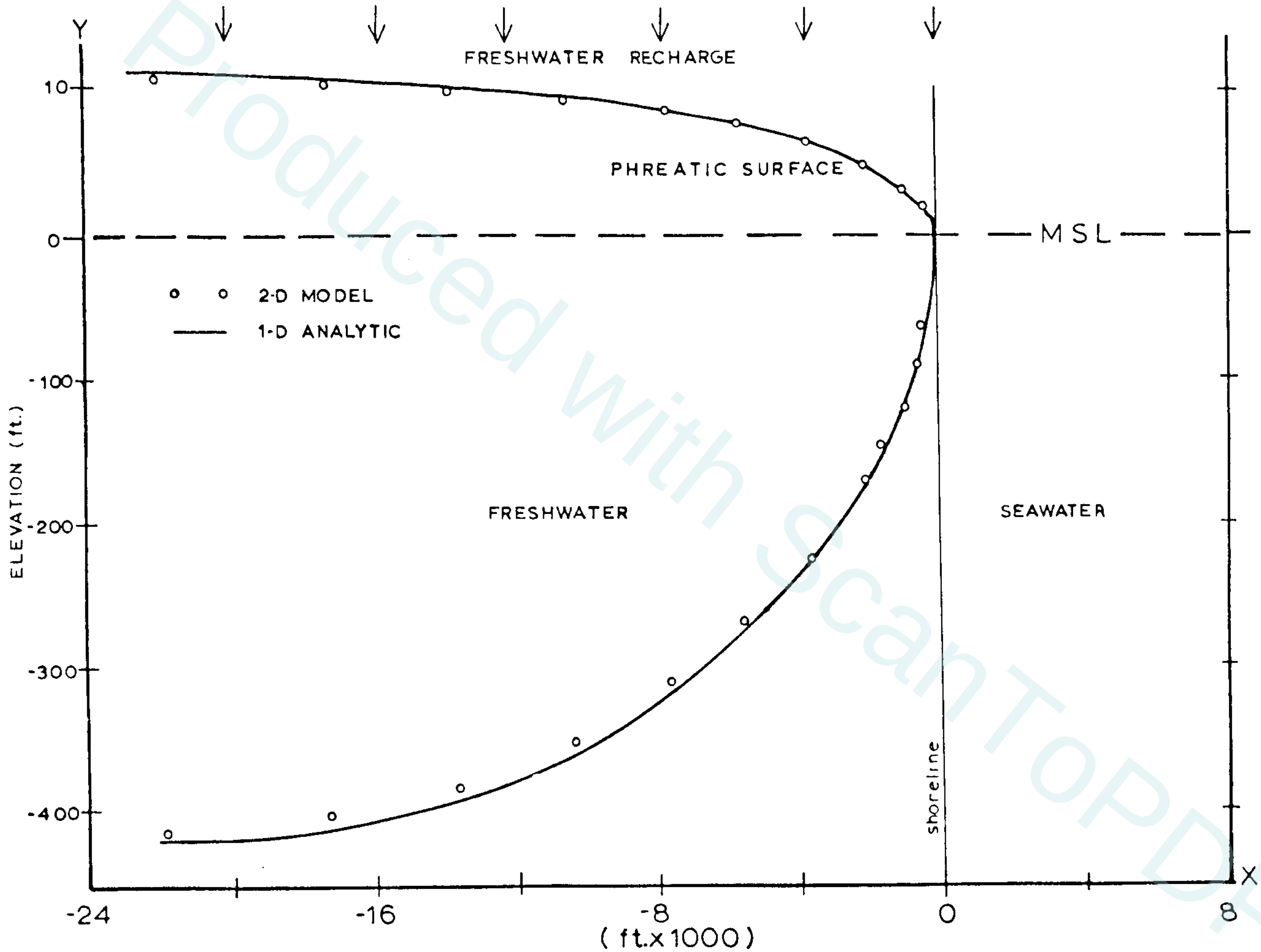

Figure 3. Comparison of 1-D analytic and 2-D model results for an unconfined aquifer.

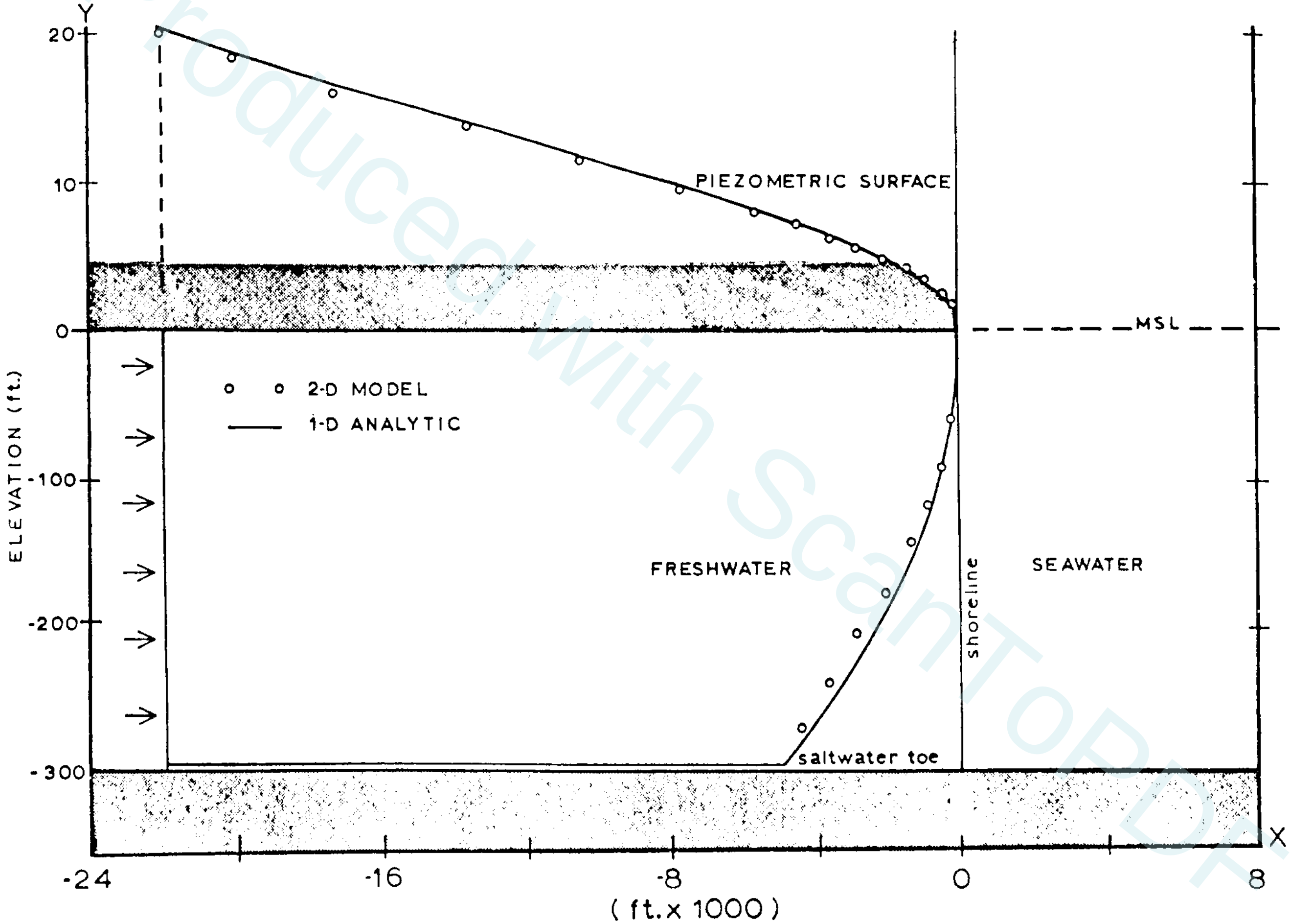

Figure 4. Comparison of 1-D analytic and 2-D model results for a confined aquifer.

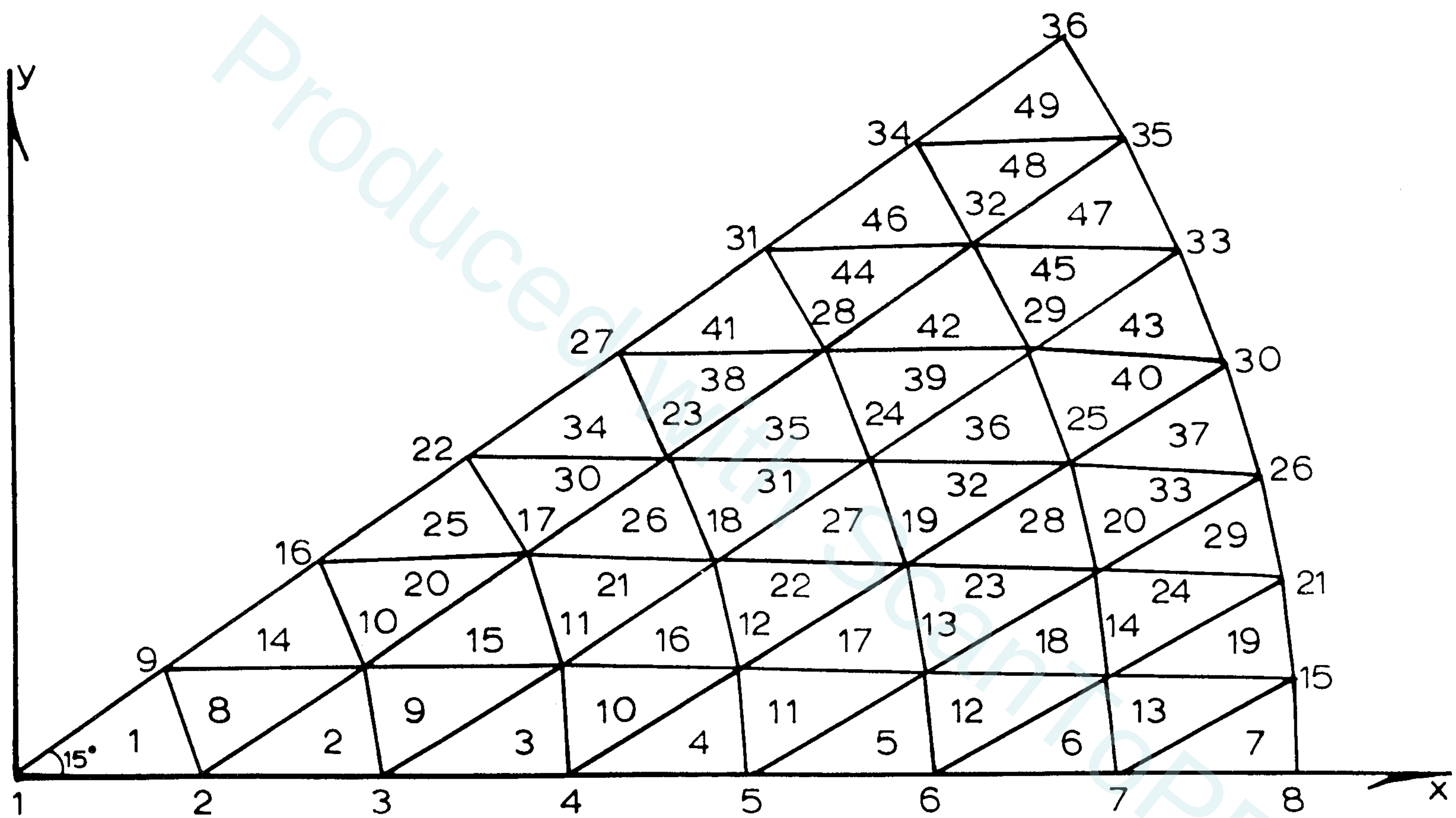

Figure 5. Element network for radial flow to a pump.

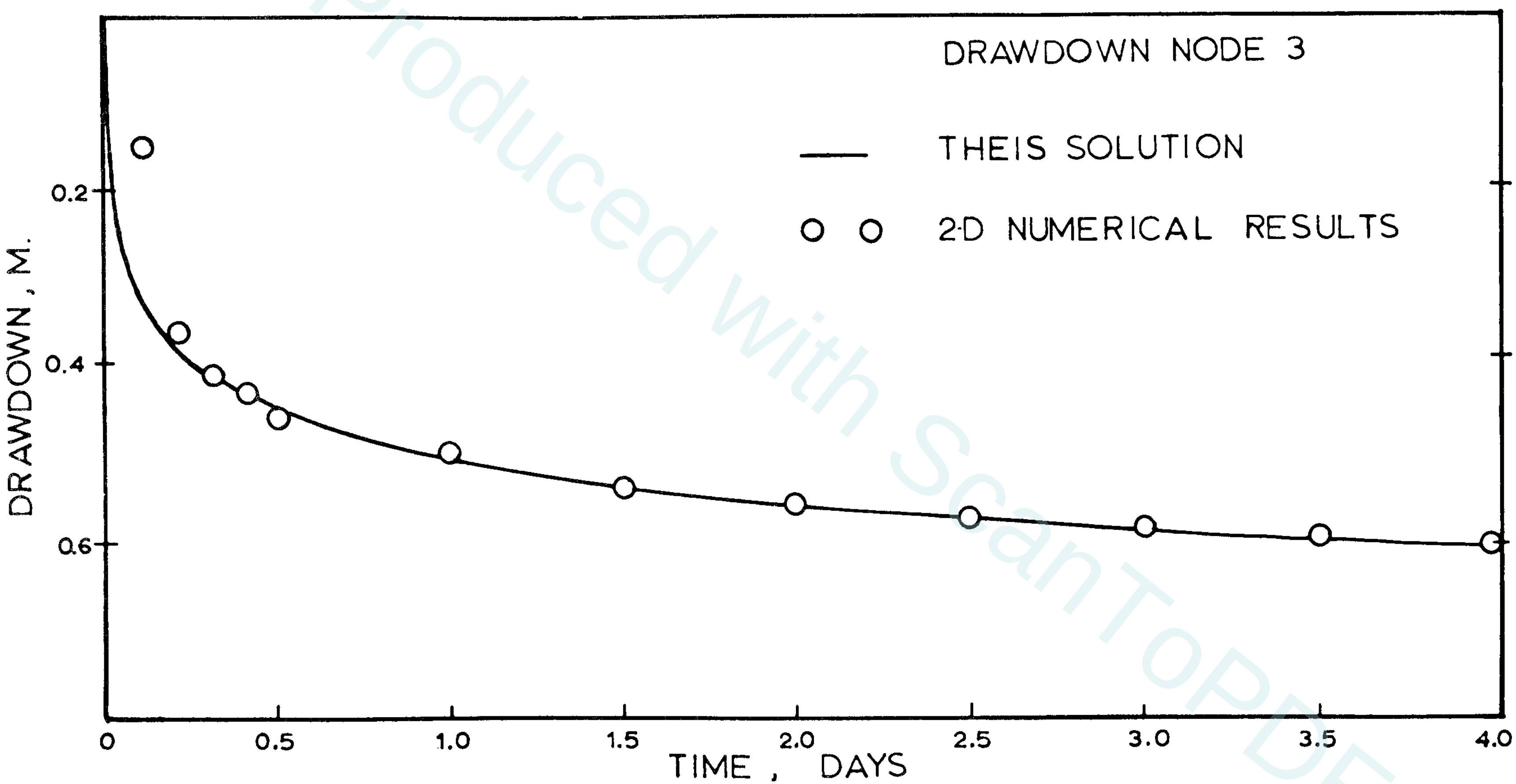

Figure 6. Comparison of 2-D numerical results with the Theis solution as a function of time.

gradients in the initial stages, the numerical solution does not match the analytic solution. However, the results after t = 0.3 days do follow the theoretical solution. Figure 7 shows a comparison of the 2-D numerical results with the Theis solution as a function of radius at t = 2 days. The agreement is good at all radii except at the pump itself. This is because the Theis solution calculates the water level in the well from the pump diameter, whereas the numerical program does not use the pump diameter, in its internal calculations. The results of the numerical program close to the well could, of course, be improved by refining the element mesh in that neighborhood.

The final flow simulation deals with the gravitational segregation problem. Figure 8 shows a schematic of this problem. At time t = 0, there exists a vertical interface separating freshwater on the left and saltwater on the right. With passage of time, this interface rotates in a clockwise direction about the midpoint. The analytical solution to this problem (Gelhar et al., 1972) shows that the interface remains a straight line at any given time. This flow case was simulated by the model using constant head boundaries at both ends. The following parameters for the problem were used:

$$K_x^f = 39.024 \text{ m/day},\ K_x^s = 40.\text{m/day},\ n= 0.3,\ b = 10\text{m},\ \Delta t = 1 \text{ day}$$

Simulation was begun with the interface at $t = t_o = 7.87$ days. The initial conditions are given below:

x(m)	θ^f(m)	θ^s(m)	ζ (m)
-16 and less	5.125	4.7561	-10.
-12	5.12305	4.7847	- 8.75
- 8	5.11718	4.8094	- 7.5
- 4	5.10742	4.8304	- 6.25
0	5.09375	4.8609	- 5.0
4	5.07617	4.8609	- 3.75
8	5.05469	4.8609	- 2.50
12	5.02930	4.8761	- 1.25
16 and over	5.0	4.878	0.0

Figure 9 shows the location of the interface 10 days and 25 days after the initial conditions. The model follows the analytic solution very well.

The three different flow simulations described in this section indicate that the model behaves properly for these simple cases. The model has many capabilities that have not been tested for accuracy (e.g., flow through anisotropic media). Indeed, a model such as this may never be tested in all its features. However, based on the accuracy of the results till now, one can develop a level of confidence that new applications will be simulated properly. The purpose of developing this model was to study saltwater intrusion in the northern Guam lens. This application has been carried out successfully and a description of the results are provided in the succeeding Technical Report No. 27.

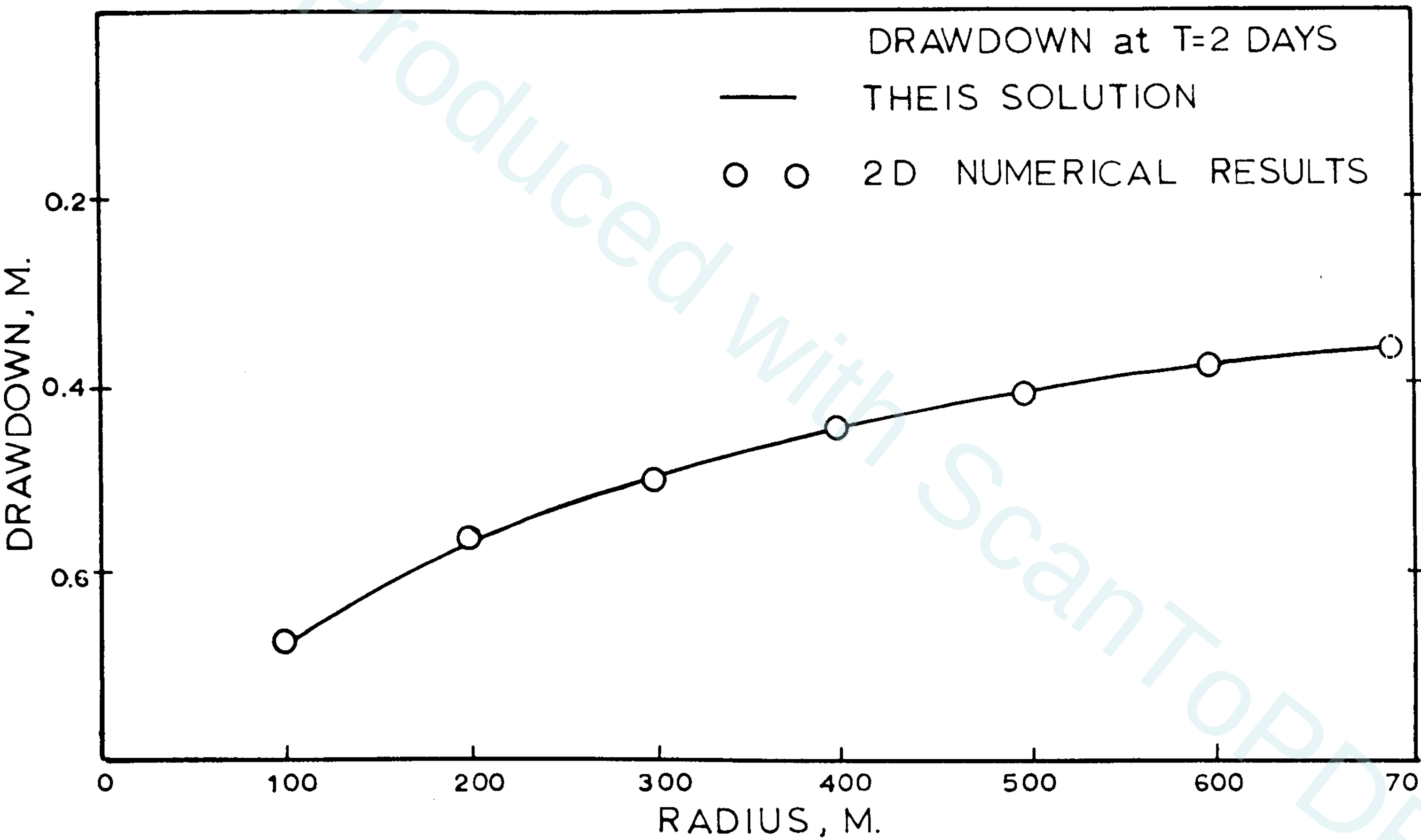

Figure 7. Comparison of 2-D numerical results with the Theis solution as a function of radius.

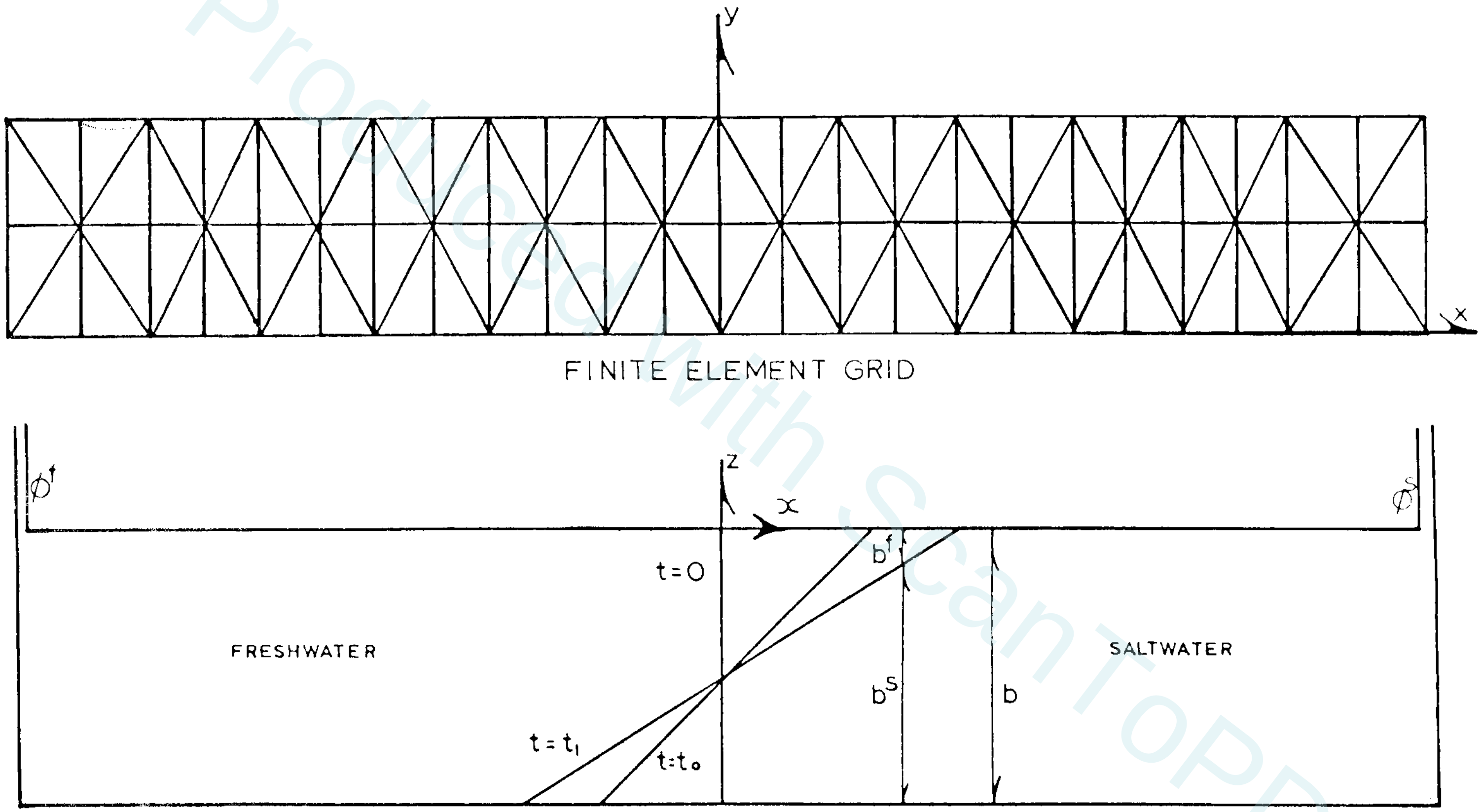

Figure 8. Schematic of the gravitational segregation problem.

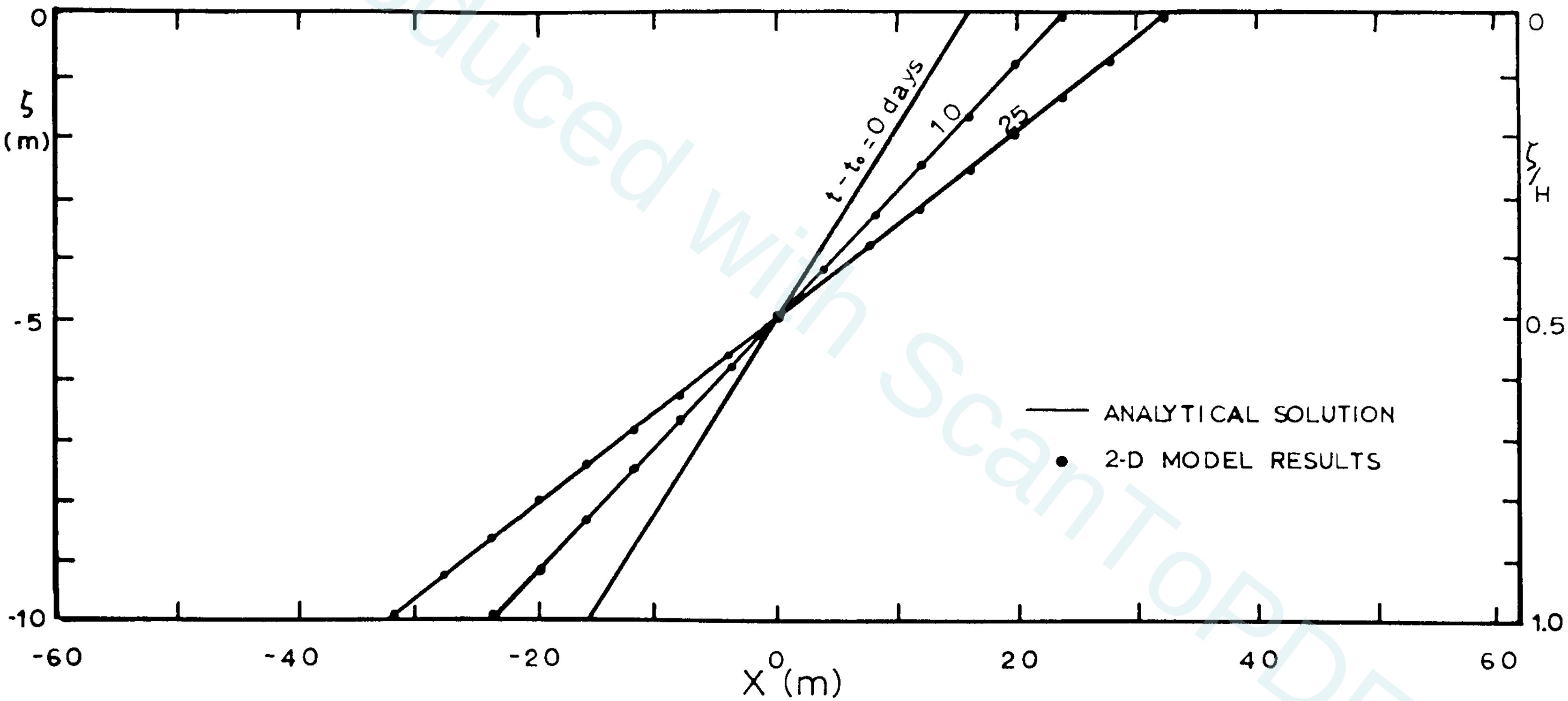

Figure 9. Comparison of 2-D model results and analytic solution of the gravitational segregation problem.

CONCLUSIONS

A computer program has been developed to simulate saltwater intrusion in groundwater systems. The main features of this program are:

1) Model uses linear triangular elements.

2) Confined or unconfined aquifers can be simulated.

3) Steady state or unsteady conditions can be handled.

4) Isotropic or anisotropic media can be studied.

5) Three types of boundary conditions can be specified.

   a) Head as a function of time.
   b) Flow as a function of time.
   c) A mixed boundary condition for freshwater flowing into a coastal boundary.

6) Freshwater wells can be located at any node.

7) Freshwater and saltwater toes can be tracked with time.

8) Direction and magnitude of the velocity in each element can be output.

9) Aquifers that contain only freshwater can also be simulated by specifying the saltwater head at all nodes to be much less than expected freshwater heads.

The program has been used to simulate three simple flow situations and the results compared with analytic solutions. In all three cases, the comparisons were good and promoted a degree of confidence in the program. The program has also been applied to large, complex, natural aquifers.

LITERATURE CITED

Bear, J. 1972. Dynamics of fluids in porous media. American Elsevier Publishing Company, New York.

Bear, J., and G. Dagan. 1963. The transition zone between fresh and saltwaters in coastal aquifers. Joint Experimental Coastal Ground Water Collectors Project, (Israel), Tech. Rept. 4.

Bear, J., and G. Dagan. 1964. Moving interface in coastal aquifers. ASCE, Jour. Hydraulics Division. 90(HY4):193-216.

Desai, C. S., and D. N. Contractor. 1977. Finite element analysis of flow, diffusion and salt water intrusion in porous media. In Bathe, K. J. (ed). Formulation and computational algorithms in finite element analysis. M. I. T. Press. p. 958-983.

Fetter, C. W., Jr. 1972. Position of the saline water interface beneath oceanic islands. Water Resources Research. 8(5):1307-1315.

Gelhar, L. W., J. L. Wilson, and J. S. Miller. 1972. Gravitational and dispersive mixing in aquifers. ASCE, Jour. Hydraulics Division. (HY12): 2135-2153.

Glover, R. E. 1959. The pattern of freshwater flow in a coastal aquifer. Jour. Geophysical Research. 64(4):457-459.

Henry, J. R. 1964. Effects of dispersion on salt encroachment in coastal aquifers. In Cooper, H. H. (ed). Sea water in coastal aquifers. Geol. Survey Water Supply Paper 1613-C:70-83.

Kashef, A. I. 1975. Management of retardation and salt water intrusion in coastal aquifers. Office of Water Research and Technology, Washington, D.C.

Pinder, G. F., and R. H. Page. 1976. Finite element simulation of salt water intrusion on the south fork of Long Island. International Conference in Finite Elements in Water Resources, Princeton.

Rumer, R. R., and D. R. F. Harleman. 1963. Intruded salt water wedge in porous media. ASCE, Jour. Hydraulics Division. 89(HY6):193-220.

Sã da Costa, A. A. G., and J. L. Wilson. 1979. A numerical model of seawater intrusion in aquifers. R. M. Parsons Laboratory for Water Resources and Hydrodynamics, Dept. Civil Engineering, MIT. Rept. No. 247.

Segol, G., G. F. Pinder, and W. G. Gray. 1975. A Galerkin-finite element technique for calculating the transient position of the saltwater front. Water Resources Research. 11(2):343-347.

Shamir, U., and G. Dagan. 1971. Motion of the seawater interface in coastal aquifers: a numerical solution. Water Resources Research. 7(3):644-657.

Theis, C. V. 1935. The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using groundwater storage. Trans. Amer. Geophysical Union. 2:519-524.

Van der Veer, P. 1976. Analytical solution for steady interface flow in a coastal aquifer involving a phreatic surface with precipitation. Jour. Hydrology. 29(3,4):315-339.

APPENDIX

List of Symbols

Symbol	Description	Dimensions
a_1	$x_3 - x_2$	L
a_2	$x_1 - x_3$	L
a_3	$x_2 - x_1$	L
A_{ij}	$(x_i y_j - x_j y_i)/2$	L^2
A	Area of element = $\frac{1}{2}(a_3 b_2 - a_2 b_3)$	L^2
A_1, A_2, A_3	Areas used to define Triangular Coordinates	L^2
b	Thickness of saturated layer	L
b_1	$y_2 - y_3$	L
b_2	$y_3 - y_1$	L
b_3	$y_1 - y_2$	L
B	A function of x and y	L
K^c	Permeability of the coastal aquifer	L/T
K_x	Permeability in the x direction	L/T
K_y	Permeability in the y direction	L/T
K_o	Permeability of aquitard overlying aquifer	L/T
K_{ij}	Matrix defined by equation 17	L
l_{ij}	Distance from node i to node j	L
L_1, L_2, L_3	Approximation Functions	L/T
n	Porosity	
n_x, n_y	Direction cosines of the outward normal	
q	Flow rate/unit length of boundary	L^2/T
q_p	Pump discharge	L/T
Q_p	Pump discharge	L^3/T
R^f, R^s	Residues of freshwater and saltwater equations	L/T
s	Distance along boundary	L
t	Time	T
x	Cartesian coordinate	L
y	Cartesian coordinate	L
γ	Specific weight of fluid	M/L^2T^2
$\Delta\gamma$	Difference in specific weight of salt and fresh water	M/L^2T^2

Appendix A - Continued.

Symbol	Description	Dimensions
Δt	Time increment	T
ϕ	Piezometric head	L
ϕ_o	Constant piezometric head of aquitard	L
ζ	Depth of interface below datum	L
	Superscripts	
e	Referring to element.	
f	Freshwater quantity.	
s	Saltwater quantity.	

APPENDIX B

INPUT DATA AND FORMAT

List of Input Variables.

TITLE (I)	Character variable for title of run.
NN	Number of nodes
NE	Number of elements
ITERMX	Maximum number of iterations for convergence at each time interval.
NPUMPS	Number of pumps in system.
NDATA	=0, Input data will not be printed out. =1, Input data will be printed out.
NOUT	=1, Output will be printed out at the end of every time counter J. =N, " " " " " " " " " " Nth time counter J.
JMAX	Maximum value of time counter J for execution of program.
JTRC	=0, for detailed matrix output not to be printed out. =J, for detailed matrix output to be printed out beginning at time counter J = JTRC
NVEL	=0, for no print out of magnitude and direction of velocities in elements. =1, for print out of magnitude and direction of velocities in elements.
NTOE	=0, for no print out of location of saltwater and freshwater toes. =1, for print out " " " " " " "
DT	Time interval for incrementing time counter J.
THETA	Time weighting factor =0.5, for Crank-Nicolson approximation. " " " =1.0, for fully implicit calculation.
TOL	Tolerance (as a fraction) used in convergence of solution.
GAMAF	Specific weight of lighter fluid (fresh water).
GAMAS	" " " denser " (saltwater).
TSTART	Time at start of program.
BTOE	Thickness of freshwater and saltwater toes.
LMNTQF	Number of boundary elements along which fresh water flow is specified.
LMNTQS	" " " " " " salt water " " "
LMNTCF	" " " " " " coastal boundary condition is specified.
NNODHF	Number of nodes at which fresh water head is specified.
NNODHS	" " " " " salt " " " " .
XC(I)	X-coordinate of node.
YC(I)	Y- " " " .
ZB(I)	Elevation of lower boundary (basement) of aquifer.
ZU(I)	" " upper " of confined aquifer. =1000.1 for phreatic aquifers.
NODE(I,1) NODE(I,2) NODE (I,3)	Node numbers of element I, specified in counter-clockwise direction.
KFFX(I)	Freshwater Permeability in the X direction in element I.
KFFY(I)	" " " " Y " " " ".
KSSX(I)	Saltwater " " " X " " " ".
KSSY(I)	" " " " Y " " " ".

Appendix B -- Continued.

PRSTY(I)	Porosity in element I.
NCONF(I)	=0 for phreatic (or unconfined) conditions in element I. =1 for confined conditions in element I.
NLMN	Number of succeeding elements with node numbers incremented by INC and having the same properties (permeability, porosity and NCONF).
INC	Increment
RECHG(I,J)	Recharge into element I, at time counter J.
NODEQF(I,1) } NODEQF(I,2) }	Nodes between which freshwater flow is specified. Nodes must be specified in anti-clockwise direction inside element.
QFWBND(I,J)	Freshwater flow between NODEQF(I,1) and NODEQF(I,2) entering aquifer between time counter (J-1) and (J).
NODEQS(I,1) } NODEQS(I,2) }	Nodes between which saltwater flow is specified. Nodes must be specified in anti-clockwise direction.
QSWBND(I,J)	Saltwater flow between NODEQS(I,1 AND NODEQS(I,2) entering aquifer between time counter (J-1) and (J).
NODECF(I,1) } NODECF(I,2) }	Nodes between which coastal boundary condition is to be applied. Nodes must be specified in anti-clockwise direction.
KC(I)	Permeability of coastal aquifer adjacent to boundary element.
NODEHF(I)	Node number at which fresh water head is to be specified.
SPECHF(I,J)	Specified freshwater head as a function of time counter J.
NODEHS(I)	Node number at which salt water head is to be specified.
SPECHS(I,J)	Specified saltwater head as a function of time counter J.
NODEP(I)	Node number at which pump is to be located.
NELEM(I)	Number of elements around pump node.
QPUMP(I,J)	Pump discharge occurring between time counter (J-1) and (J).
NLEAKY(I)	=0, when there is no leakage from aquitard above element I in aquifer. =1, when leakage can occur from aquitard above element I in aquifer.
KO(I)	Permeability of aquitard above element I in aquifer.
PHIO(I)	Constant head in aquitard above element I in aquifer.
HF(I,1) } Initial Conditions	Freshwater head at node I and time counter J=1.
HS(I,1) } Initial Conditions	Saltwater head at node I and time counter J=1.
BF(I,1) } Initial Conditions	Thickness of freshwater layer at node I and time counter J=1.
BS(I,1) } Initial Conditions	Thickness of saltwater layer at node I and time counter J=1.
LMNTYP(I)	=0, when all three nodes of element I have freshwater and saltwater thickness greater than BTOE. =1, when all three nodes of element I have freshwater thickness equal to BTOE, but saltwater thickness greater than BTOE. =2, when all three nodes of element I have saltwater thickness equal to BTOE, but freshwater thickness greater than BTOE. =3, when all three nodes of element I have freshwater and saltwater thicknesses equal to BTOE.
IO(I)	=0, when output of program is not to be printed out for node I. =1, when " " " " to be " " " " ".
I1	Counter.

Appendix B -- Continued.

List and Format of Input

No. of cards	List of Variables	Format
1	TITLE(I), I=1,20	20A4
1	NN,NE,ITERMX, NPUMPS,NDATA,NOUT,JMAX,JTRC,NVEL,NTOE	16I5
1	DT,THETA,TOL,GAMAF,GAMAS,TSTART,BTOE	E10·3,7F10·4
1	LMNTQF,LMNTQS,LMNTCF,NNODHF,NNODHS	16I5
NN	(XC(I),YC(I),ZB(I),ZU(I),I=1,NN)	10X,4F10·4
N, where $\sum_{i=1}^{N}(1+NLMN_i)=NE$	(NODE(I,1),NODE(I,2),NODE(I,3),KFFX(I),KFFY(I), KSSX(I),KSSY(I),PRSTY(I),NCONF(I),NLEAKY(I), NLMN,INC)	5X,3I5,5F10·2, I1,I2,2I3
N(1+JMAX/8) where $\sum_{i=1}^{N}(1+INC_i)=NE$	I,INC RECHG(J),J=1,JMAX } For all elements, for which NCONF(I)=0.	16I5 8F10·4
	IF LMNTQF>0	
LMNTQF(1+JMAX/8)	I1,NODEQF(I,2),NODEQF(I,2) (QFWBND(I,J),J=1,JMAX)	3I5 8F10·4
	IF LMNTQS>0.	
LMNTQS(1+JMAX/8)	I1,NODEQS(I,1),NODEQS(I,2) (QSWBND(I,J),J=1,JMAX)	3I5 8F10·4
	IF LMNTCF>0.	
LMNTCF	I1,NODECF(I,1),NODECF(I,2),KC(I)	3I5,5X,F15·13
	IF NNODHF>0.	
NNODHF(1+JMAX/8)	I1,NODEHF(I) (SPECHF(I,J),J=1,JMAX)	2I5 8F10·4
	IF NNODHS>0.	
NNODHS(1+JMAX/8)	I1,NODEHS(I) (SPECHS(I,J),J=1,JMAX	2I5 8F10·4
	IF NPUMPS>0.	
NPUMPS(1+JMAX/8)	NODEP(I),NELEM(I) (QPUMP(I,J),J=1,JMAX)	16I5
	FOR ALL ELEMENTS FOR WHICH NCONF(I)=1 AND NLEAKY(I)=1	
N, where $\sum_{i=1}^{N}(1+NLMN_i)=NE$	I,KO(I),PHIO(I),NLMN	I5,5X,2F10·4,I5
N, where $\sum_{i=1}^{N}(1+NND_i)=NN$	I,HF(I,1),NND	I5,5X,F10·4,I5
N, where $\sum_{i=1}^{N}(1+NND_i)=NN$	I,HS(I,1),NND	I5,5X,F10·4,I5

Appendix B -- Continued.

List and Format of Input

No. of cards	List of Variables	Format
N, where $\sum_{i=1}^{N} (1+NND_i)=NN$	I,BF(I,1),NND	I5,5X,F10.4,I5
N, where $\sum_{i=1}^{N} (1+NND_i)=NN$	I,BS(I,1),NND	I5,5X,F10.4,I5
NE/80	LMNTYP(I),I=1,NE	80I1
NN/80	IO(I),I=1,NN	80I1

APPENDIX C

SOURCE LISTING OF SWIGS2D.

```
C
C
C
C     ****************************************************************************
C     *  A TWO-DIMENSIONAL MODEL OF SALT WATER INTRUSION INTO GROUNDWATER SYSTEMS.*
C     *  A SHARP INTERFACE IS ASSUMED TO EXIST BETWEEN THE FRESH AND SALT WATER.  *
C     *  DEPTH-AVERAGED EQUATIONS FOR FRESH AND SALT WATER ARE SOLVED.            *
C     *  LINEAR, TRIANGULAR ELEMENTS ARE USED TO DISCRETIZE THE AQUIFER           *
C     *  THE GALERKIN METHOD IS USED TO DEVELOP THE ELEMENT EQUATIONS.            *
C     *  THIS MODEL WAS DEVELOPED FOR THE NORTHERN GUAM LENS STUDY                *
C     *  BY DINSHAW N. CONTRACTOR, WATER AND ENERGY RESEARCH INSTITUTE OF THE     *
C     *  WESTERN PACIFIC, UNI. OF GUAM, COLLEGE STATION, MANGILAO, GUAM,96913.    *
C     *  MODEL DESCRIPTION PROVIDED IN TECH. REPT. 26 , WERI.,UOG,GUAM.           *
C     *  MODEL CAN SIMULATE CONFINED OR UNCONFINED AQUIFERS AND STEADY OR         *
C     *  UNSTEADY CONDITIONS, WITH A VARIETY OF BOUNDARY CONDITIONS.              *
C     ****************************************************************************
C
      DIMENSION TITLE(20),XC(192),YC(192),ZB(192),PRSTY(300),NODE(300,3)
      DIMENSION ZU(192),HF(192,2),HS(192,2),BF(192,2),BS(192,2)
      DIMENSION QFWBND(044,48),QSWBND(04,004),NODEQF(44,2),NODEQS(040,2)
      DIMENSION SPECHF(40,048),NODECF(40,2),NODEHF(40),NODEHS(040)
      DIMENSION SPECHS(040,048),QPUMP(72,048),NODEP(72),NELEM(72)
      DIMENSION PHIU(004),A(300,3),B(300,3),ELQF(044)
      DIMENSION ELQS(04),ELCF(40),R(380),AA(380,380),RECHG(300,048)
      DIMENSION R1(4),A1(4,3),A2(4,3),AAA(25000),IO(192)
      DIMENSION ZETA(192),AREA(300),LMNTYP(300),NCONF(300),NLEAKY(300)
      COMMON R,AAA
      DIMENSION N7(120),I2(380),HFNEW(192),HSNEW(192)

      REAL KIJF(300,3,3),KIJS(300,3,3),KFX(300),KFY(300),KSX(300)
      REAL KO(004),KFIJ(300,3,3),KSIJ(300,3,3),KSY(300),KFFX(300)
      REAL KOIJ(004,3,3),KC(040),KM(4,3),KFFY(300),KSSX(300),KSSY(300)
      INTEGER RD,WR

      DATA RD,WR,KM(1,1),KM(1,2),KM(1,3),KM(2,1),KM(2,2),KM(2,3),KM(3,1)
     1,KM(3,2),KM(3,3)/1,3,.1666667,.08333333,.08333333,.08333333,.16666
     167,.08333333,.08333333,.08333333,.1666667/
C     ****************************************************************************
C     *  READ TITLE CARD
C     ****************************************************************************
      READ(RD,1000)(TITLE(I),I=1,20)
      WRITE(WR,1010)
      WRITE(WR,1020)(TITLE(I),I=1,20)
```

```
C     ****************************************************************************SWIG0027
C     *  READ CONTROL CARDS                                                        SWIG0028
C     ****************************************************************************SWIG0029
      READ(RD,1030)NN,NE,ITERMX,NPUMPS,NDATA,NOUT,JMAX,JTRC,NVEL,NTOE         SWIG0030
      IF(NDATA.EQ.1)WRITE(WR,1040)NN,NE,ITERMX,NPUMPS,                          SWIG0031
     1NDATA,NOUT,JMAX,JTRC,NVEL,NTOE                                            SWIG0032
      READ(RD,1045)DT,THETA,TOL,GAMAF,GAMAS,TSTART,BTOE                        SWIG0033
      IF(NDATA.EQ.1)WRITE(WR,1060)DT,THETA,TOL,GAMAF,GAMAS,TSTART,BTOE         SWIG0034
                                                                                SWIG0035
      READ (RD,1030)LMNTQF,LMNTQS,LMNTCF,NNODHF,NNODHS                         SWIG0036
      IF(NDATA.EQ.1)WRITE(WR,1065)LMNTQF,LMNTQS,LMNTCF,NNODHF,NNODHS           SWIG0037
C     ****************************************************************************SWIG0038
C     *  READ NODAL DATA. X & Y CO-ORDINATES, BOTTOM AND TOP ELEV. OF AQUI.SWIG0039
C     ****************************************************************************SWIG0040
      READ(RD,1070)(   XC(I),YC(I),ZB(I),ZU(I),I=1,NN)                         SWIG0041
      IF(NDATA.EQ.1)WRITE(WR,1080)(I,XC(I),YC(I),ZB(I),ZU(I),I=1,NN)           SWIG0042
C     ****************************************************************************SWIG0043
C     *                                                                         SWIG0044
C     *  READ ELEMENT DATA. NODE NOS. IN ELEM.,PERM.,PRSTY,NEW ELEM. GENERASWIG0045
C     ****************************************************************************SWIG0046
      I=1                                                                       SWIG0047
10    READ(RD,1090)NODE(I,1),NODE(I,2),NODE(I,3),KFFX(I),KFFY(I),KSSX(I)SWIG0048
     1,KSSY(I),PRSTY(I),NCONF(I),NLEAKY(I),NLMN,INC                            SWIG0049
      IF(NLMN.GT.0)GO TO 20                                                     SWIG0050
      I=I+1                                                                     SWIG0051
      IF(I.GT.NE)GO TO 40                                                       SWIG0052
      GO TO 10                                                                  SWIG0053
20    DO 30 K=1,NLMN                                                            SWIG0054
      I1=I+K                                                                    SWIG0055
      KINC=K*INC                                                                SWIG0056
      NODE(I1,1)=NODE(I,1)+KINC                                                 SWIG0057
      NODE(I1,2)=NODE(I,2)+KINC                                                 SWIG0058
      NODE(I1,3)=NODE(I,3)+KINC                                                 SWIG0059
      KFFX(I1)=KFFX(I)                                                          SWIG0060
      KFFY(I1)=KFFY(I)                                                          SWIG0061
      KSSX(I1)=KSSX(I)                                                          SWIG0062
      KSSY(I1)=KSSY(I)                                                          SWIG0063
      NCONF(I1)=NCONF(I)                                                        SWIG0064
30    PRSTY(I1)=PRSTY(I)                                                        SWIG0065
      I=I+NLMN+1                                                                SWIG0066
      IF(I.GT.NE)GO TO 40                                                       SWIG0067
      GO TO 10                                                                  SWIG0068
40    CONTINUE                                                                  SWIG0069
      IF(NDATA.EQ.1)WRITE(WR,1100)(I,NODE(I,1),NODE(I,2),NODE(I,3),KFFX(SWIG0070
     1I),KFFY(I),KSSX(I),KSSY(I),PRSTY(I),NCONF(I),NLEAKY(I),I=1,NE)         SWIG0071
C     ****************************************************************************SWIG0072
C     *  READ IN RECHARGE DATA                                                  SWIG0073
C     ****************************************************************************SWIG0074
      I=0                                                                       SWIG0075
29    I=I+1                                                                     SWIG0076
```

```
      IF(NCONF(I).EQ.1)GO TO 31
      IF(NDATA.EQ.1)WRITE(WR,1105)
      READ(RD,1030)I,INC
      READ(RD,1050)(RECHG(I,J),J=1,JMAX)
      IF(NDATA.EQ.1)WRITE(WR,1106)I,INC,(J,RECHG(I,J),J=1,JMAX)
      IF(I.EQ.NE)GO TO 32
      IF(INC.EQ.0)GO TO 29
      I1=I+1
      KINC=I+INC
      DO 33 K=I1,KINC
      DO 33 J=1,JMAX
33    RECHG(K,J)=RECHG(I,J)
      I=KINC
31    IF(I.EQ.NE)GO TO 32
      GO TO 29
C   ******************************************************************
C   *  READ DATA ON BOUNDARY CONDITIONS
C   ******************************************************************
C   *
C   *  BOUNDARIES ALONG WHICH FRESHWATER FLOW IS SPECIFIED.
C   ******************************************************************
32    IF(LMNTQF.EQ.0)GO TO 41
      IF(NDATA.EQ.1)WRITE(WR,1114)
      DO 34 I=1,LMNTQF
      READ(RD,1110)I1,NODEQF(I,1),NODEQF(I,2),(QFWBND(I,J),J=1,JMAX)
34    IF(NDATA.EQ.1)WRITE(WR,1115)     (I,NODEQF(I,1),NODEQF(I,2),(J,
     1QFWBND(I,J),J=1,JMAX))
C   ******************************************************************
C   *  BOUNDARIES ALONG WHICH SALTWATER FLOW IS SPECIFIED.
C   ******************************************************************
41    IF(LMNTQS.EQ.0)GO TO 42
      IF(NDATA.EQ.1)WRITE(WR,1124)
      DO 35 I=1,LMNTQS
      READ(RD,1110)I1,NODEQS(I,1),NODEQS(I,2),(QSWBND(I,J),J=1,JMAX)
35    IF(NDATA.EQ.1)WRITF(WR,1125)(I,NODEQS(I,1),NODEQS(I,2),(J,QSWBND(I
     1,J),J=1,JMAX))
C   ******************************************************************
C   *  COASTAL BOUNDARIES ALONG WHICH MIXED BOUND. COND. IS TO BE APPLIED
C   ******************************************************************
42    IF(LMNTCF.EQ.0)GO TO 43
      IF(NDATA.EQ.1)WRITE(WR,1127)
      DO 36 I=1,LMNTCF
      READ(RD,1130)I1,NODECF(I,1),NODECF(I,2),KC(I)
36    IF(NDATA.EQ.1)WRITE(WR,1126)I,NODECF(I,1),NODECF(I,2),KC(I)
C   ******************************************************************
C   *  NODES AT WHICH FRESHWATER HEAD IS SPECIFIED.
C   ******************************************************************
43    IF(NNODHF.EQ.0)GO TO 44
      IF(NDATA.EQ.1)WRITE(WR,1134)
      DO 37 I=1,NNODHF
```

```
      READ(RD,1140)I1,NODEHF(I),(SPECHF(I,J),J=1,JMAX)
37    IF(NDATA.EQ.1)WRITE(WR,1135)I,NODEHF(I),(J,SPECHF(I,J),J=1,JMAX)
C ********************************************************************
C *  NODES AT WHICH SALTWATER HEAD IS SPECIFIED.
C ********************************************************************
44    IF(NNODHS.EQ.0)GO TO 45
      IF(NDATA.EQ.1)WRITE(WR,1144)
      DO 38 I=1,NNODHS
      READ(RD,1140)I1,NODEHS(I),(SPECHS(I,J),J=1,JMAX)
38    IF(NDATA.EQ.1)WRITE(WR,1145)(I,NODEHS(I),(J,SPECHS(I,J),J=1,JMAX))
C ********************************************************************
C *  READ IN PUMP DATA.
C ********************************************************************
C
45    IF(NPUMPS.EQ.0)GO TO 46
      IF(NDATA.EQ.1)WRITE(WR,1160)NPUMPS
      I=1
50    READ(RD,1030)NODEP(I),NELEM(I)
      READ(RD,1050)(QPUMP(I,J),J=1,JMAX)
      IF(NDATA.EQ.1)WRITE(WR,1150)I,NODEP(I),NELEM(I)
      IF(NDATA.EQ.1)WRITE(WR,1165)(J,QPUMP(I,J),J=1,JMAX)
      I=I+1
      IF(I.LE.NPUMPS)GO TO 50
C ********************************************************************
C *  DATA ON LEAKY AQUITARD ABOVE AQUIFER.
C ********************************************************************
46    I=1
58    IF(NCONF(I).EQ.0.OR.NLEAKY(I).EQ.0)GO TO 54
53    READ(RD,1190)I,KO(I),PHIO(I),NLMN
      IF(NLMN.EQ.0)GO TO 54
      DO 52 J=1,NLMN
      I1=I+J

      KO(I1)=KO(I)
      PHIO(I1)=PHIO(I)
52    CONTINUE
      I=I+NLMN+1
      IF(I.GT.NE)GO TO 55
      GO TO 53
54    I=I+1
      IF(I.GT.NE)GO TO 55
      IF(NCONF(I-1).EQ.0)GO TO 58
      GO TO 53
55    I=1
      J=1
57    IF(NCONF(I).EQ.0.OR.NLEAKY(I).EQ.0)GO TO 56
      IF(NDATA.EQ.1.AND.J.EQ.1)WRITE(WR,1200)
      IF(NDATA.EQ.1)WRITE(WR,1205) I,KO(I),PHIO(I)
      J=2
56    I=I+1
```

```
      IF(I.GT.NE)GO TO 48
      GO TO 57
C     ******************************************************************
C     * READ IN INITIAL CONDITIONS
C     ******************************************************************
48    J=1
47    I=1
51    READ(RD,1180)I,ZETA(I),NND
      IF(NND.GT.0)GO TO 60
      I=I+1
      IF(I.GT.NN)GO TO 70
      GO TO 51
60    DO 61 K=1,NND
      I1=I+K
61    ZETA(I1)=ZETA(I)
      I=I+NND+1
      IF(I.GT.NN)GO TO 70
      GO TO 51
70    CONTINUE
      GO TO (80,90,100,110,115),J
80    DO 81 I=1,NN
      HF(I,1)=ZETA(I)
81    CONTINUE
      J=2
      GO TO 47
90    DO 91 I=1,NN
      HS(I,1)=ZETA(I)
91    CONTINUE
      J=3
      GO TO 47
100   DO 101 I=1,NN
101   BF(I,1)=ZETA(I)
      J=4
      GO TO 47
110   DGAMA=GAMAS-GAMAF
      DO 111 I=1,NN
      BS(I,1)=ZETA(I)
111   CONTINUE
C
C     ******************************************************************
C     * READ IN ELEMENT TYPE AND OUTPUT INFORMATION.
C     ******************************************************************
115   READ(RD,1225)((  LMNTYP(I)),I=1,NE)

      IF(NDATA.EQ.1)WRITE(WR,1230)(I,LMNTYP(I),I=1,NE)
      READ(RD,1225)(IO(I),I=1,NN)
      IF(NDATA.EQ.1)WRITE(WR,1170)
      IF(NDATA.EQ.1)WRITE(WR,1175)(I,HF(I,1),HS(I,1),BF(I,1),BS(I,1),
     1I=1,NN)
C
```

```
C  ****************************************************************************
C  *  CALCULATE CONSTANTS FOR EACH ELEMENT
C  ****************************************************************************
120   DO 150 I=1,NE
      N1=NODE(I,1)
      N2=NODE(I,2)
      N3=NODE(I,3)
      A(I,1)=XC(N3)-XC(N2)
      A(I,2)=XC(N1)-XC(N3)
      A(I,3)=XC(N2)-XC(N1)
      B(I,1)=YC(N2)-YC(N3)
      B(I,2)=YC(N3)-YC(N1)
      B(I,3)=YC(N1)-YC(N2)
      AREA(I)=0.5*(A(I,3)*B(I,2)-A(I,2)*B(I,3))
      IF(AREA(I).LT.0.)WRITE(WR,1510)I,AREA(I)
150   CONTINUE

      IF(LMNTQF.EQ.0)GO TO 135
      DO 131 J=1,LMNTQF
132   N1=NODEQF(J,1)
      N2=NODEQF(J,2)
      ELQF(J)=SQRT((XC(N1)-XC(N2))**2+(YC(N1)-YC(N2))**2)

131   CONTINUE
135   IF(LMNTQS.EQ.0)GO TO 140
      DO 136 J=1,LMNTQS
137   N1=NODEQS(J,1)
      N2=NODEQS(J,2)
      ELQS(J)=SQRT((XC(N1)-XC(N2))**2+(YC(N1)-YC(N2))**2)

136   CONTINUE
140   IF(LMNTCF.EQ.0)GO TO 145
      DO 141 J=1,LMNTCF
142   N1=NODECF(J,1)
      N2=NODECF(J,2)
      ELCF(J)=SQRT((XC(N1)-XC(N2))**2+(YC(N1)-YC(N2))**2)

141   CONTINUE
145   CONTINUE

      IF(JTRC.GT.0)WRITE(WR,1240)(I,A(I,1),A(I,2),A(I,3),B(I,1),B(I,2),
     1B(I,3),AREA(I),I=1,NE)
      IF(JTRC.GT.0.AND.LMNTQF.GT.0)WRITE(WR,1250)(I,NODEQF(I,1),NODEQF(I
     1,2),         ELQF(I),I=1,LMNTQF)
      IF(JTRC.GT.0.AND.LMNTQS.GT.0)WRITE(WR,1260)(I,NODEQS(I,1),NODEQS(I
     1,2),         ELQS(I),I=1,LMNTQS)
      IF(JTRC.GT.0.AND.LMNTCF.GT.0)WRITE(WR,1270)(I,NODECF(I,1),NODECF(I
     1,2)          ,ELCF(I),I=1,LMNTCF)
C  ****************************************************************************
C  *  CALCULATE ELEMENT MATRICES THAT DO NOT CHANGE WITH EACH ITERATION
```

```
C     ****************************************************************
C     *
C     *  MATRIX KFIJ(I,J,K)   AND   KSIJ(I,J,K)
C     ****************************************************************
C
      GAMASF=GAMAS/GAMAF
      DO 160 I=1,NE
      DO 160 J=1,3
      DO 160 K=1,3
      KFIJ(I,J,K)=PRSTY(I)*GAMAF*AREA(I)/(DT*DGAMA)*KM(J,K)
      IF(ABS(KFIJ(I,J,K)).LT.1.0E-10)KFIJ(I,J,K)=0.
160   KSIJ(I,J,K)=KFIJ(I,J,K)*GAMASF
      IF(JTRC.GT.0)WRITE(WR,1280)(I,KFIJ(I,1,1),KFIJ(I,1,2),KFIJ(I,1,3),
     1KFIJ(I,2,1),KFIJ(I,2,2),KFIJ(I,2,3),KFIJ(I,3,1),KFIJ(I,3,2),
     2KFIJ(I,3,3),I=1,NE)
      IF(JTRC.GT.0)WRITE(WR,1290)(I,((KSIJ(I,J,K),K=1,3),J=1,3),I=1,NE)
C
C     ****************************************************************
C     *  MATRIX KOIJ(I,J,K)
C     ****************************************************************
      DO 170 I=1,NE
      IF(NCONF(I).EQ.0.OR.NLEAKY(I).EQ.0)GO TO 170
      DO 170 J=1,3
      DO 170 K=1,3
      KOIJ(I,J,K)=KO(I)*AREA(I)*KM(J,K)
170   CONTINUE
      IF(JTRC.EQ.0)GO TO 180
      WRITE(WR,1300)
      DO 175 I=1,NE
      IF(              NCONF(I).EQ.1.AND.NLEAKY(I).EQ.1)WRITE(WR,1305)I,
     1((KOIJ(I,J,K),K=1,3),J=1,3)
175   CONTINUE
180   JT=1
      T=TSTART
      THETA1=1.-THETA
      NN2=NN*2
C     ****************************************************************
C     *  ADVANCE TIME T BY DT
C     ****************************************************************
200   T=T+DT
      ITER=1
      JT=JT+1
      DO 210 I=1,NN
      HF(I,2)=HF(I,1)
      HS(I,2)=HS(I,1)
      BF(I,2)=BF(I,1)
210   BS(I,2)=BS(I,1)
215   DO 225 I=1,NE
      KFX(I)=KFFX(I)
      KFY(I)=KFFY(I)
```

```
      KSX(I)=KSSX(I)
      KSY(I)=KSSY(I)
      IF(LMNTYP(I).EQ.1)KFX(I)=KFFX(I)/1.E05
      IF(LMNTYP(I).EQ.1)KFY(I)=KFFY(I)/1.E05
      IF(LMNTYP(I).EQ.2)KSX(I)=KSSX(I)/1.E05
      IF(LMNTYP(I).EQ.2)KSY(I)=KSSY(I)/1.E05
      IF(LMNTYP(I).NE.3)GO TO 216
      KFX(I)=KFFX(I)/1.E05
      KFY(I)=KFFY(I)/1.E05
      KSX(I)=KSSX(I)/1.E05
      KSY(I)=KSSY(I)/1.E05
216   DO 225 J=1,3
      DO 225 K=1,3
      KIJF(I,J,K)=-(KFX(I)*B(I,J)*B(I,K)+KFY(I)*A(I,J)*A(I,K))/(4.*AREA(
     1I))
225   KIJS(I,J,K)=-(KSX(I)*B(I,J)*B(I,K)+KSY(I)*A(I,J)*A(I,K))/(4.*AREA(
     1I))
      IF(JTRC.GT.0)WRITE(WR,1310)(I,((KIJF(I,J,K),K=1,3),J=1,3),I=1,NE)
      IF(JTRC.GT.0)WRITE(WR,1320)(I,((KIJS(I,J,K),K=1,3),J=1,3),I=1,NE)
C     ************************************************************************
C     *  SET GLOBAL MATRIX COEFICIENTS TO ZERO
C     ************************************************************************
      DO 220 I=1,NN2
      R(I)=0.
      DO 220 J=1,NN2
      AA(I,J)=0.
220   CONTINUE
C     ************************************************************************
C     *  LOOP THROUGH EACH ELEMENT TO CALCULATE ELEMENT MATRIX COEFICIENTS
C     *  AND ADD THEM UP TO FORM THE GLOBAL MATRIX.
C     ************************************************************************
      DO 500 I=1,NE
      N1=NODE(I,1)
      N2=NODE(I,2)
      N3=NODE(I,3)
      BFE=(0.5        *(BF(N1,1)+BF(N2,1)+BF(N3,1))+0.5   *(BF(N1,2)+
     1BF(N2,2)+BF(N3,2)))/3.
      BSE=(0.5    *(BS(N1,1)+BS(N2,1)+BS(N3,1))+0.5   *(BS(N1,2)+BS(N2,2)+
     1BS(N3,2)))/3.
      IF(JT.EQ.JTRC)WRITE(WR,1330)I,N1,N2,N3,BFE,BSE
      DO 230 J=1,3
      R1(J)=-BFE*THETA1*(KIJF(I,J,1)*HF(N1,1)+KIJF(I,J,2)*HF(N2,1)+KIJF(
     1I,J,3)    *HF(N3,1))                       -(KFIJ(I,J,1)*HF(N1,1)+
     2KFIJ(I,J,2)*HF(N2,1)+KFIJ(I,J,3)*HF(N3,1))+(KSIJ(I,J,1)*HS(N1,1)+
     3KSIJ(I,J,2)*HS(N2,1)+KSIJ(I,J,3)*HS(N3,1))
      IF(NCONF(I).EQ.0)R1(J)=R1(J)-(RECHG(I,JT)*AREA(I)/3.)
      IF(NCONF(I).EQ.1.AND.NLEAKY(I).EQ.1)R1(J)=R1(J)
     1                        +KO(I)*PHIO(I)*AREA(I)/3.-THETA1*KO(I)*
     2AREA(I)*(KM(J,1)*HF(N1,1)+KM(J,2)*HF(N2,1)+KM(J,3)*HF(N3,1))
      IF(NPUMPS.EQ.0)GO TO 231
```

```
      DO 232 L=1,NPUMPS                                                  SWIG0378
      IF(NODEP(L).EQ.NODE(I,J))R1(J)=R1(J)-QPUMP(L,JT)/FLOAT(NELEM(L))   SWIG0379
232   CONTINUE                                                           SWIG0380
231   CONTINUE                                                           SWIG0381
      DO 230 K=1,3                                                       SWIG0382
      A1(J,K)=BFE*THETA*KIJF(I,J,K)-KFIJ(I,J,K)                          SWIG0383
      IF(NCONF(I).EQ.1.AND.NLEAKY(I).EQ.1)A1(J,K)=A1(J,K)+KOIJ(I,J,K)    SWIG0384
      A2(J,K)=KSIJ(I,J,K)                                                SWIG0385
230   CONTINUE                                                           SWIG0386
      IF(JT.NE.JTRC)GO TO 239                                            SWIG0387
      WRITE(WR,1350)I,A1(1,1),A2(1,1),A1(1,2),A2(1,2),A1(1,3),A2(1,3),R1SWIG0388
     1(1)                                                                SWIG0389
      WRITE(WR,1350)I,A1(2,1),A2(2,1),A1(2,2),A2(2,2),A1(2,3),A2(2,3),R1SWIG0390
     1(2)                                                                SWIG0391
      WRITE(WR,1350)I,A1(3,1),A2(3,1),A1(3,2),A2(3,2),A1(3,3),A2(3,3),R1SWIG0392
     1(3)                                                                SWIG0393
239   CONTINUE                                                           SWIG0394
      DO 240 J=1,3                                                       SWIG0395
      R(2*NODE(I,J)-1)=R1(J)+R(2*NODE(I,J)-1)                            SWIG0396
      DO 240 K=1,3                                                       SWIG0397
      AA(2*NODE(I,J)-1,2*NODE(I,K)-1)=A1(J,K)+AA(2*NODE(I,J)-1,2*NODE(I,SWIG0398
     1K)-1)                                                              SWIG0399
      AA(2*NODE(I,J)-1,2*NODE(I,K))=A2(J,K)+AA(2*NODE(I,J)-1,2*NODE(I,K)SWIG0400
     1)                                                                  SWIG0401
240   CONTINUE                                                           SWIG0402
      DO 250 J=1,3                                                       SWIG0403
                                                                         SWIG0404
      R1(J)=-BSE*THETA1*(KIJS(I,J,1)*HS(N1,1)+KIJS(I,J,2)*HS(N2,1)+KIJS(SWIG0405
     1I,J,3)     *HS(N3,1))-(KSIJ(I,J,1)*HS(N1,1)+KSIJ(I,J,2)*HS(N2,1)+  SWIG0406
     2KSIJ(I,J,3)*HS(N3,1))+(KFIJ(I,J,1)*HF(N1,1)+KFIJ(I,J,2)*HF(N2,1)+  SWIG0407
     3KFIJ(I,J,3)*HF(N3,1))                                              SWIG0408
251   DO 250 K=1,3                                                       SWIG0409
      A1(J,K)=BSE*THETA*KIJS(I,J,K)-KSIJ(I,J,K)                          SWIG0410
      A2(J,K)=KFIJ(I,J,K)                                                SWIG0412
250   CONTINUE                                                           SWIG0413
      IF(JT.NE.JTRC)GO TO 259                                            SWIG0414
      WRITE(WR,1350)I,A2(1,1),A1(1,1),A2(1,2),A1(1,2),A2(1,3),A1(1,3),   SWIG0415
     1R1(1)                                                              SWIG0416
      WRITE(WR,1350)I,A2(2,1),A1(2,1),A2(2,2),A1(2,2),A2(2,3),A1(2,3),   SWIG0417
     1R1(2)                                                              SWIG0418
      WRITE(WR,1350)I,A2(3,1),A1(3,1),A2(3,2),A1(3,2),A2(3,3),A1(3,3),   SWIG0419
     1R1(3)                                                              SWIG0420
259   CONTINUE                                                           SWIG0421
      DO 260 J=1,3                                                       SWIG0422
      R(2*NODE(I,J))=R1(J)+R(2*NODE(I,J))                                SWIG0423
      DO 260 K=1,3                                                       SWIG0424
      AA(2*NODE(I,J),2*NODE(I,K))=A1(J,K)+AA(2*NODE(I,J),2*NODE(I,K))    SWIG0425
      AA(2*NODE(I,J),2*NODE(I,K)-1)=A2(J,K)+AA(2*NODE(I,J),2*NODE(I,K)-1SWIG0426
     1)                                                                  SWIG0427
260   CONTINUE                                                           SWIG0428
```

```
C    **************************************************************************SWIG0429
C    *  CORRECT MATRIX FOR BOUNDARY CONDITIONS.                                 SWIG0430
C    **************************************************************************SWIG0431
      IF(LMNTQF.EQ.0)GO TO 272                                                  SWIG0432
      DO 270 L=1,LMNTQF                                                         SWIG0433
      IF(NODEQF(L,1).EQ.NODE(I,1).AND.NODEQF(L,2).EQ.NODE(I,2))GO TO 271SWIG0434
      IF(NODEQF(L,1).EQ.NODE(I,2).AND.NODEQF(L,2).EQ.NODE(I,3))GO TO 271SWIG0435
      IF(NODEQF(L,1).EQ.NODE(I,3).AND.NODEQF(L,2).EQ.NODE(I,1))GO TO 271SWIG0436
      GO TO 270                                                                 SWIG0437
271   R(2*NODEQF(L,1)-1)=R(2*NODEQF(L,1)-1)-QFWBND(L,JT)*ELQF(L)/2.            SWIG0438
      R(2*NODEQF(L,2)-1)=R(2*NODEQF(L,2)-1)-QFWBND(L,JT)*ELQF(L) /2.           SWIG0439
      GO TO 272                                                                 SWIG0440
270   CONTINUE                                                                  SWIG0441
272   IF(LMNTQS.EQ.0)GO TO 282                                                  SWIG0442
      DO 280 L=1,LMNTQS                                                         SWIG0443
      IF(NODEQS(L,1).EQ.NODE(I,1).AND.NODEQS(L,2).EQ.NODE(I,2))GO TO 281SWIG0444
      IF(NODEQS(L,1).EQ.NODE(I,2).AND.NODEQS(L,2).EQ.NODE(I,3))GO TO 281SWIG0445
      IF(NODEQS(L,1).EQ.NODE(I,3).AND.NODEQS(L,2).EQ.NODE(I,1))GO TO 281SWIG0446
      GO TO 280                                                                 SWIG0447
281   R(2*NODEQS(L,1))=R(2*NODEQS(L,1))-QSWBND(L,JT)*ELQS(L)/2.                SWIG0448
      R(2*NODEQS(L,2))=R(2*NODEQS(L,2))-QSWBND(L,JT)*ELQS(L)/2.                SWIG0449
      GO TO 282                                                                 SWIG0450
280   CONTINUE                                                                  SWIG0451
282   IF(LMNTCF.EQ.0)GO TO 294                                                  SWIG0452
      DO 290 L=1,LMNTCF                                                         SWIG0453
      IF(NODECF(L,1).EQ.NODE(I,1).AND.NODECF(L,2).EQ.NODE(I,2))GO TO 291SWIG0454
      IF(NODECF(L,1).EQ.NODE(I,2).AND.NODECF(L,2).EQ.NODE(I,3))GO TO 291SWIG0455
      IF(NODECF(L,1).EQ.NODE(I,3).AND.NODECF(L,2).EQ.NODE(I,1))GO TO 291SWIG0456
      GO TO 290                                                                 SWIG0457
291   NCFL1=NODECF(L,1)                                                         SWIG0458
      NCFL2=NODECF(L,2)                                                         SWIG0459
      IF(JT.NE.JTRC)GO TO 292                                                   SWIG0460
      WRITE(WR,1350)NCFL1,AA(2*NCFL1-1,2*NCFL1-1),AA(2*NCFL1-1,2*NCFL2-1SWIG0461
     1),AA(2*NCFL1-1,2*NCFL1),AA(2*NCFL1-1,2*NCFL2),R(2*NCFL1-1)              SWIG0462
      WRITE(WR,1350)NCFL2,AA(2*NCFL2-1,2*NCFL1-1),AA(2*NCFL2-1,2*NCFL2-1SWIG0463
     1),AA(2*NCFL2-1,2*NCFL1),AA(2*NCFL2-1,2*NCFL2),R(2*NCFL2-1)              SWIG0464
292   CONTINUE                                                                  SWIG0465
      DO 295 LL=1,NNODHS                                                        SWIG0466
      IF(NODEHS(LL).EQ.NCFL1)L1=LL                                              SWIG0467
      IF(NODEHS(LL).EQ.NCFL2)L2=LL                                              SWIG0468
295   CONTINUE                                                                  SWIG0469
      FACT=KC(L)*ELCF(L)                                                        SWIG0470
      R(2*NCFL1-1)=R(2*NCFL1-1)+FACT*0.5   *(HF(NCFL1,1)/3.+HF(NCFL2,1)/SWIG0471
     16.-SPECHS(   L1,JT-1)/3.-SPECHS(   L2,JT-1)/6.)                         SWIG0472
      R(2*NCFL2-1)=R(2*NCFL2-1)+FACT*0.5   *(HF(NCFL1,1)/6.+HF(NCFL2,1)/SWIG0473
     13.-SPECHS(   L1,JT-1)/6.-SPECHS(   L2,JT-1)/3.)                         SWIG0474
      AA(2*NCFL1-1,2*NCFL1-1)=AA(2*NCFL1-1,2*NCFL1-1)-FACT/3.*0.5              SWIG0475
      AA(2*NCFL1-1,2*NCFL2-1)=AA(2*NCFL1-1,2*NCFL2-1)-FACT/6.*0.5              SWIG0476
      AA(2*NCFL1-1,2*NCFL1)=   AA(2*NCFL1-1,2*NCFL1)   +FACT/3.*0.5            SWIG0477
      AA(2*NCFL1-1,2*NCFL2)=   AA(2*NCFL1-1,2*NCFL2)   +FACT/6.*0.5            SWIG0478
```

```
      AA(2*NCFL2-1,2*NCFL1-1)=AA(2*NCFL2-1,2*NCFL1-1)-FACT/6.*0.5       SWIG0479
      AA(2*NCFL2-1,2*NCFL2-1)=AA(2*NCFL2-1,2*NCFL2-1)-FACT/3.*0.5       SWIG0480
      AA(2*NCFL2-1,2*NCFL1)  =AA(2*NCFL2-1,2*NCFL1)  +FACT/6.*0.5       SWIG0481
      AA(2*NCFL2-1,2*NCFL2)  =AA(2*NCFL2-1,2*NCFL2)  +FACT/3.*0.5       SWIG0482
      IF(JT.NE.JTRC)GO TO 293                                           SWIG0483
      WRITE(WR,1350)NCFL1,AA(2*NCFL1-1,2*NCFL1-1),AA(2*NCFL1-1,2*NCFL2-1SWIG0484
     1),R(2*NCFL1-1)                                                    SWIG0485
      WRITE(WR,1350)NCFL2,AA(2*NCFL2-1,2*NCFL1-1),AA(2*NCFL2-1,2*NCFL2-1SWIG0486
     1),R(2*NCFL2-1)                                                    SWIG0487
  293 CONTINUE                                                          SWIG0488
  290 CONTINUE                                                          SWIG0489
  294 IF(JT.NE.JTRC)GO TO 500                                           SWIG0490
      N1=2*NODE(I,1)-1                                                  SWIG0491
      N2=2*NODE(I,1)                                                    SWIG0492
      N3=2*NODE(I,2)-1                                                  SWIG0493
      N4=2*NODE(I,2)                                                    SWIG0494
      N5=2*NODE(I,3)-1                                                  SWIG0495
      N6=2*NODE(I,3)                                                    SWIG0496
      WRITE(WR,1340)N1,N2,N3,N4,N5,N6                                   SWIG0497
      WRITE(WR,1350)N1,AA(N1,N1),AA(N1,N2),AA(N1,N3),AA(N1,N4),AA(N1,N5)SWIG0498
     1,AA(N1,N6),R(N1)                                                  SWIG0499
      WRITE(WR,1350)N2,AA(N2,N1),AA(N2,N2),AA(N2,N3),AA(N2,N4),AA(N2,N5)SWIG0500
     1,AA(N2,N6),R(N2)                                                  SWIG0501
      WRITE(WR,1350)N3,AA(N3,N1),AA(N3,N2),AA(N3,N3),AA(N3,N4),AA(N3,N5)SWIG0502
     1,AA(N3,N6),R(N3)                                                  SWIG0503
      WRITE(WR,1350)N4,AA(N4,N1),AA(N4,N2),AA(N4,N3),AA(N4,N4),AA(N4,N5)SWIG0504
     1,AA(N4,N6),R(N4)                                                  SWIG0505
      WRITE(WR,1350)N5,AA(N5,N1),AA(N5,N2),AA(N5,N3),AA(N5,N4),AA(N5,N5)SWIG0506
     1,AA(N5,N6),R(N5)                                                  SWIG0507
      WRITE(WR,1350)N6,AA(N6,N1),AA(N6,N2),AA(N6,N3),AA(N6,N4),AA(N6,N5)SWIG0508
     1,AA(N6,N6),R(N6)                                                  SWIG0509
  500 CONTINUE                                                          SWIG0510
      IF(JT.NE.JTRC)GO TO 340                                           SWIG0511
      N1=NN2/120+1                                                      SWIG0512
      DO 310 I=1,N1                                                     SWIG0513
      N2=120                                                            SWIG0514
      IF(I.EQ.N1)N2=NN2-(N1-1)*120                                      SWIG0515
      DO 320 J=1,NN2                                                    SWIG0516
      N3=(I-1)*120+1                                                    SWIG0517
      N4=N3+N2-1                                                        SWIG0518
      DO330 K=N3,N4                                                     SWIG0519
                                                                        SWIG0520
      IF(AA(J,K).LT.-1000.)N7(K)=1                                      SWIG0521
      IF(AA(J,K).GT.-1000..AND.AA(J,K).LE.-100.)N7(K)=2                 SWIG0522
      IF(AA(J,K).GT.-100.0.AND.AA(J,K).LE.-10.0)N7(K)=3                 SWIG0523
      IF(AA(J,K).GT.-10.00.AND.AA(J,K).LT.0.   )N7(K)=4                 SWIG0524
      IF(AA(J,K).EQ.0.)N7(K)=0                                          SWIG0525
      IF(AA(J,K).GT.0..AND.AA(J,K).LE.10.00)N7(K)=6                     SWIG0526
      IF(AA(J,K).GT.10.00.AND.AA(J,K).LE.100.0)N7(K)=7                  SWIG0527
      IF(AA(J,K).GT.100.0.AND.AA(J,K).LE.1000.)N7(K)=8                  SWIG0528
```

```
330    IF(AA(J,K).GT.1000.)N7(K)=9
       WRITE(WR,1360)J,(N7(K),K=N3,N4),J
320    CONTINUE
310.   CONTINUE
       N1=NN2/10*10-9
       N2=N1+10
       N3=NN2-N2
       DO 335 I=1,NN2

       WRITE(WR,1450)((I,J,AA(I,J),(AA(I,J+K),K=1,9)),J=1,N1,10)
335    WRITE(WR,1450)(I,N2,AA(I,N2),(AA(I,N2+K),K=1,N3))
       WRITE(WR,1460)((I,R(I),(R(I+K),K=1,9)),I=1,N1,10)
       WRITE(WR,1460)(N2,R(N2),(R(N2+K),K=1,N3))
340    CONTINUE
C  **********************************************************************
C  *   REDUCE MATRIX SIZE TO TAKE INTO ACCOUNT NODES AT WHICH THE FRESHWA
C  *   AND SALTWATER HEADS ARE SPECIFIED
C  **********************************************************************
       DO 501 I=1,NN2
       I2(I)=I
501    CONTINUE
       IF(NNODHF.EQ.0)GO TO 520
       DO 510 I=1,NNODHF
       J=2*NODEHF(I)-1
       DO 510 L=1,NN2
510    R(L)=R(L)-AA(L,J)*SPECHF(I,JT)
520    IF(NNODHS.EQ.0)GO TO 535
       DO 530 I=1,NNODHS
       J=2*NODEHS(I)
       DO 530 L=1,NN2
530    R(L)=R(L)-AA(L,J)*SPECHS(I,JT)
535    IF(NNODHF.EQ.0)GO TO 600
       DO 590 I=1,NNODHF
       NN2S=NN2-I
       N1=NN2S+1
       DO 540 J=1,N1
       IF(I2(J).EQ.(2*NODEHF(I)-1))GO TO 550
540    CONTINUE
550    DO 560 K=J,NN2S
       R(K)=R(K+1)
       I2(K)=I2(K+1)
       DO 560 L=1,N1
560    AA(K,L)=AA(K+1,L)
       DO 570 L=1,NN2S
       DO 570 K=J,NN2S
570    AA(L,K)=AA(L,K+1)
590    CONTINUE
600    IF(NNODHS.EQ.0)GO TO 700
       DO 690 I=1,NNODHS
       NN2S=NN2-NNODHF-I
```

```
      N1=NN2S+1                                                          SWIG0579
      DO 610 J=1,N1                                                      SWIG0580
      IF(I2(J).EQ.(2*NODEHS(I)))GO TO 620                                SWIG0581
610   CONTINUE                                                           SWIG0582
620   IF(I2(J).EQ.NN2)GO TO 690                                          SWIG0583
      DO 630 K=J,NN2S                                                    SWIG0584
      R(K)=R(K+1)                                                        SWIG0585
      I2(K)=I2(K+1)                                                      SWIG0586
      DO 630 L=1,N1                                                      SWIG0587
630   AA(K,L)=AA(K+1,L)                                                  SWIG0588
      DO 640 L=1,NN2S                                                    SWIG0589
      DO 640 K=J,NN2S                                                    SWIG0590
640   AA(L,K)=AA(L,K+1)                                                  SWIG0591
690   CONTINUE                                                           SWIG0592
700   IF(JT.NE.JTRC)GO TO 680                                            SWIG0593
      N1=NN2S/120+1                                                      SWIG0594
      DO 650 I=1,N1                                                      SWIG0595
      N2=120                                                             SWIG0596
      IF(I.EQ.N1)N2=NN2S-(N1-1)*120                                      SWIG0597
      DO 660 J=1,NN2S                                                    SWIG0598
      N3=(I-1)*120+1                                                     SWIG0599
      N4=N3+N2-1                                                         SWIG0600
      N5=I2(J)                                                           SWIG0601
      DO 670 K=N3,N4                                                     SWIG0602
      IF(AA(J,K).LE.-1000.)N7(K)=1                                       SWIG0603
      IF(AA(J,K).GT.-1000..AND.AA(J,K).LE.-100.0)N7(K)=2                 SWIG0604
      IF(AA(J,K).GT.-100.0.AND.AA(J,K).LE.-10.00)N7(K)=3                 SWIG0605
      IF(AA(J,K).GT.-10.00.AND.AA(J,K).LT.0.0000)N7(K)=4                 SWIG0606
      IF(AA(J,K).EQ.0.)N7(K)=0                                           SWIG0607
      IF(AA(J,K).GT.0.000.AND.AA(J,K).LE.10.00)N7(K)=6                   SWIG0608
      IF(AA(J,K).GT.10.00.AND.AA(J,K).LE.100.0)N7(K)=7                   SWIG0609
      IF(AA(J,K).GT.100.0.AND.AA(J,K).LE.1000.)N7(K)=8                   SWIG0610
      IF(AA(J,K).GT.1000.)N7(K)=9                                        SWIG0611
670   CONTINUE                                                           SWIG0612
      WRITE(WR,1360)N5,(N7(K),K=N3,N4),N5                                SWIG0613
660   CONTINUE                                                           SWIG0614
650   CONTINUE                                                           SWIG0615
      N1=NN2S/10*10-9                                                    SWIG0616
      N2=N1+10                                                           SWIG0617
      N3=NN2S-N2                                                         SWIG0618
      DO 675 I=1,NN2S                                                    SWIG0619
      WRITE(WR,1450)((I,J,AA(I,J),(AA(I,J+K),K=1,9)),J=1,N1,10)          SWIG0620
675   WRITE(WR,1450)(I,N2,AA(I,N2),(AA(I,N2+K),K=1,N3))                  SWIG0621
      WRITE(WR,1460)((I,R(I),(R(I+K),K=1,9)),I=1,N1,10)                  SWIG0622
      WRITE(WR,1460)(N2,R(N2),(R(N2+K),K=1,N3))                          SWIG0623
      WRITE(WR,1490)((I,I2(I),(I2(I+K),K=1,9)),I=1,N1,10)                SWIG0624
      WRITE(WR,1490)(N2,I2(N2),(I2(N2+K),K=1,N3))                        SWIG0625
680   CONTINUE                                                           SWIG0626
C  *********************************************************************SWIG0627
C  *   REDUCE MATRIX TO FORM REQUIRED BY GAUSS ELIMINATION ALGORITHM     SWIG0628
```

```
C     ******************************************************************
      DO 682 I=1,NN2S
      IF( ABS(R(I)).LT.1.0E-10)R(I)=0.
      DO 682 J=1,NN2S
      IF( ABS(AA(I,J)).LT.1.0E-10)AA(I,J)=0.
682   CONTINUE
      IF(ITER.GT.1.OR.JT.GT.2)GO TO 725
      MAXBW=0
      MINBW=0
      DO 710 I=1,NN2S
      DO 710 J=1,NN2S
      IF(AA(I,J).EQ.0.)GO TO 710
      IF(J-I)715,710,720
715   IF((I-J).GT.MINBW)MINBW=I-J
      GO TO 710
720   IF((J-I).GT.MAXBW)MAXBW=J-I
710   CONTINUE
      NBWMIN=1+MINBW
      NBWMAX=1+MAXBW
      NBWTOT=1+MAXBW+MINBW
      MC=MINO(NN2S,NBWTOT)
      ML=MC-NBWMIN
      MU=MC-NBWMAX
      MA=NN2S*MC-ML*(ML+1)/2
      ME=MA-MU*(MU+1)/2
      IF(JTRC.GT.00)WRITE(WR,1030)MC,ML,MU,MA,ME
725   DO 730 I=1,MA
      AAA(I)=0.
730   CONTINUE
      IF(NN2S.LT.NBWTOT)GO TO 761
      NBW=MAXBW
      IC=0
      DO 740 I=1,NBWMIN
      NBW=NBW+1
      DO 740 J=1,NBW
      IC=IC+1
      AAA(IC)=AA (I,J)
740   CONTINUE
      NBWU=1+NBWMIN
      NBWL=NN2S-NBWMAX
      IF(NN2S.EQ.(1+NBWTOT))GO TO 751
      DO 750 I=NBWU,NBWL
      DO 750 J=1,NN2S
      IF(J.LE.(I-NBWMIN).OR.J.GT.(I+MAXBW))GO TO 750
      IC=IC+1
      AAA(IC)=AA (I,J)
750   CONTINUE
751   NBWL=NBWL+1
      NBW=NBWTOT
      DO 760 I=NBWL,NN2S
```

```
      NBW=NBW-1
      DO 760 J=1,NN2S
      IF(J.LT.(NN2S-NBW))GO TO 760
      IC=IC+1
      AAA(IC)=AA (I,J)
760   CONTINUE
      GO TO 762
761   NBW=MAXBW
      IC=0
      N2=NN2S-NBWMAX
      DO 763 I=1,N2
      NBW=NBW+1
      DO 763 J=1,NBW
      IC=IC+1
      AAA(IC)=AA(I,J)
763   CONTINUE
      NBWU=N2+1
      IF(N2.EQ.0)NBWU=2
      NBWL=NBWMIN
      DO 765 I=NBWU,NBWL
      DO 765 J=1,NN2S
      IC=IC+1
      AAA(IC)=AA(I,J)
765   CONTINUE
      NBWL=NBWL+1
      NBW=NN2S
      IF(NN2S.LT.NBWL)GO TO 762
      DO 766 I=NBWL,NN2S
      NBW=NBW-1
      DO 766 J=1,NN2S
      IF(J.LE.(NN2S-NBW))GO TO 766
      IC=IC+1
      AAA(IC)=AA(I,J)
766   CONTINUE
762   IF(JTRC.GT.00)WRITE(WR,1370)NN2S,MINBW,MAXBW,IC
      IF(JT.NE.JTRC)GO TO 775
      N1=IC/10*10-9
      N2=N1+10
      N3=IC-N2
      WRITE(WR,1460)((I,AAA(I),(AAA(I+K),K=1,9)),I=1,N1,10)
      WRITE(WR,1460)(N2,AAA(N2),(AAA(N2+K),K=1,N3))
775   CONTINUE
C  ***********************************************************************
C  *   SOLVE MATRIX USING GELB
C  ***********************************************************************
      CALL GELB(       NN2S,1,MAXBW,MINBW,MA,1.0E-15,IER)
      IF(IER.EQ.0)GO TO 770
      WRITE(WR,1430)IER
      STOP
770   CONTINUE
```

```
      DO 780 I=1,NN2S
      IC=(I2(I)+1)/2
      IF((2*IC).EQ.(I2(I)+1))GO TO 771
      HSNEW(IC)=R(I)
      IF(JTRC.GT.0)WRITE(WR,1500)I,I2(I),IC,HSNEW(IC)
      GO TO 780
771   HFNEW(IC)=R(I)
      IF(HFNEW(IC).LT.0.)HFNEW(IC)=BTOE
      IF(JTRC.GT.0)WRITE(WR,1500)I,I2(I),IC,HFNEW(IC)
780   CONTINUE
      IF(NNODHF.EQ.0)GO TO 773
      DO 772 I=1,NNODHF
      N1=NODEHF(I)
772   HFNEW(N1)=SPECHF(I,JT)
773   IF(NNODHS.EQ.0)GO TO 776
      DO 774 I=1,NNODHS
      N1=NODEHS(I)
774   HSNEW(N1)=SPECHS(I,JT)
776   IF(JTRC.EQ.0)GO TO 783
      DO 781 I=1,NN2S
      FWRES=0.
      DO 782 J=1,NN2S
782   FWRES=AA(I,J)*R(J)+FWRES
      WRITE(WR,1450)I,I2(I),FWRES
781   CONTINUE
C    ***********************************************************************
C    *  CALCULATE NEW VALUES OF BF(I,2),BS(I,2) AND ZETA(I)
C    ***********************************************************************
783   DO 800 I=1,NN
      ZETA(I)=(GAMAS*HSNEW(I)-GAMAF*HFNEW(I))/DGAMA
      IF(ZETA(I).LE.(ZB(I)+BTOE))GO TO 820
      IF(INT(ZU(I)).NE.1000.AND.ZETA(I).GT.ZU(I))ZETA(I)=ZU(I)-BTOE
      BS(I,2)=ZETA(I)-ZB(I)
      BF(I,2)=HFNEW(I)-ZETA(I)
      IF(INT(ZU(I)).NE.1000)BF(I,2)=ZU(I)-ZETA(I)
      IF(BF(I,2).LT.BTOE)BF(I,2)=BTOE
      GO TO 830
820   ZETA(I)=ZB(I)+BTOE
      BF(I,2)=HFNEW(I)-ZB(I)-BTOE
      IF(INT(ZU(I)).NE.1000)BF(I,2)=ZU(I)-ZB(I)-BTOE
      BS(I,2)=BTOE
      IF(BF(I,2).LT.BTOE)BF(I,2)=BTOE
830   CONTINUE
800   CONTINUE
      DO 840 I=1,NE
      N1=NODE(I,1)
      N2=NODE(I,2)
      N3=NODE(I,3)
      LMNTYP(I)=0
      IF(BS(N1,2).EQ.BTOE.AND.BS(N2,2).EQ.BTOE.AND.BS(N3,2).EQ.BTOE)
```

```
     1LMNTYP(I)=2
      IF(BF(N1,2).EQ.BTOE.AND.BF(N2,2).EQ.BTOE.AND.BF(N3,2).EQ.BTOE)
     1LMNTYP(I)=1
      IF(BS(N1,2).EQ.BTOE.AND.BS(N2,2).EQ.BTOE.AND.BS(N3,2).EQ.BTOE.AND.
     1BF(N1,2).EQ.BTOE.AND.BF(N2,2).EQ.BTOE.AND.BF(N3,2).EQ.BTOE)LMNTYP
     2(I)=3
840   CONTINUE
C  **********************************************************************
C  *  CALCULATE ERROR TO CHECK IF IT IS .LE. TOL
C  **********************************************************************
      ERROR=0.
      BASE=0.
      DO 790 I=1,NN
      ERROR=ERROR+ SQRT((HS(I,2)-HSNEW(I))**2+(HF(I,2)-HFNEW(I))**2)
      BASE=BASE+SQRT(HS(I,1)**2+HF(I,1)**2)
790   CONTINUE
      IF((ERROR/BASE).LE.TOL)GO TO 810
      DO 793 I=1,NN
      HF(I,2)=HFNEW(I)
      HS(I,2)=HSNEW(I)
793   CONTINUE
      IF(JTRC.EQ.0)GO TO 860
      WRITE(WR,1380)JT,T,ITER,ERROR,BASE
      DO 792 I=1,NN
      WRITE(WR,1390)I,HF(I,2),HS(I,2),BF(I,2),BS(I,2),ZETA(I)
792   CONTINUE
860   IF(ITER.GE.ITERMX)GO TO 810
      WRITE(WR,1380)JT,T,ITER,ERROR,BASE
      ITER=ITER+1
      GO TO 215
810   DO 850 I=1,NN
      HF(I,2)=HFNEW(I)
      HS(I,2)=HSNEW(I)
      HF(I,1)=HF(I,2)
      HS(I,1)=HS(I,2)
      BF(I,1)=BF(I,2)
      BS(I,1)=BS(I,2)
850   CONTINUE
      IF(JT/NOUT*NOUT.EQ.JT)GO TO 900
      GO TO 990
C  **********************************************************************
C  *  OUTPUT OF RESULTS
C  **********************************************************************
900   WRITE(WR,1380)JT,T,ITER,ERROR,BASE
      WRITE(WR,1400)
      DO 910 I=1,NN
      IF(IO(I).EQ.0)GO TO 910
      WRITE(WR,1390)I,HF(I,2),HS(I,2),BF(I,2),BS(I,2),ZETA(I)
910   CONTINUE
      ZETA1=0.
```

```
      ZETA2=0.
      FRAC1=0.
      DO 915 I=1,NE
      ZETA1=ZETA1+AREA(I)
      ZETA2=ZETA2+(BF(NODE(I,1),2)+BF(NODE(I,2),2)+BF(NODE(I,3),2))*
     1AREA(I)/3.
      FRAC1=FRAC1+PRSTY(I) *(BF(NODE(I,1),2)+BF(NODE(I,2),2)+BF(NODE(I,3
     1),2))*AREA(I)/3.
915   CONTINUE
      ZETA3=ZETA2/ZETA1
      WRITE(WR,1520)ZETA1,FRAC1,ZETA3
      IF(NTOE.EJ.0)GO TO 927
      DO 921 I=1,NE
      N1=NODE(I,1)
      N2=NODE(I,2)
      N3=NODE(I,3)
      N4=1
      IF(BS(N1,2).GT.BTOE.AND.BS(N2,2).EQ.BTOE.AND.BS(N3,2).EQ.BTOE)N4=2
      IF(BS(N1,2).EQ.BTOE.AND.BS(N2,2).GT.BTOE.AND.BS(N3,2).EQ.BTOE)N4=2
      IF(BS(N1,2).EQ.BTOE.AND.BS(N2,2).EQ.BTOE.AND.BS(N3,2).GT.BTOE)N4=2
      IF(BS(N1,2).EQ.BTOE.AND.BS(N2,2).GT.BTOE.AND.BS(N3,2).GT.BTOE)N4=2
      IF(BS(N1,2).GT.BTOE.AND.BS(N2,2).EQ.BTOE.AND.BS(N3,2).GT.BTOE)N4=2
      IF(BS(N1,2).GT.BTOE.AND.BS(N2,2).GT.BTOE.AND.BS(N3,2).EQ.BTOE)N4=2
      GO TO (921,922),N4
922   ZETA1=(GAMAS*HS(N1,2)-GAMAF*HF(N1,2))/DGAMA
      ZETA2=(GAMAS*HS(N2,2)-GAMAF*HF(N2,2))/DGAMA
      ZETA3=(GAMAS*HS(N3,2)-GAMAF*HF(N3,2))/DGAMA
      FRAC1=(ZETA1-ZB(N1))/(ZETA2-ZB(N2))
      FRAC2=(ZETA2-ZB(N2))/(ZETA3-ZB(N3))
      FRAC3=(ZETA3-ZB(N3))/(ZETA1-ZB(N1))
      IF(FRAC1.LT.0.)F1=ABS(FRAC1)/(1.+ABS(FRAC1))
      IF(FRAC2.LT.C.)F2=ABS(FRAC2)/(1.+ABS(FRAC2))
      IF(FRAC3.LT.0.)F3=ABS(FRAC3)/(1.+ABS(FRAC3))
      IF(FRAC1.LT.0.)WRITE(WR,1420)I,N1,N2,F1
      IF(FRAC2.LT.0.)WRITE(WR,1420)I,N2,N3,F2
      IF(FRAC3.LT.0.)WRITE(WR,1420)I,N3,N1,F3
921   CONTINUE
      DO 923 I=1,NE
      N1=NODE(I,1)
      N2=NODE(I,2)
      N3=NODE(I,3)
      N4=1
      IF(BF(N1,2).GT.BTOE.AND.BF(N2,2).EQ.BTOE.AND.BF(N3,2).EQ.BTOE)N4=2
      IF(BF(N1,2).EQ.BTOE.AND.BF(N2,2).GT.BTOE.AND.BF(N3,2).EQ.BTOE)N4=2
      IF(BF(N1,2).EQ.BTOE.AND.BF(N2,2).EQ.BTOE.AND.BF(N3,2).GT.BTOE)N4=2
      IF(BF(N1,2).EQ.BTOE.AND.BF(N2,2).GT.BTOE.AND.BF(N3,2).GT.BTOE)N4=2
      IF(BF(N1,2).GT.BTOE.AND.BF(N2,2).EQ.BTOE.AND.BF(N3,2).GT.BTOE)N4=2
      IF(BF(N1,2).GT.BTOE.AND.BF(N2,2).GT.BTOE.AND.BF(N3,2).EQ.BTOE)N4=2
      GO TO (923,924),N4
924   ZETA1=HF(N1,2)
```

```
      ZETA2=HF(N2,2)                                                    SWIG0879
      ZETA3=HF(N3,2)                                                    SWIG0880
      IF(NCONF(I).EQ.1)ZETA1=(GAMAS*HS(N1,2)-GAMAF*HF(N1,2))/DGAMA      SWIG0881
      IF(NCONF(I).EQ.1)ZETA2=(GAMAS*HS(N2,2)-GAMAF*HF(N2,2))/DGAMA      SWIG0882
      IF(NCONF(I).EQ.1)ZETA3=(GAMAS*HS(N3,2)-GAMAF*HF(N3,2))/DGAMA      SWIG0883
      FRAC1=(ZETA1-ZB(N1))/(ZETA2-ZB(N2))                               SWIG0884
      FRAC2=(ZETA2-ZB(N2))/(ZETA3-ZB(N3))                               SWIG0885
      FRAC3=(ZETA3-ZB(N3))/(ZETA1-ZB(N1))                               SWIG0886
      IF(NCONF(I).EQ.1)FRAC1=(ZETA1-ZU(N1))/(ZETA2-ZU(N2))              SWIG0887
      IF(NCONF(I).EQ.1)FRAC2=(ZETA2-ZU(N2))/(ZETA3-ZU(N3))              SWIG0888
      IF(NCONF(I).EQ.1)FRAC3=(ZETA3-ZU(N3))/(ZETA1-ZU(N1))              SWIG0889
      IF(FRAC1.LT.0.)F1=ABS(FRAC1)/(1.+ABS(FRAC1))                      SWIG0890
      IF(FRAC2.LT.0.)F2=ABS(FRAC2)/(1.+ABS(FRAC2))                      SWIG0891
      IF(FRAC3.LT.0.)F3=ABS(FRAC3)/(1.+ABS(FRAC3))                      SWIG0892
      IF(FRAC1.LT.0.)WRITE(WR,1440)I,N1,N2,F1                           SWIG0893
      IF(FRAC2.LT.0.)WRITE(WR,1440)I,N2,N3,F2                           SWIG0894
      IF(FRAC3.LT.0.)WRITE(WR,1440)I,N3,N1,F3                           SWIG0895
  923 CONTINUE                                                          SWIG0896
  927 IF(NVEL.EQ.0)GO TO 990                                            SWIG0897
      WRITE(WR,1470)                                                    SWIG0898
      DO 930 I=1,NE                                                     SWIG0899
      N1=NODE(I,1)                                                      SWIG0900
      N2=NODE(I,2)                                                      SWIG0901
      N3=NODE(I,3)                                                      SWIG0902
      FACT=-1.000/(2.*AREA(I))                                          SWIG0903
      XVELF=FACT*(B(I,1)*HFNEW(N1)+B(I,2)*HFNEW(N2)+B(I,3)*HFNEW(N3))   SWIG0904
     1*KFX(I)                                                           SWIG0905
      YVELF=FACT*(A(I,1)*HFNEW(N1)+A(I,2)*HFNEW(N2)+A(I,3)*HFNEW(N3))   SWIG0906
     1*KFY(I)                                                           SWIG0907
      XVELS=FACT*(B(I,1)*HSNEW(N1)+B(I,2)*HSNEW(N2)+B(I,3)*HSNEW(N3))   SWIG0908
     1*KSX(I)                                                           SWIG0909
      YVELS=FACT*(A(I,1)*HSNEW(N1)+A(I,2)*HSNEW(N2)+A(I,3)*HSNEW(N3))   SWIG0910
     I*KSY(I)                                                           SWIG0911
      VELF=SQRT(XVELF**2+YVELF**2)                                      SWIG0912
      VELS=SQRT(XVELS**2+YVELS**2)                                      SWIG0913
      IF(XVELF.EQ.0.)GO TO 931                                          SWIG0914
      TANF=ATAN2(YVELF,XVELF)*57.3                                      SWIG0915
      GO TO 932                                                         SWIG0916
  931 IF(YVELF.GT.0.)TANF=90.                                           SWIG0917
      IF(YVELF.LT.0.)TANF=-90.                                          SWIG0918
      IF(YVELF.EQ.0.)TANF=0.                                            SWIG0919
  932 IF(XVELS.EQ.0.)GO TO 925                                          SWIG0920
      TANS=ATAN2(YVELS,XVELS)*57.3                                      SWIG0921
      GO TO 926                                                         SWIG0922
  925 IF(YVELS.GT.0.)TANS=90.                                           SWIG0923
      IF(YVELS.LT.0.)TANS=-90.                                          SWIG0924
      IF(YVELS.EQ.0.)TANS=0.                                            SWIG0925
  926 WRITE(WR,1480)I,XVELF,YVELF,VELF,TANF,XVELS,YVELS,VELS,TANS       SWIG0926
  930 CONTINUE                                                          SWIG0927
  990 IF(JT.GE.JMAX)GO TO 999                                           SWIG0928
```

```
      GO TO 200                                                         SWIG0929
999   STOP                                                              SWIG0930
1000  FORMAT(20A4)                                                      SWIG0931
1010  FORMAT('0',/,75X,'****',/,75X,'*',3X,'***',/,74X,'*',7X,'*',/,    SWIG0932
     173X,'*',9X,'*',/,72X,'*',11X,'*',/,71X,'*',13X,'*',/,70X,'*',     SWIG0933
     215X,'*',/,70X,'*',16X,'*',/,70X,'*',17X,'*',/,70X,'*',17X,        SWIG0934
     3'*',/,70X,'*',18X,'*',/,70X,'*',18X,'*  **',/,70X,'*',19X,'**  ***SWIG0935
     4******',/,70X,'*',31X,'*',/,70X,'*',30X,'*',/,69X,'*',31X,        SWIG0936
     5'*',/,69X,'*',31X,'*',/,68X,'*',31X,'*',/,67X,'*',31X,'*',/,      SWIG0937
     666X,'*',31X,'*',/,65X,'*',32X,'*',/,64X,'*',33X,'*',/,64X,'*',    SWIG0938
     733X,'*',/,63X,'*',33X,'*',/,63X,'*',32X,'*',/,62X,'*',33X,'*',/   SWIG0939
     8,61X,'*',34X,'*',/,61X,'*',33X,'*',/,62X,'*',32X,'*',/,62X,'*',31XSWIG0940
     9,'*',/,62X,'*',28X,'***',/,61X,'*',25X,'****',/,53X,'*********',  SWIG0941
     *24X,'**',/,53X,'*',                                               SWIG0942
     *   31X,'*',/,52X,'*',32X,'*',/,54X,'*',29X,'*',/,54X,'*',25X,'**',SWIG0943
     1/,53X,'*',25X,'*',/,53X,'*',25X,'*',/,42X,'*',9X,'*',  25X,'*',   SWIG0944
     2/,41X,'* *********',24X,'*',/,42X,'*',33X,'*',/,43X,'*',31X,      SWIG0945
     3'*',/,44X,'*',23X,'*',/,45X,'*',24X,'***',/,46X,'*',23X,'*',/,    SWIG0946
     448X,'*',20X,'*',/,50X,'*',17X,'*',/,52X,'*',12X,'***',/,54X,'*'   SWIG0947
     5,9X,'*',/,56X,'***********')                                      SWIG0948
1020  FORMAT(///,48X,'NORTHERN GUAM LENS STUDY',//,27X,' A TWO-DIMENSIONSWIG0949
     UAL MODEL OF SALT WATER INTRUSION INTO GROUNDWATER SYSTEMS',//,33X,SWIG0950
     2' WATER AND ENERGY RESEARCH INSTITUTE OF THE WESTERN PACIFIC',//, SWIG0951
     350X,' UNIVERSITY OF GUAM',//,40X,' COLLEGE STATION, MANGILAO, GUAMSWIG0952
     3, 96913',///,25X,20A4)                                            SWIG0953
1030  FORMAT(16I5)                                                      SWIG0954
1040  FORMAT('0INPUT DATA',     //,5X,' NUMBER OF NODES, NN =',I5,/,5X,SWIG0955
     1' NUMBER OF ELEMENTS, NE =',I5,/,                                 SWIG0956
     2                                       5X,' MAX. NO. OF ITERASWIG0957
     3TIONS, ITERMX = ',I5,/,5X,' TOTAL NO. OF PUMPS, NPUMPS =',I5,/,5X,SWIG0958
     4' INPUT DATA TO BE PRINTED OUT ? , NDATA =',I5,/,5X,' RESULTS TO BSWIG0959
     5E PRINTED OUT EVERY',I3,2X,'TIME STEPS',/,5X,' MAX. NO. OF TIME STSWIG0960
     6EPS,JMAX =',I5,/,5X,' DETAILED MATRIX OUTPUT TO BE PRINTED AT JT=JSWIG0961
     7TRC =',I5,/,  5X,' OUTPUT OF VELOCITIES IN ELEMENTS ?  NVEL=',I5, SWIG0962
     8/,5X,' OUTPUT FOR LOCATION OF TOE(S) ?  , NTOE =',I5)             SWIG0963
1045  FORMAT(E10.3,7F10.4)                                              SWIG0964
1050  FORMAT(8F10.4)                                                    SWIG0965
1060  FORMAT('0    TIME INTERVAL DT =',E15.7,/,5X,' WEIGHTING FACTOR THSWIG0966
     1ETA =',F5.3,/,5X,' TOLERANCE (AS A FRACTION) USED IN CONVERGENCE, SWIG0967
     2 TOL =',F6.4,/,5X,' SPECIFIC WEIGHT OF FRESH WATER, GAMAF =',F7.3,SWIG0968
     3/,5X,' SPECIFIC WEIGHT OF SALT WATER, GAMAS =',F7.3,/,5X,' STARTINSWIG0969
     4G TIME, TSTART =',E15.7,/,5X,' THICKNESS OF FW OR SW LAYER BEYOND SWIG0970
     5TOE, BTOE =',F10.5)                                               SWIG0971
1065  FORMAT('0NO. OF ELEMENTS THAT HAVE FRESHWATER FLOW SPECIFIED ALONGSWIG0972
     1ONE SIDE, LMNTQF =',I5,/,' NO. OF ELEMENTS THAT HAVE SALTWATER FLOSWIG0973
     2W SPECIFIED ALONG ONE SIDE, LMNTQS =',I5,/,' NO. OF ELEMENTS THAT SWIG0974
     3HAVE A COASTAL BOUNDARY CONDITION SPECIFIED ALONG ONE SIDE, LMNTCFSWIG0975
     4 =',I5,/,' NO. OF NODES THAT HAVE FRESHWATER HEAD SPECIFIED, NNODHSWIG0976
     5F=',I5,/,' NO. OF NODES THAT HAVE SALTWATER HEAD SPECIFIED, NNODHSSWIG0977
     6=',I5)                                                            SWIG0978
```

```
1070  FORMAT(10X,4F10.4)
1080  FORMAT('ONODAL DATA',//,' NODE       X CO-ORD. Y CO-ORD. BOTTOM ELE
     1V.  TOP ELEV. OF AQUIFER',/,(I5,5X,2F10.0,2F10.2))
1090  FORMAT(5X,3I5,5F10.0,I1,I2,2I3)
1100  FORMAT('O ELEMENT DATA',//,' ELEM. NO. NODE(I,1) NODE(I,2) NODE(I,
     13) F.W.PERM. X   F.W.PERM. Y   S.W.PERM. X   S.W.PERM. Y      POR
     2OSITY  NCONF(I) NLEAKY(I)',/,(4(I5,5X),4(F12.7,2X),5X,F10.8,2I5))
1105  FORMAT('ORECHARGE TO UNCONFINED AQUIFER AS A FUNCTION OF TIME')
1106  FORMAT('OELEMENT I =',I5,5X,'NO. OF SUCCEEDING ELEMENTS WITH SAME
     1RECHARGE, INC =',I5,/,(8(2X,I2,1X,F11.9)))


1110  FORMAT(3I5,/,8(F10.4))
1114  FORMAT('OBOUNDARIES ALONG WHICH F.W. FLOW IS SPECIFIED.')
1115  FORMAT(             /,'             I =',I5,5X,' NODEQF(I,1)=',I5,5X,
     1'NODEQF(I,2)=',I5, /,(8(2X,I2,1X,F11.7)))
1124  FORMAT('OBOUNDARIES ALONG WHICH S.W. FLOW IS SPECIFIED.')
1125  FORMAT(             /,'             I =',I5,5X,'NODEQS(I,1) =',I5,5X,
     1'NODEQS(I,2) ='I5, /,(8(2X,I2,1X,F11.9)))
1130  FORMAT(3I5,5X,F15.13)
1127  FORMAT('OCOASTAL BOUNDARY ALONG WHICH MIXED B.C. IS SPECIFIED')
1126  FORMAT(                       /,'         NODECF(I,1) NODECF(I,2
     1) PERMEABILITY KC',/,(3(I5,5X),F12.10))
1140  FORMAT(2I5,/,(8F10.4))
1134  FORMAT('ONODES AT WHICH FRESHWATER HEAD IS SPECIFIED')
1135  FORMAT(//,I5,5X,'NODEHF(I) =',I5,/,(8(2X,I2,1X,F11.4)))
1144  FORMAT('ONODES AT WHICH SALTWATER HEAD IS SPECIFIED')
1145  FORMAT(//,I5,5X,'NODEHS(I) =',I5,/,(8(2X,I2,1X,F11.4)))
1160  FORMAT('OPUMP DATA ',/,'OTOTAL NUMBER OF PUMPS IN AQUIFER =',I5)
1150  FORMAT('0',I5,' NODE AT WHICH PUMP IS LOCATED, NODEP(I) =',I5,'
     1NUMBER OF ELEMENTS AROUND NODE, NELEM(I) =',I5)
1165  FORMAT(/,8(2X,I2,1X,F11.3))
1190  FORMAT(I5,5X,2F10.4,I5)
1200  FORMAT('ODATA ON LEAKY AQUITARD ABOVE AQUIFER',//,' ELEMENT   KC(I)
     1        PHIO(I)',/)
1205  FORMAT(' ', I5,10X,F12.10,F10.2)
1180  FORMAT(I5,5X,F10.4,I5)
1170  FORMAT('OINITIAL CONDITIONS',/,'   NODE     HF(I,1)    HS(I,1)    BF
     1(I,1)    BS(I,1)  ')
1175  FORMAT(' ',I5,4F10.3)
1220  FORMAT(2I5)
1225  FORMAT(80I1)
1230  FORMAT('OELEMENT TYPE AT TIME T = TSTART ',/,' ELEMENT  LMNTYP(I)'
     1,/,(' ',10(2I5)))
1240  FORMAT('OAREAS OF ELEMENTS',/,4X,'I',10X,'A(I,1)',9X,'A(I,2)',9X,'
     1A(I,3)',9X,'B(I,1)',9X,'B(I,2)',9X,'B(I,3)',9X,'AREA(I)',/,
     2(I5,5X,7E15.7))
1250  FORMAT('ODISTANCE BETWEEN FRESHWATER FLOW NODES      ',//,'     I
     1NODEQF(I,1)  NODEQF(I,2)  ELQF(I)'   ,/,(3(I5,5X),E15.7))
1260  FORMAT('ODISTANCE BETWEEN SALTWATER FLOW NODES      ',//,'     I
```

```
     1NODEQS(I,1)  NODEQS(I,2)  ELQS(I)'     ,/,(3(I5,5X),E15.7))      SWIG1029
1270 FORMAT('ODISTANCE BETWEEN COASTAL BOUNDARY NODES               ', SWIG1030
     1/,'    I        NODECF(I,1)  NODECF(I,2)  ELCF(I)',/,(3(I5,5X),E15.7))SWIG1031
1280 FORMAT('OELEMENTS OF MATRIX KFIJ',/,4X,'I',10X,'I,1,1',7X,'I,1,2',SWIG1032
     17X,'I,1,3',7X,'I,2,1',7X,'I,2,2',7X,'I,2,3',7X,'I,3,1',7X,'I,3,2',SWIG1033
     27X,'I,3,3',/,(I5,5X,9(E12.5)))                                   SWIG1034
1290 FORMAT('OELEMENTS OF MATRIX KSIJ',/,4X,'I',10X,'I,1,1',7X,'I,1,2',SWIG1035
     17X,'I,1,3',7X,'I,2,1',7X,'I,2,2',7X,'I,2,3',7X,'I,3,1',7X,'I,3,2',SWIG1036
     27X,'I,3,3',/,(I5,5X,9(E12.5)))                                   SWIG1037
1300 FORMAT('OELEMENTS OF MATRIX KOIJ',/,4X,'I',10X,'I,1,1',7X,'I,1,2',SWIG1038
     27X,'I,1,3',7X,'I,2,1',7X,'I,2,2',7X,'I,2,3',7X,'I,3,1',7X,'I,3,2',SWIG1039
     37X,'I,3,3',/)                                                    SWIG1040
1305 FORMAT(' ',I5,5X,9(E12.5))                                        SWIG1041
1310 FORMAT('OELEMENTS OF MATRIX KIJF',/,4X,'I',10X,'I,1,1',7X,'I,1,2',SWIG1042
     17X,'I,1,3',7X,'I,2,1',7X,'I,2,2',7X,'I,2,3',7X,'I,3,1',7X,'I,3,2',SWIG1043
     27X,'I,3,3',/,(I5,5X,9(E12.5)))                                   SWIG1044
1320 FORMAT('OELEMENTS OF MATRIX KIJS',/,4X,'I',10X,'I,1,1',7X,'I,1,2',SWIG1045
     27X,'I,1,3',7X,'I,2,1',7X,'I,2,2',7X,'I,2,3',7X,'I,3,1',7X,'I,3,2',SWIG1046
     37X,'I,3,3',/,(I5,5X,9(E12.5)))                                   SWIG1047
1330 FORMAT('ODETAILED OUTPUT FOR ELEMENT I =',I5,'    N1=',I3,2X,'N2=',SWIG1048
     1I3,2X,'N3=',I3,2X,'BFE=',E15.7,5X,'BSE=',E15.7)                  SWIG1049
1340 FORMAT(20X,6(I10,3X))                                             SWIG1050
1350 FORMAT(10X,I5,5X,7(E11.4,2X))                                     SWIG1051
1360 FORMAT(' ',I3,2X,120I1,2X,I3)                                     SWIG1052
1370 FORMAT(' MATRIX DIMENSIONS, NN2S=',   I5,/,' NO. OF LOWER CO-DIAGOSWIG1053
     1NALS, MINBW=',I5,/,' NO. OF UPPER CO-DIAGONALS, MAXBW =',I5,/,' TOSWIG1054
     2TAL NO. OF ELEMENTS IN BANDED MATRIX, IC =',I10)                 SWIG1055
1380 FORMAT(' TIME COUNTER, JT =',I3,5X,' TIME T =',E15.7,5X,'ITERATIONSWIG1056
     1 NO., ITER =',I5,/,' ERROR =',E15.7,5X,'BASE =',E15.7)           SWIG1057
1390 FORMAT(I5,5X,5(F10.3,5X))                                         SWIG1058
1400 FORMAT(' NODE',5X,'HF(I,2)',8X,'HS(I,2)',8X,'BF(I,2)',8X,'BS(I,2)'SWIG1059
     1,8X,'ZETA(I)')                                                   SWIG1060
1420 FORMAT(' SALTWATER TOE OCCURS IN ELEMENT I =',I4,5X,'FRACTIONAL DISWIG1061
     1STANCE FROM NODE',I4,'  TO NODE',I4,'  IS ',F6.4)                SWIG1062
1430 FORMAT(' ERROR IN SUBROUTINE GELB, IER =',I5)                     SWIG1063
1440 FORMAT(' FRESH WATER TOE OCCURS IN ELEMENT I =',I5,5X,'FRACTIONAL SWIG1064
     1DISTANCE FROM NODE',I4,'  TO NODE',I4,'  IS ',F6.4)              SWIG1065
1450 FORMAT((' ',I5,I4,10(1X,E11.4)))                                  SWIG1066
1460 FORMAT((' ',I5,4X,10(1X,E11.4)))                                  SWIG1067
1470 FORMAT('ODIRECTION AND MAGNITUDE OF DISCHARGE VELOCITY OF FRESHWATSWIG1068
     1ER AND SALTWATER IN ELEMENTS.  ',/,4X,'I',5X,'XVELF',5X,'YVELF',5XSWIG1069
     1,'VELF',6X,'TANF',9X,'XVELS',5X,'YVELS',5X,'VELS',6X,'TANS')     SWIG1070
1480 FORMAT(' ',I4,5X,4E10.3,3X,4E10.3)                                SWIG1071
1490 FORMAT((' ',I5,10(9X,I3)))                                        SWIG1072
1500 FORMAT(' ',4X,3(I5,5X),F10.4)                                     SWIG1073
1510 FORMAT('OAREA OF ELEMENT I = ',I3,' IS EQUAL TO  ',E15.7)         SWIG1074
1520 FORMAT('OAREA OF AQUIFER =',E15.7,/,' VOLUME OF FRESH WATER IN AQUSWIG1075
     1IFER =',E15.7,/,' AVERAGE DEPTH OF FRESH WATER LENS IN AQUIFER =',SWIG1076
     2E15.7)                                                           SWIG1077
     END                                                               SWIG1078
```

```
C        SUBROUTINE GELB
C        PURPOSE
C           TO SOLVE A SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS WITH A
C        COEFFICIENT MATRIX OF BAND STRUCTURE.
C
C        USAGE
C        CALL GELB(R,A,M,N,MUD,MLD,EPS,IER)
C
C        DESCRIPTION OF PARAMETERS
C        R         - M BY N RIGHT HAND SIDE MATRIX (DESTROYED).
C                    ON RETURN, R CONTAINS THE SOLUTION OF THE EQUATIONS.
C        A         - M BY M COEFFICIENT MATRIX WITH BAND STRUCTURE
C                    (DESTROYED).
C        M         - THE NUMBER OF EQUATIONS IN THE SYSTEM.
C        N         - THE NUMBER OF RIGHT HAND SIDE VECTORS.
C        MUD       - THE NUMBER OF UPPER CODIAGONALS (THAT MEANS
C                    CODIAGONALS ABOVE THE MAIN DIAGONAL).
C        MLD       - THE NUMBER OF LOWER CODIAGONALS (THAT MEANS
C                    CODIAGONALS BELOW MAIN DIAGONAL).
C        EPS       - AN INPUT CONSTANT WHICH IS USED AS RELATIVE
C                    TOLERANCE FOR TEST ON LOSS OF SIGNIFICANCE.
C        IER       - RESULTING ERROR PARAMETER CODED AS FOLLOWS
C                    IER=0   - NO ERROR,
C                    IER=-1  - NO RESULT BECAUSE OF WRONG INPUT PARAMETERS
C                              M,MUD,MLD, OR BECAUSE OF PIVOT ELEMENT
C                              AT ANY ELIMINATION STEP EQUAL TO ZERO,
C                    IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFICANCE
C                             INDICATED AT ELIMINATION STEP K+1,
C                             WHERE PIVOT ELEMENT WAS LESS THAN OR
C                             EQUAL TO INTERNAL TOLERANCE EPS TIMES
C                             ABSOLUTELY GREATEST ELEMENT OF MATRIX A.
C
C        REMARKS
C           BAND MATRIX A IS ASSUMED TO BE STORED ROWWISE IN THE FIRST
C            ME SUCCESSIVE STORAGE LOCATIONS OF TOTALLY NEEDED MA
C             STORAGE LOCATIONS, WHERE
C                MA=M*MC-ML*(ML+1)/2    AND    ME=MA-MU*(MU+1)/2   WITH
C               MC=MIN(M,1+MUD+MLD),   ML=MC-1-MLD,   MU=MC-1-MUD.
C               RIGHT HAND SIDE MATRIX R ASSUMED TO BE STORED COLUMNWISE
C               IN N*M SUCCESSIVE STORAGE LOCATIONS.  ON RETURN, SOLUTION
C               MATRIX IS STORED COLUMNWISE TOO.
C               INPUT PARAMETERS M,MUD,MLD SHOULD SATISFY THE FOLLOWING
C               RESTRICTIONS.     MUD NOT LESS THAN ZERO
C                      MLD NOT LESS THAN ZERO
C                     MUD+MLD NOT GREATER THAN 2*M-2.
C            NO ACTION BESIDES ERROR MESSAGE IER=-1 TAKES PLACE IF THESE
C            RESTRICTIONS ARE NOT SATISFIED.
C            THE PROCEDURE GIVES RESULTS IF THE RESTICTIONS ON INPUT
C            PARAMETERS ARE SATISFIED AND IF PIVOTELEMENTS AT ALL
C            ELIMINATION STEPS ARE DIFFERENT FROM 0.  HOWEVER, WARNING
```

```
C          IER=K - IF GIVEN - INDICATES POSSIBLELOSS OF SIGNIFICANCE.     GEL00530
C          IN CASE OF A WELL SCALED MATRIX A AND APPROPRIATE TOLERANCE    GEL00540
C          EPS, IER=K MAY BE INTERPRETED THAT MATRIX A HAS THE RANK K.    GEL00550
C          NO WARNING IS GIVEN IF MATRIX A HAS NO LOWER CODIAGONAL.       GEL00560
C                                                                         GEL00570
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     GEL00580
C       NONE                                                              GEL00590
C                                                                         GEL00600
C       METHOD                                                            GEL00610
C          SOLUTION IS DONE BY MEANS OF GAUSS ELIMINATION WITH            GEL00620
C          COLUMN PIVOTING ONLY, IN ORDER TO PRESERVE BAND STRUCTURE      GEL00630
C          IN REMAINING COEFFICIENT MATRICES.                             GEL00640
C                                                                         GEL00650
C       ..................................................................GEL00660
C                                                                         GEL00670
        SUBROUTINE GELB(     M,N,MUD,MLD,MA,EPS,IER)                      GEL00680
C                                                                         GEL00690
C                                                                         GEL00700
        DIMENSION R(380),A(25000)                                         GEL00710
        COMMON R,A                                                        GEL00720
C       TEST ON WRONG INPUT PARAMETERS                                    GEL00730
        IF(MLD)47,1,1                                                     GEL00740
  1     IF(MUD)47,2,2                                                     GEL00750
  2     MC=1+MLD+MUD                                                      GEL00760
        IF(MC+1-M-M)3,3,47                                                GEL00770
C                                                                         GEL00780
C       PREPARE INTEGER PARAMETERS                                        GEL00790
C          MC=NUMBER OF COLUMNS IN MATRIX A                               GEL00800
C          MU=NUMBER OF ZEROS TO BE INSERTED IN FIRST ROW OF MATRIX A     GEL00810
C          ML=NUMBER OF MISSING ELEMENTS IN LAST ROW OF MATRIX A          GEL00820
C          MR=INDEX OF LAST ROW IN MATRIX A WITH MC ELEMENTS              GEL00830
C          MZ=TOTAL NUMBER OF ZEROS TO BE INSERTED IN MATRIX A            GEL00840
C          MA=TOTAL NUMBER OF STORAGE LOCATIONS NECESSARY FOR MATRIX A    GEL00850
C          NM=NUMBER OF ELEMENTS IN MATRIX A                              GEL00860
  3     IF(MC-M)5,5,4                                                     GEL00870
  4     MC=M                                                              GEL00880
  5     MU=MC-MUD-1                                                       GEL00890
        ML=MC-MLD-1                                                       GEL00900
        MR=M-ML                                                           GEL00910
        MZ=(MU*(MU+1))/2                                                  GEL00920
        MA=M*MC-(ML*(ML+1))/2                                             GEL00930
        NM=N*M                                                            GEL00940
C       DIMENSION S(M),Z(MA)                                              GEL00950
C       MOVE ELEMENTS BACKWARD AND SEARCH FOR ABSOLUTELY GREATEST ELEMENT GEL00960
C       (NOT NECESSARY IN CASE OF A MATRIX WITHOUT LOWER CODIAGONALS)     GEL00970
        IER=0                                                             GEL00980
        PIV=0.                                                            GEL00990
        IF(MLD)14,14,6                                                    GEL01000
  6     JJ=MA                                                             GEL01010
        J=MA-MZ                                                           GEL01020
```

```
      KST=J
      DO 9 K=1,KST
      TB=A(J)
      A(JJ)=TB
      TB= ABS(TB)
      IF(TB-PIV)8,8,7
 7    PIV=TB
 8    J=J-1
 9    JJ=JJ-1
C
C     INSERT ZEROS IN FIRST MU ROWS (NOT NECESSARY IN CASE MZ=0)
      IF(MZ)14,14,10
 10   JJ=1
      J=MZ+1
      IC=1+MUD
      DO 13 I=1,MU
      DO 12 K=1,MC
      A(JJ)=0.
      IF(K-IC)11,11,12
 11   A(JJ)=A(J)
      J=J+1
 12   JJ=JJ+1
 13   IC=IC+1
C
C     GENERATE TEST VALUE FOR SINGULARITY
 14   TOL=EPS*PIV
C
C     START DECOMPOSITION LOOP
      KST=1
      IDST=MC
      IC=MC-1
      DO 38 K=1,M
      IF(K-MR-1)16,16,15
 15   IDST=IDST-1
 16   ID=IDST
      ILR=K+MLD
      IF(ILR-M)18,18,17
 17   ILR=M
 18   II=KST
C
C     PIVOT  SEARCH IN FIRST COLUMN(ROW INDEXES FROM I=K TO I=ILR)
      PIV=0.
      DO 22 I=K,ILR
      TB= ABS(A(II))
      IF(TB-PIV)20,20,19
 19   PIV=TB
      J=I
      JJ=II
 20   IF(I-MR)22,22,21
 21   ID=ID-1
```

```
 22     II=II+ID                                                        GEL01540
C                                                                       GEL01550
C       TEST ON SINGULARITY                                             GEL01560
        IF(PIV)47,47,23                                                 GEL01570
 23     IF(IER)26,24,26                                                 GEL01580
 24     IF(PIV-TOL)25,25,26                                             GEL01590
 25     IER=K-1                                                         GEL01600
 26     PIV=1./A(JJ)                                                    GEL01610
C                                                                       GEL01620
C       PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R    GEL01630
        ID=J-K                                                          GEL01640
        DO 27 I=K,NM,M                                                  GEL01650
        II=I+ID                                                         GEL01660
        TB=PIV*R(II)                                                    GEL01670
        R(II)=R(I)                                                      GEL01680
 27     R(I)=TB                                                         GEL01690
C                                                                       GEL01700
C       PIVOT TOW REDUCTION AND ROW INTERCHANGE IN COEFFICIENT MATRIX A GEL01710
        II=KST                                                          GEL01720
        J=JJ+IC                                                         GEL01730
        DO 28 I=JJ,J                                                    GEL01740
        TB=PIV*A(I)                                                     GEL01750
        A(I)=A(II)                                                      GEL01760
        A(II)=TB                                                        GEL01770
 28     II=II+1                                                         GEL01780
C                                                                       GEL01790
C       ELEMENT REDUCTION                                               GEL01800
        IF(K-ILR)29,34,34                                               GEL01810
 29     ID=KST                                                          GEL01820
        II=K+1                                                          GEL01830
        MU=KST+1                                                        GEL01840
        MZ=KST+IC                                                       GEL01850
        DO 33 I=II,ILR                                                  GEL01860
C                                                                       GEL01870
C       IN MATRIX A                                                     GEL01880
        ID=ID+MC                                                        GEL01890
        JJ=I-MR-1                                                       GEL01900
        IF(JJ)31,31,30                                                  GEL01910
 30     ID=ID-JJ                                                        GEL01920
 31     PIV=-A(ID)                                                      GEL01930
        J=ID+1                                                          GEL01940
        DO 32 JJ=MU,MZ                                                  GEL01950
        A(J-1)=A(J)+PIV*A(JJ)                                           GEL01960
 32     J=J+1                                                           GEL01980
        A(J-1)=0.                                                       GEL01990
C                                                                       GEL02000
C       IN MATRIX R                                                     GEL02010
        J=K                                                             GEL02020
        DO 33 JJ=I,NM,M                                                 GEL02030
        R(JJ)=R(JJ)+PIV*R(J)                                            GEL02040
```

```
 33     J=J+M                                                           GEL02050
 34     KST=KST+MC                                                      GEL02060
        IF(ILR-MR)36,35,35                                              GEL02070
 35     IC=IC-1                                                         GEL02080
 36     ID=K-MR                                                         GEL02090
        IF(ID)38,38,37                                                  GEL02100
 37     KST=KST-ID                                                      GEL02110
 38     CONTINUE                                                        GEL02120
C       END OF DECOMPOSITION LOOP                                       GEL02130
C                                                                       GEL02140
C                                                                       GEL02140
C       BACK SUBSTITUTION                                               GEL02150
        IF(MC-1)46,46,39                                                GEL02160
 39     IC=2                                                            GEL02170
        KST=MA+ML-MC+2                                                  GEL02180
        II=M                                                            GEL02190
        DO 45 I=2,M                                                     GEL02200
        KST=KST-MC                                                      GEL0220
        II=II-1                                                         GEL02220
        J=II-MR                                                         GEL02230
        IF(J)41,41,40                                                   GEL02240
 40     KST=KST+J                                                       GEL02250
 41     DO 43 J=II,NM,M                                                 GEL02260
        TB=R(J)                                                         GEL02270
        MZ=KST+IC-2                                                     GEL02280
        ID=J                                                            GEL02300
        DO 42 JJ=KST,MZ                                                 GEL02310
        ID=ID+1                                                         GEL02320
 42     TB=TB-A(JJ)*R(ID)                                               GEL02330
 43     R(J)=TB                                                         GEL02340
        IF(IC-MC)44,45,45                                               GEL02350
 44     IC=IC+1                                                         GEL02360
 45     CONTINUE                                                        GEL02370
 46     RETURN                                                          GEL02380
C                                                                       GEL02390
C                                                                       GEL02390
C       ERROR RETURN                                                    GEL02400
 47     IER=-1                                                          GEL02410
        RETURN                                                          GEL02420
        END                                                             GEL02430
```

APPENDIX D.

SAMPLE LISTING AND OUTPUT.

```
 STEADY STATE SOLUTION OF SALT WATER INTRUSION INTO A PHREATIC AQUIFER.
    32    40    10     0     1     4     4     0     1     0
+0.180E+24 1.0      .005       62.4       64.0      0.          0.1
     0     0     1     0     2
1          -22000.  0.         -600.      1000.1
2          -22000.  500.       -600.      1000.1
3          -20000.  250.       -600.      1000.1
4          -17100.  0.         -600.      1000.1
5          -17100.  500.       -600.      1000.1
6          -15300.  250.       -600.      1000.1
7          -13500.  0.         -600.      1000.1
8          -13500.  500.       -600.      1000.1
9          -12000.  250.       -600.      1000.1
10         -10500.  0.         -600.      1000.1
11         -10500.  500.       -600.      1000.1
12         -9100.   250.       -600.      1000.1
13         -7800.   0.         -600.      1000.1
14         -7800.   500.       -600.      1000.1
15         -6600.   250.       -600.      1000.1
16         -5500.   0.         -600.      1000.1
17         -5500.   500.       -600.      1000.1
18         -4500.   250.       -600.      1000.1
19         -3500.   0.         -600.      1000.1
20         -3500.   500.       -600.      1000.1
21         -2800.   250.       -600.      1000.1
22         -2100.   0.         -600.      1000.1
23         -2100.   500.       -600.      1000.1
24         -1500.   250.       -600.      1000.1
25         -1000.   0.         -600.      1000.1
26         -1000.   500.       -600.      1000.1
27         -600.    250.       -600.      1000.1
28         -300.    0.         -600.      1000.1
29         -300.    500.       -600.      1000.1
30         -100.    250.       -600.      1000.1
31         0.       0.         -600.      1000.1
32         0.       500.       -600.      1000.1
1        1     4     3 .01      .01       0.01025    .01025    0.3       0 0    9   3
11       1     3     2 .01      .01       0.01025    .01025    0.3       0 0    9   3
21       3     4     5 .01      .01       0.01025    .01025    0.3       0 0    9   3
31       3     5     2 .01      .01       0.01025    .01025    0.3       0 0    9   3
     1    39
.0000001  .0000001  .0000001  .0000001  .0000001  .0000001  .0000001  .0000001
     1    31    32       .01
     1    31
```

```
 2    32

  1     20.00       1
  3     18.53       0
  4     16.41       1
  6     15.09       0
  7     13.77       1
  9     12.66       0
 10     11.56       1
 12     10.53       0
 13     9.58        1
 15     8.70        0
 16     7.89        1
 18     7.09        0
 19     6.28        1
 21     5.58        0
 22     4.87        1
 24     4.04        0
 25     3.24        1
 27     2.55        0
 28     1.80        1
 30     1.04        0
 31     0.20        1
  1     0.0        31
  1     299.9      16
 18     276.5       0
 19     245.6       1
 21     217.7       0
 22     189.76      1
 24     158.26      0
 25     130.9       1
 27     99.         0
 28     73.         1
 30     42.         0
 31     8.57        1
 01     0.10       16
 18     23.5        0
 19     55.0        1
 21     82.4        0
 22     110.3       1
 24     141.        0
 25     169.        1
 27     201.        0
 28     226.        1
 30     258.        0
 31     289.3       1
2222222222222222
1111111111111111111111111111111
/*
/&
```

APPENDIX D

CONTINUED

STEADY STATE SOLUTION OF SALTWATER INTRUSION INTO A PHREATIC AQUIFER

NORTHERN GUAM LENS STUDY

A TWO-DIMENSIONAL MODEL OF SALTWATER INTRUSION INTO GROUNDWATER SYSTEMS

WATER AND ENERGY RESEARCH INSTITUTE OF THE WESTERN PACIFIC
UNIVERSITY OF GUAM
UNIVERSITY STATION, MANGILAO, GUAM 96913

STEADY STATE SOLUTION OF SALT WATER INTRUSION INTO A PHREATIC AQUIFER.

INPUT DATA

```
     NUMBER OF NODES, NN =    32
     NUMBER OF ELEMENTS, NE =    40
     MAX. NO. OF ITERATIONS, ITERMX =     10
     TOTAL NO. OF PUMPS, NPUMPS =      0
     INPUT DATA TO BE PRINTED OUT ? , NDATA =     1
     RESULTS TO BE PRINTED OUT EVERY  4  TIME STEPS
     MAX. NO. OF TIME STEPS,JMAX =     4
     DETAILED MATRIX OUTPUT TO BE PRINTED AT JT=JTRC =     0
     OUTPUT OF VELOCITIES IN ELEMENTS ?  NVEL=     1
     OUTPUT FOR LOCATION OF TOE(S) ?  , NTOE =     0

     TIME INTERVAL DT =  0.1799999E 24
     WEIGHTING FACTOR THETA =1.000
     TOLERANCE (AS A FRACTION) USED IN CONVERGENCE,  TOL =0.0050
     SPECIFIC WEIGHT OF FRESH WATER, GAMAF = 62.400
     SPECIFIC WEIGHT OF SALT WATER, GAMAS = 64.000
     STARTING TIME, TSTART =  0.0
     THICKNESS OF FW OR SW LAYER BEYOND TOE, BTOE =    0.10000

NO. OF ELEMENTS THAT HAVE FRESHWATER FLOW SPECIFIED ALONGONE SIDE, LMNTQF =     0
NO. OF ELEMENTS THAT HAVE SALTWATER FLOW SPECIFIED ALONG ONE SIDE, LMNTQS =     0
NO. OF ELEMENTS THAT HAVE A COASTAL BOUNDARY CONDITION SPECIFIED ALONG ONE SIDE, LMNTCF =     1
NO. OF NODES THAT HAVE FRESHWATER HEAD SPECIFIED, NNODHF=     0
NO. OF NODES THAT HAVE SALTWATER HEAD SPECIFIED, NNODHS=     2
```

NODAL DATA

NODE	X CO-ORD.	Y CO-ORD.	BOTTOM ELEV.	TOP ELEV. OF AQUIFER
1	-22000.	0.	-600.00	1000.10
2	-22000.	500.	-600.00	1000.10
3	-20000.	250.	-600.00	1000.10
4	-17100.	0.	-600.00	1000.10
5	-17100.	500.	-600.00	1000.10
6	-15300.	250.	-600.00	1000.10
7	-13500.	0.	-600.00	1000.10
8	-13500.	500.	-600.00	1000.10
9	-12000.	250.	-600.00	1000.10
10	-10500.	0.	-600.00	1000.10
11	-10500.	500.	-600.00	1000.10
12	-9100.	250.	-600.00	1000.10
13	-7800.	0.	-600.00	1000.10
14	-7800.	500.	-600.00	1000.10
15	-6600.	250.	-600.00	1000.10
16	-5500.	0.	-600.00	1000.10
17	-5500.	500.	-600.00	1000.10
18	-4500.	250.	-600.00	1000.10
19	-3500.	0.	-600.00	1000.10
20	-3500.	500.	-600.00	1000.10
21	-2800.	250.	-600.00	1000.10
22	-2100.	0.	-600.00	1000.10
23	-2100.	500.	-600.00	1000.10
24	-1500.	250.	-600.00	1000.10
25	-1000.	0.	-600.00	1000.10
26	-1000.	500.	-600.00	1000.10
27	-600.	250.	-600.00	1000.10
28	-300.	0.	-600.00	1000.10
29	-300.	500.	-600.00	1000.10
30	-100.	250.	-600.00	1000.10
31	0.	0.	-600.00	1000.10
32	0.	500.	-600.00	1000.10

AREA OF AQUIFER = 0.1100000E 08
VOLUME OF FRESH WATER IN AQUIFER = 0.1139740E 10
AVERAGE DEPTH OF FRESH WATER LENS IN AQUIFER = 0.3453765E 03

DIRECTION AND MAGNITUDE OF DISCHARGE VELOCITY OF FRESHWATER AND SALTWATER IN ELEMENTS.

I	XVELF	YVELF	VELF	TANF	XVELS	YVELS	VELS	TANS
1	0.560E-06	-0.381E-07	0.562E-06	-0.389E 01	0.0	0.0	0.0	0.0
2	0.160E-05	-0.440E-07	0.160E-05	-0.157E 01	0.0	0.0	0.0	0.0
3	0.256E-05	-0.529E-07	0.256E-05	-0.119E 01	0.0	0.0	0.0	0.0
4	0.362E-05	-0.693E-07	0.362E-05	-0.110E 01	0.0	0.0	0.0	0.0
5	0.490E-05	-0.100E-06	0.490E-05	-0.117E 01	0.0	0.0	0.0	0.0
6	0.662E-05	-0.163E-06	0.662E-05	-0.141E 01	0.0	0.0	0.0	0.0
7	0.902E-05	-0.295E-06	0.903E-05	-0.187E 01	0.0	0.0	0.0	0.0
8	0.128E-04	-0.658E-06	0.128E-04	-0.294E 01	0.0	0.0	0.0	0.0
9	0.209E-04	-0.201E-05	0.210E-04	-0.550E 01	0.0	0.0	0.0	0.0
10	0.535E-04	-0.129E-04	0.550E-04	-0.135E 02	0.0	0.0	0.0	0.0
11	0.555E-06	0.664E-09	0.555E-06	0.685E-01	0.0	0.0	0.0	0.0
12	0.159E-05	0.174E-09	0.159E-05	0.624E-02	0.0	0.0	0.0	0.0
13	0.255E-05	0.521E-10	0.255E-05	0.117E-02	0.0	0.0	0.0	0.0
14	0.360E-05	0.558E-10	0.360E-05	0.888E-03	0.0	0.0	0.0	0.0
15	0.488E-05	0.0	0.488E-05	0.0	0.0	0.0	0.0	0.0
16	0.658E-05	0.0	0.658E-05	0.0	0.0	0.0	0.0	0.0
17	0.892E-05	0.0	0.892E-05	0.0	0.0	0.0	0.0	0.0
18	0.125E-04	-0.163E-10	0.125E-04	-0.743E-04	0.0	0.0	0.0	0.0
19	0.196E-04	-0.366E-10	0.196E-04	-0.107E-03	0.0	0.0	0.0	0.0
20	0.374E-04	0.0	0.374E-04	0.0	0.0	0.0	0.0	0.0
21	0.564E-06	0.162E-09	0.564E-06	0.164E-01	0.0	0.0	0.0	0.0
22	0.161E-05	0.868E-10	0.161E-05	0.310E-02	0.0	0.0	0.0	0.0
23	0.257E-05	0.521E-10	0.257E-05	0.116E-02	0.0	0.0	0.0	0.0
24	0.363E-05	0.0	0.363E-05	0.0	0.0	0.0	0.0	0.0
25	0.492E-05	-0.710E-10	0.492E-05	-0.827E-03	0.0	0.0	0.0	0.0
26	0.666E-05	0.0	0.666E-05	0.0	0.0	0.0	0.0	0.0
27	0.913E-05	-0.209E-10	0.913E-05	-0.131E-03	0.0	0.0	0.0	0.0
28	0.132E-04	-0.391E-10	0.132E-04	-0.170E-03	0.0	0.0	0.0	0.0
29	0.226E-04	0.0	0.226E-04	0.0	0.0	0.0	0.0	0.0
30	0.856E-04	0.305E-11	0.856E-04	0.204E-05	0.0	0.0	0.0	0.0
31	0.560E-06	0.390E-07	0.562E-06	0.398E 01	0.0	0.0	0.0	0.0
32	0.160E-05	0.442E-07	0.160E-05	0.158E 01	0.0	0.0	0.0	0.0
33	0.256E-05	0.530E-07	0.256E-05	0.119E 01	0.0	0.0	0.0	0.0
34	0.362E-05	0.694E-07	0.362E-05	0.110E 01	0.0	0.0	0.0	0.0
35	0.490E-05	0.999E-07	0.490E-05	0.117E 01	0.0	0.0	0.0	0.0
36	0.662E-05	0.163E-06	0.662E-05	0.141E 01	0.0	0.0	0.0	0.0
37	0.902E-05	0.295E-06	0.903E-05	0.187E 01	0.0	0.0	0.0	0.0
38	0.128E-04	0.658E-06	0.128E-04	0.294E 01	0.0	0.0	0.0	0.0
39	0.209E-04	0.201E-05	0.210E-04	0.550E 01	0.0	0.0	0.0	0.0
40	0.535E-04	0.129E-04	0.550E-04	0.135E 02	0.0	0.0	0.0	0.0

```
TIME COUNTER, JT =  2      TIME T =  0.1799999E 24     ITERATION NO., ITER =    1
ERROR = 0.5795598E 02      BASE =  0.2770085E 03
TIME COUNTER, JT =  2      TIME T =  0.1799999E 24     ITERATION NO., ITER =    2
ERROR =  0.4849392E 01     BASE =  0.2770085E 03
TIME COUNTER, JT =  2      TIME T =  0.1799999E 24     ITERATION NO., ITER =    3
ERROR =  0.1486197E 01     BASE =  0.2770085E 03
TIME COUNTER, JT =  3      TIME T =  0.3599999E 24     ITERATION NO., ITER =    1
ERROR =  0.4153691E 01     BASE =  0.2157357E 03
TIME COUNTER, JT =  3      TIME T =  0.3599999E 24     ITERATION NO., ITER =    2
ERROR =  0.2405495E 01     BASE =  0.2157357E 03
TIME COUNTER, JT =  3      TIME T =  0.3599999E 24     ITERATION NO., ITER =    3
ERROR =  0.1769759E 01     BASE =  0.2157357E 03
TIME COUNTER, JT =  4      TIME T =  0.5399998E 24     ITERATION NO., ITER =    1
ERROR =  0.2322071E 01     BASE =  0.2131382E 03
TIME COUNTER, JT =  4      TIME T =  0.5399998E 24     ITERATION NO., ITER =    2
ERROR =  0.2802158E 01     BASE =  0.2131382E 03
TIME COUNTER, JT =  4      TIME T =  0.5399998E 24     ITERATION NO., ITER =    3
ERROR =  0.2140439E 01     BASE =  0.2131382E 03
TIME COUNTER, JT =  4      TIME T =  0.5399998E 24     ITERATION NO., ITER =    4
ERROR =  0.1121272E 01     BASE =  0.2131382E 03
TIME COUNTER, JT =  4      TIME T =  0.5399998E 24     ITERATION NO., ITER =    5
ERROR =  0.2349969E 00     BASE =  0.2131382E 03
```

NODE	HF(I,2)	HS(I,2)	BF(I,2)	BS(I,2)	ZETA(I)
1	11.026	0.0	441.044	169.982	-430.018
2	11.026	0.0	441.042	169.984	-430.016
3	10.915	0.0	436.599	174.316	-425.684
4	10.752	0.0	430.062	180.689	-419.311
5	10.752	0.0	430.062	180.690	-419.310
6	10.465	0.0	418.586	191.878	-408.122
7	10.176	0.0	407.023	203.153	-396.847
8	10.176	0.0	407.023	203.153	-396.847
9	9.793	0.0	391.726	218.067	-381.933
10	9.408	0.0	376.323	233.085	-366.915
11	9.408	0.0	376.323	233.085	-366.915
12	8.904	0.0	356.147	252.756	-347.244
13	8.432	0.0	337.280	271.152	-328.848
14	8.432	0.0	337.280	271.152	-328.848
15	7.847	0.0	313.871	293.976	-306.024
16	7.306	0.0	292.222	315.083	-284.917
17	7.306	0.0	292.222	315.083	-284.917
18	6.648	0.0	265.920	340.728	-259.272
19	5.982	0.0	239.291	366.691	-233.309
20	5.982	0.0	239.291	366.691	-233.309
21	5.358	0.0	214.329	391.029	-208.971
22	4.719	0.0	188.777	415.942	-184.058
23	4.719	0.0	188.777	415.942	-184.058
24	3.967	0.0	158.663	445.303	-154.697
25	3.309	0.0	132.363	470.947	-129.053
26	3.309	0.0	132.363	470.946	-129.054
27	2.523	0.0	100.928	501.595	-98.404
28	1.846	0.0	73.828	528.017	-71.983
29	1.846	0.0	73.828	528.017	-71.983
30	1.098	0.0	43.916	557.182	-42.818
31	0.241	0.0	9.657	590.584	-9.415
32	0.241	0.0	9.657	590.584	-9.415

```
RECHARGE TO UNCONFINED AQUIFER AS A FUNCTION OF TIME

ELEMENT I =     1      NO. OF SUCCEEDING ELEMENTS WITH SAME RECHARGE, INC =   39
  1 0.000000100   2 0.000000100   3 0.000000100   4 0.000000100

COASTAL BOUNDARY ALONG WHICH MIXED B.C. IS SPECIFIED

         NODECF(I,1) NODECF(I,2) PERMEABILITY KC
   1         31          32      0.0099999979

NODES AT WHICH SALTWATER HEAD IS SPECIFIED


   1      NODEHS(I) =     31
  1       0.0        2       0.0        3       0.0        4       0.0


   2      NODEHS(I) =     32
  1       0.0        2       0.0        3       0.0        4       0.0

ELEMENT TYPE AT TIME T = TSTART
ELEMENT   LMNTYP(I)
    1     2    2    2    3    2    4    2    5    2    6    2    7    2    8    2    9    2   10    2
   11     2   12    2   13    2   14    2   15    2   16    2   17    2   18    0   19    0   20    0
   21     0   22    0   23    0   24    0   25    0   26    0   27    0   28    0   29    0   30    0
   31     0   32    0   33    0   34    0   35    0   36    0   37    0   38    0   39    0   40    0

INITIAL CONDITIONS
  NODE      HF(I,1)     HS(I,1)    BF(I,1)     BS(I,1)
    1       20.000      0.0        299.900     0.100
    2       20.000      0.0        299.900     0.100
    3       18.530      0.0        299.900     0.100
    4       16.410      0.0        299.900     0.100
    5       16.410      0.0        299.900     0.100
    6       15.090      0.0        299.900     0.100
    7       13.770      0.0        299.900     0.100
    8       13.770      0.0        299.900     0.100
    9       12.660      0.0        299.900     0.100
   10       11.560      0.0        299.900     0.100
   11       11.560      0.0        299.900     0.100
   12       10.530      0.0        299.900     0.100
   13        9.580      0.0        299.900     0.100
   14        9.580      0.0        299.900     0.100
   15        8.700      0.0        299.900     0.100
   16        7.890      0.0        299.900     0.100
   17        7.890      0.0        299.900     0.100
   18        7.090      0.0        276.500    23.500
   19        6.280      0.0        245.600    55.000
   20        6.280      0.0        245.600    55.000
   21        5.580      0.0        217.700    82.400
   22        4.870      0.0        189.760   110.300
   23        4.870      0.0        189.760   110.300
   24        4.340      0.0        158.260   141.000
   25        3.240      0.0        130.900   169.000
   26        3.240      0.0        130.900   169.000
   27        2.550      0.0         99.300   201.000
   28        1.800      0.0         73.000   226.000
   29        1.800      0.0         73.000   226.000
   30        1.040      0.0         42.000   253.000
   31        0.200      0.0          8.570   289.300
   32        0.200      0.0          8.570   289.300
```

ELEMENT DATA

ELEM. NO.	NODE(I,1)	NODE(I,2)	NODE(I,3)	F.W.PERM. X	F.W.PERM. Y	S.W.PERM. X	S.W.PERM. Y	POROSITY	NCONF(I)	NLEAKY(I)
1	1	4	3	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
2	4	7	6	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
3	7	10	9	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
4	10	13	12	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
5	13	16	15	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
6	16	19	18	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
7	19	22	21	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
8	22	25	24	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
9	25	28	27	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
10	28	31	30	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
11	1	3	2	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
12	4	6	5	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
13	7	9	8	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
14	10	12	11	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
15	13	15	14	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
16	16	18	17	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
17	19	21	20	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
18	22	24	23	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
19	25	27	26	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
20	28	30	29	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
21	3	4	5	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
22	6	7	8	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
23	9	10	11	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
24	12	13	14	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
25	15	16	17	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
26	18	19	20	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
27	21	22	23	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
28	24	25	26	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
29	27	28	29	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
30	30	31	32	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
31	3	5	2	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
32	6	8	5	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
33	9	11	8	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
34	12	14	11	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
35	15	17	14	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
36	18	20	17	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
37	21	23	20	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
38	24	26	23	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
39	27	29	26	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0
40	30	32	29	0.0100000	0.0100000	0.0102500	0.0102500	0.29999995	0	0