UNITED STATES

DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Finite-Difference Model to Simulate the Areal Flow of Salt Water and Fresh Water Separated by an Interface

by J. W. Mercer, S. P. Larson, C. R. Faust

Open File Report 80-407

Reston VA
April 1980

For Additional Information write to:

Chief Hydrologist
U.S. Geological Survey, WRD
MS. # 411 National Center
Reston, Virginia 22092

Request, at cost, for the Card Deck listed at the end of the report should be directed to:

Ralph N. Eicher, Chief
Office of Teleprocessing
M. S. # 805, National Center
U.S. Geological Survey
Reston, Virginia 22092

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

ILLUSTRATIONS

TABLES

Finite-Difference Model to Simulate the Areal Flow of Salt Water and Fresh Water Separated by an Interface

by

James W. Mercer, Steven P. Larson, and Charles R. Faust

ABSTRACT

Model documentation is presented for a two-dimensional (areal) model capable of simulating ground-water flow of salt water and fresh water separated by an interface. The partial differential equations are integrated over the thicknesses of fresh water and salt water resulting in two equations describing the flow characteristics in the areal domain. These equations are approximated using finite-difference techniques and the resulting algebraic equations are solved for the dependent variables, fresh-water head and salt-water head. An iterative solution method was found to be most appropriate.

The program is designed to simulate time-dependent problems such as those associated with the development of coastal aquifers, and can treat water-table conditions or confined conditions with steady-state leakage of fresh water. The program will generally be most applicable to the analysis of regional aquifer problems in which the zone between salt water and fresh water can be considered a surface (sharp interface). Example problems and a listing of the computer code are included.

INTRODUCTION

The modeling of ground water in coastal aquifers is an important and difficult problem in water resources investigations. The primary difficulty involves the efficient and accurate simulation of the salt-water front. The convective-dispersive solute-transport equation probably best describes the movement of this front, however this equation can be difficult to solve. For certain problems where the transition zone caused by hydrodynamic dispersion is relatively narrow (when compared with the aquifer extent and thickness), the simulation can sometimes be simplified by assuming that the two fluids are immiscible and separated by a sharp interface. This assumption is particularly useful for large scale areal problems.

This interface modeling approach combined with the vertical integration was first presented by Shamir and Dagan (1971). They considered a vertical cross section and vertical integration resulted in one-dimensional equations, which were solved using a finite-difference approximation. For this one-dimensional case, they were able to track the salt-water toe and regenerate their grid for each time step; thus, solving two equations on the seaward side of the toe and only one equation on the landward side. Bonnet and Sauty (1975) extended the work of Shamir and Dagan to two dimensions. The resulting areal model was approximated using finite-difference techniques. Finally, Pinder and Page (1976) used the same equations given in Bonnet and Sauty, only approximating them using a Galerkin, finite-element approach.

In this report, the equations describing two-dimensional, areal flow of immiscible fresh water and salt water are developed by integrating the flow equations over the vertical dimension. These equations are approximated using finite-difference techniques and the resulting nonlinear algebraic equations are linearized and are solved using an iterative method. The computer program that implements this solution scheme is described and examples are presented to demonstrate its use.

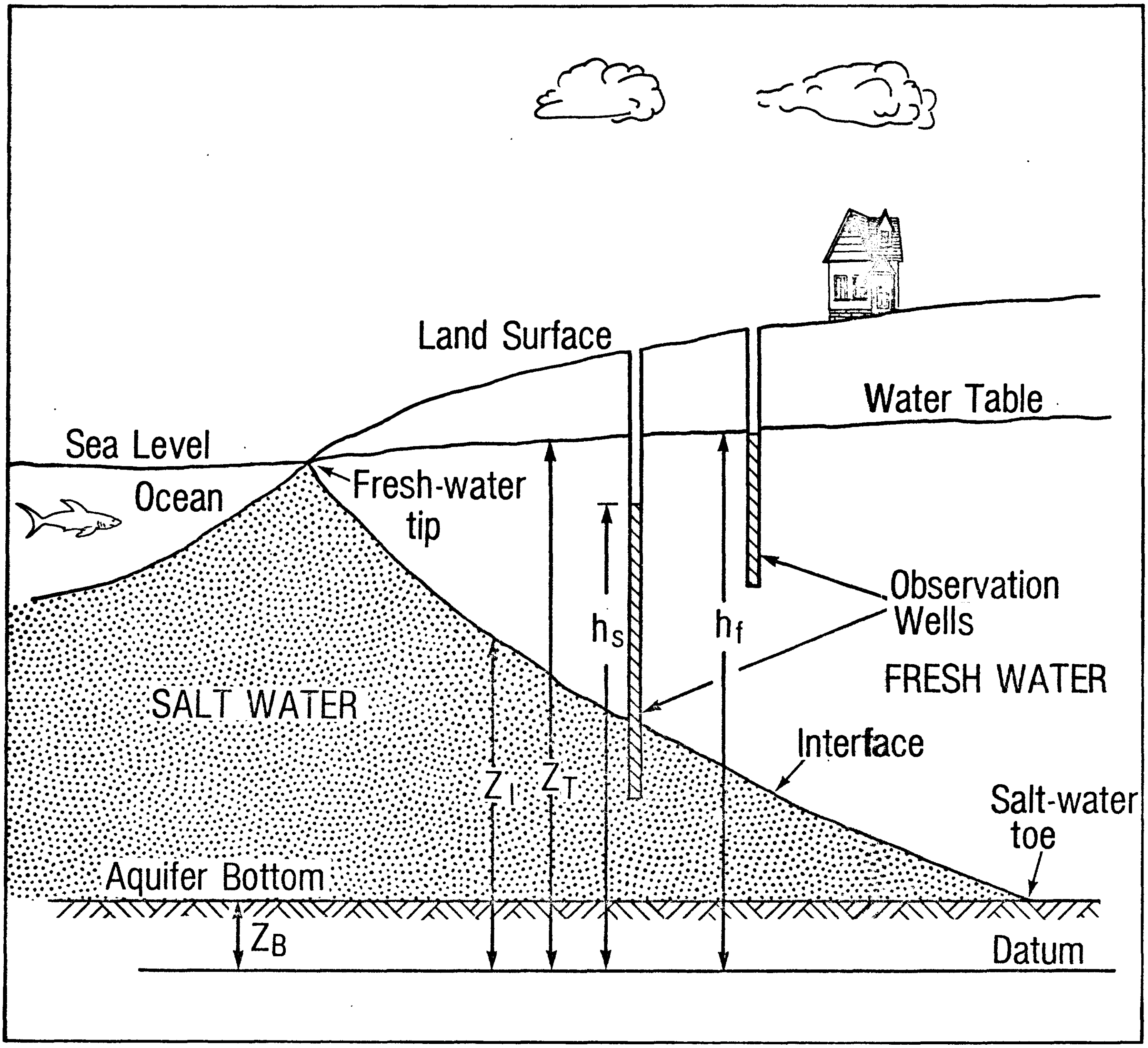

Figure 1. Diagrammatic representation of salt-water interface in a coastal aquifer.

MODEL DESCRIPTION

Governing Equations

To simulate the transient position of the salt-water interface shown in figure 1, it is necessary to solve simultaneously the equations describing the fresh-water and salt-water hydrodynamics.

Mass balance. The mass balance for salt water, s, and fresh water, f, may be written as follows:

$$S_s \frac{\partial h_s}{\partial t} + \nabla \cdot \bar{q}_s - Q_s = 0 \ , \qquad (1)$$

and

$$S_f \frac{\partial h_f}{\partial t} + \nabla \cdot \bar{q}_f - Q_f = 0 \ , \qquad (2)$$

where h is hydraulic head, $\bar{q}$ is the Darcy velocity (a vector quantity), S is the specific storage, Q is a source/sink term (negative for sink), and $\nabla \equiv \frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k}$ (where $\bar{i}$, $\bar{j}$, and $\bar{k}$ are unit vectors in the x,y, and z directions, respectively).

Momentum balance. It is assumed that Darcy's equation may be used as simplified momentum balances. For salt water and fresh water these are

$$\bar{q}_s = - \bar{\bar{K}}_s \cdot \nabla h_s \ , \qquad (3)$$

and

$$\bar{q}_f = - \bar{\bar{K}}_f \cdot \nabla h_f \ , \qquad (4)$$

where $\bar{\bar{K}}$ is hydraulic conductivity (a tensor quantity).

Three-dimensional equations. Substitution of equations (3) and (4) into (1) and (2) respectively yields

$$S_s \frac{\partial h_s}{\partial t} - \nabla\cdot(\bar{\bar{K}}_s\cdot\nabla h_s) - Q_s = 0 \ , \qquad (5)$$

and

$$S_f \frac{\partial h_f}{\partial t} - \nabla\cdot(\bar{\bar{K}}_f\cdot\nabla h_f) - Q_f = 0, \qquad (6)$$

which are the three-dimensional, ground-water flow equations for salt water and fresh water.

Vertical integration. To obtain the areal two-dimensional form of equations 5 and 6, they are integrated in the vertical dimension over the regions for which they are defined (fig. 1). Integration of (5) and (6) yields

$$\int_{z_B}^{z_I} [S_s \frac{\partial h_s}{\partial t} - \nabla\cdot (\bar{\bar{K}}_s\cdot\nabla h_s) - Q_s]dz = 0, \qquad (7)$$

and

$$\int_{z_I}^{z_T} [S_f \frac{\partial h_f}{\partial t} - \nabla\cdot (\bar{\bar{K}}_f\cdot\nabla h_f) - Q_f]dz = 0, \qquad (8)$$

where z_B is the elevation of the aquifer base, z_I is the elevation of the interface, and z_T is the elevation of the top of the fresh water region. To evaluate (7) and (8) we use Leibnitz' rule in the following form to reverse the order of integration and differentiation (Korn and Korn, 1961, p. 100):

$$\int_{z_1}^{z_2} \frac{\partial \Psi}{\partial x}\, dz = \frac{\partial}{\partial x}\int_{z_1}^{z_2} \Psi dz + \Psi(x,y,z,t)\Big|_{z_1} \frac{\partial z_1}{\partial x}$$

$$- \Psi(x,y,z,t)\Big|_{z_2} \frac{\partial z_2}{\partial x} . \tag{9}$$

It is also helpful to define a quantity averaged in the z-dimension as

$$\langle\Psi\rangle = \frac{1}{b}\int_{z_1}^{z_2} \Psi dz. \tag{10}$$

Application of Leibnitz' rule to the time derivative in (7) yields

$$\int_{z_B}^{z_I} S_s \frac{\partial h_s}{\partial t}\, dz = S_s\left(\frac{\partial}{\partial t}\int_{z_B}^{z_I} h_s dz + h_s\Big|_{z_B} \frac{\partial z_B}{\partial t} - h_s\Big|_{z_I} \frac{\partial z_I}{\partial t}\right), \tag{11}$$

where we have assumed S_s is invarient with respect to depth. In terms of averaged quantities defined by equation (10) and assuming z_B is invariant with time, equation (11) may be rewritten as

$$\int_{z_B}^{z_I} S_s \frac{\partial h_s}{\partial t} dz = S_s \left[\frac{\partial}{\partial t} (b_s \langle h_s \rangle) - h_s\Big|_{z_I} \frac{\partial z_I}{\partial t}\right], \tag{12}$$

where $b_s = (z_I - z_B)$. Using the Dupuit approximation that the hydraulic head defined at z_B and z_I does not differ significantly from the average, $\langle h_s \rangle$, and expanding the time derivative on the right side of (12) yields

$$\begin{aligned}\int_{z_B}^{z_I} S_s \frac{\partial h_s}{\partial t} dz &= S_s\left[b_s \frac{\partial \langle h_s \rangle}{\partial t} + \langle h_s \rangle \frac{\partial b_s}{\partial t} - h_s\Big|_{z_I} \frac{\partial z_I}{\partial t}\right] \\ &= S_s\left[b_s \frac{\partial \langle h_s \rangle}{\partial t} + \langle h_s \rangle \frac{\partial z_I}{\partial t} - h_s\Big|_{z_I} \frac{\partial z_I}{\partial t}\right] \qquad (13) \\ &= S_s b_s \frac{\partial \langle h_s \rangle}{\partial t}\end{aligned}$$

Integration of the x-component of $\nabla \cdot (\bar{\bar{K}}_s \cdot \nabla h_s)$ in (7) using Leibnitz rules gives

$$\begin{aligned}\int_{z_B}^{z_I} \frac{\partial}{\partial x} \left(K_{sx} \frac{\partial h_s}{\partial x}\right) dz &= \frac{\partial}{\partial x} \left(b_s K_{sx} \left\langle\frac{\partial h_s}{\partial x}\right\rangle\right) + K_{sx} \frac{\partial h_s}{\partial x}\Big|_{z_B} \frac{\partial z_B}{\partial x} \\ &\quad - K_{sx} \frac{\partial h_s}{\partial x}\Big|_{z_I} \frac{\partial z_I}{\partial x}, \qquad (14)\end{aligned}$$

where we have assumed that the hydraulic conductivity is colinear with the coordinate axis and is invarient with depth. Similarly, the integrated y- component is

$$\int_{z_B}^{z_I} \frac{\partial}{\partial y} \left(K_{sy} \frac{\partial h_s}{\partial y}\right) dz = \frac{\partial}{\partial y} \left(b_s K_{sy} \left\langle\frac{\partial h_s}{\partial y}\right\rangle\right) + K_{sy} \left.\frac{\partial h_s}{\partial y}\right|_{z_B} \frac{\partial z_B}{\partial y} - K_{sy} \left.\frac{\partial h_s}{\partial y}\right|_{z_I} \frac{\partial z_I}{\partial y} . \tag{15}$$

For the z-component, integration leads to

$$\int_{z_B}^{z_I} \frac{\partial}{\partial z} \left(K_{sz} \frac{\partial h_s}{\partial z}\right) dz = K_{sz} \left.\frac{\partial h_s}{\partial z}\right|_{z_I} - K_{sz} \left.\frac{\partial h_s}{\partial z}\right|_{z_B} . \tag{16}$$

Substitution of (13) through (16) into (7) and use of equation (3) gives

$$S_s b_s \frac{\partial \langle h_s \rangle}{\partial t} - \frac{\partial}{\partial x}\left(b_s K_{sx} \left\langle\frac{\partial h_s}{\partial x}\right\rangle\right) - \frac{\partial}{\partial y} \left(b_s K_{sy} \left\langle\frac{\partial h_s}{\partial y}\right\rangle\right) - b_s \langle Q_s \rangle$$

$$- q_{sx}\Big|_{z_I} \frac{\partial z_I}{\partial x} - q_{sy}\Big|_{z_I} \frac{\partial z_I}{\partial y} + q_{sz}\Big|_{z_I}$$

$$+ q_{sx}\Big|_{z_B} \frac{\partial z_B}{\partial x} + q_{sy}\Big|_{z_B} \frac{\partial z_B}{\partial y} - q_{sz}\Big|_{z_B} = 0, \tag{17}$$

which is the vertically-integrated flow equation for salt water. Note that the source/sink term has also been integrated.

Using a similar development, the vertically-integrated fresh-water equation is

$$S_f b_f \frac{\partial \langle h_f \rangle}{\partial t} - \frac{\partial}{\partial x}(b_f K_{fx} \langle \frac{\partial h_f}{\partial x} \rangle) - \frac{\partial}{\partial y}(b_f K_{fy} \langle \frac{\partial h_f}{\partial y} \rangle) - b_f \langle Q_f \rangle$$

$$+ q_{fx}\Big|_{z_I} \frac{\partial z_I}{\partial x} + q_{fy}\Big|_{z_I} \frac{\partial z_I}{\partial y} - q_{fz}\Big|_{z_I}$$

$$- q_{fx}\Big|_{z_T} \frac{\partial z_T}{\partial x} - q_{fy}\Big|_{z_T} \frac{\partial z_T}{\partial y} + q_{fz}\Big|_{z_T} = 0, \qquad (18)$$

where $b_f = z_T - z_I$. It can be shown (Hantush, 1964, p. 300) that

$$q_{sx}\Big|_{z_B} \frac{\partial z_B}{\partial x} + q_{sy}\Big|_{z_B} \frac{\partial z_B}{\partial y} - q_{sz}\Big|_{z_B} = 0 \qquad (19)$$

for an impermeable base and thus these terms can be eliminated from (17). Also, it can be shown (Faust and Mercer, 1979, p. 27) that the average of the derivative ($<\frac{\partial h_s}{\partial x}>$) will be equal to the derivative of the average ($\frac{\partial <h_s>}{\partial x}$) if the Dupuit approximation is used (i.e. $<h_s> \cong h_s\Big|_{z_B}$). The remaining terms that are evaluated at z_I and z_T can be replaced by considering the conditions of a material surface (the interface or the water table).

Surface conditions. To evaluate the movement of the water table and the interface, the following relationships are necessary. Suppose the surface at time t is represented by

$$F(x,y,z,t) = 0.$$

At a later time, $t+\delta t$, a particle at point (x,y,z) will be displaced to $(x+\delta x,\ y+\delta y,\ z+\delta z) = (x+v_x\delta t,\ y+v_y\delta t,\ z+v_z\delta t)$ where v_x, v_y, and v_z are the velocity components at (x,y,z) on the interface. The new interface is, therefore, given by

$$F(x+ v_x\delta t, y + v_y\delta t, z + v_z\delta t,\ t + \delta t) = 0 = F\ (x,y,z,t)$$

and therefore,

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + \bar{v} \cdot \nabla F = 0, \qquad (20)$$

where $\frac{D}{Dt}$ is the material (substantial) derivative or derivative following the motion. Equation (20) is known as the Kelvin relation governing the the motion in a liquid on a surface containing a given set of particles.

For the salt water-fresh water interface, $F = z(t) - z_I(x,y,t)$, which when substituted into (20) yields

$$\frac{D}{Dt}(z-z_I)\Big|_{z_I} = \left(v_z - \frac{\partial z_I}{\partial t} - v_x \frac{\partial z_I}{\partial x} - v_y \frac{\partial z_I}{\partial y}\right)\Big|_{z_I} = 0. \tag{21}$$

Multiplying (21) by porosity, ϕ, and rearranging terms gives

$$\phi \frac{\partial z_I}{\partial t} = q_z\Big|_{z_I} - q_x\Big|_{z_I} \frac{\partial z_I}{\partial x} - q_y\Big|_{z_I} \frac{\partial z_I}{\partial y}. \tag{22}$$

Thus, this equation can be used to replace terms in (17) and (18) that are evaluated at z_I.

Let us temporarily define the top surface as $H(x,y,z,t)$ recognizing that $H = z_T(x,y)$ for confined problems and $H = h_f(x,y,z,t)$ for unconfined problems. The surface is thus described as, $F = z(t) - H(x,y,z,t)$, and taking the material derivative,

$$\frac{DF}{Dt} = \frac{Dz}{Dt} - \frac{DH}{Dt} = 0$$

or

$$\frac{dz}{dt} = \frac{\partial H}{\partial t} + \frac{\partial H}{\partial x}\frac{dx}{dt} + \frac{\partial H}{\partial y}\frac{dy}{dt} + \frac{\partial H}{\partial z}\frac{dz}{dt},$$

and thus,

$$v_z = \frac{\partial H}{\partial t} + v_x \frac{\partial H}{\partial x} + v_y \frac{\partial H}{\partial y} + v_z \frac{\partial H}{\partial z} \quad .$$

Multiplying by porosity, ϕ, and introducing α to distinguish between confined and unconfined conditions ($\alpha = 1$ for unconfined, $\alpha = 0$ for confined) gives

$$\phi v_z \left(1 - \frac{\partial H}{\partial z}\right)\Bigg|_H - \alpha\phi \frac{\partial h_f}{\partial t}\Bigg|_H = q_x \frac{\partial H}{\partial x}\Bigg|_H + q_y \frac{\partial H}{\partial y}\Bigg|_H \qquad (23)$$

Note that we have substituted $h_f\big|_H$ for the time derivative of H. Recall that for confined problems this term vanishes (because it is not a function of time) and for unconfined problems, this term becomes the time derivative of h_f. By introducing α, we obtain these desired results. Also, from the Dupuit approximation, the derivative $\frac{\partial H}{\partial z}$ can be considered negligible for the unconfined case and is zero for the confined case; therefore, the term containing this derivative is ignored. Finally, the term ϕv_z represents the total velocity of a particle at the surface H and is composed of $q_z\big|_H$ and recharge or leakage if they exist. Thus,

$$\phi v_z = q_z\Big|_H - \alpha R + (1-\alpha)\ K'/b'(\langle h_f\rangle - h'), \qquad (24)$$

where R is the rate of recharge to the water table, K' is the hydraulic conductivity of an overlying confining bed, b' is its thickness and h' is the head on the opposite side of the confining bed. Substituting (24) into (23) and recalling from the Dupuit approximation that $h_f \big|_H \cong \langle h_f \rangle$ gives

$$q_z \Big|_H - q_x \Big|_H \frac{\partial H}{\partial x} - q_y \Big|_H \frac{\partial H}{\partial y} = \alpha R - (1-\alpha) \frac{K'}{b'} (\langle h_f \rangle - h') + \alpha\phi \frac{\partial \langle h_f \rangle}{\partial t} . \quad (25)$$

Substituting (22) and (25) into (18) gives

$$S_f b_f \frac{\partial \langle h_f \rangle}{\partial t} - \frac{\partial}{\partial x} (b_f K_{fx} \frac{\partial \langle h_f \rangle}{\partial x}) - \frac{\partial}{\partial y} (b_f K_{fy} \frac{\partial \langle h_f \rangle}{\partial y}) - b_f \langle Q_f \rangle$$

$$- \phi \frac{\partial z_I}{\partial t} + \alpha R - (1-\alpha) \frac{K'}{b'} (\langle h_f \rangle - h') + \alpha\phi \frac{\partial \langle h_f \rangle}{\partial t} = 0. \quad (26)$$

Also, substituting (19) and (22) into (17) gives

$$S_s b_s \frac{\partial \langle h_s \rangle}{\partial t} - \frac{\partial}{\partial x} (b_s K_{sx} \frac{\partial \langle h_s \rangle}{\partial x}) - \frac{\partial}{\partial x} (b_s K_{sy} \frac{\partial \langle h_s \rangle}{\partial y}) - b_s \langle Q_s \rangle$$

$$+ \phi \frac{\partial z_I}{\partial t} = 0. \quad (27)$$

Final equations. The last terms that need to be evaluated are the time derivatives of the interface. A stability requirement (what keeps one fluid from pushing into the other) is that the pressures of the two fluids at the interface are equal:

$$\rho_s g\,(h_s - z_I) = \rho_f g(h_f - z_I), \tag{28}$$

where g is the gravitational constant and ρ_f and ρ_s are fresh-water and salt-water densities, respectively. Rearranging terms, recalling the assumption that head is uniform with depth and solving for z_I gives

$$z_I = \frac{\langle h_f \rangle \rho_f - \langle h_s \rangle \rho_s}{\rho_f - \rho_s}\ . \tag{29}$$

Taking the time derivative gives

$$\frac{\partial z_I}{\partial t} = \rho_f^* \ \frac{\partial \langle h_f \rangle}{\partial t} - \rho_s^* \ \frac{\partial \langle h_s \rangle}{\partial t}\ , \tag{30}$$

where $\rho_f^* = \rho_f/(\rho_f-\rho_s)$ and $\rho_s^* = \rho_s/ (\rho_f - \rho_s)$.

Equation (30) indicates that the rate of change in the elevation of the interface is proportional to the rates of change in heads. Changes in fresh-water and salt-water heads could be off setting, resulting in little or no change in the interface.

We assume that hydraulic conductivity is measured in terms of fresh water, therefore, $K_f = K = k\rho_f g/\mu_f$ and $K_s = k\rho_s g/\mu_s$, where k is the intrinsic permeability and μ_f and μ_s are the fresh-water and salt-water dynamic viscosities, respectively. Combining these and rearranging terms gives

$$K_s = K_f \; \mu_f \rho_s/\rho_f \mu_s, \tag{31}$$

Similarly for specific storage, $S_f = \rho_f g(\alpha' + \phi\beta)$ and $S_s = \rho_s g(\alpha' + \phi\beta)$, and

$$S_s = S_f \; \rho_s/\rho_f, \tag{32}$$

where we have assumed that the vertical compressibility coefficient of the medium, α', and the compressibility coefficient of the fluid, β, are approximately the same for salt water and fresh water. Substitution of (30), (31) and (32) into (27), and (30) into (26) yields for the salt-water and fresh-water equations respectively,

$$\frac{\rho_s}{\rho_f} S_f b_s \frac{\partial h_s}{\partial t} + \phi(\rho_f^* \frac{\partial h_f}{\partial t} - \rho_s^* \frac{\partial h_s}{\partial t})$$

$$- \frac{\partial}{\partial x} (b_s K_{fx} \frac{\mu_f}{\rho_f} \frac{\rho_s}{\mu_s} \frac{\partial h_s}{\partial x}) - \frac{\partial}{\partial y} (b_s K_{fy} \frac{\mu_f}{\rho_f} \frac{\rho_s}{\mu_s} \frac{\partial h_s}{\partial y}) \qquad (33)$$

$$- Q_{ws} \delta(x-x_i, y-y_i) = 0,$$

and

$$S_f b_f \frac{\partial h_f}{\partial t} - \phi(\rho_f^* \frac{\partial h_f}{\partial t} - \rho_s^* \frac{\partial h_s}{\partial t}) + \alpha\phi \frac{\partial h_f}{\partial t}$$

$$- \frac{\partial}{\partial x} (b_f K_{fx} \frac{\partial h_f}{\partial x}) - \frac{\partial}{\partial y} (b_f K_{fy} \frac{\partial h_f}{\partial y}) \qquad (34)$$

$$- Q_{wf} \delta(x-x_i, y-y_i) + \alpha R - (1-\alpha) \frac{K'}{b'} (h_f - h') = 0,$$

which are the final vertically-integrated equations, where the averaging brackets $< >$ have been omitted. Equations (33) and (34) are formulated in terms of h_f and h_s to facilitate treatment of constant head boundaries. However, had we solved (28) for h_s, our final equations could have been formulated in terms of h_f and z_I. Similarly, the final equations could have been formulated in terms of h_s and z_I. Note that for free surface conditions, S_f and S_s are generally very small compared with porosity. Also, $b_f <Q_f>$ has been replaced with a source-sink term $Q_{wf}\ \delta(x-x_i, y-y_i)$, where $\delta(x,y)$ is the Dirac delta function.

Numerical Development

The technique used to solve equations (33) and (34) is based on the finite-difference method. For this method the areal extent of the reservoir is subdivided into rectangular grid blocks in which the fluid and reservoir properties are assumed uniform. The continuous derivatives in equations (33) and (34) are approximated by finite-difference expressions at points (nodes) in the centers of the blocks. This results in a nonlinear system of 2NB equations with 2NB unknowns (the values of fresh-water and salt-water heads at the nodes) where NB is the number of nodes. The general finite-difference representation and solution procedure for this system of nonlinear equations are outlined below.

Finite difference representation. Equations (33) and (34) may be written in compact, implicit finite-difference form as

$$\Delta_x (T_{sx} \Delta_x h_s^{n+1}) + \Delta_y (T_{sy} \Delta_y h_s^{n+1}) + AQ_{ws}$$

$$- C\Delta_t h_f - D\Delta_t h_s = 0, \qquad (35)$$

and

$$\Delta_x (T_{fx} \Delta_x h_f^{n+1}) + \Delta_y (T_{fy} \Delta_y h_f^{n+1}) + AQ_{wf}$$

$$- A\alpha R - A(1-\alpha) \frac{K'}{b'} (h_{i,j}^{n+1} - h'_{i,j})$$

$$- F\Delta_t h_f - B\Delta_t h_s = 0, \qquad (36)$$

where i and j are indices in the x- and y-directions, n is the index for the time level, and the grid block area is defined as $A = \Delta x\ \Delta y$, where Δx and Δy are the spacings in the x- and y-directions, respectively. The equations have been multiplied by the grid block area to produce equations having units of volumetric rate. It can be shown that the resulting equations have a block symmetric form, which is an important consideration with regard to the application of solution procedures. The transmissive terms are given by

$$T_{sx} = (K_{fx}/\Delta x) b_s \Delta y \mu_f \rho_s / \rho_f \mu_s, \tag{37a}$$

$$T_{sy} = (K_{fy}/\Delta y) b_s \Delta x \mu_f \rho_s / \rho_f \mu_s, \tag{37b}$$

$$T_{fx} = (K_{fx}/\Delta x) b_f \Delta y, \tag{37c}$$

$$T_{fy} = (K_{fy}/\Delta y) b_f \Delta x. \tag{37d}$$

The difference operator acts as follows in the x- direction:

$$\Delta_x (T_{sx}\ \Delta_x h_s^{n+1}) = T_{sx_{i+\frac{1}{2},j}} (h_{s_{i+1,j}}^{n+1} - h_{s_{i,j}}^{n+1}) - T_{sx_{i-\frac{1}{2},j}} (h_{s_{i,j}}^{n+1} - h_{s_{i-1,j}}^{n+1}). \tag{38}$$

The interblock transmissive terms (values at $i \pm \frac{1}{2}$ and $j \pm \frac{1}{2}$) are composed of two parts: that which is a function of space only (for example, $K_{fx}/\Delta x$) and that which is a function of head (for example, b_s).

To approximate these terms requires averaging or weighting of the various components over each grid block. For the space dependent part, this is accomplished by using a harmonic mean, for example,

$$(K_{fx}/\Delta x)_{i+\frac{1}{2}} = 2K_{fx_{i+1}} K_{fx_i} / (K_{fx_i} \Delta x_{i+1} + K_{fx_{i+1}} \Delta x_i). \tag{39}$$

The head dependent part of the these terms is computed by a weighted arithmetic average, for example,

$$b_{s_{i+\frac{1}{2}}} = \beta b_{s_{i+1}} + (1-\beta) b_{s_i}, \tag{40}$$

where β may be determined such that the upstream value is used. The upstream node is located by comparing the heads at nodes i and i+1, and β is set to give b_s the value at the node having the larger head. Alternatively, other values of β can be used; for example, $\beta = \frac{1}{2}$ produces a midpoint weighting. Of the two procedures, upstream weighting yields a lower-order approximation of the spatial derivatives but exhibits a more stable solution. The time coefficients in equations (35) and (36) are given by

$$C = \Delta x \Delta y \phi \rho_f^*, \tag{41a}$$

$$D = \Delta x \Delta y (S_f b_s \rho_s / \rho_f - \phi \rho_s^*), \tag{41b}$$

$$F = \Delta x \Delta y (S_f b_f - \phi \rho_f^* + \alpha \phi), \tag{41c}$$

$$B = \Delta x \Delta y \phi \rho_s^*. \tag{41d}$$

The difference operator acts over the time domain as follows:

$$C\Delta_t h_f = C_{i,j}(h_{f_{i,j}}^{n+1} - h_{f_{i,j}}^{n})/\Delta t, \qquad (42)$$

which is a backward difference approximation.

Solution procedure. A block form of the line-successive over-relaxation technique (Varga, 1962), is used to solve the set of algebraic equations. A block consists of the pair of equations at each node. The 2 by 2 sub-matrices multiplying dependent variables in rows j-1 and j+1 are treated explicitly resulting in a reduced coefficient matrix of block tri-diagonal form. The set of matrix equations for each row are solved sequentially to produce new estimates of h_f and h_s. The explicit terms are updated and the process repeated until satisfactory convergence is achieved. Rewritting equations (35) and (36) in matrix form gives

$$[A]_j\{h^{n+1}\}_j^{\tau} = \{Q^n\}_j - [B]_{j-1}\{h^{n+1}\}_{j-1}^{k+1} - [B]_{j+1}\{h^{n+1}\}_{j+1}^{k},$$

$$j = 1,\ldots, NY \qquad (43)$$

where NY is the number of rows. Matrix $[A]_j$ is block tri-diagonal and is composed of coefficients multiplying values of the dependent variable within row j. Matrices $[B]_{j-1}$ and $[B]_{j+1}$ are block diagonal and are composed of coefficients multiplying the dependent variables in rows j-1 and j+1, respectively. Note that h for row j-1 appears at iteration k+1 and h for row j+1 is at iteration level k. Both terms are known, either from the solution of equation (43) for the previous row (j-1) or from the previous iteration (k). Vector $\{Q^n\}_j$ is composed of source-sink terms and other

terms in (35) and (36) that are not functions of h_f or h_s.
Thus all terms on the right hand side of equation (43) are known and the equation can be solved for $\{h^{n+1}\}_j^{\tau}$. For improved accuracy, differences in head may be solved. Introducing

$$\{\Delta h^{n+1}\} = \{h^{n+1}\} - \{h^n\}, \tag{44}$$

equation (43) becomes

$$[A]_j\{\Delta h^{n+1}\}_j^{\tau} = \{Res^n\}_j - [B]_{j-1}\{\Delta h^{n+1}\}_{j-1}^{k+1}$$
$$-[B]_{j+1}\{\Delta h^{n+1}\}_{j+1}^{k}, \quad j = 1, \ldots, NY \tag{45}$$

where $\{Res^n\}_j$ is the residual of the difference equations at time level n. Iteration level τ is an intermediate level (between k and k+1) prior to over-relaxation. Over-relaxation is used to accelerate convergence and takes the form

$$\{\Delta h^{n+1}\}_j^{k+1} = \omega\{\Delta h^{n+1}\}_j^{\tau} + (1-\omega)\ \{\Delta h^{n+1}\}_j^{k} \tag{46}$$

where ω is a relaxation parameter ($0<\omega<2$). Values of ω are commonly determined by trial and are generally between 1.6 and 1.9. The solution of equation (45) followed by (46) for each row (in sequence) completes an iteration. Iterations are terminated when successive estimates of Δh are acceptably similar.

Well terms. For general problems, a well may penetrate both salt and fresh water, and pumping removes both fluids. The amount of each fluid removed is, in part, determined by the interface elevation in the well, because this determines what portion of the well's open interval is occupied by each fluid. As the interface elevation changes with changes in head (hence, with time) the relative quantities of fluids removed also changes.

A simple approximation for allocating the relative amounts of each fluid that are discharged is to linearly apportion the total discharge by the relative length of open interval that is exposed to each fluid.

In the model, these terms could be determined explicitly, that is, using the interface elevation in the well at the beginning of the time step (time level n) to determine the amount of each fluid removed. Because the interface elevation changes over the time step, this explicit approach gives mass balance errors that may become unstable (the errors grow rapidly with time) unless the time step is sufficiently small.

In order to obtain stable solutions without having a restrictive time step limitation, it is necessary to treat the well terms more implicitly, that is, evaluate them closer to the n+1 time level. One way to accomplish this is to incorporate the changes in fluid discharge with respect to salt-water and fresh-water head as follows:

$$Q_{wf}^{n+1} = Q_{wf}^{n} + \frac{\partial Q_{wf}^{n}}{\partial h_f} \Delta h_f + \frac{\partial Q_{wf}^{n}}{\partial h_s} \Delta h_s, \qquad (47a)$$

and

$$Q_{ws}^{n+1} = Q_{ws}^{n} + \frac{\partial Q_{ws}^{n}}{\partial h_f} \Delta h_f + \frac{\partial Q_{ws}^{n}}{\partial h_s} \Delta h_s, \qquad (47b)$$

where the derivatives are computed at the beginning of the time step and incorporated into the main diagonal block of appropriate equations (45). In order to compute the derivatives, consider figure 2, which shows a well penetrating both salt and fresh water. Note that both confined or water-table conditions are considered. For water-table conditions, it is assumed that the well is completely open to the top of the water table. Assuming that total flow contributions from salt and fresh water are determined linearly

$$QWF = (TWS - ZI)QT/LS \qquad (48a)$$

and

$$QWS = (ZI - WB)QT/LS, \qquad (48b)$$

where QT is the total well discharge, TWS = [αHF + (1-α)WT1] is the top of the well's screened or open interval, LS = TWS - WB is the total length of the screened or open interval, QWF = AQ_{wf}, and QWS = AQ_{ws}. Recall that $\alpha = 0$ implies confined conditions (TWS = WT1) and $\alpha = 1$ implies water-table or unconfined conditions (TWS = HF). Using equation (29) and substituting for the interface elevation (ZI) in equations (48) gives

$$QWF = (TWS - \rho_f^* HF + \rho_s^* HS)QT/LS \qquad (49a)$$

and

$$QWS = (\rho_f^* HF - \rho_s^* HS - WB)QT/LS \qquad (49b)$$

where HF and HS are the array names for fresh-water and salt-water heads, respectively, that are used in the FORTRAN computer program. Taking the derivatives gives

$$\frac{\partial QWF}{\partial HF} = \{LS(\alpha - \rho_f^*) - (TWS - \rho_f^* HF + \rho_s^* HS)\alpha\}QT/LS^2 \qquad (50a)$$

$$\frac{\partial QWF}{\partial HS} = \rho_s^* QT/LS, \qquad (50b)$$

and

$$\frac{\partial QWS}{\partial HF} = \{LS\rho_f^* - (\rho_f^* HF - \rho_s^* HS - WB)\alpha\}QT/LS^2 = -\frac{\partial QWF}{\partial HF} \qquad (51a)$$

$$\frac{\partial QWS}{\partial HS} = -\rho_s^* QT/LS = -\frac{\partial QWF}{\partial HS}\ . \qquad (51b)$$

The derivatives (equations (50) and (51)) are added to the diagonal blocks of $[A]_j$ in equation (45) and the known terms (Q_{wf}^n and Q_{ws}^n) become part of the residual ($\{Res^n\}_j$). Implicit treatment of well terms is discussed in more detail in Mercer and others [1980].

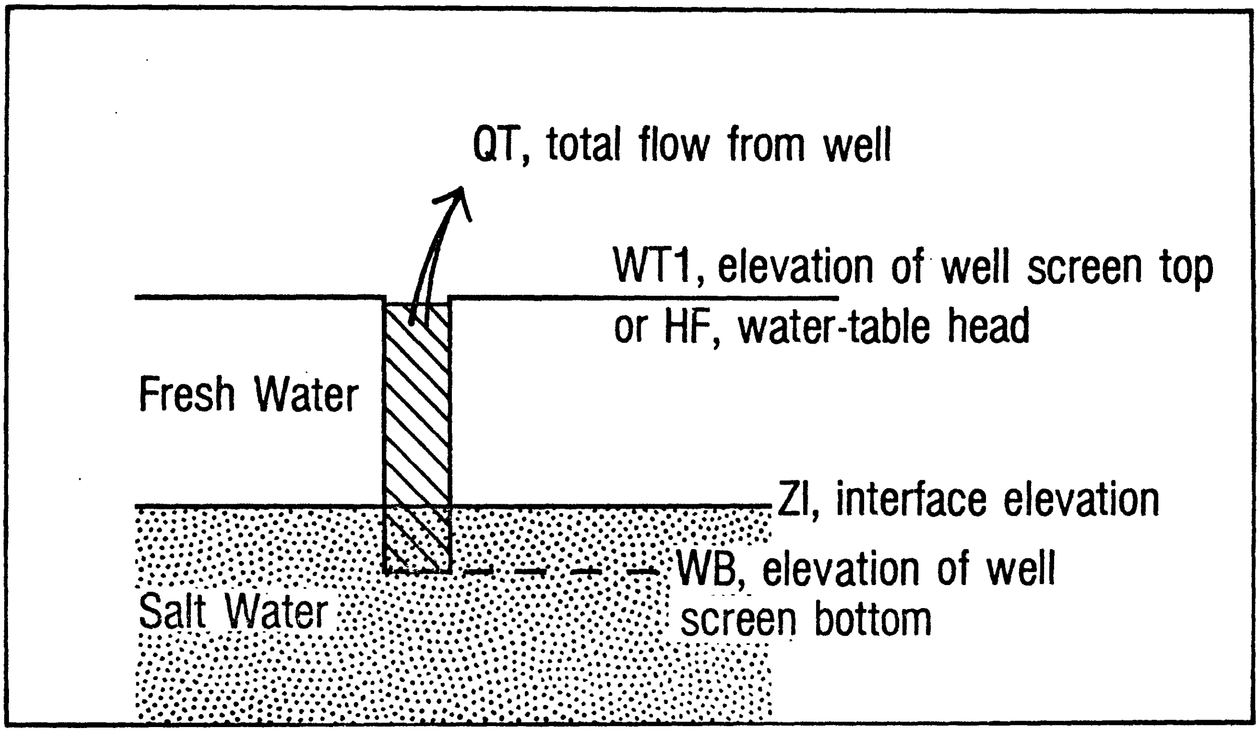

Figure 2. Well with opening to both salt water and fresh water.

MODEL DOCUMENTATION

Notes on use of program. The user may find the following comments helpful in understanding the program:

(1) Although calculations are performed in terms of salt-water and fresh-water heads, the initial input data required is fresh-water head and interface elevation. The reason being that these latter two variables are generally measured in a field situation. Salt water head is then calculated using equation (28).

(2) Well terms are considered positive for injection and negative for withdrawal. As indicated previously, the volume of salt water and fresh water removed (or injected) is determined linearly using the location of the interface and the length and position of the well's open interval. If one wishes to withdraw (or inject) only fresh water, the elevation of the well screen bottom must be set above the interface elevation. For withdrawal (or injection) of salt water only, the top of the well screen must be set below the interface elevation. Also, note that for unconfined conditions, the fresh-water head, HF, is assumed to be the top of the open interval. If this is not the case, adjustments should be made to the computer program.

(3) Both well terms and recharge are input in L^3/T units.

(4) For steady-state simulations, small values of specific storage and porosity can be used to increase the rate of interface translation and thus reduce the computational effort required to obtain the steady-state solution. There will be a limit, however, on how much these parameters can be reduced for a particular time step size.

If these parameters are chosen such that there is an attempt to translate the fresh-water tip or salt-water toe more than one grid location from their current position, the solution may become unstable. This condition can also produce instability in the well allocation scheme. If these conditions occur, the time step size must be decreased or the porosity (and possibly the specific storage) must be increased.

(5) Computer storage arrays must be sufficiently dimensioned (the code in the appendix is dimensioned for 34 columns and 32 rows). Note that variables AX, AY, BX, BY, TX, and TY require an extra row or column. To minimize core requirements, the dimensions of several arrays should be adjusted to their minimum values as follows

Array	Dimension
AF, AS	4 x NB
RF, RS, DELHF, DELHS, CI11, CI12, CI21, CI22	NB
NFE	NX + 1
RFT, RST	NX

where NB is the number of active grid points (equations) and NX is the number of columns in the grid. The total number of words required to store most of the program variables is approximately 48NB.

(6) The time required for the program to execute will depend upon the characteristics of the simulation (number of time steps, number of iterations required per step, etc.) and upon the type of computer being used. Some example processing times (per node per iteration) are: Cyber 175, 10^{-4} seconds; IBM 370/155, $6x10^{-4}$ seconds; and Harris S135, 10^{-3} seconds.

(7) The convergence criterion for iteration sequence is specified as 10^{-6}, (card A1620) that is, the maximum change in head during the iteration must be less than 10^{-6} for convergence. This number was found to be satisfactory for test problems, but if necessary, this value may easily be changed by repunching the card where this check is perfomed.

(8) To improve convergence, an over-relaxation factor, WØ, (see equation 46, p. 22) is read. This value may be varied by trial and error to determine the value that gives optimal convergence rate. In most problems, the optimal value was found to range between 1.6 to 1.9.

(9) A weighting factor, WT, (see equation 40, p. 20) is also read that weights the thicknesses of the salt and fresh water used in the transmissive terms between blocks. A value of 1.0 weights the thickness using the value from the upstream block, whereas a value of 0.5 weights the two blocks equally (midpoint or central weighting). The upstream weighting gives a less accurate solution, but may be more stable for certain problems.

(10) To reduce the number of lines in the output, only computed fresh-water head and interface elevation are printed every ITP time step, where ITP is read. If salt-water head is also desired, IPT must be input as 1.0. In addition, if the matrix information is desired, ID4 must be input as 1.0.

(11) The time step may either be uniform (IDELT = 1.0) or increase with time (IDELT = 0.0). The increase is computed automatically in MAIN using ZMAX, which is input with IDELT. The program allows a maximum increase factor of 1.2. To change this restriction, the appropriate cards must be repunched (cards A570 and A600).

(12) The program uses feet and seconds as the units for labelling output. However, any consistent set of units may be used if the output is interpreted appropriately.

Description of the subroutines. The FORTRAN IV code contains a main program and 8 subroutines, which are shown diagrammatically in figure 3. The purpose of each subroutine is listed below.

MAIN Driving program for the subroutines. In addition, the final space coefficients are computed and both the time loop and iteration loop are executed.

GDATA Reads and writes problem information according to the formats listed in the input section of this report.

READ Reads the two dimensional arrays containing data for each finite-difference block.

TCALC Computes harmonic part of transmissive terms. Also initializes boundary transmissive terms to zero.

CØEF Computes the thickness part of the transmissive terms and the time derivative coefficients.

FØRMEQ Forms the components of the reduced matrix equations for the iterative scheme.

BSBAND Solves reduced matrix equation for one row using the Gauss-Doolittle method.

PDATA Prints time step information and computed salt-water head, fresh-water head and interface elevation, if desired.

BAL Computes and prints salt and fresh water mass balance.

A generalized flow chart showing the approximate order that the subroutines are used is shown in figure 4.

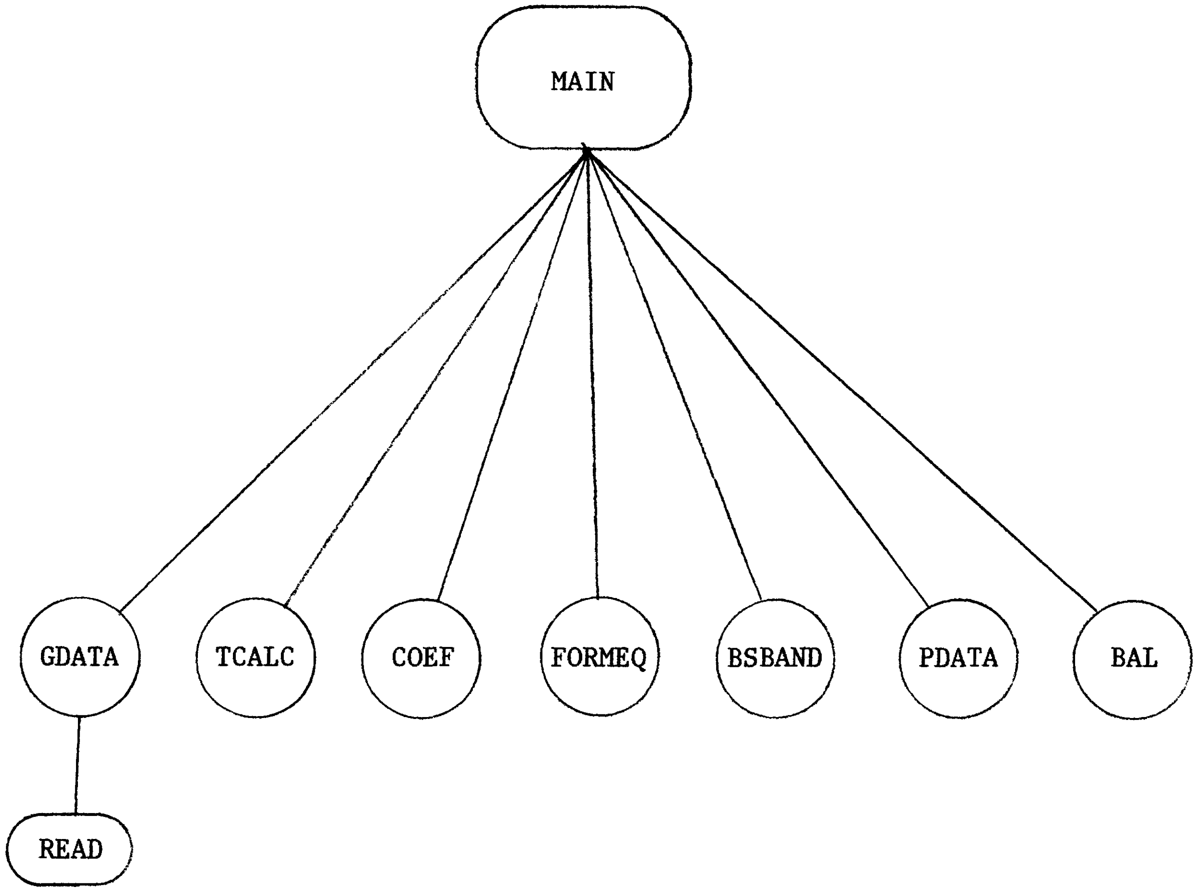

Figure 3. Program subroutines showing order and link of calling.

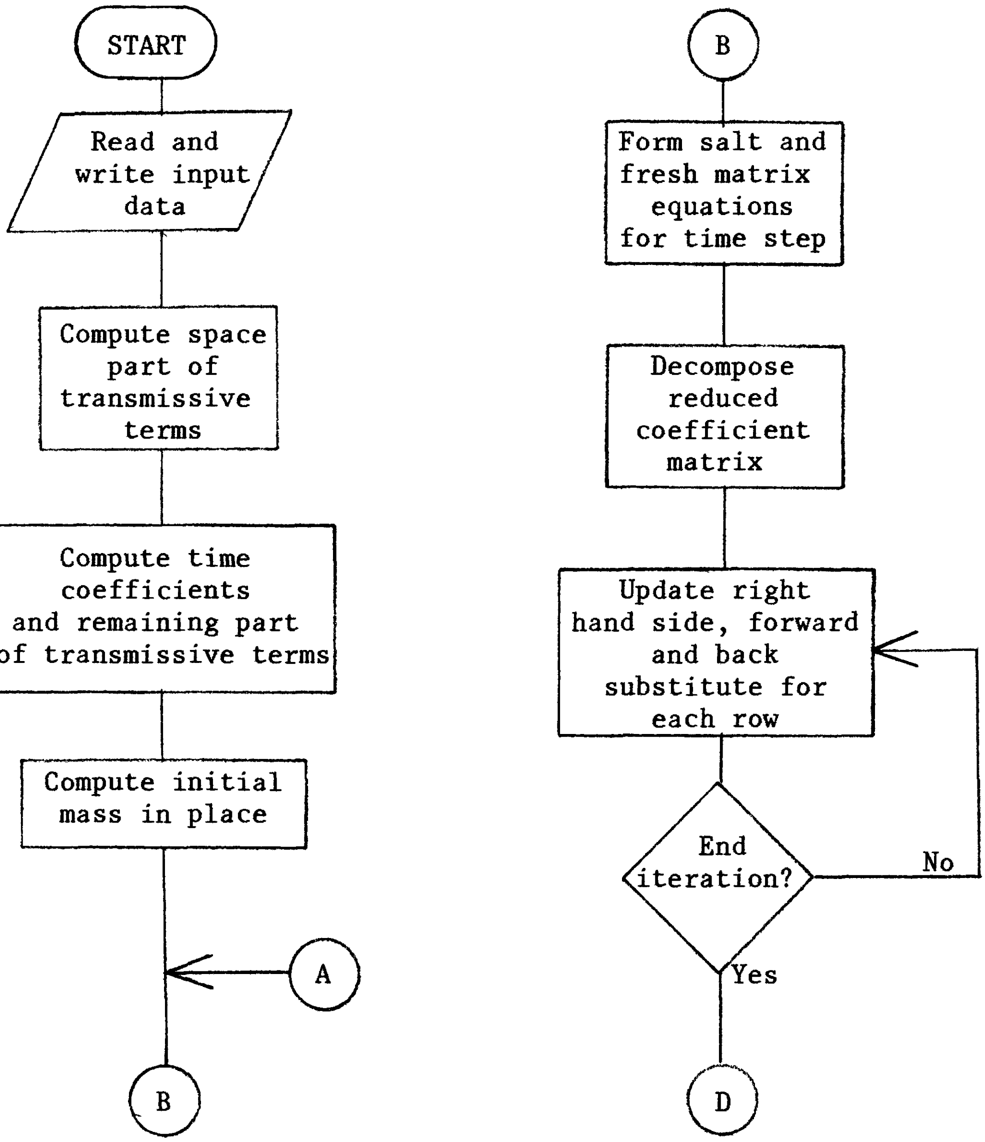

Figure 4. Generalized flow chart.

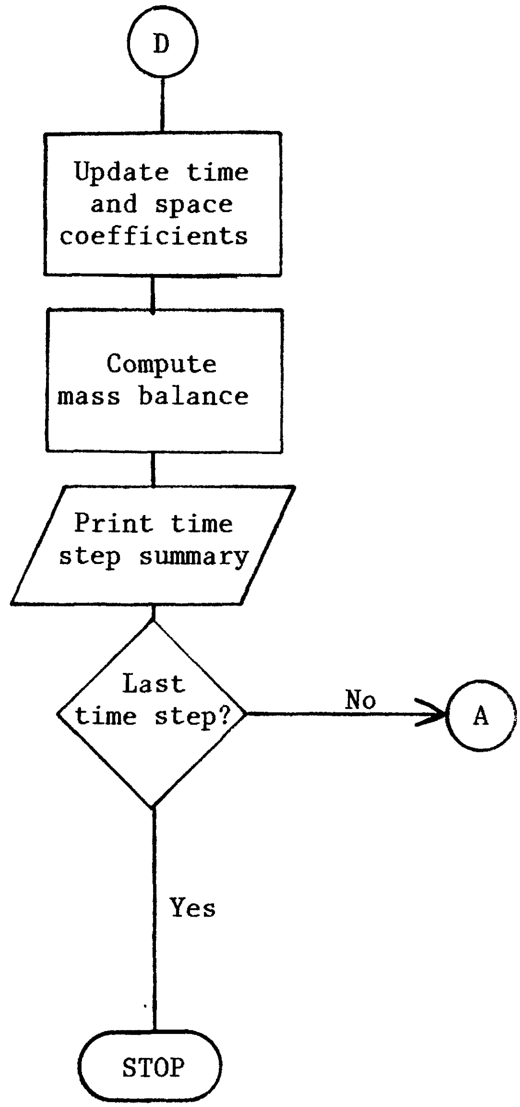

Figure 4. Generalized flow chart (cont.).

Input

Columns	Format	Name	Description
Card 1 - Title			
1 - 60	20A3	TITLE	Problem title
Card 2 - Finite difference data			
1-4	I4	NX	Number of columns (x-direction)
5-8	I4	NY	Number of rows (y-direction)
9-12	I4	NB	Number of active (nonzero porosity) grid blocks
13-16	I4	MBE	Not used.
17-20	I4	NT	Number of time steps
21-24	I4	NF	Number of sources and sinks
25-28	I4	NHSC	Number of constant salt-water head nodes
29-32	I4	NHFC	Number of constant fresh-water head nodes
33-36	I4	NIT	Maximum number of iterations allowed
Card 3 - Time parameters			
1-10	G10.0	DELT	Initial time step (seconds)
11-20	G10.0	WT	Weighting factor (1.0 for upstream)
21-30	G10.0	WØ	Over-relaxation factor

Columns	Format	Name	Description
Card 4			
1-10	G10.0	ALPHA	Read 1.0 for water-table; 0.0 for confined
11-14	I4	IPT	Read 1 to print salt-water heads
15-18	I4	ILK	Read 1 for leakage
19-22	I4	ITP	Number of time steps between printing output
23-26	I4	ID4	Read 1 to print matrix information
Card 5 - Data codes			
1-4	I4	K1	Read 1 if x-spacing is uniform
5-8	I4	K2	Read 1 if y-spacing is uniform
9-12	I4	K3	Read 1 if initial fresh-water head is uniform
13-16	I4	K4	Read 1 if initial interface elevation is uniform
17-20	I4	K5	Read 1 if x-hydraulic conductivity is uniform
21-24	I4	K6	Read 1 if y-hydraulic conductivity is uniform
25-28	I4	K7	Read 1 if specific storage is uniform
29-32	I4	K8	Read 1 if elevation of aquifer base is uniform
33-36	I4	K9	Read 1 if porosity is uniform
37-40	I4	K10	Read 1 if aquifer thickness is uniform
Card 6 - Multiplication factors			
1-10	G10.0	F1	Multiplication factor for x-spacing
11-20	G10.0	F2	Multiplication factor for y-spacing

Columns	Format	Name	Description
21-30	G10.0	F3	Multiplication factor for initial fresh-water head
31-40	G10.0	F4	Multiplication factor for initial interface elevation
41-50	G10.0	F5	Multiplication factor for x-hydraulic conductivity
51-60	G10.0	F6	Multiplication factor for y-hydraulic conductivity
61-70	G10.0	F7	Multiplication factor for specific storage
71-80	G10.0	F8	Multiplication factor for elevation of aquifer base

Card 7 - Multiplication factors (continued)

1-10	G10.0	F9	Multiplication factor for porosity
11-20	G10.0	F10	Multiplication factor for aquifer thickness

Data set 1 - X-spacing

1-80	8G10.0	DX(I)	Spacing in the x-direction (NX values); if uniform, K1 = 1 and only read one value

Data set 2 - Y-spacing

1-80	8G10.0	DY(J)	Spacing in the y-direction (NY values); if uniform, K2 = 1 and only read one value

Data set 3 - Initial fresh water heads[1/]

1-80	8G10.0	HF(I,J)	Initial fresh water heads; if uniform, K3 = 1 and only read one value

1/ Start new card for the beginning of each new row leaving blanks for missing blocks.

Columns	Format	Name	Description
Data Set 4 - Initial interface elevation[1/]			
1-80	8G10.0	ZI(I,J)	Initial interface elevation; if uniform, K4 = 1 and only read one value
Data set 5 - X-hydraulic conductivity[1/]			
1-80	8G10.0	XK(I,J)	X-hydraulic conductivity; if uniform, K5 = 1 and only read one value
Data set 6 - Y-hydraulic conductivity[1/]			
1-80	8G10.0	YK(I,J)	Y-hydraulic conductivity; if uniform, K6 = 1 and only read one value
Data set 7 - Specific storage[1/]			
1-80	8G10.0	S(I,J)	Specific storage; if uniform, K7 = 1 and only read one value
Data set 8 - Elevation of aquifer base[1/]			
1-80	8G10.0	ZB(I,J)	Elevation of aquifer base; if uniform, K8 = 1 and only read one value

1/ Start new card for the beginning of each new row leaving blanks for missing blocks.

Columns	Format	Name	Description
Data set 9 - Porosity[1/]			
1-80	8G10.0	PØR(I,J)	Porosity; if uniform, K9 = 1 and only read one value. Note, porosity is used to key in on inactive blocks - a porosity of zero must be read for these blocks.
Data set 10 - Aquifer thickness[1/] [2/]			
1-80	8G10.0	TH(I,J)	Aquifer thickness; if uniform, K10 = 1 and only read one value
Card 8 Data codes (continued)[3/]			
1-4	I4	K11	Read 1 if head in overlying aquifer is uniform
5-8	I4	K12	Read 1 if leakance (K'/b') is uniform
Card 9 Multiplication factors (continued)[3/]			
1-10	G10.0	F11	Multiplication factor for confining bed head
11-20	G10.0	F12	Multiplication factor for confining bed leakance

1/ Start new card for the beginning of each new row leaving blanks for missing blocks.
2/ Read only if ALPHA = 0.0, confined conditions.
3/ Read only if ALPHA = 0.0 and ILK = 1 (card 4), indicating that leakage is included.

Columns	Format	Name	Description
Data set 11 - Head in overlying aquifer[1/] [2/]			
1-80	8G10.0	BH(I,J)	Head in overlying aquifer; if uniform, K11 = 1 and only read one value
Data set 12 - Confining bed leakance[1/] [2/]			
1-80	8G10.0	BL(I,J)	Confining bed leakance (K'/b'); if uniform, K12 = 1 and only read one value
Card 10 - Densities and viscosities			
1-10	G10.0	DENF	Fresh-water density
11-20	G10.0	DENS	Salt-water density
21-30	G10.0	VF	Fresh-water viscosity
31-40	G10.0	VS	Salt-water viscosity
Card 11 - Recharge			
1-10	G10.0	RECHG	Recharge (uniform value applied to each node)

1/ Read only if ALPHA = 0.0 and ILK = 1(card 4), indicating that leakage is included.
2/ Start new card for the beginning of each new row leaving blanks for missing blocks.

Columns	Format	Name	Description

Data set 13 - Source/sink rates

1-4	I4	I	Column number of well
5-8	I4	J	Row number of well
9-18	G10.0	QT(I,J)	Volumetric rate of source/sink at block i,j (negative for a sink)
19-28	G10.0	WT1(I,J)	Elevation of top of well screen
29-38	G10.0	WB(I,J)	Elevation of bottom of well screen

Note: NF cards read; if NF = 0, this data set is omitted.

Card 12

1-4 I4	IDELT		Read 1 for constant time step; 0 for auto time step.
8-14	G10.0	ZMAX	Maximum change in interface elevation used to compute auto time step

Data set 14 - Constant salt water head nodes

1-4	I4	I	Column number of constant salt-water head node
5-8	I4	J	Row number of constant salt-water head node

Note: NHSC cards read; if NHSC = 0, this data set is omitted.

Data set 15 - Constant fresh water head nodes

1-4	I4	I	Column number of constant fresh-water head node
5-8	I4	J	Row number of constant fresh-water head node

Note: NHFC cards read; if NHFC = 0, this data set is omitted.

Output

To aid the user in detecting errors associated with data input, data read by the program is immediately printed out; thus, output appears in the following order:

Title of Problem

Finite Difference Data

Time Parameters

Multiplication Factors

Spacing in X-Direction

Spacing in Y-Direction

Initial Fresh-water Heads 1/

Initial Interface Elevation 1/

X-Hydraulic Conductivity 1/

Y-Hydraulic Conductivity 1/

Specific Storage 1/

Elevation of Aquifer Base 1/

Porosity 1/

Aquifer Thickness 1/ 2/

Multiplication Factors 1/ 3/

Confining Bed Head 1/ 3/

1/ Printed by rows with row number printed to left of data; if uniform, only one value is printed.
2/ For confined conditions
3/ For leakage and confined conditions

Confining Bed Leakance 1/ 2/

Densities and Viscosities

Recharge

Source/Sink Values and Locations

Constant Salt-water Head Locations

Constant Fresh-water Head Locations

Grid Block Numbers 3/

Initial Volume of Fresh Water and Salt Water in Place

In addition, for every time step, the following is printed:

Iteration Information

Fresh- and Salt-water Balances

Time Step Information

Finally, for every ITP time step, the following is also printed:

Fresh-water Head

Interface Elevation

Salt-water Head 4/

Water Quality Map

1/ Printed by rows with row number printed to left of data; if uniform, only one value is printed.
2/ For leakage and confined conditions
3/ If ID4 equals one
4/ If IPT equals one

Iteration information. The maximum difference error for each iteration is printed. This information gives an indication of the rate of convergence.

Fresh-water and Salt-water Balance information. The following rates and total volumes (for each time step) are given for both fresh and salt water:

Change in Storage

Source/Sink

Constant Head Nodes

Vertical Leakage

Recharge

Error, in percent

Totals

Equation Change in Storage

The rates and totals are indicated as being in the units ft^3/s and ft^3, respectively. In actuality, these are printed in whatever consistent units were used for the input. The error, in percent, is based on a total net inflow volume. The total value is simply the difference between all inflows and outflows.

The mass balance summary includes two values of change in storage: the true change in storage and the change in storage expressed by the equations. These are given to help assess mass balance problems along the fresh-water tip or salt-water toe (Mercer and others, 1980). The equation change in storage should always balance with the net of the other inflow and outflow terms. This indicates that the iterative method has successfully solved the algebraic equations. The mass balance error, however, is computed using the true or physical change in storage and may be significant even if the equations have been properly solved. This error reflects the effect of physically unrealistic interface conditions along the fresh-water tip or salt-water toe. A more complete discussion of this problem is given in Mercer and others (1980).

APPLICATIONS

To verify that the finite-difference approach is adequately approximating the partial differential equations, a comparison is made with an analytical solution. To test some of the assumptions that compose our mathematical model, a laboratory experiment is also simulated. In addition to verifying our approach, these applications serve as examples of how to set up problems and use the computer program.

Example 1

Consider a confined aquifer of uniform thickness, D (see figure 5). At time t = 0 there is a vertical interface, maintained by a gate at x = 0, separating salt water from fresh water. At t = 0 the gate is removed, and the interface begins to move owing to the density difference. After time $t = t_1$, the toe has moved a distance of $L\ (t_1)$. Keulegan (1954) gives an analytical solution for the motion of the interface, z_I^*, and for the location of the toe, L(t):

$$z_I^*\ (x,t) = \tfrac{1}{2}D\{1+x/[(\Delta\rho K_f Dt)/(\phi\rho_f)]^{\frac{1}{2}}\}, \tag{52}$$

and

$$L(t) = (t\Delta\rho K_f D/\phi\rho_f)^{\frac{1}{2}}, \tag{53}$$

where $\Delta\rho = \rho_s - \rho_f$. The data for this problem are given in table 1 and the formated model data are given in table 2.

The numerical solution was started with a linear interface ending at $L(t_o)$ = 20 meters (following Shamir and Dagan, 1971), which was then allowed to move. This value of L corresponds to t_o = 12.28 days. The numerical model used for this problem consisted of 20 finite-difference blocks, with a spacing of 5.0 meters each. Backward difference was used for the time derivative and a constant time step of one day was specified. The program was run for 20 days using both midpoint and upstream weighting. The results for t = 32.28 days are shown in figure 6. As may be seen, the results compare well, with upstream weighting showing slightly more smearing at the top and base of the aquifer.

Example 2

To test the vertical integration or Dupuit assumption, we consider a Hele-Shaw experiment performed by Bear and Dagan (1964). In these experiments, the interface began in a fixed position maintained by a constant fresh-water recharge at the left end (fig. 7). At t = 0 the recharge was abruptly increased to a new constant value. The length of the experimental apparatus was 400 cm and other pertinent data are given in table 3.

For this problem, backward difference is used for the time derivative and the time step was a constant 5 seconds. The spacing was 10 cm and midpoint weighting was used. The upstream boundary condition indicated by Bear and Dagan is one of constant flux (18.8 cm^2/s) for $t > 0$. However, it was assumed that a constant head tank was probably used to produce this flux. In an attempt to simulate the experiment more precisely, the steady-state head at x = 200 cm for a flux of 18.8

cm^2/s was computed by the model. This head value was used in transient simulations as the upstream boundary condition at x = 200 cm. Only 200 cm of the experimental apparatus was considered because the salt-water wedge is contained within this region during the entire experiment. At the downstream (right) end, the seepage face was approximated using a leakance and head that were computed to give the correct interface location at the initial and final flow conditions and assuming the downstream salt-water head was constant at 27 cm.

Using these boundary conditions and the formated data in table 4, locations of the interface were computed for each 5-second time interval. Comparison with the results of Bear and Dagan are shown in figure 8. The line labeled t = 0 is the steady-state result that is used as initial conditions. The computed initial condition for the interface is in close agreement with the observed and thus it appears that the Dupuit approximation is valid for this interface shape. As the simulation progresses, however, the computed interface lags behind the observed interface. Shamir and Dagan (1971) also simulated this experiment, with computed results virtually identical to those presented in this paper.

The discrepancy between computed and observed could be the result of several factors: 1) error introduced from the Dupuit assumption; 2) error introduced in boundary condition approximations (that is, the way the leakance and head values are assigned); and 3) not knowing the exact experimental procedures.

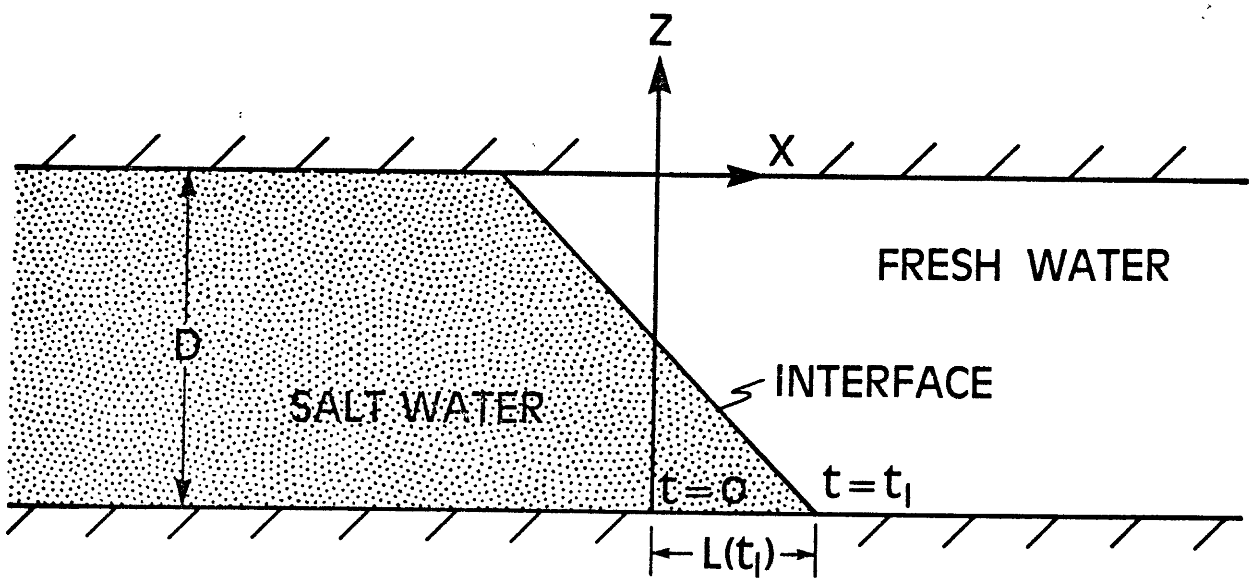

Figure 5. Diagram of linear interface in example 1.

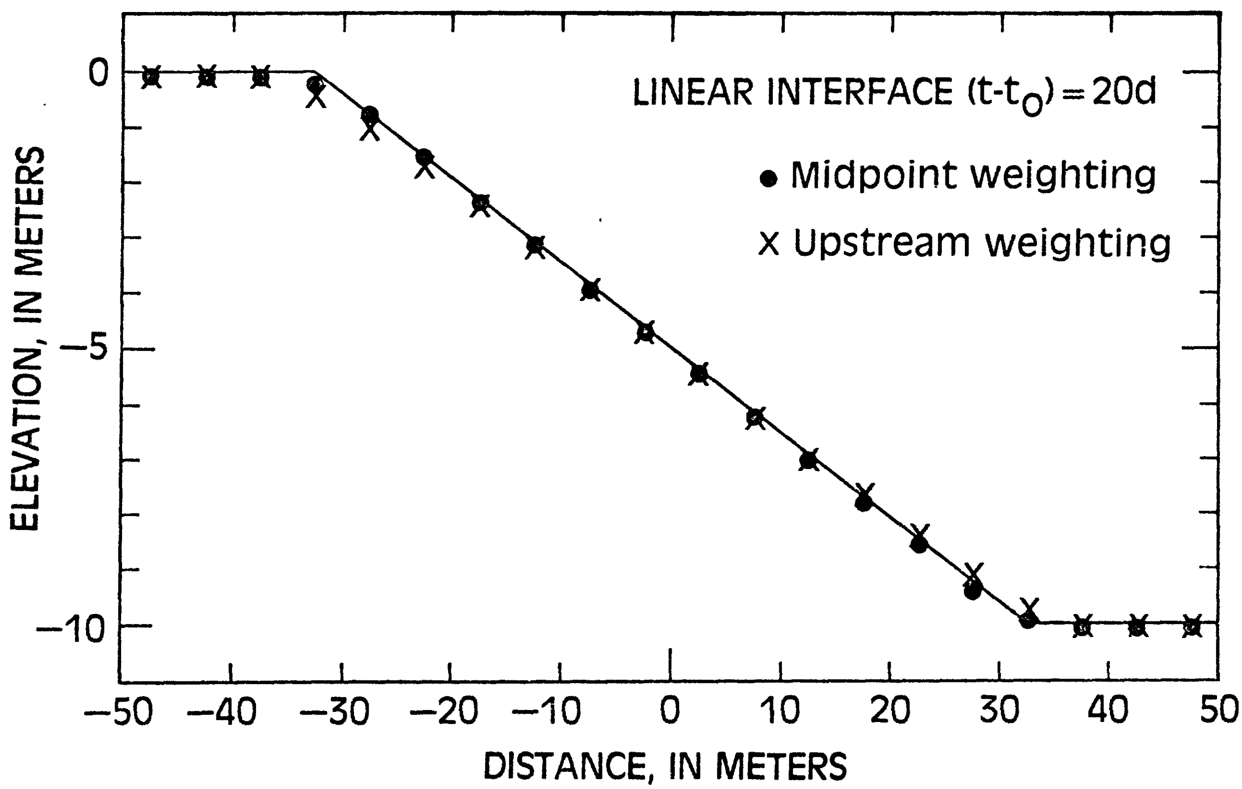

Figure 6. Computed results for midpoint and upstream weighting compared with Keulegan's analytical solution for example 1.

Table 1. Data for example 1 (linear interface).

D	$= 10$ m	K_f	$= 39.024$ m/d
ρ_f	$= 1.0$ g/cm^3	Φ	$= 0.3$
ρ_s	$= 1.025$ g/cm^3		

Initial conditions: $L(t_o) = 20$ m

$t_o = 12.28$ d

Table 2. Formated data for example 1.

```
CARD 1..........      TEST USING KEULEGAN'S ANALYTICAL SOLUTION (FINITE DIFFERENCE)
CARD 2.......... 20   1  20   7  20   0   0   0 150
CARD 3..........   86400.      1.0     1.0
CARD 4..........      1.0   1   0    1   1
CARD 5..........  1   1   0   0   1   1   1   1   1   1
CARD 6..........      1.0       1.0       1.0       1.0       1.0       1.0       1.0       1.0
CARD 7..........      1.0       1.0
DATA SET 1......      5.0
DATA SET 2......      5.0
DATA SET 3......   .06250    .06250    .06250    .06250    .06250    .06250    .06322    .06712
           .....   .07493    .08665    .10085    .11257    .12038    .12428    .12500    .12500
           .....   .12500    .12500    .12500    .12500
DATA SET 4......    0.000     0.000     0.000     0.000     0.000     0.000    -0.625    -1.875
           .....   -3.125    -4.375    -5.625    -6.875    -8.125    -9.375   -10.000   -10.000
           .....  -10.000   -10.000   -10.000   -10.000
DATA SET 5......  .00045167
DATA SET 6......  .00045167
DATA SET 7......    .0001
DATA SET 8......    -10.0
DATA SET 9......      0.3
CARD 10.........      1.0     1.025       1.0       1.0
CARD 11.........      0.0
CARD 12.........  1         0.0
```

Figure 7. Diagram of Hele-Shaw experiment of Bear and Dagan (1964).

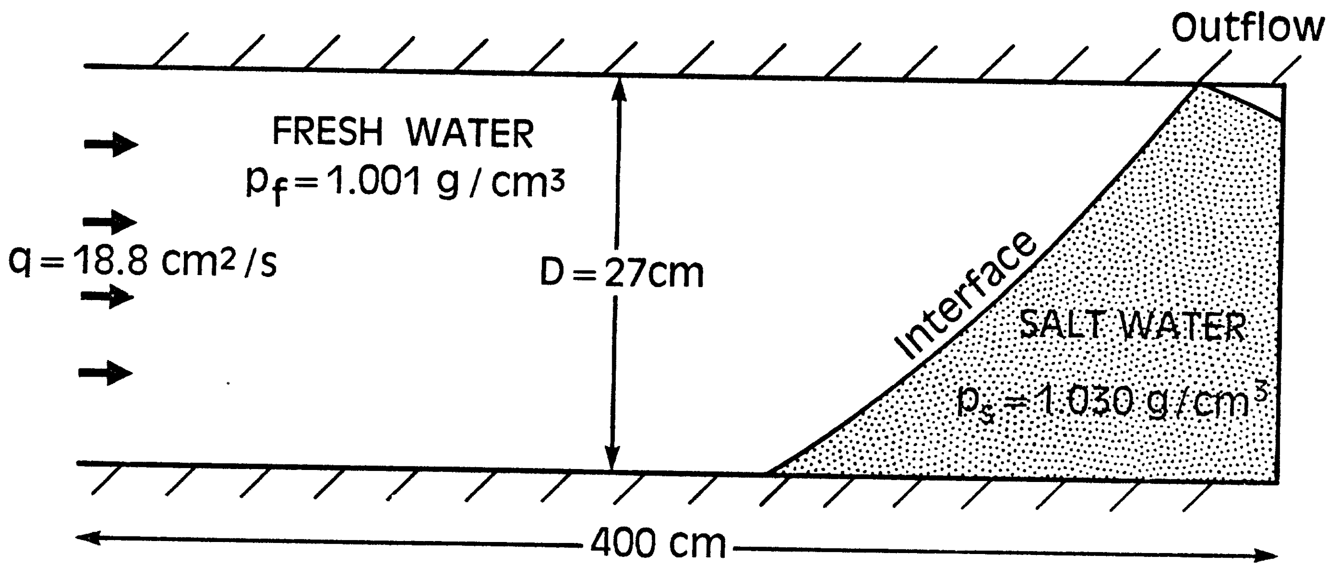

Table 3. Data for example 2 (Hele-Shaw experiment).

K_f = 69 cm/s	ρ_f = 1.001 g/cm^3
ϕ = 1.0	ρ_s = 1.030 g/cm^3
D = 27 cm	

For $t < 0$, q_{fb} = 3.9 cm^2/s

For $t \geqq 0$, q_{fb} = 18.8 cm^2/s

$h_f(x,0)$ and $z(x,0)$, computed steady state for q_{fb} = 3.9 cm^2/s.

Table 4. Formated data for example 2.

```
CARD 1..........      HELE SHAW EXPERIMENTS        TRANSIENT
CARD 2..........   20   1  20   2   8   0   1   1 150
CARD 3..........        5.          .5          1.0
CARD 4..........        0.    1   1
CARD 5..........    1   1   0   0   1   1   1   1   1   1
CARD 6..........       1.0         1.0         1.0         1.0         1.0         1.0         1.0         1.0
CARD 7..........       1.0         1.0
DATA SET 1.....       10.0
DATA SET 2.....        1.0
DATA SET 3.....     29.496      27.790      27.769      27.748      27.727      27.704      27.681      27.658
          .....     27.633      27.607      27.581      27.552      27.523      27.491      27.458      27.421
          .....     27.381      27.337      27.285      27.222
DATA SET 4.....         0.          0.       .4304      1.1592      1.9027      2.6693      3.4617      4.2820
          .....     5.1334      6.0200      6.9471      7.9200      8.9471     10.0380      11.204      12.463
          .....     13.841      15.381      17.161      19.346
DATA SET 5.....        69.
DATA SET 6.....        69.
DATA SET 7.....    1.E-06
DATA SET 8.....         0.
DATA SET 9.....        1.0
DATA SET 10....        27.
CARD 8..........    1   0
CARD 9..........       1.0         1.0
DATA SET 11....     27.136
DATA SET 12....         0.          0.          0.          0.          0.          0.          0.          0.
           ....         0.          0.          0.          0.          0.          0.          0.          0.
           ....         0.          0.          0.      4.6087
CARD 10.........     1.001       1.030         1.0         1.0
CARD 11.........       0.0
CARD 12.........    1
DATA SET 14....    20   1
DATA SET 15....     1   1
```

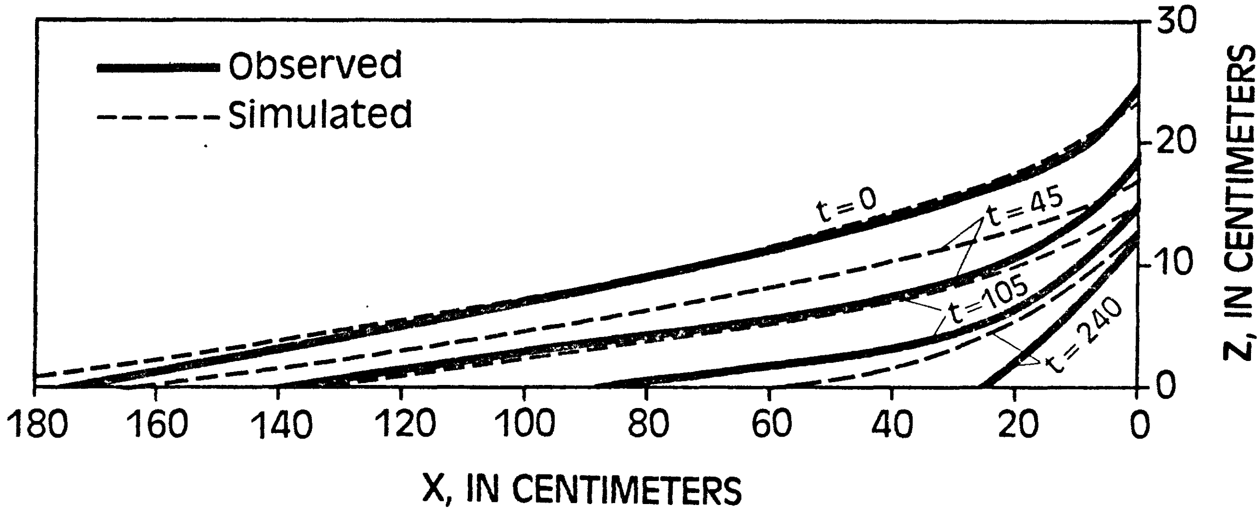

Figure 8. Computed results compared to Hele-Shaw data in example 2.

NOTATION

Parameter	Coded Name	Description
b	TH	aquifer thickness
h_f	HF	fresh-water head
h_s	HS	salt-water head
h'	BH	head in overlying aquifer
K'/b'	BL	leakance
K_{fx}	XK	x-hydraulic conductivity
K_{fy}	YK	y-hydraulic conductivity
-	QT	total discharge
R	RECHG	recharge
S_f	S	specific storage
WB	WB	well bottom
WT1	WT1	well top
z_b	ZB	elevation of aquifer base
z_I	ZI	interface elevation
α	ALPHA	confined/water-table indicator
β	WT	upstream/mid-point weighting indicator
μ_f	VF	fresh-water viscosity
μ_s	VS	salt-water viscosity
ρ_f	DENF	fresh-water density
ρ_s	DENS	salt-water density
ω	WØ	relaxation parameter
ϕ	PØR	porosity

REFERENCES

Bear, J., and Dagan, G., 1964, Moving interface in coastal aquifers: American Society of Civil Engineers, Journal of the Hydraulics Division v. 90, no. HY4, p. 193-216.

Bonnet, M., and Sauty J.P., 1975, Un modele simplfie pour la simulation des nappes avec intrusion saline: A.I.H.S. Symposium-Application of Mathematical Models in Hydrology and Water Resources Systems, Bratislava, AIHS Publication No. 115, 9 p.

Faust, C. R., and Mercer, J. W., 1979, Geothermal reservoir simulation 1. Mathematical models for liquid- and vapor-dominated hydrothermal systems: Water Resources Research, v. 15, no. 1, p. 23-30.

Hantush, M. S., 1964, Hydraulics of wells: in Advances in Hydroscience, V.T. Chow, ed., New York, Academic Press, p. 281-432.

Keulegan, H. G., 1954, An example report on model laws for density currents: U.S. National Bureau of Standards, Gaithersburg, Maryland, 16 p.

Korn, G. A., and Korn, T. M., 1961, Mathematical handbook for scientists and engineers: New York, McGraw-Hill, p. 100.

Mercer, J. W., Larson, S. P., and Faust, C. R., 1980, Simulation of salt-water interface motion: Groundwater, v. 18, no. 4.

Pinder, G. F., and Page, R. H., 1976, Finite element simulation of salt water intrusion on the South Fork of Long Island: Proceedings of the First International Conference on Finite Elements in Water Resources, Princeton University, p. 251-270.

Shamir, V., and Dagan, G., 1971, Motion of the seawater interface in coastal aquifers: a numerical solution: Water Resources Research, v. 7 no.3, 644-657.

Varga, R. S., 1962, Matrix iterative analysis: New Jersey, Prentice-Hall, 322 p.

Program listing and sample output

```
C     MAIN PROGRAM
C     ************
C
C     PURPOSE:  TO SIMULATE A FRESH WATER-SALT WATER INTERFACE OVER AN
C               AREAL DOMAIN USING FINITE-DIFFERENCE TECHNIQUES
C     THERE MAY BE PARTS OF THIS PROGRAM THAT ARE NOT THOROUGHLY
C     DEBUGGED;  THE AUTHORS AND THE GEOLOGICAL SURVEY WILL TAKE CREDIT
C     FOR ANY SUCCESSFUL RUNS OF THE MODEL, BUT THE USER ASSUMES FULL
C     RESPONSIBILITY FOR ALL RUNS THAT BOMB!
C     ------------------------------------------------------------------
C
      COMMON /LGE/ AF(4352),RF(1088),AS(4352),RS(1088),CI11(1088),CI12(1
     1088),CI21(1088),CI22(1088)
C
      COMMON /INPUT/ S(34,32),XK(34,32),YK(34,32),HF(34,32),HS(34,32),DX
     1(34),DY(32),ZB(34,32),ZI(34,32),QF(34,32),QT(34,32),POR(34,32),TH(
     234,32),NP(34,32)
      COMMON /CONTRO/ TIME,DELT,DENS,DENF,VS,VF,WT,ZMAX,ALPHA,WO,NK,NX,N
     1Y,NXX,NYY,NB,NBB,NT,MBW,MBE,IPT,NIT,PECHG
      COMMON /WORK/ AX(35,32),BX(35,32),AY(34,33),BY(34,33),TX(35,32),TY
     1(34,33),F(34,32),B(34,32),C(34,32),D(34,32),FT(34,32),ST(34,32),BH
     2(34,32),BL(34,32),NHS(34,32),NHF(34,32),DQF(34,32),DQS(34,32),WT1(
     334,32),WB(34,32)
      DIMENSION DELHF(1088), DELHS(1088), ERR(150), NFE(41), RFT(40), RS
     1T(40)
C     ------------------------------------------------------------------
C
C     READ AND WRITE PROBLEM DATA -
      CALL GDATA(IDELT,ITP,NFE)
C     ************************
C
C     CALCULATE TRANSMISSIBILITY TERMS -
      CALL TCALC(DY,DX,POR,XK,YK,TY,TX,NY,NX,NYY,NXX)
C     **********************************************
C
C     COMPUTE THE REST OF THE SPACE COEFFICIENT AND
C     TIME COEFFICIENTS -
      CALL COEF(1,IDELT,DZMX)
C     **********************
      DO 20 K=1,NB
      DELHF(K)=0.
   20 DELHS(K)=0.
C
C    CALCULATE INITIAL MASS IN PLACE
      CALL BAL(0,DELHF,DELHS)
C     **********************
C
C
C  BEGIN TIME LOOP
      DO 190 LT=1,NT
C
C     FOR INCREASING TIME STEP -
      IF (IDELT.EQ.1) GO TO 40
      IF (DZMX.LT.0.E0) GO TO 40
      IF (DZMX.EQ.0.E0) GO TO 30
```

```
      FAC=ZMAX/DZMX
      IF (FAC.GT.1.2) FAC=1.2
      DELT=DELT*FAC
      GO TO 40
   30 FAC=1.2
      DELT=DELT*FAC
   40 CONTINUE
C
C     INITIALIZE BOUNDARY ARRAYS -
      DO 50 I=1,NX
      AY(I,1)=0.E0
      AY(I,NYY)=0.E0
      BY(I,1)=0.E0
   50 BY(I,NYY)=0.E0
C
      DO 60 J=1,NY
      AX(1,J)=0.E0
      BX(1,J)=0.E0
      AX(NXX,J)=0.E0
   60 BX(NXX,J)=0.E0
C
C     COMPUTE FINAL SPACE COEFFICIENTS -
      DO 70 J=1,NY
      DO 70 I=2,NX
      ALS=WT
      ALD=WT
      IF (HF(I-1,J).GT.HF(I,J)) ALD=1.E0-WT
      IF (HS(I-1,J).GT.HS(I,J)) ALS=1.E0-WT
      AX(I,J)=(ALD*FT(I,J)+(1.E0-ALD)*FT(I-1,J))*TX(I,J)
   70 BX(I,J)=(ALS*ST(I,J)+(1.E0-ALS)*ST(I-1,J))*TX(I,J)
C
      DO 80 I=1,NX
      DO 80 J=2,NY
      ALS=WT
      ALD=WT
      IF (HF(I,J-1).GT.HF(I,J)) ALD=1.E0-WT
      IF (HS(I,J-1).GT.HS(I,J)) ALS=1.E0-WT
      AY(I,J)=(ALD*FT(I,J)+(1.E0-ALD)*FT(I,J-1))*TY(I,J)
   80 BY(I,J)=(ALS*ST(I,J)+(1.E0-ALS)*ST(I,J-1))*TY(I,J)
C
C     FORM FINAL MATRIX EQUATION -
      CALL FORMEQ(AF,AS,RF,RS)
C     ************************
C   FACTOR MATRICES FOR EACH ROW
      DO 90 J=1,NY
      N=NFE(J+1)-NFE(J)
      IF (N.LE.1) GO TO 90
      ID=4*NFE(J)-3
      K=NFE(J)-1
C
C    FACTOR ROW J
      CALL BSBAND(AF(ID),AS(ID),RFT,RST,CI11,CI12,CI21,CI22,N,K,1)
   90 CONTINUE
C     ************************************************************
C   BEGIN ITERATIONS
```

```
      DO 160 KK=1,NIT
      ERR(KK)=0.
      ERMAX=0.
      DO 150 J=1,NY
      K=NFE(J)-1
      N=NFE(J+1)-NFE(J)
      IF (N.EQ.0) GO TO 150
      ID=4*NFE(J)-3
C  UPDATE RIGHT HAND SIDE
      L=0
      DO 120 I=1,NX
      IF (POR(I,J).LE.0.) GO TO 120
      L=L+1
      KL=K+L
      RFT(L)=RF(KL)
      RST(L)=RS(KL)
      IF (J.EQ.1) GO TO 100
      NJM=NP(I,J-1)
      IF (NJM.EQ.0) GO TO 100
      RFT(L)=RFT(L)-AY(I,J)*DELHF(NJM)
      RST(L)=RST(L)-BY(I,J)*DELHS(NJM)
  100 IF (J.EQ.NY) GO TO 110
      NJP=NP(I,J+1)
      IF (NJP.EQ.0) GO TO 110
      RFT(L)=RFT(L)-AY(I,J+1)*DELHF(NJP)
      RST(L)=RST(L)-BY(I,J+1)*DELHS(NJP)
  110 IF (NHF(I,J).EQ.1) RFT(L)=0.
      IF (NHS(I,J).EQ.1) RST(L)=0.
  120 CONTINUE
C
C  COMPUTE SOLUTION FOR ROW J
      CALL BSBAND(AF(ID),AS(ID),RFT,RST,CI11,CI12,CI21,CI22,N,K,2)
C     **************************************************************
C  OVER-RELAX AND FIND MAXIMUM CHANGE
      KL=K
      DO 140 L=1,N
      KL=KL+1
      ERF1=RFT(L)-DELHF(KL)
      ERF=ABS(ERF1)
      ERS1=RST(L)-DELHS(KL)
      ERS=ABS(ERS1)
      DELHF(KL)=W0*RFT(L)+(1.-W0)*DELHF(KL)
      DELHS(KL)=W0*RST(L)+(1.-W0)*DELHS(KL)
      IF (ERF.LT.ERMAX) GO TO 130
      ERMAX=ERF
      ERR(KK)=ERF1
  130 IF (ERS.LT.ERMAX) GO TO 140
      ERMAX=ERS
      ERR(KK)=ERS1
  140 CONTINUE
  150 CONTINUE
      IF (ERMAX.LT.1.E-6) GO TO 170
  160 CONTINUE
      PRINT 200
      PRINT 210, (ERR(K),K=1,NIT)
```

```
      STOP
  170 CONTINUE
      PRINT 210, (ERR(K),K=1,KK)
      K=0
      DO 180 J=1,NY
      DO 180 I=1,NX
      IF (POR(I,J).LE.0.) GO TO 180
      K=K+1
      HF(I,J)=HF(I,J)+DELHF(K)
      HS(I,J)=HS(I,J)+DELHS(K)
  180 CONTINUE
      LL=LT+1
C
C**  UPDATE SPACE COEFICIENTS
      CALL COEF(LL,IDELT,DZMX)
C     ***********************
C
C**  CALCULATE BALANCE -
      CALL BAL(1,DELHF,DELHS)
C     **********************
      TIME=TIME+DELT
C
C**  PRINT RESULTS -
      CALL PDATA(LT,ITP)
C     ****************
  190 CONTINUE
C
C  END TIME LOOP
      STOP
C
  200 FORMAT (46H ITERATIONS EXCEEDED, PROGRAM ABORTED*********)
  210 FORMAT (39H MAXIMUM HEAD CHANGE FOR EACH ITERATION,/,(2X,10G12.5))
      END
      SUBROUTINE GDATA(IDELT,ITP,NFE)
C     ******************************
C
C     CALLED FROM MAIN
C     PURPOSE:  TO READ AND WRITE PROBLEM DATA
C     ------------------------------------------------------------------
      DIMENSION TITLE(20), NFE(1)
C
      COMMON /INPUT/ S(34,32),XK(34,32),YK(34,32),HF(34,32),HS(34,32),DX
     1(34),DY(32),ZB(34,32),ZI(34,32),QF(34,32),QT(34,32),POR(34,32),TH(
     234,32),NP(34,32)
      COMMON /CONTRO/ TIME,DELT,DENS,DENF,VS,VF,WT,ZMAX,ALPHA,WO,NK,NX,N
     1Y,NXX,NYY,NB,NBB,NI,MBW,MBE,IPT,NIT,RECHG
      COMMON /WORK/ AX(35,32),BX(35,32),AY(34,33),BY(34,33),TX(35,32),TY
     1(34,33),F(34,32),B(34,32),C(34,32),D(34,32),FT(34,32),ST(34,32),BH
     2(34,32),BL(34,32),NHS(34,32),NHF(34,32),DQF(34,32),DQS(34,32),WT1(
     334,32),WB(34,32)
C     ------------------------------------------------------------------
C
C     READ AND WRITE UNITS -
      NR=7
      NW=6
```

```
C
C     ITITIALIZE SIMULATION TIME -
      TIME=0.E0
C
C     READ AND WRITE PROBLEM TITLE -
      READ (NR,740) TITLE
      WRITE (NW,290)
      WRITE (NW,300) TITLE
C
C     FINITE-DIFFERENCE DATA -
      READ (NR,750) NX,NY,NB,MBE,NT,NF,NHSC,NHFC,NIT
C     NX - NUMBER OF COLUMNS (X-DIRECTION)
C     NY - NUMBER OF ROWS (Y-DIRECTION)
C     NB - NUMBER OF NON-ZERO BLOCKS
C     MBE - NOT USED
C     NT - MAXIMUM NUMBER OF TIME STEPS
C     NF - NUMBER OF SOURCES
C     NHSC - NUMBER OF CONSTANT SALT WATER HEAD NODES
C     NHFC - NUMBER OF CONSTANT FRESH WATER HEAD NODES
C     NIT - NUMBER OF LAGGING ITERATIONS
      IF (NX.NE.0) GO TO 10
      WRITE (NW,310)
      STOP
   10 IF (NY.NE.0) GO TO 20
      WRITE (NW,320)
      STOP
   20 IF (NB.NE.0) GO TO 30
      WRITE (NW,330)
      STOP
   30 CONTINUE
      WRITE (NW,340)
      WRITE (NW,350) NX,NY,NB,MBE,NT,NF,NHSC,NHFC,NIT
C
      NXX=NX+1
      NYY=NY+1
C
C     TIME PARAMETERS -
      READ (NR,760) DELT,WT,WO
C     DELT - INITIAL TIME STEP IN SECONDS
C     WT - WEIGHTING FACTOR (1. FOR UPSTREAM; .5 FOR MIDPOINT)
C     WO - OVER-RELAXATION  FACTOR (1.0-1.8)
      WRITE (NW,360) DELT,WT,WO
C
      READ (NR,370) ALPHA,IPT,ILK,ITP,ID4
C     ALPHA - 1.0 FOR WATER-TABLE CONDITIONS;
C             0.0 FOR CONFINED CONDITIONS.
C     IPT - READ 1 TO PRINT SALT WATER HEADS
C     ILK - READ 1 IF LEAKAGE IS DESIRED (CONFINED CASE)
C     ITP - NUMBER OF TIME STEPS BETWEEN PRINTING OUTPIUT
C     ID4 - READ 1 TO PRINT  MATRIX INFORMATION
      IF (ITP.EQ.0) ITP=1
      IF (ALPHA.EQ.1.0) WRITE (NW,380) ALPHA
      IF (ALPHA.EQ.0.0) WRITE (NW,390) ALPHA
C
C     DATA CODES -
```

```
      READ (NR,750) K1,K2,K3,K4,K5,K6,K7,K8,K9,K10
C     K1 - READ 1 IF X-SPACING IS UNIFORM
C     K2 - READ 1 IF Y-SPACING IS UNIFORM
C     K3 - READ 1 IF INITIAL FRESH WATER HEAD IS UNIFORM
C     K4 - READ 1 IF INITIAL INTERFACE ELEVATION IS UNIFORM
C     K5 - READ 1 IF X-HYDRAULIC CONDUCTIVITY IS UNIFORM
C     K6 - READ 1 IF Y-HYDRAULIC CONDUCTIVITY IS UNIFORM
C     K7 - READ 1 IF SPECIFIC STORAGE IS UNIFORM
C     K8 - READ 1 IF ELEVATION OF AQUIFER BASE IS UNIFORM
C     K9 - READ 1 IF POROSITY IS UNIFORM
C     K10 - READ 1 IF AQUIFER THICKNESS IS UNIFORM
      WRITE (NW,400)
      IF (K1.EQ.1) WRITE (NW,410)
      IF (K2.EQ.1) WRITE (NW,420)
      IF (K3.EQ.1) WRITE (NW,430)
      IF (K4.EQ.1) WRITE (NW,440)
      IF (K5.EQ.1) WRITE (NW,450)
      IF (K6.EQ.1) WRITE (NW,460)
      IF (K7.EQ.1) WRITE (NW,470)
      IF (K8.EQ.1) WRITE (NW,480)
      IF (K9.EQ.1) WRITE (NW,490)
      IF (K10.EQ.1.AND.ALPHA.EQ.0.) WRITE (NW,500)
C
C     MULTIPLICATION FACTORS -
      READ (NR,760) F1,F2,F3,F4,F5,F6,F7,F8
      READ (NR,760) F9,F10
      IF (F1.EQ.0.) F1=1.0
      IF (F2.EQ.0.) F2=1.0
      IF (F3.EQ.0.) F3=1.0
      IF (F4.EQ.0.) F4=1.0
      IF (F5.EQ.0.) F5=1.0
      IF (F6.EQ.0.) F6=1.0
      IF (F7.EQ.0.) F7=1.0
      IF (F8.EQ.0.) F8=1.0
      IF (F9.EQ.0.) F9=1.0
      IF (F10.EQ.0.) F10=1.0
C     F1 - MULTIPLICATION FACTOR FOR X-SPACING
C     F2 - MULTIPLICATION FACTOR FOR Y-SPACING
C     F3 - MULTIPLICATION FACTOR FOR INITIAL FRESH WATER HEAD
C     F4 - MULTIPLICATION FACTOR FOR INITIAL INTERFACE ELEVATION
C     F5 - MULTIPLICATION FACTOR FOR X-HYDRAULIC CONDUCTIVITY
C     F6 - MULTIPLICATION FACTOR FOR Y-HYDRAULIC CONDUCTIVITY
C     F7 - MULTIPLICATION FACTOR FOR SPECIFIC STORAGE
C     F8 - MULTIPLICATION FACTOR FOR ELEVATION OF AQUIFER BASE
C     F9 - MULTIPLICATION FACTOR FOR POROSITY
C     F10 - MULTIPLICATION FACTOR FOR AQUIFER THICKNESS
      WRITE (NW,510) F1,F2,F3,F4,F5,F6,F7,F8,F9,F10
C
C     SPACING -
      IF (K1.EQ.1) GO TO 40
      READ (NR,760) (DX(I),I=1,NX)
      GO TO 60
   40 READ (NR,760) DX1
      DO 50 I=1,NX
   50 DX(I)=DX1
```

```
   60 DO 70 I=1,NX
   70 DX(I)=DX(I)*F1
      WRITE (NW,520)
      WRITE (NW,530) (DX(I),I=1,NX)
C
      IF (K2.EQ.1) GO TO 80
      READ (NR,760) (DY(J),J=1,NY)
      GO TO 100
   80 READ (NR,760) DY1
      DO 90 J=1,NY
   90 DY(J)=DY1
  100 DO 110 J=1,NY
  110 DY(J)=DY(J)*F2
      WRITE (NW,540)
      WRITE (NW,530) (DY(J),J=1,NY)
C
C     INITIAL FRESH WATER HEADS -
      WRITE (NW,550)
      CALL READ(HF,K3,NY,NX,F3)
C     ************************
C
C     INITIAL INTERFACE ELEVATION -
      WRITE (NW,560)
      CALL READ(ZI,K4,NY,NX,F4)
C     ************************
C
C     X-HYDRAULIC CONDUCTIVITY -
      WRITE (NW,570)
      CALL READ(XK,K5,NY,NX,F5)
C     ************************
C
C     Y-HYDRAULIC CONDUCTIVITY -
      WRITE (NW,580)
      CALL READ(YK,K6,NY,NX,F6)
C     ************************
C
C     SPECIFIC STORAGE -
      WRITE (NW,590)
      CALL READ(S,K7,NY,NX,F7)
C     ***********************
C
C     ELEVATION OF AQUIFER BASE -
      WRITE (NW,600)
      CALL READ(ZB,K8,NY,NX,F8)
C     ************************
C
C     POROSITY -
      WRITE (NW,610)
      CALL READ(POR,K9,NY,NX,F9)
C     *************************
C     NOTE:  POROSITY IS USED TO KEY IN ON ZERO BLOCKS
      DO 120 I=1,NX
      DO 120 J=1,NY
      BH(I,J)=0.
  120 BL(I,J)=0.
```

```
C
C     AQUIFER THICKNESS -
      IF (ALPHA.EQ.1.0) GO TO 130
      WRITE (NW,620)
      CALL READ(TH,K10,NY,NX,F10)
C     ***************************
C
C     LEAKAGE -
      IF (ILK.EQ.0) GO TO 130
      READ (NR,750) K11,K12
C     K11 - READ 1 IF CONFINING BED HEAD IS UNIFORM
C     K12 - READ 1 IF CONFINING BED  LEAKANCE (K/B) IS UNIFORM
C
      READ (NR,760) F11,F12
      IF (F11.EQ.0.) F11=1.0
      IF (F12.EQ.0.) F12=1.0
C     F11 - MULTIPLICATION FACTOR FOR CONFINING BED HEAD
C     F12 - MULTIPLICATION FACTOR FOR CONFINING BED LEAKANCE
C     CONFINING BED HEAD -
      WRITE (NW,630)
      CALL READ(BH,K11,NY,NX,F11)
C     ***************************
C
C     CONFINING BED LEAKANCE -
      WRITE (NW,640)
      CALL READ(BL,K12,NY,NX,F12)
C     ***************************
  130 CONTINUE
C
C     DENSITIES AND VISCOSITIES -
      READ (NR,760) DENF,DENS,VF,VS
C     DENF - FRESH WATER DENSITY
C     DENS - SALT WATER DENSITY
C     VF - FRESH WATER VISCOSITY
C     VS - SALT WATER VISCOSITY
      WRITE (NW,650) DENF,DENS,VF,VS
C
C     RECHARGE -
      WRITE (NW,660)
      READ (NR,760) RECHG
      PRINT 770, RECHG
      DO 140 J=1,NY
      DO 140 I=1,NX
      WT1(I,J)=0.0
      WB(I,J)=0.0
      DQS(I,J)=0.
      DQF(I,J)=0.
      QT(I,J)=0.E0
  140 QF(I,J)=0.E0
C
C     CUBIC FEET PER SECOND
C     SOURCE/SINK (NEGATIVE IMPLIES OUT) -
      IF (NF.EQ.0) GO TO 160
      WRITE (NW,670)
      WRITE (NW,680)
```

```
      DO 150 K=1,NF
      READ (NR,780) I,J,GT(I,J),WT1(I,J),WB(I,J)
  150 PRINT 790, I,J,GT(I,J),WT1(I,J),WB(I,J)
C        WT1 - ELEVATION OF TOP OF WELL SCREEN
C        WB - ELEVAITON OF BOTTOM OF WELL SCREEN
  160 CONTINUE
      READ (NR,800) IDELT,ZMAX
      PRINT 810, IDELT,ZMAX
C     IDELT - READ 1 FOR CONSTANT TIME STEP;
C                  0 FOR AUTO TIME STEP (BASED ON A MAXIMUM CHANGE IN
C                    INTERFACE ELEVATION OF ZMAX)
C
C     CONSTANT SALT WATER HEAD NODES -
      DO 170 I=1,NX
      DO 170 J=1,NY
  170 NHS(I,J)=0
      IF (NHSC.EQ.0) GO TO 190
      WRITE (NW,690)
      DO 180 K=1,NHSC
      READ (NR,750) I,J
      WRITE (NW,700) I,J
  180 NHS(I,J)=1
  190 CONTINUE
C
C     CONSTANT FRESH WATER HEAD NODES -
      DO 200 I=1,NX
      DO 200 J=1,NY
  200 NHF(I,J)=0
      IF (NHFC.EQ.0) GO TO 220
      WRITE (NW,710)
      DO 210 K=1,NHFC
      READ (NR,750) I,J
      WRITE (NW,700) I,J
  210 NHF(I,J)=1
  220 CONTINUE
C
C     NUMBER GRID BLOCKS -
C     NP(I,J) - SEQUENCE NUMBERING OF BLOCKS I,J
      IBN=0
      NFE(1)=1
      DO 250 J=1,NY
      DO 240 I=1,NX
      IF (POR(I,J).EQ.0.) GO TO 230
      IBN=IBN+1
      NP(I,J)=IBN
      GO TO 240
  230 NP(I,J)=0
  240 CONTINUE
      NFE(J+1)=IBN+1
  250 CONTINUE
      NB=IBN
      IF (IO4.NE.1) GO TO 270
      WRITE (NW,720)
      DO 260 J=1,NY
  260 WRITE (NW,730) (NP(I,J),I=1,NX)
```

```
  610 FORMAT (/////,11X,8HPOROSITY/11X,8(1H-)/)
  620 FORMAT (/////,11X,17HAQUIFER THICKNESS/11X,17(1H-)/)
  630 FORMAT (/////,11X,18HCONFINING BED HEAD/11X,18(1H-)/)
  640 FORMAT (/////,11X,22HCONFINING BED LEAKANCE/11X,22(1H-)/)
  650 FORMAT (/////,11X,19HFRESH WATER DENSITY,G12.4/11X,18HSALT WATER D
     1ENSITY,G13.4/11X,21HFRESH WATER VISCOSITY,G10.4/11X,20HSALT WATER
     2VISCOSITY,G11.4/)
  660 FORMAT (/////,11X,8HRECHARGE/11X,8(1H-)/)
  670 FORMAT (/////,11X,11HSOURCE/SINK/11X,11(1H-)/)
  680 FORMAT (/,13X,15HI    J        QT,9X,3HWT1,10X,2HWB/)
  690 FORMAT (/////,11X,30HCONSTANT SALT WATER HEAD NODES/11X,30(1H-)/)
  700 FORMAT (11X,2HI=I4,4H, J=I4)
  710 FORMAT (/////,11X,31HCONSTANT FRESH WATER HEAD NODES/11X,31(1H-)/)
  720 FORMAT (///11X,12HGRID NUMBERS/11X,12(1H-))
  730 FORMAT (11X,16I5)
  740 FORMAT (20A3)
  750 FORMAT (16I4)
  760 FORMAT (8G10.4)
  770 FORMAT (//10X,12HRECHARGE =  ,G10.4)
  780 FORMAT (2I4,3G10.4)
  790 FORMAT (11X,2I4,3(2X,G10.3))
  800 FORMAT (I4,G10.4)
  810 FORMAT (//11X,7HIDELT= ,I4,2X,6HZMAX= ,G10.4)
      END
      SUBROUTINE READ(DUM,KODE,NY,NX,FAC)
C     **********************************
C
C     CALLED FROM GDATA
C     PURPOSE:  TO READ TWO-DIMENSIONAL ARRAYS
C     ------------------------------------------------------------------
      DIMENSION DUM(34,32)
C     ------------------------------------------------------------------
      NR=7
      NW=6
      IF (KODE.EQ.1) GO TO 20
      DO 10 J=1,NY
   10 READ (NR,110) (DUM(I,J),I=1,NX)
      GO TO 40
   20 READ (NR,110) DUM1
      DO 30 J=1,NY
      DO 30 I=1,NX
   30 DUM(I,J)=DUM1
   40 DO 50 J=1,NY
      DO 50 I=1,NX
   50 DUM(I,J)=DUM(I,J)*FAC
      IF (KODE.EQ.1) GO TO 70
      DO 60 J=1,NY
   60 WRITE (NW,90) J,(DUM(I,J),I=1,NX)
      GO TO 80
   70 WRITE (NW,100) DUM(1,1)
   80 CONTINUE
      RETURN
C
C
   90 FORMAT (/7X,I2,2X,8(G12.5,2X),/,(11X,8(G12.5,2X)))
```

```
C
C
C     COMPUTE INITIAL SALT WATER HEAD -
  270 DO 280 J=1,NY
      DO 280 I=1,NX
      IF (POR(I,J).LE.0.) GO TO 280
      HS(I,J)=DENF/DENS*HF(I,J)+(DENS-DENF)/DENS*ZI(I,J)
  280 CONTINUE
C
      RETURN
C
  290 FORMAT (1H1,///25X,36HSALT WATER-FRESH WATER FLOW ANALYSIS//)
  300 FORMAT (11X,70(1H*)//11X,20A3//11X,70(1H*)////)
  310 FORMAT (//5X,39H***** FATAL ERROR -- ZERO COLUMNS *****)
  320 FORMAT (//5X,36H***** FATAL ERROR -- ZERO ROWS *****)
  330 FORMAT (//5X,47H***** FATAL ERROR -- ZERO NON-ZERO BLOCKS *****)
  340 FORMAT (/11X,22HFINITE DIFFERENCE DATA/11X,22(1H-)/)
  350 FORMAT (1H ,10X,11HNUMBER OF -,2X,7HCOLUMNS,I23/21X,1H-,2X,4HROWS,
     1I26/21X,1H-,2X,15HNON-ZERO BLOCKS,I15/21X,1H-,2X,9HNOT USED ,I21/2
     21X,1H-,2X,18HMAXIMUM TIME STEPS,I12/21X,1H-,2X,15HSOURCES        ,
     3I15/21X,1H-,2X,21HCONSTANT HEADS (SALT),I9/21X,1H-,2X,22HCONSTANT
     4HEADS (FRESH),I8/21X,1H-,2X,18HLAGGING ITERATIONS,I12///)
  360 FORMAT (///,11X,15HTIME PARAMETERS/11X,15(1H-)//11X,28HINITIAL TIM
     1E STEP IN SECONDS,G22.6//11X,19HWEIGHTING PARAMETER,G31.2/11X,22HO
     2VER-RELAXATION FACTOR,G28.2/)
  370 FORMAT (G10.0,4I4)
  380 FORMAT (//,11X,6HALPHA=,F4.1,1X,22HWATER-TABLE CONDITIONS)
  390 FORMAT (//,11X,6HALPHA=,F4.1,1X,21HCONFINED CONDITIONS -/22X,56HMU
     1ST READ AQUIFER THICKNESS AND NONZERO SPECIFIC STORAGE)
  400 FORMAT (/////11X,26HTHE FOLLOWING ARE UNIFORM:)
  410 FORMAT (40X,9HX-SPACING)
  420 FORMAT (40X,9HY-SPACING)
  430 FORMAT (40X,25HINITIAL FRESH WATER HEADS)
  440 FORMAT (40X,27HINITIAL INTERFACE ELEVATION)
  450 FORMAT (40X,24HX-HYDRAULIC CONDUCTIVITY)
  460 FORMAT (40X,24HY-HYDRAULIC CONDUCTIVITY)
  470 FORMAT (40X,16HSPECIFIC STORAGE)
  480 FORMAT (40X,25HELEVATION OF AQUIFER BASE)
  490 FORMAT (40X,8HPOROSITY)
  500 FORMAT (40X,17HAQUIFER THICKNESS)
  510 FORMAT (/////,11X,26HMULTIPLICATION FACTORS FOR/11X,26(1H-)//11X,9
     1HX-SPACING,G27.3/11X,9HY-SPACING,G27.3/11X,24HINITIAL FRESH WATER
     2HEAD,G12.3/11X,27HINITIAL INTERFACE ELEVATION,G9.3/11X,24HX-HYDRAU
     3LIC CONDUCTIVITY,G12.3/11X,24HY-HYDRAULIC CONDUCTIVITY,G12.3/11X,1
     46HSPECIFIC STORAGE,G20.3/11X,25HELEVATION OF AQUIFER BASE,G11.3/11
     5X,8HPOROSITY,G28.3/11X,17HAQUIFER THICKNESS,G19.3/)
  520 FORMAT (/////,11X,22HSPACING IN X-DIRECTION/11X,22(1H-)/)
  530 FORMAT (/(11X,8(G12.5,2X)))
  540 FORMAT (/////,11X,22HSPACING IN Y-DIRECTION/11X,22(1H-)/)
  550 FORMAT (/////,11X,25HINITIAL FRESH WATER HEADS/11X,25(1H-)/)
  560 FORMAT (/////,11X,27HINITIAL INTERFACE ELEVATION/11X,27(1H-)/)
  570 FORMAT (/////,11X,24HX-HYDRAULIC CONDUCTIVITY/11X,24(1H-)/)
  580 FORMAT (/////,11X,24HY-HYDRAULIC CONDUCTIVITY/11X,24(1H-)/)
  590 FORMAT (/////,11X,16HSPECIFIC STORAGE/11X,16(1H-)/)
  600 FORMAT (/////,11X,25HELEVATION OF AQUIFER BASE/11X,25(1H-)/)
```

```
  100 FORMAT (/11X,G12.5)
  110 FORMAT (8G10.0)
      END
      SUBROUTINE TCALC(DY,DX,POR,XK,YK,TY,TX,NY,NX,NYY,NXX)
C     **************************************************
C
C     CALLED FROM MAIN
C     PURPOSE:  TO COMPUTE TRANSMISSIVITY TERMS -
C     ------------------------------------------------------------------
      DIMENSION DY(1), DX(1), XK(34,32), YK(34,32), TY(34,33), TX(35,32)
     1, POR(34,32)
C     ------------------------------------------------------------------
C
C     COMPUTE TRANSMISSIVITY TERMS IN THE X-DIRECTION -
      DO 10 J=1,NY
C     IF ONLY ONE COLUMN, SKIP CALCULATIONS -
      IF (NX.EQ.1) GO TO 20
      DO 10 I=2,NX
      TX(I,J)=0.
      IF (POR(I,J).LE.0..OR.POR(I-1,J).LE.0.) GO TO 10
      TXC=DY(J)/DX(I-1)
      TXD=DY(J)/DX(I)
      TXA=TXC*XK(I-1,J)
      TXB=TXD*XK(I,J)
      TTT=TXA+TXB
C     PERMEABILITY TERM -
      TX(I,J)=2.0*TXA*TXB/TTT
   10 CONTINUE
   20 CONTINUE
C
C     COMPUTE TRANSMISSIVITY TERMS IN THE Y-DIRECTION -
      DO 30 I=1,NX
      IF (NY.EQ.1) GO TO 40
      DO 30 J=2,NY
      TY(I,J)=0.
      IF (POR(I,J).LE.0..OR.POR(I,J-1).LE.0.) GO TO 30
      TYC=DX(I)/DY(J-1)
      TYD=DX(I)/DY(J)
      TYA=TYC*YK(I,J-1)
      TYB=TYD*YK(I,J)
      TTT=TYA+TYB
      TY(I,J)=2.0*TYA*TYB/TTT
   30 CONTINUE
   40 CONTINUE
C
C     SET TRANSMISSIVITY OF BOUNDARY BLOCKS (REGULAR RECTANGULAR MESH)
C     TO ZERO (NO-FLOW) -
      DO 50 I=1,NX
      TY(I,1)=0.E0
      TY(I,NYY)=0.E0
   50 CONTINUE
C
      DO 60 J=1,NY
      TX(1,J)=0.E0
      TX(NXX,J)=0.E0
```

```
   60 CONTINUE
      RETURN
      END
      SUBROUTINE COEF(L,IDELT,DZMX)
C     ***************************
C
C     CALLED FROM MAIN
C     PURPOSE:  TO COMPUTE THE REST OF THE SPACE COEFFICIENTS
C               (OTHER THAN THOSE COMPUTED IN TCALC)
C               AND TIME COEFFICIENTS
C     ------------------------------------------------------------------
      COMMON /INPUT/ S(34,32),XK(34,32),YK(34,32),HF(34,32),HS(34,32),DX
     1(34),DY(32),ZB(34,32),ZI(34,32),QF(34,32),QT(34,32),POR(34,32),TH(
     234,32),NP(34,32)
      COMMON /CONTRO/ TIME,DELT,DENS,DENF,VS,VF,WT,ZMAX,ALPHA,WO,NK,NX,N
     1Y,NXX,NYY,NB,NBB,NT,MBW,MBE,IPT,NIT,RECHG
      COMMON /WORK/ AX(35,32),BX(35,32),AY(34,33),BY(34,33),TX(35,32),TY
     1(34,33),F(34,32),R(34,32),C(34,32),D(34,32),FT(34,32),ST(34,32),BH
     2(34,32),BL(34,32),NHS(34,32),NHF(34,32),DQF(34,32),DQS(34,32),WT1(
     334,32),WB(34,32)
C     ------------------------------------------------------------------
      RAF=DENF/(DENF-DENS)
      RAS=DENS/(DENF-DENS)
      DZMX=-1.0
      IF (L.NE.1) GO TO 20
C
C     INITIALIZE SPACE COEFFICIENTS -
      DO 10 I=1,NX
      DO 10 J=1,NY
      FT(I,J)=0.E0
      ST(I,J)=0.E0
C     FT - COEFFICIENT FOR FRESH WATER
C     ST - COEFFICIENT FOR SALT WATER
   10 CONTINUE
   20 CONTINUE
      DO 50 I=1,NX
      DO 50 J=1,NY
      IF (POR(I,J).EQ.0.0) GO TO 50
      K=NP(I,J)
      HHF=HF(I,J)
      HHS=HS(I,J)
      ZIE=RAF*HHF-RAS*HHS
      IF (ZIE.LT.ZB(I,J)) ZIE=ZB(I,J)
      FTH=HHF-ZIE
      IF (FTH.LE.0.) FTH=0.
      STH=ZIE-ZB(I,J)
      IF (ALPHA.EQ.1.0) GO TO 30
      TOP=ZB(I,J)+TH(I,J)
      FTH=TOP-ZIE
      IF (FTH.LT.0.) FTH=0.
      STH=TH(I,J)-FTH
   30 CONTINUE
C
C     COMPUTE TIME COEFFICIENTS -
C
```

```
      AREA=DX(I)*DY(J)
C     FRESH WATER EQUATION
C        FRESH WATER DERIVATIVE -
      F(I,J)=AREA*(S(I,J)*FTH-POR(I,J)*RAF+ALPHA*POR(I,J))
C        SALT WATER DERIVATIVE -
      B(I,J)=AREA*POR(I,J)*RAS
C
C     SALT WATER EQUATION -
C        FRESH WATER DERIVATIVE -
      C(I,J)=AREA*POR(I,J)*RAF
C        SALT WATER DERIVATIVE -
      D(I,J)=AREA*(DENS/DENF*S(I,J)*STH-POR(I,J)*RAS)
C
C     COMPUTE SPACE COEFFICIENTS -
      FT(I,J)=FTH
      ST(I,J)=STH*VF/VS*DENS/DENF
      IF (L.EQ.1) GO TO 40
      IF (IDELT.EQ.1) GO TO 40
C
C     COMPUTE TIME STEP INCREASING FACTOR -
      DZI=ZI(I,J)-ZIE
      IF (DZI.LT.0.E0) DZI=-DZI
      IF (DZI.GT.DZMX) DZMX=DZI
   40 CONTINUE
   50 CONTINUE
      RETURN
      END
      SUBROUTINE FORMEQ(AF,AS,RF,RS)
C     ******************************
C
C     CALLED FROM MAIN
C     PURPOSE: FORM FINAL MATRIX EQUATIONS
C     ----------------------------------------------------------------
      DIMENSION AF(1), AS(1), RF(1), RS(1)
      COMMON /INPUT/ S(34,32),XK(34,32),YK(34,32),HF(34,32),HS(34,32),DX
     1(34),DY(32),ZB(34,32),ZI(34,32),QF(34,32),QT(34,32),POR(34,32),TH(
     234,32),NP(34,32)
      COMMON /CONTRO/ TIME,DELT,DENS,DENF,VS,VF,WT,ZMAX,ALPHA,WO,NK,NX,N
     1Y,NXX,NYY,NB,NBB,NT,MBW,MBE,IPT,NIT,RECHG
      COMMON /WORK/ AX(35,32),BX(35,32),AY(34,33),BY(34,33),TX(35,32),TY
     1(34,33),F(34,32),B(34,32),C(34,32),D(34,32),FT(34,32),ST(34,32),BH
     2(34,32),BL(34,32),NHS(34,32),NHF(34,32),DQF(34,32),DQS(34,32),WT1(
     334,32),WB(34,32)
C     ----------------------------------------------------------------
      RAF=DENF/(DENF-DENS)
      RAS=DENS/(DENF-DENS)
      DO 10 K=1,NB
      RF(K)=0.
      RS(K)=0.
      DO 10 L=1,4
      N=4*(K-1)
      AF(N+L)=0.
   10 AS(N+L)=0.
      K1=0
      K=-3
```

```
      DO 60 J=1,NY
      DO 60 I=1,NX
      IF (POR(I,J).EQ.0.) GO TO 60
      K1=K1+1
      K=K+4
      AREA=DX(I)*DY(J)
C   FRESH WATER EQUATION
C   DIAGONAL
      DIAF=-AX(I,J)-AX(I+1,J)-AY(I,J)-AY(I,J+1)-BL(I,J)*AREA
      RHSF=-DIAF*HF(I,J)-BL(I,J)*BH(I,J)*AREA-RECHG*AREA
      DIAF=DIAF-F(I,J)/DELT
C   OFF DIAGONAL
      AF(K+2)=AX(I+1,J)
C   COMPLETE RIGHT HAND SIDE
      IF (I.GT.1) RHSF=RHSF-AX(I,J)*HF(I-1,J)
      IF (I.LT.NX) RHSF=RHSF-AX(I+1,J)*HF(I+1,J)
      IF (J.GT.1) RHSF=RHSF-AY(I,J)*HF(I,J-1)
      IF (J.LT.NY) RHSF=RHSF-AY(I,J+1)*HF(I,J+1)
C   SALT WATER EQUATION
C   DIAGONAL
      DIAS=-BX(I,J)-BX(I+1,J)-BY(I,J)-BY(I,J+1)
      RHSS=-DIAS*HS(I,J)
      DIAS=DIAS-B(I,J)/DELT
C   OFF DIAGONAL
      AS(K+3)=BX(I+1,J)
C   COMPLETE RIGHT HAND SIDE
      IF (I.GT.1) RHSS=RHSS-BX(I,J)*HS(I-1,J)
      IF (I.LT.NX) RHSS=RHSS-BX(I+1,J)*HS(I+1,J)
      IF (J.GT.1) RHSS=RHSS-BY(I,J)*HS(I,J-1)
      IF (J.LT.NY) RHSS=RHSS-BY(I,J+1)*HS(I,J+1)
C   WELL TERMS
      FDQB=1.0
      DQDF=0.
      DQDS=0.
      IF (QT(I,J).EQ.0.) GO TO 30
      IF (ZI(I,J).LT.WB(I,J)) GO TO 20
      FDQB=0.
      WTOP=WT1(I,J)
      IF (ALPHA.EQ.1.) WTOP=HF(I,J)
      IF (ZI(I,J).GT.WTOP) GO TO 20
      WTH=WTOP-WB(I,J)
      IF (WTH.LE.0.) GO TO 30
      WFTH=WTOP-RAF*HF(I,J)+RAS*HS(I,J)
      FDQB=WFTH/WTH
      DQDF=(WTH*(ALPHA-RAF)-ALPHA*WFTH)/(WTH*WTH)*QT(I,J)
      DQDS=RAS/WTH*QT(I,J)
   20 RHSF=RHSF-QT(I,J)*FDQB
      RHSS=RHSS-QT(I,J)*(1.-FDQB)
      DIAF=DIAF+DQDF
      DIAS=DIAS-DQDS
      QF(I,J)=QT(I,J)*FDQB
      DQF(I,J)=DQDF
      DQS(I,J)=DQDS
   30 CONTINUE
C   CONSTANT HEAD CONDITIONS
```

```
C   FRESH WATER
       IF (NHF(I,J).NE.1) GO TO 40
       RHSF=0.
       DIAF=DIAF*1.E16
C   SALT WATER
    40 IF (NHS(I,J).NE.1) GO TO 50
       RHSS=0.
       DIAS=DIAS*1.E16
    50 CONTINUE
C   COMPLETE LOADING
       AF(K+1)=DQDS-B(I,J)/DELT
       AS(K)=-DQDF-C(I,J)/DELT
       AF(K)=DIAF
       AS(K+1)=DIAS
       RF(K1)=RHSF
       RS(K1)=RHSS
    60 CONTINUE
       RETURN
       END
       SUBROUTINE BSBAND(A1,A2,B1,B2,CI11,CI12,CI21,CI22,NEQ,NT,INTRY)
C      *****************************************************************
C
C      CALLED FROM MAIN
C      PURPOSE: FACTOR BLOCK SYMMETRIC MATRIX
C               FORWARD AND BACK SUBSTITUTE
C      -----------------------------------------------------------------
       DIMENSION A1(1), A2(1), B1(1), B2(1), CI11(1), CI12(1), CI21(1), C
      1I22(1)
C      -----------------------------------------------------------------
       NR=NEQ-1
       NI=NEQ
       IF (NEQ.EQ.1) GO TO 50
       IF (INTRY.EQ.2) GO TO 20
       I1=-3
       DO 10 I=1,NR
       I1=I1+4
       I2=I1+1
       I3=I1+2
       I4=I1+3
       L1=I1+4
       L2=L1+1
       IT=NT+I
       L=I+1
       D1=A1(I1)*A2(I2)-A1(I2)*A2(I1)
       C1=1./D1
       P11=A2(I2)*C1
       P12=-A1(I2)*C1
       P21=-A2(I1)*C1
       P22=A1(I1)*C1
       IF (L.GT.NEQ) GO TO 10
       C11=A1(I3)*P11+A1(I4)*P21
       C12=A1(I3)*P12+A1(I4)*P22
       C21=A2(I3)*P11+A2(I4)*P21
       C22=A2(I3)*P12+A2(I4)*P22
       CI11(IT)=C11
```

```
      CI12(IT)=C12
      CI21(IT)=C21
      CI22(IT)=C22
      A1(L1)=A1(L1)+(-C11*A1(I3)-C12*A2(I3))
      A1(L2)=A1(L2)+(-C11*A1(I4)-C12*A2(I4))
      A2(L1)=A2(L1)+(-C21*A1(I3)-C22*A2(I3))
      A2(L2)=A2(L2)+(-C21*A1(I4)-C22*A2(I4))
      D11=P11*A1(I3)+P12*A2(I3)
      D12=P11*A1(I4)+P12*A2(I4)
      D21=P21*A1(I3)+P22*A2(I3)
      D22=P21*A1(I4)+P22*A2(I4)
      A1(I3)=D11
      A1(I4)=D12
      A2(I3)=D21
      A2(I4)=D22
   10 CONTINUE
      RETURN
C
C***   FORWARD SUBSTITUTION
C
   20 CONTINUE
      I1=-3
      DO 40 I=1,NR
      I1=I1+4
      I2=I1+1
      IT=NT+1
      L=I+1
      D1=A1(I1)*A2(I2)-A1(I2)*A2(I1)
      C1=1./D1
      P11=A2(I2)*C1
      P12=-A1(I2)*C1
      P21=-A2(I1)*C1
      P22=A1(I1)*C1
      IF (L.GT.NEQ) GO TO 30
      B1(L)=B1(L)-CI11(IT)*B1(I)-CI12(IT)*B2(I)
      B2(L)=B2(L)-CI21(IT)*B1(I)-CI22(IT)*B2(I)
   30 BB1=P11*B1(I)+P12*B2(I)
      BB2=P21*B1(I)+P22*B2(I)
      B1(I)=BB1
      B2(I)=BB2
   40 CONTINUE
C
C***   BACKWARD SUBSTITUTION
C
   50 N1=4*NR+1
      N2=N1+1
      DN=A1(N1)*A2(N2)-A1(N2)*A2(N1)
      CN=1./DN
      B2N=(A1(N1)*B2(NI)-A2(N1)*B1(NI))*CN
      B1N=(A2(N2)*B1(NI)-A1(N2)*B2(NI))*CN
      B1(NI)=B1N
      B2(NI)=B2N
      IF (NEQ.EQ.1) RETURN
      K3=N2+1
      DO 60 I=1,NR
```

```
      K3=K3-4
      K4=K3+1
      K=NEQ-I
      L=K+1
      B2(K)=B2(K)-A2(K3)*B1(L)-A2(K4)*B2(L)
      B1(K)=B1(K)-A1(K3)*B1(L)-A1(K4)*B2(L)
   60 CONTINUE
      RETURN
      END
      SUBROUTINE PDATA(L,ITP)
C     ***********************
C
C     CALLED FROM MAIN
C     PURPOSE:  TO PRINT COMPUTED OUTPUT
C     ------------------------------------------------------------------
      DIMENSION MAF(34,32)
      COMMON /INPUT/ S(34,32),XK(34,32),YK(34,32),HF(34,32),HS(34,32),DX
     1(34),DY(32),ZB(34,32),ZI(34,32),QF(34,32),QT(34,32),POR(34,32),TH(
     234,32),NP(34,32)
      COMMON /CONTRO/ TIME,DELT,DENS,DENF,VS,VF,WT,ZMAX,ALPHA,WO,NK,NX,N
     1Y,NXX,NYY,NB,NBB,MT,MBW,MBE,IPT,NIT,RECHG
      COMMON /WORK/ AX(35,32),BX(35,32),AY(34,33),BY(34,33),TX(35,32),TY
     1(34,33),F(34,32),B(34,32),C(34,32),D(34,32),FT(34,32),ST(34,32),BH
     2(34,32),BL(34,32),NHS(34,32),NHF(34,32),DQF(34,32),DQS(34,32),WT1(
     334,32),NB(34,32)
C     ------------------------------------------------------------------
C
      TMN=TIME/60.
      THR=TMN/60.
      TDA=THR/24.
      TYR=TDA/365.
      PRINT 90, L,TIME,TMN,THR,TDA,TYR
C
C     COMPUTE INTERFACE ELEVATION -
      RAF=DENF/(DENF-DENS)
      RAS=DENS/(DENF-DENS)
      DO 10 I=1,NX
      DO 10 J=1,NY
      IF (POR(I,J).EQ.0.) GO TO 10
      ZI(I,J)=RAF*HF(I,J)-RAS*HS(I,J)
   10 CONTINUE
      IF (MOD(L,ITP).NE.0) GO TO 80
      PRINT 100
      DO 20 J=1,NY
   20 PRINT 110, J,(HF(I,J),I=1,NX)
      PRINT 120
      DO 30 J=1,NY
   30 PRINT 110, J,(ZI(I,J),I=1,NX)
      IF (IPT.NE.1) GO TO 50
      PRINT 130
      DO 40 J=1,NY
   40 PRINT 110, J,(HS(I,J),I=1,NX)
   50 CONTINUE
      DO 60 I=1,NX
      DO 60 J=1,NY
```

```
      MAP(I,J)=0
      ZIT=ZI(I,J)
      ZBT=ZB(I,J)
      IF (ZIT.GT.ZBT) MAP(I,J)=1
      IF (ABS(ZIT-ZBT).LE.0.01) MAP(I,J)=0
      TOP=ZBT+TH(I,J)
      IF (ALPHA.EQ.1) TOP=HF(I,J)
      IF (ZIT.GT.TOP) MAP(I,J)=2
      IF (ABS(ZIT-TOP).LE.0.01) MAP(I,J)=2
      IF (POR(I,J).EQ.0.) MAP(I,J)=9
   60 CONTINUE
      PRINT 140
      DO 70 J=1,NY
   70 PRINT 150, (MAP(I,J),I=1,NX)
   80 CONTINUE
      RETURN
C
   90 FORMAT (///11X,11HSTEP NUMBER,I4,10X,26HSIMULATION TIME IN SECONDS
     1,E10.3/11X,15(1H*),26X,10HIN MINUTES,E10.3/52X,10HIN HOURS   ,E10.3
     2/52X,10HIN DAYS    ,E10.3/52X,10HIN YEARS   ,E10.3//)
  100 FORMAT (11X,16HFRESH WATER HEAD/11X,16(1H-)//)
  110 FORMAT (/7X,I2,2X,8(G12.5,2X),/,(11X,8(G12.5,2X)))
  120 FORMAT (//11X,15HINTERFACE LEVEL/11X,15(1H-)//)
  130 FORMAT (//11X,15HSALT WATER HEAD/11X,15(1H-)//)
  140 FORMAT (//11X,42HWATER QUALITY MAP:  0=FRESH, 1=MIX, 2=SALT/11X,17
     1(1H-)//)
  150 FORMAT (11X,40I2)
      END
      SUBROUTINE BAL(ICX,DELHF,DELHS)
C     ******************************
C
C     CALLED FROM MAIN
C     PURPOSE:  TO COMPUTE MASS BALANCE
C     ----------------------------------------------------------
      DIMENSION DELHF(1), DELHS(1)
C
      COMMON /INPUT/ S(34,32),XK(34,32),YK(34,32),HF(34,32),HS(34,32),DX
     1(34),DY(32),ZB(34,32),ZI(34,32),QF(34,32),QT(34,32),POR(34,32),TH(
     234,32),NP(34,32)
      COMMON /CONTRO/ TIME,DELT,DENS,DENF,VS,VF,WT,ZMAX,ALPHA,WO,NK,NX,N
     1Y,NXX,NYY,NB,NBB,NT,MBW,MBE,IPT,NIT,RECHG
      COMMON /WORK/ AX(35,32),BX(35,32),AY(34,33),BY(34,33),TX(35,32),TY
     1(34,33),F(34,32),B(34,32),C(34,32),D(34,32),FT(34,32),ST(34,32),BH
     2(34,32),BL(34,32),NHS(34,32),NHF(34,32),DQF(34,32),DQS(34,32),WT1(
     334,32),WB(34,32)
C     ----------------------------------------------------------
C
C  COMPUTE TOTAL MASS IN THE SYSTEM AND CHANGE IN MASS
      TMFW=0.
      TMSW=0.
      DMFW=0.
      DMSW=0.
      DMFWE=0.
      DMSWE=0.
      DO 10 I=1,NX
```

```
      DO 10 J=1,NY
      IF (POR(I,J).LE.0.) GO TO 10
      AREA=DX(I)*DY(J)
      FTH=FT(I,J)
      STH=ST(I,J)*VS/VF*DENF/DENS
      TMFW=TMFW+FTH*POR(I,J)*AREA
      TMSW=TMSW+STH*POR(I,J)*AREA
      IF (ICX.EQ.0) GO TO 10
      K=NP(I,J)
C   SUM TOTAL CHANGE FROM EQUATION VIEWPOINT
      DMFWE=DMFWE+F(I,J)*DELHF(K)+B(I,J)*DELHS(K)
      DMSWE=DMSWE+C(I,J)*DELHF(K)+D(I,J)*DELHS(K)
C   SUM THE STORAGE PART OF THE TRUE MASS CHANGE
      DMFW=DMFW+S(I,J)*FTH*DELHF(K)*AREA
      DMSW=DMSW+S(I,J)*DENS/DENF*STH*DELHS(K)*AREA
   10 CONTINUE
      IF (ICX.EQ.0) GO TO 50
      CHF=0.
      CHS=0.
      RQ=0.
      QFT=0.
      QST=0.
      QLK=0.
C   BEGIN LOOP TO CALCULATE RATES OF OTHER COMPONENTS
      DO 40 I=1,NX
      DO 40 J=1,NY
      IF (POR(I,J).LE.0.) GO TO 40
      K=NP(I,J)
      AREA=DX(I)*DY(J)
C   CONSTANT HEAD FLUX--FRESH WATER
      IF (NHF(I,J).NE.1) GO TO 20
      X=HF(I,J)
      CHF=CHF+B(I,J)/DELT*DELHS(K)+AY(I,J+1)*X+AY(I,J)*X+AX(I+1,J)*X+AX(
     1I,J)*X
      IF (J.GT.1) CHF=CHF-AY(I,J)*HF(I,J-1)
      IF (J.LT.NY) CHF=CHF-AY(I,J+1)*HF(I,J+1)
      IF (I.GT.1) CHF=CHF-AX(I,J)*HF(I-1,J)
      IF (I.LT.NX) CHF=CHF-AX(I+1,J)*HF(I+1,J)
C   CONSTANT HEAD FLUX--SALT WATER
   20 IF (NHS(I,J).NE.1) GO TO 30
      X=HS(I,J)
      CHS=CHS+C(I,J)/DELT*DELHF(K)+BY(I,J)*X+BY(I,J+1)*X+BX(I,J)*X+BX(I+
     11,J)*X
      IF (J.GT.1) CHS=CHS-BY(I,J)*HS(I,J-1)
      IF (J.LT.NY) CHS=CHS-BY(I,J+1)*HS(I,J+1)
      IF (I.GT.1) CHS=CHS-BX(I,J)*HS(I-1,J)
      IF (I.LT.NX) CHS=CHS-BX(I+1,J)*HS(I+1,J)
   30 IF (NHF(I,J).NE.1) RQ=RQ+RECHG*AREA
C   SOURCE-SINK TERMS
      DELQ=DQF(I,J)*DELHF(K)+DQS(I,J)*DELHS(K)
      QFT=QFT+QF(I,J)+DELQ
      QST=QST+QT(I,J)-QF(I,J)-DELQ
C   LEAKAGE TERM
      IF (NHF(I,J).NE.1) QLK=QLK+BL(I,J)*(BH(I,J)-HF(I,J))*AREA
   40 CONTINUE
```

```
C   VOLUME TOTALS
      TCHF=CHF*DELT
      TCHS=CHS*DELT
      TRQ=RQ*DELT
      TQF=QFT*DELT
      TQS=QST*DELT
      TQLK=QLK*DELT
      DELMF=-TMFW+TMFW1-DMFW
      DELMS=-TMSW+TMSW1-DMSW
      TMFW1=TMFW
      TMSW1=TMSW
C   STORAGE CHANGE RATES
      X=1./DELT
      DELMFR=DELMF*X
      DELMSR=DELMS*X
      DELMFE=-DMFWE*X
      DELMSE=-DMSWE*X
C   PERCENT ERROR
      SUMP=0.
      SUMN=0.
      IF (TQLK.GT.0.) SUMP=SUMP+TQLK
      IF (TQLK.LE.0.) SUMN=SUMN+TQLK
      IF (TCHF.GT.0.) SUMP=SUMP+TCHF
      IF (TCHF.LE.0.) SUMN=SUMN+TCHF
      IF (TRQ.GT.0.) SUMP=SUMP+TRQ
      IF (TRQ.LE.0.) SUMN=SUMN+TRQ
      IF (DELMF.GT.0.) SUMP=SUMP+DELMF
      IF (DELMF.LE.0.) SUMN=SUMN+DELMF
      IF (TQF.GT.0.) SUMP=SUMP+TQF
      IF (TQF.LE.0.) SUMN=SUMN+TQF
      RES1=SUMN+SUMP
      PCEFM=RES1*100.
      IF (SUMP.GT.0.) PCEFM=PCEFM/SUMP
      SUMP=0.
      SUMN=0.
      IF (TCHS.GT.0.) SUMP=SUMP+TCHS
      IF (TCHS.LE.0.) SUMN=SUMN+TCHS
      IF (DELMS.GT.0.) SUMP=SUMP+DELMS
      IF (DELMS.LE.0.) SUMN=SUMN+DELMS
      IF (TQS.GT.0.) SUMP=SUMP+TQS
      IF (TQS.LE.0.) SUMN=SUMN+TQS
      RES3=SUMN+SUMP
      PCESM=RES3*100.
      IF (SUMP.GT.0.) PCESM=PCESM/SUMP
      PRINT 60, DELMFR,DELMF,DELMSR,DELMS,QFT,TQF,QST,TQS,CHF,TCHF,CHS,T
     1CHS,QLK,TQLK,RQ,TRQ,PCEFM,PCESM
      RES2=RES1*X
      RES4=RES3*X
      PRINT 70, RES2,RES1,RES4,RES3
      PRINT 80, DELMFE,DELMSE
      RETURN
   50 TMFW1=TMFW
      TMSW1=TMSW
      PRINT 90, TMFW,TMSW
      RETURN
```

```
C
   60 FORMAT (//46X,13HFRESH BALANCE,33X,12HSALT BALANCE/46X,13(1H-),33X
     1,12(1H-)/41X,10HRATE (CFS),5X,10HTOTAL (CF),19X,10HRATE (CFS),5X,1
     20HTOTAL (CF)//11X,17HCHANGE IN STORAGE,13X,G12.5,2X,G12.5,18X,G12.
     35,2X,G12.5/11X,11HSOURCE/SINK,19X,G12.5,2X,G12.5,18X,G12.5,2X,G12.
     45/11X,19HCONSTANT HEAD NODES,11X,G12.5,2X,G12.5,18X,G12.5,2X,G12.5
     5/11X,16HVERTICAL LEAKAGE,14X,G12.5,2X,G12.5/11X,8HRECHARGE,22X,G12
     6.5,2X,G12.5/11X,14HPER CENT ERROR,30X,G12.5,32X,G12.5/)
   70 FORMAT (11X,6HTOTALS,24X,G12.5,2X,G12.5,18X,G12.5,2X,G12.5)
   80 FORMAT (11X,21HERR CHANGE IN STORAGE,9X,G12.5,32X,G12.5/)
   90 FORMAT (//11X,20HINITIAL FRESH VOLUME,G12.5,1X,10HCUBIC FEET/11X,2
     10HINITIAL  SALT VOLUME,G12.5,1X,10HCUBIC FEET/)
      END

EOF..
```

```
                SALT WATER-FRESH WATER FLOW ANALYSIS

****************************************************************************

      HELE SHAW EXPERIMENTS       TRANSIENT

****************************************************************************


FINITE DIFFERENCE DATA
----------------------

NUMBER OF -  COLUMNS                   20
          -  ROWS                       1
          -  NON-ZERO BLOCKS           20
          -  NOT USED                   2
          -  MAXIMUM TIME STEPS         3
          -  SOURCES                    0
          -  CONSTANT HEADS (SALT)      1
          -  CONSTANT HEADS (FRESH)     1
          -  LAGGING ITERATIONS       150


TIME PARAMETERS
---------------

INITIAL TIME STEP IN SECONDS        5.0000000

WEIGHTING PARAMETER                      0.50
OVER-RELAXATION FACTOR                    1.0


ALPHA= 0.0 CONFINED CONDITIONS -
           MUST READ AQUIFER THICKNESS AND NONZERO SPECIFIC STORAGE


THE FOLLOWING ARE UNIFORM:
                              X-SPACING
                              Y-SPACING
                              X-HYDRAULIC CONDUCTIVITY
                              Y-HYDRAULIC CONDUCTIVITY
                              SPECIFIC STORAGE
```

```
ELEVATION OF AQUIFER BASE
POROSITY
AQUIFER THICKNESS


MULTIPLICATION FACTORS FOR
--------------------------

X-SPACING                      1.00
Y-SPACING                      1.00
INITIAL FRESH WATER HEAD       1.00
INITIAL INTERFACE ELEVATION    1.00
X-HYDRAULIC CONDUCTIVITY       1.00
Y-HYDRAULIC CONDUCTIVITY       1.00
SPECIFIC STORAGE               1.00
ELEVATION OF AQUIFER BASE      1.00
POROSITY                       1.00
AQUIFER THICKNESS              1.00


SPACING IN X-DIRECTION
----------------------

   10.000   10.000   10.000   10.000   10.000   10.000   10.000   10.000
   10.000   10.000   10.000   10.000   10.000   10.000   10.000   10.000
   10.000   10.000   10.000   10.000


SPACING IN Y-DIRECTION
----------------------

   1.0000


INITIAL FRESH WATER HEADS
-------------------------

1  29.496   27.790   27.769   27.748   27.727   27.704   27.681   27.658
   27.633   27.607   27.581   27.552   27.523   27.491   27.458   27.421
   27.381   27.337   27.285   27.222
```

INITIAL INTERFACE ELEVATION

1	0.0000	0.0000	0.43040	1.1592	1.9027	2.6693	3.4617	4.2820
	5.1334	6.0200	6.9471	7.9200	8.9471	10.038	11.204	12.463
	13.841	15.381	17.161	19.346				

X-HYDRAULIC CONDUCTIVITY

69.000

Y-HYDRAULIC CONDUCTIVITY

69.000

SPECIFIC STORAGE

0.10000E-05

ELEVATION OF AQUIFER BASE

0.0000

POROSITY

1.0000

AQUIFER THICKNESS

27.000

CONFINING BED HEAD

27.136

CONFINING BED LEAKANCE

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	4.6087				

FRESH WATER DENSITY 1.001
SALT WATER DENSITY 1.030
FRESH WATER VISCOSITY 1.000
SALT WATER VISCOSITY 1.000

RECHARGE

```
RECHARGE =   0.000

 IDELT=    1  ZMAX=  0.000


 CONSTANT SALT WATER HEAD NODES
 ------------------------------

 I=  20, J=   1


 CONSTANT FRESH WATER HEAD NODES
 -------------------------------

 I=   1, J=   1

 INITIAL FRESH VOLUME  3916.9     CUBIC FEET
 INITIAL  SALT VOLUME  1483.1     CUBIC FEET
MAXIMUM HEAD CHANGE FOR EACH ITERATION
   1.5979      0.0000
```

	FRESH BALANCE		SALT BALANCE	
	RATE (CFS)	TOTAL (CF)	RATE (CFS)	TOTAL (CF)
CHANGE IN STORAGE	-14.227	-71.135	14.226	71.131
SOURCE/SINK	0.0000	0.0000	0.0000	0.0000
CONSTANT HEAD NODES	20.131	100.66	-14.372	-71.862
VERTICAL LEAKAGE	-5.7580	-28.790		
RECHARGE	0.0000	0.0000		
PER CENT ERROR		0.72647		-1.0280
TOTALS	0.14625	0.73124	-0.14625	-0.73124
EQN CHANGE IN STORAGE	-14.373		14.372	

```
STEP NUMBER   1         SIMULATION TIME IN SECONDS 0.500E+01
***************                         IN MINUTES 0.833E-01
                                        IN HOURS   0.139E-02
                                        IN DAYS    0.579E-04
                                        IN YEARS   0.159E-06
```

```
FRESH WATER HEAD
----------------

1   29.496      29.388      29.280      29.171      29.062      28.952      28.841      28.730
    28.618      28.504      28.390      28.274      28.157      28.038      27.917      27.794
    27.667      27.536      27.401      27.261

INTERFACE LEVEL
---------------

1  0.37253E-08  -0.73127E-01  0.25958   0.93855     1.6738      2.4329      3.2171      4.0283
    4.8697       5.7448      6.6586     7.6162      8.6248      9.6928      10.830      12.051
    13.371      14.809      16.374      18.002

SALT WATER HEAD
---------------

1   28.666      28.558      28.463      28.376      28.291      28.205      28.120      28.034
    27.949      27.864      27.778      27.693      27.607      27.522      27.436      27.350
    27.264      27.178      27.090      27.000

WATER QUALITY MAP:  0=FRESH, 1=MIX, 2=SALT
-----------------

    0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
MAXIMUM HEAD CHANGE FOR EACH ITERATION
  -1.5975     0.0000

                             FRESH BALANCE                           SALT BALANCE
                             -------------                           ------------
                          RATE (CFS)     TOTAL (CF)              RATE (CFS)     TOTAL (CF)

CHANGE IN STORAGE          -12.912       -64.562                  12.912        64.562
SOURCE/SINK                 0.0000        0.0000                  0.0000        0.0000
CONSTANT HEAD NODES         20.046        100.23                 -13.003       -65.014
VERTICAL LEAKAGE           -7.0437       -35.219
RECHARGE                    0.0000        0.0000
PER CENT ERROR                            0.45025                              -0.69901

TOTALS                     0.90259E-01   0.45130             -0.90260E-01   -0.45130
EQN CHANGE IN STORAGE      -13.003                                13.003
```

```
STEP NUMBER   2          SIMULATION TIME IN SECONDS 0.100E+02
***************                          IN MINUTES 0.167E+00
                                         IN HOURS   0.278E-02
                                         IN DAYS    0.116E-03
                                         IN YEARS   0.317E-06


FRESH WATER HEAD
----------------


  1   29.496      29.368      29.281      29.173      29.064      28.955      28.845      28.734
      28.623      28.510      28.397      28.282      28.165      28.048      27.928      27.806
      27.681      27.553      27.421      27.289

INTERFACE LEVEL
---------------


  1  0.37253E-08 -0.11826     0.11812     0.72966     1.4494      2.2003      2.9763      3.7789
      4.6105      5.4747      6.3759      7.3188      8.3099      9.3564      10.467      11.651
      12.919      14.273      15.690      17.039

SALT WATER HEAD
---------------


  1   28.666      28.558      28.460      28.372      28.287      28.202      28.117      28.032
      27.947      27.862      27.777      27.691      27.606      27.521      27.436      27.351
      27.265      27.179      27.091      27.000

WATER QUALITY MAP:  0=FRESH, 1=MIX, 2=SALT
-----------------


      0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
MAXIMUM HEAD CHANGE FOR EACH ITERATION
 -0.69953E-02  0.0000


                                       FRESH BALANCE                       SALT BALANCE
                                       -------------                       ------------
                                  RATE (CFS)   TOTAL (CF)            RATE (CFS)   TOTAL (CF)

CHANGE IN STORAGE                  -11.927      -59.633               11.927       59.633
SOURCE/SINK                         0.0000       0.0000               0.0000       0.0000
CONSTANT HEAD NODES                 19.976       99.878              -11.969      -59.843
VERTICAL LEAKAGE                   -8.0071      -40.035
RECHARGE                            0.0000       0.0000
```

```
PER CENT ERROR                               0.21027                                               -0.35218

TOTALS                          0.42002E-01   0.21001                         -0.42002E-01  -0.21001
EQN CHANGE IN STORAGE          -11.969                                         11.969


STEP NUMBER   3            SIMULATION TIME IN SECONDS 0.150E+02
***************                            IN MINUTES 0.250E+00
                                           IN HOURS   0.417E-02
                                           IN DAYS    0.174E-03
                                           IN YEARS   0.476E-06

FRESH WATER HEAD
----------------

1   29.496       29.389       29.282       29.174       29.066       28.957       28.848       28.738
    28.627       28.515       28.402       28.288       28.173       28.056       27.937       27.816
    27.693       27.567       27.439       27.310

INTERFACE LEVEL
---------------

1  0.57253E-08  -0.13926      0.44071E-02  0.53412      1.2306       1.9715       2.7393       3.5333
    4.3556       5.2093       6.0983       7.0274       8.0018       9.0281      10.113       11.264
    12.485       13.771       15.090       16.318

SALT WATER HEAD
---------------

1   28.666       28.557       28.457       28.368       28.282       28.198       28.113       28.028
    27.944       27.859       27.774       27.690       27.605       27.520       27.435       27.350
    27.265       27.179       27.091       27.000

WATER QUALITY MAP:  0=FRESH, 1=MIX, 2=SALT
-----------------

  0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```


☆ U.S. GOVERNMENT PRINTING OFFICE: 1980 O— 311-344/86