INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 94S28S1

A finite-difference model of three-dimensional granular displacement

Burbey, Thomas J., Ph.D.

University of Nevada, Reno, 1994

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

University of Nevada

Reno

A Finite-Difference Model of Three-Dimensional Granular Displacement

A dissertation submitted in partial fulfillment of the
requirements for the degreee of Doctor of Philosophy
in Hydrology/Hydrogeology

by

Thomas J. Burbey

Donald C. Helm, Dissertation Advisor

May 1994

The Dissertation of Thomas J. Burbey is approved:

Donald C. Hel

Dissertation Advisor

Department Chair

Dean, Graduate School

University of Nevada

Reno

May 1994

ACKNOWLEDGEMENTS

I would like to express sincere thanks to the U.S. Geological Survey, my employer, for allowing me to pursue this degree and for their confidence in me and their commitment to the work of this project. The U.S. Geological Survey, the Nevada Division of Water Resources and the Las Vegas Valley Water District have funded this research and without their support through this long process, this product would never have come to fruition. Many thanks go to Bill Carswell and Jon Nowlin for believing in me and allowing me to remain here at UNR during the pursuit of this degree.

In addition I would like to thank my committee members, Steve Wheatcraft, Gene LeMay, Bob Watters, Jim Carr, and particularly Don Helm who served as my advisor and mentor. Don has been most gracious during this endeavor, often allowing me to interrupt his work to discuss problems as we filled many-a-blackboard with ideas and developments. His continued leadership and faith in me gave me the push I often needed when simulations and boundary-condition problems left me frazzled. Above all, Don has been a true friend.

I want to thank my family; my wife Ingrid who is my very best friend, companion, and prayer partner, who encouraged me continually through the good and hard times; and my children Rena, Ariel, and Ryan whose cheerful faces and constant exuberance allowed me to forget the troubles and problems and to remember what is really important in life.

Above all I must thank my Lord God who is my constant source of strength and hope who has taught me much about myself and His graciousness to me though this whole process. Thanks also go to all those who prayed for me over these years.

ABSTRACT

Recent advances in aquifer mechanics have shown that the hydrodynamic processes associated with land subsidence and earth fissuring due to fluid withdrawal in unconsolidated aquifers are three dimensional in scope. Previous mathematical and numerical models that use hydraulic head or volume strain as the principal unknown variable have traditionally been one dimensional with respect to changes in storage and strain because they assume no horizontal strain. These one-dimensional models can accurately simulate the total vertical compaction of interbeds in a confined aquifer, but they have no way of predicting horizontal changes in strain or granular movement, and hence can not estimate where damaging fissures may occur over time. This report describes a new three-dimensional finite-difference numerical model that has been developed and integrated into the U.S. Geological Survey's modular ground-water flow model. The displacement field of solids is the principal unknown variable within the new model. Because the displacement field of solids is a vector quantity, granular displacement resulting from imposed stresses on an unconsolidated aquifer can be simulated in three dimensions. The new model is not limited to confined or homogeneous aquifers, but can be readily applied to unconfined and heterogeneous aquifers with complex boundaries.

The three-dimensional governing equation is decoupled and each component direction is solved for, first independently, then corporately with the other principal directions. Each of the three decoupled equations is expressed numerically using a Crank-Nicolson scheme. Solution of the set of equations is accomplished with a dual-loop successive overrelaxation technique, while taking advantage of Chebyshev acceleration. Simulation of horizontal displacements compare accurately with analytic solutions for a homogeneous, isotropic confined aquifer. Simulation of vertical displacements of fine-grained interbeds within a confined aquifer compare favorably with results obtained using the one-dimensional interbed storage model. The inclusion of an overlying horizontal barrier to vertical flow results in an increase in calculated

subsidence along the edges of the barrier and a decrease in subsidence directly above the pumped well. A vertical barrier to horizontal flow tends to increase subsidence above the pumped well. The horizontal location of the wellbore tends to be drawn toward the barrier resulting in compressional strain between the barrier and the pumped well. Displacement is significant on the side of the barrier opposite the pumped well.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

v

LIST OF TABLES

Si:

INTRODUCTION

Increasing awareness of the connection between horizontal aquifer movement and earth fissures in pumped unconsolidated aquifers has triggered concern over the effectiveness of available hydrologic models to predict such phenomena. While vertical compaction or subsidence models have been successfully developed and applied to field settings (Gambolati and Freeze, 1973; Gambolati and others, 1974; Helm, 1972, 1975, 1976; Narasimhan and Witherspoon, 1977: Lewis and Schrefler, 1978; Leake and Prudic, 1988; Leake, 1990), available horizontal or three-dimensional displacement models have been intractable or require significant physical limitations for practical application under field conditions (Biot, 1941, 1955; Safri and Pinder, 1979, 1980: Bear and Corapcioglu, 1981). Helm (1979, 1982, 1984, 1987) has developed a fundamental theory of granular movement based on an extension of Darcy's law that has the displacement-field of solids as the lone dependent variable. Helm's approach makes calculating the displacement field more tractable because it does not require the prior calculation of transient values of hydraulic head, effective stress, or pressure.

Both field and analytical studies have shown that horizontal movement can be of the same order of magnitude as vertical compaction (Poland and Davis, 1969; Yerkes and Castle, 1969; Bear and Corapcioglu, 1981). In fact, field evidence suggests that horizontal movement of the granular matrix may occur where no vertical compaction is measured (Wolff, 1970). Although geologic influences such as differential movement along buried faults (Bell and others, 1992) and shallow bedrock knobs (Carpenter, 1991) may influence the location of fissure development, hydraulic mechanisms, specifically horizontal movement, influence the magnitude and severity of fissure development (Helm, 1993).

The purpose of this dissertation is to develop a model that is capable of simulating time-dependent granular movement in three space dimensions and that is tractable at the field scale. Such a model would greatly expand the state-of-the-art and allow for greater understanding in evaluating displacements in complex geologic settings (anisotropic and heterogeneous aquifer properties and application of multiple pumping and recharge wells) and in cases where stresses, such as those caused by pumping, are changing in time. From the resulting displacement field one may be able to predict where earth fissures would most likely occur. Hence, such a tool would benefit not only scientists but water managers who are interested in minimizing potential risks of structural damage from fissure development

This dissertation incorporates Helm's general theory of three-dimensional granular movement into the U.S. Geological Survey's modular finite-difference ground-water flow model (McDonald and Harbaugh, 1988). The result is a fully three-dimensional granular displacement model that is semi-independent from the ground-water flow equations used in MODFLOW. That is, transient values of hydraulic head (MODFLOW output) are not required to obtain the directional components of displacement, but they are used in the specification of the water-table boundary. The displacements are calculated independently from MODFLOW's numerical algorithms approximating the ground-water flow equation.

A detailed discussion of the theoretical and numerical developments are presented. Following these developments the model is compared with existing analytic solutions in two space dimensions. Additional three-dimensional simulations are presented and include a discussion of the development of the water-table boundary. Model limitations and assumptions are also discussed. Finally, the detailed model documentation is provided in the appendix for the three modules developed for this study: (1) a module to calculate the initial and ultimate bulk flux, (2) a module to calculate displacement and volume strain in each space dimension, and finally (3) a module to plot vectors of either displacement or bulk flux in order to analyze and review the large amount of data that is produced.

THEORETICAL DEVELOPMENT

Darcy-Gersevanov-Helm Law

Ground-water hydrologists typically ignore the movement of the granular matrix of the aquifer in their analysis of ground-water flow. Helm (1979, 1984, 1987) has shown that such limitations in the evaluation of matrix compression preclude the determination of directional components of the displacement field of solids. The approach taken in this report involves developing governing equations of granular movement in three dimensions and begins with Darcy's Law, which is expressed in vector form by the relation:

$$\vec{q} = -\bar{K}\nabla h \qquad (1)$$

where $\vec{q}$ is the specific discharge, $\bar{K}$ is the hydraulic conductivity tensor, and h is the hydraulic head. Gersevanov (1934) deduced that Darcy's law describes the flow of ground water relative to the skeletal matrix and should be written more completely as:

$$\vec{q} \equiv n(\vec{v}_w - \vec{v}_s) = -\bar{K}\nabla h \qquad (2)$$

where n is the porosity, $\vec{v}_w$ is the velocity of water, and $\vec{v}_s$ is the velocity of solids (solid phase of the aquifer). Biot (1941, 1955) independently deduced eq. 2 as being the correct expression of Darcy's law. On the basis of volume fraction, Helm (1984, 1987) defined the bulk flux for a saturated medium as:

$$\vec{q}_b = n\vec{v}_w + (1-n)\vec{v}_s. \qquad (3)$$

Using Gersevanov's generalization (eq. 2) with Helm's equation for the bulk flux (eq. 3) yields a new Darcian expression in terms of the velocity of solids and the bulk flux, referred to here as the Darcy-Gersevanov-Helm Law, namely:

$$\vec{q}_b = \vec{v}_s + \vec{q} = \vec{v}_s - \bar{K}\nabla h, \qquad (4)$$

or

$$\vec{v}_s - \bar{K}\nabla h = \vec{q}_b. \qquad (5)$$

Equation 5 is a simple yet powerful expression of Darcy's law describing both the motion of solids and interstitial fluid in an unconsolidated saturated aquifer. For a bulk volume V where no mass is produced or destroyed, Helm (1987, 1994) showed that the divergence of the bulk flux for a two-phase saturated porous media is expressed as

$$\frac{n}{\rho_w}\left[\frac{\partial \rho_w}{\partial t} + v_w \bullet \nabla \rho_w\right] + \frac{1-n}{\rho_s}\left[\frac{\partial \rho_s}{\partial t} + v_s \bullet \nabla \rho_s\right] + \nabla \bullet [n\hat{v}_w + (1-n)v_s] = 0 \qquad ‹6›$$

where ρ_w and ρ_s is the density of water and the density of the individual solids, respectively. The last bracketed term on the left side of eq. 6 is the definition of the bulk flux. If we assume that the individual solid grains are incompressible, that is $\hat{\rho}_s \equiv constant$, and that the fluid is also incompressible, that is $\hat{\rho}_w = constant$, then mass conservation for incompressible bulk flow yields

$$\nabla \bullet \vec{q}_b = 0. \qquad ‹7›$$

Equation 7 states that if the bulk flux is known along the boundary (such as at a well) it is known at all points within the aquifer. Note that $\vec{q}_b$ can be a function of time. In addition, eq. 7 implicitly states that if the divergence of the velocity of solids is nonzero, then the compressibility of the solid particles and interstitial water are much smaller than the compressibility of the skeletal matrix. This is a valid assumption for unconsolidated aquifers. Matrix compressibility for shallow sand-dominated aquifers is one to three orders of magnitude greater than the compressibility of water and is two to four orders of magnitude greater for clay-rich aquitards (Scott, 1963).

Bulk Flux

When a stress such as that caused by pumping is applied to an aquifer system the stress immediately responds as a body force on the incompressible constituent materials throughout the aquifer. At this initial instant of applied stress, $t = 0^+$, both fluids and solids move as a single incompressible mass toward the pumping well at the same velocity. This velocity is equal to q_b (defined by eq. 3) and is dependent upon the pumping rate and the distance to the point of interest from the pumping well The initial bulk flux may also be affected by the ratio of vertical to horizontal hydraulic conductivity. That is, anisotropy may affect the magnitude of the bulk flux for a given radial distance from the pumping well. The change in bulk flux due to anisotropies can be easily derived and is discussed in the following paragraphs.

Heterogeneities such as confining beds also affect the distribution of bulk flux. From eq. 5 one can deduce that once a stressed aquifer system reaches a new steady-state condition, that is after the solid matrix has come to rest, the final values of bulk flux are identical to the new steady-state distribution of specific discharge. These final or ultimate values of bulk flux will reflect any boundary and initial conditions that influence the final flow field of water and may therefore be different from the initial values of bulk flux. Empirical evidence suggests that the transition from the initial to ultimate bulk flux is rapid (Francis Riley, U.S. Geological Survey, oral commun., 1993) and occurs while the aquifer matrix remains physically in motion. This evidence originates from field measurements of water-level reversals in observation wells separated from a pumping well by an impermeable fault. The rise in head in the observation well occurs almost instantaneously from the inception of pumping. No ultimate hydraulic gradient exists in the part of the aquifer separated from the pumping well by the impermeable boundary. What does exist in this part of the aquifer is an initial strain field within the solid matrix resulting from the initial bulk flux. The transition from the initial to ultimate bulk flux is reflected in the head fluctuation observed in the observation well. Further research is needed to define the time dependency of this transient change in bulk flux from initial to ultimate values. The transient nature of bulk flux is beyond the scope of this study but numerical simulations suggest that the change may be dependent not only on time but also on hydraulic diffusivity. Only the initial and final bulk flux values are used in this study and, based on empirical evidence, the transition is considered to be nearly instantaneous. Figure 1 shows the relative magnitude and direction of initial and ultimate values of bulk flux in a heterogeneous system being pumped from a single well.

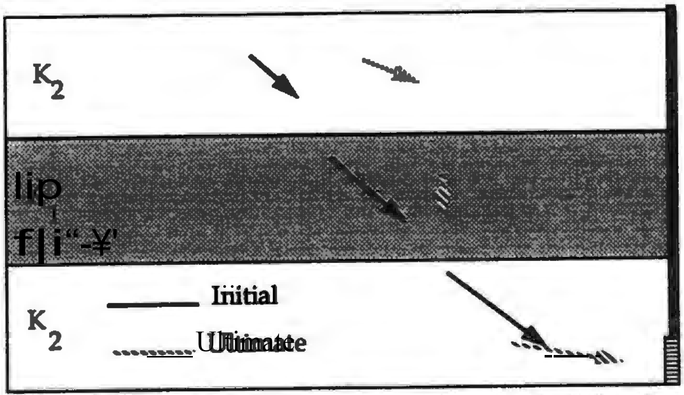


Figure 1. Schematic diagram illustrating how the initial and ultimate values of bulk flux may change for a heterogeneous aquifer system. Arrow lengths indicate relative magnitude of bulk flux. The shaded unit has low hydraulic conductivity relative to the units above and below it.

The ultimate bulk flux can be evaluated by determining the steady-state hydraulic head values that are produced from MODFLOW for a steady-state simulation of the aquifer being simulated. From these heads, the ultimate specific discharge or bulk flux values can be evaluated. The ultimate bulk flux is substituted for the initial bulk flux after approximately one minute (this is an arbitrary time that is used to establish an initial field of maximum velocity of solids). These ultimate values of bulk flux are then used throughout the remainder of the simulation.

In a contiguous three dimensional homogeneous, isotropic incompressible and undifferentiated media with a single pumping well defined as a point sink, mass balance requires that initially

$$Q = 4\pi r^2 q_b \tag{8}$$

where Q is the pumping rate and r is the radial distance from the well to a point of interest. For spherically symmetric flow

$$q_b = \frac{Q}{4\pi r^2}. \tag{9}$$

Because the bulk flux is known at the pumping well, the bulk flux can be determined at any location within the aquifer according to eq. 7. If we assume $K_r \gg K_z$, where the subscripts *r* and *z* refer to the radial and vertical space dimensions, respectively, then the hydraulic gradient migrates outward from the pumping well elliptically. The permeability ellipse is expressed as

$$\frac{b}{a} = \sqrt{\frac{K_z}{K_r}} \qquad (10)$$

where *a* is the length of the major axis and *b* is the length of the minor axis (fig. 2). Whether the aquifer is confined or unconfined does not matter at $t = 0^+$. At the initial instant when the well is turned on, the aquifer will behave as a confined aquifer until the first pressure transient reaches the water table. After this time the aquifer will behave as an unconfined aquifer. However, the initial bulk flux is evaluated at the moment the well is turned on. Because an expression for the bulk flux has been defined at the well it can now be evaluated directly for any point in the aquifer system.

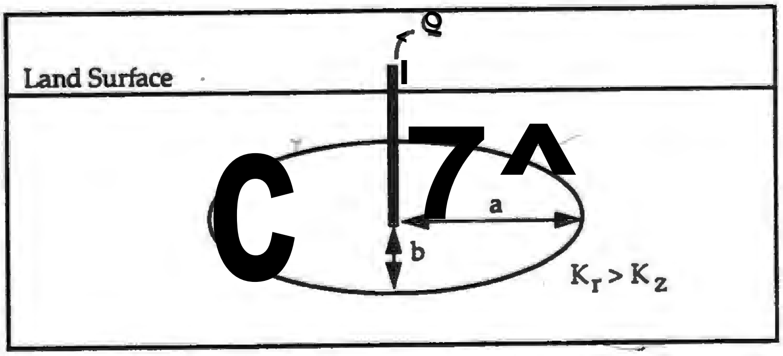


Figure 2. Schematic diagram illustrating a permeability ellipse with major axis a and minor axis b. The aquifer is being pumped at a constant rate Q.

The U.S. Geological Survey's modular ground-water flow model makes use of a rectangular finite-difference grid network. Hence, radial or spherical coordinates of initial bulk flux must be transformed to cartesian coordinates. The eccentricity of an ellipse is expressed as:

$$\tau = \frac{\sqrt{a^2 - b^2}}{a}. \qquad (11)$$

Neither *a* nor *b* are known explicitly; however, the ratio of a/b (or b/a) is known from eq. 10. Equation 11 can be expressed in terms of this ratio,

$$\tau = \left[1 - \left(\frac{b}{a}\right)^2\right]^{\frac{1}{2}} \qquad (12)$$

When the ellipse is aligned such that its rotation is about the major axis (aligned in the radial or horizontal direction) the ellipse is said to be oblate. The formula for an oblate spheroid is given by:

$$\frac{(x-x_o)^2}{a^2} + \frac{(y-y_o)^2}{a^2} + \frac{(z-z_o)^2}{b^2} = 1 \qquad (13)$$

This equation assumes that the aquifer system is transversely isotropic. That is $K_{xx} \equiv K_{yy}$. Now there are two equations and two unknowns so that *a* and *b* can be determined explicitly as follows:

$$a = \left[(x-x_o)^2 + (y-y_o)^2 + \frac{(z-z_o)^2}{1-\tau^2}\right]^{\frac{1}{2}} \qquad (14)$$

and,

$$b = a\sqrt{1-\tau^2}, \qquad (15)$$

where *x*, *y*, and z are the spatial locations of a point of interest in the aquifer system (equivalent to the column, row, and layer at a cell center within a finite-difference grid network). The variables x_o, y_o, and z_o represent the spatial locations of the pumping well. The surface area of an oblate spheroid is given by

$$S = 2\pi a^2 + \frac{\pi b^2}{\tau} \ln\left(\frac{1+\tau}{1-\tau}\right). \qquad (16)$$

Equation 9 written in vector form in cartesian coordinates is

$$\vec{q}_b = -\frac{Q}{S}\hat{e} \qquad (17)$$

where $\hat{e}$ represents a unit vector. An expression for the bulk flux can be written for each component direction as:

$$q_{bx} = \frac{-Q(x-x_o)}{Sr}\text{, and} \tag{18}$$

$$q_{by} = \frac{-Q(y-y_o)}{Sr}\text{, and} \tag{19}$$

$$q_{bz} = \frac{-Q(z-z_o)}{Sr}, \tag{20}$$

where r represents the radial distance from the well (the cell containing the well) to the cell of concern in cartesian coordinate space, which is defined as

$$r = [(x-x_o)^2 + (y-y_o)^2 + (z-z_o)^2]^{\frac{1}{2}}. \tag{21}$$

The bulk flux initial condition is now established for a point source or sink. These expressions can be applied for any number of point sources or sinks because in general,

$$q_{b\zeta_i} = \sum_{i=1}^{m} \frac{-Q_i(\zeta_i-\zeta_{oi})}{S_i r_i} \tag{22}$$

where m is the number of pumping wells and ζ_{oi} represents the location of the sink or source in the coordinate direction of concern, ζ_i is the current location within the grid system for the coordinate direction of concern, and i represents the current source or sink.

Equation 22 is valid except at the wellbore where the bulk flux approaches infinity when $x_i - x_{oi} = 0$ and $y_i - y_{oi} = 0$. The x and y components of bulk flux are automatically set to zero at the wellbore. At the wellbore where no horizontal component of bulk flux exists (and where r is set to unity), unusually large values of bulk flux in the z direction occur when $z_i - z_{oi} \neq 0$ (where q_{bx} and q_{by} are zero). To mitigate these problematic values of bulk flux, a five point average of the z component of bulk flux is used by calculating its value at each edge of the cell and at the cells center at the x and y cell location containing the well. This averaging of the z component of bulk flux along the wellbore (for $z_i - \hat{z}_{oi} \neq 0$) smooths the ultimate displacement values obtained by making the z component of displacement an average over the entire surface area of the cell instead of at a point.

Equations 16 through 22 are valid for any three dimensional setting. A modification to this general expression occurs when only a one-dimensional case is assumed with axial symmetry. In this case the resulting surface area is no longer a spheroid but is modeled rather as a cylinder with one layer. The cylindrical surface area allows no vertical component of bulk flux. That is, only horizontal components of flow and bulk flux are calculated and used in the calculation of horizontal displacement. This situation occurs when simulating a single confined aquifer (a single layer within a finite-difference model) with impermeable top and bottom. For this scenario, mass balance requires that the bulk flux be expressed for axially symmetric flow as follows,

$$q_b = \frac{Q}{2\pi r b} \qquad (23)$$

where b is the thickness of the aquifer in question. In cartesian coordinates the bulk flux for each component direction can be defined as follows:

$$q_{bx} = \frac{Q(x-x_o)}{2\pi r^2 b}, \text{and} \qquad (24)$$

$$q_{by} = \frac{Q(y-y_o)}{2\pi r^2 b}. \qquad (25)$$

For this one-dimensional case with axial symmetry bulk flux does not vary with time. Both the initial and final values are given by eqs. 23 and 24. The directional components of bulk flux are used in the governing equation for the displacement field of solids developed in the next section.

Governing Equations for the Displacement Field of Solids

The governing equation for granular displacement can be developed from equations of motion without invoking mass balance. As seen in the proceeding section, mass balance is used essentially for evaluating $\vec{q}_b$.

We begin with the expression for Darcy's law in terms of bulk flux (eq. 5). The ensuing discussion describes primarily the transformation of dependent variables from the hydraulic head to the displacement field of solids.

In a fixed coordinate system, Hubbert (1940) separated the total hydraulic head, h, into a pressure head and elevation head assuming irrotational flow as:

$$h = \frac{p}{\rho_w g} + z \qquad ‹26›$$

where p is pressure, ρ_w is the density of the interstitial water, g is the gravity constant, and z is the elevation from some known datum. Taking the gradient of each side of this expression and assuming that the density of water is a constant, eq. 26 can be written as

$$\nabla h = \frac{\nabla p}{\rho_w g} + \hat{k} \qquad ‹27›$$

S

where $\hat{k}$ is a unit normal vector in the vertical direction. Substituting this expression into eq. 5 yields

$$\vec{v}_s - \bar{K}\left[\frac{\nabla p}{\rho_w g} + \hat{k}\right] = \vec{q}_b. \qquad ‹28›$$

Pressure is also related to the total and effective stress according to Terzaghi (1960, p. 45), namely

$$\bar{\sigma} = \bar{\sigma}' + Ip, \qquad ‹29›$$

where $\bar{\sigma}$ and $\bar{\sigma}'$ represent the total and effective stress tensors, respectively, and I is the identity matrix. The mean normal values for total and effective stress, and pressure, can be obtained from the first invarient (trace) of these tensors to yield

$$\sigma_m = \sigma'_m + p, \qquad ‹30›$$

where σ_m and σ'_m are the scalar quantities of the mean total and mean effective stresses, respectively. Taking the gradient of eq. 30 and, rearranging and substituting this expression into eq. 28 yields

$$\vec{v}_s - \bar{K}\left[\hat{k} + \frac{\nabla(\sigma_m - \sigma'_m)}{\rho_w g}\right] = \vec{q}_b \ . \qquad ‹31›$$

The total and effective stress tensors can be decomposed into an initial hydrostatic stress condition and an incremental stress event, namely

$$\sigma_m = \delta\sigma_m + \sigma_{mo} \qquad ‹32›$$

and

$$\sigma'_m = \delta\sigma'_m + \sigma'_{mo} \qquad ‹33›$$

where σ_{mo} and σ'_{mo} represent the initial unstrained mean total and mean effective stress condition, respectively; whereas $\delta\sigma_m$ and $\delta\sigma'_m$ are the cumulative stress increments of interest.

If we assume an elastic or Hookian stress-strain constitutive relation for the stress increment of interest, then

$$\nabla\delta\sigma'_m = -\left[\frac{1}{\alpha_{xx}}\frac{\partial\varepsilon}{\partial x}\hat{i} + \frac{1}{\alpha_{yy}}\frac{\partial\varepsilon}{\partial y}\hat{j} + \frac{1}{\alpha_{zz}}\frac{\partial\varepsilon}{\partial z}\hat{k}\right] \qquad ‹34›$$

where ε is the volume strain, α_{xx}, α_{yy}, and α_{zz} are the bulk compressibilities of the skeletal matrix in the *x*, *y*, and *z* principal stress directions. Because transverse isotropy is assumed with respect to matrix compressibility, $\alpha_{xx} = \alpha_{yy} \neq \alpha_{zz}$. The minus sign is included because ε is positive for expansion and $\delta\sigma'$ is positive for compression. Substituting these relations into eq. 31 results in the following expression;

$$\vec{v}_s - \bar{K}\left[\hat{k} + \frac{\nabla(\delta\sigma_m + \sigma_{mo} - \sigma'_{mo})}{\rho_w g} + \frac{1}{\rho_w g}\left(\frac{1}{\alpha_{xx}}\frac{\partial\varepsilon}{\partial x}\hat{i} + \frac{1}{\alpha_{yy}}\frac{\partial\varepsilon}{\partial y}\hat{j} + \frac{1}{\alpha_{zz}}\frac{\partial\varepsilon}{\partial z}\hat{k}\right)\right] = \vec{q}_b \ . \qquad ‹35›$$

Total volume strain, ε, is related to the displacement field of solids, $\vec{u}_s$, as

$$\varepsilon = \nabla\cdot\vec{u}_s \ . \qquad ‹36›$$

By rearranging terms, rewriting the velocity of solids as the time derivative of the displacement field, assuming homogeneous isotropic skeletal compressibility, and incorporating eq. 36, the governing equation now assumes the form:

$$\frac{d\vec{u}_s}{dt} - \frac{\bar{K}}{\rho_w g \bar{\bar{\alpha}}}[\nabla(\nabla\bullet\vec{u}_s)] = \vec{q}_b + \bar{K}\left[\hat{k} + \frac{\nabla(\sigma_{mo} - \sigma'_{mo})}{\rho_w g}\right] + \frac{\bar{K}\nabla\delta\sigma_m}{\rho_w g}, \qquad \langle 37 \rangle$$

where

$$\bar{\alpha} = \frac{1}{\lambda + 2G}, \qquad \langle 38 \rangle$$

and λ and G are Lame's constants.

Recognizing that the bracketed term in the right hand side of eq. 37 is simply the initial hydrostatic gradient of hydraulic head (which equals zero), the fully three-dimensional governing equation for the displacement of solids can be written more simply as

$$\frac{d\vec{u}_s}{dt} - \frac{\bar{K}}{\rho_w g \bar{\bar{\alpha}}}[\nabla(\nabla\bullet\vec{u}_s)] = \vec{q}_b + \frac{\bar{K}\nabla\delta\sigma_m}{\rho_w g}, \qquad \langle 39 \rangle$$

The last term on the right hand side of eq. 38 can be neglected if we assume that the change in the gradient of mean total stress is small. Jacob (1950) assumes $\delta\sigma_m$ itself is negligibly small. Equation 37 is essentially the same expression as given in Helm (1987, eq. 16) and can now be written in simplified form as

$$\frac{d\vec{u}_s}{dt} - \frac{\bar{K}}{\rho_w g \bar{\bar{\alpha}}}[\nabla(\nabla\bullet\vec{u}_s)] = \vec{q}_b. \qquad \langle 40 \rangle$$

In place of assuming

$$\nabla(\delta\sigma_m)\left[\equiv \frac{1}{3}\frac{\partial}{\partial x_j}(\delta\sigma_{kk})\right] = 0 \qquad \langle 41 \rangle$$

as was done to go from eq. 38 to eq. 39, we now assume stress equilibrium (Biot, 1941, 1955), namely

$$\frac{\partial \sigma_{ij}}{\partial X_j} = 0. \tag{42}$$

The resulting equation of motion for an isotropic homogeneous aquifer matrix (Helm, written commun., 1994) is

$$\frac{d\vec{u}_s}{dt} - \frac{\bar{K}}{\rho_w g \alpha'} \left[\nabla \left(\nabla \cdot \vec{u}_s \right) \right] = \vec{q}_b \tag{43}$$

where

$$\alpha' = \frac{3}{3\lambda + 2G} \tag{44}$$

In other words, for an isotropic homogeneous poro-elastic aquifer, assuming eq. 41 yields eq. 38 as a physical interpretation of specific storage per unit weight of water, whereas assuming eq. 42 yields eq. 44.

Equation 38 is similar in form to the ground-water flow equation. Both are parabolic differential equations. The primary difference is that the principle unknown quantity $\vec{u}_s$, the displacement field of solids, is a vector. The principal unknown in the ground-water flow equation is hydraulic head, a scalar quantity. One way to simplify eq. 39 so that the dependent variable can be treated as a scalar variable is to decouple the governing equation into three equations, one expression for each component direction. By decoupling eq. 40, the three expressions are written as

$$\frac{du_x}{dt} - \frac{K_{xx}}{\rho_w g} \left[\frac{1}{\alpha_{xx}} \cdot \frac{\partial^2 u_x}{\partial x^2} + \frac{1}{\alpha_{yy}} \cdot \frac{\partial^2 u_y}{\partial x \partial y} + \frac{1}{\alpha_{zz}} \cdot \frac{\partial^2 u_z}{\partial x \partial z} \right] = q_{bx} \, , \tag{45}$$

$$\frac{du_y}{dt} - \frac{K_{yy}}{\rho_w g} \left[\frac{1}{\alpha_{xx}} \cdot \frac{\partial^2 u_x}{\partial x \partial y} + \frac{1}{\alpha_{yy}} \cdot \frac{\partial^2 u_y}{\partial y^2} + \frac{1}{\alpha_{zz}} \cdot \frac{\partial^2 u_z}{\partial y \partial z} \right] = q_{by} \, , \tag{46}$$

and

$$\frac{du_z}{dt} - \frac{K_{zz}}{\rho_w g} \left[\frac{1}{\alpha_{xx}} \cdot \frac{\partial^2 u_x}{\partial x \partial z} + \frac{1}{\alpha_{yy}} \cdot \frac{\partial^2 u_y}{\partial y \partial z} + \frac{1}{\alpha_{zz}} \cdot \frac{\partial^2 u_z}{\partial z^2} \right] = q_{bz} \, . \tag{47}$$

The left-hand side of equations 45-47 can be further simplified if the compressibility of the aquifer is transversely isotropic, that is, $\alpha_h = \alpha_{xx} = \alpha_{yy}$. The specific storage in the horizontal direction can now be written as $S_{sh} = \rho_w g \alpha_h$, and the specific storage in the vertical direction can likewise be written as $S_{sv} = \rho_w g \alpha_{zz}$, where we have assumed incompressible bulk constituents. Typically, hydrologists assume a compressible interstitial fluid. However, because the matrix compressibility is generally at least two orders of magnitude greater than fluid compressibility, the specific storage is not significantly affected by the omission of fluid compressibility within the expression for bulk flux. The set of equations is now stated simply as:

$$\frac{du_x}{dt} - \frac{K_{xx}}{S_{sh}}\left[\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_y}{\partial x \partial y}\right] - \frac{K_{xx}}{S_{sv}}\left[\frac{\partial^2 u_z}{\partial x \partial z}\right] = q_{bx} \ , \qquad ‹48›$$

$$\frac{du_y}{dt} - \frac{K_{yy}}{S_{sh}}\left[\frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_x}{\partial y \partial x}\right] - \frac{K_{yy}}{S_{sv}}\left[\frac{\partial^2 u_z}{\partial y \partial z}\right] = q_{by}, \qquad ‹49›$$

and

$$\frac{du_z}{dt} - \frac{K_{zz}}{S_{sh}}\left[\frac{\partial^2 u_y}{\partial z \partial y} + \frac{\partial^2 u_x}{\partial z \partial x}\right] - \frac{K_{zz}}{S_{sv}}\left[\frac{\partial^2 u_z}{\partial z^2}\right] = q_{bz} \ . \qquad ‹50›$$

The left-hand side of each equation (eqs. 48-50) contains three dependent variables. The right-hand side contains the directional components of bulk flux which are known. Each component direction contains cross-product derivatives of the remaining space dimensions. The x, y, and z component directions of displacement are required for each expression. The temptation is to assume the cross-product derivative terms in each expression are negligible. This would yield three second order diffusion equations in one space dimension that would be easy to solve directly. However, the cross-product derivatives can not be ignored as the following discussion will show. For the purposes of example, only two dimensions (*x* and *y*) are considered in the following discussion.

Total volume strain is an invariant and can be written for axially symmetric displacement in two dimensions as

$$\varepsilon \equiv \varepsilon_{rr} + \varepsilon_{\theta\theta} \quad , \qquad \langle 51 \rangle$$

where the normal strains are

$$\varepsilon_{rr} = \frac{\partial u_r}{\partial r} \qquad \langle 52 \rangle$$

and

$$\varepsilon_{\theta\theta} = \frac{u_r}{r} \qquad \langle 53 \rangle$$

Figure 3 illustrates these normal components of strain. The tail length of the arrows schematically represents the distance the aquifer matrix displaces horizontally during a unit time interval. The radial strain occurs along a flow line and results when, at a specified time of interest, a front grain has moved either a greater or lesser cumulative distance toward the pumping well than the neighboring grain behind it. If the front grain has moved farther extension or extensional strain has occurred, if it has not moved as far as the back grain compression or compressional strain has occurred. In fig. 3 radial compression occurs inside the circle, while radial extension occurs outside the circle. The circle itself represents a circumference of zero radial strain. That is, the front grain and the neighboring back grain move the same distance along the flow path during the specified time interval. In addition to radial strain, tangential strain also occurs as a result of grains moving closer together along two converging flow lines. Tangential strain is everywhere compressive in response to pumping.

In cartesian coordinates the relation within brackets expressed in eq. 48

$$\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_y}{\partial x \partial y}$$

can be written in terms of normal strains following eq. 36 as

$$\frac{\partial}{\partial x}(\varepsilon_{xx}) + \frac{\partial}{\partial x}(\varepsilon_{yy}) ,$$

where

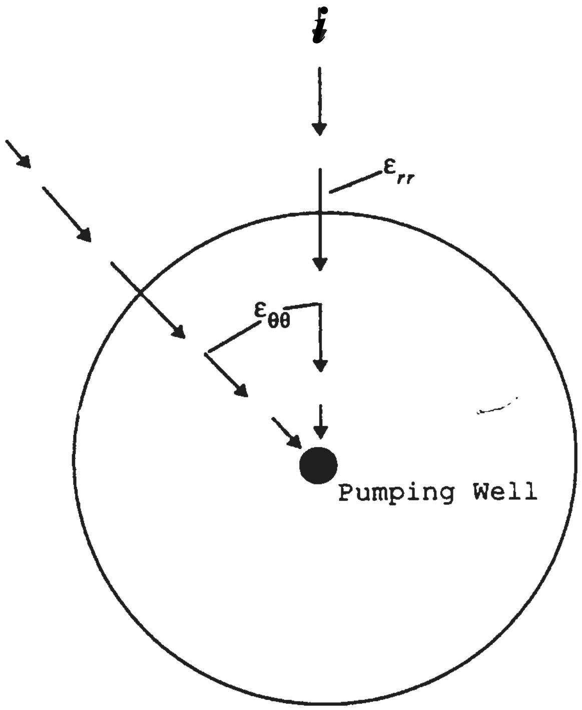


Figure 3. Schematic diagram illustrating the components of radial and tangential strains. The arrows indicate direction of movement, their tails indicate relative magnitude of strain displacement. Compressional radial strain occurs inside the circle, extensional radial strain occurs outside the circle.

$$\varepsilon = \varepsilon_{xx} + \varepsilon_{yy} \tag{54}$$

and where ε_{xx} and ε_{yy} represent the cartesian coordinate equivalent of the radial and tangential strain components, respectively. Thus, because the coordinate directions do not necessarily coincide with the flow lines, the cross-product derivatives of eqs. 48-50 can not be neglected. Equations 48-50 represent the governing equations used to simulate displacement in the granular displacement model developed in the next section.

form in the same manner; hence, the numerical approximation to only one space dimension (eq. 50) is given below. The other two dimensions can be easily written on the basis of the space dimension provided. The central difference approximation to eq. 50 for all interior cells (cells not at the boundary) in the z direction is:

$$\frac{K_{zz}}{S_{sv}}\left(\frac{2\,(uz_{j,i,k+1}-uz_{j,i,k})^{m+1}}{\Delta z_k(\Delta z_{k+1}+\Delta z_k)}-\frac{2\,(uz_{j,i,k}-uz_{j,i,k-1})^{m+1}}{\Delta z_k(\Delta z_{k-1}+\Delta z_k)}\right)+\frac{K_{zz}}{S_{sh}}\left(\frac{-uy^{m+1}_{j,i+1,k+1}}{DVC(DCC,DCF)}-\right.$$

$$\frac{uy^{m+1}_{j,i,k+1}}{DVC(DCF,-DCB)}+\frac{uy^{m+1}_{j,i-1,k+1}}{DVC(DCC,DCB)}+\frac{uy^{m+1}_{j,i+1,k-1}}{DVC(DCC,DCF)}-\frac{uy^{m+1}_{j,i,k-1}}{DVC(-DCF,DCB)}-$$

$$\frac{uy^{m+1}_{j,i-1,k-1}}{DVC(DCC,DCB)}-\frac{ux^{m+1}_{j+1,i,k+1}}{DVC(DRC,DRF)}-\frac{ux^{m+1}_{j,i,k+1}}{DVC(DRF,-DRB)}+\frac{ux^{m+1}_{j-1,i,k+1}}{DVC(DRC,DRB)}$$

$$\left.+\frac{ux^{m+1}_{j+1,i,k-1}}{DVC(DRC,DRF)}-\frac{ux^{m+1}_{j,i,k-1}}{DVC(-DRF,DRB)}-\frac{ux^{m+1}_{j-1,i,k-1}}{DVC(DRC,DRB)}\right)=-qbz_{j,i,k}+$$

$$\frac{uz^{m+1}_{j,i,k}-uz^{m}_{j,i,k}}{\Delta t} \tag{55}$$

where,

ux, *uy*, and *uz* are the displacement in the *x*, *y*, and z directions, respectively;

qbz is the component of bulk flux in the z direction;

j, *i*, and *k* represent the grid spacing in the x, y, and z directions, respectively;

m is the time-step indicator;

Δt is the length of the current time step;

DRC $= \Delta x_j + 0.5\,(\Delta x_{j+1} + \Delta x_{j-1})$;

DCF $= \Delta y_i + 0.5\,(\Delta y_{i+1} + \Delta y_i)$;

DCC $= \Delta y_i + 0.5\,(\Delta y_{i+1} + \Delta y_{i-1})$;

DCB $= \Delta y_i + 0.5\,(\Delta y_i + \Delta y_{i-1})$;

DVF $= \Delta z_k + 0.5\,(\Delta z_{k+1} + \Delta z_k)$;

DVC $= \Delta z_k + 0.5\,(\Delta z_{k+1} + \Delta z_{k-1})$;

DVB $= \Delta z_k + 0.5\,(\Delta z_k + \Delta z_{k-1})$.

The comma delimiter in the denominator is synonymous to the word "or". Only one of the two terms in the parentheses is used in the numerical approximation. The term used depends upon whether a boundary has been encountered in the *y* or *x* directions. Boundary conditions are discussed in the following section.

Equation 55 is second-order correct in z but is only first-order correct with respect to time. To make the time derivative second order correct and unconditionally stable the Crank-Nicolson scheme (Remson and others, 1971) is applied to eq. 50 (and synonymous expressions for the *y* and *x* space dimensions). This technique divides the time step into two parts so that the finite-difference equations are essentially evaluated at the *m+1/2* and the *m+1* time intervals. This is accomplished by averaging the second-order space derivatives at the *m* and *m+1* time steps. Increased accuracy is obtained by such an approximation. Equation 50 written with the Crank-Nicolson scheme results in the expression:

$$\frac{K_{zz}}{S_{sv}}\left([A]\,uz^{m+1}_{j,i,k-1} - [B]\,uz^{m+1}_{j,i,k} + [C]\,uz^{m+1}_{j,i,k+1} + [A]\,uz^{m}_{j,i,k-1} - [B]\,uz^{m}_{j,i,k} +\right.$$

$$\left.[C]\,uz^{m}_{j,i,k+1}\right) + \frac{K_{zz}}{S_{sh}}\left(-[D]\,uy^{m+1}_{j,i+1,k+1} - [E]\,uy^{m+1}_{j,i,k+1} + [F]\,uy^{m+1}_{j,i-1,k+1} +\right.$$

$$+ [G]\,uy^{m+1}_{j,i+1,k-1} - [H]\,uy^{m+1}_{j,i,k-1} - [I]\,uy^{m+1}_{j,i-1,k-1} - [D]\,uy^{m}_{j,i+1,k+1} - [E]\,uy^{m}_{j,i,k+1} +$$

$$[F]\,uy^{m}_{j,i-1,k+1} + [G]\,uy^{m}_{j,i+1,k-1} - [H]\,uy^{m}_{j,i,k-1} - [I]\,uy^{m}_{j,i-1,k-1} - [J]\,ux^{m+1}_{j+1,i,k+1}$$

$$- [K]\,ux^{m+1}_{j,i,k+1} + [L]\,ux^{m+1}_{j-1,i,k+1} + [M]\,ux^{m+1}_{j+1,i,k-1} - [N]\,ux^{m+1}_{j,i,k-1} - [O]\,ux^{m+1}_{j-1,i,k-1}$$

$$- [J]\,ux^{m}_{j+1,i,k+1} - [K]\,ux^{m}_{j,i,k+1} + [L]\,ux^{m}_{j-1,i,k+1} + [M]\,ux^{m}_{j+1,i,k-1}$$

$$\left.- [N]\,ux^{m}_{j,i,k-1} - [O]\,ux^{m}_{j-1,i,k-1}\right) \equiv \frac{uz^{m+1}_{j,i,k}}{\Delta t} - \frac{uz^{m}_{j,i,k}}{\Delta t} - qbz_{j,i,k} \qquad (56)$$

where,

$A=1/(\Delta z_k(\Delta z_{k-1}+\Delta z_k));$

$C=1/(\Delta z_k(\Delta z_{k+1}+\Delta z_k));$

$B=A+C;$

$D=1/(2DVC(DCC,DCF));$

$E=1/(2DVC(DCF,-DCB));$

$F=1/(2DVC(DCC,DCB));$

$G=1/(2DVC(DCC,DCF));$

$H=1/(2DVC(-DCF,DCB));$

$I=1/(2DVC(DCC,DCB));$

$J=1/(2DVC(DRC,DRF));$

$K=1/(2DVC(DRC,-DRB));$

$L=1/(2DVC(DRC,DRB));$

$M=1/(2DVC(DRC,DRF));$

$N=1/(2DVC(-DRF,DRB));$

$O=1/(2DVC(DRC,DRB)).$

The equations for the *x* and y space dimensions (eqs. 48 and 49) are written in similar fashion.

Boundary Conditions

After the general expressions of the finite-difference equations have been developed, boundary conditions need to be applied. For the granular displacement model two types of boundary conditions have been implemented in terms of displacement: (1) a zero-displacement (or zero solid velocity) boundary, and (2) a water table boundary which is used for all quasi and fully three-dimensional simulations. Other boundaries such as constant or general head are inherently included in the ultimate bulk flux terms.

The zero-displacement boundary is used at the bottom (base of model grid) and sides (lateral extent) of model grid. Because of the type of settings (namely, sedimentary basins) in which the model would most likely be applied, this boundary is a logical choice. A zero-displacement boundary refers to a cell edge (for one space dimension) where there is no granular displacement in a direction orthogonal to the boundary. Such a condition would likely occur at a bedrock contact. Most basin models are designed to extend to the basin fill-bedrock contact; hence the zero-displacement boundary is a natural choice for the perimeter of the modeled area. However, the program allows the user to specify zero-displacement cells if bedrock zones are to be included in the model. Orthogonal moving lateral or bottom boundaries have not yet been coded into the program.

Zero-displacement boundaries are implemented perpendicular to the cell edge representing the outer perimeter of the grid. This is accomplished by using a central-difference scheme with image theory in order to specify the zero displacement at the cell boundary and not at the cell center. For example, if a zero-displacement boundary is encountered in the *x* direction, the dependent variable of the image cell become the negative value of the *x* displacement in the active cell adjacent to the boundary as follows:.

$ux_0 = -ux_1$

and

$ux_{NCOL+1} = -ux_{NCOL}$,

where *NCOL* is the total number of columns in the grid. Because the nodes are block centered, this approach is needed to establish a zero-displacement boundary at the *j=1/2* and *j=NCOL+1/2* grid locations (*j* represents the column number). Thus eq. 56 is rewritten in the program to include these boundary conditions when *j=1* or *j=NCOL* (or for the boundary grid cell in the x direction). Similar expressions are used for the numerical expression of the governing equation in the *y* and *z* directions.

The fact that a boundary has zero displacement in the x direction does not mean that the *y* or *z* component of displacement is zero. In fact, the tangential components of strain (*y* and *z*) are not zero at the boundaries when evaluating displacement in the *x* direction, unless a similar zero-displacement boundary is encountered in the y or z principal directions. Thus, it is common for granular movement to occur parallel to a nonmoving boundary.

In order for the zero displacement boundary to also be an impermeable boundary (that is, $q=0$), the bulk flux would also have to be zero at the boundary according to eq. 5. This type of boundary as well as any heterogeneity is inherently included in the ultimate bulk flux values that supersede the initial values of bulk flux. Because these ultimate values are identical to the values of steady-state specific discharge, they contain all boundary conditions used within MODFLOW. Hence the ultimate bulk flux values also contain all the hydraulic-type boundaries of MODFLOW.

The second type of displacement boundary condition included in the model is a water table or free water surface. To adequately approximate this boundary two assumptions must be made. The first assumption is that the observed change of hydraulic head at the water table is a measure of the vertical velocity of a particle of water that lies on the water table and moves relative to the local skeletal frame, namely

$$\frac{dh}{dt} = v_{wz} - v_{sz} \qquad ‹57›$$

where z refers to the water table. The second assumption is that based on the standard definition of specific discharge as a relative velocity term (see eq. 2). In the vertical direction the expression is

$$q_z = n(v_{wz} - v_{sz}), \qquad ‹58›$$

where in this water-table case n is the specific yield representing the fraction of water that is drained from the overlying aquifer. Equations 57 and 58 can be substituted into eq. 4 to yield the final expression for the water table in the z direction

$$\frac{du_z}{dt} - n\frac{dh}{dt} = q_{bz} \qquad ‹59›$$

Equation 59 can be expressed numerically as follows:

$$uz_{j,i,k}^{m+1} = qbz_{j,i,k}\Delta t + n\,(h_{j,i,k}^{m+1} - h_{j,i,k}^{m}) + uz_{j,i,k}^{m} \qquad ‹60›$$

where drawdown is positive because displacement is positive downward and n is the specific yield. This expression is used only for the z direction in the topmost active layer at the water table. Because block-centered nodes are used the displacement is typically calculated at the center of the cell of question. In order to simulate the total displacement accurately along the water table and to apply the boundary condition given in eq. 60 appropriately, the finite-difference expression needs to be evaluated at the water table To accomplish this, the cell center for the topmost active layer of cells is calculated at the water table. In effect, the cell center is moved up by one half the thickness of the topmost active layer.

Solution Technique

Once the numerical expressions for all interior and boundary cells have been written, a suitable solution algorithm must be used to solve the set of equations that is established for each cell in the finite-difference network. Numerous methods have been developed for solving parabolic finite-difference equations in two and three dimensions. However, virtually all the literature in the physical sciences describes solving for a scalar dependent variable. Vector dependent variables, such as displacement, create complex situations in which standard techniques become invalid or essentially intractable numerically. Equation 56 (and corresponding equations for the x and y directions) contain three dependent variables and three equations and where both the initial and ultimate values of bulk flux are known. With the included boundary conditions this becomes a well posed problem but typical solution techniques using banded matrices become so large for this type of problem that it becomes unwieldy.

The approach taken here is to solve each directional component of displacement independently from the other two directions. In this way, the component directions not being solved for are held constant. A successive relaxation (SOR) solution scheme with Chebychev acceleration is used to solve the directional components of displacement at each model cell location. After each displacement within each component direction has been solved for and convergence met, the three independent equations are solved collectively as a whole. If convergence is not met for this outward iteration loop, each directional component equation is solved independently again. This process is repeated until outer convergence is met. The procedure is referred to here as a dual-loop SOR technique.

In the z direction the SOR algorithm can be written as:

$$uz^{n+1}_{j,i,k} = uz^{n}_{j,i,k} - \omega \cdot \frac{\zeta_{j,i,k}}{\Phi} \qquad ‹61›$$

where n is the iteration number, ω is the relaxation parameter, $\zeta_{j,i,k}$ is the residual at a given cell of concern, and Φ is the coefficient of the dependent variable of concern; that is, $uz^{n+1}_{j,i,k}$. The equations of the SOR algorithm are identical for the dependent variable in the x and y directions.

The residual can be easily obtained from eq. 56 by rearranging the expression as follows:

$$\zeta_{j,i,k} = \frac{K_{zz}\Delta t}{S_{sv}} \{ ([A]\, uz^{m+1}_{j,i,k-1} - [B]\, uz^{m+1}_{j,i,k} + [C]\, uz^{m+1}_{j,i,k+1} + [A]\, uz^{m}_{j,i,k-1} - [B]\, uz^{m}_{j,i,k} +$$

$$[C]\, uz^{m}_{j,i,k+1}) + \frac{S_{sv}}{S_{sh}} (- [D]\, uy^{m+1}_{j,i+1,k+1} - [E]\, uy^{m+1}_{j,i,k+1} + [F]\, uy^{m+1}_{j,i-1,k+1}$$

$$+ [G]\, uy^{m+1}_{j,i+1,k-1} - [H]\, uy^{m+1}_{j,i,k-1} - [I]\, uy^{m+1}_{j,i-1,k-1} - [D]\, uy^{m}_{j,i+1,k+1} - [E]\, uy^{m}_{j,i,k+1} +$$

$$[F]\, uy^{m}_{j,i-1,k+1} + [G]\, uy^{m}_{j,i+1,k-1} - [H]\, uy^{m}_{j,i,k-1} - [I]\, uy^{m}_{j,i-1,k-1} - [J]\, ux^{m+1}_{j+1,i,k+1}$$

$$- [K]\, ux^{m+1}_{j,i,k+1} + [L]\, ux^{m+1}_{j-1,i,k+1} + [M]\, ux^{m+1}_{j+1,i,k-1} - [N]\, ux^{m+1}_{j,i,k-1} - [O]\, ux^{m+1}_{j-1,i,k-1}$$

$$- [J]\, ux^{m}_{j+1,i,k+1} - [K]\, ux^{m}_{j,i,k+1} + [L]\, ux^{m}_{j-1,i,k+1} + [M]\, ux^{m}_{j+1,i,k-1}$$

$$- [N]\, ux^{m}_{j,i,k-1} - [O]\, ux^{m}_{j-1,i,k-1}) \} + qbx_{j,i,k}\Delta t + ux^{m}_{j,i,k} - ux^{m+1}_{j,i,k}. \qquad ‹62›$$

The coefficient, Φ, then becomes

$$\Phi = 1 + \frac{K_{zz}\Delta t}{S_{sv}} [B]. \qquad ‹63›$$

To hasten convergence, Chebyshev acceleration is implemented as outlined by Press and others (1992 p. 860). In this technique odd-even ordering is used. This process involves dividing the grid into odd and even cells. At each half iteration, n=1/2, the odd cells are updated. Then during the next half iteration sweep the even cells are updated with the newly calculated odd values. At each half sweep the relaxation parameter is updated according to the following prescription:

$$\omega^0 = 1$$

$$\omega^{1/2} = 1/(1-\rho_{Jacobi}^2/2)$$

$$\omega^{n+1/2} = 1/(1-\rho_{Jacobi}^2\omega^n/4) \quad \langle 64\rangle$$

$$\omega^\infty \rightarrow \omega_{optimal}$$

where ρ_{Jacobi}^2 is the spectral radius of the Jacobi iteration. The spectral radius is a number ranging from zero to one. A formal discussion of the approximation of this value for a given problem is given by Press and others (1992) and Remson and others (1971). For most examples illustrated in this study the optimum spectral radius is determined to be approximately 0.998 for all component directions.

The SOR iterative method was tested on a one dimensional problem with a direct solution technique. The SOR method produced nearly identical values of displacement as the direct method. One drawback to the SOR method as opposed to other possible iterative methods is that it is not very efficient for large problems. However, because of the vector dependent variable, it is possible that all iterative solution techniques would be slow to converge.

A generalized flow chart is outlined in fig. 4 showing the formulation and solution scheme used in the computer program for calculation of directional components of displacement.

SIMULATION OF GRANULAR MOVEMENT IN TWO DIMENSIONS

Model Evaluation

The numerical model must be analyzed to determine whether it produces accurate results for a given set of initial conditions and aquifer properties. This is accomplished by comparing the simulation results with analytic solutions of displacement developed for the Theis aquifer. The Theis-type aquifer is a one-dimensional radial or two dimensional cartesian coordinate system where vertical flow is ignored and the aquifer top and bottom is completely impermeable. Analytic solutions have been developed by Helm (1994) for granular displacement in a Theis-type aquifer that is pumped at a constant rate. In a contiguous unconsolidated media where the aquifer is confined and of infinite radial extent, mass balance requires that the bulk flux of incompressible constituent materials responds to a volume rate of discharge as

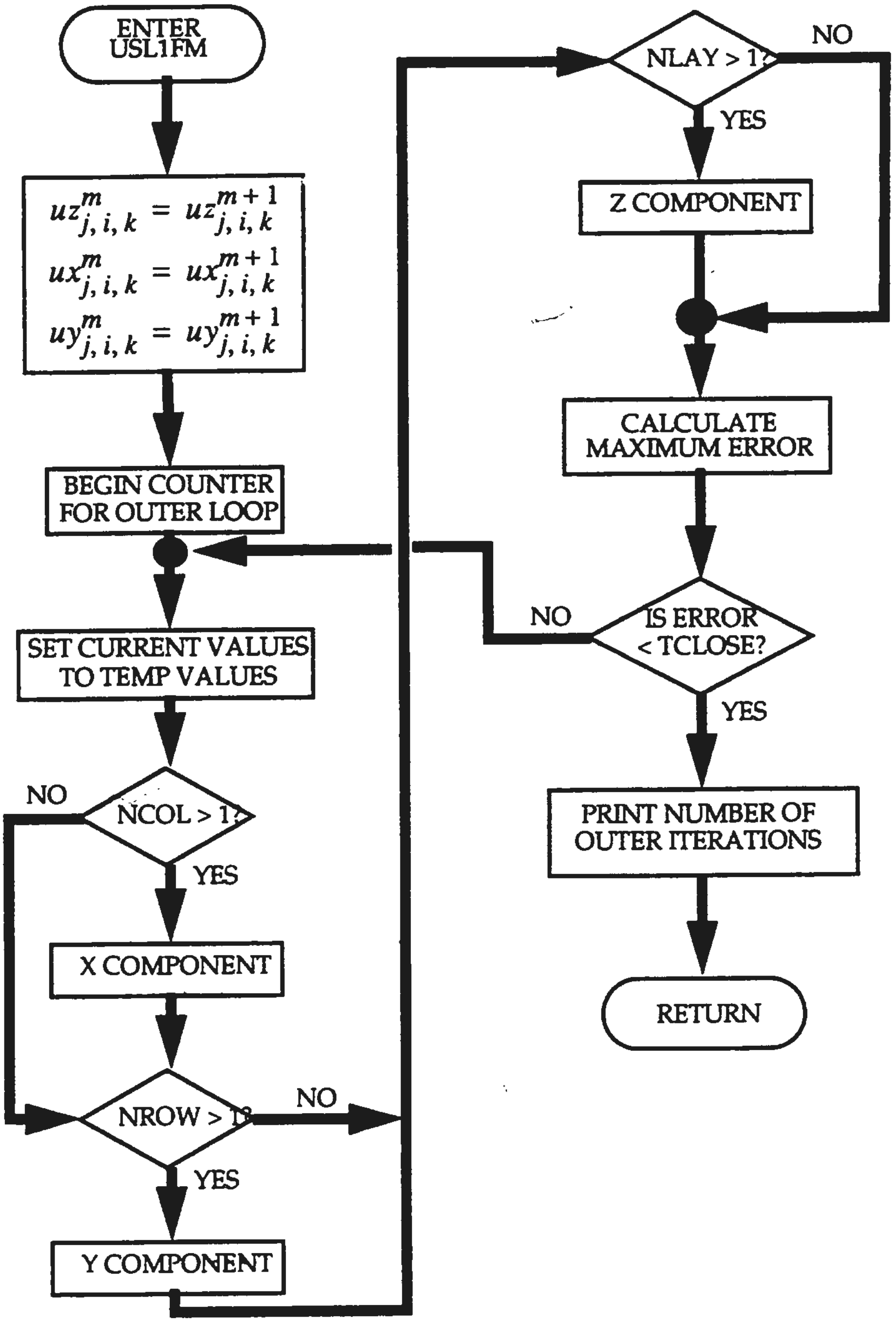
$$Q = 2\pi rb\vec{q}_b \qquad \langle 65 \rangle$$

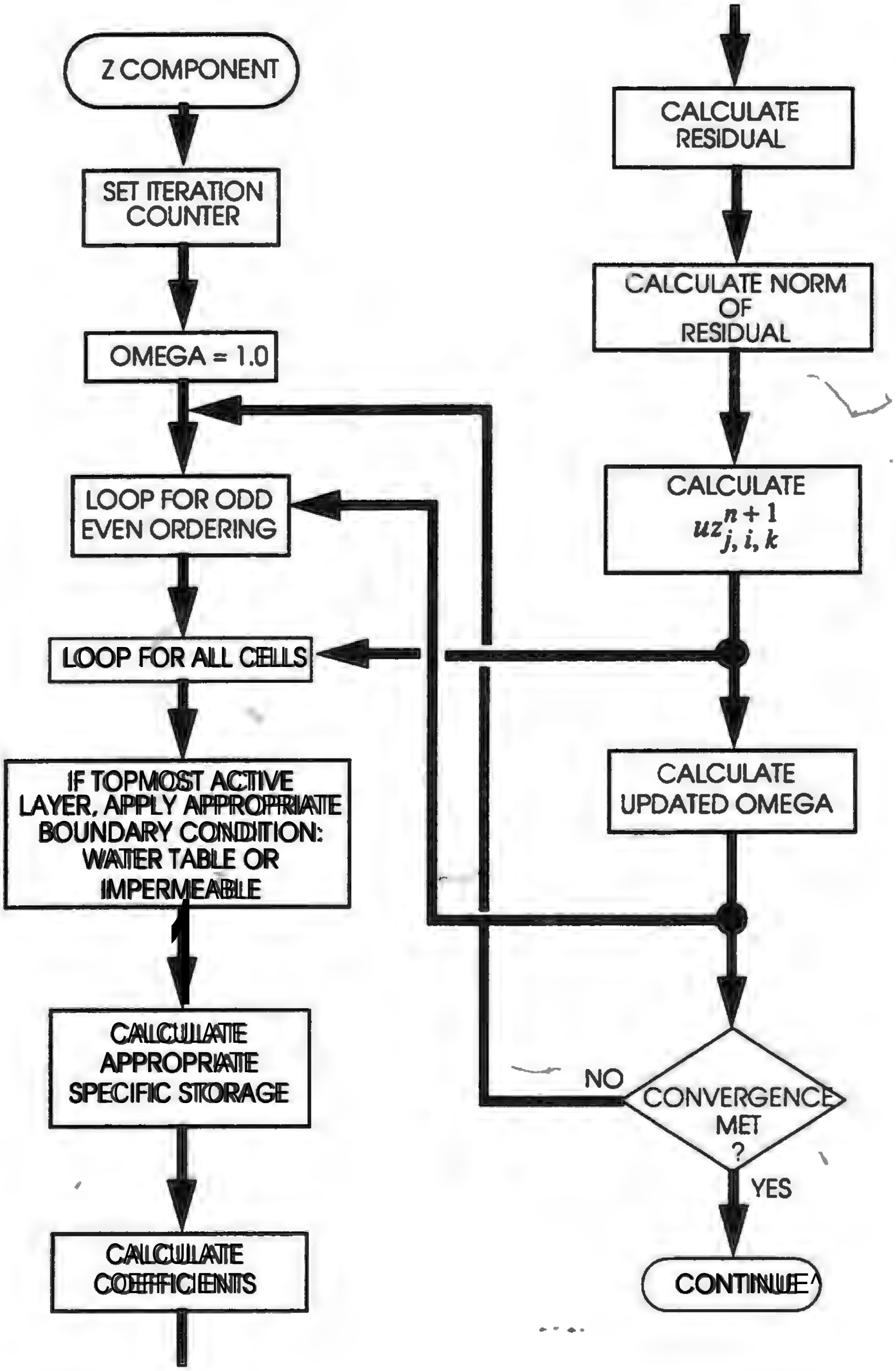
where Q is the pumping rate, b is the aquifer thickness and r is the radial distance from the well. For axially symmetric radial flow

$$q_b = \frac{Q}{2\pi rb} \qquad \langle 66 \rangle$$

where q_b is the radial component of $\vec{q}_b$. For the purposes of this report, horizontal flow is assumed, and the aquifer is isotropic and homogeneous. However, the numerical model as written does not require these limiting assumptions.

Figure 4. Generalized flow chart of granular displacement model. Includes inner-loop flow chart of Z direction only. X and Y inner loops are synonymous to Z.




Z COMPONENT
SET ITERATION COUNTER
OMEGA = 1.0
LOOP FOR ODD EVEN ORDERING
LOOP FOR ALL CELLS
IF TOPMOST ACTIVE LAYER, APPLY APPROPRIATE BOUNDARY CONDITION: WATER TABLE OR IMPERMEABLE
CALCULATE APPROPRIATE SPECIFIC STORAGE
CALCULATE COEFFICIENTS
CALCULATE RESIDUAL
CALCULATE NORM OF RESIDUAL
CALCULATE $uz_{j,i,k}^{n+1}$
CALCULATE UPDATED OMEGA
NO
CONVERGENCE MET ?
YES
CONTINUE

The U.S. Geological Survey's modular ground-water flow model, MODFLOW (McDonald and Harbaugh, 1988), makes use of a rectangular finite-difference grid network in cartesian coordinates. Thus, the bulk flux must be transformed from polar to cartesian coordinates. The horizontal component of bulk flux in cartesian coordinates is given by the relations:

$$q_{bx} = \frac{Q(x-x_o)}{Sr} \tag{67}$$

and

$$q_{by} = \frac{Q(y-y_o)}{Sr} \tag{68}$$

where $x-x_o$ and $y-y_o$ represent the distance along the principal coordinate direction from the pumping well to any point of concern within the aquifer, namely, to each grid cell center. S represents the surface area of equal values of bulk flux that emanate outward radially from the pumping well. For an idealized Theis aquifer (one that is homogeneous, isotropic, and of constant thickness) pressure transients migrate outward over time in a circular fashion. Therefore, the surface area is expressed as $2\pi rb$. The radial distance r from the pumping well to the point of interest is expressed in cartesian coordinates as

$$r = \left[(x-x_o)^2 + (y-y_o)^2\right]^{\frac{1}{2}} . \tag{69}$$

The bulk flux initial condition is now established for a point source or sink and remains constant over time as long as Q remains constant. A new bulk flux would need to be calculated if the pumping rate is changed.

Helm (1994) uses the integral form of the Theis solution (Lohman, 1972, p. 8) to develop equations for displacement. By taking the divergence of the Theis solution and using eqs. 5 and 66 he obtains a Theis-Thiem expression for a confined aquifer. The displacement field of solids can be determined by integrating the velocity of solids with respect to time. Helm's resulting expression for the displacement field of solids is

$$u_s = \frac{QrS_s}{8\pi Kb}\left(\frac{1-e^{-u}}{u}+\int_u^\infty \frac{e^{-m}}{m}\,dm\right) \qquad (70)$$

where $u = r^2S_s/(4Kt)$. Helm (1994) develops expressions of dimensionless displacement from equation 71 both for a fixed radial distance where time is variable, and for a fixed point in time where the radial distance from the well is allowed to change. For a fixed radial distance from the pumping well the dimensionless displacement, u_{td}, and the dimensionless time, t_d, are

$$u_{td} = \frac{8\pi Kbu_s}{QrS_s}, \qquad (71)$$

and

$$t_d = \frac{Kt}{S_sr^2}. \qquad (72)$$

Likewise, for a fixed time, the dimensionless displacement, u_{rd}, and dimensionless distance, r_d, are calculated by Helm to be:

$$u_{rd} = \frac{2\pi b\sqrt{K}u_s}{Q\sqrt{S_st}}, \qquad (73)$$

and

$$r_d = \frac{\sqrt{S_s}r}{\sqrt{Kt}}. \qquad (74)$$

To compare the numerical results to these analytic solutions developed by Helm, a finite-difference grid network is first constructed. To take advantage of radial symmetry, only one-fourth of a radial flow field is simulated to optimize computational time. A 30x30 finite-difference cartesian-coordinate grid is used for the simulations in this report. The grid increases in size geometrically outward from the pumping well. The pumping well is located at grid location x=1 and y=1 with grid dimensions of 0.328 m.

The geometric relations that determine the dimensions of the adjacent cell (outward from the pumping well) are x+0.5x and y+0.5y where x and y represent the current cell dimensions. The geometrically increasing cell dimensions are used to approximate an infitute radial aquifer (assumed for the Theis solution) and to assure that the zero-displacement boundary does not affect simulation results near the pumping well.

For purposes of comparison to the analytic solutions, arbitrary but realistic initial conditions and aquifer parameters are chosen. Table 1 lists the initial conditions and aquifer parameters used for the simulation. Helm's dimensionless expressions are converted to real time and real distance with the data used for the numerical simulation.

Parameter	Value(s)
Pumping rate, Q	2,450 m^3/day
Hydraulic diffusivity, K/S_s	1.866x10^3 m^2/day 1.866x10^4 m^2/day 1.866x10^5 m^2/day
Aquifer thickness, b	30 m

Table 1: Initial conditions and aquifer parameters used for model evaluation

Figure 5 compares the analytic solution displacements to the simulated displacements as a function of distance from the pumping well for the three hydraulic diffusivities listed in table 1 after 10 days of pumping. The results indicate that the numerical model accurately approximates the governing equations for granular movement (eqs. 51 and 52 without the *z* cross-product terms). The results are not significantly affected by time-step size. Only minor differences are noticeable when the 10 day pumping period is divided into two time steps or 50 time steps. The illustration

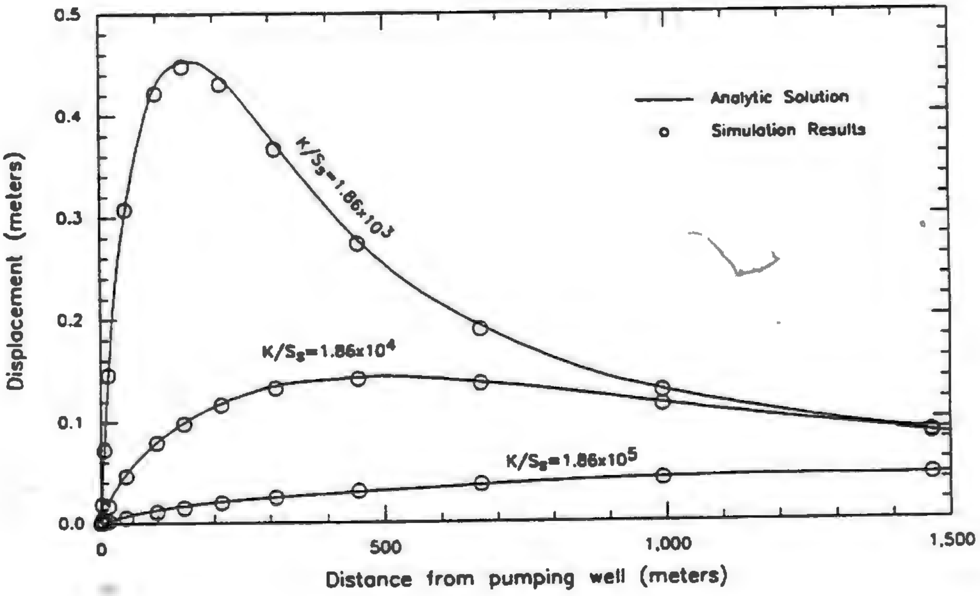


Figure 5. Plot showing displacement as a function of radial distance from pumping well for analytic solution and simulation with granular displacement model using aquifer properties listed in table 1.

reveals that the displacement field produces a wave form with respect to distance from the pumping well. This is a wave-like phenomenon because all materials (water and skeletal frame) are moving radially inward. In plan view, the maximum amplitude of this wave represents a circumference centered on the pumping well. This circumference of maximum displacement moves outward as a function of time as pumping continues. The shape and amplitude of the wave is dependent upon the hydraulic diffusivity, pumping rate, and pumping time. For a specified time the value of maximum displacement decreases with an increase in hydraulic diffusivity. Similarly, the point of maximum displacement migrates outward from the well with an increase in hydraulic diffusivity (fig. 5). The slope at any point on the displacement curve represents the radial strain at that point r for a given instant in time. The point of maximum displacement represents a circumference of zero radial strain. From the pumping well to the point of zero radial strain the aquifer is experiencing radial compression. That is, porosity and hydraulic conductivity decrease as the grains are rearranged to a more closely packed configuration. Beyond the region of zero radial strain the aquifer is experiencing radial extension. For a given point at distance r from the pumping well, however, there will be initially a period of radial extension followed by an instant of zero radial strain, followed by a period of radial compression as the wave of maximum displacement moves outward. The total length of time for the episodes of extension and compression depends upon the pumping rate and the hydraulic diffusivity.

Strictly speaking, for an idealized Theis aquifer no drawdown (and no change in volume strain) occurs only at an infinite distance from the well. According to Helm (1994) at a radial distance of $r_o \equiv 3.1\sqrt{Kt/S_s}$ the change in volume strain, $\delta\varepsilon$, can be considered negligibly small. In other words, outward from the boundary identified by r_o is a region where porosity volume does not change significantly even though the shape of the pore spaces is changing. This occurs because radial extensional strain in this region is equal to (or nearly so) tangential compressional strain. Beyond r_o water does not flow relative to the solid matrix and hence specific discharge, q, remains zero. Consequently, there is no drawdown; that is, $\hat{q} = -K\nabla h = 0$.

From Helm's analytical solution and resulting curve relating dimensionless distance to dimensionless cumulative displacement, the product $u_{rd} \cdot r_d$ is approximately equal to unity within this outer region. Thus, from eqs. 73 and 74 a displacement can be calculated for the radius at this outer circumference. That is,

$$u_r = \frac{Qt}{2\pi b n r_o} \tag{75}$$

From equation 75 analytically derived values of displacement can be easily calculated for the three hydraulic diffusivities listed in table 1 for this outermost region where $r \geq r_o$.

Table 2 gives radii, r_o, to the outer boundary representing zero change in volume strain and zero drawdown. These radii are accompanied by the analytically derived (eq. 75), and simulated (eqs. 61-64), displacements at this outermost boundary for the three values of hydraulic diffusivity listed in table 1 after 10 days of pumping. The last column in table 2 lists the simulated distance from the pumping well where the drawdown becomes nearly zero (less than 0.15 m) for the three hydraulic diffusivity values after 10 days of pumping. These radii of zero drawdown were simulated from the modular ground-water flow model, which is independent of the displacement model.

Several important observations can be made from the results shown in table 2. First, the aquifer is in motion and displacement does occur in the region of the aquifer beyond what has traditionally been defined as the "radius of influence" of the pumping well (based on hydraulic head alone). Helm (1994) refers to this outer boundary as the transient radius of influence because this boundary moves outward from the pumping well with the square root of time. It also represents the circumference of maximum radial strain in tension. Field data verify that displacement must occur beyond the "radius of influence" of the well because earth fissures are often known to form beyond where there is any significant drawdown (Anderson, 1989). The simulation results accurately reflect the values obtained from the analytic solutions. Second, the point at which drawdowns were simulated to return to their prepumping state (zero drawdown) closely corresponds to this outer region of zero change in volume strain (at r_o). Hence, not only is the displacement model able to duplicate the analytic results, but the analytic solutions and displacement model predict where the simulated "radius of influence" will reach in accordance with the ground-water flow model.

Hydraulic Diffusivity *(m^3/day)*	Analytically derived distance from pumping well (m)	Analytically derived displacement (m)	Model's calculated displacement (m)	Ground-water model's calculated radius of influence (<0.15 m drawdown) (m)
1.86×10^3	455	0.99	0.97	426
1.86×10^4	1,438	0.31	0.30	1,345
1.86×10^5	4,547	0.01	0.10	4,264

Table 2: Analytic and simulated values for displacement and "radius of influence "at the boundary representing negligibly small change in volume strain after 10 days of pumping.

It is worth emphasizing that this outward moving "radius of influence" corresponds not only to where there is negligible drawdown but also to where there is maximum extensional radial strain which corresponds to the inflection point on the curve in fig. 5. The standard ground-water model can not predict such a correlation, whereas the equation of motion of aquifer material (eq. 39) does.

Figure 6 shows the analytic and simulation results for displacement at a fixed radial distance of approximately 490 m from the pumping well for the three hydraulic diffusivities listed in table 2. Again, the simulation results indicate that the numerical model is capable of accurately approximating the analytic solutions developed by Helm (1994) for displacement versus time. Therefore, the model represents a good numerical approximation to the partial differential equations that define granular movement. The plot indicates that the grains initially begin from rest and would eventually reach an asymptotic value (ultimate displacement) after a long period of pumping. That is, as total pumping time increases, the velocity of solids decreases. The displacements shown are cumulative.

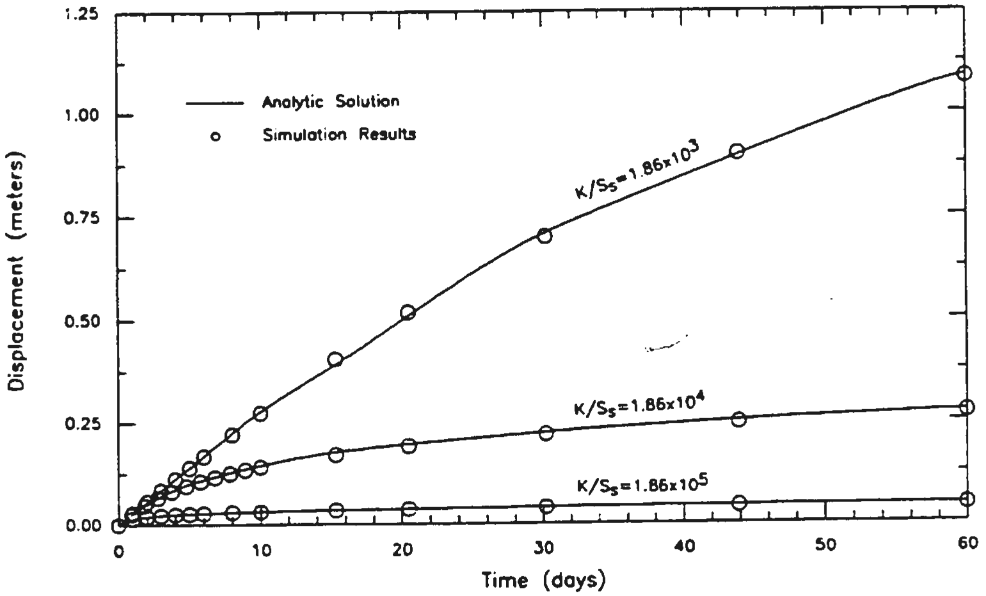


Figure 6. Plot showing displacement as a function of time since the onset of pumping for analytic solution and simulation with the granular displacement model using aquifer properties listed in table 1.

Nonconstant Pumping, Relaxation, and Injection

The analytic solutions of displacement for the Theis-type aquifer (Helm, 1994) are limited to a constant rate of pumping. The numerical displacement model, however, is capable of simulating displacement fields for any Q (pumping, relaxation, or injection) one chooses. Simulation results are presented for three scenarios in which Q is allowed to change. The first scenario uses the evaluation simulation (10 days of pumping) with a hydraulic diffusivity of $1.86\text{x}10^4$ m^2/day followed by a 50 day relaxation period (no pumping). The second scenario (presented here) uses a combination of pumping and relaxation periods in order to evaluate how turning the well on and off will affect granular movement. The third scenario uses a combination of pumping and injection periods where the injection rate is equal to the pumping rate. Because injection of potable water is becoming more common in arid-zone cities during seasons of low water use, this scenario will evaluate how such management practices may affect granular movement. The results of these three scenarios are presented below.

Simulation results of the first scenario are shown in figures 7a and 7b. The maximum displacement for the set of initial conditions after 10 days of pumping occurs at a distance of 541m from the pumping well. Figure 7b graphically shows the drawdown curve corresponding to the displacement curve in figure 7a after 10 days of pumping. After 10 days of pumping the well is shutoff but the grains do not become stationary. Rather, they begin to return to their prepumping state (fig. 7a). Simulation results indicate that the grains in the aquifer will eventually reach their prepumping state after nearly 2 years of relaxation. The relaxation of the granular matrix reflects the elastic stress-strain constitutive relation used in the theoretical development (eq. 34). Water levels return to their prepumping levels much more rapidly than the grains do (compare figs. 7a and 7b).

In the second scenario, the aquifer is subjected to a series of pumping stresses followed by episodes of relaxation. Figures 8a and 8b illustrate how this pumping and relaxation pattern influences the displacement field of solids and drawdown, respectively. The maximum displacement after 2.5 days of pumping at 5.0 m^3/day (stress period 1) is essentially the same as the maximum displacement after 10 days of

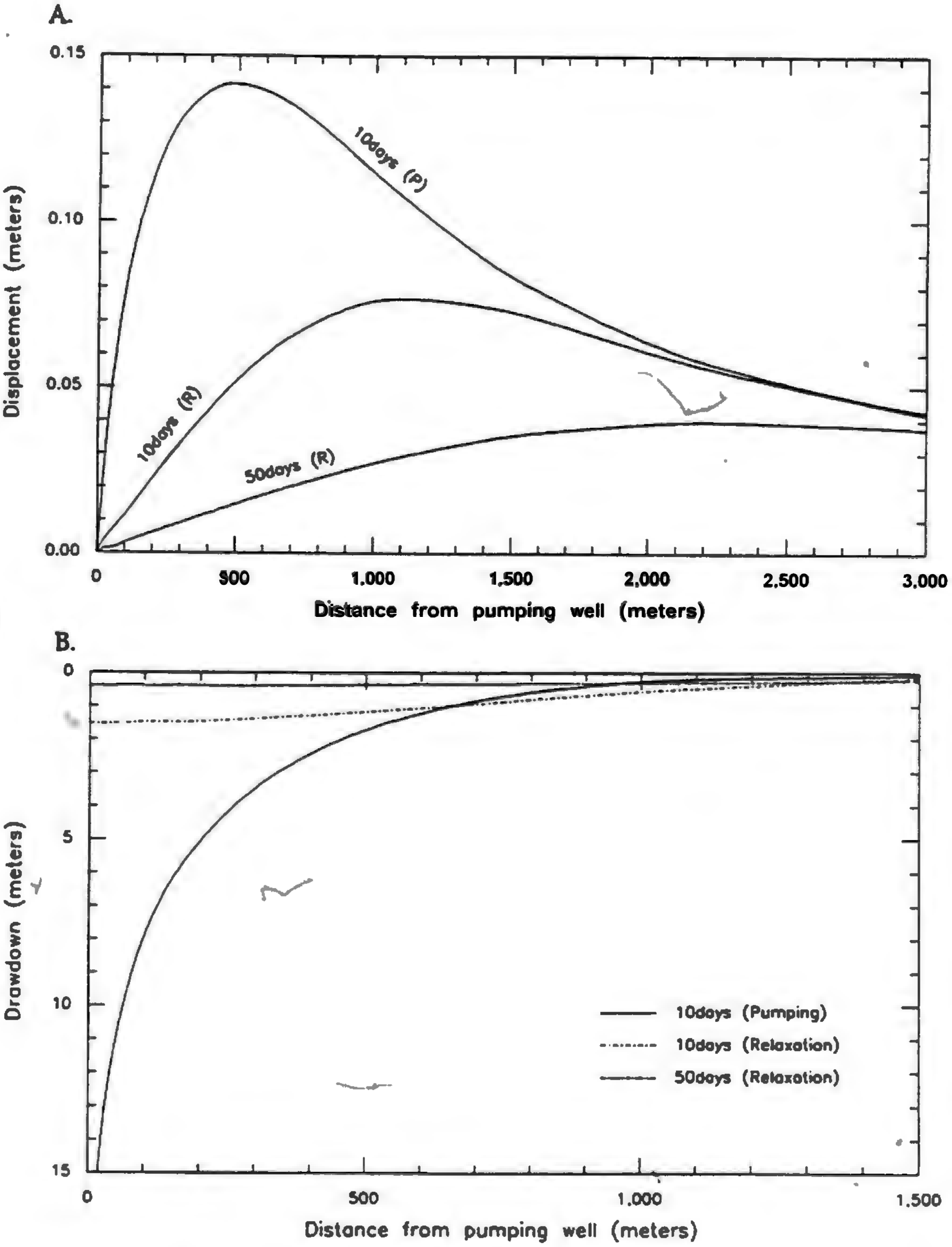


Figure 7. Simulation results showing A. displacement as a function of radial distance from pumping well for one period of pumping followed by a prolonged period of relaxation; and B. drawdowns as calculated using MODFLOW.

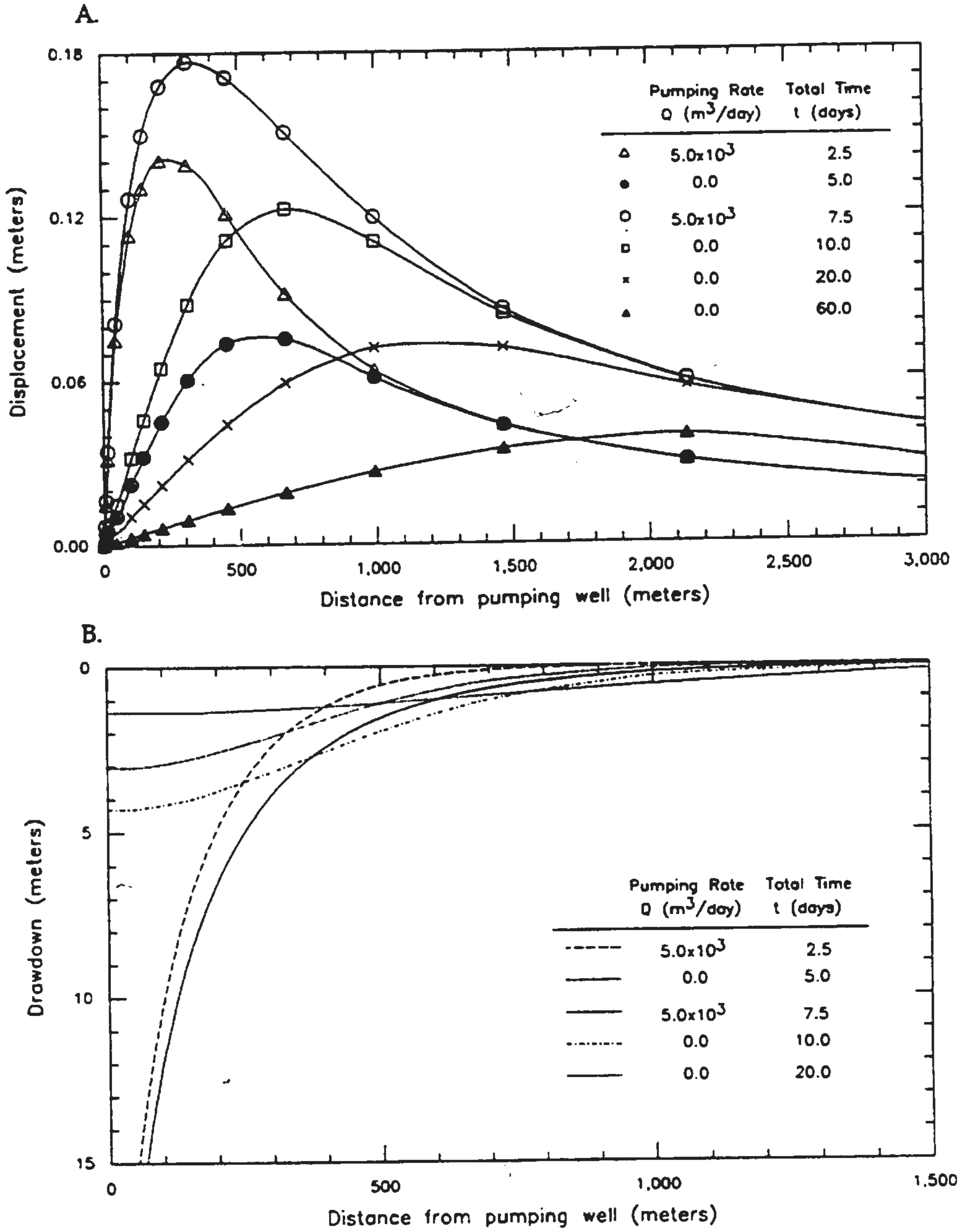


Figure 8. Simulation results showing A. displacement as a function of radial distance from pumping well for cycles of pumping and relaxation; and B. drawdowns as calculated using MODFLOW.

pumping at 2.5 m^3/day (fig. 7a). The different pumping rate is reflected in the radial distance at which the maximum displacement occurs. In this scenario the radial distance where this maximum displacement occurs is at 271 m, or one half the distance that occurs in the first scenario. Of interest is the fact that the displacement at which the drawdown is "zero" is 0.10 m., the same amoint as observed in the first pumping scenario. Furthermore, the radius representing zero volume strain, $\hat{r}_o$, is almost identical as in the first pumping scenario after 10 days.

The maximum displacement following the second pumping period (stress period 3) increases to 0.19 m (fig. 8a). The total volume of water pumped after 7.5 days is identical to the volume of water pumped after 10.0 days in the first scenario (fig. 7a), yet there is 20 percent more displacement. The radial distance at which this maximum displacement occurs is approximately 360 m from the pumping well, or nearly 100 m farther than after the first pumping period (stress period 1), but 180 m closer to the pumping well than the radius of maximum displacement in the first scenario (fig. 8a). Note, in figure 9a, that after the second and fourth stress periods (2.5 day relaxation periods) the radius of maximum displacement moves outward to more than twice the distance from the pumping well than that which is simulated after the corresponding pumping period (stress periods 1 and 2). Although the maximum displacement is farther from the pumping well, its maximum displacement is greatly reduced. After 20 days (additional 10 days of relaxation) the displacement curve is nearly the same as that of fig. 8a for the same simulation time. After 60 days (50 days relaxation), the curves in figs. 7a and 8a are identical, suggesting that regardless of the pumping patterns, the displacement curves during relaxation will eventually look the same for an equal volume of pumped water.

Figure 8a shows that the practice of turning the well on and off (second scenario) is potentially more damaging to the aquifer (may lead to a higher incidence of fissure development and shearing of well casings) than leaving the pump on at a lower pumping rate (first scenario). Not only is the maximum displacement increased in the second scenario, but the granular matrix is more dynamically active, experiencing greater strain and movement both toward and subsequently away from the pumping well. This scenario has significance because intermittent pumping is practiced in Las Vegas and other arid and semi-arid regions where water use varies significantly from season to season. After many years of implementing this practice, the total displacements may be significantly greater than if a constant but lower pumping rate was used. To what effect this practice may have on fissure development is beyond the scope of this report. What is clear however, is that both displacements and strains induced by this practice are greater, and greater strains and displacements can be expected to ultimately have adverse effects on the aquifer itself and on any structures located on or within the aquifer.

The third scenario is similar to the second one discussed above. The difference is that the 2.5 day relaxation intervals are replaced by injection at the same rate as pumping and the 50 day relaxation period is omitted (table 3). Figure 9 is designed to reveal the rapid influence injection has on granular movement and strain. The top curve is identical to the curve shown in fig. 8a after 2.5 days (stress period 1). Injection is immediately implemented after the initial 2.5 days of pumping. The remaining curves in fig. 9 show the displacements that occur radially outward from the well as injection proceeds through time. The time slices are small to illustrate how quickly the grains respond to this 2.5 day period of injection (stress period 2). In the immediate vicinity of the well, displacements become negative; that is, they move farther from the well than their initial location (prior to pumping). Within about a hundred meters of the well the grains move dramatically (up to 0.2 meter in less than 2.0 days) and from compressional strain to extensional strain. Figure 10a shows the displacement curves for 2 cycles of pumping and injection (after 4 stress periods). This illustration reveals the tremendous amount of lateral movement that occurs within 300 m of the pumping well within a short period of time (2.5 days). Combined with the large strains and changes in strain from compressional to extensional and back again, the overall impact on the aquifer can be expected to be great. Figure 10b shows the resulting drawdowns from the ground-water flow model after each of the four stress periods.

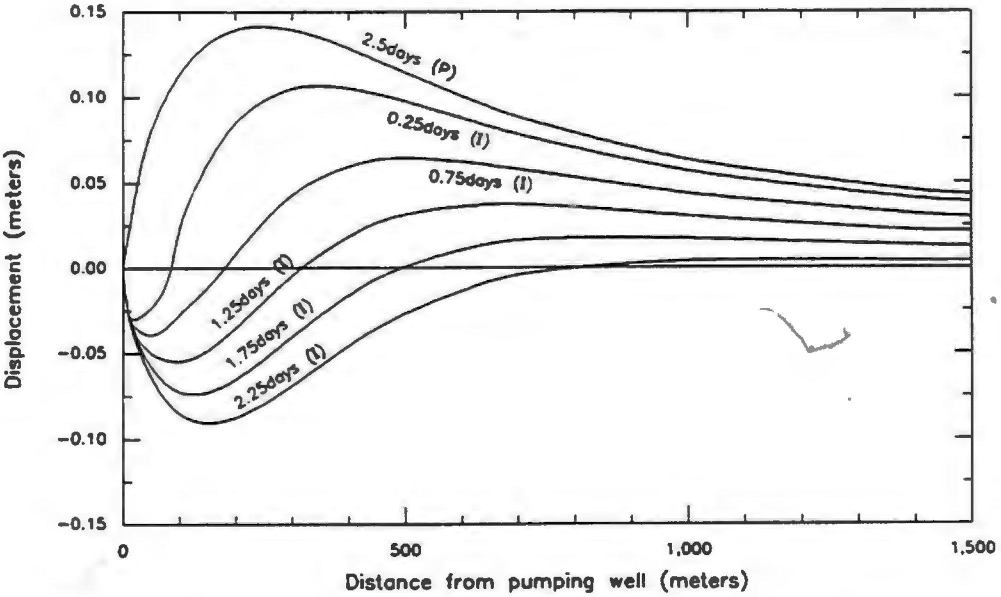


Figure 9. Plot showing displacement as a function of distance from the pumping well after 2.5 days of pumping immediately followed by 2.5 days of injection. Incremental time plots are shown during injection to indicate how quickly the aquifer responds to stress reversals.

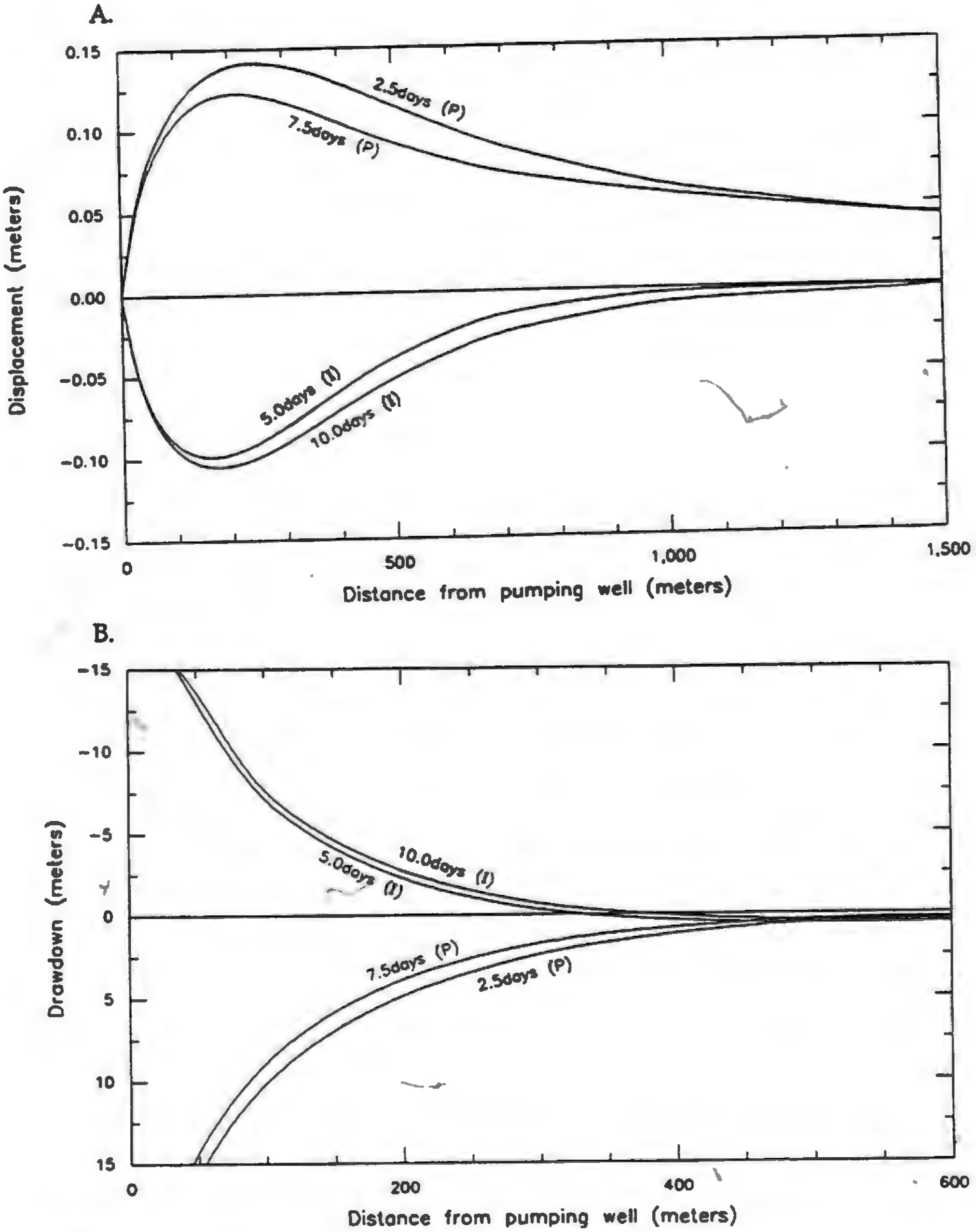


Figure 10. Simulation results showing A. displacement as a function of radial distance from pumping well for cycles of pumping and injection; and B. drawdowns as calculated using MODFLOW.

Pumping rate m^3/day	Stress period	Length of stress period (days)	Total simulation time (days)
2.0 (pumping)	1	2.5	2.5
2.0 (injection)	2	2.5	5.0
2.0 (pumping)	3	2.5	7.5
2.0 (injection)	4	2.5	10.0

Table 3: Stress period information used for the third scenario.

Such severe granular movements resulting from cycles of pumping and injection, may have long-term consequences. As injection (artificial recharge) becomes a more popular mechanism to store water during seasons of low water demand, increased strain on the granular matrix in the vicinity of the well or wells may not only weaken the soil structure, but may weaken or rupture nearby well casings and other structures. In Las Vegas, for instance, artificial recharge commenced in 1989 and has increased annually to a volume of 20,000 acre-ft. in water-year 1993 (Oct.-May), or approximately one third of the total annual volume pumped. Evidence of buckled and sheared well casings in the vicinity of the main well field in the valley point to horizontal movement or displacement as the potential cause. Although it is not known at this time whether horizontal granular movement is responsible for such failures, more field data and further well casing failures in the future may very well point to this overlooked phenomenon.

SIMULATION OF THREE-DIMENSIONAL GRANULAR MOVEMENT

Model Evaluation

Displacement in a two-dimensional Theis-type aquifer has been accurately simulated using the granular displacement model. The approximation of the differential equations and numerical scheme used is adequate for approximating the analytic solutions developed by Helm (1994) for an areally infinite aquifer. The same approach is used to extend the model to three dimensions. The numerical approximations are the same as those used for two dimensions. The major difference arises due to the water table boundary condition and the use of ultimate values of bulk flux for the water table and all heterogeneities present in the aquifer system being evaluated.

No known models exist that use displacement as the principal unknown to evaluate granular movement resulting from imposed stresses in a fully three dimensional setting. Existing models such as COMPAC (Helm, 1975, 1976), the interbed storage model (Leake and Prudic, 1988) only evaluate vertical compaction or strain on the basis of effective stress changes within fine-grained interbeds. These vertical strain models convert effective stress changes to an equivalent change in thickness of a compressible interbed as follows:

$$\Delta b = \frac{\delta\sigma'}{\rho_w g} S_{sk} b_o \qquad (76)$$

where Δb is the change in thickness of the interbed, S_{sk} is the skeletal specific storage, and b_o is the original thickness of the interbed. The skeletal specific storage may be elastic or inelastic (virgin) depending upon the previous maximum effective stress imposed on the aquifer. If the stress (measured as drawdown) exceeds its previous maximum value then the skeletal specific storage is in the virgin range, otherwise it is elastic.

These vertical strain models take each compressible interbed as a separate entity and then sums the results (eq. 76 is summed for all interbeds). For example, if a series of doubly draining clay lenses lies within a single aquifer (see layers 4 and 5 of fig. 11), the midplane of each lens (interbed) is represented as impermeable due to vertical symmetry of water flow relative to this midplane. Vertical compression of the interbed (relative to the midplane) is opposite and equal to the rate of incompressible water that is squeezed out from interbed storage. In other words, for an interbed storage model, the implicit equation for velocity of solids (relative to the interbed midplane) is

$$v_{sz} = -q_z = K_{zz}\frac{\partial h}{\partial z} . \tag{77}$$

Note that eq. 77 differs from eq. 5 by the bulk flux term, q_{bz}. That is, if q_{bz} was included in this interbed storage model (as it should be) it would specify the rate at which the midplane moves vertically relative to a regionally specified point that is fixed in space (usually the bedrock bottom beneath the aquifer being simulated, as is done with the granular displacement model).

For the granular displacement model mass balance is ensured by the bulk flux term $\vec{q}_b$ satisfying eq. 6 and strain compatibility is inherently ensured because the dependent variable is the displacement field $\vec{u}_s$. For the interbed storage model, however, although vertical mass balance is ensured within each individual interbed, it is not necessarily ensured for the system as a whole. In order to approximate strain compatibility, the interbed storage model sums the vertical strain of all material (namely, for a vertical stack of N interbeds) that is modeled to be compressible within each vertical column from the water table downwards. In other words, the interbed storage model selects each i^{th} interbed and N total interbeds within a column of interest where

$$u_{ztotal} = \sum_{i=1}^{N} u_{zi} = \sum_{i=1}^{N} \int \varepsilon_{zzi}\, dz \tag{78}$$

This sum of strains (or "floating" interbeds) is assumed to accumulate at the top of the column and, by implication, to represent the vertical subsidence of the land surface. The vertical deformation of any material that lies between basement rock and the water table that is not modeled as a compressible interbed will escape being included in the interbed storage calculation of subsidence.

The vertical velocity of solids for the granular displacement model is expressed as

$$v_z = \frac{K}{S_s}\left(\frac{\partial^2 u_x}{\partial z \partial x} + \frac{\partial^2 u_y}{\partial z \partial y} + \frac{\partial^2 u_z}{\partial z^2}\right) + q_{bz} \qquad ‹79›$$

and is expressed relative to a fixed regional boundary. Therefore, not only can a net change in strain within a layer be calculated, but the total change relative to other layers (from a fixed reference) can be calculated so that both extensional and compressional strains are simulated and are manifested in the distribution of the displacement field. Variations in the strain field relative to a fixed point in space (namely, strain compatibility requirements) are included in eq. 79 but not with eqs. 77 or 78. The cross-product terms in x and y are inherently absent in eq. 77 but are included in eq. 79. The influence of these terms becomes large enough near the pumping well that they can not be ignored and result in calculated vertical displacements that differ from the compaction calculated from the interbed storage model. The granular displacement model would tend to have lower measured displacements within clay interbeds in a confined aquifer near the pumping well because the strain field (squeezed water volume) is not entirely contained in the vertical component as it is for the interbed storage model. Both horizontal components contain part of the volume strain field, particularly near the pumping well.

In order to use the interbed storage model to evaluate the accuracy of the vertical displacements of the granular displacement model, a suitable simulation must be set up that is not impacted by the limitations, assumptions, and differences between the two models. One possible approach is to evaluate the compaction or displacement within one set of interbeds within a confined aquifer. The elastic and inelastic components of skeletal specific storage must be equal in order to avoid the differences in methods in which inelastic storage is invoked in the two models. Figure 11 outlines the conceptual model used to evaluate the vertical displacements calculated using the granular displacement model. Table 4 lists the aquifer properties used for the evaluation. Layer 3 serves as the confining unit with a horizontal hydraulic conductivity four orders of magnitude less than the layers either above or below the confining bed. A constant pumping rate of $2.45x10^{-3} m^3/d$ for a 10 day period was used in the evaluation. A longer time period was not used because no recharge was implemented to offset the large pumping rate. This was done so that all the water would come from storage of the interbeds.

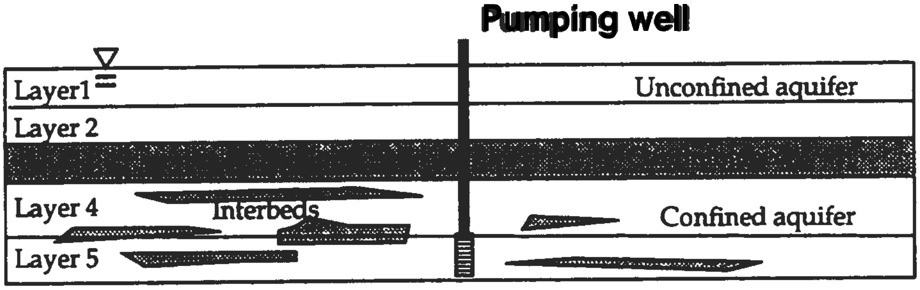


Figure 11. Conceptual model used to evaluate accuracy of vertical displacements calculated using the granular displacement model. The displacements were compared with compaction within a single layer containing interbeds in a confined aquifer using the interbed storage package (Leake and Prudic, 1988).

Property	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Horizontal K (m/d)	60.	60.	0.006	6.0	6.0
Vertical K (m/d)	60.	60.	0.0006	6.0	6.0
Specific Storage (1/m)	3.28e-9	3.28e-9	6.5e-6	6.5e-6	6.5e-7
Initial layer thickness (m)	15.2	15.2	15.2	15.2	15.2
Vertical conductance (1/t)	4.0	0.00004	0.00004	0.4	-------

Table 4: Aquifer properties used for each layer in evaluating the granular displacement model by comparing vertical displacements of layer 4 with compaction calculated in layer 4 using the interbed storage package (Leake and Prudic, 1991).

The results of the evaluation are shown qualitatively using the vector plotting package in fig. 12 where the plot represents a cross section along the y direction at row 26. The actual measured displacements within the region of concern in layer 4 are listed in table 5.

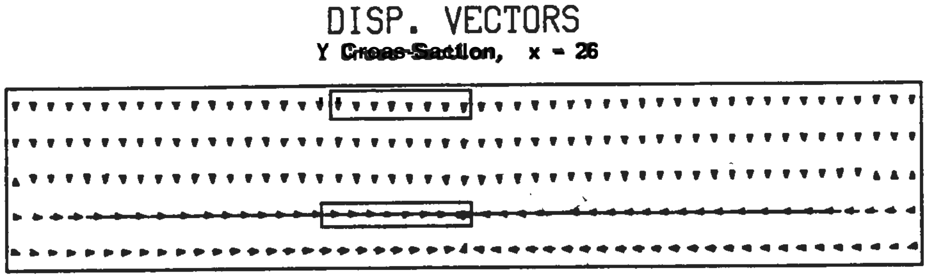


Figure 12. Vector plot showing relative magnitude (arrow length) and direction of displacement (arrow head) for the conceptual model described in fig. 11 and table 4. The boxes indicate where comparisons where made between the interbed storage model and the granular displacement model as listed in table 5.

Package	Col 19	Col 20	Col 21	Col 22	Col 23	Col 24	Col 25	Col 26
IBS Layer 4	.0120	.0137	.0152	.0215	.0275	.0350	.0488	.0823
GDM Layer 4	.0117	.0135	.0148	.0185	.0256	.0311	.0381	.0540
IBS Subsidence	.0186	.0228	.0277	.0344	.0433	.0554	.0792	.1280
GDM Subsidence	.0280	.0353	.0430	.0542	.0642	.0794	.0929	.1042

Table 5: Downward (z) displacements in centimeters calculated with the granular displacement model (GDM) and the interbed storage model (IBS) for model layer 4 and at the land surface representing total subsidence. Cell spacing is 1524 m. The pumping well is located in row and column 26.

Net displacements were compared with compaction of layer 4 by subtracting the calculated displacements of layer 5 from those of layer 4. Net differences had to be used for comparison to compaction values calculated with the interbed storage model. Using the net difference eliminated the influence of the location of a fixed boundary at depth. Small differences in the total displacement versus compaction (total strain over a thickness interval of interest) should be expected because of the differences in which the two values are calculated. In addition, MODFLOW uses the leakage between layers to determine vertical hydraulic conductivity of a layer. That is to say the harmonic mean of adjoining layers is used to estimate vertical hydraulic conductivity. The granular displacement model uses the vertical hydraulic conductivity of each layer without regard of the aquifer properties of adjoining layers. Consequently, low leakage values had to be specified for both layers 2 and 3 to assure that the proper confinement of layer 3 was simulated. This difference alone could account for the differences in the two models reported in table 5.

Calculated vertical compaction within layer 4 is nearly identical for the two models (table 5) except near the wellbore where the interbed storage model has larger calculated values of compaction. These larger values are expected due to the volume strain field contained completely within the vertical space dimension for the interbed storage model (eqs. 77 and 78). The inclusion of the cross product derivatives used for the granular displacement model (eq. 79) distributes the volume strain so that the horizontal components account for some of the overall displacement.

At points away from the well, larger total differences in subsidence or displacement are calculated with the granular displacement model. This is in part due to the fixed reference frame (the aquifer bottom) which is used and tends to pull the aquifer toward this underlying fixed plane.

The granular displacement model provides information that can not be obtained with the interbed storage model. Displacement data indicate that vertical compressional strain exists from the confining unit downward while vertical extensional strain exists within the unconfined aquifer after the 10 day simulation period. As time increases the unconfined aquifer also becomes dominated by vertical compressional strain.

This numerical experiment indicates that under controlled conditions where net changes in displacement (compaction) obtained with the granular displacement model are compared with results of the interbed storage model, nearly identical results are achieved. Both horizontal and vertical simulated displacements using the newly developed granular displacement model have been evaluated against existing analytic and numerical models. In the following section, various scenarios are developed to analyze how displacement fields are influenced by various boundary conditions and heterogeneities.

Flow Barriers, Heterogeneity, and Multiple Wells

Countless variations in aquifer properties and wells can be used to simulate granular movement in three dimensions. Only several will be discussed. In the model evaluation we have already discussed how the granular matrix responds to a confined aquifer beneath an unconfined aquifer separated by a relatively thick confining unit. In the following discussion, granular matrix response to other confined and unconfined settings will be analyzed. The impact of heterogeneity and multiple pumping and injection wells on granular movement will also be discussed briefly. Results will be presented qualitatively as opposed to quantitatively. That is, actual displacement values will not be discussed but rather relative displacements (differences between one simulation and another).

Figures 13 and 14 are vector plots of an unconfined homogeneous aquifer pumped from layer 4. Figure 13 is a plan view of layer 4 showing the radial symmetry of displacement vectors toward the pumping well. The greatest magnitude of the horizontal displacement occurs in layer 4. Radially, the greatest radial displacement within layer 4

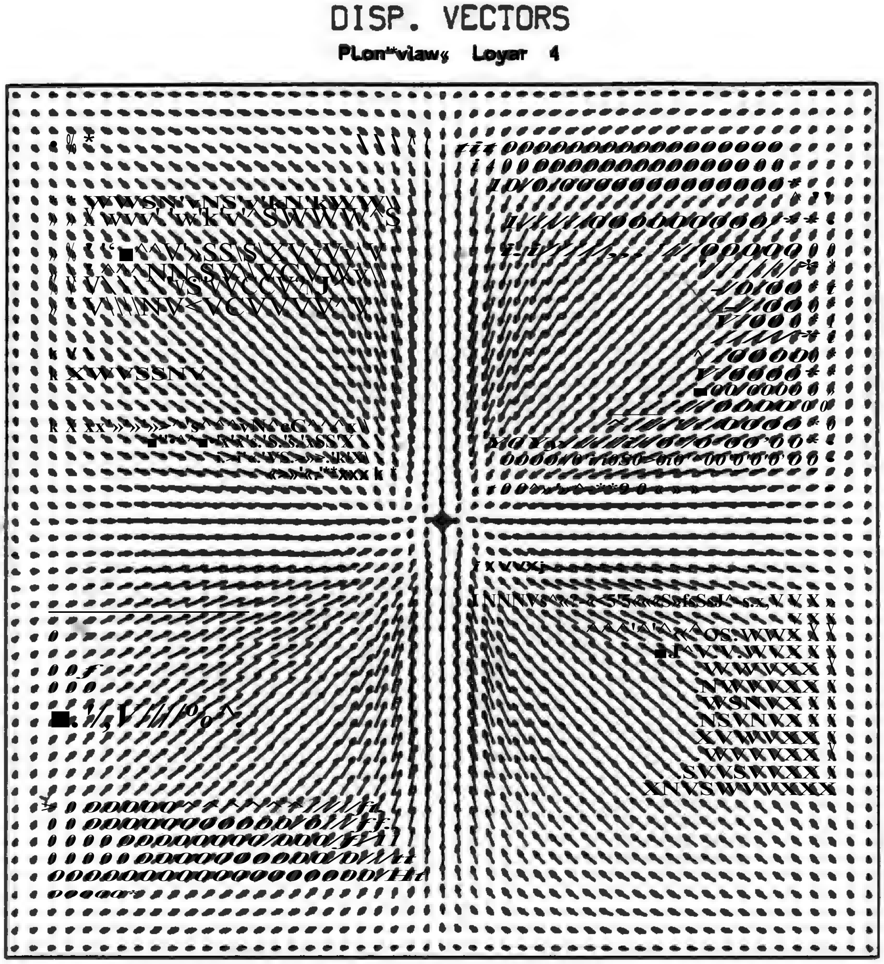


Figure 13. Planimetric vector plot showing simulated granular displacements of model layer 4 for a homogeneous aquifer pumped from the center of the model grid.

occurs about 2,500 m from the pumped well. This reflects the length of the pumping period and the value of the hydraulic diffusivity (see discussion on two-dimensional evaluation for explanation for the displacement field shown in fig. 13). Figure 14 shows a cross section through the pumping well and indicates that much of the vertical displacement occurs in the vicinity of the pumped well.

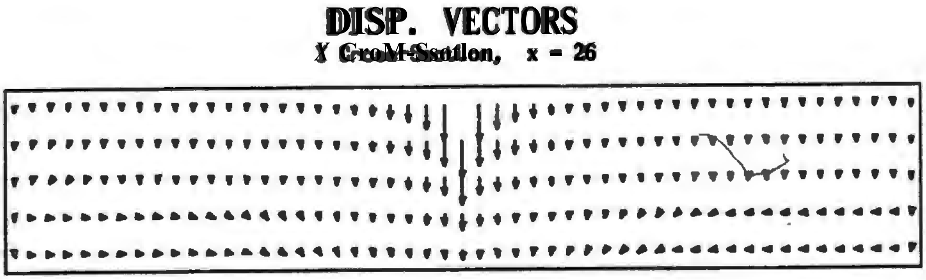


Figure 14. Cross-sectional vector plot along column 26 (pumping well is located at row 26 and column 26) showing granular displacement in a homogeneous aquifer pumped from layer 4.

These two illustrations will serve as reference plots for the four test simulations that will follow. The reader should be aware that the vector lengths, representing relative magnitude of displacement, between planimetric and cross sectional plots (as well as from simulation to simulation) can not be compared. The magnitude and subsequent vector length is calculated for each plot and is independent of other plots. This can be readily seen in viewing the vector lengths of layer 4 in fig. 14. The lengths are considerably less than those of fig. 13. This is due to the fact that vertical displacement near the pumping well is much greater than the horizontal displacement of layer 4. Displacement directions can be readily compared between different plots. The author will point out instances where displacement magnitude is noticeably increased or decreased due to a heterogeneity or implemented barrier.

The first test simulation (figs. 15 and 16) evaluates granular movement with a vertical flow barrier (zero permeability) through layers 2, 3, and 4 along columns 32 and 33. Simulated displacements along layer 3 (fig. 15) show how greatly the displacement vectors are altered by the impermeable barrier (compare with fig. 13). Displacements

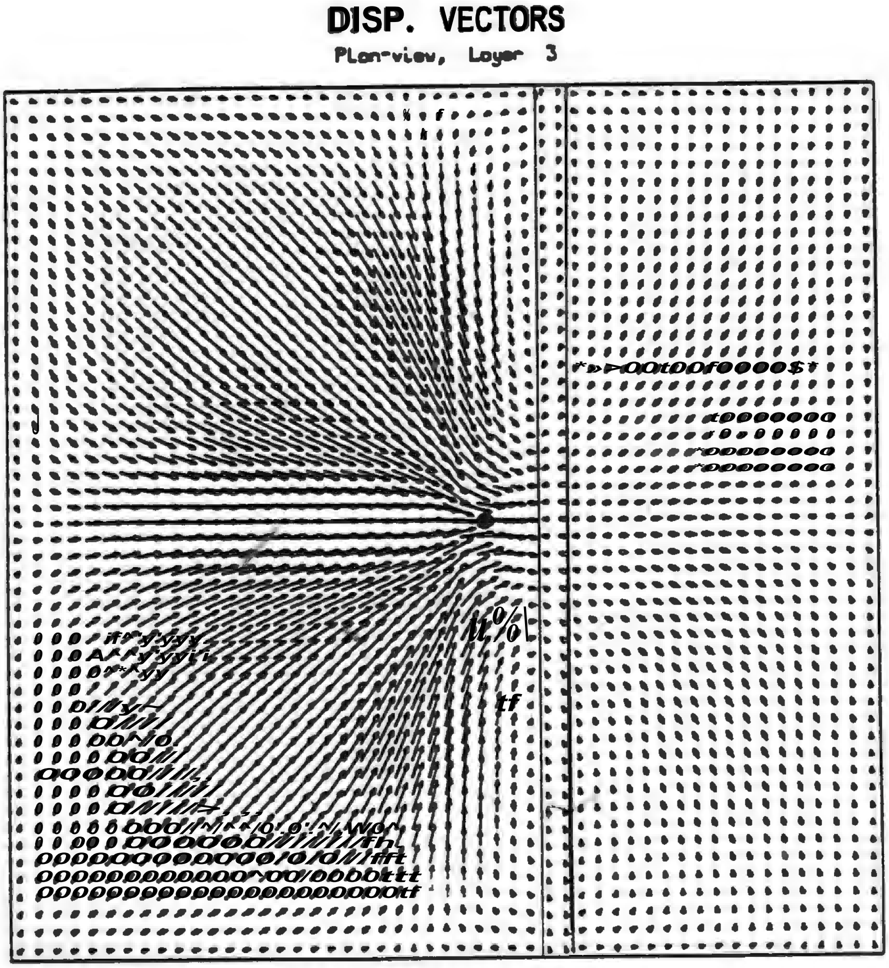


Figure 15. Planimetric vector plot showing the effect of a vertical linear flow barrier shown by the narrow rectangular box) on granular movement along layer 3.

tend to move toward the pumped well. However within about 2,000 m of the pumped well displacements appear to move toward the flow barrier. In fact the location of the wellbore itself actually moves toward the flow barrier causing compressional strain between the wellbore and the barrier. Upon initial inspection this may seem a contradiction. The wellbore, however, is not fixed. The fixed boundaries are at the margins of the grid or basin. The wellbore is actually moving, or being displaced, toward the flow barrier. On the side of the barrier opposite the well, displacements tend to be radially toward the pumped well. This is largely due to the initial values of bulk flux which form a radial pattern toward the well. Maximum displacement in this area is not near the barrier but at a distance of 2,300 m from the pumped well. Beyond this distance, the aquifer is experiencing radial extension.

Figure 16 is the cross-sectional vector plot along row 26 and is perpendicular to the flow barrier. Because of the large vertical displacements near the wellbore the vector tails in the horizontal direction are subdued, but vertical displacement or subsidence is enhanced between the well and the flow barrier (not readily seen distinguished on fig. 16). Subsidence is increased by over 100 percent in columns 30 and 31 (directly adjacent to the flow barrier). To the right of the flow barrier (opposite the pumped well) displacements tend to move downward and beneath the barrier. However, directly adjacent to the barrier granular movement is vertically upwards parallel to the barrier and over its top.

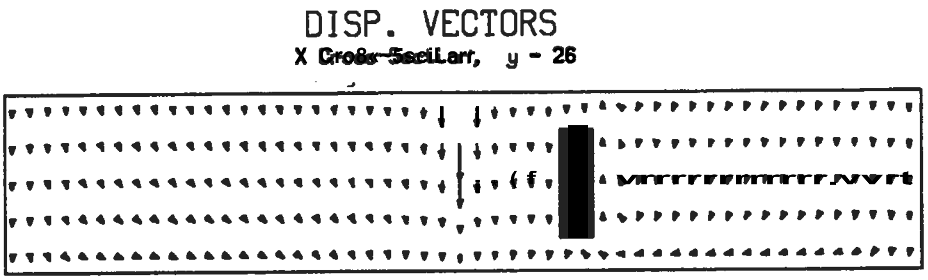


Figure 16. Cross-sectional vector plot showing relative granular movement with the inclusion of an impermeable flow barrier extending along the length of columns 32 and 33. The box indicates the location of the barrier.

The second test simulation involves implementing a horizontal flow barrier along layer 2 within the central part of the model grid of a homogeneous aquifer. In other words the barrier does not extend to the perimeter of the grid but only occupies the middle 27 cells in the row and column directions. Twelve model cells extending inward from the perimeter do not contain the impermeable horizontal barrier. The aquifer is pumped from the grid center in layer 4. Figure 17 shows simulated granular movement after pumping for a 30 day period. The lateral extent of the barrier is also shown.

Results show that the greatest downward vertical displacements or subsidence occur at the edge of the barrier (fig. 17) even though pumping does not occur in the vicinity of greatest subsidence. This is a classic example of strain compatibility. Above the flow barrier granular movement is upwards. These seemingly anomalous displacements may simply result from mass balance because the initial bulk flux values above the barrier are much greater than the ultimate bulk flux values. The specific discharge after a short period of time is downward, thus requiring the vecocity of solids (and displacement) to be upwards. Another explanation might be that the boundary condition uses hydraulic heads from MODFLOW which are assumed to be correct for the granular displacement model. If pumping were to continue for a longer period of time and a greater specific storage value was assigned to the lowest three layers, the displacements above the barrier would ultimately tend downwards.

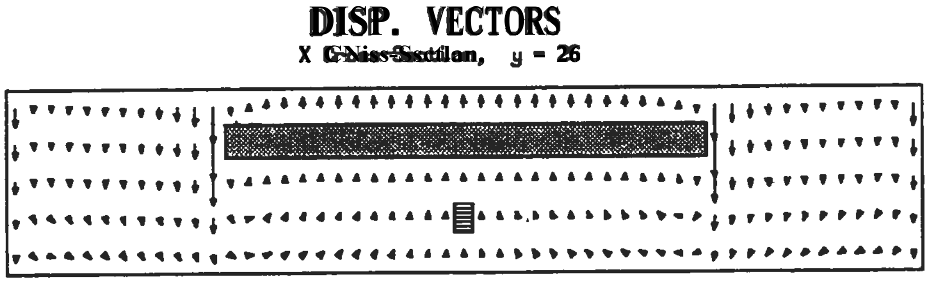


Figure 17. Cross-sectional vector plot showing location of horizontal flow barrier (shaded horizontal rectangle) and pumping well (hachured box in layer 4) and resulting displacement vectors.

This simulation is a generalized depiction of what may be occurring in Las Vegas Valley. The near-surface aquifer is separated from the principal aquifer by a thick confining unit (represented by the barrier in fig. 17). This confining unit covers much of the eastern two-thirds of the valley but is largely absent or less significant in the western part of the valley. According to Bell (1981) a significant amount of subsidence has occurred west of the thick highly-compressible confining unit. Figure 17 indicates that such observations are not anomalous or due to over generalization, but represent real physical phenomena.

The third test scenario evaluates the effect of multiple wells. In one simulation two pumping wells are placed in a homogeneous isotropic aquifer to evaluate granular movement when more than one well is involved in stressing the aquifer. In a second simulation, a pumping well and an injection well (equal but opposite rates) are used in the same aquifer. Figures 18 and 19 show the displacement vectors for layer 4 (layer from which pumping or injection occurs) for the two-pumping and one-pumping-one-injection well simulations, respectively.

Results for the first simulation (two pumped wells) indicate that much of the horizontal movement occurs toward the midpoint of the line connecting the two wells as opposed to the actual well locations (fig. 18). Little horizontal movement occurs along the line connecting the two pumped wells. This occurs because of the bulk flux established for each individual well tending to cause movement toward each individual well. The sum of the bulk flux for both wells tends to move the center of mass toward the midpoint of the line connecting the two pumped wells and not towards a single well.

Results for the second simulation (one pumped and one injection well) reveal that large horizontal displacements occur along the line connecting the two wells (fig. 19). Based on the locations of the injection and pumped wells one may conclude that the vectors (or the wells) do not line up correctly with the magnitude and direction of displacement. To understand that these horizontal movements are correct, it is best to analyze the effect of each individual well on granular movement and then sum the results. The pumping well tends to pull the grains toward it from all locations within the

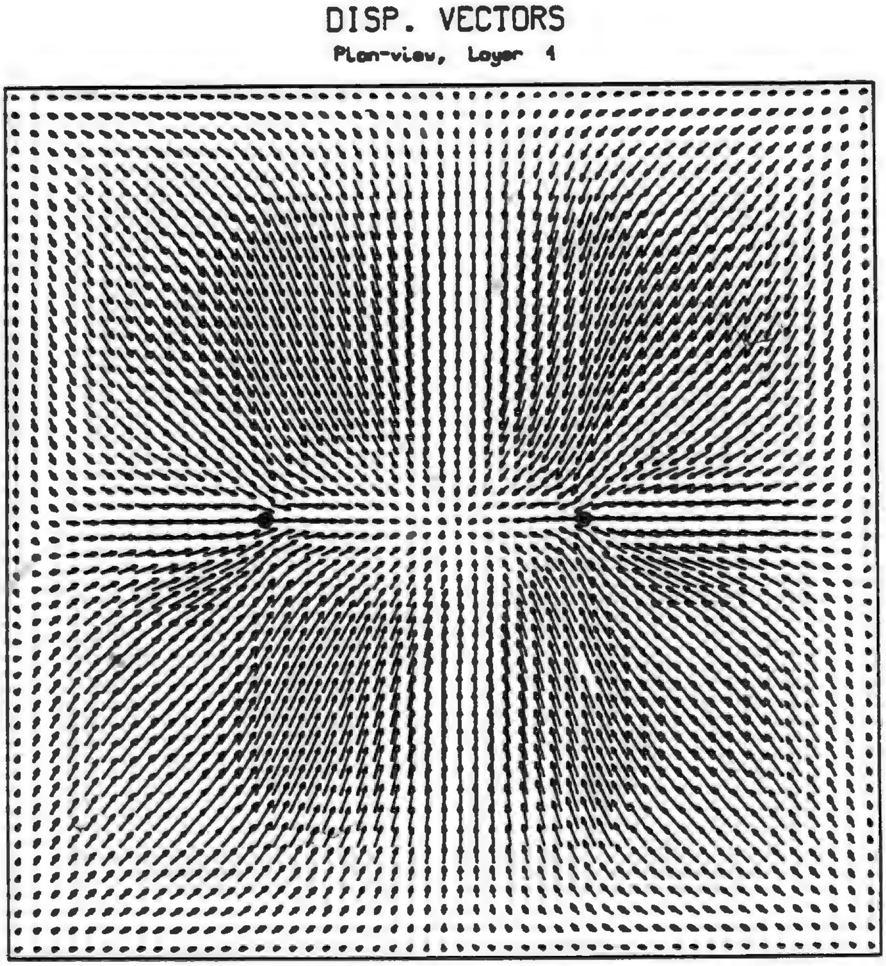


Figure 18. Planimetric plot of layer 4 showing granular displacement in a homogeneous isotropic unconfined aquifer with two pumping wells (black dots) located in layer 4.

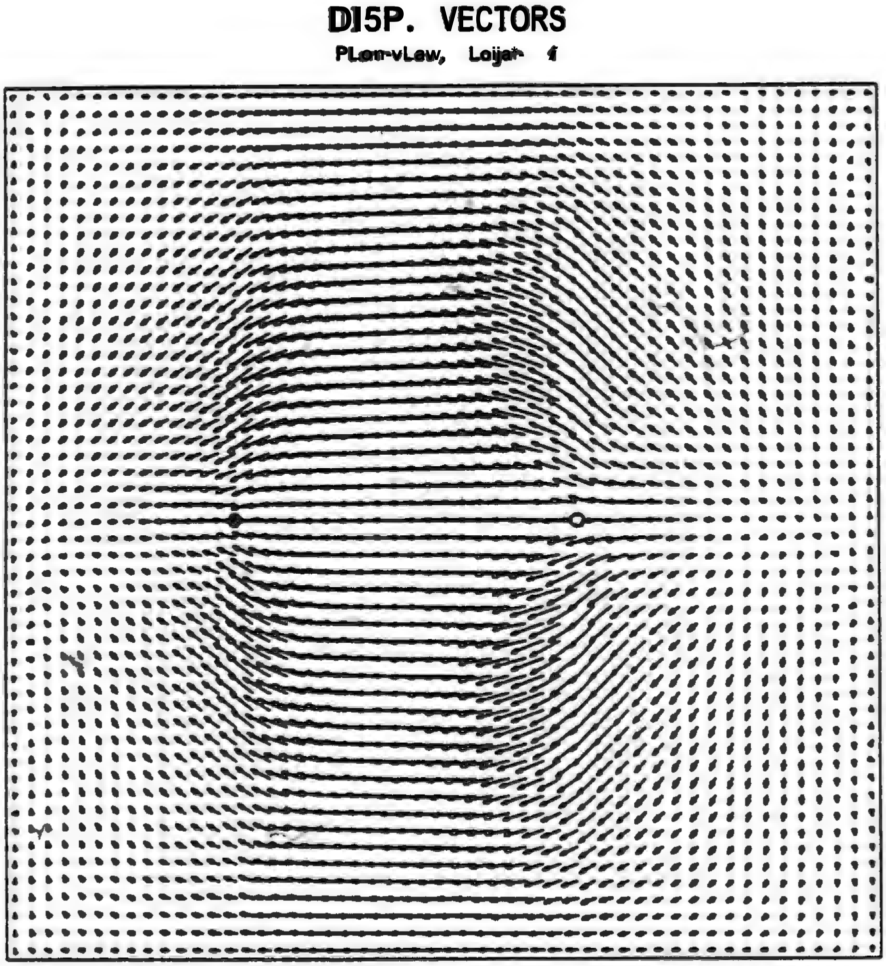


Figure 19. Planimetric plot of layer 4 showing granular displacement in a homogeneous isotropic aquifer with one pumped well and one injection well (black dot is pumping well, open dot is injection well) located in layer 4.

aquifer, so immediately the grains in the vicinity of the injection well have already been displaced toward the pumping well without considering injection. Now when coupling injection with the effects of pumping, movement away from the injection well is exacerbated making it appear as though the well should be located farther to the right edge of the boundary. In a similar way, the injection well tends to push the grains and wellbore farther from the pumped well. Hence, the final result is the correct displacement configuration for a pumped- and injection-well setting.

The fourth and final test simulation involves using a similar pumping pattern and location of the first simulation of test simulation three; that is, two pumping wells located along row 26. However, the hydraulic conductivity and vertical leakage of the right half of the aquifer is two orders of magnitude less than the left half, and pumping is from layer 3. Each half of the aquifer is homogeneous and anisotropic (vertical hydraulic conductivity is one order of magnitude less than the horizontal hydraulic conductivity), yet the system as a whole is heterogeneous because of the sharp hydraulic conductivity contrast of the two halves. This test is done to inquire as to how granular movement may respond to abrupt facies changes in alluvial basins.

Simulation results (fig. 20) indicate that horizontal granular movement is largely influenced by the pumped well within the half of the aquifer with the lower hydraulic conductivity (skeletal specific storage has been kept uniform for both halves of the aquifer). In the left half of fig. 20, note that the simulated vectors tend toward the pumped well on the left side but then are pulled toward the pumped well on the right side forming a type of rounded-step pattern. Displacement vectors tend to be nearly orthogonal to the bounary (facies change). By way of contrast, vertical displacements are larger on the side of the aquifer with the larger hydraulic conductivity (20-80 percent greater depending on location relative to the pumped well). This may be due to the gradient of hydraulic head calculated by way of MODFLOW.

From a practical standpoint, these results suggest that pumped wells from aquifers of lower hydraulic conductivity (or in aquifers with lithologies that inhibit horizontal ground-water flow) will tend to exacerbate horizontal displacement leading to

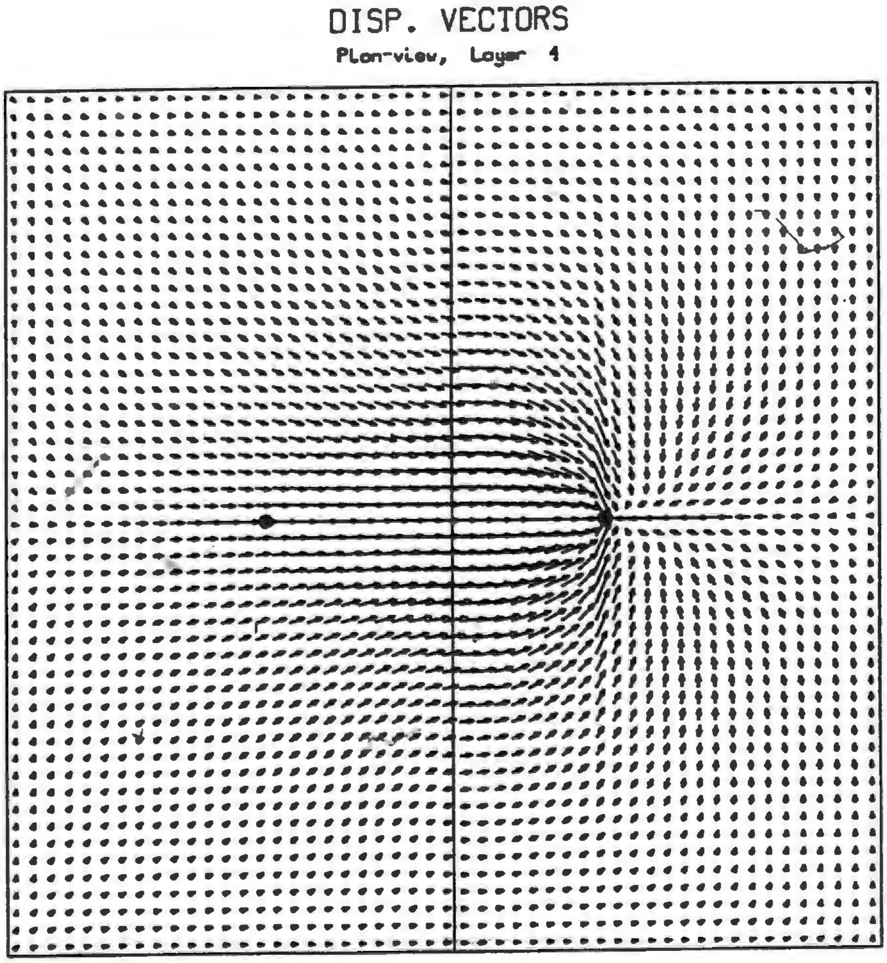


Figure 20. Planimetric plot showing relative granular movement and direction due to an abrupt facies change where the hydraulic conductivity is decreased by two orders of magnitude (right-hand side is lower). Each well is pumped the same rate. The dots indicate the location of the pumped wells. Wells are pumped from layer 3.

a potentially higher likelihood of fissure development in the vicinity of the well pumped from low-transmissivity half of the aquifer. Although vertical displacements were greater in the more highly transmissive part of the aquifer, had skeletal specific storage values been increased for the half of the aquifer with lower transmissivity, vertical displacements would probably be greater on the side of the aquifer with lower transmissivity as well. Eliminating the dependency of the gradient of hydraulic head from MODFLOW may also influence the vertical displacements.

MODEL LIMITATIONS

Due to the complex nature of the mathematical and numerical models developed to simulate three-dimensional granular movement, several limitations and caveats are inherent in the mathematical model and computer program Understanding the basis of these limitations will help the user avoid certain pitfalls and erroneous assumptions in developing a conceptual model designed for evaluating granular movement.

As with any numerical model, the quality of the data that goes into the model reflects the data that is produced by the model. Great care should be taken in properly developing the conceptual model for granular displacement. Unlike MODFLOW which is commonly used as a quasi three-dimensional model, the granular displacement model produces more accurate results when a more detailed fully three-dimensional model is used. That is, confining beds and other low permeable units are best treated as individual layers as opposed to employing only a leakage term for estimating vertical hydraulic conductivity. Thus, it is better to simulate the upper layer as a water table regardless of the nature of the topmost hydrogeologic unit being simulated. One drawback to this approach is that more data is needed for the simulation. When specifying a LAYCON=0, the user only needs to know the transmissivity of the unit but does not explicitly need to know the hydraulic conductivity or the thickness of the unit being simulated. When specifying this layer type (completely confined) for the granular displacement model, however, the user must enter the top and bottom elevations of the unit as well as the hydraulic conductivity of the unit. The granular displacement model is based on a fixed coordinate system and therefore requires the exact volume extent of each cell for all layers. This may be problematic in poorly defined systems where only estimates of transmissivities are known. In addition, information on the thickness and horizontal and vertical hydraulic conductivities are rarely known because aquifer-test data, when available, does not usually provide information about these low permeable units unless they were specifically designed for this purpose. The granular displacement model requires more detailed data about the system; and a fully three-dimensional model with these more detailed data provide more accurate displacements within all units of the system.

A second limitation with the granular displacement model is that it requires the user to simulate at least two layers to obtain vertical displacements or subsidence. Many two-dimensional ground-water models have been developed for specific aquifer systems and are properly calibrated to field data. To extend these existing models by applying the granular displacement program would require the user to either add another layer to the system which would take considerable time for conceptual reevaluation and implementation, or to simply use the granular displacement model to calculate only horizontal displacements while using the interbed storage package (Leake and Prudic, 1991) to simulate subsidence (vertical displacements). This latter approach is the recommended method for two-dimensional planimetric models.

The water-table boundary condition developed for the granular displacement model presently requires newly updated values of hydraulic head to calculate vertical displacements. In regions where thick units of highly compressible clay interbeds occur the standard MODFLOW program would estimate erroneous values of hydraulic head. Poland and others (1975, p. H38) report that an average of one third of the total water pumped from wells in the San Joaquin Valley of California comes from the compaction of these clay interbeds. The standard MODFLOW program does not consider subsidence and would therefore produce greater drawdowns than would be measured in the field. To overcome this problem, the user is recommended to use the interbed storage package and implement the necessary specific storage values to simulate the improved values of hydraulic head. These boundary heads at the water table are presently used in the granular displacement model. The advantage of this approach is that subsidence and compaction data are available to compare with displacements. The disadvantages are that (1) the heads calculated with the interbed storage package are likely to be somewhat different than heads calculated with a granular displacement mathematical model because of the three-dimensional nature of the model. In particular, (2) reversal in water levels that are observed in the field can not be simulated by the standard version of MODFLOW or the interbed storage model. Hence, under *in situ* conditions when the water table would physically rise initially (when a pump is turned on), the water table boundary condition will be in error. (3) Additional data are needed due to the implementation of an additional package (or set of subroutines). Not only does this add development time, but also computer processing time is increased. In the future, a wate-table boundary condition will be developed that is contained intirely within the granular displacement model and permits initial water-table reversals to occur.

The granular displacement model requires that steady-state hydraulic heads be calculated for a given stress period and used to calculate the ultimate specific discharge or bulk flux values. For complex settings this may prove to be quite cumbersome as a new set of steady-state heads are required for each stress period being simulated. Depending on the nature of the pumping patterns invoked, the set of steady-state hydraulic heads calculated during one stress period may be able to be used with another stress period if cyclic stress periods are used. This would eliminate the need for additional steady-state simulations. However, once these ultimate values are calculated and the input data set is developed, the granular displacement becomes a powerful tool for evaluating three-dimensional displacements, strains, and head reversals in a complex aquifer system. One powerful aspect of the use of ultimate bulk flux is that these values contain all boundary conditions and heterogeneities present in the conceptual and numerical model.

During the lengthy testing process for the granular displacement model, it was discovered that the z cross-product terms in the x and y directions (last term on left-hand side of eqs. 48 and 49) cause symmetry problems with the calculated displacements in the x and y directions. In addition, the inclusion of these z cross-product terms increases simulation time by as much as 15 times. Therefore, they have been omitted from the simulations presented in this dissertation but they have been retained in the program documentation. These z cross-product terms can be easily removed or added as the user desires simply by adding or removing the +ZCON term from the expression for RESID in calculations for displacement in the x and y directions. Although this problem was evaluated extensively, the reasons for the non-symmetry and increased computer processing time is not completely understood. The problem may reside not in the mathematical model but rather in the numerical model and may be related to the implementation of the Crank-Nicolson scheme used. The Crank-Nicolson scheme is known to produce accurate and stable results for two dimensional problems, but it is uncertain if this same approach is completely valid for three-dimensional problems. The weighting factor of 0.5 used for two dimensional problems may need to be reduced to one third or some other weight for three-dimensional problems.

CONCLUSIONS

Granular movement in unconsolidated aquifers is typically associated with vertical compaction of fine-grained interbeds. The significance of granular horizontal movement is often overlooked or ignored. The occurrence and location of earth fissures resulting from overdraft of unconsolidated aquifer systems in many arid and semiarid regions can not be explained or predicted with currently available hydrologic or subsidence models. These models are one-dimensional in scope with respect to strain (vertical only) and are not capable of calculating the total strain or displacement based on a fixed point in space because they use hydraulic head as the principal unknown and do not incorporate the bulk flux. Earth fissures are known to be controlled by horizontal granular movement. Thus, to accurately describe the strain and displacement fields due to applied stresses within an aquifer system of interest, a unified model is needed that accounts for changes in volume strain based on a fixed point or plane such as the bedrock basin-fill contact at the base of the aquifer system.

This study presents the development and documentation of a fully three-dimensional granular displacement model that has the displacement field of solids as the dependent variable and incorporates the initial and ultimate bulk flux values to account for boundaries and heterogeneities within the aquifer system of interest.

Evaluation of the model is accomplished by first comparing horizontal granular movement in a homogeneous isotropic confined aquifer with available analytic solutions, secondly, a test simulation is developed to compare net change in vertical strain (displacement within a model layer) within fine-grained interbeds using the already published interbed storage model and the granular displacement model. Simulation results from the granular displacement model compare favorably with those of the interbed storage model for the test simulation.

Simulation results of horizontal granular movement within a confined aquifer have shown that large displacements can occur in the vicinity of the pumping well. These displacements are exacerbated by cyclically turning the well on and off. When a well is pumped and then followed by an episode of injection, the aquifer experiences a rapid change from compressional strain to extensional strain. In addition, large displacements can occur within a short period of time. The change in the rate of pumping or injection places immense strain on the aquifer matrix which over time may weaken the granular structure making it susceptible to failure, particularly along planes of weakness such as preexisting lineaments, faults, or other subvertical heterogeneities.

In distal regions from the main pumping center, where drawdowns are minimal or nonexistent, extensional radial strains are at their maximum during pumping and horizontal displacements may be surprisingly large. However, the driving mechanism remains the same as in regions near the pumping center; namely, the bulk force that operates on both the interstitial fluid and granular matrix. Horizontal granular movement beyond the radius of influence (determined by hydraulic head) is likely to be responsible for most fissure development along the perimeter of heavily pumped aquifers. In this outer area the change from extensional to compressional strain may not be the controlling factor for fissure development. The main factor may be the increasing magnitude of radial displacement itself coupled with geologic influences such as shallow bedrock knobs and subvertical range-front faults that create potential plains of discontinuity where the front grain (closest to pumping well) freely moves while the rear grain remains stationary. The important concept to remember is that the granular matrix beyond where drawdowns occur in heavily pumped aquifers is vary much in motion.

Vertical and horizontal flow barriers were incorporated into a homogeneous unconfined aquifer system to evaluate how these impermeable barriers may influence granular movement. The vertical linear barrier tends to increase vertical and horizontal movement in the vicinity of the pumped well. The net displacement of the well is toward the barrier (compressional strain) yet some radially convergent horizontal displacement occurs on the side of the barrier away from the pumped well. The simulated displacement field indicates that pumped wells near hydrologic barriers tend to increase subsidence between the well and the flow barrier. Potential fissuring is most likely to occur away from the barrier in the region of zero radial strain.

A horizontal flow barrier within the interior cells of the model in layer two resulted in large vertical displacements at the edge of the barrier even though pumping was at the center of the grid network far from the edge of the barrier. This finding tends to confirm what is occurring in Las Vegas Valley where a large amount of subsidence has been measured in the region where a thick confining units transition into coarser grained materials in the western part of the valley. Typical subsidence models would not predict subsidence to occur in this region.

Simulation results from multiple pumping or injection wells reveals clearly that the locations of greatest displacements do not occur at intuitively obvious locations. The displacement field from each individual well are additive. The result is a displacement field that does not tend to move directly toward or away from any single well.

This work points to the need to focus data collection efforts on both horizontal and vertical displacements within a complex hydrogeologic setting. These data will help to further evaluate the effectiveness of the granular displacement model in calculating the displacement field of solids and potential for fissure development in heavily pumped arid-zone aquifers. Although fissure development is typically an arid-zone phenomenon, horizontal movements in unconsolidated aquifers are a consequence of Darcy's law regardless of climatic conditions.

References

Anderson, M.P., and H.F. Wang, *Introduction to groundwater modeling*, San Francisco, W.H. Freeman and Co., 237 pp., 1982.

Anderson, S.R., Potential for aquifer compaction, land subsidence, and earth fissures in Avra Valley, Pima and Pinal Counties, Arizona, *U.S. Geol. Sur. Hydro. Inv. Atlas HA-718, 1:250,000, 3 sh., 1989.*

Bear, J., and M.Y. Corapcioglu, Mathematical model for regional land subsidence due to pumping 2. Integrated aquifer subsidence equations for vertical and horizontal displacements, *Water Resour. Res.*, *17*, 947-958, 1981.

Bell, J.W., J.G. Price, and M.D. Mifflin, Subsidence-induced fissuring along pre-existing faults in Las Vegas Valley, Nevada, in *Engineering Geology into the 21st Century, pp. 66-75*, Proceedings of the 25th Annual Meeting of the Assoc. of Eng. Geol., Long Beach, Calif., 1992.

Biot, M.A., General theory of three-dimensional consolidation, *J. of App. Phys., 12(2),* 155-164, 1941.

Biot, M.A., Theory of elasticity and consolidation for a porous anisotropic solid, *J. of App. Phy., 26(2), 182-185, 1955.*

Carpenter, M.C., Earth-fissures movements associated with fluctuations in ground-water levels near the Picacho Mountains, south-central Arizona, 1980-84, *U.S. Geol. Surv. Open File Rep. 90-561,* 62 pp., 1991.

Gambolati, G., and R.A. Freeze, Mathematical simulation of the subsidence of Venice, 1: Theory, *Water Resour. Res., 9,* 721-733, 1973.

Gambolati, G., P. Gatto, and R.A. Freeze, Mathematical simulation of the subsidence of Venice, 2: Results, *Water Resour. Res., 10,* 563-577, 1974.

Gersevanov, N.M., *The foundation of dynamics of soils,* (in Russian), Stroiizdat, Leningrad, 1934

Helm, D.C., A postulated relation between granular movement and Darcy's law for transient flow, in Evaluation and Prediction of Subsidence, S.K. Saxena (ed.), p. 417-440, American Society of Civil Engineers, New York, 1979.

Helm, D.C., One-dimensional simulation of aquifer system compaction near Pixley, Calif., 1: constant parameters, *Water Resour. Res., 11,* 465-478, 1975.

Helm, D.C., One-dimensional simulation of aquifer system compaction near Pixley, Calif., 2: stress-dependent parameters, *Water Resour. Res., 12,* 375-391, 1976.

Helm, D.C., Simulation of aquitard compaction due to changes in stress, *Trans. Am. Geophys. Union, 53(11), 979,* 1972.

Helm, D.C., Conceptual aspects of subsidence due to fluid withdrawal, in *Recent trends in hydrogeology,* T.N. Narasimhan (ed.), pp. 103-139, Geol. Soc. Amer. Spec. Paper 189, 1982.

Helm, D.C, Analysis of sedimentary skeletal deformation in a confined aquifer and the resulting drawdown, in *Groundwater hydraulics*, J.S. Rosenshein and G.D. Bennett, (eds.), pp. 29-82, Am. Geophys. Union Water. Resour. Mono. 9, 1984.

Helm, D.C., Three-dimensional consolidation theory in terms of the velocity of solids, *Geotechnique, 37(3)*, 369-392, 1987.

Helm, D.C, Hydraulic forces that play a role in generating fissures at depth, *Bull. Assoc. Eng. Geologists*, (in print), 1993.

Helm, D.C., Calculation of horizontal aquifer movement, within a Theis-Thiem confined system, *Water Resour. Res.*, 30(4), p. 953-964, 1994.

Hubbert, M.K., The theory of ground-water motion, *J. of Geol*, 48, 785-944, 1969

Jacob, C.E., Flow of groundwater, in *Engineering Hydraulics*, H. Rouse, ed., 321-386, John Wiley, New York, 1950.

Leake, S.A., and D.E. Prudic, Documentation of a computer program to simulate aquifer-system compaction using the modular finite-difference ground-water flow model, *U.S. Geol. Surv. Open File Rep. 88-482*, 80 pp. 1988.

Leake, S.A., Interbed storage changes and compaction in models of regional groundwater flow, *Water Resour. Res., 26*, 1939-1950, 1990.

Lewis R.W., and B. Schrefler, A fully coupled consolidation model of the subsidence of Venice, *Water Resour. Res., 14*, 223-230, 1978.

Lohman, S.W., Ground-water hydraulics, *U.S. Geol. Surv. Prof. Paper 708*, 70 pp., 1972

McDonald, M.G., and A.W. Harbaugh, A modular three-dimensional finite-difference ground-water flow model, *U.S. Geol. Surv. Tech. Water Resour. Invest.*, Book 6 Chap. A1, 586 pp., 1988.

Narasimhan, T.N., and P.A. Witherspoon, Numerical model for land subsidence in shallow groundwater systems, *Land Subsidence*, A.I. Johnson (ed.), pp. 131-144. International Association of Sci. Hydro. Publ. 121, New York; Unipubl., 1977.

Peaceman, D.W., *Fundamentals of numerical reservoir simulation*, New York, Elsevier Sci. Pub. Co., 176, 1977.

Poland, J.F., B.E. Lofgren, R.L. Ireland, and R.G. Pugh, Land subsidence in the San Joaquin Valley, California, as of 1972, *U.S. Geol. Surv. Prof. Paper 437-H,* 78 p., 1975.

Poland, J.F., and G.H. Davis, Land subsidence due to fluid withdrawal of fluids, *Rev. Eng. Geol.,* 2, 187-269, 1969.

Press, W.H., S.A. Teulolsky, W.T.Vetterling, and B.P. Flannery, *Numerical recipes in Fortran,* Second Ed., Cambridge Univ. Press, 963 pp.

Safri, N.M., and G. Pinder, Vertical and horizontal land deformation in a desaturating porous medium, *Adv. Water Resour.,* 2, 19-25, 1979.

Scott, R.F., *Principles of soil mechanics,* Palo Alto Calif., Addison-Wesley Pub. Co., Inc., 550 p., 1963

Terzaghi, K., *From theory to practice in soil mechanics,* New York, Wiley, 425 pp., 1960.

Verruijt, A., Elastic storage of aquifers, in *Flow through Porous Media,* R.J.M. DeWiest (ed.), New York, Academic Press, p. 331-377, 1969

Wolff, R.G., Relationship between horizontal strain near a well and reverse water level fluctuation, *Water Resour. Res.,* 6, 1721-1728, 1970.

Yerkes, R.F., and R.O. Castle, Surface deformation associated with oil and gas field operations in the United States, *in* L.J. Tison, ed., Land subsidence, v. 1: *Internat. Assoc. Sci. Hydro. Pub. 88,* 55-66., 1969

APPENDIX A:

MODEL DOCUMENTATION

The documentation of the granular displacement model includes a complete discussion of each of the three modules written. The order of each module discussion is as follows: (1) bulk flux module, (2) displacement module, and (3) vector plotting module. Each module includes an in depth description of subroutines used, design, all variables and parameters, and flow paths outlining each routine. Also included are input instructions for using the modules. Finally, the main program of MODFLOW is included because modifications were made to this segment of the code. All sections of the main program written in small letters are additions required by the granular displacement model. The portions of the main program in capital letters are part of the standard code and published modules already available. No documentation for the main code is included. The user is referred to the MODFLOW instruction manuel (McDonald and Harbaugh, 1988) for further discussion of the main program.

Bulk Flux Package Input

Input for Bulk Flux Package (QBK)is read from the unit specified in IUNIT(16).

FOR EACH SIMULATION

QBK1AL

1. Data:	IQBKOC	IQBTYP	IQBSS
Format;	I10	I10	I10

QBK1RP

2. Data:	IQBKFM	IQBKUN
Format:	I10	I10

The following arrays (items 3-5) describe each layer. Whether an array is read depends on the layer type code (LAYCON)

FOR ALL LAYER TYPE CODES

3. Data: RATIO
Module: U2DREL

IF THE LAYER TYPE CODE IS ZERO OR TWO

4. Data: BASE
Module: U2DREL
5. Data: QSURF
Module: U2DREL

FOR EACH STRESS PERIOD

QBK1ST

Read data set 6 once for every layer

6. Data: HSS
Module: U2DREL

QBK1OT

7. Data:	IQBKPR	IQBKSV
Format	I10	I10

Explanation of Fields Used in Input Instructions

IQBKOC-is a flag for printing or saving displacement values

If IQBKOC ≥ 0 Bulk flux data wdll be read and printed or saved.

If IQBKOC<=0 Bulk flux data will not be calculated, printed, or saved.

IQBSS-is the flag for reading and using ultimate specific discharge or bulk flux values

If IQBSS ≡ 0 Ultimate values of bulk flux are not read or used.

If IQBSS ≥0 Ultimate values of bulk flux are read and used.

IQBTYP-is a flag for the part of an aquifer simulated when only one pumping well location is used. This flag adjusts the bulk flux values according to adjusted pumping rates for simulations using one half or one quarter of the areal extent of the aquifer.

If IQBTYP=0 the entire aquifer is simulated. This type is used when multiple pumping well locations are used.

If IQBTYP=1 one half of the aquifer is simulated and the pumping location is centered along one boundary.

If IQBTYP=2 one quarter of the aquifer is simulated and the pumping location is centered at the corner of two converging boundaries.

IQBKFM-is the print format code for the bulk flux values in all three component directions. The print codes are listed in the modular model documentation p. 14-3.

IQBKUN-is the unit number where bulk flux values will be saved

If IQBKUN=0 bulk flux values will not be saved

If IQBKUN≥0 bulk flux values will be saved on the unit number specified according to the time step flag IQBKSV described below.

RATIO-is the ratio of vertical to horizontal hydraulic conductivity of the layer being simulated. RATIO can not be greater than 1.0. For a single layer simulation use 1.0.

BASE-is the elevation of the bottom of the layer, It is synonymous to BOT but is used with layer types zero and two.

QSURF-is the elevation of the top of the layer. It is synonymous to TOP but is used with layer types zero and two.

HSS-are the ultimate steady-state hydraulic head values obtained by simulating steady-state conditions with the current aquifer properties while assuming either a constant head or constant recharge rate to the topmost active layer. If a rate is used it must be equal to the total discharge from the modeled area.

IQBKPR-is the output flag for printing bulk flux values

If IQBKPR≥0 bulk flux values for each layer will be printed

If IQBKPR<=0 bulk flux values will not be printed

IQBKSV-is the output flag for saving bulk flux values.

If IQBKSV<=0 bulk flux values are not saved for all layers

If IQBKSV≥0 bulk flux values are saved for all layers

Module Documentation for the Bulk Flux Package

The bulk flux package (QBK) has four primary modules and three submodules. All the primary modules are called by the MAIN program.

Primary Modules.

QBK1AL	Allocates space for data arrays. Reads bulk flux calculation flag and type of simulation invoked.
QBK1RP	Reads print and save flags. Sets active cells for displacement. Initializes bulk flux arrays to zero. Reads information needed to calculate eccentricity and cell thickness. Calls submodule SQBK1L.
QBK1FM	Calculates cell centers where pumping and injection occurs. Calls SQBK1E. Calculates the bulk flux for each cell where recharge is specified.
QBK1ST	Reads ultimate steady-state heads and calls submodule SQBK1U.
QBK1OT	Reads print and save flags for bulk flux. Prints or saves bulk flux values for all three component directions after each stress period when print or save flags are set.

Submodules

SQBK1L	Calculates initial thickness of all cells in the grid, even those where transmissivity has been specified.
SQBK1E	Calculates eccentricity, major and minor ellipsoid axes, surface area of ellipsoid and finally bulk flux for each well. Adds component contribution of bulk flux for all pumping or injection wells specified for a given stress period.
SQBK1W	Calculates adjustment to bulk flux in z direction at the wellbore.
SQBK1U	Calculates the ultimate specific discharge which is equivalent to the ultimate bulk flux.

Narrative for Module QBK1AL

This module allocates space for data arrays for the bulk flux package. It also reads flag for calculating bulk flux terms and whether a single well quadrant or half space is simulated, or whether a full aquifer simulation is to be simulated.

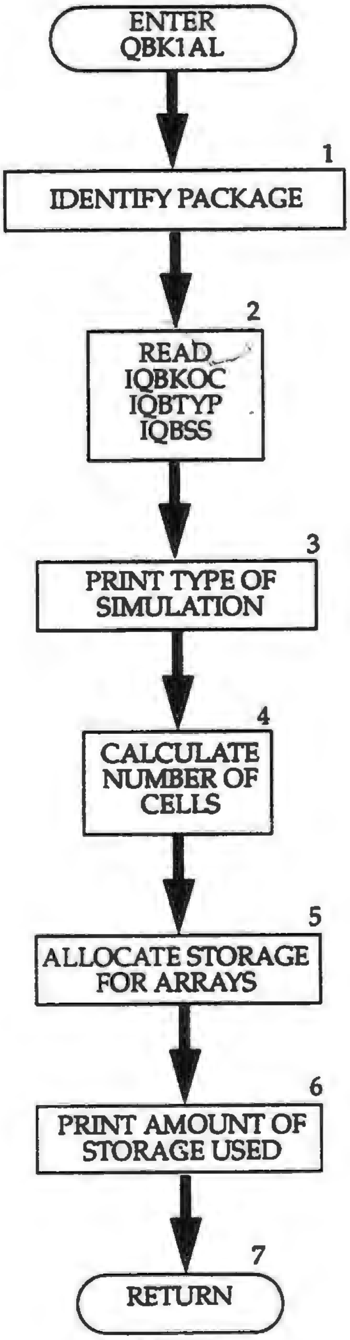
1. Identify package.
2. Read flags for calculation of bulk flux terms, and type of simulation.
3. Print the type of simulation.
4. Calculate total number of cells in the model grid.
5. Allocate storage for the following arrays:
    - QBX Bulk flux in the X direction for each cell in the grid.
    - QBY Bulk flux in the Y direction for each cell in the grid.
    - QBZ Bulk flux in the Z direction for each cell in the grid.
    - SPX Specific discharge in the X direction for each cell in the grid.
    - SPY Specific discharge in the Y direction for each cell in the grid.
    - SPZ Specific discharge in the Z direction for each cell in the grid.
    - QX Volume fluid flux in the X direction for each cell in the grid.
    - QY Volume fluid flux in the Y direction for each cell in the grid.
    - QZ Volume fluid flux in the Z direction for each cell in the grid.
    - HSS Ultimate steady-state heads for each cell in the grid.
    - RATIO Ratio of vertical to horizontal hydraulic conductivity for all cells.
    - BASE The bottom altitude of the cell where transmissivities are specified.
    - QSURF The top altitude of the cell where transmissivities are specified.
    - DELL Cell thickness.
    - IACT Boundary array for displacement.
6. Print amount of storage used by the bulk flux package.
7. RETURN.

Flow Chart for Module USL1AL

QBK1OC is the bulk flux calculation flag.
If QBK1OC ≥0 Bulk flux calculations will be made.
If QBK1OC <=0 Bulk flux calculations will not be made.

IQBTYP is the flag identifying the type of simulation.
If IQBTYP=0 A full aquifer for single or multiple wells is simulated.
If IQBTYP=1 The aquifer is represented as a half circle with the well at the center of the half circle. This is a single well simulation only.
If IQBTYP=2 The aquifer is represented as a quarter circle with the well at the center of the wedge. This is a single well simulation only. This approach is used to save space for symmetric single well simulations.

IQBSS is the flag indicating whether ultimate steady-state heads are read.



```
      SUBROUTINE QBK1AL(ISUM,LENX,LCQBX,LCQBY,LCQBZ,LCRAX,LCBASE,
     1LCQSURF,LCDELL,NCOL,NROW,NLAY,IN,IOUT,IQBKOC,LCIACT,IQBTYP,
     2LCSPX,LCSPY,LCSPZ,LCQX,LCQY,LCQZ,LCHSS,IQBSS
C
C     *******************************************************************
C     ALLOCATE ARRAY STORAGE FOR QBULK PACKAGE
C     *******************************************************************
C
C     SPECIFICATIONS:
C     ---------------------------------------------------------------
C     ---------------------------------------------------------------
C
C1------IDENTIFY PACKAGE
      WRITE(IOUT,1)IN
    1 FORMAT(//1X,'QBK1 -- QBULK PACKAGE SETS UP INITIAL CONDITIONS
     1 FOR GRANULAR FLOW MODEL,  INPUT READ FROM',I3)
C
C2------READ FLAG FOR CALCULATING BULK FLUX TERMS  TYPE OF
     1 SIMULATION, AND WHETHER ULTIMATE HEADS ARE NEEDED.
      READ(IN,2) IQBKOC,IQBTYP,IQBSS
    2 FORMAT(3I10)
      IF(IQBKOC.GT.0) WRITE(IOUT,10)
   10 FORMAT(1X,'OUTPUT CONTROL RECORDS FOR QBK1 PACKAGE WILL
     1 BE READ EACH TIME STEP.')
      IF(IQBSS.EQ.0) WRITE(IOUT,8)
    8 FORMAT(1X,'ULTIMATE HEADS ARE NOT READ')
      IF(IQBSS.NE.0) WRITE(IOUT,9)
    9 FORMAT(1X,'ULTIMATE HEADS ARE READ FOR EACH STRESS PERIOD')
C
C3------PRINT TYPE OF SIMULATION
      IF(IQBTYP.GT.2) IQBTYP=0
      IF(IQBTYP.EQ.0) WRITE(IOUT,12)
   12 FORMAT(1X,'FULL CIRCLE SIMULATION')
      IF(IQBTYP.EQ.1) WRITE(IOUT,14)
   14 FORMAT(1X,'HALF CIRCLE SIMULATION')
      IF(IQBTYP.EQ.2) WRITE(IOUT,16)
   16 FORMAT(1X,'QUARTER CIRCLE SIMULATION')
```

```
C
C4------CALCULATE TOTAL NUMBER OF CELLS
      NRCL=NROW*NCOL*NLAY
C
C5------ALLOCATE STORAGE FOR ARRAYS
      IQBLK=ISUM
      LCQBX=ISUM
      ISUM=ISUM+NRCL
      LCQBY=ISUM
      ISUM=ISUM+NRCL
      LCQBZ=ISUM
      ISUM=ISUM+NRCL
      LCRAT=ISUM
      ISUM=ISUM+NRCL
      LCBASE=ISUM
      ISUM=ISUM+NRCL
      LCQSURF=ISUM
      ISUM=ISUM+NRCL
      LCDELL=ISUM
      ISUM=ISUM+NRCL
      LCIACT=ISUM
      ISUM=ISUM+NRCL
      LCHSS=ISUM
      ISUM=ISUM+NRCL
      LCQX=ISUM
      ISUM=ISUM+NRCL
      LCQY=ISUM
      ISUM=ISUM+NRCL
      LCQZ=ISUM
      ISUM=ISUM+NRCL
      LCSPX=ISUM
      ISUM=ISUM+NRCL
      LCSPY=ISUM
      ISUM=ISUM+NRCL
      LCSPZ=ISUM
      ISUM=ISUM+NRCL
C
```

```
C6-----PRINT AMOUNT OF STORAGE USED BY THE BULK FLUX PACKAGE
      ISP=ISUM-IQBLK
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I8,'ELEMENTS USED IN QBULK PACKAGE')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I8,'ELEMENTS IN X ARRAY USED OUT OF',I8)
C-----IF THERE ISN'T ENOUGH SPACE IN THE X ARRAY THEN PRINT
C-------A WARNING MESSAGE.
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C7------RETURN
      RETURN
      END
```

List of Variables for Module QBK1AL

Variable	Range	Definition
IN	Package	Primary unit number from which input for this package will be read
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IQBKOC	Package	Flag for calculating bulk flux terms. >0 calculate bulk flux terms. <= 0 do not calculate bulk flux terms.
IQBLK	Module	Before this module allocates space, IQBLK is set equal to ISUM. After allocation, IQBLK is subtracted from ISUM to get ISP, the amount of space in the X array allocated by this module.
IQBTYP	Package	Flag indicating type of simulation. =0 Full aquifer is simulated =1 One half aquifer is simulated with well at center of circle. =2 One quarter of aquifer is simulated with well at center of aquifer wedge.
ISP	Module	Number of words in the X array allocated by this module
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	ISUM-1
LCBASE	Package	Location in the X array of the first element of array BASE.
LCDELL	Global	Location in the X array of the first element of array DELL.
LCHSS	Package	Location of the first element of array HSS.
LCIACT	Global	Location in the X array of the first element of array IACT.
LCSPX	Global	Location in the X array of the first element of array SPX.
LCSPY	Global	Location in the Y array of the first element of array SPY.
LCSPZ	Global	Location in the Z array of the first element of array SPZ
LCQBX	Global	Location in the X array of the first element of array QBX.
LCQBY	Global	Location in the Y array of the first element of array QBY.
LCQBZ	Global	Location in the Z array of the first element of array QBZ.
LCQSURF	Package	Location of the first element of array QSURF.
LCQX	Package	Location in the X array of the first element of array QX.
LCQY	Package	Location in the Y array of the first element of array QY.
LCQZ	Package	Location in the Z array of the first element of array QZ.
LCRAT	Global	Location of the first element of array RATIO.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRCL	Module	Number of cells in the grid.
NROW	Global	Number of rows in the grid.

ilm:

Narrative for Module QBK1RP

This module reads the format and unit numbers for printing and saving the bulk flux terms at the end of each stress period. It also sets the boundary array for displacement, initializes bulk flux arrays, reads the ratio of vertical to horizontal hydraulic conductivity and altitudes of the top and bottom of cells when transmissivity is used. These altitudes are used along with those specified in the BCF package to calculate cell thickness.

Module QBK1RP calls submodule SQBK1L and performs its tasks in the following order:

1. Read format and unit numbers for printing or saving bulk flux values, respectively.
2. Set active boundary cells for displacement.
3. Initialize bulk flux values to zero.
4. Read the ratio of vertical to horizontal hydraulic conductivity for all cells in the grid. Read the top and bottom cell elevations when LAYCON = 0 or LAYCON = Z
5. Calculate the initial thickness of all cells in the grid.
6. RETURN.

Flow Chart for Module QBK1RP

IQBKFM is the flag identifying the format type for printing bulk flux values. The flag number is identified in McDonald and Harbaugh (1988, pg. 14-5).

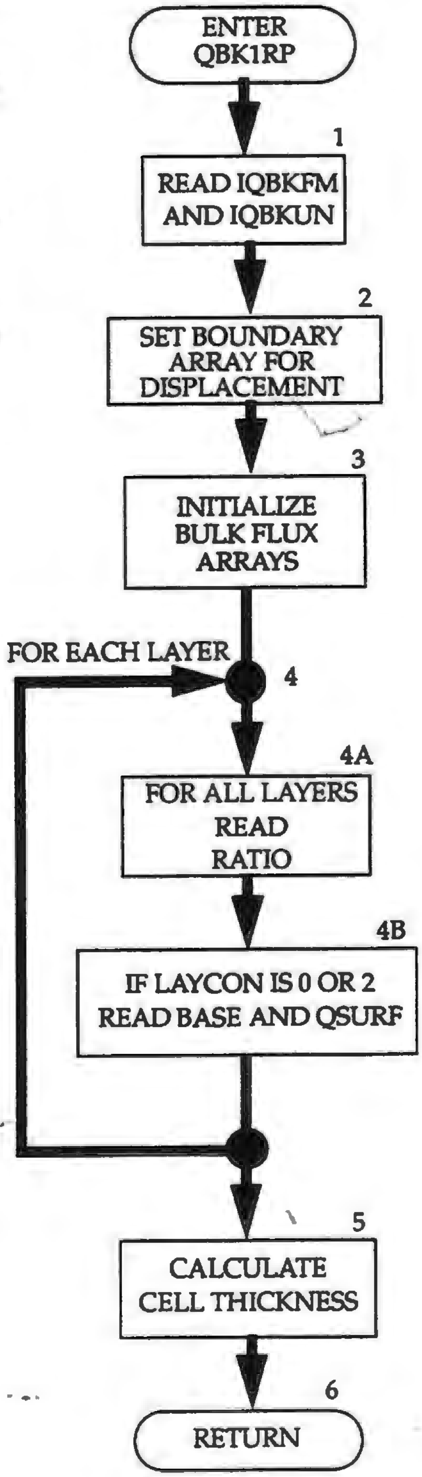
IQBKUN is a flag and unit number to which the bulk flux terms will be written.
If IQBKUN<=0 the bulk flux terms are not saved.
If IQBKUN>0 the bulk flux terms are saved on the unit number specified.

LAYCON is a layer-type code (one for each layer).
- 0 - confined
- 1 - unconfined
- 2 - confined/unconfined but transmissivity is constant
- 3 - confined/unconfined

RATIO is the ratio of vertical to horizontal hydraulic conductivity. It must not exceed 1.0.

BASE is the elevation of the bottom of the cell. It is read only when the LAYCON is 0 or Z.

QSURF is the altitude of the top of the cell. It is read only when the LAYCON is 0 or Z.



```
      SUBROUTINE QBK1RP(QBX,QBY,QBZ,RATIO,BASE,QSURF,NODES,NCOL,
     1NROW,NLAY,IN,IOUT,IACT,IBOUND,IQBKOC,IQBKFM,IQBKUN,DELC,
     2 DELR,DELL, TOP,BOT,HNEW)
C
C     ***********************************************************************
C     INITIALIZE QBULK ARRAYS AND READ KV/KH RATIO, BASE AND
C     SURFACE ELEVATIONS
C     ***********************************************************************
C
C   SPECIFICATIONS:
C---------------------------------------------------------------
      DOUBLE PRECISION HNEW
      CHARACTER*4 ANAME
C
      DIMENSION ANAME(6,3),RATIO(NODES),BASE(NODES),QSURF(NODES),
     1 QBX(NCOL,NROW,NLAY),QBY(NCOL,NROW,NLAY),QBZ(NCOL,
     2NROW,NLAY),IBOUND(NODES),IACT(NODES),TOP(NCOL,NROW,
     3NLAY),BOT(NCOL,NROW,NLAY),HNEW(NCOL,NROW,NLAY),
     4 DELC(NROW), DELR(NCOL),DELL(NCOL,NROW,NLAY)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1   ANAME(6,1) /'   V','ERTI','TO H','ORIZ.','CON','DUCT'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1   ANAME(6,2) /'    ','CELL',' BA','SE E','LEVA','TION'/
      DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),
     1   ANAME(6,3) /'  CE','LL S','URFA','CE E','LEVA','TION'/
C
       COMMON /FLWCOM/LAYCON(80)
C---------------------------------------------------------------
C
C1------READ FORMAT AND UNIT NUMBER FOR PRINTING OR SAVING QBULK
      1  TERMS.
      IF(IQBKOC.LE.0) GO TO 500
      READ(IN,5) IQBKFM,IQBKUN
    5  FORMAT(2I10)
      WRITE(IOUT,6) IQBKFM
    6  FORMAT(1X,'  QBULK PRINT FORMAT IS NUMBER',I4)
```

```
      IF(IQBKUN.GT.0) WRITE(IOUT,7) IQBKUN
    7 FORMAT(1X,'UNIT FOR SAVING QBULK VALUES IS',I4)
C--------CALCULATE NUMBER OF CELLS PER LAYER
  500 NRC=NCOL*NROW
C
C2------SET IACT=IBOUND; TEST WHERE IACT SHOULD BE INACTIVE.
      DO 25 I=1,NODES
      IACT(I)=IBOUND(I)
C-------IF IBOUND=9 SET ACTIVE CELL TO ZERO DISPLACEMENT
      IF(IBOUND(I).EQ.9) IACT(I)=0
   25 CONTINUE
C
C3-------INITIALIZE QB(X,Y,Z) ARRAYS TO ZERO
      DO 26 K=1,NLAY
      DO 26 I=1,NROW
      DO 26 J=1,NCOL
      QBX(J,I,K)=0.
      QBY(J,I,K)=0.
      QBZ(J,I,K)=0.
   26 CONTINUE
C
C4-------FOR EACH LAYER IN THE GRID:
C4A------READ IN KZZ/KHH RATIO DATA FOR EACH CELL
      KR=0
      DO 50 K=1,NLAY
      IF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) KR=KR+1
      KK=K
      LOC=1+(K-1)*NRC
      LOCR=1+(KR-1)*NRC
      CALL U2DREL(RATIO(LOC),ANAME(1,1),NROW,NCOL,KK,IN,IOUT)
      IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ. 3) GOTO 50
C4B--------READ THE SURFACE AND BASE ELEVATIONS OF CELL WHERE
C         LAYCON IS 0 OR 2.
      CALL U2DREL(BASE(LOCR),ANAME(1,2),NROW,NCOL,KK,IN,IOUT)
      CALL U2DREL(QSURF(LOCR),ANAME(1,3),NROW,NCOL,KK,IN,IOUT)
   50 CONTINUE
C
```

```
C5------ DETERMINE CELL THICKNESS (DELL) FOR ALL CELLS IN THE GRID.
      KB=0
      KT=0
      KR=0
      DO 150 K=1,NLAY
      KK=K
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
   IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) KB=KB+1
      IF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) KR=KR+1
C------CALL SUBMODULE TO CALCULATE THE VERTICAL DIMENSIONS OF
C     GRID
      CALL SQBK1L(KK,KT,KB,KR,TOP,BOT,BASE,QSURF,DELL,NCOL,NROW,
      1  NLAY,HNEW)
  150 CONTINUE
C
C6------RETURN
      RETURN
      END
```

List of Variables for Module OBKIRP

Variable	Range	Definition
BASE	Package	DIMENSION (NCOL,NROW,NLAY), Elevation of cell bottom of each cell in layers where LAYCON is 0 or 2.
BOT	Global	DIMENSION (NCOL,NROW,NLAY), Elevation of the bottom of each cell in layers where LAYCON is 1 or 3.
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains width of row I.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains width of column J.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Initial estimate of head in each cell in order to determine thickness for topmost layer.
I	Module	Index for nodes and rows
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array identifying active cells in which displacement is calculated.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. <0 constant-head cell =0 inactive cell >0 variable-head cell
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IQBKFM	Package	Flag for identifying what format to use to print bulk flux values
IQBKCC	Package	Flag for calculating bulk flux terms. >0 calculate bulk flux terms. <=0 do not calculate bulk flux terms.
IQBKUN	Package	Flag identifying unit number to which bulk flux terms will be saved.
J	Module	Index for columns
K	Module	Index for layers
KB	Module	Counter for the number of layers for which the bottom elevation is needed (LAYCON = 1 or 3).
KK	Module	Temporary variable set equal to K. KK is used as an actual argument in subroutine calls to avoid using the DO loop variable K as an argument, which causes problems for some compilers.
KR	Module	Counter for the number of layers for which BASE and QSURF arrays need to be read (LAYCON = 0 or 2).
KT	Module	Counter for the number of layers for which the top elevation is needed (LAYCON = 1or 3).
LAYCON	Global	DIMENSION (80), Layer-type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined.

List of Variables for Module OBKIRP (Continued)

Variable	Range	Definition
		2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity is variable).
LOC	Module	Pointer to parts of the RATIO arrays corresponding to particular layers.
LCXZR	Module	Pointer to parts of the elevation arrays corresponding to particular layers.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NODES	Module	Number of cells in the grid.
NRC	Module	Number of cells in a layer.
NROW	Global	Number of rows in the grid.
QSURF	Package	DIMENSION (NCOL,NROW,NLAY), Elevation of cell top of each cell in layers where LAYCON is 0 or 2.
QBX	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the X direction.
QBY	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Y direction.
QBZ	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Z direction.
RATIO	Global	DIMENSION (NCOL,NROW,NLAY), Ratio of vertical to horizontal hydraulic conductivity specified for all LAYCON types.
TOP	Global	DIMENSION (NCOL,NROW,NLAY), Elevation of the top of each cell in layers where LAYCON is 2 or 3.

Narrative for Module OBKIST

This module reads the ultimate steady-state heads if IQBSS is set. These heads are ultimately used to calculate the steady-state specific discharge or ultimate bulk flux values which is done in submodule SQBK1U which is called by this module.

This module performs its tasks in the following order:

1. Check IQBSS flag
2. Read steady-state heads if IQBSS is set.
3. Call SQBK1U if IQBSS is set.
4. RETURN

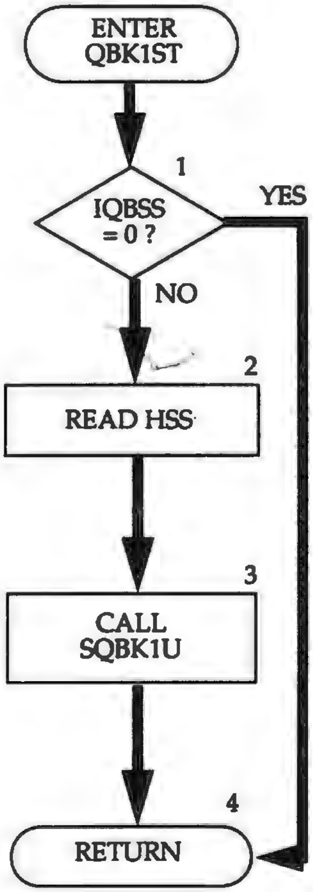
Flow Chart for Module QBK1ST

IQBSS is a flag indicating whether ultimate steady-state heads are read and ultimate bulk fluxes are calculated.

=0 HSS not read and ultimate bulk fluxes not calculated

≠0 HSS read and ultimate bulk fluxes are calculated

HSS is the ultimate steady-state head array. One array is read for each layer



```
      SUBROUTINE QBK1ST(HSS,IBOUND,TRAN,HY,CV,TOP,BOT,DELL,DELR,
     1 DELC,NCOL,NROW,NLAY,SPX,SPY,SPZ,QX,QY,QZ,IN,IOUT,IQBSS)
C
C     *******************************************************************************
C     THIS SUBROUTINE READS THE STEADY STATE HYDRAULIC HEAD
C     VALUES FOR THE IMPOSED STRESSES ON THE SYSTEM. THIS
C     ROUTINE ALSO CALLS THE ROUTINE TO CALCULATE ULTIMATE
C     BULK FLUX VALUES.
C     *******************************************************************************
C
C     SPECIFICATIONS:
C-----------------------------------------------------------------------------------
      CHARACTER*4 ANAME
C
      DIMENSION HSS(NODES),IBOUND(NCOL,NROW,NLAY),TRAN(NCOL,
     1 NROW,NLAY),HY(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY),
     2 TOP(NCOL,NROW,NLAY),BOT(NCOL,NROW,NLAY),DELL(NCOL,
     6 NROW,NLAY),DELR(NCOL),DELC(NROW),SPX(NCOL,NROW,NLAY),
     7 SPY(NCOL,NROW,NLAY),SPZ(NCOL,NROW,NLAY),QX(NCOL,NROW,NLAY),
     8 QY(NCOL,NROW,NLAY),QZ(NCOL,NROW,NLAY),ANAME(6,1)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1 ANAME(6,1) /'    '/    '/  UL','TIMA'/TE H'/EADS'7
C
      COMMON /FLWCOM/LAYCON(80)
C--------------------------------------------------
C1----CHECK TO SEE IF ULTIMATE HEADS ARE READ AND ULTIMATE QBULK
C     VALUES CALCULATED
      IF(IQBSS.EQ.0) GOTO 70
C2----READ ULTIMATE STEADY STATE HEAD VALUES FOR THIS STRESS PERIOD
      NCR=NCOL*NROW
      DO 15 K=1,NLAY
      KK=K
      LOC=1+(KK-1)*NCR

      CALL U2DREL(HSS(LOC),ANAME(1,1),NROW,NCOL,KK,IN,IOUT)
   15 CONTINUE
```

```
C
C3—CALL ROUTINE TO CALCULATE ULTIMATE BULK FLUX VALUES
      CALL SQBK1U(HSS,IBOUND,TRAN,HY,CV,TOP,BOT,DELL,DELR,DELC,
     1 NCOL,NROW,NLAY,SPX,SPY,SPZ,QX,QY,QZ)
C
C4—RETURN
   70 RETURN
      END
```

List of Variables for Module OBK1ST

Variable	Range	Definition
ANAME	Module	Label for printout of input array.
BOT	Global	DIMENSION (NCOL,NROW,NBOT), Elevation of the bottom of each layer. (NBOT is the number of layers for which LAYCON = 1 or 3.)
CV	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the layer direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains width of row I.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains width of column J.
HSS	Package	DIMENSION (NCOL,NROW,NLAY), Ultimate steady-state heads used to calculate the ultimate bulk flux values.
HY	Global	DIMENSION(NCOL,NROW,NLAY), Hydraulic conductivity in layers specified as LAYCON = 1 or 3.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT=6.
IQBSS	Package	Flag indicating whether ultimate steady-state heads are read and ultimate bulk fluxes are calculated and used.
K	Module	Index for layers
KK	Module	Temporary variable set equal to K.
LAYCON	Global	DIMENSION (80) Layer type code: 0 - Layer strictly confined 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity constant). 3 - Layer confined/unconfined (transmissivity variable).
LOG	Module	Pointer to parts of the HSS arrays corresponding to particular layers.
NCOL	Global	Number of columns in the grid.
NCR	Module	Number of cells in a layer.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
SPX	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge or ultimate bulk flux in the X direction.
SPY	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge or ultimate bulk flux in the Y direction.
SPZ	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge or ultimate bulk flux in the Z direction.

List of Variables for Module OBKIST (Continued)

Variable	Range	Definition
QX	Package	DIMENSION(NCOL,NROW,NLAY), Volume fluid flux across the right cell face in the X direction
QY	Package	DIMENSION(NCOL,NROW,NLAY), Volume fluid flux across the front cell face in the Y direction
QZ	Package	DIMENSION (NCOL,NROW,NLAY), Volume fluid flux across the bottom cell face in the Z direction.
TRAN	Global	DIMENSION(NCOL,NROW,NLAY), Transmissivity specified when LAYCON = 0 or 2.

Narrative for Module QBKIFM

This module performs the calculations for the bulk flux terms that are needed at the start of each stress period.The initial stresses that produce a bulk flux within the aquifer are pumping, injection, or recharge. For pumping, a bulk flux is determined on the basis of the pumping rate and the eccentricity at each well location. The calculated bulk flux for each well is summed to produce a final bulk flux value that is used in the displacement model. For recharge, only the z component of bulk flux is affected because it is assumed that recharge occurs from the top of the cell. If recharge is desired along a different face then the Well Package should be used to simulate recharge.

Module QBK1FM calls submodule SQBK1E and performs its tasks in the following order:

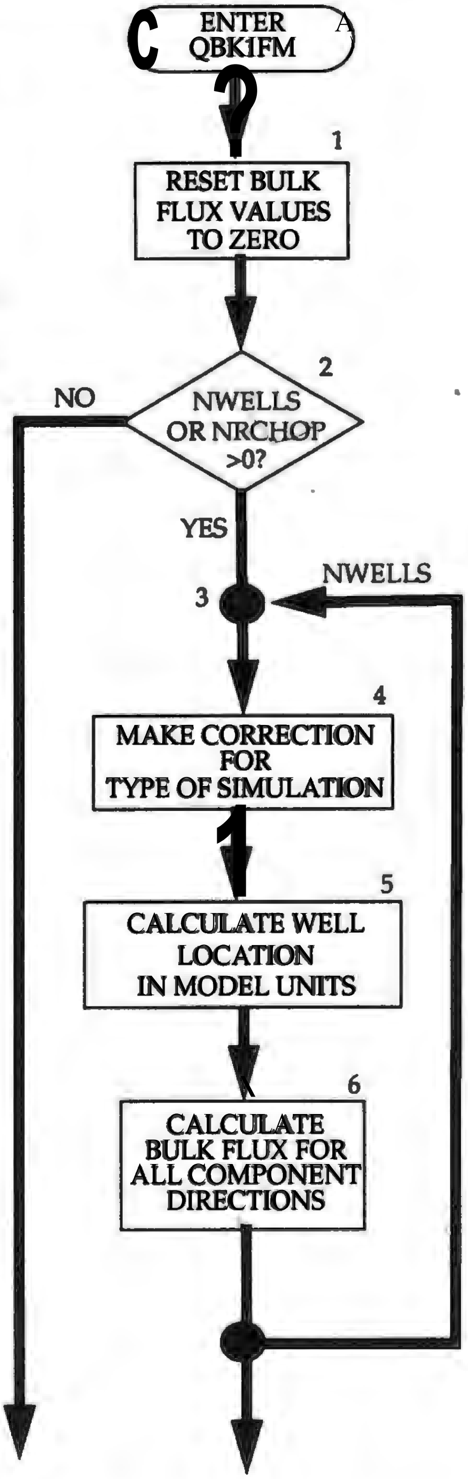
1. Reset bulk flux values to zero at the start of each new stress period.
2. Check to see if initial stress conditions exist.
3. Loop through each well and calculate a bulk flux term for each component direction for each well.
4. Make correction in pumping rate for the type of simulation
5. Calculate the actual well location in model units
6. Calculate the total bulk flux for each component direction due to pumping.
7. Check to see if recharge is present. If recharge exists add bulk flux components resulting from recharge.
8. RETURN.

Flow Chart for Module QBK1FM

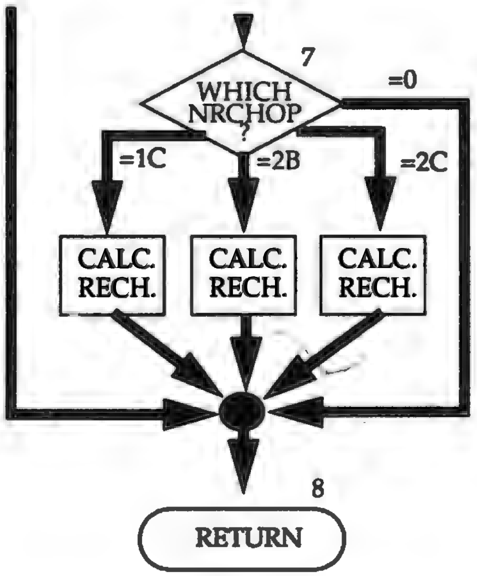
NWELLS is the number of wells specified in the Well Package for the current stress period.

NRCHOP is the recharge option code specified in the Recharge Package.

- =1 Recharge is only to the top grid layer.
- =2 Vertical distribution of recharge is specified in array IRCH.
- =3 Recharge is applied to the highest active cell in each vertical column. A constant-head node intercepts recharge and prevents deeper infiltration.



Flow Chart for Module QBKIFM (Continued)



```
      SUBROUTINE QBK1FM(NWELLS,DELR,DELC,DELL,WELL,QBX,QBY,QBZ,
     1 RATIO,IBOUND,NROW,NCOL,NLAY,NRCHOP,IRCH,RECH,IACT,TOTIM,
     2 KPER,IQBTYP)
C
C    ******************************************************************
C    CALCULATE DIRECTIONAL COMPONENTS OF QBULK AS INFLUENCED
C    BY PUMPING AND RECHARGE
C    ******************************************************************
C
C    SPECIFICATIONS:
C------------------------------------------------------------------------
C
      DOUBLE PRECISION HNEW
      DIMENSION WELL(4,NWELLS),QBX(NCOL,NROW,NLAY),QBY(NCOL,
     1 NROW,NLAY),QBZ(NCOL,NROW,NLAY),RATIO(NCOL,NROW,NLAY),
     2 IBOUND(NCOL,NROW,NLAY),DELR(NCOL),DELC(NROW),DELL(NCOL,
     3 NROW,NLAY),RECH(NCOL,NROW),IRCH(NCOL,NROW),
     4 IACT(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C------------------------------------------------------------------------
C
C1------RESET QBULK VALUES TO ZERO EACH TIME STEP
        DO 23 K=1,NLAY
        DO 23 I=1,NROW
        DO 23 J=1,NCOL
        QBX(J,I,K)=0.
        QBY(J,I,K)=0.
        QBZ(J,I,K)=0.
   23   CONTINUE
C
C2------CHECK TO SEE IF INITIAL STRESS CONDITIONS EXIST
        IF(NWELLS.LE.0 .AND. NRCHOP.LE.0) RETURN
C
C3------LOOP THROUGH NWELLS AND CALCULATE BULK FLUX TERMS
        IF(NWELLS.LE.0) GOTO 1000
        DO 50 LL=1,NWELLS
```

```
          IR=WELL(2,LL)
          IC=WELL(3,LL)
          IL=WELL(1,LL)
          Q=WELL(4,LL)
C
C4------MAKE CORRECTION FOR TYPE OF SIMULATION
          IF(IQBTYP.EQ.1) Q=Q*Z
          IF(IQBTYP.EQ.2) Q=Q*4.
C
C5------CALCULATE ACTUAL WELL LOCATION IN MODEL UNITS
          DC = 0.
          DR = 0.
          DL = 0.
C
          DO 20 I=1,IR
          DC=DC+DELC(I)
   20     CONTINUE
          DO 25 J=1,IC
          DR=DR+DELR(J)
   25     CONTINUE
          DO 30 K=1,IL
          DL=DL+DELL(IC,IR,K)
   30     CONTINUE
C
          YO1=DC-DELC(IR)
           YO2=DC
          X01=DR-DELR(IC)
           X01=DR
          Z01=DL - DELL(IC,IR,IL)
           Z02=DL
C
C6------CALL SUBROUTINE TO CALCULATE QBULK FOR AN ELLIPSOID
          CALL SQBK1E(QBX,QBY,QBZ,QB4T,YO1,YO2,XO1,XO2,ZO1;Z02,QINCOL,
          1NROW,NLAY,RATIOE,ER,DELC,DELL,IL,IR,IC)
   50     CONTINUE
C
C7------NOW ADD EFFECTS OF RECHARGE TO QBULK DEPENDING ON
```

```
C       NRCHOP.
 1000   IF(NRCHOP.GT.3 .OR. NRCHOP.LT.1) GOTO 2000
        DO 2 IR=1,NROW
        DO 2 IC=1,NCOL
C
C7A-----CALCULATE LOCATION AND RATE OF RECHARGE FOR NRCHOP=1
        IF(NRCHOP.NE.1) GOTO 100
        IF(IBOUND(IC,IR,1).EQ.0) GOTO 2
        IF(RECH(IC,IR).EQ.0.) GOTO 2
        QBZ(IC,IR,1)=QBZ(IC,IR,1)-RECH(IC,IR)/(DELR(IC)*DELC(IR))
        GOTO 2
C
C7B-----CALCULATE LOCATION AND RATE OF RECHARGE FOR NRCHOP=2
  100   IF(NRCHOP.NE.2) GOTO 200
        IL=IRCH(IC,IR)
        IF(IBOUND(IC,IR,IL).LE.0) GOTO 2
        IF(RECH(IC,IR).EQ.0.) GOTO 2
        QBZ(IC,IR,IL)=QBZ(IC,IR,IL)-RECH(IC,IR)/(DELR(IC)*DELC(IR))
        GOTO 2
C
C7C-----CALCULATE LOCATION AND RATE OF RECHARGE FOR NRCHOP=3
  200   DO 4 IL=1,NLAY
        IF(IBOUND(IC,IR,IL).LT.0) GOTO 2
        IF(IBOUND(IC,IR,IL).EQ.0) GOTO 4
        IF(RECH(IC,IR).EQ.0.) GOTO 2
        QBZ(IC,IR,IL)=QBZ(IC,IR,IL)-RECH(IC,IR)/(DELR(IC)*DELC(IR))
        GOTO 2
C
    4   CONTINUE
    2   CONTINUE
C
C8------RETURN
 2000   RETURN
        END
```

List of Variables for Module QBK1FM

Variable	Range	Definition
DC	Module	Location of the well in model units in the column direction
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains width of row I.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains width of column J.
DL	Module	Location of the well in model units in the layer direction.
DR	Module	Location of the well in model units in the row direction.
I	Module	Index for rows.
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array identifying active cells in which displacement is calculated.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell. <0 constant-head cell =0 inactive cell >0 variable-head cell
IC	Module	Index for column location of pumping or discharging well.
IL	Module	Index for layer location of pumping or discharging well.
IQBTYP	Package	Flag indicating type of simulation. =0 Full aquifer is simulated =1 One half aquifer is simulated with well at center of circle. =2 One quarter of aquifer is simulated with well at center of aquifer wedge.
IR	Module	Index for row location of pumping or discharging well.
J	Module	Index for columns.
K	Module	Index for layers.
KPER	Global	Stress period counter.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NRCHOP	Global	Recharge option: =1 Recharge is to the top grid layer. =2 Recharge is to the grid layer specified in array IRCH. =3 Recharge is to the highest variable-head cell which is not below a constant-head cell.
NROW	Global	Number of rows in the grid.
NWELLS	Global	Number of wells active during the current stress period.
Q	Global	Rate at which the well adds water to the aquifer (negative for discharging wells).
QBX	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the X-direction.
QBY	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Y-direction.
QBZ	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Z-direction.
RATIO	Global	DIMENSION (NCOL,NROW,NLAY), Ratio of vertical to

List of Variables for Module QBKIFM (Continued)

Variable	Range	Definition
		horizontal hydraulic conductivity.
RECH	Global	DIMENSION (NCOL, NROW), Recharge flow rate. Recharge flux is read into RECH and then multiplied by cell area to obtain recharge flow rate.
TOTIM	Global	Total simulation time.
WELL	Global	DIMENSION (4, MXWELL), For each well: layer, row, column, and recharge rate of the well.
XO1	Module	Distance in the X direction to the left edge of the cell containing the well.
XO2	Module	Distance in the X direction to the right edge of the cell containing the well
YO1	Module	Distance in the Y direction to the front edge of the cell containing the well.
YO2	Module	Distance in the Y direction to the back edge of the cell containing the well.
ZO1	Module	Distance in the Z direction to the top edge of the cell containing the well
ZO2	Module	Distance in the Z direction to the bottom edge of the cell containing the well

Narrative for Module QBKIOT

This module prints or saves bulk flux terms in all three component directions according to flags set by the user for each stress period. If flag IQBKPR is set for a stress period, then bulk flux values will be printed according to the format set by flag IQBKFM. First the X component values of bulk flux will be printed, followed by the Y and then Z component values, respectively. Similarly, if flag IQBKSV is set for a stress period, then unformated bulk flux values will be saved to disk.

Module QBKIOT is called each stress period and performs its functions in the following order:

1. Read flags for printing and saving bulk flux terms
2. Print bulk flux terms if flag IQBKPR is set.
3. Save bulk flux terms if flag IQBKSV is set.
4. RETURN.

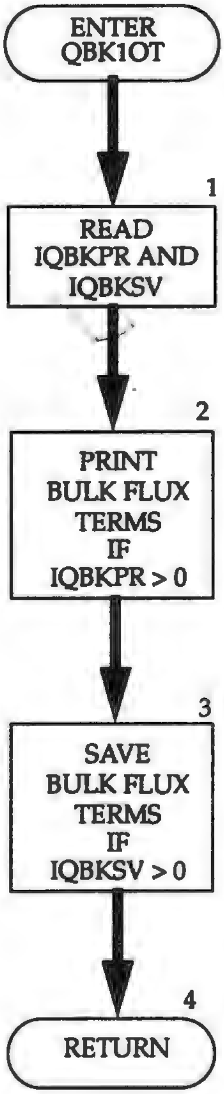
Flow Chart for Module QBK1OT

IQBKPR is the print flag for the current stress period.

- >0 print bulk flux terms
- <=0 do not print bulk flux terms.

IQBKSV is the save flag for the current stress period.

- >0 save bulk flux terms
- <= 0 do not save bulk flux terms



```
      SUBROUTINE QBK1OT(QBX,QBY,QBZ,NCOL,NROW,NLAY,IQBKOC,I
     1 QBKFM,IQBKUN,IN,IOUT,KPER,KSTP,PERTIM,TOTIM)
C
C     ***************************************************************
C     PRINTS DIRECTIONAL COMPONENTS OF QBULK
C     ***************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 TEXT
      DIMENSION QBX(NCOL,NROW,NLAY),QBY(NCOL,NROW,NLAY),
     1 QBZ(NCOL,NROW, NLAY),TEXT(4,3)                  X  ,
      DATA TEXT(1,1),TEXT(2,1),TEXT(3,1),TEXT(4,1) /'    ','    ',
     1 'X-Q','BULK'/,TEXT(1,2),TEXT(2,2),TEXT(3,2),TEXT(4,2) /'    ',
     2 '    ','Y-Q','BULK'/,TEXT(1,3),TEXT(2,3),TEXT(3,3),TEXT(4,3) /
     3 '    ','    ','Z-Q','BULK'/
C     ------------------------------------------------------------------
C
C1------READ FLAGS FOR PRINTING AND SAVING
      IF(IQBKOC.LE.0) GOTO 500
      READ(IN,5) IQBKPR,IQBKSV
    5 FORMAT(2I10)
      WRITE(IOUT,6) IQBKPR,IQBKSV
    6 FORMAT(/,1X,'FLAGS FOR PRINTING AND STORING QBULK VALUES:'/
     1 '  IQBKPR    IQBKSV  '/
     2 '-------------------'/
     3 I6,I10)
C
C2------PRINT QBULK VALUES IF IQBKPR IS SET
      IF(IQBKPR.LE.0) GOTO 20
      DO 10 K=1,NLAY
      IF(IQBKFM.LT.0) THEN
      CALL ULAPRS(QBX(1,1,K),TEXT(1,1),KSTP,KPER,NCOL,NROW,K,-IQBKFM,
     1   IOUT)
      CALL ULAPRS(QBY(1,1,K),TEXT(1,2),KSTP,KPER,NCOL,NROW,K,-IQBKFM,
     1   IOUT)
      CALL ULAPRS(QBZ(1,1,K),TEXT(1,3),KSTP,KPER,NCOL,NROW,K,-IQBKFM,
```

```
      1   IOUT)
      ENDIF
      IF(IQBKFM.GE.0) THEN
      CALL ULAPRW(QBX(1,1,K),TEXT(1,1),KSTP,KPER,NCOL,NROW,K,IQBKFM,
      1   IOUT)
      CALL ULAPRW(QBY(1,1,K),TEXT(1,2),KSTP,KPER,NCOL,NROW,K,IQBKFM,
      1   IOUT)
      CALL ULAPRW(QBZ(1,1,K),TEXT(1,3),KSTP,KPER,NCOL,NROW,K,IQBKFM,
      1   IOUT)
      ENDIF
   10 CONTINUE
C
C3------SAVE QBULK VALUES IF IQBKSV IS SET
   20 IF(IQBKSV.LE.0) GOTO 500
      DO 15 K=1,NLAY
      CALL ULASAV(QBX(1,1,K),TEXT(1,1),KSTP,KPER,PERTIM,TOTIM,NCOL,
      1 NROW,K,IQBKUN)
      CALL ULASAV(QBY(1,1,K),TEXT(1,2),KSTP,KPER,PERTIM,TOTIM,NCOL,
      1 NROW,K,IQBKUN)
      CALL ULASAV(QBZ(1,1,K),TEXT(1,3),KSTP,KPER,PERTIM,TOTIM,NCOL,
      1 NROW,K,IQBKUN)
   15 CONTINUE
C
C4-------RETURN
  900 RETURN
      END
```

List of Variables for Module QBK1OT

Variable	Range	Definition
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IQBKFM	Package	Flag for identifying what format to use to print bulk flux values
IQBKOC	Package	Flag for calculating bulk flux terms. >0 calculate bulk flux terms. <=0 do not calculate bulk flux terms.
IQBKPR	Module	Flag for printing after current stress period. >0 print bulk flux values. <=0 do not print bulk flux values.
IQBKSV	Module	Flag for saving after current stress period. >0 save bulk flux values. <=0 do not save bulk flux values.
IQBKUN	Package	Flag identifying unit number to which bulk flux terms will be saved.
K	Module	Counter for layers.
KPER	Global	Stress period counter.
KSTP	Global	Time step counter.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
PERTIM	Global	Elapsed time during current stress period.
QBX	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the X direction.
QBY	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Y direction.
QBZ	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Z direction.
TEXT	Module	Label for printout of input array.
TOTIM	Global	Elapsed time in the simulation.

Narrative for Module SQBKIL

This submodule is called by QBK1RP once for each layer in the grid to calculate the thickness of all the cells in the active grid. Modflow only calculates cell thickness when the LAYCON type is 1 or 3. This module assigns thickness to cells where the LAYCON type is 0 and 2 because the displacement model requires that cell thickness be known. The initial thickness for the water table layer (when LAYCON = 1) is set to be the difference between the steady-state water level in the cell minus the base elevation of that cell. Hence, it is assumed that the steady-state water levels have already been established before the displacement model is used. It should also be noted that if the water level rises above this initial thickness, the adjustment is made in the displacement package.

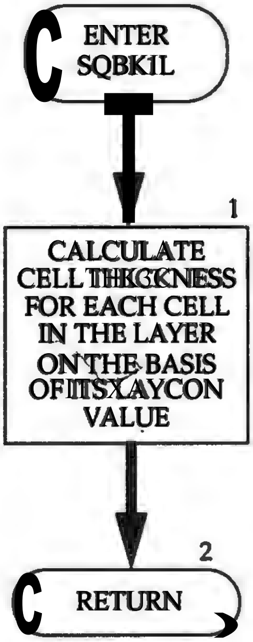
The Module SQBK1L performs its tasks in the following order:

1. Calculate cell thickness based on LAYCON type for the given layer.
2. RETURN.

Flow Chart for Module SQBK1L

LAYCON is a layer-type code (one for each layer).

0 - confined
1 - unconfined
2 - confined/unconfined but transmissivity is constant
3 - confined/unconfined



```
      SUBROUTINE SQBK1L(KK,KT,KB,KR,TOP,BOT,BASE,QSURF,DELL,NCOL,
     1 NROW,NLAY,HNEW)
C
C    ******************************************************************
C    COMPUTE VERTICAL GRID SPACING AT ROW AND COLUMN LOCATIONS
C    ******************************************************************
C
C    SPECIFICATIONS:
C-----------------------------------------------------------------------
      DOUBLE PRECISION HNEW
      DIMENSION TOP(NCOL,NROW,NLAY),BOT(NCOL,NROW,NLAY),
     1 BASE(NCOL,NROW,NLAY),QSURF(NCOL,NROW,NLAY),DELL(NCOL,
     2 NROW,NLAY),HNEW(NCOL,NROW,NLAY)
C
    COMMON /FLWCOM/LAYCON(80)
C-----------------------------------------------------------------------
C
C1------CALCULATE THICKNESS OF EACH CELL IN THE LAYER.
      DO25I=1,NROW
      DO25J=1,NCOL
      IF(LAYCON(KK).EQ.0 .OR. LAYCON(KK).EQ. 2) GOTO 100
      IF(LAYCON(KK).EQ.1) GOTO 40
      IF(KK.EQ.1)THEN
        DELL(J,I,KK)=HNEW(J,I,KK)-BOT(J,I,KB)
      ELSE
         DELL(J,I,KK)=TOP(J,I,KT)-BOT(J,I,KB)
      ENDIF
      GOTO 25
   40 DELL(J,I,KK)=HNEW(J,I,KK)-BOT(J,I,KB)
      GOTO 25
C
  100 DELL(J,I,KK)=QSURF(J,I,KR)-BASE(J,I,KR)
   25 CONTINUE
C
C2------RETURN
      RETURN
      END
```

List of Variables for Module SOBKIL

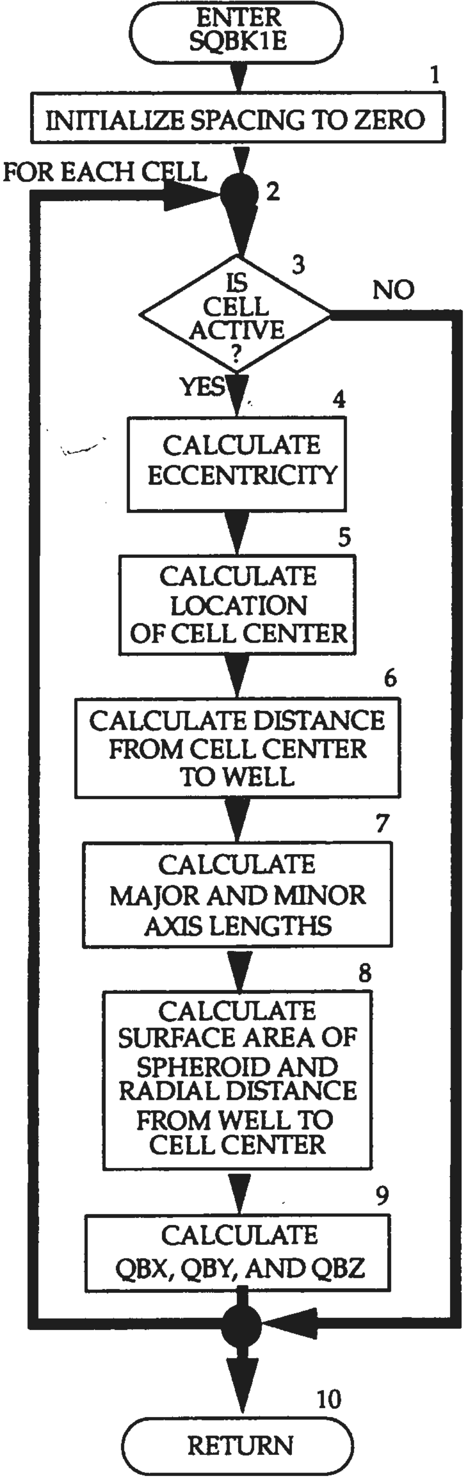
Variable	Range	Definition
BASE	Package	DIMENSION (NCOL,NROW,NLAY), Elevation of the bottom of cells in layers where LAYCON is 0 or 2.
BOX	Global	DIMENSION (NCOL,NROW,NLAY), Elevation of the bottom of cells in layers where LAYCON is 1 or 3.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Initial steady-state head value initially assigned in the Basic Package.
I	Module	Index for rows.
J	Module	Index for columns.
KB	Module	Counter for layers where BOX is needed (when LAYCON is 1 or 3).
KK	Module	Index for layers
KR	Module	Counter for layers whete BASE and QSURF are needed (when LAYCON is 0 or 2).
KT	Module	Counter for layers where TOP is needed (when LAYCON is 2 or 3).
LAYCON	Global	DIMENSION (80), Layer type code: 0 - Layer is strictly confined. 1 - Layer is strictly unconfined. 2 - Layer is confined/unconfined (transmissivity is constant 3 - Layer is confined/unconfined (transmissivity is variable).
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
QSURF	Package	DIMENSION (NCOL,NROW,NLAY), Elevation of the top of cells in layers where LAYCON is 0 or 2.
TOP	Global	DIMENSION (NCOL,NROW,NLAY), Elevation of the top of cells in layers where LAYCON is 2 or 3

Flow Chart for Module SQBK1E

QBX is the component of bulk flux in the X direction.

QBY is the component of bulk flux in the Y direction.

QBZ is the component of bulk flux in the Z direction.



```
      SUBROUTINE SQBK1E(QBX,QBY,QBZ,IACT,YO1,YO2,XO1,XO2,ZO1,ZO2,Q,
     1 NCOL,NROW,NLAY,RATIO,DELR,DELC,DELL,ILIR,IC)
C
C
C ******************************************************************************
C   CALCULATE QBULK IN THREE DIMENSIONS ASSUMING A PROLATE
C   SPHEROID AT THE CENTER OF EACH CELL. EXCEPTION IS AT THE WATER
C   TABLE WHERE QBULK IS CALCULATED AT THE BOUNDARY
C
C ******************************************************************************
C
C   SPECIFICATIONS:
C-------------------------------------------------------------------------------
      DIMENSION QBX(NCOL,NROW,NLAY),QBY(NCOL,NROW,NLAY),
     1 QBZ(NCOL,NROW,NLAY),IACT(NCOL,NROW,NLAY),RATIO(NCOL,
     2 NROW,NLAY),DELR(NCOL),DELC(NROW),DELL(NCOL,NROW,NLAY)
C
       PI=4.*ATAN(1.0)
C  -----------------------------------------------------------------------------
C
C1--------SET INITIAL SPACING TO ZERO
      DC = 0.
      DR = 0.
      DL = 0.
C
C2------LOOP THROUGH ENTIRE ACTIVE GRID AND CALCULATE QBULK
      DO 60 KK=1,NLAY
      DO 60 II=1,NROW
      DO 60 JJ=1,NCOL
C
C3--------CHECK TO SEE IF THE CELL IS ACTIVE
      IF(IACT(JJ,II,KK).EQ.0) GOTO 60
C
C4------CALCULATE THE ECCENTRICITY AT EACH CELL
      ECC=SQRT(1.0-(RATIO(JJ,II,KK)))
      ESQR=1.0-ECC*ECC
       DO 35 L=1,II
```

```
      DC=DC+DELC(L)
   35  CONTINUE
      DO 45 M=1,JJ
        DR=DR+DELR(M)
   45  CONTINUE
      DO 55 N=1,KK
      DL=DL+DELL(JJ,II,N)
   55  CONTINUE
C
C5----COMPUTE CELL-CENTER LOCATIONS
      Y=DC-DELC(II)/2.
      X=DR-DELR(JJ)/2.
       IF(KK.EQ.1 .OR. (KK.GT.1 .AND. IACT(JJ,II,KK-1).EQ.0)) THEN
       Z=DL-DELL(JJ,II,KK)
       ELSE
      Z=DL-DELL(JJ,II,KK)/2.
       ENDIF
       IF(NLAY.EQ.1) Z=DL-DELL(JJ,II,KK)/2.
      DC=0.
      DR=0.
      DL=0.
C
C6------CALCULATE THE DISTRANCE FROM THE WELL TO THE CELL OF
C     INTEREST AND CALCULATE ITS SQUARE
      IF(JJ.LT.IC) THEN
      XXO=X-XO1
      ELSEIF(JJ.GT.IC) THEN
      XXO=X-XO2
      ELSE
      XXO=0.
      ENDIF
      IF(II.LT.IR) THEN
      YYO=Y-YO1
      ELSEIF(II.GT.IR) THEN
      YYO=Y-YO2
      ELSE
      YYO=0.
```

```
      ENDIF
      IF(KK.LT.IL)THEN
      ZZO=Z-ZO1
      ELSEIF(KK.GT.IL)THEN
      ZZO=Z-ZO2
      ELSE
      ZZO=0.
      ENDIF
C
      XSQR=XXO*XXO
      YSQR=YYO*YYO
      ZSQR=ZZO*ZZO
C
C7------CALCULATE THE AXES LENGTHS OF THE OBLATE SPHERIOD
      RAD=SQRT(XSQR+YSQR+ZSQR)
      IF(RAD.LT.0.01) RAD=1.0
      IF(NLAY.EQ.1)GOTO 7
       IF(ZZO.LT.DELL(JJ,II,IL) .AND. DELL(JJ,II,IL).GT.RAD) RAD=DELL(JJ,II,IL)
       GOTO 8
    7   RAD=DELL(JJ,II,KK)
    8  AMAJ=SQRT(XSQR+YSQR+(ZSQR/ESQR))
        IF(AMAJ.LT.0.01) AMAJ=1.0
      BMIN=AMAJ*SQRT(ESQR)
C
C8------DETERMINE THE SURFACE AREA OF THE OBLATE SPHERIOD
      IF(ECC.LT.0.0001) THEN
      SA=4*PI*AMAJ*AMAJ
      ELSE
      XECC=ALOG((1+ECC)/(1-ECC))
      SA=(2*PI*AMAJ*AMAJ)+(PI*BMIN*BMIN/ECC*XECC)
      ENDIF
       IF(NLAY.EQ.1) SA=2*PI*AMAJ*AMAJ
C
C9------CALCULATE BULK FLUX FOR THIS WELL IN EACH COMPONENT
C        DIRECTION. THEN ADD THE VALUE TO ANY PREVIOUS BULK FLUX
C        FROM OTHER WELLS.
       QBX(JJ,II,KK)=QBX(JJ,II,KK)+Q*XXO/(SA*RAD)
```

```
          QBY(JJ,II,KK)=QBY(JJ,II,KK)+Q*YYO/(SA*RAD)
           IF(ZZO.EQ.0) GOTO 100
           IF(XXO.EQ.0 .AND. YYO.EQ.0) THEN
           CALL SQBK1W(ZZO,Q,ESQR,DELR(JJ),DELC(II),ECC,ZQBZ)
            QBZ(JJ,II,KK)=QBZ(JJ,II,KK)+0.80*ZQBZ+0.20*(Q*ZZO/(SA*RAD))
            GOTO 60
            ENDIF
    100   QBZ(JJ,II,KK)=QBZ(JJ,II,KK)+Q*ZZO/(SA*RAD)
     60   CONTINUE
C
C10-------RETURN
          RETURN
          END
```

List of Variables for Module SOBKIE

Variable	Range	Definition
AMAJ	Module	Length of the major axis of the oblate spheroid. The major axis is aligned in the X and Y directions. Thus it is assumed that there is no horizontal anisotropy.
BMIN	Module	Length of the minor axis of the oblate spheroid. The minor axis is aligned in the Z direction.
DC	Module	Distance to cell of interest along a column.
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
DL	Module	Distance to cell of interest along a layer.
DR	Module	Distance to cell of interest along a row.
ECC	Module	Eccentricity of the cell.
ESQR	Module	1-ECC**2.
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array for displacement. >0 cell is active <=0 cell is inactive
II	Module	Index for rows.
IC	Package	Column identifying location of pumping well.
IL	Package	Layer identifying location of pumping well
IR	Package	Row identifying location of pumping well.
JJ	Module	Index for columns.
KK	Module	Index for layers.
L	Module	Index for current row number.
M	Module	Index for current column number.
N	Module	Index for current layer number.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
PI	Module	Equivalent to π.
Q	Global	Pumping or injection rate.
QBX	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the X-direction.
QBY	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Y-direction.
QBZ	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in the Z-direction.
RAD	Module	Radial distance from the center of the cell containing the well to the center of the cell of interest.
RATIO	Global	DIMENSION (NCOL,NROW,NLAY), Ratio of vertical to horizontal hydraulic conductivity.
SA	Module	Surface area of the oblate spheroid.

List of Variables for Module SOBKIE (Continued)

Variable	Range	Definition
X	Module	Distance in the X direction to the center of the cell of interest
XECC	Module	Termporay variable to combine log-transformed eccentricity values for computing the surface area
XO1	Package	Distance in the X directionto the left edge of the cell containing the well.
XO2	Package	Distance in the X direction to the right edge of the cell containing the well
XXO	Module	Distance in the X direction between edge of cell containing the well to the center of the current cell.
XSQR	Module	XXO*XXO.
Y	Module	Distance in the Y direction to the center of the cell of interest
YO1	Package	Distance in the Y direction to the front edge of the cell containing the well.
YO2	Package	Distance in the Y direction to the back edge of the cell containing the well
YYO	Module	Distance in the Y direction between edge of cell containing the well to the center of the current cell.
YSQR	Module	YYO*YYO
Z	Module	Distance in the Z direction to the center of the cell of interest
ZO1	Package	Distance in the Z direction to the top edge of the cell containing the well.
ZO2	Package	Distance in the Z direction to the bottom edge of the cell containing the well.
ZZO	Module	Distance in the Z direction between edge of the cell containing the well to the center of the current cell.
ZSQR	Module	ZZO*ZZO
ZQBW	Module	Correction term for bulk flux in the Z direction

Narrative for Module SQBK1W

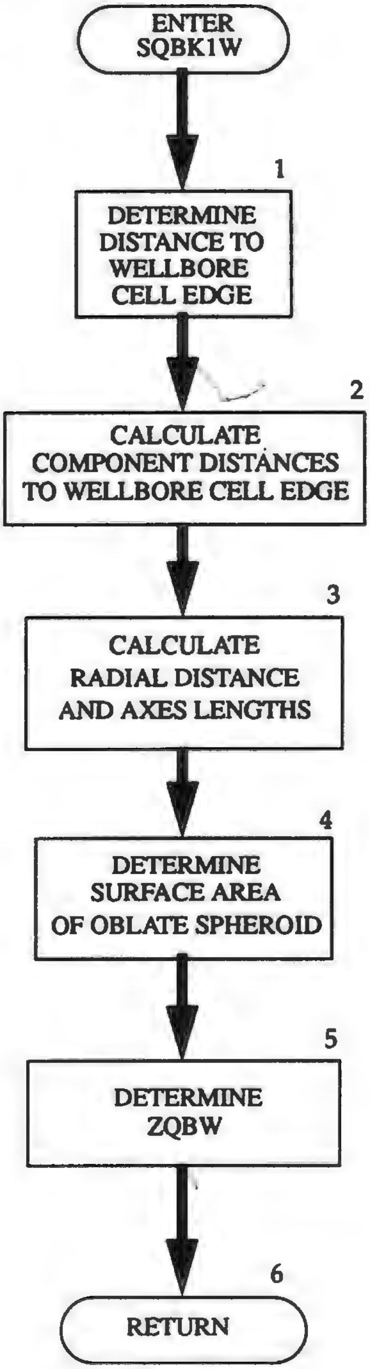
Module SQBK1W is called by module SQBK1E if a correction to the bulk flux in the z direction needs to be made at the wellbore. An adjustment is made at the row and column location containing the well so that the bulk flux is not calculated for simply a point, but rather is adjusted taking into accoint the area of the cell containing the pumping well. This correction keeps the calculated displacements from becoming unrealistically large at the wellbore.

This submodule performs its tasks in the following order:

1. Determine distance from wellbore center to cell edge containing the pumping well.
2. Calculate directional component of radius from wellbore to cell edge.
3. Calculate radial distance to cell edge and spheroid axes lengths.
4. Determine surface area of prolate spheroid.
5. Determine adjusted bulk flux in z direction
6. RETURN

Flow Chart for Module SQBK1W

ZQBW is the adjusted value of bulk flux in the Z direction at the wellbore.



List of Variables for Module SQBK1W

Variable	Range	Definition
AMAJ	Module	Length of the major axis of the oblate spheroid. The major axis is aligned in the X and Y directions. Thus it is assumed that there is no horizontal anisotropy.
BMIN	Module	Length of the minor axis of the oblate spheroid. The minor axis is aligned in the Z direction.
DCWELL	Module	Width of the column containing the pumping well
DRWELL	Module	Width of the row containing the pumping well.
ECC	Package	Eccentricity of the cell.
ESQR	Package	1-ECC**2.
PI	Module	Equivalent to π.
Q	Global	Pumping or injection rate.
RAD	Module	Radial distance from the center of the cell containing the well to the edge of the cell containing the well.
SA	Module	Surface area of the spheroid.
XECC	Module	Temporary variable to combine log-transformed eccentricity values for computing the surface area
XSQR	Module	XXOZ*XXOZ
XXOZ	Module	Radius of cell containing well in the X direction.
YSQR	Module	YYOZ*YYOZ
YYOZ	Module	Radius of cell containing well in the Y direction.
ZQBW	Module	Adjusted value of bulk flux in the z direction at the wellbore.
ZSQR	Package	ZZO*ZZO

Narrative for Module SOBKIU

This module calculates the ultimate steady-state specific discharge for a given set of stresses applied to the aquifer system. This ultimate specific discharge field is identical to the ultimate bulk flux field according to the Darcy-Gersevanov-Helm law. That is, at the new steady-state the velocity of solids is equal to zero. These new ultimate bulk flux values are substituted into the governing equation for the bulk flux after approximately 5 minutes of simulation time. The true distribution between the initial and ultimate bulk flux values needs to be determined empirically and is beyond the scope of this study.

This module is called by QBK1ST and performs its tasks in the following order:

1. Calculate volume flux through right face of cell
2. Calculate volume flux through front face of cell
3. Calculate volume flux through lower face of cell.
4. Calculate cell thickness
5. Calculate ultimate values of specific discharge in the X direction
6. Calculate ultimate values of specific discharge in the Y direction.
7. Calculate ultimate values of specific discharge in the Z direction.
8. RETURN

Flow Chart for Module SOBK1U

QX is the volume fluid flux through the right cell face in the X direction.

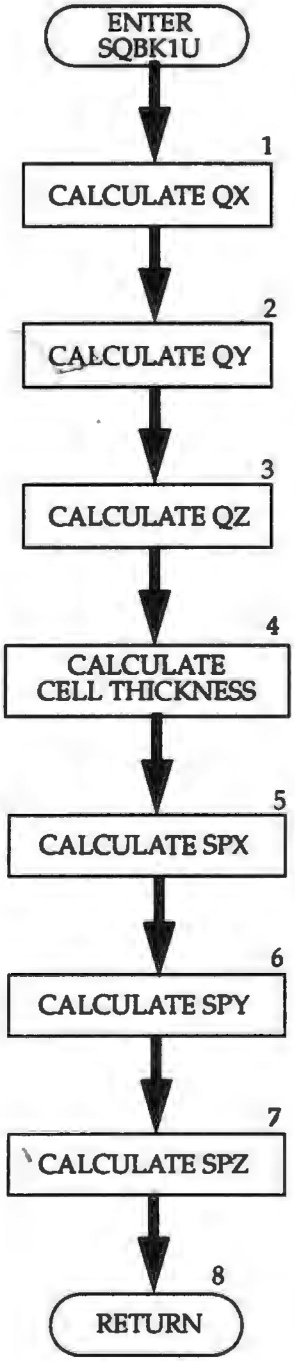
QY is the volume fluid flux through the front cell face in the Y direction.

QZ is the volume fluid flux through the lower cell face in the Z direction.

SPX is the ultimate steady-state specific storage values in the X direction

SPY is the ultimate steady-state specific storage values in the Y direction.

SPZ is the ultimate steady-state specific storage values in the Z direction.



```
      SUBROUTINE SQBK1U(HSS,IBOUND,TRAN,HY,CV,TOP,BOT,DELL,DELR,
     1DELC,NCOL,NROW,NLAY,SPX,SPY,SPZ,QX,QY,QZ)
C
C ********************************************************************************
C  THIS SUBROUTINE CALCULATES THE ULITMATE VALUE OF BULK FLUX
C  FROM STEADY-STATE HYDRAULIC HEAD VALUES FOR THE IMPOSED
C  STRESSES ON THE SYSTEM BEING ANALYZED.
C ********************************************************************************
C
C  SPECIFICATIONS:
C--------------------------------------------------------------------------------
      DIMENSION HSS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),TRAN(
     1 NCOL,NROW,NLAY),HY(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY),
     2 TOP(NCOL,NROW,NLAY),BOT(NCOL,NROW,NLAY),DELL(NCOL,
     3 NROW,NLAY),DELR(NCOL),DELC(NROW),SPX(NCOL,NROW,NLAY),
     4 SPY(NCOL,NROW,NLAY),SPZ(NCOL,NROW,NLAY),QX(NCOL,NROW,NLAY),
     5 QX(NCOL,NROW,NLAY),QY(NCOL,NROW,NLAY),QZ(NCOL,NROW,NLAY)
C
     COMMON /FLWCOM/LAYCON(80)
C ------------------------------------------------------------------------------
C
      DO 11 K=1,NLAY
      DO 11 J=1,NCOL
      DO 11 I=1,NROW
      QX(J,I,K)=0.
      QY(J,I,K)=0.
      QZ(J,I,K)=0.
   11 CONTINUE
C
      NCM1=NCOL-1
      IF(NCM1.LT.1) GOTO 105

C
C1------FOR EACH CELL CALCULATE FLOW THROUGH RIGHT FACE AND STORE
C        IN QX
      KB=0
      KT=0
```

```
      KR=0
      DO 100 K=1,NLAY
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ. 1) KB=KB+1
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
      IF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) KR=KR+1
      DO 100 I=1,NROW
      DO 100 J=1,NCM1
      IF((IBOUND(J,I,K).LE.0).AND.(IBOUND(J+1,I,K).LE.0)) GOTO 100
      HDIFF=HSS(J,I,K)-HSS(J+1,I,K)
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) THEN
      HD=HSS(J,I,K)
      IF(LAYCON(K).EQ.1) GOTO 51
      IF(HD.GT.TOP(J,I,KT)) HD=TOP(J,I,KT)
   51 T1=HY(J,I,KB)*(HD-BOT(J,I,KB))
      T2=HY(J+1,I,,KB)*(HD-BOT(J,I,KB))
      ELSEIF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) THEN
      T1=TRAN(J,I,KR)
      T2=TRAN(J+1,I,KR)
      ENDIF
      IF(T1.EQ.0. .OR. T2.EQ.0.) GOTO 100
      CR=2*T1*T2*DELC(I)/(T1*DELR(J+1)+T2*DELR(J))
      QX(J,I,K)=HDIFF*CR
  100 CONTINUE
C
  105 NRM1=NROW-1
      IF(NRM1.LT.1) GOTO 205
C
C2----FOR EACH CELL CALCULATE FLOW THROUGH FRONT FACE AND STORE
C       IN QY
      KB=0
      KT=0
      KR=0
      DO 200 K=1,NLAY
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ. 1) KB=KB+1
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
      IF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) KR=KR+1
      DO 200 I=1,NROW
```

```
      DO 200 J=1,NCOL
      IF((IBOUND(J,I,K).LE.0).AND.(IBOUND(J,I+1,K).LE.0)) GOTO 200
      HDIFF=HSS(J,I,K)-HSS(J,I+1,K)
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) THEN
      HD=HSS(J,I,K)
      IF(LAYCON(K).EQ.1) GOTO 52
      IF(HD.GT.TOP(J,I,KT)) HD=TOP(J,I,KT)
   52 T1=HY(J,I,KB)*(HD-BOT(J,I,KB))
      T2=HY(J,I+1,KB)*(HD-BOT(J,I,KB))
      ELSEIF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) THEN
      T1=TRAN(J,I,KR)
      T2=TRAN(J,I+1,KR)
      ENDIF
      IF(T1.EQ.0. .OR. T2.EQ.0.) GOTO 100
      CC=2*T1*T2*DELR(J)/(T1*DELC(I+1)+T2*DELC(I))
      QY(J,I,K)=HDIFF*CC
  200 CONTINUE
C
  205 NLM1=NLAY-1
      IF(NLM1.LT.1) GOTO 500
C
C3----FOR EACH CELL CALCULATE FLOW THROUGH LOWER FACE AND STORE
C      IN QZ
      KT=0
      DO 300 K=1,NLM1
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
      DO 300 I=1,NROW
      DO 300 J=1,NCOL
      IF((IBOUND(J,I,K).LE.0).AND.(IBOUND(J,I,K+1).LE.0)) GOTO 300
      HD=HSS(J,I,K+1)
      IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GOTO 350
      TMP=HD
      IF(TMP.LT.TOP(J,I,KT+1)) HD=TOP(J,I,KT+1)
  350 HDIFF=HSS(J,I,K)-HD
      QZ(J,I,K)=HDIFF*CV(J,I,K)
  300 CONTINUE
C4----CALCULATE CELL THICKNESS
```

```
  500 KB=0
      KT=0
      DO 20 K=1,NLAY
      IF(LAYCON(K).EQ.2 .OR. LAYCON(K).EQ.3) KT=KT+1
      IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ.3) KB=KB+1
      DO 20 I=1,NROW
      DO 20 J=1,NCOL
      IF(IBOUND(J,I,K).LE.0) GOTO 20
      IF(LAYCON(K).NE.0 .AND. LAYCON(K).NE.2) GOTO 30
      THCK=DELL(J,I,K)
      GOTO 25
   30 HD=HSS(J,I,K)
      IF(LAYCON(K).EQ.1) GOTO 28
      IF(HD.GT.TOP(J,I,KT)) HD=TOP(J,I,KT)
   28 THCK=HD-BOT(J,I,KB)
C5----CALCULATE ULTIMATE SPECIFIC DISCHARGE IN X
   25 IF(NCOL.EQ.1)GOTO 26
      QX1=0.
      QX2=0.
      IF(J.NE.1 .AND. (J-1.NE.0.AND.J-1.NE.-1)) QX1=QX(J-1,I,K)
      IF(J.NE.NCOL) QX2=QX(J,I,K)
      XAREA=DELC(I)*THCK
      SPX(J,I,K)=0.5*(QX2/XAREA+QX1/XAREA)
C6----CALCULATE ULTIMATE SPECIFIC DISCHARGE IN Y
   26 IF(NROW.EQ.1)GOTO 27
      QY1=0.
      QY2=0.
      IF(I.NE.1 .AND. (I-1.NE.0.AND.I-1.NE.-1)) QY1=QY(J,I-1,K)
      IF(I.NE.NROW) QY2=QY(J,I,K)
      YAREA=DELR(J)*THCK
      SPY(J,I,K)=0.5*(QY2/YAREA+QY1/YAREA)
C7----CALCULATE ULTIMATE SPECIFIC DISCHARGE IN z
   27 IF(NLAY.EQ.1)GOTO 20
      QZ1=0.
      QZ2=0.
      IF(K.NE.1 .AND. (K-1.NE.0.AND.J-1.NE.-1)) QZ1=QZ(J,I,K-1)
      IF(K.NE.NLAY) QZ2=QZ(J,I,K)
```

```
      ZAREA=DELR(J)*DELC(I)
      SPZ(J,I,K)=0.5*(QZ2/ZAREA+QZ1/ZAREA)
   20 CONTINUE
C
C8——RETURN
      RETURN
      END
```

List of Variables for Module SOBK1U

Variable	Range	Definition
BOT	Global	DIMENSION (NCOL,NROW,NBOT), Elevation of the bottom of each layer. (NBOT is the number of layers for which LAYCON = 1 or 3.)
CC	Module	Temporary variable for conductance in the column direction.
CR	Module	Temporary variable for conductance in the row direction.
CV	Global	DIMENSION (NCOL,NROW,NLAY), Conductance in the layer direction. CV(J,I,K) contains conductance between nodes (J,I,K) and (J,I,K+1).
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains width of row I.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains width of column J.
HD	Module	Temporary variable for HSS.
HDIFF	Module	Head difference between adjacent rows, columns, or layers.
HSS	Package	DIMENSION (NCOL,NROW,NLAY), Ultimate steady-state heads used to calculate the ultimate bulk flux values.
I	Module	Index for rows.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell <0, constant-head cell =0, inactive cell >0, variable-head cell.
J	Module	Index for columns.
K	Module	Index for layers
KB	Module	Counter for the number of layers for which the bottom elevation is needed (LAYCON = 1 or 3).
KT	Module	Counter for the number of layers for which the top elevation is needed (LAYCON = 2 or 3).
KR	Module	Counter for the number of layers for which the transmissivity is needed (LAYCON = 0 or 2).
LAYCON	Global	DIMENSION (80) Layer type code: 0 - Layer strictly confined 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity constant). 3 - Layer confined/unconfined (transmissivity variable).
NCM1	Module	NCOL-1.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NLM1	Module	NLAY-1.
NRM1	Module	NROW-1.

List of Variables for Module SOBKIU (Continued)

Variable	Range	Definition
NROW	Global	Number of rows in the grid.
QX	Package	DIMENSION(NCOL,NROW,NLAY), Volume fluid flux across the right cell face in the X direction
QX1	Module	Volume fluid flux at lowest active column with variable head.
QX2	Module	Volume fluid flux at highest active column with variable head.
QY	Package	DIMENSION(NCOL,NROW,NLAY), Volume fluid flux across the front cell face in the Y direction
QY1	Module	Volume fluid flux at lowest active row with variable head.
QY2	Module	Volume fluid flux at highest active row with variable head.
QZ	Package	DIMENSION(NCOL,NROW,NLAY), Volume fluid flux across the bottom cell face in the Z direction.
QZ1	Module	Volume fluid flux at lowest active layer with variable head.
QZ2	Module	Volume fluid flux at highest active layer with variable head.
SPX	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge or ultimate bulk flux in the X direction.
SPY	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge or ultimate bulk flux in the Y direction.
SPZ	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge or ultimate bulk flux in the Z direction.
T1	Module	Temporary variable for transmissivity at I or J location
T2	Module	Temporary variable for transmissivity at I+1 or J+1 location.
THCK	Module	Thickness of cell.
TMP	Module	Temporary variable for HD.
XAREA	Module	Cross-sectional area of cell perpendicular to the X direction.
YAREA	Module	Cross-sectional area of cell perpendicular to the Y direction.
ZAREA	Module	Cross-sectional area of cell perpendicular to the Z direction.

Displacement Package Input

Input for Displacement Package (USL) is read from the unit specified in IUNIT(13).

FOR EACH SIMULATION

USL1AL

1. Data:	IUSLOC	KUNIT		
Format:	I10	I10		
2. Data:	ISTEP	XCLOSE	IOSTP	TCLOSE
Format:	I10	F10.0	I10	F10.0

USL1RP

Data arrays (items 3 and 4) are read for each layer.

3. Data:	SSE
Module:	U2DREL
4. Data:	SSV
Module:	U2DREL

Data array (item 5) is read for LAYCON type zero or two. Item 5 is read after all of items 3 and 4 have been read

5. Data:	TRAN
Module:	U2DREL

If IUSLOC>0 then read item 6

6. Data:	NMAGFM	NUSXFM	NUSYFM	NUSZFM	NVSTFM
	NMAGUN	NUSXUN	NUSYUN	NUSZUN	NVSTUN
Format:	10I5				

FOR EACH TIME STEP

USL1OT

7. Data:	NMAGPR	NUSXPR	NUSYPR	NUSZPR	NVSTPR
	NMAGSV	NUSXSV	NUSYSV	NUSZSV	NVSTSV
Format:	10I5				

Explanation of Fields Used in Input Instructions

IUSLOC–is the output control flag for displacement

If IUSLOC>0 displacement values will be written or saved according to
flags specified in item 6.
If IUSLOC<=0 displacement values will not be written or saved.

KUNIT–is the length unit used in the simulation

If KUNIT=0 meters are used
If KUNIT ≠ 0 feet are used.

ISTEP–is the maximum number of times through the inner iteration loop in one time step in an attempt to solve the system of finite-difference equations. Two hundred iterations is generally sufficient.

XCLOSE–is the individual directional component of displacement change criterion for convergence. When the maximum absolute value of residual from all nodes during an iteration is less than or equal to XCLOSE, iteration stops.

IOSTP–is the maximum number of times through the outer iteration loop in one time step in an attempt to solve the system of finite-difference equations. Fifty iterations is generally sufficient

TCLOSE–is the total directional component of displacement change criterion for convergence. When the maximum absolute value of cumulative displacement change from all nodes during an iteration is less than or equal to TCLOSE, iteration stops.

SSE–is the elastic specific storage value in the vertical direction

SSV–is the inelastic or virgin specific storage value in the vertical direction.

TRAN–is the transmissivity specified in the BCF package for laycon types of zero or two. It is repeated here because the TRAN read by the BCF package is immediately changed to a harmonic mean so these values are never passed to another subroutine.

NMAGFM–is the output format code for magnitude of displacement.

NUSXFM–is the output format code for X-direction displacement.

NUSYFM–is the output format code for Y-direction displacement.

NUSZFM–is the output format code for Z-direction displacement.

NVSTFM–is the output format code for volume strain.

NMAGUN–is the unit number for saving magnitude of displacement

NUSXUN–is the unit number for saving X-direction displacement

NUSYUN–is the unit number for saving Y-direction displacement

NUSZUN–is the unit number for saving Z-direction displacement

NVSTUN–is the unit number for saving volume strain.

NMAGPR–is the print flag for magnitude of displacement of solids.

If NMAGPR<=0 magnitude of displacement is not printed
If NMAGPR>0 magnitude of displacement is printed.

NUSXFM–is the print flag for X-direction displacement

If NUSXPR<=0 X-direction displacement is not printed
If NUSXPR>0 X-direction displacement is printed

NUSYFM–is the print flag for Y-direction displacement

If NUSXPR<=0 Y-direction displacement is not printed
If NUSXPR>0 Y-direction displacement is printed

NUSZFM–is the print flag for Z-direction displacement.

If NUSXPR<=0 Z-direction displacement is not printed

If NUSXPR>0 Z-direction displacement is printed

NVSTFM–is the print flag for volume strain
If NUSXPR<=0 volume strain is not printed
If NUSXPR>0 volume strain is printed.

NMAGSV–is the save flag for magnitude of displacement
If NMAGSV<=0 magnitude of displacement is not saved.
If NMAGSV>0 magnitude of displacement is saved.

NUSXSV–is the save flag for magnitude of displacement
If NUSXSV<=0 X-direction displacement is not saved.
If NUSXSV>0 X-direction displacement is saved.

NUSYSV–is the save flag for magnitude of displacement
If NUSYSV<=0 Y-direction displacement is not saved.
If NUSYSV>0 Y-direction displacement is saved.

NUSZSV–is the save flag for magnitude of displacement
If NUSZSV<=0 Z-direction displacement is not saved.
If NUSZSV>0 Z-direction displacement is saved.

NVSTSV–is the save flag for magnitude of displacement
If NVSTSV<=0 volume strain is not saved.
If NVSTSV>0 volume strain is saved.

Module Documentation for the Displacement Package

The displacement package (USL1) has four primary modules, three submodules, and two utility modules. All the primary modules are called by the MAIN program.

Primary Modules

USL1AL	Allocates space for data arrays. Reads output control flags, iteration and convergence information, and length units flag.
USL1RP	Initializes displacements, strains, and print flags. Reads specific storage values, transmissivity if needed, and reads format information for printing or saving displacements and volume strains.
USL1FM	Formulates and solves numerical approximations to governing equations describing the displacement of solids using a dual loop successive overrelaxation technique with Chebyshev acceleration. Applies all boundary conditions, and calculates volume strains.
USL1OT	Reads print and save flags for displacement and volume strain. Prints or saves displacement values in each component direction, magnitude of displacement, or volume strains after each stress period when the print or save flags are set.

Submodules

SUSL1X	Calculates strain in the X direction.
SUSL1Y	Calculates strain in the Y direction.
SUSL1Z	Calculates strain in the Z direction.

Utility Modules

UBCUSL	Calculates magnitude of displacement if print or save flag set.
U2USLR	Reads transmissivity information if LAYCON=0 or 2.

Narrative for Module USL1AL

This module allocates space for data arrays for the displacement package. It also reads the output control flag, iteration and convergence information, and length units flag. This module performs its tasks in the following order:

1. Identify package
2. Read flags for calculation and printing or saving displacements, and length units.
3. Print statements for output control, length units, and space allocation
4. Calculate number of cells and direction with maximum number of cells.
5. Allocate storage for the following arrays:
    USLX Displacement in the X direction for each cell in the grid.
    USLY Displacement in the Y direction for each cell in the grid.
    USLZ Displacement in the Z direction for each cell in the grid.
    TEMPX Temporary storage for displacement in X direction
    TEMPY Temporary storage for displacement in Y direction.
    TEMPZ Temporary storage for displacement in Z direction.
    UOLDX Old value of displacement in the X direction.
    UOLDY Old value of displacement in the Y direction.
    UOLDZ Old value of displacement in the Z direction.
    HC Hydraulic conductivity of each cell in the grid.
    STRNX Strain in the X direction for each cell in the grid.
    STRNY Strain in the Y direction for each cell in the grid.
    STRNZ Strain in the Z direction for each cell in the grid.
    VSTRN Volume strain for each cell in the grid.
    PS Preconsolidation strain for each cell in the grid.
    SSE Elastic specific storage for each cell in the grid.
    SSV Virgin specific storage for each cell in the grid.
    TRAN Transmissivity for cells where LAYCON=0 or 2.
    SSK Horizontal specific storage for each cell in the grid.
    MAG Magnitude of displacement for each cell in the grid.
    UX single precision displacement in X direction for plotting.
    UY single precision displacement in Y direction for plotting.
    UZ single precision displacement in Z direction for plotting.
6. Print amount of storage used by the displacement package.
7. RETURN

Flow Chart for Module USL1AL

IUSLOC is the flag indicating whether displacement is calculated and printed or saved.
IUSLOC>1 Displacements are calculated and printed or saved.
IUSLOC<=0 Displacements are not calculated, printed, or saved.

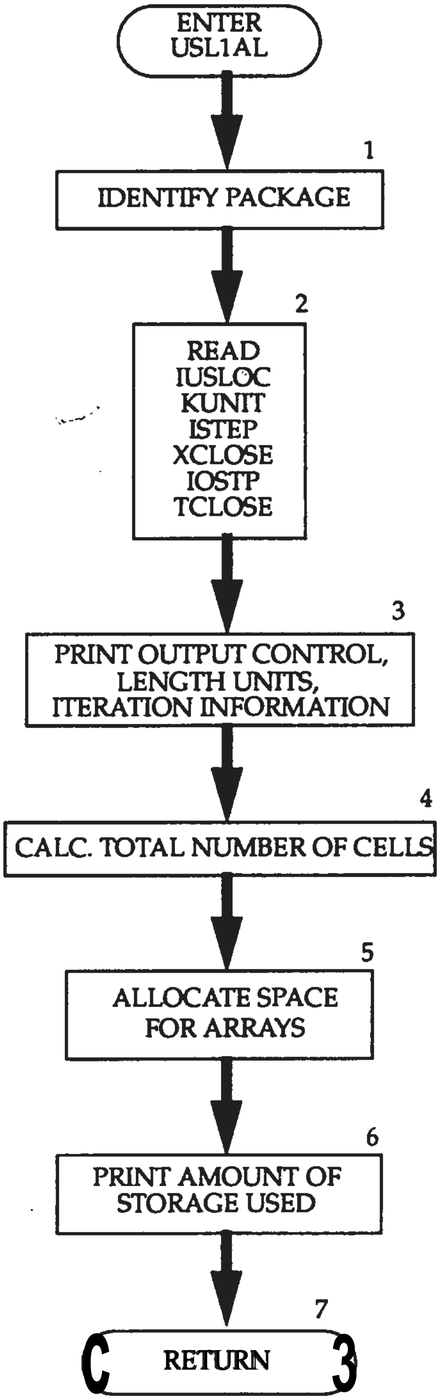
KUNIT is the length units flag.
KUNIT=0 Meters are the units for length
KUNIT=1 Feet are the units for length

ISTEP is the number of inner-loop iterations used to obtain a solution.

XCLOSE is the inner-loop closure criterion. This value should be at least one order of magnitude smaller than the smallest displacement value calculated.

IOSTP is the number of outer-loop iterations used to obtain a solution

TCLOSE is the outer-loop closure criterion. This value should small enough so that at least 5 outer iterations are performed before closure.



```
      SUBROUTINE USL1AL(ISUM,LENX,LCUSLX,LCUSLY,LCUSLZ,LCSTRNZ,LCPS,
    1  NCOL,NROW,NLAY,NMAX,LCUMAG,IN,IOUT,IUSLOC,KUNIT,LCSSE,LCSSV,
    2  LCSTRNX,LCSTRNY,LCVSTRN,LCTRAN,LCHC,LCSSK,LCUOLDX,LCUOLDY,
    3  LCUOLDZ,ISTEP,XCLOSE,IOSTP,LCTEMPX,LCTEMPY,LCTEMPZ,
    4  TCLOSE,LCUX,LCUY,LCUZ)
C
C  ********************************************************************
C  ALLOCATE ARRAY STORAGE FOR DISPLACEMENT PACKAGE
C  ********************************************************************
C
C  SPECIFICATIONS:
C  --------------------------------------------------------------------
C  --------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE
      WRITE(IOUT,1)IN
      FORMAT(1H0/USL1 -- DISR PACKAGE CALCULATES HORIZONTAL AND
    1 VERTICAL AND HORIZONTAL DISPLACEMENTS AND VELOCITIES OF
    2  SOLIDS FROM UNIT NUMBER,I4)
C
C2------READ OUTPUT CONTROL FOR DISPLACEMENT, AND UNIT OF
C        MEASUREMENT AS WELL AS NUMBER OF ITERATIONS AND CLOSURE
C        CRITERION
      READ(IN,5) IUSLOC,KUNIT
    5  FORMAT(2I10)
      READ(IN,9) ISTEP,XCLOSE,IOSTP,TCLOSE
    9  FORMAT(I10,F10.0,I10,F10.0)
C3------PRINT STATEMENTS FOR OUTPUT CONTROL, UNITS, ITERATION INFO.
      IF(IUSLOC.GT.0) WRITE(IOUT,15)
   15  FORMAT(1X,'OUTPUT CONTROL RECORDS WILL BE READ EACH TIME
     1 STEP')
      IF(IUSLOC.LE.0) WRITE(IOUT,17)
   17 FORMAT(1X,'DISPLACEMENT INFORMATION WILL NOT BE WRITTEN')
      IF(KUNIT.EQ.0) WRITE(IOUT,19)
      IF(KUNIT.NE.0) WRITE(IOUT,21)
   19 FORMAT(//,1X,'METERS WILL BE USED AS THE SPACE DIMENSION')
```

```
 21  FORMAT(/,1X,'FEET WILL BE USED AS THE SPACE DIMENSION')
     WRITE(IOUT,7) ISTEP,IOSTP
      IF(NLAY.LT.2) WRITE(IOUT,80)
 80  FORMAT(1X,'NLAY MUST BE AT LEAST 2.  USE IBS PACKAGE FOR FEWER
    1  LAYERS THAN 2.'/,1X,'ONLY HORIZONTAL DISPLACEMENT WILL BE
    2  SIMULATED')
  7  FORMAT(1X,'THE MAXIMUM NUMBER OF INNER ITERATIONS FOR
    1  CLOSURE IS', I5,/,1X,'THE NUMBER OF OUTER ITERATIONS IS',I5,)
     WRITE(IOUT,8) XCLOSE,TCLOSE
  8   FORMAT(1X,'THE CLOSURE FOR INNER DISPLACEMENT IS',E15.8,/,
     1 1X,'THE CLOSURE FOR OUTER DISPLACEMENT IS',E15.8)
C
C4---CALCULATE TOTAL NUMBER OF CELLS AND DIRECTION WITH MAX
C    CELLS
     NRCL=NROW*NCOL*NLAY
     NRC=NROW*NCOL
     NMAX=MAX0(NCOL,NROW,NLAY)
C
C5------ALLOCATE SPACE FOR STORAGE
     IUSL=ISUM
     LCUSLX=ISUM
     ISUM=ISUM+NRCL*2
     LCUSLY=ISUM
     ISUM=ISUM+NRCL*2
     LCUSLZ=ISUM
     ISUM=ISUM+NRCL*2
     LCTEMPX=ISUM
     ISUM=ISUM+NRCL*2
     LCTEMPY=ISUM
     ISUM=ISUM+NRCL*2
     LCTEMPZ=ISUM
     ISUM=ISUM+NRCL*2
     LCUOLDX=ISUM
     ISUM=ISUM+NRCL*2
     LCUOLDY=ISUM
     ISUM=ISUM+NRCL*2
     LCUOLDZ=ISUM
```

```
      ISUM=ISUM+NRCL*2
      LCHC=ISUM
      ISUM=ISUM+NRCL
      LCSTRNZ=ISUM
      ISUM=ISUM+NRCL
      LCSTRNX=ISUM
      ISUM=ISUM+NRCL
      LCSTRNY=ISUM
      ISUM=ISUM+NRCL
      LCVSTRN=ISUM
      ISUM=ISUM+NRCL
      LCPS=ISUM
      ISUM=ISUM+NRCL
      LCSSE=ISUM
      ISUM=ISUM+NRCL
      LCSSV=ISUM
      ISUM=ISUM+NRCL
      LCTRAN=ISUM
      ISUM=ISUM+NRCL
      LCSSK=ISUM
      ISUM=ISUM+NRCL
      LCUX=ISUM
      ISUM=ISUM+NRCL
      LCUY=ISUM
      ISUM=ISUM+NRCL
      LCUZ=ISUM
      ISUM=ISUM+NRCL
      IF(IUSLC.LE.0) GOTO 20
      LCUMAG=ISUM
      ISUM=ISUM+NRCL
C
C6--------PRINT NUMBER OF SPACES IN X ARRAY USED BY USL PACKAGE
   20 ISP=ISUM-IUSL
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I8,' ELEMENTS USED IN USL PACKAGE')
      ISUM1=ISUM-1
      WRITE(IOUT,3)ISUM1,LENX
```

```
    3  FORMAT(1X,I8,' ELEMENTS IN X ARRAY USED OUT OF',I8)
C
       IF(ISUM1.GT.LENX) WRITE(IOUX,6)
    6  FORMAT(1X,' ****X ARRAY MUST BE DIMENSIONED LARGER****')
C7------RETURN
       RETURN
       END
```

List of Variables for Module USL1AL

Variable	Range	Definition
IN	Package	Primary unit number from which input for this package will be read.
IOSTP	Package	Maximum number of outer-loop iterations selected for convergence.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
ISP	Module	Number of words in the X array allocated by this module.
ISTEP	Package	Maximum number of inner-loop iterations selected for convergence.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	ISUM-1
IUSL	Module	Before this module allocates space, IUSL is set equal to ISUM. After allocation, IUSL is subtracted from ISUM to get ISP, the amount of space in the X array allocated by this module.
IUSLOC	Package	Flag indicating whether displacement and volume strain information calculated and printed or saved. >0 Displacements and volume strains are calculated and printed or saved according to flags set in the output subroutine. <=0 Displacements and volume strains are not calculated.
KUNIT	Package	Flag indicating whether english or metric units are used for length. =0 Meters are used as units of length. =1 Feet are used as units of length.
LCHC	Package	Location in the X array of the first element of array HC.
LCPS	Package	Location in the X array of the first element of array PS.
LCSSK	Package	Location in the X array of the first element of array SSK.
LCSSE	Package	Location in the X array of the first element of array SSE.
LCSSV	Package	Location in the X array of the first element of array SSV.
LCSTRNX	Package	Location in the X array of the first element of array STRNX.
LCSTRNY	Package	Location in the X array of the first element of array STRNY.
LCSTRNZ	Package	Location in the X array of the first element of array STRNZ.
LCTEMPX	Package	Location in the X array of the first element of array TEMPX.
LCTEMPY	Package	Location in the X array of the first element of array TEMPY.
LCTEMPZ	Package	Location in the X array of the first element of array TEMPZ.
LCTRAN	Package	Location in the X array of the first element of array TRAN.
LCUMAG	Package	Location in the X array of the first element of array UMAG.
LCUOLDX	Package	Location in the X array of the first element of array UOLDX.
LCUOLDY	Package	Location in the X array of the first element of array UOLDY.
LCUOLDZ	Package	Location in the X array of the first element of array UOLDZ.
LCUSLX	Package	Location in the X array of the first element of array USLX.
LCUSLY	Package	Location in the X array of the first element of array USLY.

List of Variables for Module USL1AL (Continued)

Variable	Range	Definition
LCUSLZ	Package	Location in the X array of the first element of array USLZ.
LCUX	Package	Location in the X array of the first element of array UX.
LCUY	Package	Location in the X array of the first element of array UY.
LCUZ	Package	Location in the X array of the first element of array UZ.
LCVSTRN	Package	Location in the X array of the first element of array VSTRN.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NMAX	Package	Number of cells in either the row, column or layer direction—whichever is greatest.
NRC	Module	Number of cells in a layer.
NRCL	Module	Number of cells in the grid.
NROW	Global	Number of rows in the grid.
TCLOSE	Package	Closure criterion for outer-loop convergence.
XCLOSE	Package	Closure criterion for inner-loop convergence.

Narrative for Module USL1RP

This module sets the initial values of displacement and strain and initializes flags for printing and saving displacements and heads. It also reads the elastic and virgin specific storage values for each layer. Transmissivity is also read if LAYCON = 0 or 2. Finally, the format for printing and saving displacements and strains is read if IUSLOC is set.

Module USL1RP calls utility modules U2DREL and U2USLR and performs its tasks in the following order:

1. Initializes displacements and strains to zero.
2. Initializes flags for printing and saving displacements and strains to zero.
3. Reads elastic and virgin specific storage values for each layer.
4. If LAYCON = 0 or 2 read transmissivity values.
5. Read formats and unit numbers for printing or saving displacements and volume strain
6. RETURN

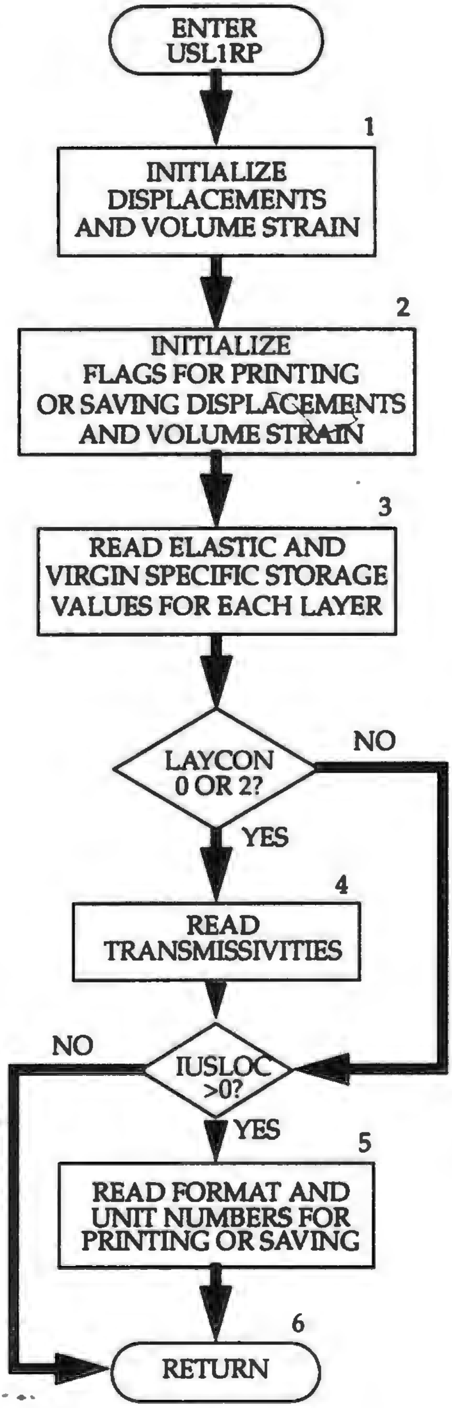
Flow Chart for Module USLIRP

LAYCON is the layer-type code (one for each layer).

- 0 - confined
- 1 - unconfined
- 2 - confined/unconfined but transmissivity is constant
- 3 - confined/unconfined transmissivity varies

IUSLOC is the flag for printing or saving displacement or volume strain information.

- >0 - displacements or volume strains are printed or saved according to specified flags
- <=0 - no displacements or volume strains are calculated written or saved



```
      SUBROUTINE USL1RP(USLX,USLY,USLZ,STRNX,STRNY,STRNZ,PS,SSE,SSV,
     1 NODES,NCOL,NROW,NLAY,IN,IOUT,IUSLOC,NMAGFM,NUSXFM,NUSYFM,
     2 NUSZFM,NMAGUN,NUSXUN,NUSYUN,NUSZUN,NVSTFM,NVSTUN,
     3 VSTRN,TRAN,NMAX)
C
C     ******************************************************************
C     INITIALIZES DISPLACEMENT AND STRAIN ARRAYS BY ASSUMING
C     UNSTRAINED CONDITIONS.  ALSO READS IN TRANSMISSIVITY VALUES IF
C     NEEDED
C     ******************************************************************
C     SPECIFICATIONS:
C-----------------------------------------------------------------------
      CHARACTER*4 ANAME
      DOUBLE PRECISION USLX,USLY,USLZ
C
      DIMENSION USLX(NODES),USLY(NODES),USLZ(NODES),SSV(NODES),
     1 SSE(NODES),ANAME(6,2),STRNX(NODES),STRNY(NODES),STRNZ(NODES),
     2 PS(NODES),VSTRN(NODES),TRAN(NODES)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1  ANAME(6,1) /'ELAS','TIC ','SPEC','IFIC',' STO','RAGE'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1  ANAME(6,2) /' INE','LAST','IC S','PEC.',' STO','RAGE'/
C
      COMMON /FLWCOM/LAYCON(80)
C-----------------------------------------------------------------------
C
C1------SET ARRAYS TO ZERO INITIAL DISPLACEMENT, VELOCITY OF SOLIDS,
C        STRAIN AND STRAIN RATE.  ALSO ASSUME THAT
C        PRECONSOLIDATION STRAIN IS ZERO.
      DO 10 K=1,NODES
      USLX(K)=0.D0
      USLY(K)=0.D0
      USLZ(K)=0.D0
      STRNX(K)=0.
      STRNY(K)=0.
      STRNZ(K)=0.
```

```
      VSTRN(K)=0.
      PS(K)=0.
   10 CONTINUE
C
C2------INITIALIZE FLAGS FOR PRINTING AND SAVING DISPLACEMENTS AND
C       STRAINS
      NMAGFM=0
      NUSXFM=0
      NUSYFM=0
      NUSZFM=0
      NVSTFM=0
      NMAGUN=0
      NUSXUN=0
      NUSYUN=0
      NUSZUN=0
      NVSTUN=0
C
C3------READ IN SPECIFIC STORAGE VALUES
      NCR=NCOL*NROW
      DO 100 K=1,NLAY
      LOC=1+(K-1)*NCR
      CALL U2DREL(SSE(LOC),ANAME(1,1),NROW,NCOL,K,IN,IOUT)
      CALL U2DREL(SSV(LOC),ANAME(1,2),NROW,NCOL,K,IN,IOUT)
  100 CONTINUE
C
C------TEST TO SEE IF TRANSMISSIVITY NEEDS TO BE READ. IF IT DOES, READ IT
      ITEST=0
      DO 33 K=1,NLAY
      IF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) ITEST=1
   33 CONTINUE
      IF(ITEST.EQ.0) GOTO 200
C
C4------READ TRANSMISSIVITY DATA IF LAYCON=0 OR 2
      KR=0
      DO 72 K=1,NLAY
      IF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) KR=KR+1
      LOCR=1+(KR-1)*NCR
```

```
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1)GOTO 72
      CALL U2USLR(TRAN(LOCR),NROW,NCOL,K,IN)
   72 CONTINUE
C
C5-----READ FORMAT AND UNIT NUMBER INFORMATION FOR SUBSIDENCE,
C          DISPLACEMENT, AND MAGNITUDE OF DISPLACEMENT IF IUSLOC IS
C          GREATER THAN ZERO.
  200 IF(IUSLOC.LE.0) GOTO 300
      READ(IN,25)  NMAGFM,NUSXFM,NUSYFM,NUSZFM,NVSTFM,NMAGUN,
     1 NUSXUN,NUSYUN,NUSZUN,NVSTUN
   25 FORMAT(10I5)
      WRITE(IOUT,30) NMAGFM,NUSXFM,NUSYFM,NUSZFM,NVSTFM
   30 FORMAT(/,'MAGNITUDE OF DISP. PRINT FORMAT IS NUMBER',I4/
     1       '  X-DISPLACEMENT PRINT FORMAT IS NUMBER',I4/
     2       '  Y-DISPLACEMENT PRINT FORMAT IS NUMBER',I4/
     3       '  Z-DISPLACEMENT PRINT FORMAT IS NUMBER',I4/
     4       '   VOLUME STRAIN PRINT FORMAT IS NUMBER',I4)
      IF(NMAGUN.GT.0) WRITE(IOUT,40) NMAGUN
   40 FORMAT(/,1X,'UNIT FOR SAVING MAGNITUDE OF DISPLACEMENT IS',I4)
      IF(NUSXUN.GT.0) WRITE(IOUT,45) NUSXUN
   45 FORMAT(/,1X,' UNIT FOR SAVING X-DIRECTION DISPLACEMENT IS',I4)
      IF(NUSYUN.GT.0) WRITE(IOUT,50) NUSYUN
   50 FORMAT(/,1X,' UNIT FOR SAVING Y-DIRECTION DISPLACEMENT IS',I4)
      IF(NUSZUN.GT.0) WRITE(IOUT,55) NUSZUN
   55 FORMAT(/,1X,' UNIT FOR SAVING Z-DIRECTION DISPLACEMENT IS',I4)
      IF(NVSTUN.GT.0) WRITE(IOUT,60) NVSTUN
   60 FORMAT(/,1X,'          UNIT FOR SAVING VOLUME STRAIN IS',I4)
C
C6-------RETURN
  300 RETURN
      END
```

List of Variables for Module USL1RP

Variable	Range	Definition
ANAME	Module	Label for printout of input array.
I	Module	Index for rows
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT ≡ 6.
ITEST	Module	Flag for indicating whether transmissivities are read or not. ≡0 Transmissivities are not read ≡1 Transmissivities are read
IUSLOC	Package	Flag for calculating, printing or saving displacements and volume strains.
J	Module	Index for columns.
K	Module	Index for layers
KR	Module	Counter for number of layers for which TRAN is read (LAYCON ≡ 0 or 2).
LAYCON	Global	DIMENSION (80), Layer-type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity is variable).
LOC	Module	Pointer to parts of the SSE and SSV arrays corresponding to particular layers.
LOCR	Module	Pointer to parts of the TRAN array corresponding to particular layers.
NCOL	Global	Number of columns in the grid.
NCR	Module	Number of cells in a layer.
NLAY	Global	Number of layers in the grid.
NMAGFM	Package	Code for format in which magnitude of displacement should be printed.
NMAGUN	Package	Unit number on which an unformated record containing magnitude of displacement should be recorded.
NODES	Module	Number of cells in the grid
NROW	Global	Number of rows in the grid.
NUSXFM	Package	Code for format in which X-displacements should be printed.
NUSXUN	Package	Unit number on which an unformated record containing X-displacements should be recorded.
NUSYFM	Package	Code for format in which Y-displacements should be printed.
NUSYUN	Package	Unit number on which an unformated record containing Y-displacements should be recorded.
NUSZFM	Package	Code for format in which Z-displacements should be printed.
NUSZUN	Package	Unit number on which an unformated record containing Z-displacement should be recorded.
NVSTFM	Package	Code for format in which volume strains should be printed.
NVSTUN	Package	Unit number on which an unformated record containing volume strains should be recorded.

List of Variables for Module USLIRP (continued)

Variable	Range	Definition
PS	Package	DIMENSION (NCOL,NROW,NLAY), Preconsolidation volume strain used to determine whether elastic or virgin specific storage should be used.
SSE	Package	DIMENSION (NCOL,NROW,NLAY), Elastic specific storage.
SSV	Package	DIMENSION (NCOL,NROW,NLAY), Virgin specific storage.
STRNX	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the X direction.
STRNY	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the Y direction.
STRNZ	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the Z direction.
USLX	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the X direction.
USLY	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the X^* direction.
USLZ	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the Z direction.
VSTRN	Package	DIMENSION (NCOL,NROW,NLAY), Volume strain.

Narrative for Module USL1FM

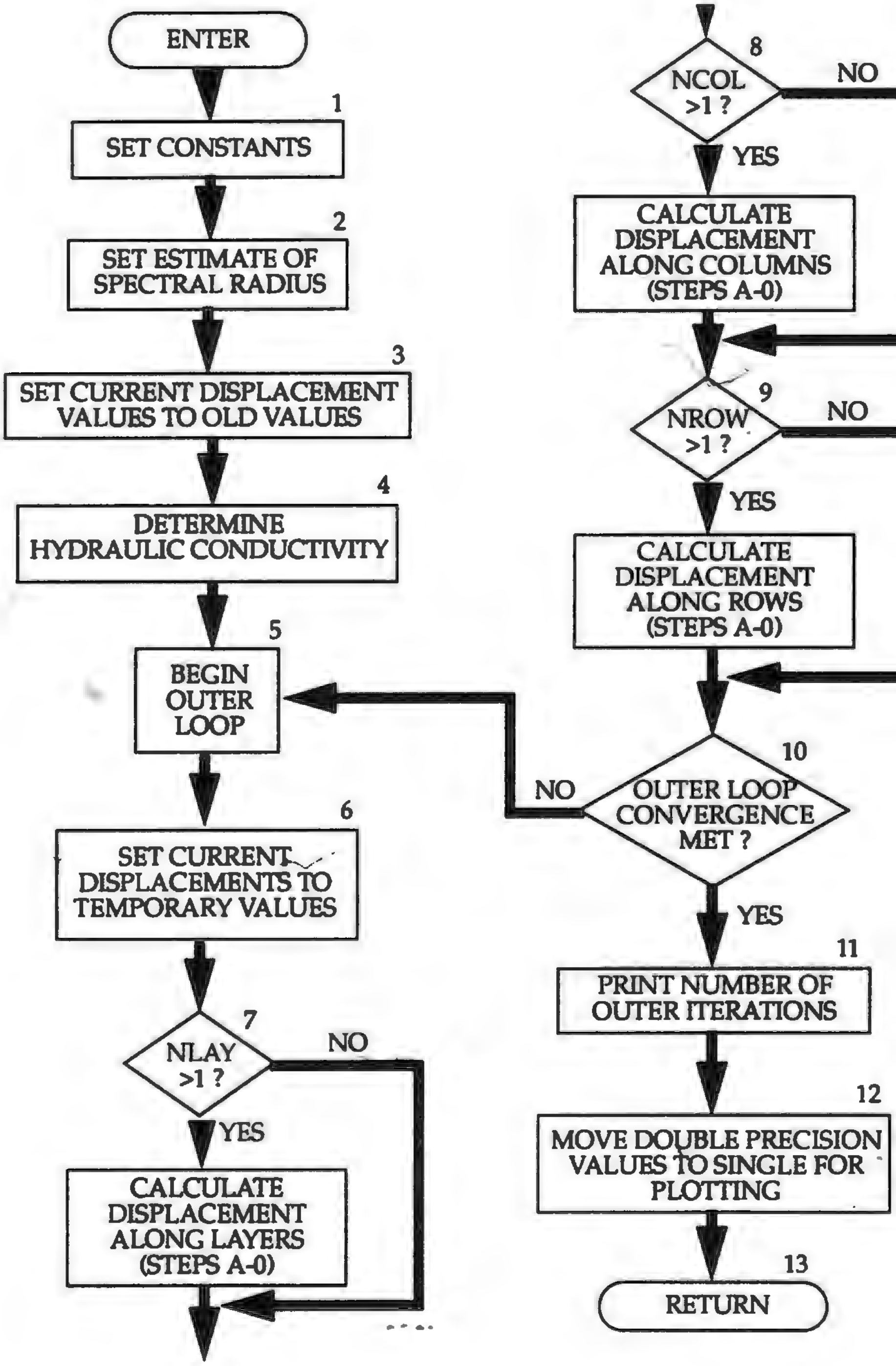
This module represents the major part of the granular displacement model. This module assembles the numerical approximations to the governing equations and calculates displacements and volume strains for each active cell in the model grid. In addition, this module contains the solution algorithm using a dual-loop successive overrelaxation iterative scheme with Chebyshev acceleration to solve the system of equations. This module performs other tasks such as calculation of hydraulic conductivity, and determines the correct value of specific storage for a given cell on the basis of the past maximum volume strain.

This module is called by the MAIN program and calls the submodules SUSL1X, SUSL1Y and SUSL1Z. Module USL1FM performs it's tasks in the following order:

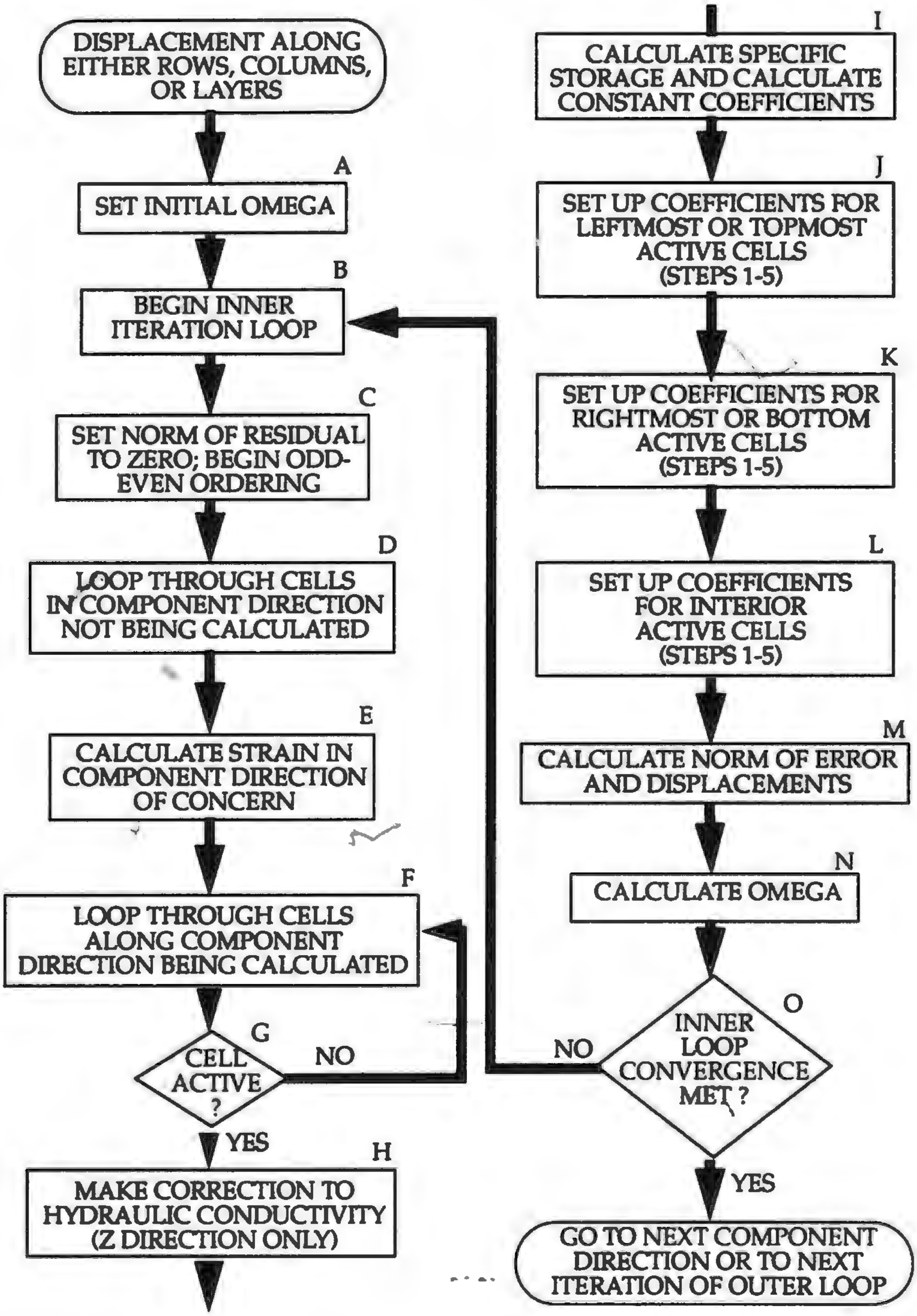
1. Set constants.
2. Set estimate of spectral radius.
3. Set current displacement equal to old values.
4. Determine hydraulic conductivity for each cell.
5. Begin outer iteration loop.
6. Set current displacement values to temporary values for error check.
7. Check if displacement for layers is necessary
    If displacement is calculated perform tasks A-O (below).
8. Check if displacement for columns is necessary.
    If displacement is calculated perform tasks A-O (below).
9. Check if displacement for rows is necessary.
    If displacement is calculated perform tasks A-O (below).
10. Check if outer loop convergence is met. If not return to step 5.
11. Print number of outer iterations necessary for convergence.
12. Move double precision displacements to single precision storage for plotting.
13. RETURN

A. Set initial omega.
B. Begin inner iteration loop.
C. Set norm of residual to zero, use odd-even ordering.
D. Loop through cells in component directions not being calculated.
E. Calculate strain in component direction of concern.
F. Loop through cells along component direction being evaluated.
G. Check for non-active cells.
H. Make correction to hydraulic conductivity (Z-direction only).
I. Calculate correct value of specific storage.
J. Set up coefficients for left-most or topmost active cells.
K. Set up coefficients for right-most or bottommost active cells.
L. Set up coefficients for interior cells.
M. Calculate norm of true error and new displacements.
N. Calculate omega.
O. Check if inner-loop convergence met. If not, go to step B.
GO TO NEXT COMPONENT DIRECTION OR NEXT OUTER LOOP CHECK.

Flow Chart for Module USL1FM (first of 3 pages)

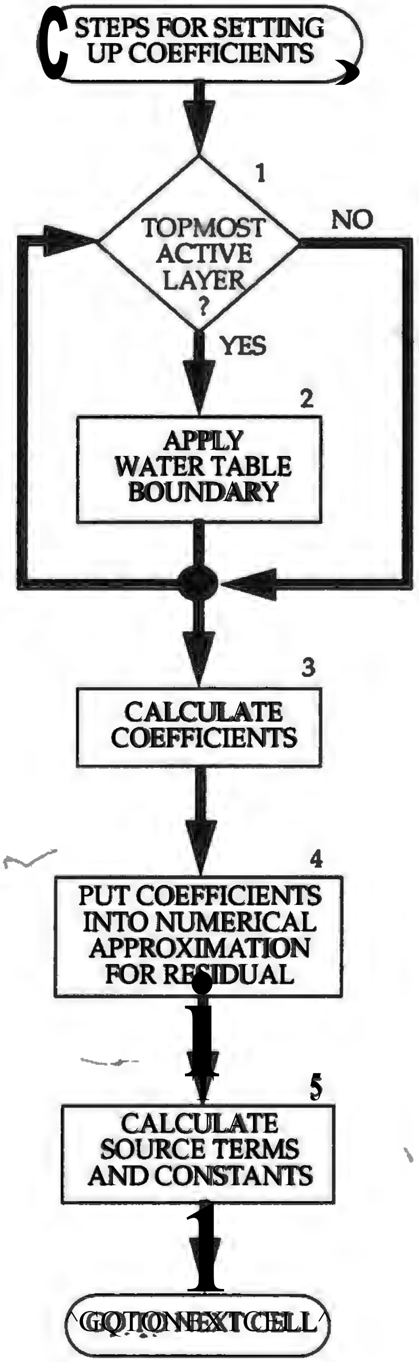


Flow Chart for Module USLIFM (second of 3 pages)



Flow Chart for Module USL1FM (third of 3 pages)

Steps 1 and 2 apply to the Z direction only.



```
      SUBROUTINE USL1FM(QBX,QBY,QBZ,IACT,DELR,DELC,DELL,NCOL,NROW,
     1  NLAY,SC1,HY,USLX,USLY,USLZ,STRNZ,PS,IOUT,IUSLOC,DELT,
     2  TOTIM,RATIO,KSTP,SSE,SSV,IUNIT,STRNX,STRNY,VSTRN,
     3  TRAN,SC2,HC,SSK,UOLDX,UOLDY,UOLDZ,ISTEP,XCLOSE,ICSTP,KPER,
     4  TEMPX,TEMPY,TEMPZ,TCLOSE,UX,UY,UZ,HNEW,BOT,IBOUND,HOLD,
     5  SPX, SPY,SPZ,NBULK)
C
C   ***********************************************************************************
C    THIS SUBROUTINE CALCULATES THE DISPLACEMENT OF SOLIDS
C    THROUGH TIME FOR EACH COMPONENT DIRECTION USING A
C    DUAL ITERATATIVE SOR SOLVER WITH A CRANK-NICOLSON
C   APPROXIMATION.  CHEBYSHEV ACCELERATION IS ALSO USED TO SPEED
C   CONVERGENCE. THIS ALGORITHM ASSUMES FIXED BOUNDARY
C   CONDITIONS AND USES A WATER TABLE BOUNDARY.
C   ***********************************************************************************
C
C    SPECIFICATIONS:
C-----------------------------------------------------------------------------------
C
      DOUBLE PRECISION USLX,USLY,USLZ,ANORM,ANORMF,UOLDX,UOLDY,
     1UOLDZ,OMEGA,RESID,COEF,TDIFF,ERR,TEMPX,TEMPY,TEMPZ
     2 XCON,YCON,ZCON,HNEW
C
      DIMENSION IACT(NCOL,NROW,NLAY),DELR(NCOL),HY(NCOL,NROW,
     1NLAY),USLX(NCOL,NROW,NLAY),QBX(NCOL,NROW,NLAY),USLY(NCOL,
     2NROW,NLAY),QBY(NCOL,NROW,NLAY),USLZ(NCOL,NROW,NLAY),QBZ
     3 (NCOL,NROW,NLAY),DELC(NROW),SC1(NCOL,NROW,NLAY),DELL(NCOL,
     4NROW,NLAY),STRNZ(NCOL,NROW,NLAY),PS(NCOL,NROW,NLAY),
     6TEMPX(NCOL,NROW,NLAY),SSE(NCOL,NROW,NLAY),SSV(NCOL,
     7NROW,NLAY),RATIO(NCOL,NROW,NLAY),STRNY(NCOL,NROW,NLAY),
     8STRNX(NCOL,NROW,NLAY),VSTRN(NCOL,NROW,NLAY),TEMPY(NCOL,
     9 NROW,NLAY),IBOUND(NCOL,NROW,NLAY),TRAN(NCOL,
     *NROW,NLAY),SC2(NCOL,NROW,NLAY),TEMPZ(NCOL,NROW,NLAY),
     1HC(NCOL,NROW,NLAY),SSK(NCOL,NROW,NLAY),UOLDX(NCOL,
     2NROW,NLAY),UOLDY(NCOL,NROW,NLAY),UOLDZ(NCOL,NROW,NLAY),
     3UX(NCOL,NROW,NLAY),UY(NCOL,NROW,NLAY),UZ(NCOL,NROW,NLAY),
     4HNEW(NCOL,NROW,NLAY),BOT(NCOL,NROW,NLAY),HOLD(NCOL,
```

```
     5 NROW,NLAY), SPX(NCOL,NROW,NLAY),SPY(NCOL,NROW,NLAY),
     6 SPZ(NCOL,NROW,NLAY),NBULK
C
    COMMON /FLWCOM/LAYCON(80)
C-----------------------------------------------------------------------
C
C1------SET CONSTANTS
      TDOLD=1.E8
C
C2------SET INITIAL SPECTRAL RADIUS
      RJACX=0.998
      RJACY=0.998
      RJACZ=0.998
C
C3------SET CURRENT DISPLACEMENT VALUES EQUAL TO OLD VALUES
      DO 68 I=1,NROW
      DO 68 J=1,NCOL
      DO 68 K=1,NLAY
      UOLDX(J,I,K)=USLX(J,I,K)
      UOLDY(J,I,K)=USLY(J,I,K)
      UOLDZ(J,I,K)=USLZ(J,I,K)
   68 CONTINUE
C
      IF(NBULK.EQ.1)THEN
      DO 61 I=1,NROW
      DO 61 J=1,NCOL
      DO 61 K=1,NLAY
      QBX(J,I,K)=SPX(J,I,K)
      QBY(J,I,K)=SPY(J,I,K)
      QBZ(J,I,K)=SPZ(J,I,K)
   61 CONTINUE
      ENDIF
C4------DETERMINE HYDRAULIC CONDUCTIVITY IN LAYERS WHERE LAYCON IS
C     EITHER 0 OR 2, SO THAT K VALUES ARE KNOWN FOR EVERY LAYER. SET
C     KB TO THE PROPER LAYER. MAKE CORRECTION TO DELL IF UNCONFINED.
      DO 2 I=1,NROW
      DO 2 J=1,NCOL
```

```
      KB=0
      KR=0
      DO 4 K=1,NLAY
      IF(LAYCON(K).EQ.0 .OR. LAYCON(K).EQ.2) KR=KR+1
      IF(LAYCON(K).NE.1 .AND. LAYCON(K).NE. 3) GOTO 3
      KB=KB+1
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) THEN
      DELL(J,I,K)=HNEW(J,I,K)-BOT(J,I,KB)
      IF(DELL(J,I,K).LT.0) DELL(J,I,K)=0.
      ENDIF
      HC(J,I,K)=HY(J,I,KB)
      GOTO 4
    3 HC(J,I,K)=TRAN(J,I,KR)/DELL(J,I,K)
    4 CONTINUE
    2 CONTINUE
C
C5-----BEGIN OUTER ITERATION LOOP FOR ALL THREE DIMENSIONS
      NUMOUT=0
  100 TDIFF=0.D0
      NUMOUT=NUMOUT+1
C
C6-----SET TEMP VALUES FOR OUTER ITERATION CONVERGENCE TEST
      DO 80 K=1,NLAY
      DO 80 I=1,NROW
      DO 80 J=1,NCOL
      TEMPX(J,I,K)=USLX(J,I,K)
      TEMPY(J,I,K)=USLY(J,I,K)
      TEMPZ(J,I,K)=USLZ(J,I,K)
   80 CONTINUE
C
C7-----CHECK IF LAYER DIRECTION ITERATION NECESSARY
      IF(NLAY.LT.2) GOTO 1000
C
C7A-----SET INITIAL OMEGA
      OMEGA=1.D0
C
C7B-----ITERATE TO SOVE FOR DISPLACEMENT IN Z DIRECTION
```

```
      NUMIT=0
   16 DIFFZ=0.
      ANORMF=0.D0
      NUMIT=NUMIT+1
C
C7C----SET NORM FOR RESIDUAL TO ZERO AND BEGIN ODD-EVEN ORDERING
      ANORM=0.D0
      IJSW=1
      DO 91 IPASS=1,2
      KSW=IJSW
C
C7D----LOOP THROUGH ROWS AND COLUMNS
      DO 90 I=1,NROW
      DO 90 J=1,NCOL
      KB=0
      ZADJ=0.
C7E----CALCULATE STRAIN IN THE Z DIRECTION, UPDATE EACH INNER LOOP
C       ITERATION
      CALL SUSLIZ(USLZ,IACT,J,I,NLAY,DELL,STRNZ,IBOUND,NCOL,NROW)
C
C7F----INNER LOOP FOR LAYERS
      DO 92 K=KSW,NLAY,2
C
C7G----CHECK FOR NON-ACTIVE CELLS AND BOUNDARY
      IF(K.EQ.1 .AND. IACT(J,I,K+1).EQ.0) GOTO 92
      IF(K.GT.1 .AND. (IBOUND(J,I,K-1).EQ.0 .AND. IACT(J,I,K+1).EQ.0))
    1 GOTO 92
      IF(K+1.GT.NLAY .AND. IBOUND(J,I,K-1).EQ.0) GOTO 92
      IF(IACT(J,I,K).EQ.0 .AND. IBOUND(J,I,K).EQ.0) GOTO 92
      IF(ZADJ.EQ.0) ZADJ=USLZ(J,I,K)
C
C7H----MAKE CORRECTION FOR VERTICAL HYDRAULIC CONDUCTIVITY
      HV=HC(J,I,K)*RATIO(J,I,K)
C
C7I----DETERMINE WHETHER SPECIFIC YIELD, OR ELASTIC OR INELASTIC
C    SPECIFIC  STORAGE VALUES ARE TO BE USED
      VSTRN(J,I,K)=STRNX(J,I,K)+STRNY(J,I,K)+STRNZ(J,I,K)
```

```
      IF(VSTRN(J,I,K).LT.PS(J,I,K)) THEN
      SSZ=SSV(J,I,K)
      PS(J,I,K)=VSTRN(J,I,K)
      ELSE
      SSZ=SSE(J,I,K)
      ENDIF
      SSK(J,I,K)=SSZ
      IF(LAYCON(K).EQ.1) SS=SSE(J,I,K)
      IF(LAYCON(K).NE.1) SS=SC1(J,I,K)/(DELL(J,I,K)*DELC(I)*DELR(J))
      IF(KUNIT.EQ.0 .AND. SS.LT.4.32E-6) SS=4.32E-6
      IF(KUNIT.NE.0 .AND. SS.LT.1.36E-6) SS=1.36E-6
C
C   SET CONSTANT COEFFICIENTS
      FAC=HV*DELT/(CON*SSK(J,I,K))
      FACH=HV*DELT/(CON*SS)
C
C7J1-----CHECK FOR TOPMOST ACTIVE CELL ALONG A LAYER
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) THEN
C
C7J2-----APPLY WATER TABLE BOUNDARY AND GO TO NEXT CELL
      IF(LAYCON(K).EQ.1) CON=SC1(J,I,K)/(DELC(I)*DELR(J))
      IF(LAYCON(K).EQ.2 .OR. LAYCON(K).EQ.3) THEN
      KB=KB+1
      CON=SC2(J,I,KB)/(DELC(I)*DELR(J))
      ENDIF
      IF(LAYCON(K).EQ.0) CON=.15
C
      USLZ(J,I,K)=QBZ(J,I,K)*DELT+CON*(HOLD(J,I,K)-HNEW(J,I,K))+
     1 UOLDZ(J,I,K)
      GOTO 92
      ENDIF
C
C7K1-----SET UP BOUNDARY COEFFICIENTS FOR BOTTOMMOST ACTIVE CELL
C       ALONG A LAYER
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
C
C   DETERMINE THE VALUES OF THE COEFFICIENTS
```

```
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVM=2*DELL(J,I,K)*(DELL(J,I,K-1)+0.5*DELL(J,I,K))
      DVB=DELL(J,I,K-1)+0.5*DELL(J,I,K)
      ELSE
      DVM=DELL(J,I,K)*(DELL(J,I,K-1)+DELL(J,I,K))
      DVB=0.5*(DELL(J,I,K)+DELL(J,I,K-1))
      ENDIF
C
      ACOEF=1./DVM
      BCOEF=1./(DELL(J,I,K)*DELL(J,I,K))+ACOEF
      FAC1=1+FAC*BCOEF
C
      IF(I.EQ.1 .OR. (I.GT.1 .AND. IACT(J,I-1,K).EQ.0)) THEN
      DCF=DELC(I)+0.5*(DELC(I)+DELC(I+1))
      PCOEF=1./(2*DVB*DCF)
      QCOEF=PCOEF
      RCOEF=0.
      GCOEF=PCOEF
      HCOEF=-PCOEF
      OCOEF=0.
C
      ELSEIF(I.EQ.NROW .OR. (I.LT.NROW .AND. IACT(J,I+1,K).EQ.0)) THEN
      DCB=DELC(I)+0.5*(DELC(I)+DELC(I-1))
      PCOEF=0.
      QCOEF=1./(2*DVB*DCB)
      RCOEF=-QCOEF
      GCOEF=0.
      HCOEF=-QCOEF
      OCOEF=-QCOEF
C
      ELSE
      DCC=DELC(I)+0.5*(DELC(I+1)+DELC(I-1))
      PCOEF=1./(2*DVB*DCC)
      QCOEF=0.
      RCOEF=PCOEF
      GCOEF=PCOEF
      HCOEF=0.
```

```
      OCOEF=PCOEF
      ENDIF
C
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J-1,I,K).EQ.0))THEN
      DRF=DELR(J)+0.5*(DELR(J)+DELR(J+1))
      P2COEF=1./(2*DVB*DRF)
      Q2COEF=P2COEF
      R2COEF=0.
      G2COEF=P2COEF
      H2COEF=-P2COEF
      O2COEF=0.
C
      ELSEIF(J.EQ.NCOL .OR. (J.LT.NCOL .AND. IACT(J+1,I,K).EQ.0)) THEN
      DRB=DELR(J)+0.5*(DELR(J)+DELR(J-1))
      P2COEF=0.
      Q2COEF=-1./(2*DVB*DRB)
      R2COEF=-Q2COEF
      G2COEF=0.
      H2COEF=-Q2COEF
      O2COEF=-Q2COEF
C
      ELSE
      DRC=DELR(J)+0.5*(DELR(J+1)+DELR(J-1))
      P2COEF=1./(2*DVB*DRC)
      Q2COEF=0.
      R2COEF=P2COEF
      G2COEF=P2COEF
      H2COEF=0.
      O2COEF=P2COEF
      ENDIF
C
C7K4----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATION TO
C      GOVERNING EQUATION
      XCON=-FACH*(P2COEF*USLX(J+1,I,K)+Q2COEF*USLX(J,I,K)-R2COEF*
     1USLX(J-1,I,K)-G2COEF*USLX(J+1,I,K-1)+H2COEF*USLX(J,I,K-1)+O2COEF*
     2 USLX(J-1,I,K-1)+P2COEF*UOLDX(J+1,I,K)+Q2COEF*UOLDX(J,I,K)-R2COEF*
     3 UOLDX(J-1,I,K)-G2COEF*UOLDX(J+1,I,K-1)+H2COEF*UOLDX(J,I,K-1)+
```

```
     4 OCOEF*UOLDX(J-1,I,K-1))
C
      YCON=-FACH*(PCOEF*USLY(J,I+1,K)+QCOEF*USLY(J,I,K)-RCOEF*
     1USLY(J,I-1,K)-GCOEF*USLY(J,I+1,K-1)+HCOEF*USLY(J,I,K-1)+OCOEF*
     2USLY(J,I-1,K-1)+PCOEF*UOLDY(J,I+1,K)+QCOEF*UOLDY(J,I,K)-RCOEF*
     3UOLDY(J,I-1,K)-GCOEF*UOLDY(J,I+1,K-1)+HCOEF*UOLDY(J,I,K-1)+OCOEF*
     4UOLDY(J,I-1,K-1))
C
      RESID=FAC*(ACOEF*USLZ(J,I,K-1)-BCOEF*USLZ(J,I,K)+ACOEF*
     1UOLDZ(J,I,K-1)-BCOEF*UOLDZ(J,I,K))+QBZ(J,I,K)*DELT+
     2 +XCON+YCON+UOLDZ(J,I,K)-USLZ(J,I,K)
C
C7K5----CALCULATE SOURCE TERMS AND CONSTANTS
      IF(NUMIT.EQ.1) COEF=XCON+YCON+QBZ(J,I,K)*DELT
C
      GOTO 91
      ENDIF
C
C7L1----SET UP COEFFICIENTS FOR ALL INTERIOR CELLS THROUGH A LAYER
C
C    DETERMINE THE VALUES OF THE COEFRICIENTS
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUNE(J,I,K-2).EQ.0)) THEN
      DVM=2*DELL(J,I,K)*(DELL(J,I,K-1)+0.5*DELL(J,I,K))
      DVC=DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1)
      ELSE
      DVM=DELL(J,I,K)*(DELL(J,I,K-1)+DELL(J,I,K))
      DVC=DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1))
      ENDIF
      DVP=DELL(J,I,K)*(DELL(J,I,K+1)+DELL(J,I,K))
C
      ACOEF=1./DVM
      CCOEF=1./DVP
      BCOEF=ACOEF+CCOEF
      FAC1=1+FAC*BCOEF
C
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J,I-1,K).EQ.0)) THEN
      DCF=DELC(J)+0.5*(DELC(J)+DELC(J+1))
```

```
      DCOEF=1./(2*DVC*DCF)
      ECOEF=DCOEF
      FCOEF=0.
      GCOEF=DCOEF
      HCOEF=-DCOEF
      OCOEF=0.
C
      ELSEIF(I.EQ.NROW .OR. (I.LT.NROW .AND. IACT(J,I+1,K).EQ.0)) THEN
      DCB=DELC(I)+0.5*(DELC(I)+DELC(I-1))
      DCOEF=0.
      ECOEF=-1./(2*DVC*DCB)
      FCOEF=-ECOEF
      GCOEF=0.
      HCOEF=-ECOEF
      OCOEF=-ECOEF
C
      ELSE
      DCC=DELC(I)+0.5*(DELC(I+1)+DELC(I-1))
      DCOEF=1./(2*DVC*DCC)
      ECOEF=0.
      FCOEF=DCOEF
      GCOEF=DCOEF
      HCOEF=0.
      OCOEF=DCOEF
      ENDIF
C
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J-1,I,K).EQ.0)) THEN
      DRF=DELR(J)+0.5*(DELR(J)+DELR(J+1))
      D2COEF=1./(2*DVC*DRF)
      E2COEF=D2COEF
      F2COEF=0.
      G2COEF=D2COEF
      H2COEF=-D2COEF
      O2COEF=0.
C
      ELSEIF(J.EQ.NCOL .OR. (J.LT.NCOL .AND. IACT(J+1,I,K).EQ.0)) THEN
      DRB=DELR(J)+0.5*(DELR(J)+DELR(J-1))
```

```
      D2COEF=0.
      E2COEF=-1./(2*DVC*DRB)
      F2COEF=-E2COEF
      G2COEF=0.
      H2COEF=-E2COEF
      02COEF=-E2COEF
C
      ELSE
      DRC=DELR(J)+0.5*(DELR(J+1)+DELR(J-1))
      D2COEF=1./(2*DVC*DRC)
      E2COEF=0.
      F2COEF=D2COEF
      G2COEF=D2COEF
      H2COEF=0.
      02COEF=D2COEF
      ENDIF
C
C7L4----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATIONS TO
C        GOVERNING EQUATION
      XCON=-FACH*(D2COEF*USLX(J+1,I,K+1)+E2COEF*USLX(J,I,K+1)-F2COEF
     1USLX(J-1,I,K+1)-G2COEF*USLX(J+1,I,K-1)+H2COEF*USLX(J,I,K-1)+
     2 02COEF*USLX(J-1,I,K-1)+D2COEF*UOLDX(J+1,I,K+1)+E2COEF
     3 UOLDX(J,I,K+1)-F2COEF*UOLDX(J-1,I,K+1)-G2COEF*UOLDX(J+1,I,K-1)
     4 +H2COEF*UOLDX(J,I,K-1)+02COEF*UOLDX(J-1,I,K-1))
C
      YCON=-FACH*(DCOEF*USLY(J,I+1,K+1)+ECOEF*USLY(J,I,K+1)-
     1 FCOEF*USLY(J,I-1,K+1)-GCOEF*USLY(J,I+1,K-1)+HCOEF*USLY(J,I,K-1)
     2 +OCOEF*USLY(J,I-1,K-1)+DCOEF*UOLDY(J,I+1,K+1)+ECOEF
     3 UOLDY(J,I,K+1)-FCOEF*UOLDY(J,I-1,K+1)-GCOEF*UOLDY(J,I+1,K-1)+
     4 HCOEF*UOLDY(J,I,K-1)+OCOEF*UOLDY(J,I-1,K-1))
C
      RESID=FAC*(CCOEF*USLZ(J,I,K+1)-BCOEF*USLZ(J,I,K)+ACOEF
     1USLZ(J,I,K-1)+CCOEF*UOLDZ(J,I,K+1)-BCOEF*UOLDZ(J,I,K)+ACOEF
     2UOLDZ(J,I,K-1))+QBZ(J,I,K)*DELT+XCON+YCON+UOLDZ(J,I,K)-
     3 USLZ(J,I,K)
C
C7L5-----CALCULATE SOURCE TERMS AND CONSTANTS
```

```
      IF(NUMIT.EQ.1) COEF=XCON+YCON+QBZ(J,I,K)*DELT
C
C7M-----CALCULATE NORM OF TRUE ERROR AND NEW VALUE OF
C        DISPLACEMENT
   91  ANORMF=ANORMF+ABS(COEF)
      ANORM=ANORM+ABS(RESID)
      USLZ(J,I,K)=USLZ(J,I,K)-OMEGA*RESID/(-FAC1)
   92 CONTINUE
C
      KSW=3-KSW
C
C     MAKE ADJUSTMENT TO VERTICAL DISPLACEMENT IN DRY CELLS
      KN=0.
      DO 88 KK=1,NLAY
      IF(IACT(J,I,KK).GT.0 .AND. IBOUND(J,I,KK).EQ.0) KN=KN+1
   88 CONTINUE
      IF(KN.EQ.0 .OR. KN+1.GT.NLAY) GOTO 90
      DO 89 KK=1,KN
      USLZ(J,I,KK)=USLZ(J,I,KK)+(USLZ(J,I,KN+1-ZADJ)
   89 CONTINUE
   90 CONTINUE
C
C7N------CALCULATE OMEGA
      IJSW=3-IJSW
      IF(NUMIT.EQ.1 .AND. IPASS.EQ.1) THEN
      OMEGA=1.D0/(1.D0-0.5D0*RJACZ**2)
      ELSE
      OMEGA=1.D0/(1.D0-0.25D0*RJACZ**2*OMEGA)
      ENDIF
   99 CONTINUE
C
C7O------CHECK TO SEE IF CONVERGENCE IS MET.
      IF(ANORM.LE.XCLOSE*ANORMF) THEN
      WRITE(IOUT,72) NUMIT
   72 FORMAT(1X,'NUMBER OF INNER ITERATIONS IN Z IS',I5)
      GOTO 3000
      ENDIF
```

```
         IF(ANORM.GT.XCLOSE*ANORMF .AND. NUMIT.LT.ISTEP) GOTO 16
         IF(NUMIT.GE.ISTEP) WRITE(IOUT,49) ISTEP,KSTP,
        1 KPER,ANORM,ANORMF,NUMOUT
      49 FORMAT(1X,'***WARNING*** CONVERGENCE NOT MET AFTER',I5,/
        1 INNER ITERATIONS, AT TIME STEP',I5,' OF STRESS PERIOD',I5,/,
        2 1X,' IN Z. FINAL NORM OF RESIDUAL CALCULATED TO BE',G12.6,/,
        3 1X,' IN Z. FINAL TRUE ERROR CALCULATED TO BE',G12.6,/,
        3 1X,'OUTER ITERATION NUMBER',I5,)
C
C8-----CHECK IF COLUMN DIRECTION ITERATION NECESSARY
   1000  IF(NCOL.EQ.1) GOTO 2000
C
C8A-----SET INITIAL OMEGA
         OMEGA=1.D0
C
C8B-----ITERATE TO SOLVE FOR THE DISPLACEMENT IN THE X DIRECTION
         NUMIT=0.
      15 DIFFX=0.
         ANORMF=0.D0
         NUMIT=NUMIT+1
C
C8C-----SET NORM OF RESIDUAL TO ZERO, AND BEGIN ODD-EVEN ORDERING
         ANORM=0.D0
         IKSW=1
         DO 98 IPASS=1,2
         JSW=IKSW
C
C8D-----LOOP THROUGH LAYERS AND ROWS
      18 DO 20 K=1,NLAY
         DO 20 I=1,NROW
C
C8E-----CALCULATE THE STRAIN IN THE X DIRECTION
         IF(NLAY.EQ.1) GOTO 11
         CALL SUSL1X(USLX,IACT,NCOL,I,K,DELR,STRNX,NROW,NLAY)
C
C8F-----INNER LOOP FOR COLUMNS
      11 DO 40 J=JSW,NCOL,2
```

```
C
C8G----CHECK FOR NON-ACTIVE CELLS
C   CHECK FOR ACTIVE CELLS AND LOOP THROUGH CELLS WHERE IACT=1,
C   BUT WE WANT TO KEEP ANY CELLS THAT MAY HAVE GONE DRY
      IF(IACT(J,I,K).EQ.0) GOTO 40
C
C8I----MAKE CORRECTION FOR PROPER SPECIFIC STORAGE VALUE. THE
C   HORIZONTAL DISPLACEMENT USES A STORAGE COEFFICIENT DIVIDED BY
C   THICKNESS, SET CONSTANTS
      IF(LAYCON(K).EQ.1) SS=SSE(J,I,K)
      IF(LAYCON(K).NE.1) SS=SC1(J,I,K)/
     1  (DELL(J,I,K)*DELR(J)*DELC(I))
      IF(KUNIT.EQ.0 .AND. SS.LT.4.32E-6) SS=4.32E-6
      IF(KUNIT.NE.0 .AND. SS.LT.1.36E-6) SS=1.36E-6
C
C   CALCULATE STORAGE FOR ALL LAYCON VALUES
      IF(NLAY.EQ.1 .OR. USLX(J,I,K).EQ.0.) SSK(J,I,K)=SS
C
C   SET CONSTANT COEFFICIENTS FOR ALL CELLS REGARDLESS OF BOUNDARY
      H=HC(J,I,K)
      FAC=H*DELT/(CON*SS)
      FACV=H*DELT/(CON*SSK(J,I,K))
C
C8J1----SET UP BOUNDARY COEFFICIENTS FOR LEFTMOST ACTIVE CELL
C    ALONG A ROW
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J-1,I,K).EQ.0))THEN
C
C   THEN DETERMINE THE VALUES OF THE COEFFICIENTS
      DRP=DELR(J)*(DELR(J+1)+DELR(J))
      DRF=0.5*(DELR(J)+DELR(J+1))
C
      CCOEF=1/DRP
      BCOEF=1/DRP+1/(DELR(J)*DELR(J))
      FAC1=1+FAC*BCOEF
C
      IF(I.EQ.1 .OR. (I.GT.1 .AND. IACT(J,I-1,K).EQ.0)) THEN
      DCF=DELC(I)+0.5*(DELC(I)+DELC(I+1))
```

```
      DCOEF=1./(2*DRF*DCF)
      ECOEF=DCOEF
      FCOEF=0.
      PCOEF=DCOEF
      QCOEF=-DCOEF
      RCOEF=0.
C
      ELSEIF(I.EQ.NROW .OR. (I.LT.NROW .AND. IACT(J,I+1,K).EQ.0)) THEN
      DCB=DELC(I)+0.5*(DELC(I)+DELC(I-1))
      DCOEF=0.
      ECOEF=-1./(2*DRF*DCB)
      FCOEF=-ECOEF
      PCOEF=0.
      QCOEF=-ECOEF
      RCOEF=-ECOEF
C
      ELSE
      DCC=DELC(I)+0.5*(DELC(I+1)+DELC(I-1))
      DCOEF=1./(2*DRF*DCC)
      ECOEF=0.
      FCOEF=DCOEF
      PCOEF=DCOEF
      QCOEF=0.
      RCOEF=DCOEF
      ENDIF
C
      ZCON=0.
      IF(NLAY.EQ.1) GOTO 33
      IF(K+1.GT.NLAY .AND. IBOUND(J,I,K-1).EQ.0) GOTO 33
C
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) GOTO 33
C
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVB=1.5*DELL(J,I,K)+DELL(J,I,K-1)
      ELSE
      DVB=DELL(J,I,K)+0.5*(DELL(J,I,K)+DELL(J,I,K-1))
```

```
      ENDIF
      SCOEF=0.
      UCOEF=1./(2*DRF*DVB)
      VCOEF=UCOEF
      X2COEF=0.
      Y2COEF=UCOEF
      Z2COEF=UCOEF
C
      ELSE
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVC=DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1)
      ELSE
      DVC=DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1))
      ENDIF
      SCOEF=1./(2*DRF*DVC)
      UCOEF=0.
      VCOEF=SCOEF
      X2COEF=SCOEF
      Y2COEF=0.
      Z2COEF=SCOEF
      ENDIF
C
      ZCON=-FACW*(SCOEF*USLZ(J+1,I,K+1)-UCOEF*USLZ(J+1,I,K)-VCOEF*USLZ
     1 (J+1,I,K-1)-X2COEF*USLZ(J,I,K+1)+Y2COEF*USLZ(J,I,K)+
     2 Z2COEF*USLZ(J,I,K-1)+SCOEF*UOLDZ(J+1,I,K+1)-
     3 UCOEF*UOLDZ(J+1,I,K)-VCOEF*UOLDZ(J+1,I,K-1)-X2COEF*UOLDZ(J,I,
     4 K+1)+Y2COEF*UOLDZ(J,I,K)+Z2COEF*UOLDZ(J,I,K-1))
C
C8J2----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATION TO
C      GOVERNING EQUATION
   33 YCON=FAC*(DCOEF*USLY(J+1,I+1,K)+ECOEF*USLY(J+1,I,K)-
     1 FCOEF*USLY(J+1,I-1,K)-PCOEF*USLY(J,I+1,K)+QCOEF*USLY(J,I,K)+
     2 RCOEF*USLY(J,I-1,K)+DCOEF*UOLDY(J+1,I+1,K)+ECOEF*UOLDY(J+1,I,K)
     3 -FCOEF*UOLDY(J+1,I-1,K)-PCOEF*UOLDY(J,I+1,K)+QCOEF*UOLDY(J,I,K)
     4 +RCOEF*UOLDY(J,I-1,K))
C
      RESID=FAC*(CCOEF*USLX(J+1,I,K)-BCOEF*USLX(J,I,K)+CCOEF*
```

```
      1 UOLDX(J+1,I,K)-BCOEF*UOLDX(J,I,K))+YCON+ZCON+QBX(J,I,K)*DELT+
      2 UOLDX(J,I,K)-USLX(J,I,K)
C
C8J3-----CALCULATE SOURCE TERMS AND CONSTANTS
      IF(NUMIT.EQ.1)COEF=YCON+ZCON+QBX(J,I,K)*DELT
C
      GOTO 39
      ENDIF
C
C8K1-----SET UP BOUNDARY COEFFICIENTS FOR RIGHTMOST ACTIVE CELL
C      ALONG A COLUMN
      IF(J.EQ.NCOL .OR. (J.LT.NCOL .AND. IACT(J+1,I,K).EQ.0)) THEN
C
C    THEN DETERMINE THE VALUES OF THE COEFFICIENTS
      DRM=DELR(J)*(DELR(J-1)+DELR(J))
      DRB=0.5*(DELR(J)+DELR(J-1))
C
      ACOEF=1/DRM
      BCOEF=1/DRM+1/(DELR(J)*DELR(J))
      FAC1=1+FAC*BCOEF
C
      IF(I.EQ.1 .OR. (I.GT.1 .AND. IACT(J,I-1,K).EQ.0)) THEN
      DCF=DELC(I)+0.5*(DELC(I)+DELC(I+1))
      GCOEF=1./(2*DRB*DCF)
      HCOEF=-GCOEF
      OCOEF=0.
      PCOEF=GCOEF
      QCOEF=GCOEF
      RCOEF=0.
C
      ELSEIF(I.EQ.NROW .OR. (I.LT.NROW .AND. IACT(J,I+1,K).EQ.0)) THEN
      DCB=DELC(I)+0.5*(DELC(I)+DELC(I-1))
      GCOEF=0.
      HCOEF=1./(2*DRB*DCB)
      OCOEF=HCOEF
      PCOEF=0.
      QCOEF=-HCOEF
```

```
      RCOEF=HCOEF
C
      ELSE
      DCC=DELC(J)+0.5*(DELC(J+1)+DELC(J-1))
      GCOEF=1./(2*DRB*DCC)
      HCOEF=0.
      OCOEF=GCOEF
      PCOEF=GCOEF
      QCOEF=0.
      RCOEF=GCOEF
      ENDIF
C
      ZCON=0.
      IF(NLAY.EQ.1) GOTO 34
      IF(K+1.GT.NLAY .AND. IBOUND(J,I,K-1).EQ.0) GOTO 34
C
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) GOTO 34
C
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVB=1.5*DELL(J,I,K)+DELL(J,I,K-1)
      ELSE
      DVB=DELL(J,I,K)+0.5*(DELL(J,I,K)+DELL(J,I,K-1))
      ENDIF
      XCOEF=1./(2*DRB*DVB)
      YCOEF=0.
      ZCOEF=XCOEF
      X2COEF=0.
      Y2COEF=XCOEF
      Z2COEF=XCOEF
C
      ELSE
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVC=DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1)
      ELSE
      DVC=DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1))
      ENDIF
```

```
        XCOEF=0.
        YCOEF=1./(2*DRB*DVC)
        ZCOEF=YCOEF
        X2COEF=YCOEF
        Y2COEF=0.
        Z2COEF=YCOEF
       ENDIF
C
        ZCON=-FACV*(X2COEF*USLZ(J,I,K+1)-Y2COEF*USLZ(J,I,K)-
       1Z2COEF*USLZ(J,I,K-1)-YCOEF*USLZ(J-1,I,K+1)+XCOEF*USLZ(J-1,I,K)+
       2 ZCOEF*USLZ(J-1,I,K-1)+X2COEF*UOLDZ(J,I,K+1)-
       3Y2COEF*UOLDZ(J,I,K)-Z2COEF*UOLDZ(J,I,K-1)-YCOEF*UOLDZ(J-1,I,K+1)+
       4XCOEF*UOLDZ(J-1,I,K)+ZCOEF*UOLDZ(J-1,I,K-1))
C
C8K2----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATION TO
C        GOVERNING EQUATION
   34   YCON=FAC*(PCOEF*USLY(J,I+1,K)+QCOEF*USLY(J,I,K)-RCOEF*
       1USLY(J,I-1,K)-GCOEF*USLY(J-1,I+1,K)+HCOEF*USLY(J-1,I,K)+OCOEF*
       2USLY(J-1,I-1,K)+PCOEF*UOLDY(J,I+1,K)+QCOEF*UOLDY(J,I,K)-RCOEF*
       3UOLDY(J,I-1,K)-GCOEF*UOLDY(J-1,I+1,K)+HCOEF*UOLDY(J-1,I,K)+
       4 OCOEF*UOLDY(J-1,I-1,K))
C
        RESID=FAC*(ACOEF*USLX(J-1,I,K)*BCOEF*USLX(J,I,K)+
       1 ACOEF*UOLDX(J-1,I,K)+BCOEF*UOLDX(J,I,K))+YCON+ZCON+
       2QBX(J,I,K)*DELT+UOLDX(J,I,K)-USLX(J,I,K)
C
C8K3----CALCULATE SOURCE TERMS AND CONSTANTS
        IF(NUMIT.EQ.1) COEF=YCON+ZCON+QBX(J,I,K)*DELT
C
        GOTO 39
        ENDIF
C
C8L1----SET UP COEFFICIENTS FOR ALL INTERIOR CELLS ALONG A ROW
C
C    DETERMINE VALUES OF THE COEFFICIENTS
        DRM=DELR(J)*(DELR(J-1)+DELR(J))
        DRP=DELR(J)*(DELR(J+1)+DELR(J))
```

```
      DRC=DELR(J)+0.5*(DELR(J-1)+DELR(J+1))
C
      CCOEF=1/DRP
      ACOEF=1/DRM
      BCOEF=ACOEF+CCOEF
      FAC1=1+FAC*BCOEF
C
      IF(I.EQ.1 .OR. (I.GT.1 .AND. IACT(J,I-1,K).EQ.0)) THEN
      DCF=DELC(I)+0.5*(DELC(I)+DELC(I+1))
      DCOEF=1./(2*DRC*DCF)
      ECOEF=DCOEF
      FCOEF=0.
      GCOEF=DCOEF
      HCOEF=-DCOEF
      OCOEF=0.
C
      ELSEIF(I.EQ.NROW .OR. (I.LT.NROW .AND. IACT(J,I+1,K).EQ.0)) THEN
      DCB=DELC(I)+0.5*(DELC(I)+DELC(I-1))
      DCOEF=0.
      ECOEF=-1./(2*DRC*DCB)
      FCOEF=-ECOEF
      GCOEF=0.
      HCOEF=-ECOEF
      OCOEF=-ECOEF
C
      ELSE
      DCC=DELC(I)+0.5*(DELC(I+1)+DELC(I-1))
      DCOEF=1./(2*DRC*DCC)
      ECOEF=0.
      FCOEF=DCOEF
      GCOEF=DCOEF
      HCOEF=0.
      OCOEF=DCOEF
      ENDIF
C
      ZCON=0.
      IF(NLAY.EQ.1) GOTO 35
```

```
      IF(K+1.GT.NLAY .AND. IBOUND(J,I,K-1).EQ.0) GOTO 35
C
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) GOTO 35
C
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVB=1.5*DELL(J,I,K)+DELL(J,I,K-1)
      ELSE
      DVB=DELL(J,I,K)+0.5*(DELL(J,I,K)+DELL(J,I,K-1))
      ENDIF
      SCOEF=0.
      UCOEF=1./(2*DRC*DVB)
      VCOEF=UCOEF
      XCOEF=UCOEF
      YCOEF=0.
      ZCOEF=UCOEF
C
      ELSE
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVC=DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1)
      ELSE
      DVC=DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1))
      ENDIF
      SCOEF=1./(2*DRC*DVC)
      UCOEF=0.
      VCOEF=SCOEF
      XCOEF=0.
      YCOEF=SCOEF
      ZCOEF=SCOEF
      ENDIF
C
      ZCON=-FACV*(SCOEF*USLZ(J+1,I,K+1)-UCOEF*USLZ(J+1,I,K)-VCOEF*
     1USLZ(J+1,I,K-1)+XCOEF*USLZ(J-1,I,K)-YCOEF*USLZ(J-1,I,K+1)+
     2ZCOEF*USLZ(J-1,I,K-1)+SCOEF*UOLDZ(J+1,I,K+1)-UCOEF*UOLDZ(J+1,I,K)
     3-VCOEF*UOLDZ(J+1,I,K-1)+XCOEF*UOLDZ(J-1,I,K)
     4-YCOEF*UOLDZ(J-1,I,K+1)+ZCOEF*UOLDZ(J-1,I,K-1))
C
```

```
C8L2----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATION TO
C       GOVERNING EQUATION
   35 YCON=FAC*(DCOEF*USLY(J+1,I+1,K)+ECOEF*USLY(J+1,I,K)-FCOEF
     1USLY(J+1,I-1,K)-GCOEF*USLY(J-1,I+1,K)+HCOEF*USLY(J-1,I,K)+
     2OCOEF*USLY(J-1,I-1,K)+DCOEF*UOLDY(J+1,I+1,K)+ECOEF*UOLDY(J+1,I,K)
     3-FCOEF*UOLDY(J+1,I-1,K)-GCOEF*UOLDY(J-1,I+1,K)+
     4HCOEF*UOLDY(J-1,I,K)+OCOEF*UOLDY(J-1,I-1,K))
C
      RESID=FAC*(CCOEF*USLX(J+1,I,K)-BCOEF*USLX(J,I,K)+ACOEF
     1USLX(J-1,I,K)+CCOEF*UOLDX(J+1,I,K)-BCOEF*UOLDX(J,I,K)+
     2ACOEF*UOLDX(J-1,I,K))+YCON+ZCON+QBX(J,I,K)*DELT+UOLDX(J,I,K)-
     3USLX(J,I,K)
C
C8L3-----CALCULATE SOURCE TERMS AND CONSTANTS
      IF(NUMIT.EQ.1)COEF=YCON+ZCON+QBX(J,I,K)*DELT
C
C8M-----CALCULATE NORM OF TRUE ERROR AND NEW VALUE OF
C        DISPLACEMENT
   39 ANORMF=ANORMF+ABS(COEF)
      ANORM=ANORM+ABS(RESID)
      USLX(J,I,K)=USLX(J,I,K)-OMEGA*RESID/(-FAC1)
   40 CONTINUE
C
      JSW=3-JSW
   20 CONTINUE
C
C8N------CALCULATE OMEGA
      IKSW=3-IKSW
      IF(NUMIT.EQ.1 .AND. IPASS.EQ.1) THEN
--->  OMEGA=1.D0/(1.D0-0.5D0*RJACX**2)
      ELSE
      OMEGA=1.D0/(1.D0-0.25D0*RJACX**2*OMEGA)
      ENDIF
   98 CONTINUE
C
C8O-----CHECK FOR CONVERGENCE
      IF(ANORM.LE.XCLOSE*ANORMF) THEN
```

```
      WRITE(IOUT,70) NUMIT
   70 FORMAT(1X,'NUMBER OF INNER ITERATIONS IN X IS',I5)
      GOTO 1000
      ENDIF
      IF(ANORM.GT.XCLOSE*ANORMF .AND. NUMIT.LT.ISTEP) GOTO 15
      IF(NUMIT.GE.ISTEP) WRITE(IOUT,48) ISTEP,KSTP,
     1 KPER,ANORM,ANORMF,NUMOUT
   48 FORMAT(1X,'***WARNING*** CONVERGENCE NOT MET AFTER',I5/
     1 INNER ITERATIONS, AT TIME STEP',I5,' OF STRESS PERIOD',I5,/
     2 1X,' IN X. FINAL NORM OF RESIDUAL CALCULATED TO BE',F10.6,/,
     3 1X,' IN X. FINAL NORM OF TRUE ERROR CALCULATED TO BE',F10.6,/,
     4 1X,'OUTER ITERATION NUMBER',I5,)
C
C9-----CHECK IF ROW DIRECTION ITERATION NECESSARY
 2000 IF(NROW.EQ.1) GOTO 3000
C
C9A---SET INITIAL OMEGA
      OMEGA=1.D0
C
C9B---ITERATE TO SOLVE FOR THE DISPLACEMENT IN THE Y DIRECTION
      NUMIT=0.
   17 DIFFY=0.
      ANORMF=0.D0
      NUMIT=NUMIT+1
C
C9C-----SET NORM FOR RESIDUAL TO ZERO AND BEGIN ODD-EVEN ORDERING
      ANORM=0.D0
      JKSW=1
      DO 97 IPASS=1,2
      ISW=JKSW
C
C9D-----LOOP THROUGH LAYERS AND COLUMNS
      DO 25 K=1,NLAY
      DO 25 J=1,NCOL
C
C9E-----CALCULATE STRAIN IN THE Y DIRECTION
      IF(NLAY.EQ.1) GOTO 12
```

```
      CALL SUSLTY(USLY,IACT,J,NROW,K,DELC,STRNY,NCOL,NLAY)
C
C9F-----INNER LOOP FOR ROWS
   12 DO 45 I=ISW,NROW,2
C
C9G-----CHECK FOR NON-ACTIVE CELLS
C     CHECK FOR ACTIVE CELLS AND LOOP THROUGH CELLS WHERE IACT=1
      IF(IACT(J,I,K).EQ.0) GOTO 45
C
C9I-----CALCULATE PROPER VALUE OF SPECIFIC STORAGE, SET CONSTANTS
      IF(LAYCON(K).EQ.1) SS=SSE(J,I,K)
      IF(LAYCON(K).NE.1) SS=SC1(J,I,K)/
     1  (DELL(J,I,K)*DELC(I)*DELR(J))
      IF(KUNIT.EQ.0 .AND. SS.LT.4.32E-6) SS=4.32E-6
      IF(KUNIT.NE.0 .AND. SS.LT.1.36E-6) SS=1.36E-6
C
C   CALCULATE STORAGE FOR ALL LAYCON VALUES
      IF(NLAY.EQ.1 .OR. USLY(J,I,K).EQ.0.) SSK(J,I,K)=SS
C
C   SET CONSTANT COEFFICIENTS FOR ALL CELLS REGARDLESS OF BOUNDARY
      H=HC(J,I,K)
      FAC=H*DELT/(CON*SS)
      FACV=H*DELT/(CON*SSK(J,I,K))
C
C9J1-----SET UP BOUNDARY COEFFICIENTS FOR LEFTMOST ACTIVE CELL
C        ALONG A COLUMN
      IF(I.EQ.1 .OR. (I.GT.1 .AND. IACT(J,I-1,K).EQ.0)) THEN
C
C   DETERMINE VALUES OF THE COEFFICIENTS
      DCP=DELC(I)*(DELC(I+1)+DELC(I))
      DCF=0.5*(DELC(I+1)+DELC(I))
C
      CCOEF=1/DCP
      BCOEF=CCOEF+1/(DELC(I)*DELC(I))
      FAC1=1+FAC*BCOEF
C
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J-1,I,K).EQ.0)) THEN
```

```
      DRF=DELR(J)+0.5*(DELR(J)+DELR(J+1))
      DCOEF=1./(2*DCF*DRF)
      ECOEF=DCOEF
      FCOEF=0.
      PCOEF=DCOEF
      QCOEF=-DCOEF
      RCOEF=0.
C
      ELSEIF(J.EQ.NCOL .OR. (J.LT.NCOL .AND. IACT(J+1,I,K).EQ.0)) THEN
      DRB=DELR(J)+0.5*(DELR(J)+DELR(J-1))
      DCOEF=0.
      ECOEF=-1./(2*DCF*DRB)
      FCOEF=-ECOEF
      PCOEF=0.
      QCOEF=-ECOEF
      RCOEF=-ECOEF
C
      ELSE
      DRC=DELR(J)+0.5*(DELR(J+1)+DELR(J-1))
      DCOEF=1./(2*DCF*DRC)
      ECOEF=0.
      FCOEF=DCOEF
      PCOEF=DCOEF
      QCOEF=0.
      RCOEF=DCOEF
      ENDIF
C
      ZCON=0.
      IF(NLAY.EQ.1) GOTO 36
      IF(K+1.GT.NLAY .AND. IBOUND(J,I,K-1).EQ.0) GOTO 36
C
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0))GOTO 36
C
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVB=1.5*DELL(J,I,K)+DELL(J,I,K-1)
      ELSE
```

```
      DVB=DELL(J,I,K)+0.5*(DELL(J,I,K)+DELL(J,I,K-1))
      ENDIF
      SCOEF=0.
      UCOEF=1./(2*DCF*DVB)
      VCOEF=UCOEF
      X2COEF=0.
      Y2COEF=UCOEF
      Z2COEF=UCOEF
C
      ELSE
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVC=DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1)
      ELSE
      DVC=DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1))
      ENDIF
      SCOEF=1./(2*DCF*DVC)
      UCOEF=0.
      VCOEF=SCOEF
      X2COEF=SCOEF
      Y2COEF=0.
      Z2COEF=SCOEF
      ENDIF
C
      ZCON=-FACV*(SCOEF*USLZ(J,I+1,K+1)-UCOEF*USLZ(J,I+1,K)-
     1VCOEF*USLZ(J,I+1,K-1)-X2COEF*USLZ(J,I,K+1)+Y2COEF*USLZ(J,I,K)
     2+Z2COEF*USLZ(J,I,K-1)+SCOEF*UOLDZ(J,I+1,K+1)
     3-UCOEF*UOLDZ(J,I+1,K)-VCOEF*UOLDZ(J,I+1,K-1)-X2COEF*UOLDZ(J,I,
     4 K+1)+Y2COEF*UOLDZ(J,I,K)-Z2COEF*UOLDZ(J,I,K-1))
C
C9J2----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATION TO
C       GOVERNING EQUATION
   36 XCON=FAC*(DCOEF*USLX(J+1,I+1,K)+ECOEF*USLX(J,I+1,K)-
     1FCOEF*USLX(J-1,I+1,K)-PCOEF*USLX(J+1,I,K)+QCOEF*USLX(J,I,K)+
     2RCOEF*USLX(J-1,I,K)+DCOEF*UOLDX(J+1,I+1,K)+ECOEF*UOLDX(J,I+1,K)
     3-FCOEF*UOLDX(J-1,I+1,K)-PCOEF*UOLDX(J+1,I,K)+QCOEF*UOLDX(J,I,K)
     4 +RCOEF*UOLDX(J-1,I,K))
C
```

```
      RESID=FAC*(CCOEF*USLY(J,I+1,K)-BCOEF*USLY(J,I,K)+CCOEF*
     1UOLDY(J,I+1,K)-BCOEF*UOLDY(J,I,K))+XCON+ZCON+QBY(J,I,K)*DELT+
     2 UOLDY(J,I,K)-USLY(J,I,K)
C
C9J3----CALCULATE SOURCE TERMS AND CONSTANTS
      IF(NUMIT.EQ.1) COEF=XCON+ZCON+QBY(J,I,K)*DELT
C
      GOTO 44
      ENDIF
C
C9K1----SET UP BOUNDARY COEFFICIENTS FOR RIGHTMOST ACTIVE CELL
C        ALONG A COLUMN
      IF(I.EQ.NROW .OR. (I.LT.NROW .AND. IACT(J,I+1,K).EQ.0))THEN
C
C    DETERMINE VALUES OF THE COEFFICIENTS
      DCM=DELC(I)*(DELC(I-1)+DELC(I))
      DCB=0.5*(DELC(I-1)+DELC(I))
C
      ACOEF=1/DCM
      BCOEF=ACOEF+1./(DELC(I)*DELC(I))
      FAC1=1+FAC*BCOEF
C
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J-1,I,K).EQ.0)) THEN
      DRF=DELR(J)+0.5*(DELR(J)+DELR(J+1))
      GCOEF=1./(2*DCB*DRF)
      HCOEF=-GCOEF
      OCOEF=0.
      PCOEF=GCOEF
      QCOEF=GCOEF
      RCOEF=0.
C
      ELSEIF(J.EQ.NCOL .OR. (J.LT.NCOL .AND. IACT(J+1,I,K).EQ.0)) THEN
      DRB=DELR(J)+0.5*(DELR(J)+DELR(J-1))
      GCOEF=0.
      HCOEF=1./(2*DCB*DRB)
      OCOEF=HCOEF
      PCOEF=0.
```

```
      QCOEF=-HCOEF
      RCOEF=HCOEF
C
      ELSE
      DRC=DELR(J)+0.5*(DELR(J+1)+DELR(J-1))
      GCOEF=1./(2*DCB*DRC)
      HCOEF=0.
      OCOEF=GCOEF
      PCOEF=GCOEF
      QCOEF=0.
      RCOEF=GCOEF
      ENDIF
C
      ZCON=0.
      IF(NLAY.EQ.1) GOTO 37
      IF(K+1.GT.NLAY .AND. IBOUND(J,I,K-1).EQ.0) GOTO 37
C
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) GOTO 37
C
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVB=1.5*DELL(J,I,K)+DELL(J,I,K-1)
      ELSE
      DVB=DELL(J,I,K)+0.5*(DELL(J,I,K)+DELL(J,I,K-1))
      ENDIF
      XCOEF=1./(2*DCB*DVB)
      YCOEF=0.
      ZCOEF=XCOEF
      X2COEF=0.
      Y2COEF=XCOEF
      Z2COEF=XCOEF
C
      ELSE
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVC=DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1)
      ELSE
      DVC=DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1))
```

```
      ENDIF
      XCOEF=0.
      YCOEF=1./(2*DCB*DVC)
      ZCOEF=YCOEF
      X2COEF=YCOEF
      Y2COEF=0.
      Z2COEF=YCOEF
      ENDIF
C
      ZCON=-FACW*(X2COEF*USLZ(J,I,K+1)-Y2COEF*USLZ(J,I,K)-
     1Z2COEF*USLZ(J,I,K-1)-YCOEF*USLZ(J,I-1,K+1)+
     2XCOEF*USLZ(J,I-1,K)+ZCOEF*USLZ(J,I-1,K-1)+X2COEF*UOLDZ(J,I,K+1)-
     3 Y2COEF*UOLDZ(J,I,K)-Z2COEF*UOLDZ(J,I,K-1)-YCOEF*UOLDZ(J,I-1,K+1)+
     4 XCOEF*UOLDZ(J,M,K)+ZCOEF*UOLDZ(J,M,K-1))
C
C9K2----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATION TO
      GOVERNING EQUATION
   37 XCON=FAC*(PCOEF*USLX(J+1,I,K)+QCOEF*USLX(J,I,K)-RCOEF*
     1USLX(J-1,I,K)-GCOEF*USLX(J+1,I-1,K)+HCOEF*USLX(J,I-1,K)+OCOEF*
     2USLX(J-1,I-1,K)+PCOEF*UOLDX(J+1,I,K)+QCOEF*UOLDX(J,I,K)-
     3 RCOEF*UOLDX(J-1,I,K)-GCOEF*UOLDX(J+1,I-1,K)+HCOEF*
     4UOLDX(J,I-1,K)+OCOEF*UOLDX(J-1,I-1,K))
C
      RESID=FAC*(ACOEF*USLY(J,I+1,K)-BCOEF*USLY(J,I,K)+
     1 ACOEF*UOLDY(J,I+1,K)-BCOEF*UOLDY(J,I,K))+XCON+ZCON+QBY(J,I,K)
     2 *DELT+UOLDY(J,I,K)-USLY(J,I,K)
C
C9K3----CALCULATE SOURCE TERMS AND CONSTANTS
      IF(NUMIT.EQ.1) COEF=XCON+ZCON+QBY(J,I,K)*DELT
C
      GOTO 44
      ENDIF
C
C9L1----SET UP COEFFICIENTS FOR ALL INTERIOR CELLS ALONG A COLUMN
C
C    DETERMINE VALUES OF THE COEFFICIENTS
      DCM=DELC(J)*(DELC(J-1)+DELC(J))
```

```
      DCP=DELC(I)*(DELC(I+1)+DELC(I))
      DCC=DELC(I)+0.5*(DELC(I+1)+DELC(I-1))
C
      CCOEF=1/DCP
      ACOEF=1/DCM
      BCOEF=ACOEF+CCOEF
      FAC1=1+FAC*BCOEF
C
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J-1,I,K).EQ.0))THEN
      DRF=DELR(J)+0.5*(DELR(J)+DELR(J+1))
      DCOEF=1./(2*DCC*DRF)
      ECOEF=DCOEF
      FCOEF=0.
      GCOEF=DCOEF
      HCOEF=-DCOEF
      OCOEF=0.
C
      ELSEIF(J.EQ.NCOL .OR. (J.LT.NCOL .AND. IACT(J+1,I,K).EQ.0)) THEN
      DRB=DELR(J)+0.5*(DELR(J)+DELR(J-1))
      DCOEF=0.
      ECOEF=-1./(2*DCC*DRB)
      FCOEF=-ECOEF
      GCOEF=0.
      HCOEF=-ECOEF
      OCOEF=-ECOEF
C
      ELSE
      DRC=DELR(J)+0.5*(DELR(J+1)+DELR(J-1))
      DCOEF=1./(2*DCC*DRC)
      ECOEF=0.
      FCOEF=DCOEF
      GCOEF=DCOEF
      HCOEF=0.
      OCOEF=DCOEF
      ENDIF
C
      ZCON=0.
```

```
      IF(NLAY.EQ.1) GOTO 38
      IF(K+1.GT.NLAY .AND. IBOUND(J,I,K-1).EQ.0) GOTO 38
C
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) GOTO 38
C
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVB=1.5*DELL(J,I,K)+DELL(J,I,K-1)
      ELSE
      DVB=DELL(J,I,K)+0.5*(DELL(J,I,K)+DELL(J,I,K-1))
      ENDIF
      SCOEF=0.
      UCOEF=1./(2*DCC*DVB)
      VCOEF=UCOEF
      XCOEF=UCOEF
      YCOEF=0.
      ZCOEF=UCOEF
C
      ELSE
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      DVC=DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1)
      ELSE
      DVC=DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1))
      ENDIF
      SCOEF=1./(2*DCC*DVC)
      UCOEF=0.
      VCOEF=SCOEF
      XCOEF=0.
      YCOEF=SCOEF
      ZCOEF=SCOEF
      ENDIF
C
      ZCON=-FACW*(SCOEF*USLZ(J,I+1,K+1)-UCOEF*USLZ(J,I+1,K)-
     1 VCOEF*USLZ(J,I+1,K-1)+XCOEF*USLZ(J,I-1,K)-YCOEF*USLZ(J,I-1,K+1)+
     2 ZCOEF*USLZ(J,I-1,K-1)+SCOEF*UOLDZ(J,I+1,K+1)-UCOEF*UOLDZ(J,I+1,K)
     3 -VCOEF*UOLDZ(J,I+1,K-1)+XCOEF*UOLDZ(J,M,K)-
     4 YCOEF*UOLDZ(J,I-1,K+1)+ZCOEF*UOLDZ(J,I-1,K-1))
```

```
C
C9L2----NOW PUT COEFFICIENTS INTO NUMERICAL APPROXIMATION TO
C        GOVERNING EQUATION
   38 XCON=FAC*(DCOEF*USLX(J+1,I+1,K)+ECOEF*USLX(J,I+1,K)-
     1FCOEF*USLX(J-1,I+1,K)-GCOEF*USLX(J+1,I-1,K)+HCOEF*USLX(J,I-1,K)
     2+OCOEF*USLX(J-1,I-1,K)+DCOEF*UOLDX(J+1,I+1,K)+ECOEF*UOLDX(J,
     3I+1,K)-FCOEF*UOLDX(J-1,I+1,K)-GCOEF*UOLDX(J+1,I-1,K)+HCOEF*
     4UOLDX(J,I-1,K)+OCOEF*UOLDX(J-1,I-1,K))
C
      RESID=FAC*(CCOEF*USLY(J,I+1,K)-BCOEF*USLY(J,I,K)+ACOEF*
     1USLY(J,I-1,K)+CCOEF*UOLDY(J,I+1,K)-BCOEF*UOLDY(J,I,K)+ACOEF*
     2UOLDY(J,I-1,K))+XCON+ZCON+QBY(J,I,K)*DELT+UOLDY(J,I,K)-
     3USLY(J,I,K)
C
C9L3----CALCULATE SOURCE TERMS AND CONSTANTS
      IF(NUMIT.EQ.1) COEF=XCON+ZCON+QBY(J,I,K)*DELT
C
C9M-----CALCULATE NORM OF TRUE ERROR AND NEW VALUE OF
C        DISPLACEMENT
   44 ANORMF=ANORMF+ABS(COEF)
      ANORM=ANORM+ABS(RESID)
      USLY(J,I,K)=USLY(J,I,K)-OMEGA*RESID/(-FAC1)
   45 CONTINUE
C
      ISW=3-ISW
   25 CONTINUE
C
C9N-----CALCULATE OMEGA
      JKSW=3-JKSW
      IF(NUMIT.EQ.1 .AND. IPASS.EQ.1) THEN
      OMEGA=1.D0/(1.D0-0.5D0*RJACY**2)
      ELSE
      OMEGA=1.D0/(1.D0-0.25D0*RJACY**2*OMEGA)
      ENDIF
   97 CONTINUE
C
C9O-----CHECK FOR CONVERGENCE
```

```
      IF(ANORM.LE.XCLOSE*ANORMF) THEN
      WRITE(IOUT,71) NUMIT
   71 FORMAT(1X,'NUMBER OF INNER ITERATIONS IN Y IS',I5)
      GOTO 2000
      ENDIF
      IF(ANORM.GT.XCLOSE*ANORMF .AND. NUMIT.LT.ISTEP) GOTO 17
      IF(NUMIT.GE.ISTEP) WRITE(IOUT,43) ISTEP,KSTP,
     1 KPER,ANORM,ANORMF,NUMOUT
   43 FORMAT(1X,'***WARNING*** CONVERGENCE NOT MET AFTER',I5/
     1 INNER ITERATIONS, AT TIME STEP',I5,' OF STRESS PERIOD',I5/,
     2 1X,' IN Y. FINAL NORM OF RESIDUAL CALCULATED TO BE',F10.6/,
     3 1X,' IN Y. FINAL NORM OF TRUE ERROR CALCULATED TO BE',F10.6/,
     4 1X,'OUTER ITERATION NUMBER',I5)
C
C10-----LOOP THROUGH ALL THE COMPONENT DIRECTIONS AGAIN UNTIL
C       CONVERGENCE
 3000 DO 81 K=1,NLAY
      DO 81 I=1,NROW
      DO 81 J=1,NCOL
      ERR=AMAX1(ABS(USLX(J,I,K)-TEMPX(J,I,K)),ABS(USLY(J,I,K)-
     1TEMPY(J,I,K)),ABS(USLZ(J,I,K)-TEMPZ(J,I,K)))
      IF(ERR.GT.TDIFF) TDIFF=ERR
   81 CONTINUE
      IF(TDOLD.LT.TDIFF) GOTO 4000
      TDOLD=TDIFF
      WRITE(IOUT,85)TDIFF
   85 FORMAT(1X,'TOTAL DIFFERENCE IS',E15.8)
      IF(TDIFF.GT.TCLOSE .AND. NUMOUT.LT.IOSTP) GOTO 100
      IF(TDIFF.LE.TCLOSE) GOTO 4000
      IF(TDIFF.GT.TCLOSE .AND. NUMOUT.EQ.IOSTP) WRITE(IOUT,82) NUMOUT,
     1 KSTP,KPER
   82 FORMAT(1X,'***WARNING, OUTER LOOP CONVERGENCE NOT MET
     1 AFTER',I5,'OUTER ITERATIONS AT TIME STEP',I5,' OF STRESS PERIOD',I5)
      GOTO 4000
C
C11-----PRINT NUMBER OF OUTER ITERATIONS NEEDED FOR CONVERGENCE
 4000 WRITE(IOUT,83) NUMOUT
```

```
   83 FORMAT(1X,'NUMBER OF OUTER ITERATIONS =',I5,)
C
C12------MOVE DOUBLE PRECISION VALUES TO SINGLE FOR PLOTING
      DO 19 K=1,NLAY
      DO 19 I=1,NROW
      DO 19 J=1,NCOL
      UX(J,I,K)=USLX(J,I,K)
      UY(J,I,K)=USLY(J,I,K)
      UZ(J,I,K)=USLZ(J,I,K)
   19 CONTINUE
C
C13------RETURN
      RETURN
      END
```

List of Variables for Module USL1FM

Variable	Range	Definition
ACOEF	Module	Coefficient of the cell to the left or above in principal direction.
ANORM	Module	True error of residual based on L1-type norm.
ANORMF	Module	L1 norm of source terms.
BCOEF	Module	Coefficient of the cell of concern in the principal direction.
BOT	Global	DIMENSION (NCOL,NROW,NLAY), Elevation of bottom of each layer. (NBOT is the number of layers for which LAYCON = 1 or 3.)
CCOEF	Module	Coefficient of the cell to right or below in principal direction.
COEF	Module	Sum of source terms and constants of cell being evaluated.
CON	Module	Multiplication factor (not used for this governing equation).
DCB	Module	Grid spacing component of coefficient of cell to left for columns along rows or layers.
DCC	Module	Grid spacing component of coefficient of cell for columns along rows or layers.
DCF	Module	Grid spacing component of coefficient of cell to right for columns along rows or layers.
DCM	Module	Grid spacing component of coefficient of cell to left along columns.
DCOEF	Module	Coefficient of USLY(J,I+1,K+1), USLY(J+1,I+1,K), and USLX(J+1,I+1,K).
DCP	Module	Grid spacing component of coefficient of cell to right along columns.
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains width of row I.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains width of column J.
DELT	Global	Length of current time step.
DRB	Module	Grid spacing component of coefficient of cell to left for rows along columns or layers.
DRC	Module	Grid spacing component of coefficient of cell for rows along columns or layers.
DRF	Module	Grid spacing component of coefficient of cell to right for rows along columns or layers.
DRM	Module	Grid spacing component of coefficient of cell to left along rows.
DRP	Module	Grid spacing component of coefficient of cell to right along rows.
DV	Module	Grid spacing component of coefficient of cell along layers.
DVB	Module	Grid spacing component of coefficient of cell above for layers along rows or columns.
DVC	Module	Grid spacing component of coefficient of cell for layers along rows or columns.

List of Variables for Module USLIFM (Continued)

Variable	Range	Definition
DVM	Module	Grid spacing component of coefficient of cell below for layers along rows or columns.
D2COEF	Module	Coefficient of USLX(J+1,I,K+1).Coefficient of USLY(J,I,K+1),
ECOEF	Module	USLY(J+1,I,K), and USLX(J,I+1,K).
ERR	Module	Error of cell measured for each outer iteration.
E2COEF	Module	Coefficient of USLX(J,I,K+1).
FAC	Module	Constant equal to correct hydraulic conductivity times DELT divided by specific storage times CON.
FACH	Module	Same as FAC but specific storage is for horizontal direction.
FACV	Module	Same as FAC but specific storage is for vertical direction.
FAC1	Module	1+FAC*BCOEF
FCOEF	Module	Coefficient of USLY(J,I-1,K+1), USLY(J+1,I-1,K), USLX(J-1,I+1,K).
F2COEF	Module	Coefficient of USLX(J-1,I,K+1).
GCOEF	Module	Coefficient of USLY(J,I+1,K-1), USLY(J-1,I+1,K), USLX(J+1,I,K).
G2COEF	Module	Coefficient of USLX(J+1,I,K-1).
H	Module	Temporary value of horizontal hydraulic conductivity.
HC	Package	DIMENSION (NCOL,NROW,NLAY), Horizontal hydraulic conductivity evaluated for each cell in the grid.
HCOEF	Module	Coefficient of USLY(J,I,K-1), USLY(J-1,I,K), and USLX(J,I-1,K).
HNEW	Global	DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell.
HOLD	Global	DIMENSION (NCOL,NROW,NLAY), Head at the start of the current time step.
HV	Module	Vertical hydraulic conductivity equal to HC*RATIO.
HY	Global	DIMENSION(NCOL,NROW,NLAY), Horizontal hydraulic conductivity for cells where LAYCON = 1 or 3.
H2COEF	Module	Coefficient of USLX(J,I,K-1).
I	Module	Index for rows.
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array identifying active cells in which displacement is calculated.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell <0, constant-head cell =0, inactive cell >0, variable-head cell
IJSW	Module	Counter in column direction for odd-even ordering.
IKSW	Module	Counter in column direction for odd-even ordering.
IOSTP	Module	Number of outer-loop iterations.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPASS	Module	Index for odd-even ordering.
ISW	Module	Index in row direction for odd-even ordering
J	Module	Index for columns.
JKSW	Module	Counter in row direction for odd-even ordering.
JSW	Module	Index for columns for odd-even ordering.

List of Variables for Module USLIFM (Continued)

Variable	Range	Definition
K	Module	Index for layers.
KB	Module	Counter for layers for which bottom elevation is needed.
KN	Module	Counter for layers that have gone dry for USLZ adjustment.
KPER	Global	Stress period counter.
KR	Module	Counter for layers for which hydraulic conductivity is needed.
KSTP	Global	Time step counter. Reset at the start of each stress period.
KSW	Module	Index for layers for odd-even ordering.
KT	Module	Counter for layers for which top elevation is needed.
KUNIT	Package	Flag indicating whether english or metric units are used for length. =0 Meters are used as units of length. =1 Feet are used as units of length.
LAYCON	Global	DIMENSION (80), Layer type code: 0 - Layer strictly confined. 1 - Layer strictly unconfined. 2 - Layer confined/unconfined (transmissivity is constant). 3 - Layer confined/unconfined (transmissivity varies).
NBULK	Global	Flag indicating whether initial or ultimate bulk fluxes are used.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NMAX	Package	Number of cells in either the row, column or layer direction--whichever is greatest.
NROW	Global	Number of rows in the grid.
NUMIT	Module	Counter for inner-loop iterations.
NUMOUT	Module	Counter for outer-loop iterations.
OCOEF	Module	Coefficient of USLY(J,I,K-1), USLY(J-1,I-1,K), USLX(J-1,I-1,K).
OMEGA	Package	Relaxation parameter for successive overrelaxation.
O2COEF	Module	Coefficient of USLX(J-1,I,K-1).
PCOEF	Module	Coefficient of USLY(J,I+1,K) and USLX(J+1,I,K).
PS	Package	DIMENSION (NCOL,NROW,NLAY), Preconsolidation strain. Used to determine whether specific storage is elastic or virgin.
P2COEF	Module	Coefficient of USLX(J+1,I,K).
QBX	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in X direction.
QBY	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in Y direction.
QBZ	Global	DIMENSION (NCOL,NROW,NLAY), Bulk flux in Z direction.
QCOEF	Module	Coefficient of USLY(J,I+1,K) and USLX(J+1,I,K).
Q2COEF	Module	Coefficient of USLX(J,I,K).
RATIO	Global	DIMENSION (NCOL,NROW,NLAY), Ratio of vertical to horizontal hydraulic conductivity.
RCOEF	Module	Coefficient of USLY(J,I-1,K) and USLX(J-1,I,K).
RESID	Module	Residual which defines the error at the cell during inner-loop iteration.

List of Variables for Module USL1FM (Continued)

Variable	Range	Definition
RJACX	Module	Estimate of the spectral radius of the Jacobi iteration in X.
RJACY	Module	Estimate of the spectral radius of the Jacobi iteration in Y.
RJACZ	Module	Estimate of the spectral radius of the Jacobi iteration in Z.
R2COEF	Module	Coefficient of USLX(J-1,I,K).
SCOEF	Module	Coefficient of USLZ(J+1,I,K+1) and USLZ(J,I+1,K+1).
SC1	Global	DIMENSION (NCOL,NROW,NLAY), Primary storage capacity of each cell (S*DELC*DELR).
SC2	Global	DIMENSION (NCOL,NROW,NLAY), Secondary storage capacity of each cell in the grid.
SPX	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge values in X, equivalent to the ultimate bulk fluxes in X
SPY	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge values in Y, equivalent to the ultimate bulk fluxes in Y
SPZ	Global	DIMENSION (NCOL,NROW,NLAY), Ultimate specific discharge values in Z, equivalent to the ultimate bulk fluxes in Z
SS	Module	Current value of specific storage at the cell being evaluated.
SSE	Package	DIMENSION (NCOL,NROW,NLAY), Elastic specific storage.
SSK	Package	DIMENSION (NCOL,NROW,NLAY), Specific storage in the Z direction
SSV	Package	DIMENSION (NCOL,NROW,NLAY), Virgin specific storage.
SSZ	Module	Temporary specific storage in Z direction.
STRNX	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the X direction.
STRNY	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the Y direction.
STRNZ	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the Z direction.
TCLOSE	Package	Closure criterion for outer-loop convergence.
TDIFF	Module	Total error for outer-loop iteration at current time step.
TDOLD	Module	Total error for outer-loop iteration at previous time step.
TEMPX	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in X direction at previous outer iteration.
TEMPY	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in Y direction at previous outer iteration.
TEMPZ	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in Z direction at previous outer iteration.
TOTIM	Global	Total simulation time.
TRAN	Package	DIMENSION (NCOL,NROW,NLAY), Transmissivity.
UCOEF	Module	Coefficient of USLZ(J,I+1,K) and USLZ(J+1,I,K).
UOLDX	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in X direction at previous time step.
UOLDY	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in Y direction at previous time step.
UOLDZ	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in Z direction at previous time step.
USLX	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in X

List of Variables for Module USLIFM (Continued)

Variable	Range	Definition
USLY	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in Y
USLZ	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in Z
UX	Package	DIMENSION (NCOL,NROW,NLAY) Single precision displacement in X direction used for plotting.
UY	Package	DIMENSION (NCOL,NROW,NLAY) Single precision displacement in Y direction used for plotting.
UZ	Package	DIMENSION (NCOL,NROW,NLAY) Single precision displacement in Z direction used for plotting.
VCOEF	Module	Coefficient of USLZ(J,I+1,K-1) and USLZ(J+1,I,K-1).
VSTRN	Package	DIMENSION (NCOL,NROW,NLAY), Volume strain.
XCLOSE	Package	Closure criterion for inner-loop convergence.
XCOEF	Module	Coefficient of USLZ(J+1,I,K) and USLZ(J-1,I,K).
XCON	Module	Contribution of cross-product derivatives of USLX for displacement in the Y or Z directions.
X2COEF	Module	Coefficient of USLZ(J,I,K+1), and USLZ(J,I,K-1).
YCOEF	Module	Coefficient of USLZ(J,I-1,K+1) and USLZ(J-1,I,K+1).
YCON	Module	Contribution of cross-product derivatives of USLY for displacement in the X or Z directions.
Y2COEF	Module	Coefficient of USLZ(J,I,K).
ZADJ	Module	Amount of displacement in the Z direction added to previously active cells.
ZCOEF	Module	Coefficient of USLZ(J-1,I,K+1) and USLZ(J,I+1,K+1).
ZCON	Module	Contribution of cross-product derivatives of USLZ for displacement in the X or Y directions.
ZIM	Package	DIMENSION (NCOL,NROW), Displacement in the layer of image cells above the water table.
ZOM	Package	DIMENSION (NCOL,NROW), Displacement in the layer of image cells above the water table at the previous time step.
Z2COEF	Module	Coefficient of USLZ(J+1,I,K).

Narrative for Module USL1OT

This module prints or saves magnitude of displacement, directional components of displacement, and volume strain according to flags set by the user for each stress period. If flag IUSLOC is set in Module USL1AL then flags are read for printing or saving these data. If the flags are greater than zero the data are saved for the stress period. The data are printed or the data saved to disk according to flags set in Module USL1AL. Each directional component of displacement has its own flag for printing or saving so that the user does not have to print out all component directions if this should be undesirable.

Module USL1OT is called each stress period by the MAIN program and calls utility modules ULAPRS and ULASAV (see McDonald and Harbaugh, 1988). Module USL1OT performs its functions in the following order:

1. Initialize print and save flags
2. Read print and save flags if IUSLOC is set.
3. Print magnitude of displacement if NMAGPR > 0.
   Save magnitude of displacement if NMAGSV > 0.
4. Print X-displacements if NUSXPR >0.
   Save X-displacements if NUSXSV > 0.
5. Print Y-displacements if NUSYPR > 0.
   Save Y-displacements if NUSYSV > 0.
6. Print Z-displacements if NUSZPR > 0.
   Save Z-displacements if NUSZSV > 0.
7. Print volume strain if NVSTPR > 0.
   Save volume strain if NVSTSV > 0.
8. RETURN

Flow Chart for Module USLIOT

IUSLOC is the flag to determine whether displacements are calculated and whether displacements and volume strain are printed or saved.

NMAGFM and NMAGSV are the flags indicating whether the magnitude of displacement is printed or saved for the stress period, respectively.

>0 values are printed or saved
<=0 values are not printed or saved.

NUSXPR and NUSXSV are the flags indicating whether the X-displacements are printed or saved for the stress period, respectively.

>0 values are printed or saved
<=0 values are not printed or saved.

NUSYPR and NUSYSV are the flags indicating whether the Y-displacements are printed or saved for the stress period, respectively.

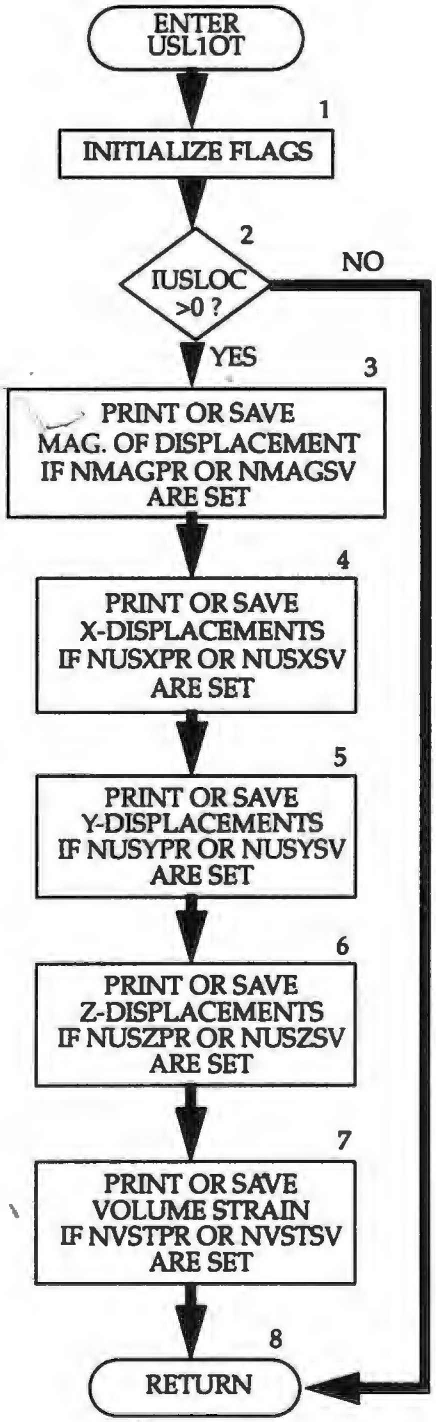
>0 values are printed or saved
<=0 values are not printed or saved.

NUSZPR and NUSZSV are flags indicating whether the Z-displacements are printed or saved for the stress period, respectively.

>0 values are printed or saved.
<=0 values are not printed or saved.

NVSTPR and NVSTSV are flags indicating whether volume strain is printed or saved for the stress period.

>0 values are printed or saved.
<=0 values are not printed or saved.



```
      SUBROUTINE USL1OT(USLX,USLY,USLZ,UMAG,NCOL,NROW,NLAY,IOUT,IN,
     1  IUSLOC,IACT,NSTRT,KPER,KSTP,NMAGFM,NUSXFM,NUSYFM,NUSZFM,
     2  NVSTFM,NMAGUN,NUSXUN,NUSYUN,NUSZUN,NVSTUN,PERTIM,TOTIM,
     3  VSTRN,BUFF)
C
C     ******************************************************************************
C     PRINT AND RECORD DISPLACEMENT OF SOLIDS INFORMATION
C     ******************************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------------------
      CHARACTER*4 TEXT
      DOUBLE PRECISION USLX,USLY,USLZ
C
      DIMENSION USLX(NCOL,NROW,NLAY),USLY(NCOL,NROW,NLAY),
     1  USLZ(NCOL,NROW,NLAY),UMAG(NCOL,NROW,NLAY),IACT(NCOL,
     2  NROW,NLAY),TEXT(4,5),VSTRN(NCOL,NROW,NLAY),BUFF(NCOL,
     3  NROW,NLAY)
C
      DATA TEXT(1,1),TEXT(2,1),TEXT(3,1),TEXT(4,1)
     1  /' DIS','P. M','AGNI','TUDE'/,TEXT(1,2),TEXT(2,2),TEXT(3,2),
     2  TEXT(4,2) /'  X-','DISP','LACE','MENT'/,TEXT(1,3),TEXT(2,3),
     3  TEXT(3,3),TEXT(4,3) /'  Y-','DISP','LACE','MENT'/,TEXT(1,4),
     4  TEXT(2,4),TEXT(3,4),TEXT(4,4) /'  Z-','DISP','LACE','MENT'/,
     5  TEXT(1,5),TEXT(2,5),TEXT(3,5),TEXT(4,5) /'   V','OLUM','E ST',
     6  'RAIN'/
C     ------------------------------------------------------------------------------
C
C1-----INITIALIZE FLAGS FOR PRINTING AND SAVING SUBSIDENCE, MAGNITUDE
C      OF DISPLACEMENT,  AND DIRECTIONAL COMPONENTS OF
C      DISPLACEMENT.
      NMAGPR=0
      NUSXPR=0
      NUSYPR=0
      NUSZPR=0
      NVSTPR=0
      NMAGSV=0
```

```
      NUSXSV=0
      NUSYSV=0
      NUSZSV=0
      NVSTSV=0
C
C2-----READ FLAGS FOR PRINTING AND SAVING IF IUSLOC IS SET
      IF(IUSLOC.LE.0) GOTO 170
      READ(IN,3) NMAGPR,NUSXPR,NUSYPR,NUSZPR,NVSTPR,NMAGSV,
     1 NUSXSV,NUSYSV,NUSZSV,NVSTSV
    3  FORMAT(10I5)
      WRITE(IOUT,6) NMAGPR,NUSXPR,NUSYPR,NUSZPR,NVSTPR,
     1 NMAGSV,NUSXSV,NUSYSV,NUSZSV,NVSTSV
    6  FORMAT(/,1X,'FLAGS FOR PRINTING AND STORING MAGNITUDE OF
     1DISPLACEMENT, AND DIRECTIONAL COMPONENTS OF DISPLACEMENT:'
     2 /' NMAGPR   NUSXPR   NUSYPR   NUSZPR   NVSTPR   NMAGSV   NUSXSV
     3NUSYSV   NUSZSV   NVSTSV'/
     4'--------------------------------------------------------------------
     5---------------------'/,I5,9I9)
C
C3-----PRINT MAGNITUDE OF DISPLACEMENT FOR ALL LAYERS
      IF(NMAGPR.LE.0) GOTO 80
      CALL UBCUSL(USLX,USLY,USLZ,NCOL,NROW,NLAY,IACT,UMAG,IUSLOC)
      DO 8 K=1,NLAY
      IF(NMAGFM.LT.0) CALL ULAPRS(UMAG(1,1,K),TEXT(1,1),KSTP,KPER,
     1 NCOL,NROW,K,-NMAGFM,IOUT)
      IF(NMAGFM.GE.0) CALL ULAPRW(UMAG(1,1,K),TEXT(1,1),KSTP,KPER,
     1 NCOL,NROW,K,NMAGFM,IOUT)
    8 CONTINUE
C
C------SAVE MAGNITUDE OF DISPLACEMENT FOR ALL LAYERS
   80 IF(NMAGSV.LE.0) GOTO 90
      CALL UBCUSL(USLX,USLY,USLZ,NCOL,NROW,NLAY,IACT,UMAG,IUSLOC)
      DO 9 K=1,NLAY
      CALL ULASAV(UMAG(1,1,K),TEXT(1,1),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1 NROW,K,NMAGUN)
    9 CONTINUE
C
```

```
C4-----PRINT X COMPONENT OF DISPLACEMENT FOR ALL LAYERS
   90 IF(NUSXPR.LE.0) GOTO 100
      DO 20 K=1,NLAY
      DO 20 I=1,NROW
      DO 20 J=1,NCOL
      UX=USLX(J,I,K)
      BUFF(J,I,K)=UX
   20 CONTINUE
      DO 11 K=1,NLAY
      IF(NUSXFM.LT.0) CALL ULAPRS(BUFF(1,1,K),TEXT(1,2),KSTP,KPER,
     1 NCOL,NROW,K,-NUSXFM,IOUT)
      IF(NUSXFM.GE.0) CALL ULAPRW(BUFF(1,1,K),TEXT(1,2),KSTP,KPER,
     1 NCOL,NROW,K,NUSXFM,IOUT)
   11 CONTINUE
C
C-------SAVE X COMPONENT OF DISPLACEMENT FOR ALL LAYERS
  100 IF(NUSXSV.LE.0) GOTO 110
      DO 21 K=1,NLAY
      DO 21 I=1,NROW
      DO 21 J=1,NCOL
      UX=USLX(J,I,K)
      BUFF(J,I,K)=UX
   21 CONTINUE
      DO 12 K=1,NLAY
      CALL ULASAV(BUFF(1,1,K),TEXT(1,2),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1 NROW,K,NUSXUN)
   12 CONTINUE
C
C5------PRINT Y COMPONENT OF DISPLACEMENT FOR ALL LAYERS
  110 IF(NUSYPR.LE.0) GOTO 120
      DO 22 K=1,NLAY
      DO 22 I=1,NROW
      DO 22 J=1,NCOL
      UY=USLY(J,I,K)
      BUFF(J,I,K)=UY
   22 CONTINUE
      DO 13 K=1,NLAY
```

```
      IF(NUSYFM.LT.0) CALL ULAPRS(BUFF(1,1,K),TEXT(1,3),KSTP,KPER,
     1  NCOL,NROW,K,-NUSYFM,IOUT)
      IF(NUSYFM.GE.0) CALL ULAPRW(BUFF(1,1,K),TEXT(1,3),KSTP,KPER,
     1  NCOL,NROW,K,NUSYFM,IOUT)
   13 CONTINUE
C
C------SAVE Y COMPONENT OF DISPLACEMENT FOR ALL LAYERS
  120 IF(NUSYSV.LE.0) GOTO 130
      DO 23 K=1,NLAY
      DO 23 I=1,NROW
      DO 23 J=1,NCOL
      UY=USLY(J,I,K)
      BUFF(J,I,K)=UY
   23 CONTINUE
      DO 14 K=1,NLAY
      CALL ULASAV(BUFF(1,1,K),TEXT(1,3),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1  NROW,K,NUSYUN)
   14 CONTINUE

C
C6------PRINT Z COMPONENT OF DISPLACEMENT FOR ALL LAYERS
  130 IF(NUSZPR.LE.0) GOTO 140
      DO 24 K=1,NLAY
      DO 24 I=1,NROW
      DO 24 J=1,NCOL
      UZ=USLZ(J,I,K)
      BUFF(J,I,K)=UZ
   24 CONTINUE
      DO 15 K=1,NLAY
      IF(NUSZFM.LT.0) CALL ULAPRS(BUFF(1,1,K),TEXT(1,4),KSTP,KPER,
     1  NCOL,NROW,K,-NUSZFM,IOUT)
      IF(NUSZFM.GE.0) CALL ULAPRW(BUFF(1,1,K),TEXT(1,4),KSTP,KPER,
     1  NCOL,NROW,K,NUSZFM,IOUT)
   15 CONTINUE
C
C------SAVE Z COMPONENT OF DISPLACEMENT FOR ALL LAYERS
  140 IF(NUSZSV.LE.0) GOTO 150
```

```
      DO 25 K=1,NLAY
      DO 25 I=1,NROW
      DO 25 J=1,NCOL
      UZ=USLZ(J,I,K)
      BUFF(J,I,K)=UZ
   25 CONTINUE
      DO 16 K=1,NLAY
      CALL ULASAV(BUFF(1,1,K),TEXT(1,4),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1  NROW,K,NUSZUN)
   16 CONTINUE
C
C7------PRINT VOLUME STRAIN FOR ALL LAYERS
  150 IF(NVSTPR.LE.0) GOTO 160
      DO 17 K=1,NLAY
      IF(NVSTFM.LT.0) CALL ULAPRS(VSTRN(1,1,K),TEXT(1,5),KSTP,KPER,
     1  NCOL,NROW,K,-NVSTFM,IOUT)
      IF(NVSTFM.GE.0) CALL ULAPRW(VSTRN(1,1,K),TEXT(1,5),KSTP,KPER,
     1  NCOL,NROW,K,NVSTFM,IOUT)
   17 CONTINUE
C
C------SAVE VOLUME STRAIN FOR ALL LAYERS
  160 IF(NVSTSV.LE.0) GOTO 170
      DO 18 K=1,NLAY
      CALL ULASAV(VSTRN(1,1,K),TEXT(1,5),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1 NROW,K,NVSTUN)
   18 CONTINUE
C
C8------RETURN
  170 RETURN
      END
```

List of Variables for Module USLIOT

Variable	Range	Definition
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
I	Module	Index for rows
IACT	Package	DIMENSION (NCOL,NROW,NLAY), Boundary array identifying active cells in which displacement is calculated.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IUSLOC	Package	Flag indicating whether displacements are calculated and displacements and volume strain printed or saved.
J	Module	Index for columns
K	Module	Index for layers
KPER	Global	Cotunter for number of stress periods.
KSTP	Global	Counter for number of time steps.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NMAGFM	Package	Code indicating format for printing magnitude of displacement
NMAGPR	Module	Flag indicating whether magnitude of displacement is printed.
NMAGSV	Module	Flag indicating whether magnitude of displacement is saved.
NMAGUN	Package	Unit number indicating where magnitude of displacement data are to be recorded.
NUSXFM	Package	Code indicating format for printing X-displacements.
NUSXPR	Module	Flag indicating whether X-displacements are printed.
NUSXSV	Module	Flag indicating whether X-displacements are saved.
NUSXUN	Package	Unit number indicating where X-displacement data are to be recorded.
NUSYFM	Package	Code indicating format for printing Y-displacements.
NUSYPR	Module	Flag indicating whether Y-displacements are printed.
NUSYSV	Module	Flag indicating whether Y-displacements are saved.
NUSYUN	Package	Unit number indicating where Y-displacement data are to be recorded.
NUSZFM	Package	Code indicating format for printing Z-displacements.
NUSZPR	Module	Flag indicating whether Z-displacements are printed.
NUSZSV	Module	Flag indicating whether Z-displacements are saved.
NUSZUN	Package	Unit number indicating where Z-displacement data are to be recorded.
NVSTFM	Package	Code indicating format for printing volume strain.
NVSTPR	Module	Flag indicating whether volume strain is printed.
NVSTSV	Module	Flag indicating whether volume strain is saved.
NVSTUN	Package	Unit number indicating where volume strain data are to be recorded.
TEXT	Module	Label for printout of input array.
TOTIM	Global	Total simulation time.

List of Variables for Module USLIOT

Variable	Range	Definition
UMAG	Package	DIMENSION (NCOL,NROW,NLAY), Magnitude of displacement.
USLX	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the X direction.
USLY	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the Y direction.
USLZ	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the Z direction.
UX	Module	Temporary variable for displacement in the X direction.
UY	Module	Temporary variable for displacement in the Y direction.
UZ	Module	Temporary variable for displacement in the Z direction.
VSTRN	Package	DIMENSION (NCOL,NROW,NLAY), Volume strain.

Narrative for Module SUSL1X

This module calculates the strain in the X direction. This value is used to calculate the volume strain which is used to determine whether elastic or virgin specific storage is used for a given cell in the model grid.

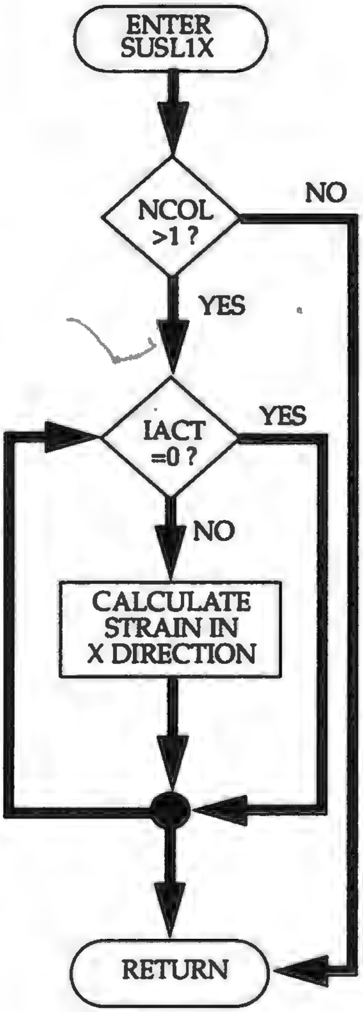
Module SUSL1X is called by USL1FM and performs the following tasks

1. Check to see if NCOL > 1.
2. Check for active cells in the grid
3. Calculate strain in the X direction.
4. RETURN

Flow Chart for Module SUSL1X

NCOL is the number of columns in the grid.

IACT is the boindary array indicating the cells where displacement is calculated.



```
      SUBROUTINE SUSL1X(USLX,IACT,NCOL,I,K,DELR,STRNX,NROW,NLAY)
C
C ****************************************************************
C     CALCULATE THE STRAIN IN THE X DIRECTION
C ****************************************************************
C
C     SPECIFICATIONS:
C-------------------------------------------------------------------------------
      DOUBLE PRECISION USLX
      DIMENSION USLX(NCOL,NROW,NLAY),IACT(NCOL,NROW,NLAY),
     1 DELR(NCOL),STRNX(NCOL,NROW,NLAY)
C -----------------------------------------------------------------------------
C
C1------LOOP THROUGH ACTIVE CELLS FOR EACH COLUMN
      IF(NCOL.GT.1) GOTO 11
      GOTO 23
   11 DO 10 J=1,NCOL
C
C2-------CHECK FOR ACTIVE CELLS
      IF(IACT(J,I,K).EQ.0) GOTO 10
C
C3--------FOR CELLS ALONG A COLUMN CALCULATE STRAIN
C     LOWEST ACTIVE COLUMN
      IF(J.EQ.1 .OR. (J.GT.1 .AND. IACT(J-1,I,K).EQ.0)) THEN
      STRNX(J,I,K)=(-USLX(J+1,I,K)+USLX(J,I,K))/(DELR(J)+0.5*
     1 (DELR(J+1)+DELR(J)))
      GOTO 10
      ENDIF
C     HIGHEST ACTIVE COLUMN
      IF(J.EQ.NCOL .OR. (J.LT.NCOL .AND. IACT(J+1,I,K).EQ.0)) THEN
      STRNX(J,I,K)=(-USLX(J,I,K)-USLX(J-1,I,K))/(DELR(J)+0.5*
     1 (DELR(J)+DELR(J-1)))
      GOTO 10
      ENDIF
C     INTERIOR ACTIVE CELLS ALONG A COLUMN
      IF(IACT(J-1,I,K).GT.0 .AND. IACT(J+1,I,K).GT.0) THEN
      STRNX(J,I,K)=(USLX(J+1,I,K)-USLX(J-1,I,K))/
```

```
     1 (DELR(J)+0.5*(DELR(J+1)+DELR(J-1)))
       ENDIF
  10   CONTINUE
C
C4-----RETURN
  23   RETURN
       END
```

List of Variables for Module SUSL1X

Variable	Range	Definition
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
I	Module	Index for rows.
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array for displacement. >0 cell is active. <=0 cell is inactive.
J	Module	Index for columns.
K	Module	Index for layers
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
STRNX	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the X direction.
USLX	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the X direction.

Narrative for Module SUSL1Y

This module calculates the strain in the Y direction. This value is used to calculate the volume strain which is used to determine whether elastic or virgin specific storage is used for a given cell in the model grid.

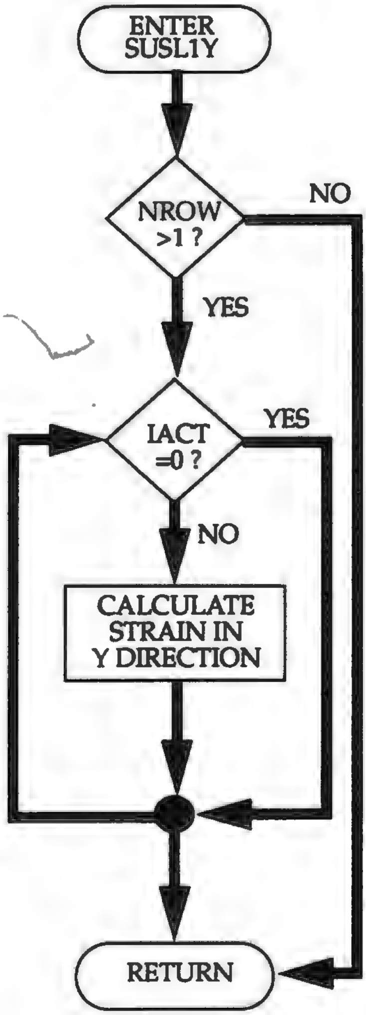
Module SUSL1Y is called by USL1FM and performs the following tasks

1. Check to see if NROW>1.
2. Check for active cells in the grid
3. Calculate strain in the Y direction.
4. RETURN

Flow Chart for Module SUSL1Y

NROW is the number of rows in the grid.

IACT is the boundary array indicating the cells where displacement is calculated.



```
      SUBROUTINE SUSLY(USLY,IACT,J,NROW,K,DELC,STRNY,NCOL,NLAY)
C
C   ***************************************************************
C   CALCULATE THE STRAIN IN THE Y DIRECTION
C   ***************************************************************
C
C   SPECEFICATIONS:
C   ----------------------------------------------------------------
      DOUBLE PRECISION USLY
      DIMENSION USLY(NCOL,NROW,NLAY),IACT(NCOL,NROW,NLAY),
     1 DELC(NROW),STRNY(NCOL,NROW,NLAY)
C   ----------------------------------------------------------------
C
C1------LOOP THROUGH ACTIVE CELLS FOR EACH ROW
   23 IF(NROW.GT.1)GOTO 22
      GOTO 33
   22 DO 20 I=1,NROW
C
C2------CHECK FOR ACTIVE CELLS
      IF(IACT(J,I,K).EQ.0) GOTO 20
C
C3------FOR CELLS ALONG A ROW CALCULATE STRAIN
C   LOWEST ACTIVE ROW
      IF(I.EQ.1 .OR. (I.GT.1 .AND. IACT(J,M,K).EQ.0)) THEN
      STRNY(J,I,K)=(USLY(J,I+1,K)-USLY(J,I,K))/(DELC(I)+0.5*
     1  (DELC(I+1)+DELC(I)))
      GOTO 20
      ENDIF
C   HIGHEST ACTIVE ROW
       IF(I.EQ.NROW .OR. (I.LT.NROW .AND. IACT(J,I+1,K).EQ.0)) THEN
      STRNY(J,I,K)=(-USLY(J,I,K)-USLY(J,I-1,K))/(DELC(I)+0.5*
     1  (DELC(I)+DELC(I-1)))
      GOTO 20
      ENDIF
C   INTERIOR ACTIVE CELLS ALONG A ROW
      IF(IACT(J,I-1,K).GT.0 .AND. IACT(J,I+1,K).GT.0) THEN
      STRNY(J,I,K)=(USLY(J,I+1,K)-USLY(J,I-1,K))/
```

```
     1 (DELC(I)+0.5*(DELC(I+1)+DELC(I-1)))
       ENDIF
    20 CONTINUE
C
C4------RETURN
    33 RETURN
       END
```

List of Variables for Module SUSL1Y

Variable	Range	Definition
DELC	Global	DIMENSION (NCOL), Cell dimension in the column direction. DELC(I) contains the width of row I.
I	Module	Index for rows.
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array for displacement. >0 cell is active. <=0 cell is inactive.
J	Module	Index for columns.
K	Module	Index for layers
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
STRNY	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the Y direction.
USLY	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the Y direction.

Narrative for Module SUSL1Z

This module calculates the strain in the Z direction. This value is used to calculate the volume strain which is used to determine whether elastic or virgin specific storage is used for a given cell in the model grid.

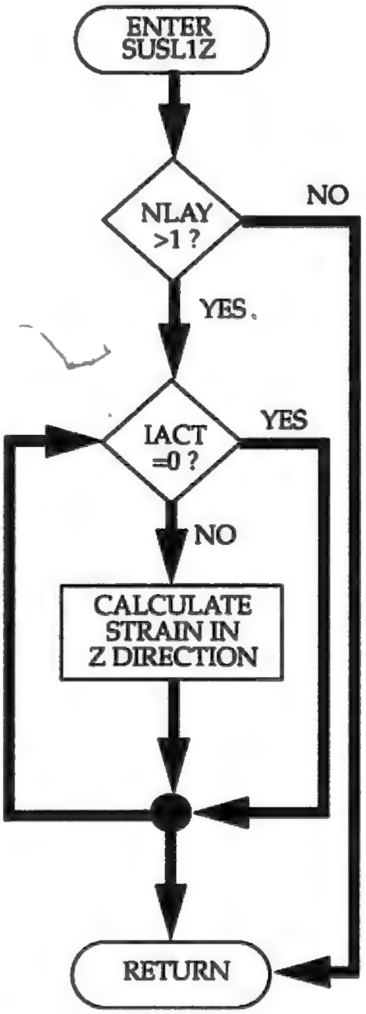
Module SUSL1Z is called by USL1FM and performs the following tasks

1. Check to see if NLAY>1.
2. Check for active cells in the grid
3. Calculate strain in the Z direction.
4. RETURN

Flow Chart for Module SUSL1Z

NLAY is the number of layers in the grid.

IACT is the boundary array indicating
the cells where displacement is calculated.



```
      SUBROUTINE SUSL1Z(USLZ,IACT,J,I,NLAY,DELL,STRNZ,IBOUND,
     1 NCOL,NROW)
C
C    ********************************************************
C    CALCULATE THE STRAIN IN THE Z DIRECTION
C    ********************************************************
C
C    SPECIFICATIONS:
C--------------------------------------------------------------
      DOUBLE PRECISION USLZ
      DIMENSION USLZ(NCOL,NROW,NLAY),IACT(NCOL,NROW,NLAY),
     1DELL(NCOL,NROW,NLAY),STRNZ(NCOL,NROW,NLAY),IBOUND(NCOL,
     2NROW,NLAY)
C    --------------------------------------------------------------
C
C1------LOOP THROUGH ACTIVE CELLS FOR EACH LAYER
   33 IF(NLAY.GT.1) GOTO 32
      GOTO 41
   32 DO 30 K=1,NLAY
C
C2------CHECK FOR ACTIVE CELLS
      IF(IACT(J,I,K).EQ.0) GOTO 30
      IF(DELL(J,I,K).EQ.0) GOTO 30
C
C3------FOR CELLS THROUGH A LAYER AND CALCULATE STRAIN
C    UPPERMOST LAYER
      IF(K.EQ.1 .OR. (K.GT.1 .AND. IBOUND(J,I,K-1).EQ.0)) THEN
      STRNZ(J,I,K)=(USLZ(J,I,K+1)-USLZ(J,I,K))/
     1 (0.5*DELL(J,I,K+1)+DELL(J,I,K))
      GOTO 30
      ENDIF
C    HIGHEST ACTIVE LAYER
      IF(K.EQ.NLAY .OR. (K.LT.NLAY .AND. IACT(J,I,K+1).EQ.0)) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      STRNZ(J,I,K)=(USLZ(J,I,K)-USLZ(J,I,K-1))/(DELL(J,I,K)+0.5*
     1 DELL(J,I,K)+DELL(J,I,K-1))
      ELSE
```

```
      STRNZ(J,I,K)=(-USLZ(J,I,K)-USLZ(J,I,K-1))/(DELL(J,I,K)+0.5*
     1  (DELL(J,I,K)+DELL(J,I,K-1)))
      ENDIF
      GOTO 30
      ENDIF
C    INTERIOR ACTIVE CELLS THROUGH ALL LAYERS
      IF(IACT(J,I,K-1).GT.0 .AND. IACT(J,I,K+1).GT.0) THEN
      IF(K-1.EQ.1 .OR. (K-1.GT.1 .AND. IBOUND(J,I,K-2).EQ.0)) THEN
      STRNZ(J,I,K)=(USLZ(J,I,K+1)-USLZ(J,I,K-1))/
     1(DELL(J,I,K)+0.5*DELL(J,I,K+1)+DELL(J,I,K-1))
      ELSE
      STRNZ(J,I,K)=(USLZ(J,I,K+1)-USLZ(J,I,K-1))/
     1  (DELL(J,I,K)+0.5*(DELL(J,I,K+1)+DELL(J,I,K-1)))
      ENDIF
      ENDIF
   30 CONTINUE
C
C4------RETURN
   41 RETURN
      END
         s
```

List of Variables for Module SUSLIX

Variable	Range	Definition
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction. DELL(J,I,K) contains the thickness of layer K.
I	Module	Index for rows.
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array for displacement. >0 cell is active. <=0 cell is inactive.
IBOUND	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell <0 cell is constant head =0 cell is inactive >0 cell is variable head
J	Module	Index for columns.
K	Module	Index for layers
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
STRNZ	Package	DIMENSION (NCOL,NROW,NLAY), Strain in the Z direction.
USLZ	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the Z direction.

Narrative for Utility Module UBCUSL

This utility module calculates the magnitude of displacement from the already calculated directional components of displacement if IUSLOC > 0. This utility module is called by USL1OT and performs its tasks in the following order:

1. Check to see if IUSLOC > 0
2. Loop through all cells in the grid checking to see if they are active
3. Calculate magnitude of displacement from USLX, USLY, and USLZ values.
4. RETURN.

Flow Chart for Utility Module UBCUSL

IUSLOC is the flag indicating whether displacement is calculated or not.
 >0 displacement is calculated and printed or saved.
 <=0 displacement is not calculated, printed or saved.

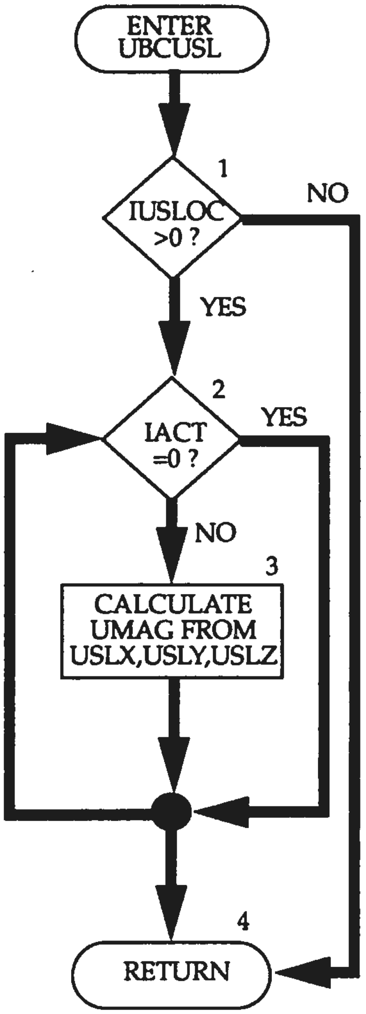
IACT is the boundary array flag for calculating displacement
 >0 cell is active
 <=0 cell is inactive

USLX is the displacement in the X direction.

USLY is the displacement in the Y direction.

USLZ is the displacement in the Z direction.

UMAG is the magnitude of displacement calculated as the square root of the sum of squares of USLX, USLY, and USLZ.



```
      SUBROUTINE UBCUSL(USLX,USLY,USLZ,NCOL,NROW,NLAY,
     1 IACT,UMAG,IUSLOC)
C  ************************************************************************
C  UTILITY SUBROUTINE TO CALCULATE THE MAGNITUDE OF
C  DISPLACEMENT WHEN IUSLOC FLAG IS SET
C  ************************************************************************
C
C  SPECIFICATIONS:
C  ------------------------------------------------------------------------
     DOUBLE PRECISION USLX,USLY,USLZ
C
      DIMENSION USLX(NCOL,NROW,NLAY),USLY(NCOL,NROW,NLAY),
     1 USLZ(NCOL,NROW,NLAY),IACT(NCOL,NROW,NLAY),
     2 UMAG(NCOL,NROW,NLAY)
C  ------------------------------------------------------------------------
C
C1------CALUCATE THE MAGNITUDE OF DISPLACEMENT IF
C        IUSLOC IS GREATER THEN 0.
      IF(IUSLOC.LE.0) GOTO 100
C
C2------LOOP THROUGH ENTIRE GRID OF ACTIVE CELLS
      DO 8 K=1,NLAY
      DO 8 I=1,NROW
      DO 8 J=1,NCOL
      IF(IACT(J,I,K).EQ.0) GOTO 8
C
C3------CALCULATE MAGNITUDE OF DISPLACEMENT
      UMAG(J,I,K)=SQRT(USLX(J,I,K)*USLX(J,I,K)+USLY(J,I,K)*USLY(J,I,K)+
     1 USLZ(J,I,K)*USLZ(J,I,K))
    8 CONTINUE
C
C4------RETURN
  100 RETURN
      END
```

List of Variables for Utility Module UBCUSL

Variable	Range	Definition
I	Module	Index for rows.
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Boundary array identifying active cells in which displacement is calculated. >0 cell is active <=0 cell is inactive
IUSLCX	Package	Flag indicating whether displacement and volume strain is calculated >0 displacements and volume strains are calculated <=0 displacements and volume strains are not calculated
J	Module	Index for columns.
K	Module	Index for layers.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
UMAG	Package	DIMENSION (NCOL,NROW,NLAY), Magnitude of displacement.
USLX	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the X direction.
USLY	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the Y direction.
USLZ	Package	DIMENSION (NCOL,NROW,NLAY), Displacement in the Z direction.

Narrative for Utility Module U2USLR

This module reads transmissivity values if LAYCON = 0 or 2. Although these values are read in the BCF package of MODFLOW the values are modified within the BCF package and are not usable for the purposes of the displacement model. This utility module functions much like U2DREL of MODFLOW.

This module is read by USL1AL and performs its tasks in the following order:

1. Read array control record.
2. Use LOCAT to see where array values come from.
3. If LOCAT = 0 set all array values equal to CNSTNT.
4. If LOCAT ≥ 0 read formatted records using format FMTIN.
5. If LOCAT ≤ 0 read unformated record containing array values.
6. RETURN

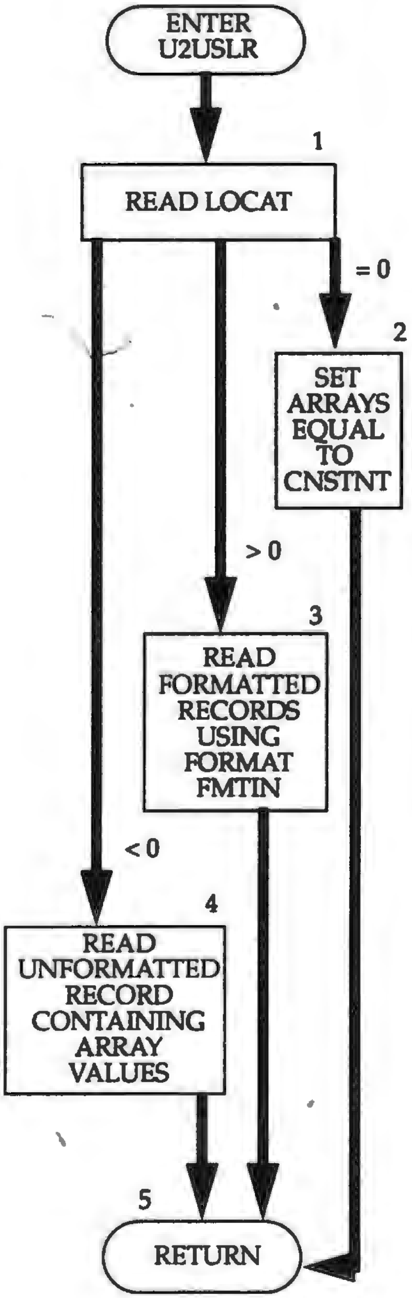
Flow Chart for Utility Module U2USLR

LOCAT indicates the location of the data which will be put in the array.

- ≥0 Represents the unit number from which data values will be read in the format specified in the third field of the array control record (FMTIN).
- =0 Every element in the array will be set equal to the value CNSTNT.
- <0 The sign reversed to give the unit number from which an unformated record will be read.

CNSTNT is the value that each element in the array is set to when LOCAT = 0.

FMTIN is the format used to read the array values.



```
      SUBROUTINE U2USLR(A,II,JJ,K,IN)
C     ********************************************************************
C     THIS SUBROUTINE READS TRANSMISSIVITY DATA THAT CAN NOT
C     BE OBTAINED FROM THE BCF PACKAGE BECAUSE IT IS ALTERED
C     ********************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------
     CHARACTER*16 FMTIN
     DIMENSION A(JJ,II)
C     ------------------------------------------
C
C1------READ ARRAY CONTROL RECORD
      READ(IN,1) LOCAT,CNSTNT,FMTIN
     1 FORMAT(I10,F10.0,A20)
C
C2------USE LOCAT TO SEE WHERE ARRAY VALUES COME FROM
      IF(LOCAT) 200,50,90
C
C3------IF LOCAT=0 THEN SET ALL ARRAY VALUES EQUAL TO CNSTNT, RETURN
   50 DO 80 I=1,II
      DO 80 J=1,JJ
   80 A(J,I)=CNSTNT
      RETURN
C
C3------IF LOCAT>0 THEN READ FORMATTED RECORDS USING FORMAT FMTIN
   90 DO 100 I=1,II
      READ(LOCAT,FMTIN) (A(J,I),J=1,JJ)
  100 CONTINUE
      GOTO 300
C
C4------IF LOCAT<0 THEN READ UNFORMATTED RECORD CONTAINING ARRAY
C       VALUES
  200 LOCAT=-LOCAT
      READ(LOCAT)
      READ(LOCAT) A
  300 IF(CNSTNT.EQ.0.) GOTO 320
```

```
      DO 310 I=1,II
      DO 310 J=1,JJ
      A(J,I)=A(J,I)*CNSTNT
  310 CONTINUE
C
C5--------RETURN
  320 RETURN
      END
```

List of Variables for Utility Module U2USLR

Variable	Range	Definition
A	Module	DIMENSION (NCOL,NROW), Represents the array of values being read. For this module this array represents transmissivities.
CNSTNT	Module	The value that each cell in the grid is set to when LOCAT = 0.
FMTIN	Module	The format that is used to read in the data from the unit specified in IN.
I	Module	Index for rows.
II	Module	Number of rows in array being read.
IN	Package	Primary unit number from which input for this package will be read.
J	Module	Index for columns.
JJ	Module	Number of columns in the array being read.
LOCAT	Module	Indicates the location of the data which will be put in the array. >0 Represents the unit number from which data values will be read in the format specified in the third field of the array-control record (FMTIN). =0 Every element in the array will be set equal to the value CNSTNT. <0 The sign reversed to give the unit number from which an unformated record will be read

Vector Plot Package Input

Input for Vector Plot Package (PLT) is read from the unit specified in IUNIT(16).

FOR EACH SIMULATION

PLT1AL

1. Data:	IPLOTV	IMANY	IDEV	IVEC
Format:	I10	I10	I10	I10

FOR EACH STRESS PERIOD

PLT1RP

2. Data:	ITYPE	LPXY
Format:	I10 ,	I10
	(Item 2 is read IMANY times)	
3. Data:	ISTR	
Format:	40I2	
	(Item 3 is read NPER times)	

Explanation of Fields Used in Input Instructions

IPLOTV--is the flag indicating whether vectors for bulk flux or displacement will be made at the end of each stress period.

If IPLOTV=1 a bulk flux vector plot will be made
If IPLOTV=2 a displacement vector plot will be made
If IPLOTV<1 or >2 no plot will be made.

IMANY--is the number of plots that will be made at the end of each stress period. Determines how many times item 2 will be read.

IDEV--is the device that the vector plots will be plotted to.

If IDEV=1 then the plot will be drawn as an X-WINDOW on the Data General Aviion Workstations.
If IDEV=2 then the plot will be stored as a postscript file and named post# where # is the number of the plot in the order designated by item 2.
If IDEV=3 then the plot will be stored as a CGM META file that can be imported directly into FRAMEMAKER.

(Note: The subroutine that calls the various platforms or file conventions can be readily modified to include the platform or file type needed by the user).

IVEC--is the flag indicating whether vector heads (arrows) will be printed

If IVEC=0 no arrow heads are drawn
If IVEC # 0 vector heads are drawn

ITYPE--is the flag indicating the type of plot drawn

If ITYPE=1 a planimetric plot will be made (x-y plot)
If ITYPE=2 a cross-sectional plot will be made (x-z plot)
If ITYPE=3 a cross-sectional plot will be made (y-z plot)
If ITYPE<1 or >3 no plot will be made.

LPXY--is the row, column, or layer designation through which a plot is drawn. For example, if ITYPE is 1 and LPXY is 3, a planimetric plot of layer three will be drawn. If ITYPE is 2 and LPXY is 25 then a cross sectional plot along plane X-Z through row 25 (Y) will be drawn.

ISTR--is the flag indicating after which stress periods plots are to be made. IMANY plots are made after each stress period when the flag is set. If more than 40 stress periods are used, continue item three on the following line.

If ISTR>0 plots will be made for the stress period indicated
If ISTR<=0 plots will not be made for the stress period indicated.

Module Documentation for the Vector Plotting Package

This plotting package plots vectors at each grid cell location identifying the magnitude and direction of either bulk flux or displacement. The length of the vector tail represents the relative magnitude of bulk flux or displacement.

The Plotting package (PLT) has three primary modules, one submodule, and one function. All the primary modules are called by the MAIN program. This package uses the graphics kernel system (GKS) to plot the data; therefore, numerous calls are made in the PLT1FM subroutine to GKS routines not described in this documentation.

Primary Modules

PLT1AL	Allocates space for data arrays. Reads the type and amount of plots that will be made each stress period.
PLT1RP	Reads all data needed by the package. Prints the type of plots that are made after each stress period.
PLT1FM	Prepares data and graphics for plotting either the bulk flux or displacements along user defined lines of section. Calls SPLT1D submodule and STR_LEN function subprogram.

Submodule

SPLT1D	Makes a CGM meta file, a postscript file, or makes a plot in the X-window environment, of the bulk fluxes or displacements according to user defined parameters.

Function

STR_LEN	Calculates the exact string length of file names or labels.

Narrative for Module PLT1AL

This module allocates space for data arrays for the Plotting package. It also reads the number and type of plots to be made. These are done in the following order.

1. Set up size parameters for array sizes.
2. Read plot type, number of plots per stress period, how plot is to be stored or printed, and whether arrowheads are to be added to vector plots.
3. Allocate storage for the following arrays.
    - XANG_ARR Angle of vector for y-z plot at each cell location.
    - YANG_ARR Angle of vector for x-z plot at each cell location.
    - ZANG_ARR Angle of vector for x-y plot at each cell location.
    - XMAX_ARR Magnitude of vector in y-z plane for each cell location.
    - YMAX_ARR Magnitude of vector in x-z plane for each cell location.
    - ZMAX_ARR Magnitude of vector in x-y plane for each cell location.
    - XCNTR Center of each grid cell in y-z plane.
    - YCNTR Center of each grid cell in x-z plane.
    - ZCNTR Center of each grid cell in x-y plane.
    - LPXY Line of section along which plot is made.
    - PTYPE Type of plot: planimetric or cross section.
    - ISTR Flag indicating whether plots are made after each stress period.
4. Print amount of space used for Plotting package.
5. RETURN

Flow Chart for Module PLT1AL

IPLOTV is the plotting flag.

If IPLOTV=1 bulk fluxes are plotted.
If IPLOTV=2 displacements are plotted.
If IPLOTV<1 or > 2 no plots are made.

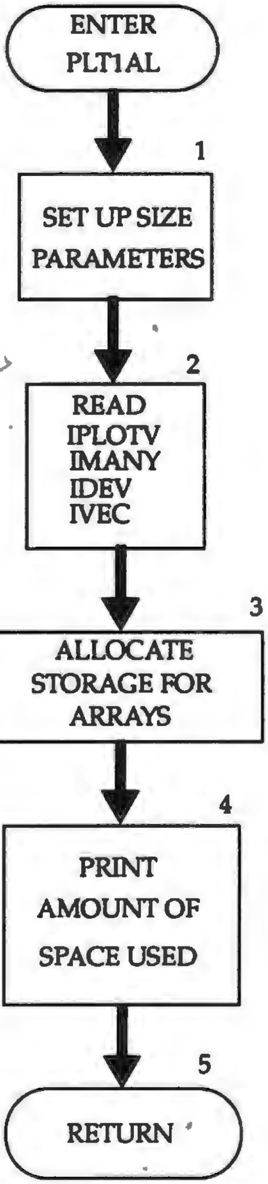
IMANY is the number of plots that are made after each selected stress period.

IDEV is the device or type of file that the plot is written to for plotting or printing.

If IDEV=1 the plot is drawn to an X-window on the DG workstation.
If IDEV=2 the plot is written to a postscript file with the prefix POST followed by the plot number designated by the program.
If IDEV=3 the plot is written to a CGM meta file with the prefix META followed by the plot number designated by the program.

IVEC is a flag indicating whether arrow heads are drawn at the end of the vectors plotted.

If IVEC=0 no arrow heads are drawn.
If IVEC ≠ 0 arrow heads are drawn.



```
    SUBROUTINE PLT1AL(ISUM,LENX,LCXANG,LCYANG,LCZANG,LCXMAX,
   1 LCYMAX,LCZMAX,IPLOTV,IMANY,LCIPXY,LCITYPE,NCOL,NROW,
   2NLAY,IN,IOUT,IDEV,IVEC,LCXCNTR,LCYCNTR,LCZCNTR,LCISTR,NPER)
C
C     *********************************************************
C     ALLOCATE ARRAY STORAGE FOR PLOTTING PACKAGE
C     *********************************************************
C
C     SPECIFICATIONS:
C     ----------------------------------------------
C     ----------------------------------------------
C
C1------SET UP SIZE PARAMETERS
      ISOLD=ISUM
      NRC=NROW*NCOL
      NRL=NROW*NLAY
      NCL=NCOL*NLAY
      NRCL=NCOL*NROW*NLAY
C
C2------READ IN TYPE OF PLOT, HOW MANY PLOTS, AND OUTPUT DEVICE
C      THE OUTPUT DEVICES ARE 1=XWINDOW; 2=POSTSCRIPT; 3=METAFILE;
C       4=EXIT WITHOUT A PLOT AND WHETHER VECTOR ARROWS ARE
C       ADDED.
      READ(IN,7) IPLOTV,IMANY,IDEV,IVEC
    7 FORMAT(4I10)
C
C3------ALLOCATE STORAGE FOR ARRAYS
      LCXANG=ISUM
      ISUM=ISUM+NRL
      LCYANG=ISUM
      ISUM=ISUM+NCL
      LCZANG=ISUM
      ISUM=ISUM+NRC
      LCXMAX=ISUM
      ISUM=ISUM+NRCL
      LCYMAX=ISUM
      ISUM=ISUM+NRCL
```

```
      LCZMAX=ISUM
      ISUM=ISUM+NRCL
      LCLPXY=ISUM
      ISUM=ISUM+IMANY
      LCITYPE=ISUM
      ISUM=ISUM+IMANY
      ISP=ISUM-ISOLD
      LCXCNTR=ISUM
      ISUM=ISUM+NCOL
      LCYCNTR=ISUM
      ISUM=ISUM+NROW
      LCZCNTR=ISUM
      ISUM=ISUM+NLAY
      LCISTR=ISUM
      ISUM=ISUM+NPER
C
C4------PRINT AMOUNT OF SPACE USED
      ISP=ISUM-ISOLD
      WRITE(IOUT,2) ISP
    2 FORMAT(1X,I8,' ELEMENTS IN X ARRAY ARE USED BY PLT')
      ISUM1=ISUM-1
      WRITE(IOUT,3) ISUM1,LENX
    3 FORMAT(1X,I8,' ELEMENTS OF X ARRAY USED OUT OF',I8)
      IF(ISUM1.GT.LENX) WRITE(IOUT,4)
    4 FORMAT(1X,'***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module PLT1AL

Variable	Range	Definition
IDEV	Package	Flag indicating device that the vector plots will be plotted to. =1 plot will be displayed in an X-window on the DG. =2 plot will be stored as a postscript file. =3 plot will be stored as a CGM meta file.
IMANY	Package	Number of plots that will be made after specified stress periods.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPLOTV	Package	Flag indicating whether vector plots will be made, =1 bulk flux vector plots will be made. =2 displacement vector plots will be made. <1 or >2 no vector plots will be made.
ISOLD	Package	Before this module allocates space, ISOLD is set equal to ISUM. After allocation, ISOLD is subtracted from ISUM to get ISP, the amount of space in the X array allocated by this module.
ISP	Package	Number of words in the X array allocated by this module.
ISUM	Global	Index number of the lowest element in the X array which has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ISUM1	Module	ISUM-1
IVEC	Package	Flag indicating whether arrow heads are drawn on vectors. =0 no arrow heads are drawn. ≠ 0 arrow heads are drawn.
LCISTR	Package	Location in the X array of the first element of array ISTR.
LCITYPE	Package	Location in the X array of the first element of array ITYPE.
LCLPXY	Package	Location in the X array of the first element of array LPXY.
LCXANG	Package	Location in the X array of the first element of array XANG_ARR.
LCXCNTR	Package	Location in the X array of the first element of array XCNTR.
LCXMAX	Package	Location in the X array of the first element of array XMAX_ARR.
LCYANG	Package	Location in the X array of the first element of array YANG_ARR.
LCYCNTR	Package	Location in the X array of the first element of array YCNTR.
LCYMAX	Package	Location in the X array of the first element of array YMAX_ARR.
LCZANG	Package	Location in the X array of the first element of array ZANG_ARR.
LCZCNTR	Package	Location in the X array of the first element of array ZCNTR.
LCZMAX	Package	Location in the X array of the first element of array ZMAX_ARR.
LENX	Global	Length of the X array in words. This should always be equal to the dimension of X specified in the MAIN program.
NCL	Module	Number of cells through a row of the grid.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NPER	Global	Number of stress periods in the simulation.
NRC	Module	Number of cells in a layer.
NRCL	Module	Number of cells in the grid.

List of Variables for Module PLT1AL

Variable	Range	Definition
NRL	Module	Number of cells through a column of the grid.
NROW	Global	Number of rows in the grid.

Narrative for Module PLTIRP

This module prints the number and type of vector plots that will be made after specified stress periods. This module also reads the information dealing with the line of section where the plot will be taken from. In addition, the module reads the flag to determine after which stress periods the vector plots will be drawn.

Module PLTIRP performs its tasks in the following order:

1. Print plotting information. Prints whether plots will be bulk fluxes or displacements, and prints the number of plots that will be made for the stress period.
2. Set counter for stress period plots
3. Read line-of-section information for each plot that will be made.
4. Read stress period plotting flag to determine whether plots are made for this stress period.
5. RETURN

Flow Chart for Module PLTIRP

ITYPE is a line-of-section flag.

If ITYPE=1 a planimetric plot will be made (x-y plot).
If ITYPE=2 a cross-sectional plot will be made (x-z plot).
If ITYPE=3 a cross sectional plot will be made (y-z plot).
If ITYPE>3 or <1 no plots are made

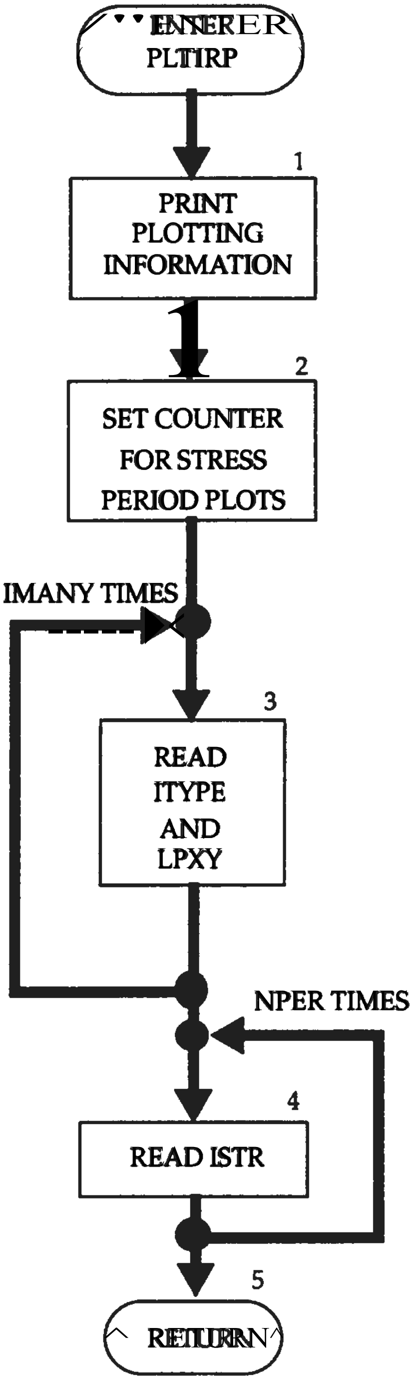
LPXY is the row, column, or layer designation through which a plot is drawn. For ITYPE = LPXY represents a layer number. For ITYPE = 2 LPXY represents row number. For ITYPE = 3 LPXY represents a column number.

IMANY is the number of plots that are drawn for the stress period when ISTR is set

ISTR is a flag indicating that IMANY plots will be made for the stress period

If ISTR>0 vector plots are made for this stress period.
If ISTR<=0 plots will not be made for this stress period.

NPER is the number of stress periods specified in the Basic Package Input.



```
      SUBROUTINE PLT1RP(IPLOTV,ITYPE,LPXY,IMANY,IN,IOUT,IDEV,ISTR,NPER,
     1KPE)
C     ***********************************************************************
C     PRINT TYPE OF VECTOR PLOT TO BE MADE
C     READ AND INITIALIZE DATA FOR THE TYPE OF PLOTS DESIRED
C     ***********************************************************************
C
C     SPECIFICATIONS:
C     ---------------------------------------------------------------
      DIMENSION ITYPE(IMANY),LPXY(IMANY),ISTR(NPER)
C     ---------------------------------------------------------------
C
C1------PRINT OUT PLOTTING INFORMATION
      IF(IPLOTV.LE.0 .OR. IPLOTV.GE.3) THEN
      WRITE(IOUT,8)
    8 FORMAT(1X,' NO PLOTS WILL BE MADE')
      ENDIF
      IF(IPLOTV.EQ.1) THEN
      WRITE(IOUT,9) IMANY
    9 FORMAT(1X,I5,' QBULK VECTOR PLOT(S) WILL BE MADE')
      ENDIF
      IF(IPLOTV.EQ.2)THEN
      WRITE(IOUT,10) IMANY
   10 FORMAT(1X,I5,' DISPLACMENT VECTOR PLOT(S) WILL BE MADE')
      ENDIF
C
C2------SET COUNTER FOR STRESS PERIOD PLOTS
      KPE=0
C
C3------READ ITYPE AND LPXY (SECTION LINE) IMANY TIMES.
      DO 50 I=1,IMANY
      READ(IN,5) ITYPE(I),LPXY(I)
    5 FORMAT(2I10)
   50 CONTINUE
C
C4------SET FLAG FOR PRINTING AFTER SPECIFIED STRESS PERIODS
      READ(IN,55)(ISTR(K),K=1,NPER)
```

```
   55 FORMAT(40I2)
C
C5------RETURN
      RETURN
      END
```

List of Variables for Module PLTIRP

Variable	Range	Definition
I	Module	Index for number of plots (IMANY) per stress period.
IDEV	Package	Flag for device that the vector plots will be plotted to. =1 plot will be displayed in an X-window on the DG. =2 plot will be stored as a postscript file. =3 plot will be stored as a CGM meta file.
IMANY	Package	Number of plots that will be made after specified stress periods.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPLOTV	Package	Flag indicating whether vector plots will be made. =1 bulk flux vector plots will be made. =2 displacement vector plots will be made. <1 or >2 no vector plots will be made.
ISTR	Package	DIMENSION (NPER), Flag indicating whether plots are to be made for the specified stress period. If ISTR>0 plots are made for this stress period. If ISTR<=0 plots are not made for this stress period.
ITYPE	Package	DIMENSION (IMANY), Flag indicating whether plot is planimetric or cross sectional in x or y. If ITYPE=1 planimetric plot will be made (x-y plot). If ITYPE=2 cross-sectional plot will be made (x-z plot). If ITYPE=3 cross-sectional plot will be made (y-z plot).
K	Module	Index for number of stress periods
KPE	Package	Counter for stress periods where plots are to be made.
LPXY	Package	DIMENSION (IMANY), Row, column or layer designation through which plot is drawn. Whether LPXY is a row, column or layer depends on the value of ITYPE.

Narrative for Module PLTIFM

This module makes two dimensional plots of either displacements or bulk fluxes. This is accomplished by calculating the magnitude and direction of each vector relative to the maximum displacement or bulk flux in the grid; hence the vector tails represent the relative magnitude of the value of bulk flux or displacement. The user can make a plot along any user defined line of section. IMANY plots are made after each stress period where ISTR is set. The plots can be drawn directly to the screen in the X-windows environment, or they can be directed to a postscript or CGM meta file.

Module PLTIFM calls submodule SPLTID and function STR_LEN. Module PLTIFM performs its tasks in the following order:

1. Set constants.
2. Check to see if plots are made for this stress period.
3. If plot flag is set write plot number and plot type.
4. Calculate the maximum displacement for each plane in the grid.
5. Determine the real world dimensions of the grid.
6. Calculate the ratio of map length to real world length to obtain a scale.
7. Determine titles for each IMANY plots for each stress period.
8. Set plot number counter. This is to distinguish file names when writing to disks.
9. Check ITYPE to determine the line of section where plot is made. If ITYPE = 1 make a planimetric (x-y plot). If ITYPE = 2 make a cross-sectional plot (x-z plot). If ITYPE = 3, make a cross-sectional plot (y-z plot). Items 10-15 below are read for each ITYPE specified.
10. Set up box and labels by using the GKS software routines.
11. Determine angle for each vector in the grid.
12. Print the maximum displacement for the plane of interest.
13. Determine location for drawing each vector relative to grid size.
14. Calculate magnitude of each vector in the grid.
15. Draw the vectors.
16. RETURN.

Flow Chart for Module PLT1FM

ISTR is a flag indicating whether plots are made for the current stress period.
<=0 no plots are made
>0 plots are made.

IMANY is the number of plots to be made for each stress period.

IJK is the index for the plot number for the entire simulation.

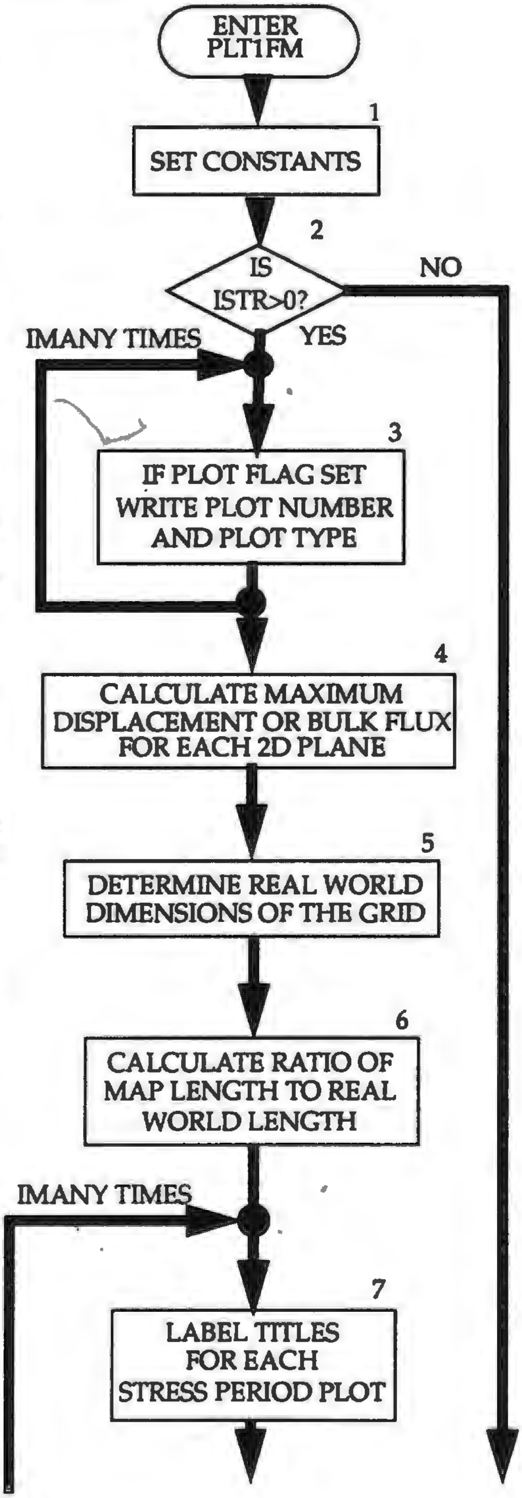
ITYPE is a flag indicating whether plot is planimetric or cross sectional in x or y.
=1 planimetric plot will be made (x-y plot).
=2 cross-sectional plot will be made (x-z plot).
=3 cross-sectional plot will be made (y-z plot).

XANG_ARR, YANG_ARR, and ZANG_ARR are arrays of angles of bulk flux or displacements for each component direction.

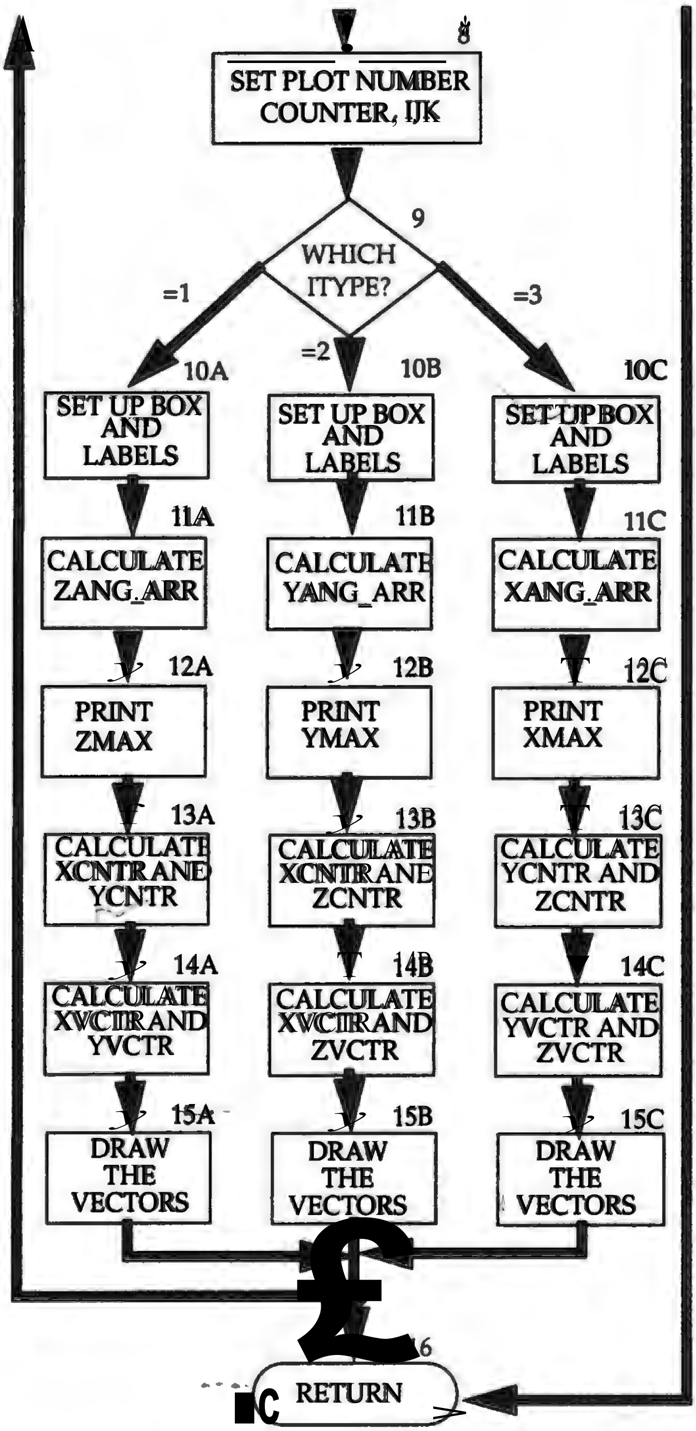
XMAX, YMAX, AND ZMAX are the largest values of bulk flux or displacement in thi x, y, and z directions, respectively.

XCNTR, YCNTR, and ZCNTR represent the locations of each grid center in the x, y, and z directions, respectively.

XVCTR, YVCTR, and ZCTR represent the location of each vector endpoint whose origin is the grid cell center.



Flow Chart for Module PLTIFM (Continued)



```
      SUBROUTINE PLT1FM(XVAL,YVAL,ZVAL,DELR,DELC,XANG_ARR,YANG_ARR,
     1 ZANG_ARR,XMAX_ARR,YMAX_ARR,ZMAX_ARR,DELL,NCOL,NROW,NLAY,
     2 IPLOTV,ITYPE,LPXY,IMANY,IN,IOUT,IDEV,IVEC,IACT,XCNTR,YCNTR,
     3 ZCNTR,ISTR,NPER,KPER,KPE)
C
C ********************************************************************************
C     THIS SUBROUTINE WILL PLOT QBULK OR DISPLACEMENT VECTORS ALONG
C     A USER DEFINED LINE OF SECTION.
C ********************************************************************************
C
C   SPECIFICATIONS:
C------------------------------------------------------------------------------------
      DIMENSION XVAL(NCOL,NROW,NLAY),YVAL(NCOL,NROW,NLAY),
     1 ZVAL(NCOL,NROW,NLAY),DELR(NCOL),DELC(NROW),
     2 XANG_ARR(NROW,NLAY), YANG_ARR(NCOL,NLAY),ZANG_ARR(NCOL,
     3 NROW),ZMAX_ARR(NCOL,NROW,NLAY), XMAX_ARR(NCOL,NROW,NLAY),
     4 YMAX_ARR(NCOL,NROW,NLAY),ITYPE(IMANY),LPXY(IMANY),
     5 DELL(NCOL,NROW,NLAY),IACT(NCOL,NROW,NLAY),
     6  XCNTR(NCOL),YCNTR(NROW),ZCNTR(NLAY),ISTR(NPER)
C ------------------------------------------------------------------------------------
C
      PARAMETER (
     ^        XORG    = 0.,
     ^        YORG    = 0.,
     ^        ZORG    = 0.,
     ^        XLEN    = 10.,
     ^        YLEN    = 10.,
     ^        ZLEN    = 2.0,
     ^        PENTHK  = .0001,
     ^        HHEIGHT = .25,
     ^        PAGEX   = 12.,
     ^        PAGEY   = 12.,
     ^        PAGEZ   = 12.
     ^                 )
C
      CHARACTER TITLEP*24, TITLEY*24, TITLEX*24, VIEW*16, CNCHAR*2
C
```

```
      DOUBLE PRECISION PI
C     ------------------------------------------------------------------
C
C1------SET CONSTANTS
      IXYMAX=AMAX1(NCOL,NROW)
      PLIER=XLEN/FLOAT(IXYMAX)
      PI=4.*ATAN(1.)
C
C2------CHECK STRESS PERIOD TO SEE IF PLOTS ARE TO BE MADE. IF ISTR=0
C        RETURN,
C       IF ISTR>0 INCREMENT KPE.
      IF(ISTR(KPER).LE.0) GOTO 99999
      KPE=KPE+1
      WRITE(IOUT,99) KPER
   99 FORMAT(//,1X,'STRESS PERIOD',I3)
C
C3---PRINT PLOT NUMBER, SECTION TYPE, AND LOCATION OF PLOT
     DO 50 I=1,IMANY
     IJK=(KPE-1)*IMANY+I
     WRITE(IOUT,12) IJK
   12 FORMAT(1X,' PLOT NUMBER',I3,' WILL BE A')
      IF(ITYPE(I).LE.0 .OR. ITYPE(I).GT.3) GOTO 50
      IF(ITYPE(I).EQ.1) THEN
     WRITE(IOUT,6) LPXY(I)
    6 FORMAT(1X,' PLANIMETRIC X-Y PLOT ALONG LAYER',I3)
     ENDIF
     IF(ITYPE(I).EQ.2) THEN
    WRITE(IOUT,7) LPXY(I)
    7 FORMAT(1X,' X-Z SECTION PLOT ALONG ROW',I3)
    ENDIF
    IF(ITYPE(I).EQ.3) THEN
    WRITE(IOUT,11) LPXY(I)
   11 FORMAT(1X,' Y-Z SECTION PLOT ALONG COLUMN',I3)
     ENDIF
  50 CONTINUE
C
C4----CALCULATE THE MAXIMUM DISPLACEMENTS FOR EACH PLANE
```

```
      ZMAX=0.
      YMAX=0.
      XMAX=0.
      DO 35 K=1,NLAY
       DO 35 I=1,NROW
        DO 35 J=1,NCOL
        IF(IACT(J,I,K).EQ.0)GOTO 35
        ZMAX_ARR(J,I,K) = SQRT(XVAL(J,I,K)**2 + YVAL(J,I,K)**2)
        ZMAX = AMAX1(ZMAX,ZMAX_ARR(J,I,K))
        YMAX_ARR(J,I,K) = SQRT(XVAL(J,I,K)**2 + ZVAL(J,I,K)**2)
        YMAX = AMAX1(YMAX,YMAX_ARR(J,I,K))
        XMAX_ARR(J,I,K) = SQRT(YVAL(J,I,K)**2 + ZVAL(J,I,K)**2)
        XMAX = AMAX1(XMAX,XMAX_ARR(J,I,K))
   35 CONTINUE
C
C5---DETERMINE THE REAL WORLD DIMENSIONS OF THE GRID
      SUMR=0.
      SUMC=0.
      SUML=0.
      DO 36 J=1,NCOL
      SUMR=SUMR+DELR(J)
   36 CONTINUE
      DO 37 I=1,NROW
      SUMC=SUMC+DELC(I)
   37 CONTINUE
      DELB=0.
      DO 38 I=1,NROW
      DO 38 J=1,NCOL
      IF(DELL(J,I,NLAY).GT.DELB) THEN
       DELB=DELL(J,I,NLAY)
       JJJ=J
       III=I
      ENDIF
   38 CONTINUE
      DO 39 K=1,NLAY
      SUML=SUML+DELL(JJJ,III,K)
   39 CONTINUE
```

```
C
C6----CALCULATE RATIO OF MAP LENGTH TO REAL WORLD LENGTH TO
C        OBTAIN A SCALE
      XRAT=XLEN/SUMR
      YRAT=YLEN/SUMC
      ZRAT=ZLEN/SUML

C7----DETERMINE TITLES FOR PLOTS
      DO 210 II=1,IMANY
      IF (ITYPE(II).EQ.1) THEN
        VIEW = 'PLAN-VIEW'
      ELSE
        IF (ITYPE(II) .EQ. 2) THEN
          VIEW = 'X CROSS-SECTION'
        ELSE
          IF (ITYPE(II) .EQ. 3) THEN
            VIEW = 'Y CROSS-SECTION'
          ENDIF
        END IF
      ENDIF

      LENV = STR_LEN(VIEW)
C
C8----SET PLOT NUMBER COUNTER
      IJK=(KPE-1)*IMANY+II
C
C9A----IF ITYPE IS 1 THEN MAKE AN X-Y PLOT ALONG A SPECIFIED LAYER Z
C10A---SET UP BOX AND LABELS
      IF(ITYPE(II) .EQ.1) THEN
      WRITE(CNCHAR,'(I2)') LPXY(II)
      TITLEP = VIEW(1:LENV)//'  LAYER '//CNCHAR
      LEN = STR_LEN(TITLEP)

      CALL SPLTID (IJK, IN, IOUT, IDEV, 99999)
      CALL NOBRDR
      CALL PHYSOR (.9,1.)
      CALL PAGE (PAGEX,PAGEY)
```

```
      CALL AREA2D (PAGEX,PAGEY)
      CALL HEIGHT (HHEIGHT)
      IF(IPLOTV.EQ.1) THEN
       CALL MESSAG ('QBULK VECTORS$',13,3.7,10.65)
      ENDIF
      IF(IPLOTV.EQ.2)THEN
       CALL MESSAG ('DISP. VECTORS$',13,3.7,10.65)
      ENDIF
      CALL HEIGHT (HHEIGHT * .6)
      CALL MESSAG (TITLEP(1:LEN),LEN,3.8,10.3)
      CALL STRTPT (XORG,YORG)
      CALL CONNPT (XLEN,XORG)
      CALL CONNPT (XLEN,YLEN)
      CALL CONNPT (XORG,YLEN)
      CALL CONNPT (XORG,YORG)
C
C11A----DETERMINE ANGLE FOR EACH VECTOR IN THE GRID
      DO 10084 N=LPXY(II),LPXY(II)
       DO 10088 M=1,NROW
        DO 10088 L=1,NCOL
         IF(IACT(L,M,N).EQ.0)GOTO 10088
         IF (XVAL(L,M,N) .EQ. 0. .OR. YVAL(L,M,N) .EQ. 0.) THEN
          IF (XVAL(L,M,N) .EQ. 0.) THEN
           IF (YVAL(L,M,N) .GT. 0.) THEN
            ZANG_ARR(L,M) = PI/2.
           ENDIF
           IF (YVAL(L,M,N) .LT. 0) THEN
            ZANG_ARR(L,M) = 3*PI/2.
           ENDIF
          ENDIF
          IF (YVAL(L,M,N) .EQ. 0.) THEN
           IF (XVAL(L,M,N) .GE. 0.) THEN
            ZANG_ARR(L,M) = 0.
           ENDIF
           IF (XVAL(L,M,N) .LT. 0.) THEN
            ZANG_ARR(L,M) = PI
           ENDIF
```

```
           END IF
          ELSE
           IF (XVAL(L,M,N) .GT. 0. .AND. YVAL(L,M,N) .GT. 0.) THEN
            ZANG_ARR(L,M) = ATAN(YVAL(L,M,N)/XVAL(L,M,N))
           END IF
           IF (XVAL(L,M,N) .LT. 0 .AND. YVAL(L,M,N) .NE. 0.) THEN
            ZANG_ARR(L,M) = PI + ATAN(YVAL(L,M,N)/XVAL(L,M,N))
           END IF
           IF (XVAL(L,M,N) .GT. 0. .AND. YVAL(L,M,N) .LT. 0.) THEN
            ZANG_ARR(L,M) = 2*PI + ATAN(YVAL(L,M,N)/XVAL(L,M,N))
           END IF
          END IF
10088    CONTINUE
10084   CONTINUE
C
C12A-----PRINT THE MAXIMUM DISPLACEMENT
      WRITE(IOUT,124) ZMAX
 124  FORMAT(1X,'MAXIMUM DISPLACEMENT IN THE X-Y PLANE IS',1PE12.5)
C
C13A---DETERMINE LOCATION FOR DRAWING EACH VECTOR RELATIVE TO
C       GRID SIZE
      XLINE=0.
      YLINE=0.
      DO 51 J=1,NCOL
      XSTEP=XRAT*DELR(J)
      XLINE=XLINE+XSTEP
      XCNTR(J)=XLINE-(XSTEP*0.5)
  51  CONTINUE
      DO 52 I=1,NROW
      YSTEP=YRAT*DELC(I)
      YLINE=YLINE+YSTEP
      YCNTR(I)=YLINE-(YSTEP*0.5)
  52  CONTINUE
C
C14A---CALCULATE MAGNITUDE OF EACH VECTOR IN THE GRID
      DO 10060 N=LPXY(1),LPXY(U)
      DO 10040 M=1,NROW
```

```
      DO 10040 L=1,NCOL
      IF(IACT(L,M,N).EQ.0) GOTO 10040
      IF (ZMAX_ARR(L,M,N).EQ.0. .OR. ZMAX_ARR(L,M,N).EQ.ZMAX) THEN
       VCTRLEN = 0.
      ELSE
       VCTRLEN = PLIER*(1./(ALOG(ZMAX/ZMAX_ARR(L,M,N))))
      END IF
      IF(VCTRLEN .GT. 2*PLIER) VCTRLEN=2*PLIER
C
      XVCTR = XCNTR(L) + COS(ZANG_ARR(L,M)) * VCTRLEN
      YVCTR = YCNTR(M) + SIN(ZANG_ARR(L,M)) * VCTRLEN
C
C15A-----DRAW THE VECTORS
      IF(IVEC.EQ.0)THEN
       CALL STRTPT(XCNTR(L), YCNTR(M))
       CALL CONNPT(XVCTR, YVCTR)
      ELSE
       CALL VECTOR(XCNTR(L),YCNTR(M),XVCTR,YVCTR,1101)
      ENDIF
C
10040   CONTINUE
10060  CONTINUE
      ELSE
C
C9B-----IF ITYPE IS 2 THEN MAKE A X-Z PLOT ALONG A GIVEN SECTION OF Y
C10B----SET UP BOX AND LABELS
      IF (ITYPE(II) .EQ. 2) THEN
     WRITE(CNCHAR,'(I2)') LPXY(II)
    TITLEY = VIEW(1:LENVY)//'  Y = '//CNCHAR
  - LEN = STR_LEN(TITLEY)
    CALL SPLTID (IJK, IN, IOUT, IDEV, *99999)
     CALL NOBRDR
     CALL PHYSOR(1.,5.)
     CALL PAGE (PAGEX,PAGEZ)
     CALL AREA2D (PAGEX,PAGEZ)
     CALL HEIGHT (HHEIGHT)
     IF(IPLOTV.EQ.1)THEN
```

```
      CALL MESSAG ('QBULK VECTORS$',M,3.3,2.6)
      ENDIF
      IF(IPLOTV.EQ.2)THEN
      CALL MESSAG ('DISP. VECTORS$',M,3.3,2.6)
      ENDIF
      CALL HEIGHT (HHEIGHT * .6)
      CALL MESSAG (TITLEY(1:LEN),LEN,3.2,2.3)
      CALL STRTPT (XORG,ZORG)
      CALL CONNPT (XORG,ZLEN)
      CALL CONNPT (XLEN,ZLEN)
      CALL CONNPT (XLEN,ZORG)
      CALL CONNPT (ZORG,XORG)
C
CUB-----CALCULATE ANGLE OF VECTORS FOR EACH CELL IN THE GRID
       DO 10095 M=LPXY(II),LPXY(II)
        DO 10099 N=1,NLAY
        DO 10099 L=1,NCOL
         IF(IACT(L,M,N).EQ.0)GOTO 10099
         IF (XVAL(L,M,N) .EQ. 0. .OR. ZVAL(L,M,N) .EQ. 0.) THEN
          IF (XVAL(L,M,N) .EQ. 0.) THEN
           IF (ZVAL(L,M,N) .GT. 0.) THEN
            YANG_ARR(L,N) = -PI/2.
           ENDIF
           IF (ZVAL(L,M,N) .LT. 0) THEN
            YANG_ARR(L,N) = -3*PI/2.
           ENDIF
          ENDIF
          IF (ZVAL(L,M,N) .EQ. 0.) THEN
           IF (XVAL(L,M,N) .GE. 0.) THEN
            YANG_ARR(L,N) = 0.
           ENDIF
           IF (XVAL(L,M,N) .LT. 0.) THEN
            YANG_ARR(L,N) = -PI'
           ENDIF
          END IF
         ELSE
          IF (XVAL(L,M,N) .GT. 0. .AND. ZVAL(L,M,N) .GT. 0.) THEN
```

```
          YANG_ARR(L,N) = -ATAN(ZVAL(L,M,N)/XVAL(L,M,N))
         END IF
         IF (XVAL(L,M,N) .LT. 0 .AND. ZVAL(L,M,N) .NE. 0.) THEN
          YANG_ARR(L,N) = -PI - ATAN(ZVAL(L,M,N)/XVAL(L,M,N))
         END IF
         IF (XVAL(L,M,N) .GT. 0. .AND. ZVAL(L,M,N) .LT. 0.) THEN
          YANG_ARR(L,N) = -2*PI - ATAN(ZVAL(L,M,N)/XVAL(L,M,N))
         END IF
         END IF
10099    CONTINUE
10095 CONTINUE

C12B------PRINT THE MAGNITUDE OF THE MAXIMUM Y DISPLACEMENT
        WRITE(IOUX,122) YMAX
  122  FORMAT(1X,'MAXIMUM DISPLACMENT IN THE X-Z PLANE IS ',1PE12.5)

C13B----DETERMINE LOCATION FOR DRAWING EACH VECTOR RELATIVE TO
C        GRID SIZE
      XLINE=0.
      ZLINE=0.
      DO 61 J=1,NCOL
      XSTEP=XRAT*DELR(J)
      XLINE=XLINE+XSTEP
      XCNTR(J)=XLINE-(XSTEP*0.5)
   61 CONTINUE
      DO 62 K=1,NLAY
      KK=NLAY+1-K
      ZSTEP=ZRAT*DELL(JJ,II,KK)
      ZLINE=ZLINE+ZSTEP
      ZCNTR(KK)=ZLINE-(ZSTEP*0.5)
   62 CONTINUE
C
C14B----CALCULATE VECTOR MAGNITUDE FOR EACH CELL IN THE GRID
      DO 10071 I=LPXY(II),LPXY(II)
      DO 10073 J=1,NLAY
      DO 10073 K=1,NCOL
       IF(IACT(K,I,J).EQ.0)GOTO 10073
```

```
        IF (YMAX_ARR(K,I,J).EQ.0. .OR. YMAX_ARR(K,I,J).EQ.YMAX) THEN
         VCTRLEN = 0.
        ELSE
          VCTRLEN = PLIER*(1./(ALOG(YMAX/YMAX_ARR(K,I,J))))
        END IF
        IF(VCTRLEN .GT. 2*PLIER) VCTRLEN=2*PLIER
C
        XVCTR = XCNTR(K) + COS(YANG_ARR(K,J)) * VCTRLEN
        ZVCTR = ZCNTR(J) + SIN(YANG_ARR(K,J)) * VCTRLEN
C
C15B------DRAW THE VECTORS
         IF(IVEC.EQ.0) THEN
         CALL STRTPT(XCNTR(K), ZCNTR(J))
         CALL CONNPT(XVCTR, ZVCTR)
        ELSE
         CALL VECTOR(XCNTR(K),ZCNTR(J),XVCTR,ZVCTR,1101)
        ENDIF

10073   CONTINUE
10071  CONTINUE
       ELSE
C
C9C------IF ITYPE IS 3 THEN MAKE A Y-Z PLOT ALONG A GIVEN SECTION X
C10C-----SET UP BOX AND LABELS
        IF (ITYPE(II) .EQ. 3) THEN
       WRITE(CNCHAR,'(I2)') LPXY(II)
       TITLEX = VIEW(1:LENVY)//'  X = '//CNCHAR
       LEN = STR_LEN(TITLEX)
       CALL SPLTID (IJK, IN, IOUT, IDEV, *99999)
       CALL NOBRDR
       CALL PHYSOR(1.,5.)
       CALL PAGE (PAGEX, PAGEY)
       CALL AREA2D(PAGEX,PAGEY)
       CALL HEIGHT (HHEIGHT)
       IF(IPLOTV.EQ.1)THEN
        CALL MESSAG ('QBULK VECTORS$',13,3.5,2.6)
      ENDIF
```

```
      IF(IPLOTV.EQ.2) THEN
       CALL MESSAG ('DISP. VECTORS$',100,3.5,2.6)
      ENDIF
      CALL HEIGHT (HHEIGHT * .6)
      CALL MESSAG (TITLEX(1:LEN),LEN,3.4,2.3)
      CALL STRTPT (ZORG,XORG)
      CALL CONNPT (ZORG,ZLEN)
      CALL CONNPT (XLEN,ZLEN)
      CALL CONNPT (XLEN,XORG)
      CALL CONNPT (ZORG,XORG)
C
C11C--CALCULATE ANGLE OF EACH VECTOR IN THE GRID
      DO 10077 L=LPXY(1),LPXY(1)
       DO 10075 N=1,NLAY
       DO 10075 M=1,NROW
         IF(IACT(L,M,N).EQ.0)GOTO 10075
         IF (YVAL(L,M,N) .EQ. 0. .OR. ZVAL(L,M,N) .EQ. 0.) THEN
          IF (YVAL(L,M,N) .EQ. 0.) THEN
           IF (ZVAL(L,M,N) .GT. 0.) THEN
            XANG_ARR(M,N) = -PI/2.
           ENDIF
           IF (ZVAL(L,M,N) .LT. 0) THEN
            XANG_ARR(M,N) = -3*PI/2.
           ENDIF
          ENDIF
          IF (ZVAL(L,M,N) .EQ. 0.) THEN
           IF (YVAL(L,M,N) .GE. 0.) THEN
            XANG_ARR(M,N) = 0.
           ENDIF
           IF (YVAL(L,M,N) .LT. 0.) THEN
            XANG_ARR(M,N) = -PI
           ENDIF
          ENDIF
         ELSE
          IF(YVAL(L,M,N) .GT. 0. .AND. ZVAL(L,M,N) .GT. 0.)THEN
           XANG_ARR(M,N) = -ATAN(ZVAL(L,M,N)/YVAL(L,M,N))
          ENDIF
```

```
          IF(YVAL(L,M,N) .LT. 0 .AND. ZVAL(L,M,N) .NE. 0.) THEN
           XANG_ARR(M,N) = -PI - ATAN(ZVAL(L,M,N)/YVAL(L,M,N))
          END IF
          IF(YVAL(L,M,N) .GT. 0. .AND. ZVAL(L,M,N) .LT. 0.)THEN
           XANG_ARR(M,N) =-2*PI- ATAN(ZVAL(L,M,N)/YVAL(L,M,N))
          END IF
         END IF

10075       CONTINUE
10077      CONTINUE
C
C12C---PRINT THE MAXIMUM DISPLACMENT
       WRITE(IOUT,126)XMAX
 126   FORMAT(1X,'MAXIMUM DISPLACEMENT IN THE Y-Z PLANE IS',1PE12.5)
C
C13C---DETERMINE LOCATION FOR DRAWING EACH VECTOR RELATIVE TO
C        GRID SIZE
      YLINE=0.
      ZLINE=0.
      DO 63 I=1,NROW
      YSTEP=YRAT*DELC(I)
      YLINE=YLINE+YSTEP
      YCNTR(I)=YLINE-(YSTEP*0.5)
  63  CONTINUE
      DO 64 K=1,NLAY
      KK=NLAY+1-K
     ZSTEP=ZRAT*DELL(JJJ,III,KK)
     ZLINE=ZLINE+ZSTEP
     ZCNTR(KK)=ZLINE-(ZSTEP*0.5)
  64  CONTINUE
C
C14C----CALCULATE THE MAGNITUDE OF EACH VECTOR IN THE GRID
      DO 10070 L=LPXY(II),LPXY(II)
       DO 10072 J=1,NLAY
       DO 10072 K=1,NROW
       IF(IACT(L,K,J).EQ.0) GOTO 10072
       IF (XMAX_ARR(L,K,J).EQ.0. .OR. XMAX_ARR(L,K,J).EQ.XMAX) THEN
```

```
       VCTRLEN = 0.
     ELSE
       VCTRLEN = PLIER*(1./(ALOG(XMAX/XMAX_ARR(L,K,J))))
     END IF
     IF(VCTRLEN .GT. 2*PLIER) VCTRLEN=2*PLIER
     YVCTR = YCNTR(K) + COS(XANG_ARR(K,J)) * VCTRLEN
     ZVCTR = ZCNTR(J) + SIN(XANG_ARR(K,J)) * VCTRLEN
C
C15C-----DRAW THE VECTORS
     IF(IVEC.EQ.0)THEN
      CALL STRTPT(YCNTR(K), ZCNTR(J))
      CALL CONNPT(YVCTR, ZVCTR)
     ELSE
      CALL VECTOR(YCNTR(K),ZCNTR(J)),YVCTR,ZVCTR,1101)
     ENDIF
10072   CONTINUE
10070  CONTINUE

    ENDIF
    ENDIF
    ENDIF
C
    CALL ENDPL (0)
 210 CONTINUE
    CALL DONEPL
C
C16------RETURN
99999 RETURN
      END
```

List of Variables for Module PLT1FM

Variable	Range	Definition
CNCHAR	Module	String length for label defining location of line-of-section for plots.
DELB	Module	Maximum cell thickness of all cells in the grid.
DELC	Global	DIMENSION (NROW), Cell dimension in the column direction. DELC(I) contains the width of row I.
DELL	Global	DIMENSION (NCOL,NROW,NLAY), Cell dimension in the layer direction.
DELR	Global	DIMENSION (NCOL), Cell dimension in the row direction. DELR(J) contains the width of column J.
HHEIGHT	Module	Height in inches of labels for plots.
I	Module	Index for rows and number of plots
IACT	Global	DIMENSION (NCOL,NROW,NLAY), Status of each cell for displacement. <=0 inactive >0 active
IDEV	Package	Flag for device that the vector plots will be plotted to. =1 plot will be displayed in an X-window on the DG. =2 plot will be stored as a postscript file. =3 plot will be stored as a CGM meta file.
II	Module	Index for IMANY plots.
III	Module	Index for row location of cell with maximum thickness.
IJK	Module	Index for plot number (all stress periods).
IMANY	Package	Number of plots that will be made after specified stress periods.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IPLOTV	Package	Flag indicating whether vector plots will be made. =1 bulk flux vector plots will be made. =2 displacement vector plots will be made. <1 or >2 no vector plots will be made.
ISTR	Package	DIMENSION (NPER), Flag indicating whether plots are to be made for the specified stress period. >0 plots are made for this stress period. <=0 plots are not made for this stress period.
ITYPE	Package	DIMENSION (IMANY), Flag indicating whether plot is planimetric or cross sectional in x or y. =1 planimetric plot will be made (x-y plot). =2 cross-sectional plot will be made (x-z plot). =3 cross-sectional plot will be made (y-z plot).
IVEC	Package	Flag indicating whether vector arrow heads are to be drawn >0 draw arrow heads <=0 no arrow heads are drawn
IXYMAX	Module	Constant equal to the largest value of either NCOL or NROW
J	Module	Index for columns or layers.

List of Variables for Module PLTIFM (Continued)

Variable	Range	Definition
JJJ	Module	Index for column location of cell with maximum thickness
K	Module	Index for layers or columns
KK	Module	NLAY+1-K.
KPE	Module	Counter for stress periods where plots are to be made.
KPER	Global	Stress period counter.
L	Module	Index for columns, LPXY(II).
LEN	Module	Character length of title for plot.
LENV	Module	Location of last character in title string.
LPXY	Package	DIMENSION (IMANY), Row, column or layer designation through which plot is drawn. Whether LPXY is a tow, column or layer depends on the value of ITYPE.
N	Module	Index for LPXY(II) or number of layers.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
PAGEX	Module	Page size in inches in X direction for plots.
PAGEY	Module	Page size in inches in Y direction for plots.
PAGEZ	Module	Page size in inches in Z direction for plots.
PENTHK	Module	Parameter identifying line thickness for plots.
PI	Module	Constant equal to π.
PLIER	Module	Constant identifying the ratio of plot length in inches divided by IXYMAX.
SUMC	Module	Sum of length of all DELC(I) in the grid.
SUML	Module	Sum of length of all DELL(J,I,K) in the grid.
SUMR	Module	Sum of length of all DELR(J) in the grid.
TITLEP	Module	Title for planimetric plot (x-y plot).
TITLEX	Module	Title for cross-sectional plot (y-z plot).
TITLEY	Module	Title for cross-sectional plot (x-z plot).
VCTRLEN	Module	Length of vector multiplied by angle to obtain vector endpoint.
VIEW	Module	Character string containing title for plots.
XANG_ARR	Package	DIMENSION (NCOL,NROW,NLAY), Angle in radians of XVAL for each vector in the grid.
XCNTR	Package	DIMENSION (NCOL), Center of each cell in the grid in the X direction.
XLEN	Module	Length of plotting window in X direction.
XLINE	Module	Cumulative distance to center of each grid cell in X direction.
XMAX	Module	Maximum magnitude of displacement or bulk flux in y-z plane.
XMAX_ARR	Package	DIMENSION (NCOL,NROW,NLAY), Magnitude of displacement or bulk flux in y-z plane.
XORG	Module	Origin of plotting window in X direction.
XRAT	Module	Ratio of XLEN to SUMR.
XSTEP	Module	Length between cell centers in X direction.

List of Variables for Module PLTIFM (Continued)

Variable	Range	Definition
XVAL	Module	DIMENSION (NCOL,NROW,NLAY), X-direction component of displacement or bulk flux.
XVCTR	Module	Location of vector endpoint in X direction for each cell in the grid.
YANG_ARR	Package	DIMENSION (NCOL,NROW,NLAY), Angle in radians of YVAL for each vector in the grid.
YCNTR	Package	DIMENSION (NROW), Center of each cell in the grid in the Y direction.
YLEN	Module	Length of plotting window in Y direction.
YLINE	Module	Cumulative distance to center of each grid cell in Y direction.
YMAX	Module	Maximum magnitude of displacement or bulk flux in x-z plane.
YMAX_ARR	Package	DIMENSION (NCOL,NROW,NLAY), Magnitude of displacement or bulk flux in x-z plane.
YORG	Module	Origin of plotting window in Y direction.
YRAT	Module	Ratio of YLEN to SUMC.
YSTEP	Module	Length between cell centers in Y direction.
YVAL	Package	DIMENSION (NCOL,NROW,NLAY), Y-direction component of displacement or bulk flux.
YVCTR	Module	Location of vector endpoint in Y direction for each cell in the grid.
ZANG_ARR	Package	DIMENSION (NCOL,NROW,NLAY), Angle in radians of ZVAL for each vector in the grid.
ZCNTR	Package	DIMENSION (NCOL), Center of each cell in the grid in the Z direction.
ZLEN	Module	Length of plotting window in Z direction.
ZLINE	Module	Cumulative distance to center of each grid cell in Z direction.
ZMAX	Module	Maximum magnitude of displacement or bulk flux in x-y plane.
ZMAX_ARR	Package	DIMENSION (NCOL,NROW,NLAY), Magnitude of displacement or bulk flux in x-y plane.
ZORG	Module	Origin of plotting window in X direction.
ZRAT	Module	Ratio of ZLEN to SUML.
ZSTEP	Module	Length between cell centers in Z direction.
ZVAL	Package	DIMENSION (NCOL,NROW,NLAY), Z-direction component of displacement or bulk flux.
ZVCTR	Module	Location of vector endpoint in X direction for each cell in the grid.

Variable	Range	Definition of GKS Graphics Subroutines Called by PLTIFM
AREA2D	Module	Defines the subplot area based on axis length.
CONNPT	Module	Connects successive points with straight lines.
DONEPL	Module	Signs off the plotting device and ends plotting.
ENDPL	Module	Terminates a plot page.
HEIGHT	Module	Changes the height of all subsequent strings, numbers and labels.

List of Variables for Module PLTIFM (Continued)

Variable	Range	Definition
MESSAG	Module	Draws the specified string at specified distance from physical origin.
PAGE	Module	Sets the page dimensions wherein the plot is centered and draws a page border.
PHYSOR	Module	Defines the physical origin.
STRTPT	Module	Moves the point without drawing a line.
VECTOR	Module	Draws a vector with the end points specified in inches from the physical origin.

Narrative for Module SPLTID

This subroutine sets the variables and parameters to make plot into one of three types. They include, a CGM meta file for importation into FRAMEMAKER, a PS postscript file that can be printed from the DG to a postscript printer or other software package that can read postscript files, or to an X-window that is automatically opened on Data General Aviion Workstations. These devices can be modified or expanded as needed.

Module SPLTID is called by module PLTIFM and makes numerous GKS graphics subroutine calls to make graphics files or plots to an X-window screen. The module SPLTID performs these functions in the following order:

1. Check IDEV type.
2. If IDEV = 1 the plot is drawn to an X-window environment on the DG workstation. IMANY plots are made after each stress period. Each successive plot will be made by pressing the ENTER key on the terminal keyboard.
3. If IDEV = 2 then make a postscript file with the file name beginning with POST followed by a suffix representing, IJK, the number of the plot for the simulation.
4. If IDEV = 3 then make a CGM meta file with the file name beginning with META followed by a suffix representing, IJK, the number of the plot for the Simulation.
5. RETURN

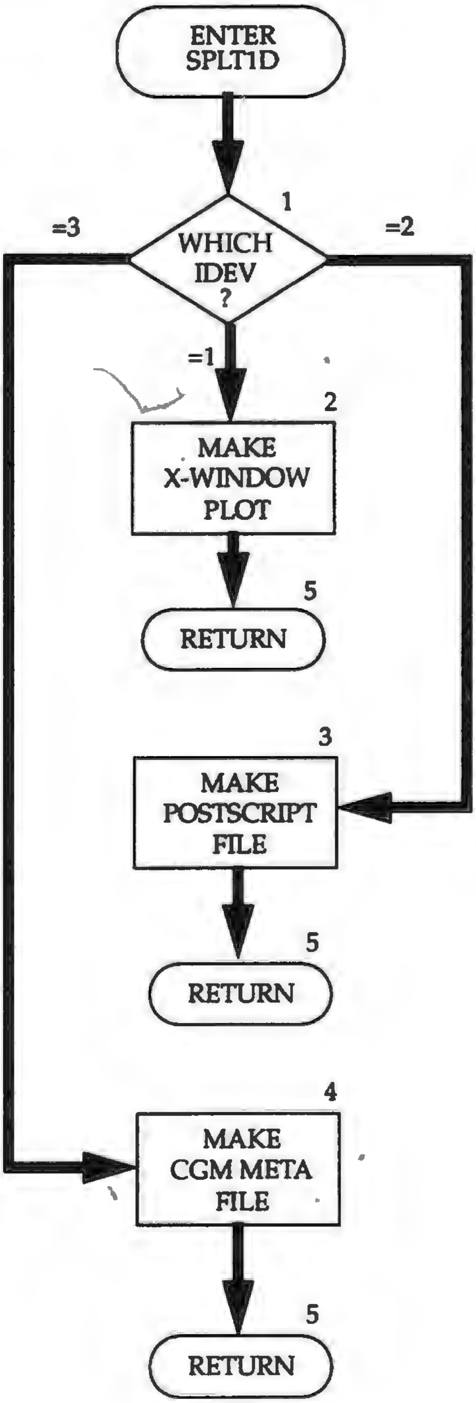
Flow Chart for Module SPLT1D

IDEV is the device or type of file that the plot is written to for plotting or printing.

If IDEV=1 the plot is drawn to an X-window on the DG workstation

If IDEV=2 the plot is written to a postscript file with the prefix POST followed by the plot number designated by the variable IJK

If IDEV=3 the plot is written to a CGM meta file with the prefix META followed by the plot number designated by the variable IJK.



```
      SUBROUTINE SPLTIO(JK, IN, IOUT, IDEV, *)
C
C    ****************************************************************
C    SETS THE VARIABLES AND PARAMETERS TO MAKE PLOT INTO ONE OF THREE
C    TYPES. CGM META FILE FOR IMPORTATION INTO FRAME, PS POSTSCRIPT
C    FILE THAT CAN BE PRINTED FROM THE DG TO A POSTSCRIPT PRINTER OR
C    ON THE SCREEN USING GS (GHOSTSCRIPT), OR THE PLOT CAN BE PLOTTED
C    DIRECTLY TO THE DG SCREEN IN X-WINDOWS. THESE DEVICES CAN BE
C    MODIFIED OR EXPANDED AS NEEDED.
C    ****************************************************************
C
C    SPECIFICATIONS:
C    -----------------------------------------------
      INTEGER I_XARG(10), I_BUF(16), J_BUF(16)
C
      CHARACTER*6 PSTFIL, MTAFIL
      CHARACTER*2 SUFFX
C
      EQUIVALENCE (J_BUF(1), PSTFIL)
C    -----------------------------------------------
C
C1-------IF IDEV = 1 THEN MAKE THE PLOT DIRECTLY TO THE X-WINDOW
C           ENVIRONMENT
       IF (IDEV .EQ.1) THEN
        I_XARG(1) = 112
        I_XARG(2) = 1
        I_XARG(3) = 1
        I_XARG(4) = 6
        I_XARG(5) = 6
        I_XARG(6) = 0
        I_XARG(7) = 3
        I_XARG(8) = -1
        I_XARG(9) = 0
        CALL XWNDOW(1,I_XARG,9)
C
C2-------IF IDEV = 2 THEN MAKE A POSTSCRIPT FILE
      ELSE
```

```
      IF(IDEV .EQ. 2) THEN
       I_BUF(1) = 5
       CALL IOMGR(I_BUF,-102)
       IF(IJK.LT.10) THEN
        WRITE(SUFFX,'(I1)') IJK
      ELSE
        WRITE(SUFFX,'(I2)') IJK
      ENDIF
       PSTFIL='POST'//SUFFX
       INQUIRE (FILE=PSTFIL, EXIST=LOGFIL, OPENED=LOGOPN)
       IF(LOGFIL) THEN
        WRITE(IOUT,100)
        RETURN 1
       ENDIF
       CALL IOMGR(I_BUF,-103)
       I_BUF(1) = 1
       CALL IOMGR(I_BUF,-104)
       CALL PSCRPT (0,0,0)
C
C3------IF IDEV = 3 THEN MAKE A CGM META FILE FOR IMPORT TO FRAME
      ELSE
       IF (IDEV .EQ. 3) THEN
        IF(IJK.LT.10) THEN
         WRITE(SUFFX,'(I1)') IJK
       ELSE
         WRITE(SUFFX,'(I2)') IJK
       ENDIF
       MTAFIL='META'//SUFFX
       INQUIRE (FILE=MTAFIL, EXIST=LOGFIL, OPENED=LOGOPN)
       IF(LOGFIL) THEN
        WRITE(IOUT,100)
        RETURN 1
       ENDIF
        LEN_MTA = STR_LEN(MTAFIL)
        CALL CGMBO (MTAFIL(1:LEN_MTA),LEN_MTA,0)
       ENDIF
      ENDIF
```

```
      END IF

 100  FORMAT(/'//** FILE ALREADY EXISTS**',/)
C
C4—RETURN
      RETURN
      END
```

List of Variables for Module

Variable	Range	Definition
I_BUF	Module	DIMENSION (16), Integer array used by GKS graphics subroutine IOMGR to prepare postscript file.
IDEV	Package	Flag for device that the vector plots will be plotted to. =1 plot will be displayed in an X-window on the DG. =2 plot will be stored as a postscript file. =3 plot will be stored as a CGM meta file.
IJK	Package	Index for plot number.
IN	Module	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
I_XARG	Module	DIMENSION (10), Integer array used by GKS graphics subroutine XWNDOW to prepare X-Window environment for plotting.
J_BUF	Module	DIMENSION (16), Integer array used by GKS graphics subroutine IOMGR to prepare postscript file.
LEN_MTA	Module	Length in bytes of meta file name.
LOGFIL	Module	Temporary file assigned in parameter list for INQUIRE statement. If LOGFIL exists (representing already existing POST or META file) return without writing plot file.
LOGOPN	Module	Temporary file opened by INQUIRE statement.
MTAFIL	Module	Name of current CGM meta file being written.
PSTFIL	Module	Name of current postscript file being written.

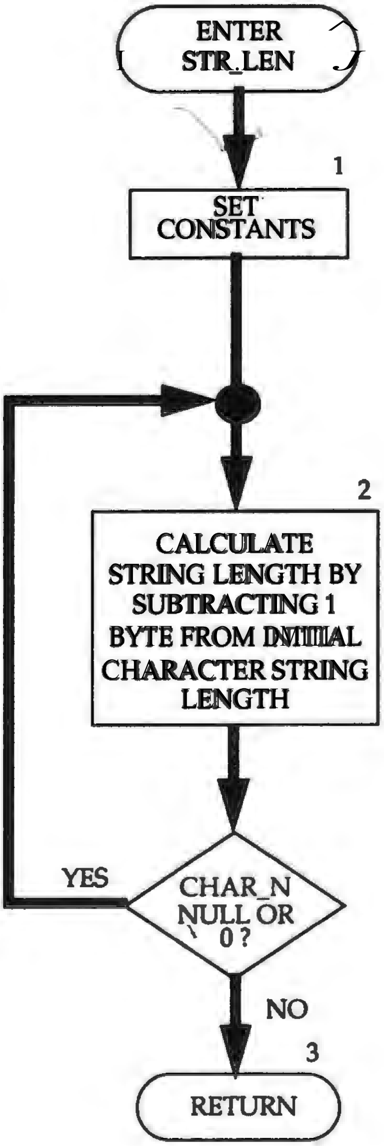
Variable	Range	Definition of GKS Graphics Subroutines Called by SPLT1D
CGMBO	Module	Stores plot information in CGM metafile.
IOMGR	Module	Sets up and queries I/O environment for graphic output.
PSCRPT	Module	Stores plot information in postscript file.
XWNDOW	Module	Sets up X-Window environment for plotting to the screen.

Narrative and Flow Chart of Function STR_LEN

This function subprogram calculates the character string length in bytes of labels and file names. It is called by modules PLTIFM and SPLTID. It perfoms its tasks in the following order:

1. Set constants.
2. Calculate exact string length.
3. Return.

CHAR_N is the length in bytes of the character string. It resides in a do loop that is NBYTE long (specified in character declaration statement). The do loop subtracts one byte through each pass of the loop until CHAR_N is the exact length of the string.



```
      FUNCTION STR_LEN (STRBUF)
C     ***************************************
C     JOHN C. WATSON, SNVCRS, 9/15/88
C     DETERMINE THE CHARACTER STRING LENGTH
C     ***************************************
C
C     SPECIFICATIONS:
C     ----------------------------
      CHARACTER*(*) STRBUF
      CHARACTER*1 CHAR_N, BLNK
C     ----------------------------
C
C1------SET CONSTANTS
      BLNK = ' '
      NBYTE = LEN(STRBUF)
      STR_LEN = NBYTE
C
C2------DETERMINE EXACT LENGTH OF STRING
      DO 100 IBYTE = NBYTE, 1, -1
        CHAR_N = STRBUF(IBYTE:IBYTE)
        IF (CHAR_N .NE. BLNK .AND. ICHAR(CHAR_N) .NE. 0) RETURN
        STR_LEN = STR_LEN-1
  100 CONTINUE
C
C3------RETURN
      RETURN
      END
```

List of Variables for Function STR_LEN

Variable	Range	Definition
BLNK	Module	Null or blank byte.
CHAR_N	Module	Current string length in bytes.
IBYTE	Module	Index for bytes.
NBYTE	Module	Number of bytes initially assigned to character string by character declaration statement.
STRBUF	Module	Buffer for holding file or label name passed into STR_LEN. If the file or label are less than the character string length specified by the character declaration statement then the string contains null characters.
STR_LEN	Package	Exact length of desired character string in bytes.

```
C     ************************************************************
C     MAIN CODE FOR MODULAR MODEL-- 7/2/92
C       BY MICHAEL G. MCDONALD AND ARLEN W. HARBAUGH
C         modified by Thomas J. Burbey
C----VERSION 0212 Feb. 18, 1994; MAIN1
C     ************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------
      COMMON X(1200000)
      COMMON /FLWCOM/LAYCON(80)
      CHARACTER*4 HEADNG,VBNM
      DIMENSION HEADNG(32),VBNM(4,20),VBVL(4,20),IUNIT(24)
      INTEGER*2 ITIM1(28),ITIM2(28)
      DOUBLE PRECISION DUMMY
      EQUIVALENCE (DUMMY,X(1))
C     ------------------------------------------------------------
C
C1------SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
      LENX=1200000
C
C2------ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
      INBAS=5
      IOUT=6
C
C3------DEFINE PROBLEM--ROWS,COLUMNS,LAYERS,STRESS PERIODS,PACKAGES
      CALL BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
     1        NODES,INBAS,IOUT,IUNIT)
C
C4------ALLOCATE SPACE IN "X" ARRAY.
       CALL BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1        LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,
     2        INBAS,ISTRT,NCOL,NROW,NLAY,IOUT)
       IF(IUNIT(1).GT.0) CALL BCF1AL(ISUM,LENX,LCSC1,LCHY,
     1   LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
     2   NCOL,NROW,NLAY,IOUT,IBCFCB)
       IF(IUNIT(2).GT.0) CALL WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,
```

```
1          IUNIT(2),IOUT,IWELCB)
 IF(IUNIT(3).GT.0) CALL DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,
1          IUNIT(3),IOUT,IDRNCB)
 IF(IUNIT(8).GT.0) CALL RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
1       NCOL,NROW,IUNIT(8),IOUT,IRCHCB)
 IF(IUNIT(5).GT.0) CALL EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
1       LCSURF,NCOL,NROW,NEVTOP,IUNIT(5),IOUT,IEVTCB)
 IF(IUNIT(4).GT.0) CALL RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
1       IUNIT(4),IOUT,IRIVCB)
 IF(IUNIT(13).GT.0) CALL STR1AL(ISUM,LENX,LCSTRM,ICSTRM,MXSTRM,
1          NSTREM,IUNIT(13),IOUT,ISTCB1,ISTCB2,NSS,NTRIB,
2           NDIV,ICALC,CONST,LCTBAR,LCTRIB,LCIVAR,LCFGAR)
 IF(IUNIT(7).GT.0) CALL GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,
1       IUNIT(7),IOUT,IGHBCB)
 IF(IUNIT(9).GT.0) CALL SIP1AL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,
1       LCHDCG,LCLRCH,LCW,MXITER,NPARM,NCOL,NROW,NLAY,
2       IUNIT(9),IOUT)
 IF(IUNIT(11).GT.0) CALL SOR1AL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,
1       LCIEQP,MXITER,NCOL,NLAY,NSLICE,MBW,IUNIT(11),IOUT)
 IF(IUNIT(14).GT.0) CALL PCG2AL(ISUM,LENX,LCV,LCSS,LCP,LCCD,
1    LCHCHG,LCLHCH,LCRCHG,LCLRCH,MXITER,ITER1,NCOL,NROW,NLAY,
2     IUNIT(14),IOUT,NPCOND)
 if(iunit(15).gt.0) call qbk1al(isum,lenx,lcqbx,lcqby,lcqbz,lcrat,
1     lcbase,lcqsurf,lcdell,ncol,nrow,nlay,iunit(15),iout,
2 ^ iqbloc,lciact,iqbtyp,lcspx,lcspy,lcspz,lcqx,lcqy,lcqz,
3     lchss,iqbss)
 if(iunit(17).gt.0) call usl1al(isum,lenx,lcuslx,lcusly,lcuslz,
1    lcstmz,lcps,ncol,nrow,nlay,ntmax,lcumag,iunit(17),
2    iout,iusloc,kunit,lcsse,lcssv,lcstmx,lcstmy,lcvstm,
3    lctran,lshc,lcssk,lcuoldx,lcuoldy,lcuoldz,istep,xclose,
    iostp,lctempx,lctempy,lctempz,tclose,lcux,lcuy,lcuz)
 if(iunit(16).gt.0) call plt1al(isum,lenx,lcxang,lcyang,lczang,
1    lcxmax,lcymax,lczmax,iplotv,imahy,lcdpxy,lcitype,ncol,nrow,
2    nlay,iunit(16),iout,idev,ivec,lcxcntr,lcycntr,lczcntr,
3    lcdstr,nper)
 IF (IUNIT(19).GT.0) CALL IBS1AL(ISUM,LENX,LCHC,LCSCE,LCSCV,
1        LCSUB,NCOL,NROW,NLAY,IIBSCB,IIBSOC,ISS,IUNIT(19),IOUT)
```

```
      IF(IUNIT(21).GT.0) CALL HYD3AL(ISUM,LENX,LCHYD3,NHYD3,IHYD3UN,
     1      IUNIT(21),IOUT)
      IF(IUNIT(22).GT.0) CALL TLK2AL(ISUM,LENX,NUMC,NCOL,NROW,NLAY,
     1       LCRAT,LCZCB,LCTLK,LCTL,LCSLU,LCSLD,LCAA,LCBB,LCALPH,
     2       LCBET,LCRM1,LCRM2,LCRM3,LCRM4,NODES1,NM1,NM2,
     3       NTM1,ITLKSV,ITLKRS,IUNIT(22),IOUT,ITLKCB)
C
C5------IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
      IF(ISUM-1.GT.LENX) STOP
C
C6------READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
      CALL BAS1RP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
     1    ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
     2    IUNIT(12),IHEDFM,IDDNFM,IHEDUN,IDDNUN,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1RP(X(LCIBOU),X(LCHNEW),X(LCSC1),
     1       X(LCHY),X(LCCR),X(LCCC),X(LCCV),X(LCDELR),
     2       X(LCDELC),X(LCBOT),X(LCTOP),X(LCSC2),X(LCTRPY),
     3       IUNIT(1),ISS,NCOL,NROW,NLAY,NODES,IOUT)
      IF(IUNIT(9).GT.0) CALL SIP1RP(NPARM,MXITER,ACCL,HCLOSE,X(LCW),
     1     IUNIT(9),IPCALC,IPRSIP,IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1RP(MXITER,ACCL,HCLOSE,IUNIT(11),
     1     IPRSOR,IOUT)
      IF(IUNIT(14).GT.0) CALL PCG2RP(MXITER,ITER1,HCLOSE,RCLOSE,
     1      NPCOND,NBPOL,RELAX,IPRPCG,IUNIT(14),IOUT,MUTPCG,
     2     NITER)
      if(iunit(15).gt.0) call qbk1rp(x(lcqbx),x(lcqby),x(lcqbz),
     1   x(lcrat),x(lcbase),x(lcqsurf),nodes,ncol,nrow,nlay,
     2   iunit(15),iout,x(lcdact),x(lcibou),iqbkoc,iqbkfm,iqbkun,
     3   x(lcdelc),x(lcdelr),x(lcdell),x(lctop),x(lcbot),x(lchnew))
      if(iunit(17).gt.0) call usl1rp(x(lcuslx),x(lcusly),x(lcuslz),
     1   x(lcstmx),x(lcstmy),x(lcstmz),x(lcpssp),x(lcsse),x(lcssv),
     2   nodes,ncol,nrow,nlay,iunit(17),iout,iusloc,
     3   nmagfm,nusxfm,nusyfm,nuszfm,nmagun,nusxun,nusyun,nuszun,
     4   nvstfm,nvstun,x(lcvstm),x(lctran),nimax)
      if(iunit(16).gt.0) call plt1rp(iplotv,x(lcitype),x(lclpxy),
     1  imany,iunit(16),iout,idev,x(lcistr),nper,kpe)
      IF(IUNIT(19).GT.0) CALL IBS1RP(X(LCDELR),X(LCDELC),X(LCHNEW),
```

```
     1    X(LCHC),X(LCSCE),X(LCSCV),X(LCSUB),NCOL,NROW,NLAY,
     2    NODES,IIBSOC,ISUBFM,ICOMFM,IHCFM,ISUBUN,ICOMUN,IHCUN,
     3    IUNIT(19),IOUT)
      IF(IUNIT(22).GT.0) CALL TLK2RP(X(LCRAT),X(LCZCB),
     1      X(LCRM1),X(LCRM2),X(LCRM3),X(LCRM4),
     2      X(LCAA),X(LCBB),X(LCBUFF),X(LCALPH),X(LCBET),
     3      NROW,NCOL,NUMC,NODES1,NM1,NM2,NTM1,ITLKRS,X(LCDELC),
     4      X(LCDELR),TOTIM,DELTM1,IUNIT(22),IOUT)
      IF(IUNIT(21).GT.0) CALL HYD3RP(X(LCHYD3),NHYD3,NUMH,IHYD3UN,
     1      X(LCDELR),X(LCDELC),NCOL,NROW,NLAY,LCHNEW,LCSUB,
     2      LCHC,IUNIT(21),IOUT)
C
C-----WRITE STARTING HYDROGRAPH RECORD
      IF(IUNIT(21).GT.0) CALL HYD3OT(X,ISUM,X(LCHYD3),NUMH,
     1                IHYD3UN,0.0)
C
C7-----SIMULATE EACH STRESS PERIOD.
      DO 300 KPER=1,NPER
      KKPER=KPER
C
      nbulk=0
      if(iunit(15).gt.0) call qbk1st(x(lchss),x(lcibou),x(lctran),
     1 x(lchy),x(lccv),x(lctop),x(lcbot),x(lcdell),x(lcdelr),
     2 x(lcdelc),ncol,nrow,nlay,x(lcspx),x(lcspy),x(lcspz),
     3 x(lcqx),x(lcqy),x(lcqz),iunit(15),iout,iqbss)
c
C7A-----READ STRESS PERIOD TIMING INFORMATION.
      CALL BAS1ST(NSTP,DELT,TSMULT,PERTIM,KKPER,INBAS,IOUT)
C
C7B-----READ AND PREPARE (formulate) INFORMATION FOR STRESS PERIOD.
      IF(IUNIT(2).GT.0) CALL WEL1RP(X(LCWELL),NWELLS,MXWELL,IUNIT(2),
     2      X(LCAA),X(LCBB),X(LCBOU),X(LCALPH),X(LCBET),
     3      NROW,NCOL,NLAY,NUMC,DELT,TOTIM,DELTM1,
     4      NM1,NM2,NTM1,IUNIT(22),IOUT)
C
C7C2-----ITERATIVELY FORMULATE AND SOLVE THE EQUATIONS.
      DO 100 KITER=1,MXITER
```

```
      KKITER=KITER
C
C7C2A—FORMULATE THE FINITE DIFFERENCE EQUATIONS.
      CALL BAS1FM(X(LCHCOF),X(LCRHS),NODES)
      IF(IUNIT(1).GT.0) CALL BCF1FM(X(LCHCOF),X(LCRHS),X(LCHOLD),
     1     X(LCSC1),X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),X(LCCV),
     2     X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCSC2),
     3     X(LCDELR),X(LCDELC),DELT,ISS,KKITER,KKSTP,KKPER,NCOL,
     4     NROW,NLAY,IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1FM(NWELLS,MXWELL,X(LCRHS),X(LCWELL),
     1     X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(3).GT.0) CALL DRN1FM(NDRAIN,MXDRN,X(LCDRAI),X(LCHNEW),
     1    X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(8).GT.0) CALL RCH1FM(NRCHOP,X(LCIRCH),X(LCRECH),
     1     X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(5).GT.0) CALL EVT1FM(NEVTOP,X(LCIEVT),X(LCEVTR),
     1      X(LCEXDP),X(LCSURF),X(LCRHS),X(LCHCOF),X(LCIBOU),
     1     X(LCHNEW),NCOL,NROW,NLAY)
      IF(IUNIT(4).GT.0) CALL RIV1FM(NRIVER,MXRIVR,X(LCRIVR),X(LCHNEW),
     1     X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(13).GT.0) CALL STR1FM(NSTREM,X(LCSTRM),X(ICSTRM),
     1          X(LCHNEW),X(LCHCOF),X(LCRHS),X(LCIBOU),
     2       MXSTRM,NCOL,NROW,NLAY,IOUT,NSS,X(LCTBAR),
     3       NTRIB,X(LCTRIB),X(LCIVAR),X(LCFGAR),ICALC,CONST)
      IF(IUNIT(7).GT.0) CALL GHB1FM(NBOUND,MXBND,X(LCBNDS),X(LCHCOF),
     1     X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(19).GT.0) CALL IBS1FM(X(LCRHS),X(LCHCOF),X(LCHNEW),
     1    X(LCHOLD),X(LCHC),X(LCSCE),X(LCSCV),X(LCIBOU),
     2    NCOL,NROW,NLAY,DELT)
      IF(IUNIT(22).GT.0) CALL TLK2FM(X(LCTL),X(LCTLK),X(LCSLU),X(LCSLD),
     1     X(LCCV),X(LCHCOF),X(LCRHS),NROW,NCOL,NLAY,NUMC,
     2     X(LCIBOU),X(LCRAT))
C
C7C2B—MAKE ONE CUT AT AN APPROXIMATE SOLUTION.
      IF(IUNIT(9).GT.0) CALL SIP1AP(X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),
     1  X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCFL),X(LCGL),X(LCV),
     2  X(LCW),X(LCHDCG),X(LCLRCH),NPARM,KKITER,HCLOSE,ACCL,ICNVG,
```

```
     3  KKSTP,KKPER,IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,
     4  IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AP(X(LCHNEW),X(LCIBOU),X(LCCR),
     1  X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCA),X(LCRES),X(LCIEQP),
     2  X(LCHDCG),X(LCLRCH),KKITER,HCLOSE,ACCL,ICNVG,KKSTP,KKPER,
     3  IPRSOR,MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
      IF(IUNIT(14).GT.0) CALL PCG2AP(X(LCHNEW),X(LCIBOU),X(LCCR),
     1    X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCV),X(LCSS),X(LCP),
     2    X(LCCD),X(LCHCHG),X(LCLHCH),X(LCRCHG),X(LCLRCH),KKITER,
     3    NITER,HCLOSE,RCLOSE,ICNVG,KKSTP,KKPER,IPRPCG,MXITER,ITER1,
     4  NPCOND,NBPOL,NSTP,NCOL,NROW,NLAY,NODES,RELAX,IOUT,MUTPCG)
C
C7C2C---IF CONVERGENCE CRITERION HAS BEEN MET STOP ITERATING.
      IF(ICNVG.EQ.1) GO TO 110
  100 CONTINUE
      KITER=MXITER
  110 CONTINUE
C
C7C3----DETERMINE WHICH OUTPUT IS NEEDED.
      CALL BAS1OC(NSTP,KKSTP,ICNVG,X(LCIOFL),NLAY,
     1  IBUDFL,ICBCFL,IHDDFL,IUNIT(12),IOUT)
C
C7C4----CALCULATE BUDGET TERMS. SAVE CELL-BY-CELL FLOW TERMS.
      MSUM=1
      IF(IUNIT(22).GT.0) CALL TLK2BD(X(LCHNEW),X(LCTLK),X(LCTL),
     1      X(LCSLU),X(LCSLD),X(LCRAT),X(LCCV),VBNM,VBVL,X(LCIBOU),
     2      MSUM,NUMC,NCOL,NROW,NLAY,DELT,KSTP,KPER,ITLKCB,ICBCFL,
     3      X(LCBUFF),IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1BD(VBNM,VBVL,MSUM,X(LCHNEW),
     1  X(LCIBOU),X(LCHOLD),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
     2  X(LCTOP),X(LCSC2),DELT,ISS,NCOL,NROW,NLAY,KKSTP,KKPER,
     3  IBCFCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1BD(NWELLS,MXWELL,VBNM,VBVL,MSUM,
     1  X(LCWELL),X(LCIBOU),DELT,NCOL,NROW,NLAY,KKSTP,KKPER,IWELCB,
     2  ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,
     1  X(LCDRAI),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
```

```
     2    KKPER,IDRNCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1BD(NRCHOP,X(LCIRCH),X(LCRECH),
     1    X(LCIBOU),NROW,NCOL,NLAY,DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,
     2    IRCHCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1BD(NEVTOP,X(LCIEVT),X(LCEVTR),
     1    X(LCEXDP),X(LCSURF),X(LCIBOU),X(LCHNEW),NCOL,NROW,NLAY,
     2   DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,IEVTCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1BD(NRIVER,MXRIVR,X(LCRIVR),X(LCIBOU),
     1    X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,
     2    KKSTP,KKPER,IRIVCB,ICBCFL,X(LCBUFF),IOUT)
     IF(IUNIT(13).GT.0) CALL STR1BD(NSTREM,X(LCSTRM),X(ICSTRM), X(LCIBOU),
     1   MXSTRM,X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,
     2   KKSTP,KKPER,ISTCB1,ISTCB2,ICBCFL,X(LCBUFF),IOUT,NTRIB,NSS,
     3   X(LCTRIB),X(LCTBAR),X(LCIVAR),X(LCFGAR),ICALC,CONST,IPTFLG)
      IF(IUNIT(7).GT.0) CALL GHB1BD(NBOUND,MXBND,VBNM,VBVL,MSUM,
     1    X(LCBNDS),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
     2    KKPER,IGHBCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(19).GT.0) CALL IBS1BD(X(LCIBOU),X(LCHNEW),X(LCHOLD),
     1     X(LCHC),X(LCSCE),X(LCSCV),X(LCSUB),X(LCDELR),X(LCDELC),
     2     NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,KSTP,KPER,IIBSCB,
     3     ICBCFL,X(LCBUFF),IOUT)
C
C-----WRITE HYDROGRAPH RECORD
       IF(IUNIT(21).GT.0) CALL HYD3OT(X,ISUM,X(LCHYD3),NUMH,
     1                   IHYD3UN,TOTIM)
C
c----Perform calculations for directional components of displacement
      if(iunit(17).gt.0) call usllfm(x(lcqbx),x(lcqby),x(lcqbz),
     1 x(lcdact),x(lcdebr),x(lcdelc),x(lcdell),ncol,nrow,nlay,
     2 x(lcsc1),x(lchy),x(lcusk),x(lcusly),x(lcuslz),x(lcstmz),
     3 x(lcps),iout,iusloc,delt,totim,x(lcrat),kkstp,x(lcsse),
     4 x(lcssv),kunit,x(lcstmx),x(lcstmy),x(lcvstm),
     5 x(lctran),x(lcsc2),x(lchc),x(lcssk),x(lcuoldx),x(lcuoldy),
     6 x(lcuoldz),istep,xclose,iostp,kkper,x(lctempx),
     7 x(lctempy),x(lctempz),tclose,x(lcux),x(lcuy),x(lcuz),x(lchnew),
     8 x(lcbot),x(lcibou),x(lchold),x(lcspx),x(lcspy),x(lcspz),nbulk)
c
```

```
c-----print and or save subsidence, magnitude of displacement and
c-----directional components of displacement.
      if(iunit(17).gt.0) call uslot(x(lcuslx),x(lcusly),x(lcuslz),
     1   x(lcumag),ncol,nrow,nlay,iout,iunit(17),iusloc,x(lciact),
     2   nstp,kper,kkstp,nmagfm,nusxfm,nusyfm,nuszfm,nvstfm,
     3   nmagun,nusxun,nusyun,nuszun,nvstun,pertim,totim,x(lcvstm),
     4   x(lcbuff))
C7C5—PRINT AND OR SAVE HEADS AND DRAWDOWNS. PRINT OVERALL
C     BUDGET.
      CALL BAS1OT(X(LCHNEW),X(LCSTRT),ISTRT,X(LCBUFF),X(LCIOFL),
     1     MSUM,X(LCIBOU),VBNM,VBVL,KKSTP,KKPER,DELT,
     2     PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
     3   IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT)
C
C7C5A--PRINT AND OR SAVE SUBSIDENCE, COMPACTION, AND CRITICAL HEAD.
      IF(IUNIT(19).GT.0) CALL IBS1OT(NCOL,NROW,NLAY,PERTIM,TOTIM,KSTP,
     1    KPER,NSTP,X(LCBUFF),X(LCSUB),X(LCHC),IIBSOC,ISUBFM,ICOMFM,
     2    IHCFM,ISUBUN,ICOMUN,IHCUN,IUNIT(19),IOUT)
C
C7C6—IF ITERATION FAILED TO CONVERGE THEN STOP.
      IF(ICNVG.EQ.0)STOP
  200 CONTINUE
c-----print and or save qbulk terms to output
      if(iunit(15).gt.0) call qbkot(x(lcqbx),x(lcqby),x(lcqbz),
     1    ncol,nrow,nlay,iqbkoc,iqbkfm,iqbkun,iunit(15),iout,
     2    kper,kkstp,pertim,totim,x(lcbuff))
c
c----plot bulk fluxes (iplotv=1) or displacements (iplotv=2)
      if(iplotv.le.0 .or.iplotv.ge.3) goto 205
      if(iunit(15).le.0 .and. iunit(17).le.0) goto 205
      if(iplotv.eq.1) then
      if(iunit(16).gt.0) call pltlfm(x(lcqbx),x(lcqby),x(lcqbz),
     1 x(lcdelr),x(lcdelc),x(lcxang),x(lcyang),x(lczang),x(lcxmax),
     2 x(lcymax),x(lczmax),x(lcdell),ncol,nrow,nlay,iplotv,x(lcitype),
     3 x(lclpxy),imany,iunit(16),iout,idev,idev,x(lciact),x(lcxcntr),
     4 x(lcycntr),x(lczcntr),x(lcistr),nper,kper,kpe)
      endif
```

```
        if(iplotv.eq.2) then
        if(iunit(16).gt.0) call pltlfm(x(lcux),x(lcuy),x(lcuz),
     1  x(lcdelr),x(lcdelc),x(lcxang),x(lcyang),x(lczang),x(lcxmax),
     2  x(lcymax),x(lczmax),x(lcdell),ncol,nrow,nlay,iplotv,x(lcdtype),
     3  x(lcpxy),imany,iunit(16),iout,idev,ivec,x(lcact),x(lcxcntr),
     4  x(lcycntr),x(lczcntr),x(lcistr),nper,kper,kpe)
        endif
  205  continue
  300  CONTINUE
C
C7C7—WRITE RESTART RECORDS
C7C7A~WRITE RESTART RECORDS FOR TLK2 PACKAGE
         IF(IUNIT(22).GT.0) CALL TLK2RS(TLKSV,X(LCRM1),X(LCRM2),
     1     X(LCRM3),X(LCRM4),NM1,NM2,DELTM1,TOTIM,IOUT)
C
C8-----END PROGRAM
       STOP
C
       END
```