HST3D: A COMPUTER CODE FOR SIMULATION OF HEAT AND SOLUTE TRANSPORT IN THREE-DIMENSIONAL GROUND-WATER FLOW SYSTEMS

By Kenneth L. Kipp, Jr.

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 86-4095

Denver, Colorado
1987

UNITED STATES DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, Secretary

GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

Project Chief, WRD, Central Region
U.S. Geological Survey,
MS 413, Box 25046,
Denver Federal Center
Denver, CO 80225

Copies of this report can
be purchased from:

U.S. Geological Survey
Books and Open-File Reports
Federal Center, Bldg. 810
Box 25425,
Denver, CO 80225
[Telephone: (303) 236-7476]

CONTENTS

FIGURES

TABLES

CONVERSION FACTORS

The HST3D simulator program performs calculations in metric units. However, it will accept input and produce output in inch-pound units. The conversion factors are listed below:

Multiply	*By*	*To obtain*
kilogram (kg)	2.204622	pound (lb)
meter (m)	3.280840	foot (ft)
millimeter(mm)	3.937008×10^{-2}	inch (in.)
second (s)	1.157407×10^{-5}	day (d)
degree Celsius (°C)	T(°F) = 1.8T(°C) + 32	degree Fahrenheit (°F)
Kelvin (K)	T(°F) = 1.8T(K) - 459.67	degree Fahrenheit (°F)
Joule (J) or Watt-second (W-s)	9.478170×10^{-4}	British Thermal Unit (BTU)
square meter (m^2)	10.76391	square foot (ft^2)
cubic meter (m^3)	35.31466	cubic foot (ft^3)
meter-second (m-s)	3.797267×10^{-5}	foot-day (ft-d)
Pascal (Pa)	1.450377×10^{-4}	pound per square inch (psi)
meter per second (m/s)	2.834646×10^{5}	foot per day (ft/d)
square meter per second (m^2/s)	9.300018×10^{5}	square foot per day (ft^2/d)
cubic meter per second (m^3/s)	3.051187×10^{6}	cubic foot per day (ft^3/d)
liter per second (ℓ/s)	3.051187×10^{3}	cubic foot per day (ft^3/d)
kilogram per second (kg/s)	1.904794×10^{5}	pound per day (lb/d)
Pascal per second (Pa/s)	12.53126	pound per square inch per day ($lb/in^2/d$)
cubic meter per cubic meter-second (m^3/m^3-s)	8.6400×10^{4}	cubic foot per cubic foot-day (ft^3/ft^3-d)
kilogram per cubic meter (kg/m^3)	6.242797×10^{-2}	pound per cubic foot (lb/ft^3)[1]
Watt per cubic meter (W/m^3)	9.662109×10^{-2}	British Thermal Unit per hour-cubic foot (BTU/h-ft^3)
Joule per kilogram (J/kg)	4.299226×10^{-4}	British Thermal Unit per pound (BTU/lb)
Joule per kilogram (J/kg)	0.3345526	foot-pound force per pound mass (ft-lbf/lbm)
cubic meter per kilogram (m^3/kg)	16.01846	cubic foot per pound (ft^3/lb)
cubic meter per square meter-second (m^3/m^2-s)	2.834646×10^{5}	cubic foot per square foot-day (ft^3/ft^2-d)
Watt per square meter (W/m^2)	0.3169983	British Thermal Unit per hour-square foot (BTU/h-ft^2)
kilogram per square meter-second (kg/m^2-s)	1.769611×10^{4}	pound per square foot-day (lb/ft^2-d)

cubic meter per meter-second ($m^3/m\text{-}s$)	9.300018×10^5	cubic foot per foot-day ($ft^3/ft\text{-}d$)
kilogram per meter-second (kg/m-s)	1,000	centipoise (cP)[2]
Joule per kilogram-meter (J/kg-m)	1.310404×10^{-4}	British Thermal Unit per pound-foot (BTU/lb-ft)
Watt per meter-degree Celsius (W/m-°C)	13.86941	British Thermal Unit per foot-hour-degree Fahrenheit (BTU/ft-h-°F)
Watt per square meter-degree Celsius ($W/m^2\text{-}°C$)	0.1761102	British Thermal Unit per hour-square foot-degree Fahrenheit ($BTU/h\text{-}ft^2\text{-}°F$)
Joule per kilogram-degree Celsius (J/kg-°C)	2.388459×10^{-4}	British Thermal Unit per pound-degree Fahrenheit (BTU/lb-°F)
Joule per cubic meter-degree Celsius ($J/m^3\text{-}°C$)	1.491066×10^{-5}	British Thermal Unit per cubic foot-degree Fahrenheit ($BTU/ft^3\text{-}°F$)
cubic meter per second-meter-Pascal ($m^3/s\text{-}m\text{-}Pa$)	6.412138×10^9	cubic foot per day-foot-pound-square inch ($ft^3/d\text{-}ft\text{-}psi$)

[1] A weight density rather than a mass density.
[2] Not inch-pound but common usage.

HST3D: A COMPUTER CODE FOR SIMULATION OF HEAT AND SOLUTE TRANSPORT IN THREE-DIMENSIONAL GROUND-WATER FLOW SYSTEMS

By Kenneth L. Kipp Jr.

ABSTRACT

The Heat- and Solute-Transport Program (HST3D) simulates ground-water flow and associated heat and solute transport in three dimensions. The HST3D program may be used for analysis of problems such as those related to subsurface-waste injection, landfill leaching, saltwater intrusion, freshwater recharge and recovery, radioactive-waste disposal, hot-water geothermal systems, and subsurface-energy storage. The three governing equations are coupled through the interstitial pore velocity, the dependence of the fluid density on pressure, temperature, and solute-mass fraction, and the dependence of the fluid viscosity on temperature and solute-mass fraction. The solute-transport equation is for only a single, solute species with possible linear-equilibrium sorption and linear decay. Finite-difference techniques are used to discretize the governing equations using a point-distributed grid. The flow-, heat- and solute-transport equations are solved, in turn, after a partial Gauss-reduction scheme is used to modify them. The modified equations are more tightly coupled and have better stability for the numerical solutions.

The basic source-sink term represents wells. A complex well-flow model may be used to simulate specified flow rate and pressure conditions at the land surface or within the aquifer, with or without pressure and flow-rate constraints. Boundary-condition types offered include specified value, specified flux, leakage, heat conduction, an approximate free surface, and two types of aquifer-influence functions. All boundary conditions can be functions of time.

Two techniques are available for solution of the finite-difference matrix equations. One technique is a direct-elimination solver, using equations reordered by alternating diagonal planes. The other technique is an iterative solver, using two-line successive overrelaxation. A restart option is available for storing intermediate results and restarting the simulation at an intermediate time with modified boundary conditions. This feature also can be used as protection against computer-system failure.

Data input and output may be in metric (SI) units or inch-pound units. Output may include tables of dependent variables and parameters, zoned-contour maps, and plots of the dependent variables versus time. The HST3D program is a descendant of the Survey Waste Injection Program (SWIP) written for the U.S. Geological Survey under contract.

1. INTRODUCTION

1.1. OVERVIEW OF THE SIMULATOR

The computer program (HST3D) described in this report simulates heat and solute transport in three-dimensional saturated ground-water flow systems. The equations that are solved numerically are: (1) The saturated ground-water flow equation, formed from the combination of the conservation of total-fluid mass and Darcy's Law for flow in porous media; (2) the heat-transport equation from the conservation of enthalpy for the fluid and porous medium; and (3) the solute-transport equation from the conservation of mass for a single-solute species, that may decay and may adsorb onto the porous medium. These three equations are coupled through the dependence of advective transport on the interstitial fluid-velocity field, the dependence of fluid viscosity on temperature and solute concentration, and the dependence of fluid density on pressure, temperature, and solute concentration.

Numerical solutions are obtained for each of the dependent variables: pressure, temperature, and mass fraction (solute concentration) in turn, using a set of modified equations that more directly link the original equations through the velocity-, density-, and viscosity-coupling terms. Finite-difference techniques are used for the spatial and temporal discretization of the equations. When supplied with appropriate boundary and initial conditions and system-parameter distributions, simulation calculations can be performed to evaluate a wide variety of heat- and solute-transport situations.

The computer code (HST3D) described in this documentation is a descendant of a computer code for calculating the effects of liquid-waste disposal into deep, saline aquifers, developed by INTERCOMP Resource Development and Engineering Inc. 1976) for the U.S. Geological Survey and revised by INTERA Environmental Consultants Inc. (1979). The parent code, known as the Survey Waste Injection Program (SWIP), has been completely rewritten with many major and minor modifications, improvements, and correction of several errors. Features included in HST3D are briefly described as follows:

1. Specified-value and specified-flux boundary conditions are independent of each other and independent of the well or aquifer-influence-function boundary conditions. The boundary conditions also may vary with time.
2. Specified heat- and solute-flux boundary conditions are available.
3. The leakage boundary conditions are generalized and a river-leakage boundary condition is available.
4. Porous-medium thermal properties, dispersivity, and compressibility, may have spatial variation defined by zones.
5. A point-distributed, finite-difference grid is employed, rather than a cell- or block-centered grid, for less truncation error and easier incorporation of boundary conditions.
6. The heat-conduction boundary condition is generalized to apply to any cell face.
7. Global-flow, and heat- and solute-balance calculations are performed including flux calculations through specified pressure, temperature, and mass-fraction boundaries.
8. A robust algorithm for the computation of the optimum overrelaxation factor for the two-line, successive-overrelaxation, matrix-solution method is used, with a convergence criterion that includes the matrix spectral-radius estimate.
9. The code is organized for a logical flow of calculation and a modular structure.
10. The code length is about 12,000 lines, using FORTRAN 77 language constructs for cleaner, more efficient coding than possible with FORTRAN 66. However, clarity has not been sacrificed for ultimate efficiency.
11. Comments have been included liberally for ease of understanding the program.
12. All arrays with lengths depending on the size of the problem are in two variably-partitioned arrays, integer and real, to facilitate double-precision arithmetic.

13. Arrays required for thermal or solute calculations exclusively are eliminated if only one of these transported quantities is being simulated, which results in a considerable decrease in computer storage.
14. Arrays used for a specific type of boundary condition or source-sink condition are dimensioned only to the length required.
15. The allocation of space for the direct-equation solver is explicitly determined during array-space allocation, rather than estimated.
16. Logical variables are used to control the flow of program execution for ease of option selection.
17. The input file is in free-format to facilitate input from terminals.
18. The input file is organized into logical groups for parameter specifications.
19. User comments can be freely incorporated into the input file for rapid identification of the data. An input-file form is available which the user can fill out at the terminal for a given simulation.
20. A read-echo file may be written to aid in locating errors in the data-input file.
21. Character plots of the porous-media zones may be created on the output file to facilitate checking the zonation.
22. Although the internal calculations of the program are performed in metric units, the input and output can be chosen to be in inch-pound units.
23. The output material is made easily understandable by avoiding variable names, by logical grouping on the page, and by including supplementary information.
24. Error tests are included to catch likely mistakes in data input.
25. Error messages are printed explicity rather than as code numbers.

26. There is no limit on the number of plots that can be created. The number of calculated points in time per plot is limited to three times the total number of grid points, while the number of observed points in time is limited to two times the number of grid points. The user can select every nth point to be plotted, if this number is limiting.

27. The solute concentration can be chosen to be the mass fraction or a scaled mass fraction that ranges from 0 to 1. This choice was available in the SWIP code, but the user was not clearly made aware of which option was selected.

28. Two types of restart option are available: a periodic check-point dump for protection against computer-system failure, and a specific dump for user review and possible modification of parameters.

29. Map-contour intervals can be automatically determined to be a multiple of 2, 5, or 10, and the contour zones are "zebra striped" for easier reading.

30. Initial-pressure conditions can be specified to be other than hydrostatic. For example, an initial water-table configuration can be used.

31. The precipitation-infiltration option is contained in the distributed flux-boundary conditions.

32. The conductive-heat-loss to overburden and underburden is a general, heat-transfer calculation, applicable to any cell face in the region.

33. The well-riser, heat-transfer calculation is based on heat transfer from a known-temperature, cylindrical boundary, and higher order assymptotic expansions have been used.

34. The well-riser calculation has been formulated to solve the total-energy and momentum balance equations simultaneously, using the Bulirsch-Stoer algorithm for integration of the ordinary differential equations.

35. The well-bore equations are implicitly coupled to the system equations for cases of cylindrical geometry.

36. The well-datum pressure and the well-flow rate allocation calculations may be performed iteratively in conjunction with the solution of the flow equation, or explicitly.
37. The full nine-component, or an approximate three-component, dispersion-coefficient tensor may be used for cross-dispersive flux calculations.

The purpose of simulation modeling the transport of heat and solute in ground-water flow systems is to gain a quantitative understanding of how the sources and sinks, the boundary conditions, and the aquifer parameters interact to cause ground-water flow patterns and consequent thermal- and solute-concentration movement in a system under investigation. Of particular interest are the magnitudes of concentrations and discharges at interfaces with the environment, for example, in cases of aquifer contamination. Naturally, the quality or degree of realism of a given simulation is strongly dependent on the quantity and quality of the parameter distribution, boundary-condition, and source-sink data. Acquiring this data can be a major task of the modeling project.

1.2. APPLICABILITY AND LIMITATIONS

The HST3D code is suitable for simulating ground-water flow and the associated heat and solute transport, in saturated, three-dimensional flow systems with variable density and viscosity. As such, the code is applicable to the study of waste injection into saline aquifers, landfill-contaminant movement, seawater intrusion in coastal regions, brine disposal, fresh-water storage in saline aquifers, heat storage in aquifers, liquid-phase geothermal systems, and similar transport situations. If desired, only the ground-water flow or only the heat- or the solute-transport equation may be solved in conjunction with ground-water flow. Three-dimensional cartesian or axisymmetric, cylindrical-coordinate systems are available.

The primary limitation of this code results from the use of finite-difference techniques for the spatial- and temporal-derivative approximations.

Where longitudinal and transverse dispersivities may be small, cell sizes will need to be small to minimize numerical dispersion or oscillation. Furthermore, if the region of solute movement is somewhat convoluted and three-dimensional, the projection of nodal lines from regions of high-nodal density will cause more nodes than are needed to appear in other regions. These two factors can combine to cause an excessive number of nodes to be involved for a given simulation, thus making the simulation prohibitively expensive because of computer-storage and computation-time requirements. In such cases, a simple model of the system, useful for investigating mechanisms and testing hypotheses, may be all that is practical.

Another limitation results from a phenomenon called grid-orientation effect (Aziz and Settari, 1979, p. 332), whereby numerical simulations of miscible displacement converge to two separate solutions, as the mesh size is refined, depending on whether the major velocity vectors are parallel to one of the coordinate directions or are diagonally oriented. The effect is more pronounced for conditions of little dispersion or piston-like displacement of the solute, and for conditions of the viscosity of the displacing fluid much less than the viscosity of the displaced fluid. The effect virtually is absent if the two viscosities are nearly equal, or if the dispersion coefficient is large. The primary cause of the grid-orientation effect appears to be the use of a seven-point difference formula for the three-dimensional-flow and solute-transport equations, because this formula restricts transport in the diagonal directions. Use of a grid where the major velocity vectors are oriented parallel to one of the coordinate directions, has been found to give more realistic simulation results (Aziz and Settari, 1979, p. 336). To completely eliminate this problem, a higher-order differencing scheme, or curvilinear coordinates need to be used, but these modifications are beyond the scope of the present version of HST3D.

There is a limitation on which boundary conditions can be used with a tilted coordinate system. The free surface and leakage boundary conditions require that the z-axis be oriented in the vertical direction.

A limitation that is secondary for most ground-water flow and transport modeling is that two types of transport phenomena exist that this type of numerical simulation has difficulty in representing quantitatively. The first phenomenon, viscous-fingering instabilities, may occur during the displacement of a resident fluid by an injected fluid with significantly less viscosity. The injected fluid forms channels or fingers through the resident fluid, as described by Aronofsky (1952), Saffman and Taylor (1958), and Sheidegger (1960). The second phenomenon may occur in the situation where a fluid of greater density overlies one of lesser density. Rayleigh-Taylor convective cells are formed that mix the two fluids (Wooding, 1959). Numerical simulation tends to predict these transport instabilities later than they occur in laboratory-scale experiments. When perturbations are present to initiate the instabilities, the general magnitudes often are calculated to be less than those that actually occur (Scheidegger and Johnson, 1963; and Dougherty, 1963). However, laboratory-scale viscous fingering and convective-cell formation may be much more unstable than the corresponding field-scale phenomenon, because of the smaller dispersivity at the laboratory scale. Therefore, at the field scale, numerical simulation may not be so much in error in representing these instabilities. Nevertheless, these limitations need to be be kept in mind when simulating fluid flow with large viscosity or density contrasts.

Another secondary limitation is that this is a rather general computer code. The variety of discretization, boundary-condition, and source-sink options make this code not as computationally efficient as a simulation code designed specifically for a given system being investigated. This limitation is compensated by the ability of the HST3D simulator to represent a wide variety of physical situations.

1.3. PURPOSE AND SCOPE

The purpose of this documentation is to provide the user with information on the theory, assumptions, and equations being numerically solved, the numerical-solution methods employed, and the various program options available. The sets of verification test problems are presented and two example problems are described in detail with input and output files. Sections on the code organization, input information, and output information, as well as a list of variable-definitions and a cross-reference map are provided. The documentation is intended to be sufficiently complete and understandable so the user easily can obtain successful simulations, diagnose most computational problems, develop remedies, and incorporate minor program additions or modifications to suit specific modeling needs.

Each release of the HST3D program code is identified by a release number. This documentation is for release 1.0, and this number will change as modifications, corrections, and additions are made to the program. Updates to the documentation will be keyed to the release number.

1.4. ACKNOWLEDGMENTS

The contributions of J.E. Carr, J.B. Gillespie, and A.H. Welch, of the U.S. Geological Survey who provided application problems that influenced program development, and of R.T. Miller and M.L. Merritt, also with the U.S. Geological Survey, who helped with program testing, are gratefully acknowledged.

2. THEORY

2.1. FLOW AND TRANSPORT EQUATIONS

Derivation of the saturated ground-water flow and heat- and solute-transport equations solved by this program can be found in references such as Bear (1972) or Huyakorn and Pinder (1983). Only the assumptions leading to these equations will be presented here. Explanations of the notation will appear after the first usage. A complete table of notation appears in chapter 9. In the report, all variables will be given with metric (SI) units of measure.

2.1.1. Ground-Water Flow Equation

The partial-differential equation of ground-water flow is based on the following assumptions:

- Ground water fully saturates the porous medium within the region of ground-water flow.
- Ground-water flow is described by Darcy's Law.
- The porous medium is compressible.
- The fluid is compressible.
- The porosity and permeability are functions of space.
- The coordinate system is chosen to be alined with the principal directions of the permeability tensor so that this tensor is diagonal for anisotropic media.
- The coordinate system is orthogonal as are the principal directions of the permeability tensor.
- The coordinate system is right-handed with the z-axis pointing vertically upward.
- The fluid viscosity is a function of space and time through dependence on temperature and solute concentration.
- Density-gradient diffusive fluxes of the bulk fluid are neglected relative to advective-mass fluxes.

- Dispersive-mass fluxes of the bulk fluid from spatial-velocity fluctuations are not included.
- Contributions to the total fluid-mass balance from pure-solute-mass sources within the region are not included.

Pressure is chosen as the dependent variable for fluid flow, because no potentiometric-head function exists for density fields that depend on temperature and solute concentration. All pressures denoted by p are expressed relative to atmospheric pressure. Absolute pressures are denoted by $\hat{p}$. The flow equation is based on the conservation of total fluid mass in a volume element, coupled with Darcy's Law for flow through a porous medium. Thus:

$$\frac{\partial(\varepsilon\rho)}{\partial t} = \nabla \cdot \rho \frac{\underline{\underline{k}}}{\mu}(\nabla p + \rho g) + q\rho^{*} \; ; \qquad (2.1.1.1a)$$

where

p is the fluid pressure (Pa);

t is the time (s);

ε is the effective porosity (-);

ρ is the fluid density (kg/m^3);

ρ^* is the density of a fluid source (kg/m^3);

$\underline{\underline{k}}$ is the porous-medium permeability tensor (m^2);

μ is the fluid viscosity (kg/m-s);

g is the gravitational constant (m/s^2); and

q is the fluid-source flow-rate intensity (m^3/m^3-s); (positive is into the region).

Equation 2.1.1.1a relates the rate of change of total mass in the fluid phase to net fluid-inflow rate, and source fluid-and-solute flow rate. Note that the density of the fluid source is ρ^* for $q>0$, and ρ for $q<0$.

The interstitial or pore velocity, $\underline{v}$ is obtained from Darcy's Law as:

$$\underline{v} = - \frac{\underline{\underline{k}}}{\varepsilon\mu} (\nabla p + \rho g) \; ; \qquad (2.1.1.1b)$$

where

$\underline{v}$ is the interstitial-velocity vector (m/s).

2.1.2. Heat-Transport Equation

The thermal-energy-balance equation, used for heat transport, is based on the following assumptions:

- Fluid kinetic energy is negligible.
- Thermal-dispersive transport takes place with a mechanism analogous to solute-dispersive transport.
- Thermal conduction occurs through the fluid and porous medium in parallel.
- Radiant-energy transfer is neglected.
- Thermal effects of chemical reactions are neglected.
- Changes in gravitational energy from diffusive and dispersive fluxes of solute species are neglected.
- Heating from viscous dissipation is neglected.
- Heat capacities are not a function of temperature or solute concentration.
- Thermal conductivities are not functions of temperature or solute concentration.
- Thermal equilibrium exists between the fluid and solid phases.
- Energy transport by a diffusive flux of solute is neglected.
- Only a single fluid phase exists.
- Pressure equilibrium exists between the fluid and porous-medium phases.
- Changes in fluid enthalpy with pressure, that is, pressure volume work, reversible work, or flow work, as a parcel of fluid moves are neglected.

- The velocity of the porous medium during compression or expansion is neglected.
- Enthalpy dependence on solute concentration is accounted for by a heat-capacity adjustment.
- The thermal expansion of the porous medium is neglected.

The energy equation is based upon the conservation of enthalpy in both the fluid and solid or porous-medium phases of a volume of the region. Enthalpy is a derived property containing both internal energy and flow energy. Temperature is the dependent variable. Thus:

$$\frac{\partial}{\partial t} (\varepsilon\rho c_f + (1-\varepsilon)\rho_s c_s)T = \nabla\cdot(\varepsilon K_f + (1-\varepsilon)K_s)\underline{\underline{I}}\ \nabla T$$

$$+ \nabla\cdot\varepsilon\underline{\underline{D}}_H\ \nabla T - \nabla\cdot\varepsilon\rho c_f\underline{v}T$$

$$+ q_H + q\rho^* c_f T^* \ ; \qquad (2.1.2.1)$$

where

T is the fluid and porous-medium temperature (°C);
T^* is the temperature of the fluid source (°C);
ρ_s is the density of the solid phase (kg/m^3);
c_f is the heat capacity of the fluid phase at constant pressure (J/kg-°C);
c_s is the heat capacity of the solid phase at constant pressure (J/kg-°C);
K_f is the thermal conductivity of the fluid phase (W/m-°C);
K_s is the thermal conductivity of the solid phase (W/m-°C);
$\underline{\underline{D}}_H$ is the thermo-mechanical dispersion tensor (W/m-°C);
q_H is the heat-source rate intensity (W/m^3); and
$\underline{\underline{I}}$ is the identity matrix of rank 3 (-).

Equation 2.1.2.1 relates the rate of change of fluid and porous-medium enthalpy to the net conductive-enthalpy flux, to the net dispersive enthalpy flux, to the net advective-enthalpy flux, to the heat source, and to the fluid source at a given temperature. It is written for a unit volume of fluid and solid phase together; that is, a unit volume of saturated, porous medium. Heat is injected at temperature, T^*, and density, ρ^*, by a fluid source; but heat is withdrawn at temperature, T, and density, ρ by a fluid sink. A detailed derivation of equation 2.1.2.1 is given in Faust and Mercer (1977).

2.1.3. **Solute-Transport Equation**

The equation for conservation of a single solute species is based on the following assumptions:

- Thermal diffusion is neglected.
- Pressure diffusion is neglected.
- Solute transport by local, interstitial, velocity-field fluctuations and mixing at pore junctions is described by a hydrodynamic-dispersion coefficient.
- Forced diffusion by gravitational, electrical, and other fields is neglected.
- The only reaction mechanism is linear decay or disappearance of solute.
- The only solute, porous-medium, interaction mechanism is linear-equilibrium sorption.
- No pure solute sources occur in the fluid or solid phases.

The solute mass fraction is taken to be the dependent variable because the density field is variable. It is an amount per unit mass of fluid, that is, a mass-based concentration. The more widely used concentration term is an amount per unit volume of fluid; that is, a volume-based concentration. But volume-based concentration is not conserved in a variable-density system. The term "solute concentration," used in this report, will refer to the mass-based concentration or mass fraction. The conservation equation for the solute in the fluid phase can be written:

$$\frac{\partial(\varepsilon\rho w)}{\partial t} = \nabla\cdot\varepsilon\rho\underline{\underline{D}}_S\nabla w + \nabla\cdot\varepsilon\rho D_m\underline{\underline{I}}\nabla w - \nabla\cdot\varepsilon\rho\underline{v}w - \lambda\varepsilon\rho w$$

$$-\rho_b R_{fs} + q\rho^* w^* \; ; \qquad (2.1.3.1a)$$

where

w is the mass fraction of solute in the fluid phase (-);

w^* is the mass fraction of solute in the fluid source (-);

$\underline{\underline{D}}_S$ is the mechanical-dispersion-coefficient tensor (m^2/s);

D_m is the effective-molecular diffusivity of the solute (m^2/s);

λ is the linear-decay rate constant (s^{-1});

R_{fs} is the transfer rate of solute from fluid to solid phase per unit mass of solid phase (kg solute/s·kg solid phase); and

ρ_b is the bulk density of the porous medium (kg/m^3).

A similar conservation equation can be written for the solute in the solid phase:

$$\frac{\partial(\rho_b\bar{w})}{\partial t} = \rho_b R_{fs} - \lambda\rho_b\bar{w} \; , \qquad (2.1.3.1b)$$

where

$\bar{w}$ is the mass fraction of solute on the solid phase (-).

The solute is immobile when it is on the solid phase. Under the assumption of linear-equilibrium sorption, the fluid-phase and solid-phase concentrations can be related by an equilibrium-distribution coefficient:

$$\bar{w} = K_d\rho w \; ; \qquad (2.1.3.1c)$$

where

K_d is the equilibrium-distribution coefficient (m^3/kg).

By combining equations 2.1.3.1a-c, we obtain the final solute-conservation equation:

$$\frac{\partial}{\partial t}(\varepsilon + \rho_b K_d)\rho w = \nabla\cdot\varepsilon\rho[\underline{\underline{D}}_S + D_m\underline{\underline{I}}]\nabla w - \nabla\cdot\varepsilon\rho\underline{v}w - \lambda(\varepsilon+\rho_b K_d)\rho w + q\rho^* w^* \; ; \quad (2.1.3.2)$$

Equation 2.1.3.2 relates the rate-of-change of solute in the fluid phase to the net dispersive and diffusive flux, the net advective flux, the solute-source rate, the solute-injection rate with a fluid source, and the solute-decay rate. The equation is written for a unit volume of fluid and solid phase together; that is, a unit volume of saturated porous medium. Note that solute is injected into the sytem at concentration, w^*, and density, ρ^*, by a fluid source; but that solute is withdrawn at concentration w, and density ρ, by a fluid sink; that is, $w^* = w$, if $q<0$.

2.2. PROPERTY FUNCTIONS AND TRANSPORT COEFFICIENTS

Before the three conservation equations can be solved, information about the fluid properties, porous-matrix properties, and transport coefficients need to be obtained. The fluid properties are density, viscosity, heat capacity, thermal conductivity, and reference-state enthalpy. The porous-matrix properties are porosity, compressibility, permeability, heat capacity, thermal conductivity, and reference-state enthalpy. The transport coefficients are heat- and solute-dispersion tensors, and the effective molecular diffusivity, decay and sorption coefficients of the solute. In the HST3D simulator, density, viscosity, and porosity are functions of the dependent variables: pressure, temperature, and solute-mass fraction. The heat- and solute-dispersion tensors are functions of space and the interstitial velocity. The other parameters are either uniform or functions of space within the simulation region.

2.2.1. Fluid-Density Function

Fluid density is assumed to be a function of pressure, temperature, and solute concentration. For fluids such as water, a linear-density function is usually adequate over the ranges of pressures, temperatures, and solute concentrations encountered. Thus, the fluid-density function incorporated into this simulation code is:

$$\rho(p,T,w) = \rho(p_o,T_o,w_o) + \frac{\partial\rho}{\partial p}\Big|_o (p-p_o) + \frac{\partial\rho}{\partial T}\Big|_o (T-T_o) + \frac{\partial\rho}{\partial w}\Big|_o (w-w_o) \; ; \qquad (2.2.1.1a)$$

or

$$\rho(p,T,w) = \rho_o + \rho_o\beta_p(p-p_o) - \rho_o\beta_T(T-T_o) + \rho_o\beta_w(w-w_o) \; ; \qquad (2.2.1.1b)$$

where

ρ_o is the fluid density at a reference pressure, p_o, temperature, T_o, and mass fraction, w_o, (kg/m^3);

β_p is the fluid compressibility (Pa^{-1});

β_T is the fluid coefficient of thermal expansion (°C^{-1}); and

β_w is the slope of the fluid density as a function of mass fraction divided by the reference fluid density (-).

Now $\rho_o\beta_w$ is given by:

$$\rho_o\beta_w = \frac{\rho(w_{max}) - \rho(w_{min})}{w_{max} - w_{min}}\Big|_{p_o,T_o} \qquad (2.2.1.1c)$$

where

w_{min} is the minimum solute-mass fraction (-); and

w_{max} is the maximum solute-mass fraction (-).

The user needs to specify w_{min} and w_{max} along with $\rho(w_{min})$ and $\rho(w_{max})$. The minimum solute-mass fraction usually will be determined by the initial conditions. If linear decay is present, w_{min} must be zero. The maximum solute-mass fraction usually will be determined by source or boundary conditions because none of the transport processes incorporated in the HST3D simulator will concentrate solute in the fluid phase. For simplicity, w_0 is taken to be equal to w_{min}.

The option is available in HST3D to use a scaled, solute, mass fraction defined by:

$$w' = \frac{w - w_{min}}{w_{max} - w_{min}} ; \qquad (2.2.1.2)$$

where

w' is the scaled solute-mass fraction (-);

The scaled solute-mass fraction also is dimensionless and ranges from 0 to 1. Commonly, for input and output of mass-fraction data, it is more convenient to deal with a scaled solute-mass fraction rather than an absolute value. With a scaled solute-mass fraction, equation 2.2.1.1b becomes:

$$\rho(p,T,w') = \rho_o + \rho_o\beta_p(p-p_o) - \rho_o\beta_T(T-T_o) + \rho_o\beta_w'w', \qquad (2.2.1.3a)$$

where

$$\rho_o\beta_w' = \rho(w_{max}) - \rho(w_{min}) . \qquad (2.2.1.3b)$$

The errors caused by assuming constant values for fluid compressibility, coefficient of thermal expansion, and variation of density with solute concentration can be assessed by looking at a density table for salt brines (Perry and others, 1963, p. 3-77).

Over a temperature range of about 100 °C and a solute-mass fraction range of 20 percent, the coefficient of thermal expansion varies by 60 percent and the density-concentration coefficient, β_w, varies by about 10 percent (Perry and others, 1963, p. 3-77). The variation of the fluid compressibility could not be checked because of lack of data. However, the density dependence on pressure for nearly incompressible fluids like water is much less than the density dependence on temperature or solute concentration. Therefore, some error will be introduced into the simulations by the linear-density function where large variations in temperature and solute concentration are involved.

The relative importance of pressure, temperature, and solute concentration for density variation can be seen from the salt-brine density table given in Perry and others (1963) and the compressibility of water. A change in pressure of 10^6 Pa results in a density change of about 0.04 percent, whereas a change in temperature of 100 °C results in a density change of about 4 percent, but a change in solute-mass fraction of 0.25 results a density change of about 20 percent. Thus, the salt concentration has the greatest effect on the density for typical ranges of the variables.

2.2.2. Fluid-Viscosity Function

Fluid viscosity is strongly dependent on temperature, and, to a lesser extent, on solute concentration. The viscosity dependence on pressure is neglected. The viscosity as a function of temperature and scaled-solute concentration is written as:

$$\mu(T,w') = 10^{-3}\ \mu\ (T_{ov},w')\ \exp\left[(B_0 w' + B_1(1-w'))\ (\frac{1}{T} - \frac{1}{T_{ov}})\right] , \quad (2.2.2.1)$$

where

$\mu(T_{ov},w')$ is the fluid viscosity at the reference temperature (kg/m-s);

B_0, B_1 are parameters describing the temperature dependence of viscosity at the concentration extremes (°C); and

T_{ov} is the reference temperature for viscosity (°C).

The scaled solute-mass fraction of equation 2.2.1.2 is used in the viscosity function as well as the density function. The parameters B_0 and B_1 are obtained from a least-squares fit of viscosity versus temperature data. If data are available only at a single temperature, the generalized viscosity versus temperature graph of Lewis and Squire as given in Perry and others (1963, p. 3-228) is used.

The concentration extremes are chosen to be the same minimum and maximum mass fractions described in section 2.2.1. The variation of viscosity with solute-mass fraction is specified in tabular form by the user. If viscosity data at only the minimum and maximum mass-fraction values are available, the equation used for viscosity as a function of concentration at a given temperature is:

$$\mu(w') = \mu_1(T_{ov})^{w'} \, \mu_0(T_{ov})^{1-w'} \, , \qquad (2.2.2.2)$$

where

μ_0 is the viscosity at the minimum-mass fraction or scaled concentration of zero (kg/m-s); and

μ_1 is the viscosity of the maximum-mass fraction or scaled concentration of one (kg/m-s).

Equation 2.2.2.2 is used with equation 2.2.2.1 or alone in the case of isothermal simulation.

The viscosity versus temperature and concentration data that could be available may be divided into three classes. Class 1 is the greatest amount available, namely $\mu(T)$ at w_{min} and w_{max} and $\mu(w)$ for a range of w from w_{min} to w_{max}. Class 2 is viscosity versus temperature, $\mu(T)$, at only w_{min} and w_{max}. Class 3 is the least amount of data required, namely two viscosity points at a given temperature at w_{min} and w_{max}.

An evaluation of the accuracy of viscosity functions given in equations 2.2.2.1 and 2.2.2.2 was presented by INTERCOMP Resource Development and

Engineering, Inc. (1976). They found errors ranging from 5 to 14 percent over the temperature range from freezing to boiling for pure water. For a solution of sodium chloride with a mass fraction ranging from 0.0 to 0.24, the different amounts of data available resulted in errors from 5 to 18 percent at a temperature of 65 °C. A sucrose solution with mass fractions ranging from 0.0 to 0.5 showed a maximum viscosity error of 30 percent. Other viscosity functions of temperature and solute concentration may be more suitable for certain situations.

2.2.3. Fluid Enthalpy

Fluid-phase enthalpy is a function of pressure, temperature, and solute concentration. The present version of the HST3D code uses the enthalpy of pure water obtained from the steam tables of Keenan and others (1969, p. 2-7 and 104-107), which can be described as:

$$H(\hat{p},T) = H(\hat{p}_{sat},0) + \int_{\hat{p}_{sat}}^{\hat{p}} \frac{1}{\rho} \, [1-\hat{T}\beta_T]d\hat{p} + \int_{o}^{T} c_{fo}dT; \qquad (2.2.3.1a)$$

where

H is the specific enthalpy of the fluid phase (J/kg);

$\hat{p}$ is the absolute pressure (Pa);

$\hat{p}_{sat}$ is the absolute pressure at saturation (Pa); and

$\hat{T}$ is the absolute temperature (K).

c_{fo} is the heat capacity of pure water at constant pressure (J/kg-°C).

The reference state for the enthalpy tables is saturated liquid water at 0 °C where the reference enthalpy is taken to be zero (Van Wylen, 1959, p. 80). The variation of enthalpy with solute concentration is treated in an approximate fashion, by adjusting the pure-water enthalpy by a factor that is the ratio of the heat capacity of the solution to the heat capacity of pure water at 0 °C, and by using an average heat capacity for the range of solute concentrations to be simulated. The heat capacity is assumed independent of temperature and pressure.

Thus,

$$H(p,T,w) = H(p,T,0)\ (\overline{c_f(w)}/c_{fo}) \qquad (2.2.3.1b)$$

where

$\overline{c_f(w)}$ is an average heat capacity (J/kg-°C).

During the simulations, the enthalpy is calculated as a variation from a reference state described by a pressure, p_{oH}, and a temperature, T_{oH}, selected by the user. The reference state is pure water so the reference mass fraction, w_{oH}, is always zero. Thus, the enthalpy equation becomes:

$$H(p,T,w) = H(\hat{p}_{oH}, T_{oH},0)\ (\overline{c_f}/c_{fo}) + \int_{p_{oH}}^{p} [1-\hat{T}\beta_T]\ \frac{dp}{\rho} + \int_{T_{oH}}^{T} \overline{c_f} dT \ ; \qquad (2.2.3.1c)$$

where

p_{oH} is a reference pressure for enthalpy (Pa);
$\hat{p}_{oH}$ is the corresponding absolute pressure (Pa); and
T_{oH} is a reference temperature for enthalpy (°C).

The $\hat{T}\beta_T$ term may be neglected for temperatures less than 100 °C (373 K) and density may be regarded as constant for pressure changes less than 10^8 Pa. The chosen reference pressure and temperature needs to be within the range to be calculated during the simulation. The heat capacity of the fluid needs to be an average value over the solute-concentration range to be simulated. More sophisticated treatments of the enthalpy of fluid mixtures are available in the literature; for example, Hougen and others (1959, p. 879).

2.2.4. **Porous-Medium Enthalpy**

Enthalpy of the porous medium is taken to be a function of only temperature in the following form:

$$H_s = H_s(T_{oH}) + c_s(T-T_{oH}) ; \tag{2.2.4.1}$$

where

H_s is the specific enthalpy of the solid phase (porous matrix) (J/kg); and

c_s is the heat capacity of the solid phase (porous matrix) (J/kg-°C).

Often, the enthalpy of the porous matrix is taken to be zero at a reference state of 0 °C.

2.2.5. **Porous-Medium Compressibility**

Many types of compressibility for porous media have been defined (Bear, 1972, p. 52, 203-213; Thomas, 1982, p. 34, 40). The porous-medium bulk compressibility, α_b (Pa^{-1}), is defined on a volumetric basis (Bear, 1972, p. 56; Eagleson, 1970, p. 268), assuming confined-aquifer conditions, and one-dimensional, vertical consolidation of the porous matrix, as:

$$\alpha_b = \frac{1}{V_b}\frac{\partial V_b}{\partial p} ; \tag{2.2.5.1}$$

where

V_b is the bulk or total volume of a fixed mass of porous medium, that is, fluid plus porous matrix (m^3).

Petroleum-reservoir engineers use the term rock compressibility, α_r (Pa^{-1}), defined as (Thomas, 1982, p. 34):

$$\alpha_r = \frac{1}{\varepsilon}\frac{\partial \varepsilon}{\partial p} \tag{2.2.5.2}$$

Rock compressibility directly expresses the variation of porosity with pressure. It is related to bulk compressibility by:

$$\alpha_r = \frac{(1-\varepsilon)}{\varepsilon} \alpha_b ; \qquad (2.2.5.3a)$$

for the case of a nondeforming control volume, where more porous medium enters the control volume, as compression takes place. It is related by:

$$\alpha_r = \frac{\alpha_b}{\varepsilon} \qquad (2.2.5.3b)$$

for the case of a deforming control volume, or where impermeable medium enters a nondeforming control volume, as compression takes place.

By combining equations 2.2.5.2 and 2.2.5.3b we obtain:

$$\frac{\partial \varepsilon}{\partial p} = \alpha_b ; \qquad (2.2.5.4)$$

which relates bulk compressibility to changes in porosity with changes in pressure.

Thus we have allowed the control volume to deform as the porous matrix and the fluid specific volumes expand or contract with changes in pressure. However, we neglect the velocity of deformation, so that the interstitial-pore velocity is calculated with respect to the fixed-coordinate system.

The specific storage is related to the compressibilities of the fluid and porous medium by (Eagleson, 1970, P. 270):

$$S_o = \rho g(\alpha_b + \varepsilon \beta_p) \qquad (2.2.5.5)$$

where

S_o is the specific storage (m^{-1}).

However, it is more convenient for our purposes to employ the compressibility parameters, because of the variable density.

2.2.6. Dispersion Coefficients

2.2.6.1. Solute Dispersion

Hydrodynamic dispersion is the name for the group of mixing mechanisms that occur on the micro or pore scale that cause the irreversible spreading of a solute tracer that is observed at the macro or field scale for the system. As described by Bear (1972, p. 580-581), flow within the porous-medium structure has variations in local flow velocity, because of the velocity profile across the pore and mixing at pore junctions. The macroscopic effect is mechanical dispersion of a tracer. Molecular diffusion also is present where solute-tracer concentration gradients exist. However, diffusion in liquids is a relatively slow process, producing significant transport rates only at very slow ground-water flow velocities. In a laminar flow regime within the pores, diffusion of solute from one flow path to another contributes to the dispersion, so the separation of dispersion into a mechanical and diffusive mechanisms is somewhat artificial. For an extensive discussion of dispersion theory and a review of previous work, see Bear (1972, ch. 10).

The form of the hydrodynamic-dispersion-coefficient tensor D^*_{Sij} (m^2/s) for the heat- and solute-transport simulation model is assumed to be, in component form:

$$D^*_{Sij} = D_{Sij} + D_m \delta_{ij} \; ; \tag{2.2.6.1.1}$$

where

D_{Sij} is the mechanical-dispersion-tensor component (m^2/s);
D_m is the effective molecular-diffusion coefficient (m^2/s); and
δ_{ij} is the Kronecker delta function.

The effective molecular-diffusion coefficient is the liquid-phase molecular diffusivity multiplied by an attenuation factor that accounts for the effect of the tortuosity of the porous medium. The form of the mechanical-dispersion coefficient is taken from the work of Scheidegger (1961) and Bear (1961) as presented by Konikow and Grove (1977) and Bear (1972, ch. 10). For an isotropic porous medium, two parameters describe the mechanical-dispersion tensor, the longitudinal dispersivity, α_L (m), and the transverse dispersivity, α_T (m). Then the nine components of the mechanical-dispersion tensor are given by:

$$D_{Sij} = (\alpha_L - \alpha_T) \frac{v_i v_j}{v} + \alpha_T v\delta_{ij} \; ; \qquad (2.2.6.1.2)$$

where

v_i is the component of interstitial velocity in the ith direction (m/s);

and $$v = (v_1^2 + v_2^2 + v_3^2)^{\frac{1}{2}} \; ; \qquad (2.2.6.1.3)$$

where

v is the magnitude of the velocity vector (m/s).

In general, the subscript 1 is associated with the x direction; the subscript 2 is associated with the y direction; and the subscript 3 is associated with the z direction. Field data have shown that longitudinal dispersivity usually is 3 to 10 times larger than transverse dispersivity (Freeze and Cherry, 1979, p. 396; Anderson, 1979), and that their magnitudes are dependent on the scale of observation distance over which the tracer is transported in the system.

Note that while flow in the porous medium may be governed by an anisotropic-permeability tensor, dispersion for heat and solute transport is assumed to be described by a dispersion tensor that applies to an isotropic-porous medium. This assumption is made because it is not feasible to obtain all the dispersivity parameters for an anisotropic medium. If dispersive transport is a second-order effect, relative to advective transport, this inconsistency should not introduce serious errors. In most cases, the errors should be less than those introduced by uncertainties in the dispersion parameters themselves.

When the longitudinal and transverse dispersivities are not equal, dispersive transport will cause a solute distribution to enlongate in the direction of flow, because the longitudinal dispersivity always is greater than or equal to the transverse dispersivity. Thus, anisotropic spreading of solute and heat can occur in an isotropic-porous medium, even under conditions of uniform, unidirectional flow.

2.2.6.2. Thermal Dispersion

A description of thermal dispersion is based on a direct analogy with solute dispersion. Energy replaces solute mass as the quantity being transported by mechanical dispersion, and thermal conduction replaces molecular diffusion. Thus, the thermo-mechanical dispersion tensor is derived from the mechanical dispersion tensor by:

$$D_{Hij} = \rho c_f D_{Sij} \; ; \qquad (2.2.6.2.1)$$

where

D_{Hij} is the thermo-mechanical-dispersion tensor component (W/m-°C).

Combining the thermo-mechanical dispersion tensor with the net thermal conductivity of the fluid and solid phases gives the thermo-hydrodynamic-dispersion coefficient tensor, D^*_{Hij} (W/m-°C), in component form:

$$D^*_{Hij} = D_{Hij} + [\varepsilon K_f + (1-\varepsilon)K_s]\delta_{ij} \qquad (2.2.6.2.2)$$

2.3. EXPANDED SYSTEM EQUATIONS

When the density function, equation 2.2.1.1b, and the porous-medium compressibility relation, equations 2.2.5.3a and 2.2.5.3b are incorporated into the system governing equations, the following expanded system equations are obtained:

For ground-water flow:

$$\varepsilon\rho_o\beta_p \frac{\partial p}{\partial t} + \varepsilon\rho_o\beta_T \frac{\partial T}{\partial t} + \varepsilon\rho_o\beta_w \frac{\partial w}{\partial t}$$

$$+ \rho\alpha_b \frac{\partial p}{\partial t} = \nabla\cdot\rho \frac{\underline{\underline{k}}}{\mu} (\nabla p + \rho g) + q\rho^* ; \qquad (2.3.1a)$$

For heat transport:

$$\varepsilon\rho_o\beta_p c_f T \frac{\partial p}{\partial t} + \varepsilon\rho_o\beta_T c_f T \frac{\partial T}{\partial t}$$

$$+ \varepsilon\rho_o\beta_w c_f T \frac{\partial w}{\partial t} + \rho\ \alpha_b c_f T \frac{\partial p}{\partial t}$$

$$+ \varepsilon\rho c_f \frac{\partial T}{\partial t} - \rho_s c_s T\ \alpha_b \frac{\partial p}{\partial t} + (1-\varepsilon)\rho_s c_s \frac{\partial T}{\partial t}$$

$$= \nabla\cdot(\varepsilon K_f + (1-\varepsilon)K_s)\underline{\underline{I}}\ \nabla T$$

$$+ \nabla\cdot\varepsilon\underline{\underline{D}}_H\ \nabla T - \nabla\cdot\varepsilon\rho c_f \underline{v} T$$

$$+ q_H + q\rho^* c_f T^* ; \qquad (2.3.1b)$$

For solute transport:

$$\rho_o \beta_p (\varepsilon + \rho_b K_d) \quad w \frac{\partial p}{\partial t} + \rho_o \; \beta_T (\varepsilon + \rho_b K_d) \quad w \frac{\partial T}{\partial t}$$

$$+ \rho_o \beta_w (\varepsilon + \rho_b K_d) \; w \frac{\partial w}{\partial t}$$

$$+ \rho \; \alpha_b \; w \quad \frac{\partial p}{\partial t} + \rho(\varepsilon + \rho_b K_d) \frac{\partial w}{\partial t} \quad = \nabla \cdot \varepsilon \rho [\underline{\underline{D}}_S + D_m \underline{\underline{I}}] \nabla w - \nabla \cdot \varepsilon \rho \underline{v} w$$

$$- \lambda(\varepsilon + \rho_b K_d) \; \rho w + q \rho^* w^* \qquad (2.3.1c)$$

The change in the product of bulk density and equilibrium-distribution coefficient, $\rho_b K_d$, with pressure is zero, because these equations were derived for a fixed mass of porous medium occupying a volume that under-goes slight deformation with variations in pressure. These three expanded equations show the implicit coupling that occurs with variable density and porosity.

2.4. SOURCE OR SINK TERMS--THE WELL MODEL

Most of the ground-water flow and heat and solute sources or sinks affect the simulations through the boundary conditions. However, a line source or sink term is used to represent injection or withdrawal by a well. Although a well is treated as a line source or sink for the flow and transport equations, a well is a finite-radius cylinder for the well-bore model.

The well model for the HST3D simulator is more sophisticated than those well models used in most ground-water flow simulators. A well can be used for fluid injection or fluid withdrawal, with associated heat and solute injection or production. It also can be used simply for observation of aquifer conditions. In the present code, the well bore can communicate with any subset of cells along the z-coordinate direction at a given x-y location. That is, the well may be screened or it may be an open hole over several intervals of its depth. Several options are available for specifying pressure or flow-rate conditions under which the well will operate. A special technique is used to relate the local pressure field around a well to the pressures in the cells with which it communicates. Finally, a mathematical model of the well riser is included to calculate pressure and heat gains and losses as fluid moves from the land surface to the uppermost screened interval, or vice versa.

The well can be divided into two parts as shown in figure 2.1. The lower part, from the bottom of the borehole to the top of the uppermost screened interval, will be referred to as the well bore; the upper part, from the top of the screened interval to the land surface, will be referred to as the well riser. The well-riser interval may or may not have a riser pipe within it, and the well-bore interval may be an open hole or have cased and screened sections. A screened section also may be just perforated casing. The term well-datum level refers to the location at the junction between the well riser and the well bore, equivalently referred to as the bottom hole.

Focusing attention on the well bore, we shall describe the linking of the well model to the simulation region as a source or sink, and then describe the pressure and flow-rate conditions that can be specified as bottom-hole conditions. The incorporation of the well-riser calculations will then be discussed.

Cell or nodal pressures represent a spatially averaged condition, when the simulation region is discretized into finite-difference cells. A well located in a cell will have a pressure at the screen at the nodal elevation that is not necessarily the same as the cell pressure. Various analytical approaches have been used to avoid the computational burden of a finer finite-difference grid around each well in the region. They are summarized by Aziz and Settari (1979, sec. 7.7) and are based on steady-state radial flow in a cylindrical-coordinate system with homogeneous aquifer properties. Another review may be found in Williamson and Chappelear (1981).

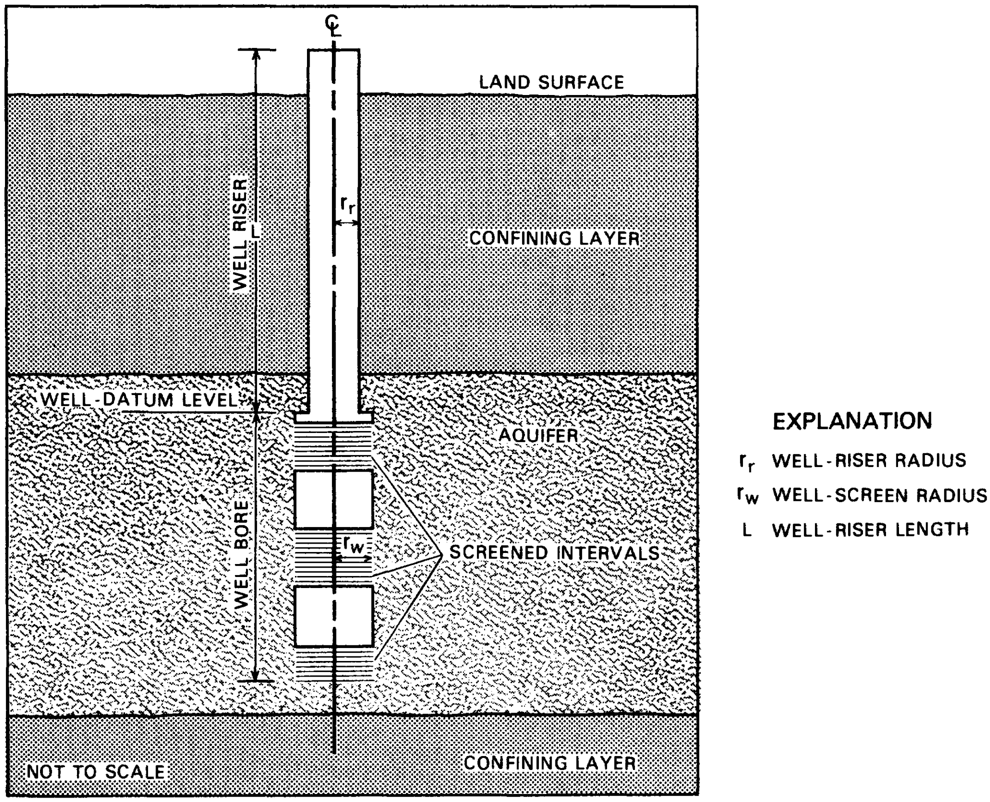


Figure 2.1.--Sketch of well-model geometry showing the well-bore and well-riser sections and the well-datum level.

2.4.1. **The Well-Bore Model**

For three-dimensional cartesian coordinates, the present version of the HST3D code uses a modification of the well-bore equation derived by Van Poolen and others (1968). Consider steady-state radial flow from a well into a homogeneous aquifer with flux across an exterior cylindrical boundary, r_e. This boundary can be regarded as a radius of influence of the well. For a cartesian-coordinate system, the exterior radius, r_e, is taken to be the radius of a circle that encloses the equivalent area to the x-y horizontal area of the cell in which the well is located. The average pressure within the annulus between the well-bore radius and the radius of influence can be calculated and the flow rate from the well per unit length of well bore can be expressed as a function of the pressure change from the well-bore pressure to this average pressure. At any given elevation, z, we have:

$$q_w = \frac{2\pi\ k_w\ (r_e^2 - r_w^2)\ (p_w - p_{av})}{\mu\ [r_e^2\ \ln\ (r_e/r_w) - 0.5\ (r_e^2 - r_w^2)]}\ ; \qquad (2.4.1.1)$$

where

p_w is the pressure at the well bore (Pa);
k_w is the average permeability between r_w and r_e (m^2);
p_{av} is the average pressure between r_w and r_e (Pa);
r_w is the well-bore radius (m);
r_e is the radius of influence of the well (m); and
q_w is the volumetric flow rate per unit length of well bore (positive is flow into the aquifer) (m^3/m-s).

The time-independent factors that affect flow from a well bore can be combined into a single term. Departing slightly from petroleum-reservoir-engineering usage, we define a modified well index as follows:

$$W_I = \frac{2\pi\ k_w\ (r_e^2 - r_w^2)}{r_e^2\ \ln\ (r_e/r_w) - 0.5\ (r_e^2 - r_w^2)}\ ; \qquad (2.4.1.2)$$

where

W_I is the well index per unit length of well bore (m^2).

The average permeability k_w is taken to be:

$$k_w = (k_x k_y)^{\frac{1}{2}} \; ; \qquad (2.4.1.3)$$

for cartesian-coordinate systems, where

k_x is the permeability in the x-direction (m^2); and
k_y is the permeability in the y-direction (m^2).

There is presently no provision for accommodating areally heterogeneous permeability distributions in the vicinity of the well bore.

Equations 2.4.1.1 to 2.4.1.3 will be modified for use with the finite-difference discretization in the numerical-implementation section 3.3.

For three-dimensional and cylindrical regions, the total specified flow rate from the well needs to be allocated over the length of well bore that communicates with the aquifer. This allocation can be done in two ways; by fluid mobility, or by the product of fluid mobility and the pressure difference between the aquifer and the well bore. Although there may be zones of cased well bore through which there is no communication with the aquifer, we shall assume for the present discussion that the well bore is screened throughout its depth. The total well flow rate from the well to the aquifer is given by:

$$Q_w = \int_{\ell_L}^{\ell_U} q_w d\ell \; ; \qquad (2.4.1.4a)$$

$$= \int_{\ell_L}^{\ell_U} \frac{W_I(\ell)}{\mu(\ell)} (p_w - p_{av}) d\ell \; ; \qquad (2.4.1.4b)$$

where

Q_w is the volumetric well flow rate (positive is from the well to the aquifer) (m^3/s);

ℓ is the distance along the well bore (m);

ℓ_L is the lower end of the screened interval (m); and

ℓ_U is the upper end of the screened interval (m).

Fluid mobility at the well can be defined as:

$$M_w(\ell) = \frac{W_I(\ell)}{\mu(\ell)} \quad ; \tag{2.4.1.5}$$

where

M_w is the well mobility per unit length of well bore (m^3/s-m-Pa).

Allocation of the specified flow rate by fluid mobility is obtained by assuming that the pressure difference in equation 2.4.1.4b is independent of depth. Then

$$q_w(\ell) = M_w(\ell) \Big/ \int_{\ell_L}^{\ell_U} M_w(\ell)\, d\ell \; ; \tag{2.4.1.6}$$

represents the allocation of the total flow rate over the well-bore length as a function of fluid mobility.

For wells drilled at an angle, θ_w, to the vertical or z-axis,

$$dz = \cos\theta_w \, d\ell \; ; \tag{2.4.1.7}$$

where

θ_w is the angle between the vertical and the well bore (degrees).

If the screened interval is not continuous from ℓ_L to ℓ_U, the mobility is set to zero over the appropriate subintervals.

The alternative method of flow-rate allocation over the well-bore length is derived by not regarding the pressure difference in equation 2.4.1.4b as constant with depth. A hydrostatic-pressure distribution in the well bore is assumed using an average fluid density. Thus, frictional hydraulic-head losses in the well bore are neglected. This yields, from equation 2.4.1.4b.

$$p_{wd} = \frac{\int_{\ell_L}^{\ell_U} M_w(\ell) \; [p_{av}(\ell) + \rho_w g(z-z_{wd})] d\ell + Q_w}{\int_{\ell_L}^{\ell_U} M_w(\ell) d\ell}; \qquad (2.4.1.8)$$

where

p_{wd} is the bottom-hole or well-datum pressure (Pa);
z_{wd} is the elevation of the well datum (m); and
ρ_w is the average fluid density in the well bore (kg/m^3).

Then the well flow rate is allocated as follows:

$$q_w(\ell) = M_w(\ell) \; [p_{wd} + \rho_w g(z_{wd} - z) - p_{av}]; \qquad (2.4.1.9)$$

This method is referred to as allocation by mobility and pressure difference. The average pressure, p_{av} , will be related to the grid-cell pressures in section 3.3.1 on numerical implementation.

The flow rate can be specified with a bottom-hole pressure-constraint condition, that may affect the source or sink flow rate applied. Allocation is by mobility and pressure difference, and equation 2.4.1.8 is used to calculate a predicted bottom-hole pressure based on the specified flow-rate. For an injection well, if the predicted pressure is greater than the bottom-hole constraint pressure, then the well is pressure-limited, and the flow rate will be less than that specified. The flow rate will be reduced to meet the pressure constraint. If the predicted bottom-hole pressure is less than that specified, then the desired flow rate is used. For a production well, if the predicted bottom-hole pressure is less than the constraint pressure, the well is pressure limited, and the flow rate will be less than desired. Otherwise, the pressure constraint is not limiting. In other words, a well bore can function as either a Dirchlet or a Neumann boundary condition, or it can switch back and forth.

When bottom-hole (well-datum) pressure is specified, equation 2.4.1.9 gives the flow-rate allocation and equation 2.4.1.4b gives the total flow rate. No constraints are applied to the calculated flow rate.

After the flow rate has been established and allocated, heat-injection and solute-injection rates are determined from the bottom-hole pressure, specified-temperature, and specified solute-mass-fraction values. Heat-withdrawal and solute-withdrawal rates are determined by the ambient pressure, temperature, and solute-mass fraction in the aquifer for each cell that communicates with a well bore.

In the case of cylindrical coordinates with a single well at the radial origin, the inner radius of the simulation region becomes the well-bore surface. Thus, a specified flow rate allocated by mobility becomes a specified-flux boundary condition. Allocation by mobility and pressure difference using equation 2.4.1.9 is not applicable here, because the well-bore pressure and the pressure at the inner radius of the region are identical. Instead, the pressure profile along the well bore is not assumed to be hydrostatic, but, rather it satisfies a steady-state momentum equation, that includes frictional pressure losses, but neglects changes in momentum by flow into or out from the well bore. Then, we have, for a differential-momentum balance along the well bore:

$$\frac{dp_w}{dz} + \rho_w g + \frac{\rho_w v^2{}_w}{4r_w} f_w = 0; \qquad (2.4.1.10)$$

where

f_w is the hydraulic-head-loss friction factor (-); and

v_w is the average velocity across the well bore at a given z-level (m/s).

The corresponding mass balance is obtained assuming no change in well-bore storage, thus:

$$\frac{d\rho v_w}{dz} + \frac{2\rho_w q_{Fw}}{r_w} = 0 \; ; \qquad (2.4.1.11)$$

where

q_{Fw} is the volumetric flux from the well bore (m^3/m^2-s).

Equations 2.4.1.10 and 2.4.1.11 can be combined to give:

$$\rho_w q_{Fw} = \frac{d}{dz} \left[\frac{2r_w{}^2}{v_w f_w} \left(\frac{dp_w}{dz} + \rho_w g \right) \right] ; \qquad (2.4.1.12)$$

Equation 2.4.1.12 is combined with the flow equation 2.1.1.1a by assuming that the aquifer pressure and well-bore pressure are equal at the well-bore radius. The flow equation at the inner radius of the region becomes:

$$\frac{\partial \varepsilon \rho}{\partial t} = \nabla \cdot \rho \frac{k}{\mu} (\nabla p + \rho g) + \frac{\partial}{\partial z} \left[\frac{2 r_w^2}{v_w f_w} \left(\frac{\partial p}{\partial z} + \rho_w g \right) \right]; \quad (2.4.1.13)$$

for the parts of the inner radius that are screened. A fluid-flux boundary condition of zero applies over the cased-off intervals.

Thus, the flow equation is still in its original form, but the coefficients of pressure gradient in the z-direction, and of the gravity term, are augmented. The flow rate to or from the well is implicitly incorporated. When the equation is converted to discrete form, the flow rate to or from the well will arise naturally at the upper boundary of the screened interval. The friction-head-loss factor is calculated as described in the well-riser model, section 2.4.2. The magnitude of the friction head-loss factor often may be very small but it needs to be non-zero, for flow to occur in the well bore.

The total flow rate to or from the well always is satisfied by this calculation method, and the pressure at the top of the screened interval in the aquifer is identical to the well-datum-level pressure. Recall that these pressures are not necessarily equal in the line-source approach used with the cartesian coordinate system. An examination of the relative magnitudes of the terms for advective momentum and frictional head-loss in the full momentum-balance equation shows that, for a producing well with uniform inflow per unit length, the advective-momentum term dominates near the bottom of the screen. The frictional head-loss term dominates at distances above the bottom of the screen that are greater than about 1,000 times the well radius. Thus, a significant region exists in which both the momentum and frictional terms are of similar magnitude. However, a more rigorous development, retaining the momentum term, is beyond the scope of this work. The present development follows that of Aziz and Settari (1979, p. 337-341).

2.4.2. The Well-Riser Model

When flow rate or pressure is specified at the land surface for a given well, the well-riser calculation needs to be performed in conjunction with the well bore flow-rate allocation described above. This calculation consists of a simultaneous solution of the macroscopic equations of total energy, momentum and mass (Bird and others, 1960, p. 209-212) for the change in pressure and temperature over the well-riser length.

The total-energy or enthalpy equation is written for steady flow either up or down the well riser as a rate of change with distance along the riser,

$$\frac{dH_r}{d\ell} + g \cos\theta_r + v_r \frac{dv_r}{d\ell} = Q_{Hr}(\ell); \qquad (2.4.2.1)$$

where

H_r is the specific enthalpy of fluid in the riser (J/kg);
v_r is the average velocity across the riser at a given ℓ-location (m/s);
θ_r is the angle between the well riser and vertical (degrees);
Q_{Hr} is the heat transfered per unit mass per unit length to the fluid in the riser (J/kg-m); and
ℓ is the distance along the well-riser casing (m).

Energy loss by viscous dissipation has been neglected. All quantities are averages across the riser-pipe cross section at a given level.

The equation for momentum along the well-riser axis also is written for steady flow as a differential balance along the well riser:

$$2\rho_r v_r \frac{dv_r}{d\ell} + \rho_r g \cos\theta_r + \frac{dp_r}{d\ell} + \frac{\rho_r v_r^2}{2r_r} f_r = 0 \; ; \qquad (2.4.2.2)$$

where

ρ_r is the fluid density in the riser (kg/m^3);

p_r is the pressure in the riser (Pa);

r_r is the internal radius of the well riser (m); and

f_r is the hydraulic-head-loss friction factor (-).

Finally, the macroscopic-mass balance, written in differential form as a rate of change along the riser, is:

$$\rho_r v_r = Q_{Fr} \big/ \pi r_r^2 \ ; \qquad (2.4.2.3a)$$

where

Q_{Fr} is the total mass-flow rate in the riser (kg/s).

Differentiation with respect to length yields:

$$\rho_r \frac{dv_r}{d\ell} + v_r \frac{d\rho_r}{d\ell} = 0 \ . \qquad (2.4.2.3b)$$

To solve equations 2.4.2.1, 2.4.2.2, 2.4.2.3a, and 2.4.2.3b, the enthalpy tables (Keenan and others, 1969, p. 2-7 and 104-107) are used for $H_r(p,T)$, equation 2.7a is used for the density equation of state, and the Fanning friction factor, using the Moody correlation (Perry and others, 1963, p. 5-20), is used to calculate f_r as a function of velocity. The enthalpy for pure water is adjusted for other fluid mixtures according to equation 2.2.3.1b. For turbulent flow, the friction factor is a function of pipe roughness. The user needs to supply a value for pipe roughness, and some typical values for pipe roughness from Shames (1962, p. 300) are given in table 2.1. Changes in viscosity with temperature along the riser are neglected.

Table 2.1.--*Pipe-roughness values*

Pipe type	Pipe roughness (millimeters)
Drawn tubing	1.3×10^{-4}
Steel or wrought iron	3.8×10^{-3}
Galvanized iron	1.3×10^{-2}
Cast iron	2.2×10^{-2}

The heat transferred to the fluid in the riser must pass from the surrounding medium to the riser pipe, then from the riser pipe to the fluid. The heat transferred per unit mass of fluid per unit length of riser is then:

$$Q_{Hr}(\ell) = \frac{2\pi r_r}{Q_{Fr}} U_T(T_a(\ell) - T_r(\ell)) \; ; \qquad (2.4.2.4)$$

where

T_r is the fluid temperature in the well riser (°C);

T_a is the ambient temperature in the medium adjacent to the riser (°C);

U_T is the overall heat-transfer coefficient for the fluid, riser pipe and surrounding medium (W/m^2-°C).

The overall heat-transfer coefficient is given by:

$$\frac{1}{r_r U_T} = \frac{1}{r_r h_r} + \frac{\Delta r_r}{r_r K_r} + \frac{1}{K_{re} F_{CJ}(t)} \; ; \qquad (2.4.2.5)$$

where

Δr_r is the wall thickness of the riser pipe (m);

$F_{CJ}(t)$ is the dimensionless part of the Carslaw and Jaeger (1959, p. 336) solution for heat flux to an infinite medium from a constant-temperature cylindrical source (-);

h_r is the local heat-transfer coefficient from the fluid to the riser pipe (W/m²-°C);

K_{re} is the thermal conductivity of the medium surrounding the riser pipe (W/m-°C); and

K_r is the thermal conductivity of the riser pipe (W/m-°C).

Equation 2.4.2.5 is a simplification of the relation for the overall heat transfer coefficient for conduction through cylindrical walls (Bird and others, 1960, p. 288) combined with the Carslaw-Jaeger solution for heat flux to an infinite medium from a cylindrical source (Carslaw and Jaeger, 1959, p. 336). It is valid for wall thicknesses that are small relative to the riser-pipe radius.

The dimensionless heat-flux function, $F_{CJ}(t)$, can be approximated by the following two series:

(1) For short time, τ, (Carslaw and Jaegar, 1959, p. 336):

$$F_{CJ} \cong F_{CJ}^{S} \; ; \quad \text{for} \quad \tau<1 \; ; \qquad (2.4.2.6a)$$

where

$$F_{CJ}^{S} \cong (\pi\tau)^{-\frac{1}{2}} + \tfrac{1}{2} - \tfrac{1}{4}\left(\frac{\tau}{\pi}\right)^{\frac{1}{2}} + \frac{\tau}{8} \; ; \qquad (2.4.2.6b)$$

and where

τ is the dimensionless time defined by:

$$\tau = \frac{D_{Hrm} t}{(r_r + \Delta r_r)^2} \quad ; \qquad (2.4.2.6c)$$

and where

D_{Hrm} is the thermal diffusivity of the medium surrounding the well riser (m^2/s).

(2) For long times, the asymptotic expansion was derived by Ritchie and Sakakura (1956):

$$F_{CJ} \cong F_{CJ}^{L} \; ; \text{ for } \tau > 3.6 \; ; \qquad (2.4.2.7a)$$

where

$$\begin{aligned} F_{CJ}^{L} = \; & 2(\ell n \; \chi)^{-1}[1 - .5772(\ell n \; \chi)^{-1} - 1.3118(\ell n \; \chi)^{-2} \\ & + .2520 \; (\ell n \; \chi)^{-3} + 3.9969 \; (\ell n \; \chi)^{-4} \\ & + 5.0637 \; (\ell n \; \chi)^{-5}] \\ & + \frac{4}{e^{2\gamma}} \; (\tau \ell n \; \chi)^{-1}[(\ell n \; \chi)^{-1} - 1.1544(\ell n \; \chi)^{-2}] \\ & - 2 \; \tau^{-1}(\ell n \; \chi)^{-3} \; ; \end{aligned} \qquad (2.4.2.7b)$$

$$\chi = \frac{4\tau}{e^{2\gamma}} \; ; \qquad (2.4.2.7c)$$

and where

γ is Euler's constant: $\cong 0.5772$.

In equation 2.4.2.7b, terms of higher order than $(\ell n\ \chi)^{-6}$ and $\tau^{-1}(\ell n\ \chi)^{-3}$ have been dropped. Carslaw and Jaeger (1959, p. 336) present a lower-order version of equation 2.4.2.7b that is accurate for dimensionless time much greater than 3.6. The estimated error is on the order of 10 percent for dimensionless time, τ, greater than 3.6. For a typical rock medium, this truncation means that the time must be greater than about 3.6×10^4 s, or about 0.4 d. The short-time approximation, equation 2.4.2.6b, is good for time less than about 0.1 d. For intermediate time, the heat-transfer function is estimated by linear interpolation between F^S_{CJ} evaluated at τ=1 and F^L_{CJ} evaluated at τ=3.6.

Note that the heat-flux function in equation 2.4.2.5 is a function of time; whereas, the mechanical and thermal-energy balances are at steady-state. This is a consistent approximation, provided it is assumed that the heat transfer from the fluid to the riser pipe and through to its outer boundary is rapid, relative to rates of change in temperature at the fluid-inlet end of the riser pipe; and, that changes in the fluid-temperature profile within the riser pipe re-equilibriate quickly, relative to induced temperature changes in the adjacent medium. This approach parallels that of Ramey (1962), with the difference being that the heat-flux solution from a cylinder at constant temperature is used, instead of the temperature solution for the constant heat-flux case. The former solution is considered to more accurately describe the physical situation.

Values for the local heat-transfer coefficient, h_r in equation 2.4.2.5, can be determined from correlations, such as those of McAdams (1954, p. 241-243) or Sieder and Tate (given in Bird and others, 1960, p. 399), between the Nusselt number, the Prandtl number, and the Reynolds number for forced convection in tubes.

The correlation from McAdams (1954, p. 219) that is valid for turbulent flow in the well-riser pipe is:

$$\frac{2r_r h_r}{K_f} = 0.023 \left[\frac{\rho_r v_r}{\mu_r}\right]^{0.8} \left[\frac{c_f \mu_r}{K_f}\right]^{0.33} ; \qquad (2.4.2.8)$$

where

μ_r is the viscosity of the fluid in the riser pipe (kg/m-s).

The well-riser calculation is developed by combining equations 2.4.2.1-2.4.2.3b with equation 2.2.1.1b and the derivative of equation 2.2.3.1a for the enthalpy function. The resulting equations are:

$$\begin{bmatrix} v_r^2 \beta_p - \frac{1}{\rho_r} & v^2{}_r \beta_T \\ \left.\frac{\partial H}{\partial \rho}\right|_T - \frac{1}{\rho_r} & \left.\frac{\partial H}{\partial T}\right|_p \end{bmatrix} \begin{bmatrix} \frac{dp_r}{d\ell} \\ \frac{dT_r}{d\ell} \end{bmatrix} = \begin{bmatrix} g \cos \theta_r + \frac{v_r^2 f_r}{2r_r} \\ \frac{2\pi r_r U_T}{Q_{Fr}} (T_a - T_r) + \frac{v_r^2 f_r}{2r_r} \end{bmatrix} \qquad (2.4.2.9)$$

Using the thermodynamic relationships:

$$\left.\frac{\partial H}{\partial p}\right|_T = \frac{1}{\rho_r} - T \left.\frac{\partial \rho_r}{\partial T}^{-1}\right|_p ; \qquad (2.4.2.10a)$$

and

$$\left.\frac{\partial H}{\partial T}\right|_p = c_f ; \qquad (2.4.2.10b)$$

we can reduce equation 2.4.2.9 to two simultaneous ordinary differential equations:

$$\begin{bmatrix} \dfrac{dp_r}{d\ell} \\ \\ \dfrac{dT_r}{d\ell} \end{bmatrix} = \begin{bmatrix} c_f & -\beta_T v_r^2 \\ \\ -\beta_T \, T_r/\rho_r & \beta_p v_r^2 - 1/\rho_r \end{bmatrix} \cdot$$

$$\begin{bmatrix} g \cos \theta_r + v_r^2 \, f_r/2r_r \cdot \\ \\ \dfrac{2\pi r_r}{Q_{Fr}}[\dfrac{1}{r_r h_r} + \dfrac{\Delta r_r}{r_r K_r} + \dfrac{1}{K_{re} F_{CJ}(t)}]^{-1} (T_a - T_r) + v_r^2 f_r/2r_r \end{bmatrix}$$

$$\left[(\beta_p v_r^2 - 1/\rho_r)c_f - T_r \beta_T^2 v_r^2/\rho_r\right]^{-1} . \qquad (2.4.2.11)$$

These equations are coupled through the density, velocity, and temperature terms. The boundary conditions are known at one end of the riser. For injection:

$$\text{at} \quad z = z_{LS} \; ; \quad p = p_{inj}; \; T = T_{inj} \; ; \qquad (2.4.2.12a)$$

For withdrawal:

$$\text{at} \quad z = z_{wd} \; ; \quad p = p_{wd}; \; T = T_{wd} \; ; \qquad (2.4.2.12b)$$

where

z_{wd} is the elevation of the well datum (m); and
z_{LS} is the elevation of the land surface (m).

Equations 2.4.2.6a-c and 2.4.2.7a-c are used to evaluate the heat-transfer function F_{CJ}.

The mass, enthalpy, and mechanical-energy-balance equations are solved either up or down along the well riser, depending on the direction of fluid flow, to obtain the pressure and temperature at the riser bottom for injection conditions, or at the riser top for production conditions. Coupling this well-riser calculation to the well-bore model enables specified pressure, temperature, and solute concentration, or specified flow-rate conditions at the land surface, to be employed.

When the flow rate at the land surface is specified as an injection, the surface temperature and solute concentration also need to be specified. The well-riser calculation will give the necessary surface pressure to achieve the specified flow rate. If a production or withdrawal flow rate is specified, the surface pressure, temperature, and solute concentration are determined by the well-bore and well-riser calculations.

When the surface pressure is specified, the well-bore and well-riser calculations determine the flow rate, surface temperature, and solute concentration for a production well. Surface temperature and solute concentration also need to be specified in the case of an injection well. The ambient-temperature profile with depth along the well riser is specified by the user.

A flow rate and pressure constraint at the surface can be specified and the slower of the specified flow rate or the flow rate that results from the specified-pressure constraint will be applied to the aquifer and apportioned as described previously.

A well also can be used as an observation well. In this case, none of the well-bore or well-riser calculations are necessary. The purpose of an observation well is to record dependent variable data (pressure, temperature, solute-mass fraction) for plotting versus time at the conclusion of the simulation. The recorded data are the aquifer values at the well-datum level, which is at the top of the uppermost screened interval.

In summary, a well can be a production well, an injection well, or an observation well. The flow rate can be specified with or without a pressure constraint, or the pressure can be specified either at the land surface or at the well-datum level. For three-dimensional cartesian coordinates, the allocation of the flow to each layer can be determined by the relative mobility of the layer, or by the product of the mobility times the pressure difference. For cylindrical coordinates, the allocation is determined by the product of the mobility times pressure difference, with allowance for gravitational effects, because the well-bore equations are solved simultaneously with the ground-water flow equations for the region adjacent to the screened intervals. Application of the well-flow terms for each layer to the ground-water flow equation can be explicit or semi-implicit in time for three-dimensional cartesian coordinates; it is fully implicit for cylindrical coordinates.

2.5. BOUNDARY CONDITIONS

2.5.1. Specified Pressure, Temperature, and Solute-Mass Fraction

The first type of boundary condition, known as a Dirchlet boundary condition, is a specified pressure condition for the ground-water flow equation, a specified temperature condition for the energy-transport equation, and a specified-mass fraction for the solute-transport equation. These conditions can be specified independently as functions of location and they also can vary independently with time. Mathematically, we have:

$$p = p_B\ (\underline{x},t), \text{ for } \underline{x} \text{ on } S_p^1\ ; \tag{2.5.1.1a}$$

$$T = T_B\ (\underline{x},t), \text{ for } \underline{x} \text{ on } S_T^1\ ; \text{ and} \tag{2.5.1.1b}$$

$$w = w_B\ (\underline{x},t), \text{ for } \underline{x} \text{ on } S_w^1\ ; \tag{2.5.1.1c}$$

where

p_B is the pressure at the specified boundary (Pa);

T_B is the temperature at the specified boundary (°C);

w_B is the mass fraction at the specified boundary (-);

S_p^1 is the part of the boundary with specified pressure;

S_T^1 is the part of the boundary with specified temperature; and

S_w^1 is the part of the boundary with specified mass fraction.

Care needs to be used in specifying the temperature and mass fraction at fluid-outflow boundaries, because, on boundary surfaces across which fluid flow occurs, the advective transport of heat and solute is assumed to dominate over any diffusive or dispersive transport. Thus, it is physically unrealistic to specify a temperature or solute concentration at an outflow boundary because the ambient fluid will determine the temperature, and solute concentration there.

2.5.2. Specified-Flux Boundary Conditions

The default boundary condition for the numerical model is no fluid, heat, or solute flux across the boundary surfaces. Normal fluxes of fluid, heat, and solute, known as Neumann boundary conditions, can be specified over parts of the boundary as functions of time and location. However, they cannot be specified independently, because, on boundary surfaces where a specified fluid flux exists, the advective transport of heat and solute is assumed to dominate over any specified diffusive or dispersive flux of these quantities. This assumption means that, on fluid-inflow boundaries, the temperature and mass fraction of the inflowing fluid needs to be specified. These specifications determine the heat and solute fluxes. At fluid-outflow boundaries, the temperature and mass fraction are determined by the ambient fluid values in the region, thus giving the heat and solute fluxes. Therefore, it is not physically realistic to specify temperatures and mass fractions at outflow boundaries. On boundary surfaces where no fluid flux is given, heat and solute fluxes may be specified. Heat fluxes represent thermal conduction and solute fluxes represent solute diffusion.

For the reasons discussed in section 2.5.1, it also is not physically realistic to specify dispersive heat or solute fluxes across boundary surfaces that have specified pressures. However, total heat or solute fluxes may be specified for inflow boundaries. These fluxes are the advective fluxes approaching the boundary from outside the region, and they are equal to the advective plus the dispersive fluxes leaving the boundary and entering the region. For outflow boundaries, the boundary condition requires that the dispersive fluxes be zero. Thus, only advective flux of heat and solute occurs at outflow boundaries. Again, the advective transport of heat and solute is assumed to dominate over dispersive flux.

Specified fluxes are expressed mathematically as:

$$q_{Fn} = (q^B_{Fx}, q^B_{Fy}, q^B_{Fz}) \text{ for } \underline{x} \text{ on } S^2_p \; ; \tag{2.5.2.1a}$$

$$q_{Hn} = (q^B_{Hx}, q^B_{Hy}, q^B_{Hz}) \text{ for } \underline{x} \text{ on } S^2_T \; ; \tag{2.5.2.1b}$$

$$q_{Sn} = (q^B_{Sx}, q^B_{Sy}, q^B_{Sz}) \text{ for } \underline{x} \text{ on } S^2_w \; ; \tag{2.5.2.1c}$$

where

q^B_{Fi} is the component of the fluid flux in the ith direction at the boundary (m^3/m^2-s);

q^B_{Hi} is the component of the heat flux in the ith direction at the boundary (W/m^2);

q^B_{Si} is the component of the solute flux in the ith direction at the boundary (kg/m^2-s);

q_{Fn} is the normal component to the boundary surface of the fluid-flux vector (kg/m^2-s);

q_{Hn} is the normal component to the boundary surface of the heat-flux vector (W/m^2);

q_{Sn} is the normal component to the boundary surface of the solute-flux vector (kg/m^2-s); and

S^2_u are parts of the boundary with specified-fluid, heat, or solute fluxes respectively; $u = p,T,w$.

Note that the specified-fluid flux, q_F, is given as a volumetric flux. A fluid density also needs to be specified for the case of inflow to the region. The density in the region at the boundary is used for computation of mass outflow rates. Also note that flux is a vector quantity with components expressed relative to the coordinate system of the simulation region. Examples of physical-boundary conditions that can be represented as specified-flux boundaries include infiltration from precipitation, lateral boundaries where the pressure gradients can be estimated, and simple steady-state flow fields where recharge- and discharge-boundary flow rates are known.

2.5.3. **Leakage Boundary Conditions**

A leakage boundary condition has the property that a fluid flux occurs in response to a difference in pressure and gravitational potential across a confining layer of finite thickness. Usually the permeability of this layer will be orders of magnitude smaller than the permeability of the simulation region and the aquifer region on the other side of the confining layer.

Representation of leakage boundary conditions is based on the approach of Prickett and Lonnquist (1971, p. 30-35), which has been generalized to include variable-density and variable-viscosity flow. The mathematical treatment of leakage boundaries is based on the following simplifying assumptions: (1) Changes in fluid storage in the confining layer are neglected; (2) confining-layer capacitance effects on heat and solute transport are neglected; (3) flow, heat, and solute transport are affected by the leakage fluxes that enter the region, but flow, heat, and solute conditions that exist on the far side of the confining layer outside the simulation region are not affected by fluxes that enter or leave the simulation region; and (4) flow and transport properties in the confining layer are based on the average of the fluid density and viscosity on either side. These assumptions are quite restrictive; but, in cases where they are not valid, some of the region outside the boundary probably needs to be included in the simulation region. Flow and transport rates are functions of differences in pressure, temperature, and solute-mass fraction at a point in time and are not affected by the previous values of these differences.

2.5.3.1. Leaky-Aquifer Boundary

A leaky-aquifer boundary can be adjacent to any part of the simulation region. For illustration, assume that it is part of the upper boundary surface that is overlain by a confining layer. Another aquifer lies above the confining layer with a pressure distribution at its contact with the confining layer which is a known function of time. The geometry is shown in figure 2.2. We are interested in the flux normal to the boundary between the confining layer and the simulation region, located at elevation z_B in figure 2.2. Under these assumptions, the leakage boundary flux is given by:

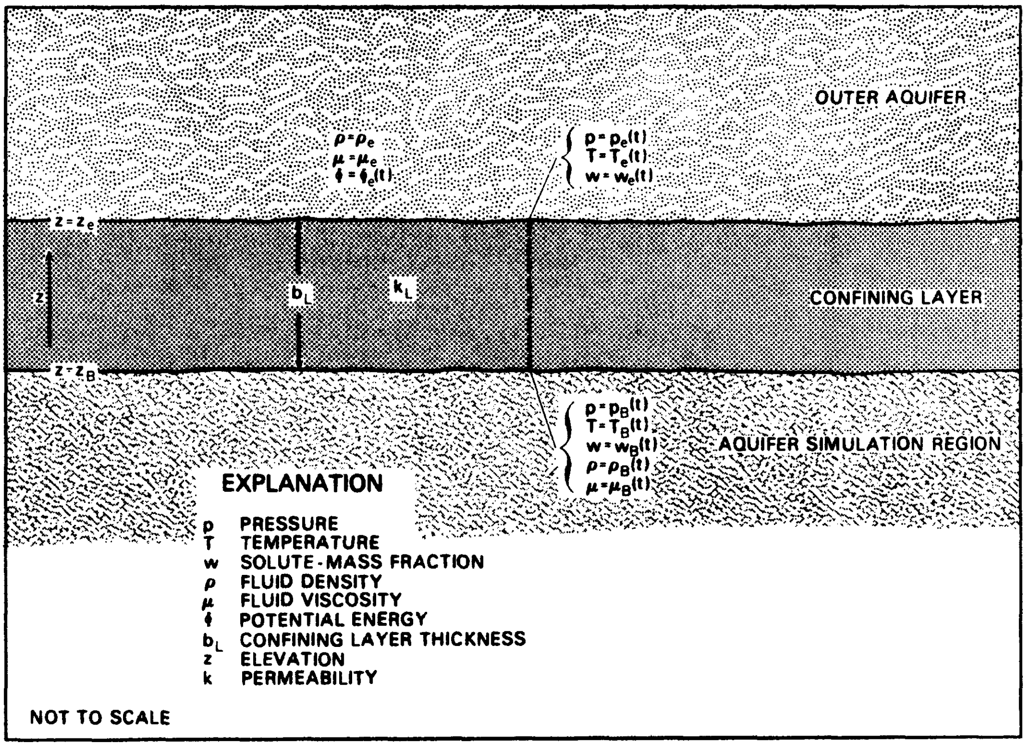


Figure 2.2.--Sketch of geometry for a leaky-aquifer boundary condition.

$$q_L = \frac{k_L}{\mu_L b_L} [\rho_e \Phi_e - (p_B + \rho_B g z_B)$$

$$-(\rho_e - \rho_B) g \, (z_e + z_B)/2], \text{ for } \underline{x} \text{ on } S^3; \qquad (2.5.3.1.1a)$$

with

$$\Phi_e = \frac{p_e}{\rho_e} + g\, z_e \;; \tag{2.5.3.1.1b}$$

$$\mu_L = \tfrac{1}{2}(\mu_B + \mu_e) \;; \tag{2.5.3.1.1c}$$

where

q_L is the fluid flux across the leakage boundary (m^3/m^2-s).
Φ_e is the potential energy per unit mass of fluid in the outer aquifer (N-m/kg);
p_B is the pressure at the simulation-region boundary (Pa);
p_e is the pressure at the top of the confining layer (Pa);
ρ_B is the fluid density at the simulation-region boundary (kg/m^3);
ρ_e is the fluid density in the outer aquifer (kg/m^3);
k_L is the permeability of the confining layer (m^2);
μ_L is the fluid viscosity in the confining layer (kg/m-s);
b_L is the thickness of the confining layer (m);
z_B is the elevation of the simulation-region boundary (m);
z_e is the elevation at the top of the confining layer (m); and
S^3 is the region boundary surface over which a leakage-boundary condition exists.

The terms Φ_e, ρ_e, k_L, μ_L, b_L, and z_B are specified functions of position along the leakage boundary; Φ_e and ρ_e also can be functions of time. The mass flux is calculated using ρ_e if the flux is into the simulation region, and using ρ_B if the flux is out from the simulation region. This choice of density is an approximation because it will take some time for the fluid in the confining layer to attain the limiting value after a change in flow direction takes place. However, this approximation is consistent with the neglect of transient flow and storage effects within the confining layer.

The heat and solute fluxes are assumed to be purely advective. They are obtained from enthalpies and mass fractions of the outer aquifer or at the boundary of the simulation region depending on the flux direction. Thus:

$$q_{HL} = H_e \rho_e q_L, \text{ if } q_L > 0, \text{ for } \underline{x} \text{ on } S^3; \tag{2.5.3.1.2a}$$

$$= H_B \rho_B q_L, \text{ if } q_L < 0 \;; \tag{2.5.3.1.2b}$$

$$q_{SL} = w_e \rho_e q_L, \text{ if } q_L > 0, \text{ for } \underline{x} \text{ on } S^3 \; ; \qquad (2.5.3.1.3a)$$

$$= w_B \rho_B q_L, \text{ if } q_L < 0 \; ; \qquad (2.5.3.1.3b)$$

where

q_{HL} is the heat flux across the leakage boundary (W/m²);
H_e is the specific enthalpy of the fluid in the outer aquifer (J/kg);
H_B is the specific enthalpy of the fluid at the region boundary (J/kg);
q_{SL} is the solute flux across the leakage boundary (kg/m²-s);
w_e is the solute mass fraction in the outer aquifer (-); and
w_B is the solute mass fraction at the region boundary (-).

Note that H_e and w_e are specified functions of position along the leakage boundary and time.

2.5.3.2. River-Leakage Boundary

The river-leakage boundary condition is a second type of leakage-boundary condition that is very similar to a leaky-aquifer condition, with the following differences: (1) This boundary condition is appropriate only for unconfined aquifer regions and is at an upper- or lateral-boundary surface; (2) the less-permeable boundary layer is now the riverbed-sediment layer, that is basically a piecewise-linear feature that traverses the upper boundary of the aquifer region; (3) a limit on the maximum flux from the river to the aquifer is imposed. Additional assumptions for the river-leakage option are: (1) The riverbed thickness is assumed constant over each cross section of the river; and (2) pressure and elevation differences and fluid properties are taken at the river centerline, representing conditions across the riverbed. An area factor is introduced to account for the fact that the riverbed area is only a fraction of the region boundary traversed by the river. The flux limit is set by not allowing the flux to increase after the aquifer pressure plus gravitational potential decreases to less than the gravitational potential at the bottom of the river. Physically, this means that if the water table declines below the bottom of the riverbed, the increased resistance to flow, because of the porous medium becoming partially saturated, prevents further increases in flux from the river to the aquifer. Thus, the flux limitation is a crude approximation to the physical situation. The simplified geometry of a

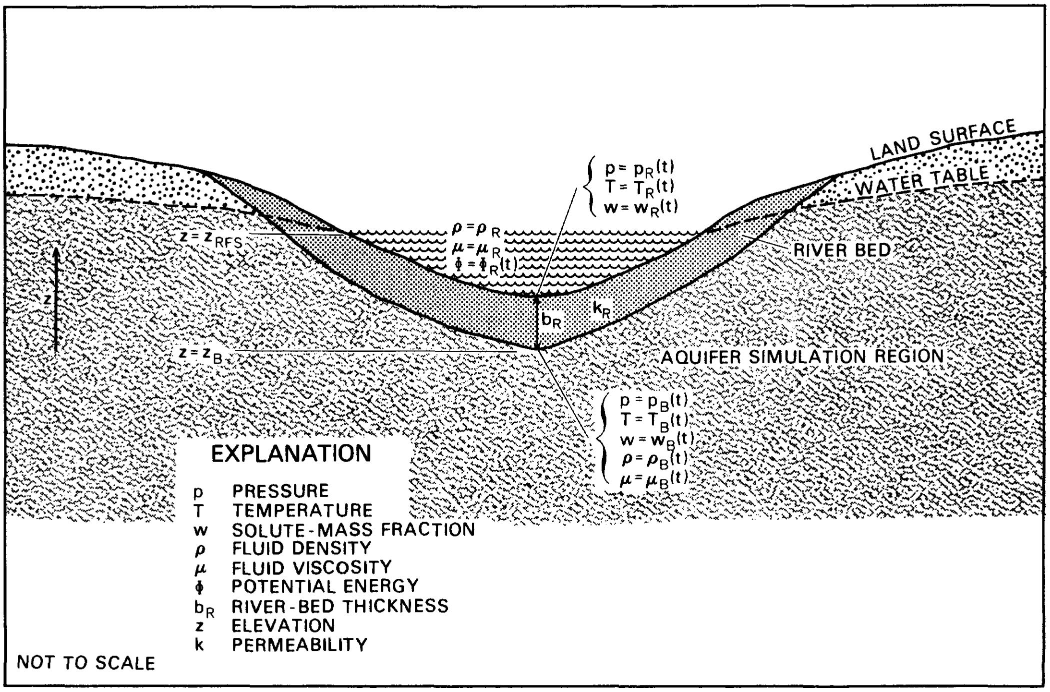


Figure 2.3.--Diagrammatic section showing geometry for a river-leakage boundary.

river-leakage boundary is shown in figure 2.3. Note that the z-axis is positive in the vertically upward direction. The present version of the HST3D program cannot simulate river leakage with a tilted coordinate system.

With the above assumptions, equations 2.5.3.1.1a and 2.5.3.1.1b become:

$$q_R = \gamma_R q_L ; \quad \text{for } \underline{x} \text{ on } S^4 ; \tag{2.5.3.2.1a}$$

with

$$\Phi_e = g z_{RFS} ; \tag{2.5.3.2.1b}$$

and

$$q_{Rmax} = q_R \Big|_{p_B = 0} ; \tag{2.5.3.2.1c}$$

where

- q_R is the fluid flux across the river-leakage boundary from the river to the aquifer (m^3/m^2-s);
- q_{Rmax} is the maximum fluid flux from the river to the aquifer (m^3/m^2-s); and
- Φ_e is the potential energy per unit mass of fluid in the river (Nt-m/kg);
- γ_R is the fraction of riverbed area per unit area of aquifer boundary (-);
- z_{RFS} is the elevation of the water surface of the river (m); and
- S^4 is the region boundary surface over which a river-leakage boundary condition exists.

Note that Φ_e, ρ_e, k_L, b_L, μ_e, z_B, and γ_R are specified as functions of position along the river length, and at Φ_e and ρ_e also can be functions of time. For calculating Φ_e, the value of atmospheric pressure can be taken as zero, because pressures are relative to atmospheric pressure. The mass flux is calculated using ρ_e if the flux is into the aquifer, and using ρ_B if the flux is out from the aquifer. The heat and solute fluxes are assumed to be purely advective, and are obtained from the enthalpies and mass fractions of the river fluid, or from the aquifer at the leakage boundary, depending on the flux direction, as given by equations 2.5.3.1.2a-b and 2.5.3.2.3a-b with q_L replaced by q_R. The enthalpy variation with pressure is neglected for the river-leakage boundary condition, as this variation is assumed to be small.

2.5.4. Aquifer-Influence-Function Boundary Conditions

The aquifer-influence-function (AIF) boundary conditions have been presented in the petroleum reservoir-simulation literature. Several methods have been used to calculate water influx at reservoir-aquifer boundaries. For a summary, the reader is referred to Craft and Hawkins (1959, ch. 5) and Aziz and Settari (1979, sec. 9.6).

The utility for ground-water flow simulation of fluid-flux calculations using aquifer-influence functions results from the fact that they enable a simulation region to be embedded within a finite or infinite surrounding region, for which the aquifer properties are known only in a general sense, and where the outer-aquifer-region flow field influences the inner-aquifer region of interest only in a general way. The primary benefit of using AIF boundary conditions is the reduction in size of the simulation region, resulting in a savings in computer-storage requirement and computation time.

Suppose that an aquifer region can be divided into subregions (fig. 2.4), where the inner-aquifer region is the one of primary interest, and the outer-aquifer region is less completely identified with respect to aquifer properties and geometrical configuration. The outer-aquifer region may completely or partially surround the inner-aquifer region, as shown in figures 2.4*A* and 2.4*B*. Variable density and nonisothermal flow may be simulated in the inner-aquifer region, but not in the outer-aquifer region. The actual simulation region may be reduced to only the inner-aquifer region, and the boundary condition at the boundary between the two regions (the AIF boundary) is taken to be the AIF boundary condition representing the outer-aquifer region. Aquifer-influence functions are analytical expressions that describe the flow rate, pressure, and cumulative flow at the boundary between the inner-aquifer and outer-aquifer regions, in response to pressure variations at the boundary. For the purposes of ground-water flow simulation described herein, the cumulative-flow aquifer-influence functions are not of concern. Flow from the outer-aquifer region is assumed to influence the inner-aquifer region of simulation, but flow to the outer-aquifer region does not affect any conditions there.

The aquifer-influence functions that describe transient flow across the AIF boundary are based upon analytical solutions to the ground-water flow equation in the outer-aquifer region. To obtain an analytical solution, the aquifer and fluid properties of the outer-aquifer region are assumed to be constant and uniform, and the geometry of the boundaries between the inner- and outer-aquifer regions need to be approximated by simple shapes.

Two types of aquifer-influence functions currently are available for the heat- and solute-transport simulator; one type treats the outer-aquifer region as a "pot;" the other type uses a transient-flow solution for simple-geometry and simple-boundary conditions. An aquifer-influence function based on the assumption of steady-state flow also exists in the petroleum-reservoir simulation literature, but it only is a restricted form of the leakage-boundary condition presented in section 2.5.3.1. Only one type of aquifer-influence function is allowed in any given simulation.

2.5.4.1. Pot-Aquifer-Influence Function

The pot-aquifer-influence function is based on the assumption of an outer-aquifer region with exterior boundaries that are impermeable (fig. 2.5). The outer-aquifer region needs to have volume and compressibility that are sufficiently small so that the pressure in this outer-aquifer region always will be virtually in equilibrium with the pressure distribution along the boundary surface between the inner- and outer-aquifer regions. Then, flow will occur in response to the rate of change of pressure at this boundary. The governing equation is obtained from mass conservation in a vertically deforming, compressible, porous medium. Using the Gauss divergence theorem (Karamcheti, 1967, p. 73) and assuming a uniform, constant, fluid density and

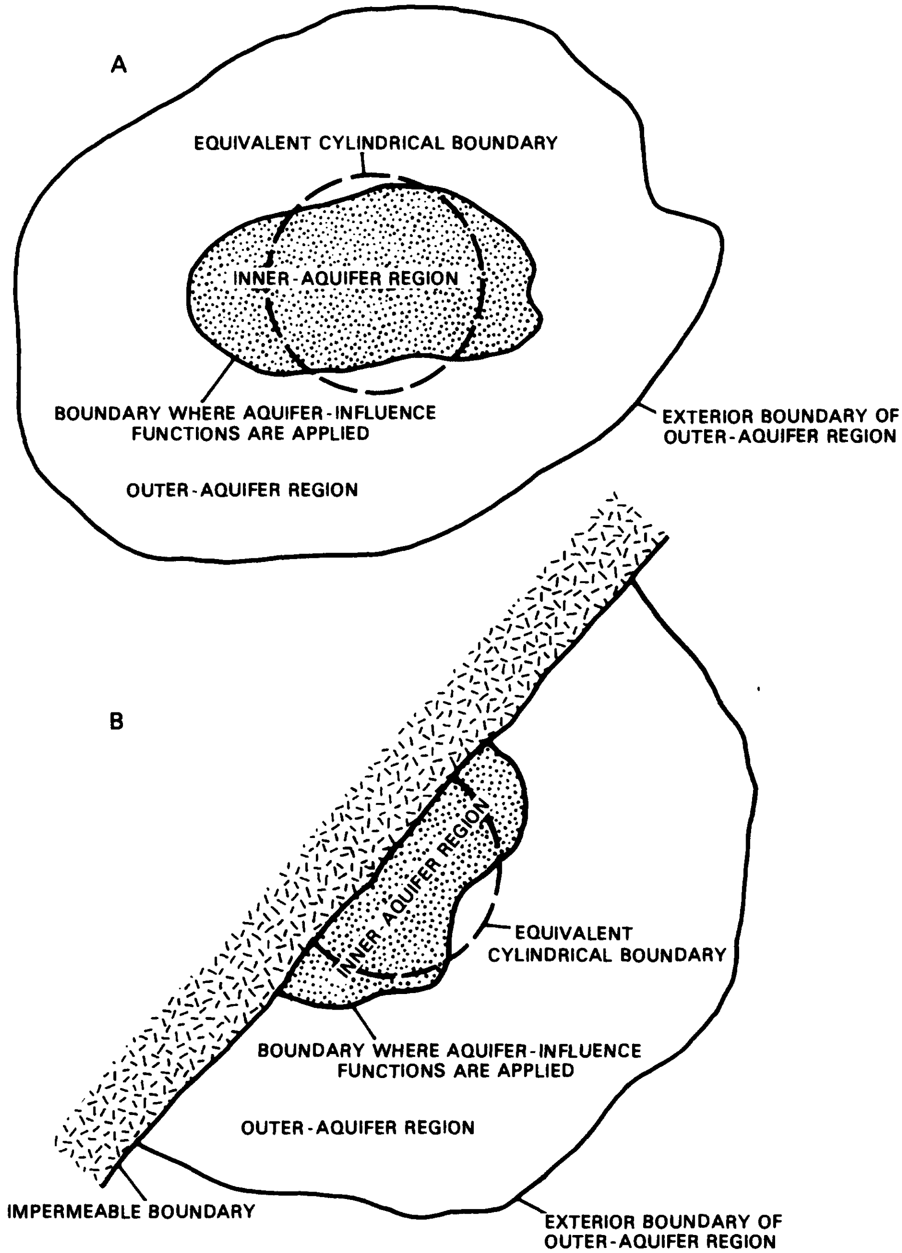


Figure 2.4.--Plan view of inner- and outer-aquifer regions and boundaries: *A*, Outer-aquifer region completely surrounding inner region; *B*, Outer-aquifer region half-surrounding inner-aquifer region.

porosity, we obtain by integration over the outer-region volume:

$$Q_A = (\alpha_{be} + \varepsilon_e \beta_{pe}) \overline{\frac{\partial p_e}{\partial t}} V_e \; ; \qquad (2.5.4.1.1)$$

where

Q_A is the volumetric flow rate across the boundary between the inner- and outer-aquifer regions; (positive is into the inner-aquifer region); (m^3/s).

$\overline{\frac{\partial p_e}{\partial t}}$ is the spatial average of the rate of pressure change in the outer region (Pa/s);

α_{be} is the bulk compressibility of the porous medium in the outer-aquifer region (Pa^{-1});

β_{pe} is the fluid compressibility in the outer-aquifer region (Pa^{-1});

ε_e is the porosity in the outer-aquifer region (-); and

V_e is the volume of the outer-aquifer region (m^3).

Because pressure equilibrium is assumed:

$$\overline{\frac{\partial p_e}{\partial t}} = \overline{\frac{\partial p_B}{\partial t}} \; ; \qquad (2.5.4.1.2)$$

where

$\overline{\frac{\partial p_B}{\partial t}}$ is the spatial average rate of pressure change at the boundary of the inner-aquifer region (Pa/s).

Equation 2.5.4.1.1 is in a form suitable for calculating overall fluid flow balances but it is difficult to distribute the flow over the AIF boundary

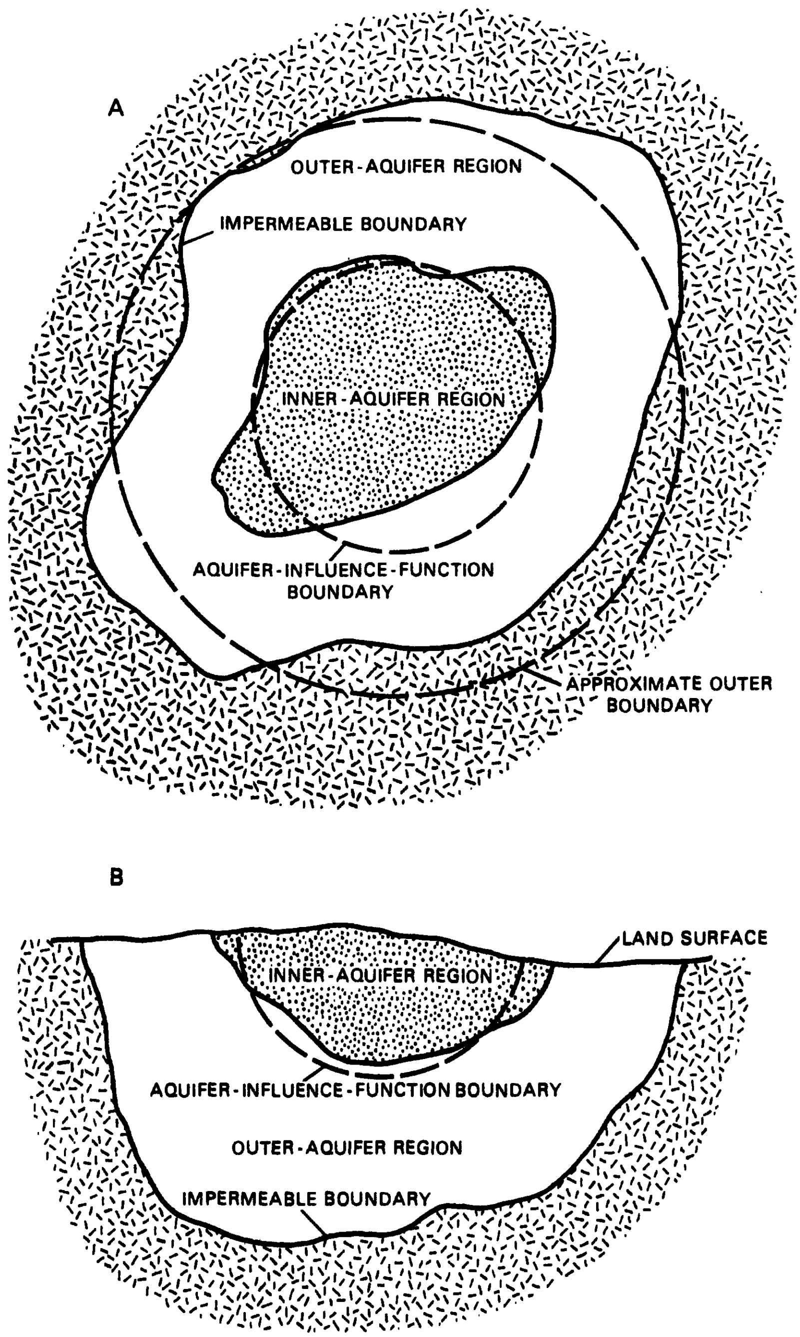


Figure 2.5.--*A*, plan view, and *B*, cross-sectional view of inner- and outer-aquifer regions with an impermeable exterior boundary for the outer-aquifer region.

of the simulation region, particularly when the rate of change of boundary pressure is not uniform over the boundary. Allocation of the flow rate will be explained in section 3.4.4 on numerical implementation.

2.5.4.2. Transient-Flow, Aquifer-Influence Function

The transient-flow solution method employs the Carter-Tracy AIF calculation technique (Carter and Tracy, 1960) and analytical solutions presented by Van Everdingen and Hurst (1949). A brief summary of the method follows. A detailed presentation is given by Kipp (1986).

Let the AIF boundary between the inner-aquifer or simulation region and the outer-aquifer region be approximated by a cylinder of a given radius and height. A plan view is presented in figure 2.5A. For a simulation region that is a rectangular prism, this boundary cylinder will be a severe approximation of the actual boundary shape. The outer boundary of this outer-aquifer region is a cylinder at a finite or infinite radius. The thickness of the outer-aquifer region is assumed to be uniform, with impermeable upper and lower boundary surfaces. Ground-water flow in this outer-aquifer region is radial at a given elevation, and the pressure satisfies:

$$(\alpha_{be} + \varepsilon_e \beta_{pe}) \frac{\partial p_e}{\partial t} = \frac{k_e}{\mu_e} \left[\frac{\partial^2 p_e}{\partial r^2} + \frac{1}{r} \frac{\partial p_e}{\partial r}\right] ; \qquad (2.5.4.2.1)$$

where

- r is the radial coordinate (m);
- p_e is the pressure in the outer-aquifer region (Pa);
- k_e is the permeability in the outer-aquifer region (m^2);
- μ_e is the viscosity in the outer-aquifer region (kg/m-s);
- α_{be} is the bulk compressibility of the porous medium in the outer-aquifer region (Pa^{-1});

β_{pe} is the compressibility of the fluid in the outer-aquifer region (Pa^{-1}); and

ε_e is the effective porosity of the outer-aquifer region (-).

The initial condition is:

$$\text{at } t = 0,\ p_e = p_e^o\ ; \qquad (2.5.4.2.2)$$

where

p_e^o is the initial uniform pressure (Pa).

Van Everdingen and Hurst (1949) used two different boundary conditions at the AIF cylindrical boundary: one condition was constant pressure, and the other condition was constant flow rate.

The boundary conditions are either specified pressure:

$$\text{at } r = r_I,\ p_e = p_B\ ; \qquad (2.5.4.2.3a)$$

where

r_I is the interior radius (m); and

p_B is the constant, specified pressure at the boundary (Pa);

or specified flow rate:

$$\text{at } r = r_I;\ \frac{\partial p_e}{\partial r} = \frac{Q_A \quad \mu_e}{2\pi r_I b_e k_e}\ ; \qquad (2.5.4.2.3b)$$

where

Q_A is the constant, specified flow rate at the boundary (positive is from the outer-aquifer region to the inner-aquifer region) (m^3/s); and

b_e is the thickness of the outer-aquifer region (m).

At the exterior cylindrical boundary, the condition is, for an infinite region:

$$\text{as } r \to \infty,\ p_e \to p_e^{o} \ ; \qquad (2.5.4.2.4a)$$

and, for a finite region, either no flow,

$$\text{at } r = r_E,\ \frac{\partial p_e}{\partial r} = 0 \ ; \qquad (2.5.4.2.4b)$$

where

r_E is the exterior radius (m);

or specified pressure:

$$\text{at } r = r_E,\ p_e = p_e^{o} \ . \qquad (2.5.4.2.4c)$$

Solutions to the dimensionless form of equations 2.5.4.2.1-2.5.4.2.4 are given in Van Everdingen and Hurst (1949) and were derived using Laplace transform techniques. For example, the flow-rate response to a unit change in pressure boundary condition (eq. 2.5.4.2.3a) and the pressure response to a unit withdrawal flow-rate boundary condition (eq. 2.5.4.2.3b) for an infinite outer-aquifer region are:

$$Q_U'(t') = \frac{4}{\pi^2} \int_o^\infty \frac{e^{-\lambda^2 t'} d\lambda}{\lambda[J_o^2(\lambda) + Y_o^2(\lambda)]} ; \qquad (2.5.4.2.5a)$$

and

$$P_U'(t') = \frac{-4}{\pi^2} \int_o^\infty \frac{1 - e^{-\lambda^2 t'} d\lambda}{\lambda^3[J_1^2(\lambda) + Y_1^2(\lambda)]} ; \qquad (2.5.4.2.5b)$$

respectively, with

$$t' = \frac{k_e \; t}{r_I^2 \; (\alpha_{be} + \varepsilon_e \beta_{pe}) \mu_e} ; \qquad (2.5.4.2.5c)$$

where

J_i is the Bessel function of the first kind of order i;
Y_i is the Bessel function of the second kind of order i; and
t' is the dimensionless time (-).

Equation 2.5.4.2.5a was presented by Jacob and Lohman (1952) in a different form as a solution to the constant drawdown problem for flow to a well. These two aquifer-influence functions will be referred to as the flow-rate response to a unit-step pressure change Q_U', and the pressure response to a unit-step withdrawal flow rate, P_U', respectively.

The concept of superposition or convolution (Tychonov and Samarski, 1964, p. 209) is used to derive the aquifer-influence functions from the unit-step response functions for a transient-pressure function at the boundary between the inner- and outer-aquifer regions. Thus, the flow-rate response of this ground-water flow system to a time-varying pressure at the inner boundary of the outer-aquifer region can be written as:

$$Q'_A = - \int_o^{t'} p'_B(\tau) \frac{\partial Q'_U(t' - \tau)}{\partial t'} d\tau ; \qquad (2.5.4.2.6a)$$

where

$$p'_B(t') = \frac{p_B(r_I,t') - p_e^o}{p_e^o - p_B^o} ; \qquad (2.5.4.2.6b)$$

and where

p' is the dimensionless pressure (-); and

p_B^o is the boundary pressure at time zero that initiates flow (Pa).

In principle, equation 2.5.4.2.6a could be used to calculate the flow rate into the simulation region from the outer-aquifer region. However, this is not practical when $p'_B(t)$ is not known in advance as in the present case of numerical simulation of ground-water flow in the inner-aquifer region. The integral needs to be recomputed repeatedly from the initial time to the current time, because time is a parameter in the integrand as well as being the upper limit of the integration variable. Carter and Tracy (1960), using an approach given by Hurst (1958), developed an approximate algorithm to avoid the repeated computation of equation 2.5.4.2.6a as the simulation calculation progresses. However, it requires the discretization of time and will be treated in section 3.4.4.2.

2.5.5. **Heat-Conduction Boundary Condition**

A boundary condition is available for pure-heat conduction without fluid flow or solute transport. This boundary condition provides for the simulation of heat gain or loss at a boundary which confines the ground-water flow. The boundary heat flux depends on the evolving temperature profile in the conducting medium exterior to the simulation region. One-dimensional conduction is assumed perpendicular to the boundary surface and conduction in the adjacent medium parallel to the boundary because of lateral temperature variation is neglected. The penetration of heat into or the withdrawal of heat from the boundary medium is assumed to progress only a finite distance out from the boundary. Beyond this distance, the temperature is assumed to remain at its initial uniform value. This assumption is for convenience in the numerical implementation. Constant, uniform thermal properties are assumed in the exterior medium. Based on these assumptions, the one-dimensional heat-conduction equation can be used to represent the boundary condition. This equation is:

$$\rho_{se} c_{se} \frac{\partial T_e}{\partial t} = K_e \frac{\partial^2 T_e}{\partial z_n^2} ; \tag{2.5.5.1a}$$

where

$\rho_{se} c_{se}$ is the heat capacity per unit volume of the adjacent medium (J/m³-°C);
K_e is the thermal conductivity of the adjacent medium (W/m-°C);
T_e is the temperature in the adjacent medium (°C); and
z_n is the coordinate in the outward normal direction to the boundary (m).

The initial condition is:

$$t = 0; \; T_e = T_e^o(z_n); \tag{2.5.5.1b}$$

where

T_e^o is the initial temperature profile (°C).

The boundary conditions are:

$$\text{at} \quad z_n = 0 \; ; \quad T_e = T_B(t) \; ; \tag{2.5.5.1c}$$

and at

$$z_n = b_{HC} \; ; \quad T_e = T_e^0 (b_{HC}) \; ; \qquad (2.5.5.1d)$$

where

T_B is the boundary temperature at the aquifer boundary (°C),

b_{HC} is the effective thickness of the conducting medium outside the region (m).

The thermal properties of the adjacent medium are assumed constant and uniform. Thus:

$$D_{He} = \frac{K_e}{\rho_{se} c_{se}} \; ; \qquad (2.5.5.2)$$

where

D_{He} is the thermal diffusivity for the adjacent medium (m^2/s).

Since the heat flux depends on the temperature profile in the exterior medium which in turn depends on the thermal history of the simulation, a simplifying approximation is used. This approximation eliminates the need to recompute or save the temperature-profile history during the course of the simulation.

The boundary-value problem specified by equations 2.5.5.1a-d can be resolved into simpler problems, as shown by Sneddon (1951, p. 162-165) or Tychonov and Samarski (1964, p. 203-209), using various forms of Duhamel's Theorem. Two simpler problems, for a general time interval, are:

$$\frac{\partial T_1}{\partial t} = D_{He} \frac{\partial^2 T_1}{\partial z_n^2}; \quad \text{on } 0 \le z_h \le b_{HC} \ \text{ and } t_o \le t \le t_1 \ . \tag{2.5.5.3a}$$

Boundary conditions:

$$\text{at } z_n = 0 \ ; \quad T_1 = 0 \ ; \tag{2.5.5.3b}$$

$$\text{at } z_n = b_{HC} \ ; \quad T_1 = 0 \ . \tag{2.5.5.3c}$$

Initial condition:

$$\text{at } t = t_0 \ ; \quad T_1 = T_e^0(z_n) \ ; \tag{2.5.5.3d}$$

and

$$\frac{\partial T_2}{\partial t} = D_{He} \frac{\partial^2 T_2}{\partial z_n^2} \quad \text{on } 0 \le z_n \le b_{HC} \ \text{ and } t_o \le t \le t_1 \tag{2.5.5.4a}$$

Boundary conditions:

$$\text{at } z_n = 0 \ ; \quad T_2 = T_B(t) \ ; \tag{2.5.5.4b}$$

$$\text{at } z_n = b_{HC} \ ; \quad T_2 = 0 \ . \tag{2.5.5.4c}$$

Initial condition:

$$\text{at } t = t_0 \ ; \quad T_2 = 0 \ ; \tag{2.5.5.4d}$$

where

$T_1(z,t)$ is the temperature solution to the first heat-conduction problem (°C); and

$T_2(t)$ is the temperature solution to the second heat-conduction problem (°C).

The total temperature solution is the sum of T_1 and T_2 and the boundary heat flux is derived from the gradient of this temperature. However, because the boundary-temperature function is not known in advance the following approximation is made

$$q_{HC} = -K_e \left. \frac{\partial T_e}{\partial z_n} \right|_{\substack{Z_n = 0 \\ T_2 = T_B(t_o)}} \tag{2.5.5.5a}$$

$$\cong - K_e \left[\left. \frac{\partial (T_1+T_2)}{\partial z_n} \right|_{\substack{z_n = 0 \\ T_2 = T_B(t_o)}} \right.$$

$$\left. + \frac{\partial}{\partial T_B} \left. \frac{\partial (T_1+T_2)}{\partial z_n} \right|_{\substack{z_n = 0 \\ T_2 = T_B(t_o)}} \delta T_B \right] ; \tag{2.5.5.5b}$$

where

q_{HC} is the heat flux at a heat-conduction boundary at a given boundary temperature and time (W/m^2); and

δT_B is the change in boundary temperature in the time interval t_o to t (°C).

Equation 2.5.5.5b is simply a Taylor-series expansion of the flux as a function of the variable boundary temperature.

By interchanging the order of differentiation and using the facts that

$$\frac{\partial T_1}{\partial T_B} = 0 \; ; \qquad (2.5.5.6a)$$

and

$$\frac{\partial T_2}{\partial T_B} = T_U(t) \; ; \qquad (2.5.5.6b)$$

where

T_U is the solution to equations 2.5.5.4a-d with $T_B = 1$ (°C);

we obtain:

$$q_{HC} = -K_e \left[\left. \frac{\partial T_e}{\partial z_n} \right|_{\substack{z_n = 0 \\ T=T_B(t_0)}} + \left. \frac{\partial T_u}{\partial z_n} \right|_{z_n = 0} \delta T_B \right] , \text{ for } \underline{x} \text{ on } S^5 \; ; \qquad (2.5.5.7)$$

where

S^5 is the part of the boundary that is a heat-conduction boundary.

The temperature, T, now satisfies equations 2.5.5.1a-d with the time dependence of the boundary condition removed in equation 2.5.5.1c. This approach to the treatment of heat-conduction boundary conditions was presented by Coats and others (1974) in the appendix to their paper. A heat-conduction boundary condition also could be treated like the transient AIF boundary condition, but that is beyond the scope of this work.

2.5.6. Unconfined Aquifer, Free-Surface Boundary Condition

For an unconfined aquifer, a free-surface boundary exists with a position in space and time that is unknown before the flow equations are solved. Therefore, two boundary conditions need to be imposed. The first is that pressure is atmospheric at the free surface. The second is the kinematic condition expressing the fact that the movement of this surface of atmospheric pressure needs to satisfy a continuity equation at the free surface.

The free-surface boundary is assumed to be a sharp interface between the fully saturated region of simulation and the unsaturated porous medium outside. The zone of capillary fringe that is partially saturated and the surfaces of seepage that exist with free-surface gravity flow are neglected. Delayed yield effects also are neglected, therefore, the specific yield is equal to the effective porosity, ε, in the vicinity of the free surface. The effective porosity under draining conditions is less than the porosity used to calculate interstitial velocity (Bear, 1972, p. 255) but the difference is assumed negligible for the HST3D simulator. Finally, the z-axis is assumed to point vertically upward when an unconfined aquifer is being simulated.

The heat- and solute-transport simulator treats the free-surface boundary in an approximate fashion. The approach follows the ideas of Prickett and Lonnquist (1971, p. 43-45) extended to a three-dimensional flow and variable-density system. The pressure condition:

$$p = 0, \text{ on } S^6(\underline{x},t); \qquad (2.5.6.1)$$

where

S^6 is the free-surface location that varies in space and time;

is employed, but the kinematic boundary condition is neglected. The absolute pressure on the free surface is atmospheric, so the relative pressure is zero. Hydrostatic conditions are assumed to exist in the immediate vicinity of the free surface. The location of the free surface is determined by interpolation in the calculated pressure field to determine the location where equation 2.5.6.1 is satisfied. Under this approximate treatment, the free surface moves in response to a net gain or loss of fluid in its vicinity. Thus, fluid mass is conserved, but the kinematics of the free-surface movement are neglected. This approximation is acceptable when the velocity of free-surface movement is small relative to the horizontal interstitial velocity.

In addition, the computational region is fixed for the duration of the simulation. Boundary pressures less than atmospheric imply that the free surface is below the boundary of the region; whereas boundary pressures greater than atmospheric imply that the free surface is above this boundary. As will be explained in section 3.4.6, the free surface is allowed to rise above the region boundary a short distance which is a function of the vertical discretization. This allowance enables the free surface to move within a reasonable range during a simulation. The fluid- and porous-matrix compressibilities usually are taken to be zero for unconfined flow systems, and the user may specify compressibility values of zero for the HST3D simulator.

2.6. INITIAL CONDITIONS

This heat- and solute-transport simulation code solves only the transient forms of the ground-water flow and the two transport equations, thus initial conditions are necessary to begin a simulation. Several options are available.

For the flow equation, an initial-pressure distribution within the region needs to be specified. This can be done as a function of position or can be set to hydrostatic conditions, with the pressure given at one elevation. In the case of nearly uniform and constant density, an initial potentiometric-head distribution can be specified, which is the water-table elevation. The water-table elevation is specified for the upper layer of cells only. No option to specify a velocity field as an initial condition exists.

For the heat-transport equation, the initial-temperature field needs to be specified. Again, this can be done as a function of position, or interpolated along the z-coordinate direction from a specified geothermal profile.

For the solute-transport equation, the initial mass-fraction field needs to be specified. This can be done only by specifying values as a function of position.

As will be described in section 4.6, pressure, temperature, and mass-fraction fields calculated by one simulation can be used as the initial conditions for another simulation, using the restart option. This often is the easiest way to establish a steady-state flow field before transport is simulated. Of course, one needs to determine whether or not an initial steady-state flow field exists for the physical situation being simulated. It should be noted that, with a hydrostatic or other estimate of initial pressure conditions, it could take some time to establish the steady-state flow field.

Mathematically, the initial conditions can be stated as follows:
At t=0:

$$p = p^0(x), \text{ in } V \; ; \tag{2.6.1a}$$

$$T = T^0(x), \text{ in } V \; ; \tag{2.6.1b}$$

$$w = w^0(x), \text{ in } V \; ; \tag{2.6.1c}$$

where

$\underline{x}$ is the vector of position (m),

V is the simulation region; and

p^0, T^0, w^0 are the initial dependent variable distributions (Pa,°C,-).

3. NUMERICAL IMPLEMENTATION

In order to perform numerical calculations that solve the governing equations, we first need to discretize the partial-differential equations and boundary condition relations in space and time. Various algorithms are used to determine parameters and to implement the boundary conditions. Then the flow and two transport equations are solved sequentially after they have been modified by a partial Gauss reduction. Finally, the sets of discretized equations are solved repeatedly, as the simulation time advances, using a direct or an iterative equation solver. This chapter will cover each of these steps for the numerical-simulation calculation.

3.1 EQUATION DISCRETIZATION

The classical method of finite differences is used to discretize the partial-differential equations and boundary conditions in space and time. Several options are available for the differencing.

The first step in spatial discretization is to construct a mesh or grid of node points and their associated cells, that covers the simulation region to a close approximation (fig. 3.1). The grid of node points is formed by specifying the distribution of nodes in each of the three coordinate directions; (two directions, if a cylindrical-coordinate system). The volume associated with each node will be called a cell; it is formed by the cell boundaries, which are planes that bisect the distance between adjacent node points. Thus, for the case of unequal nodal spacing, the node points do not lie at the centers of their respective cells. Boundaries are represented by planes containing node points. Thus, half-, quarter-, and eighth-cells appear at various sides, edges, and corners of the mesh, forming the simulation region (fig. 3.1). The minimum number of nodes required to define a region is eight, one node at each corner of the rectangular prism. The mesh or grid described is called a point-distributed grid. Other terms that have been used are face-centered mesh and lattice-centered mesh or grid. Another term that has been used for cell is block.

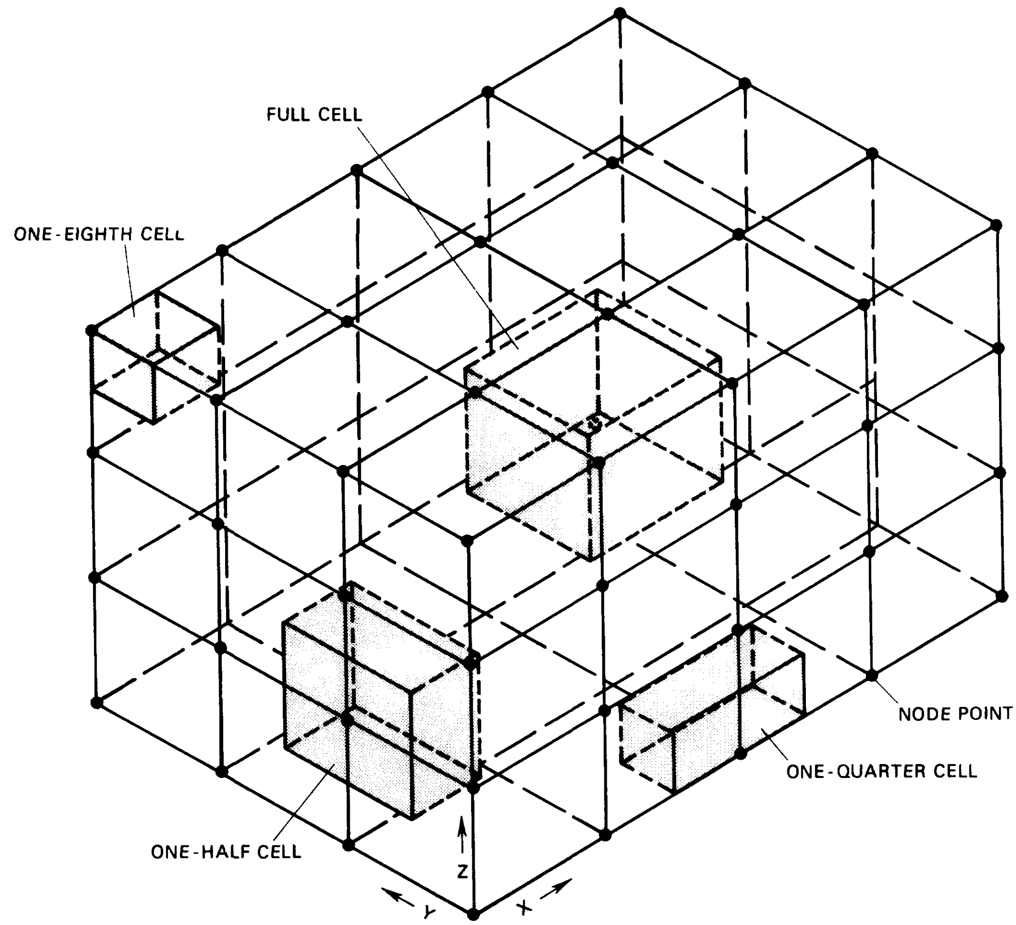


Figure 3.1.--Sketch of finite-difference spatial discretization of the simulation region.

The simulation region is discretized into rectangular prisms for the cartesian-coordinate case (fig. 3.1) and into annuli with rectangular cross sections for the cylindrical-coordinate case (fig. 3.2). Four types of regional volume subdivisions are defined (fig. 3.3). The primary subdivision is the cell that is the volume over which the flow, heat, and solute balances are made to give the nodal finite-difference equations. The second subdivision is the element that is the volume bounded by eight corner nodes in cartesian coordinates and four corner nodes in cylindrical coordinates. The element is the minimum volume with uniform porous-medium properties. The third subdivision is the zone that is a continuous set of elements with the

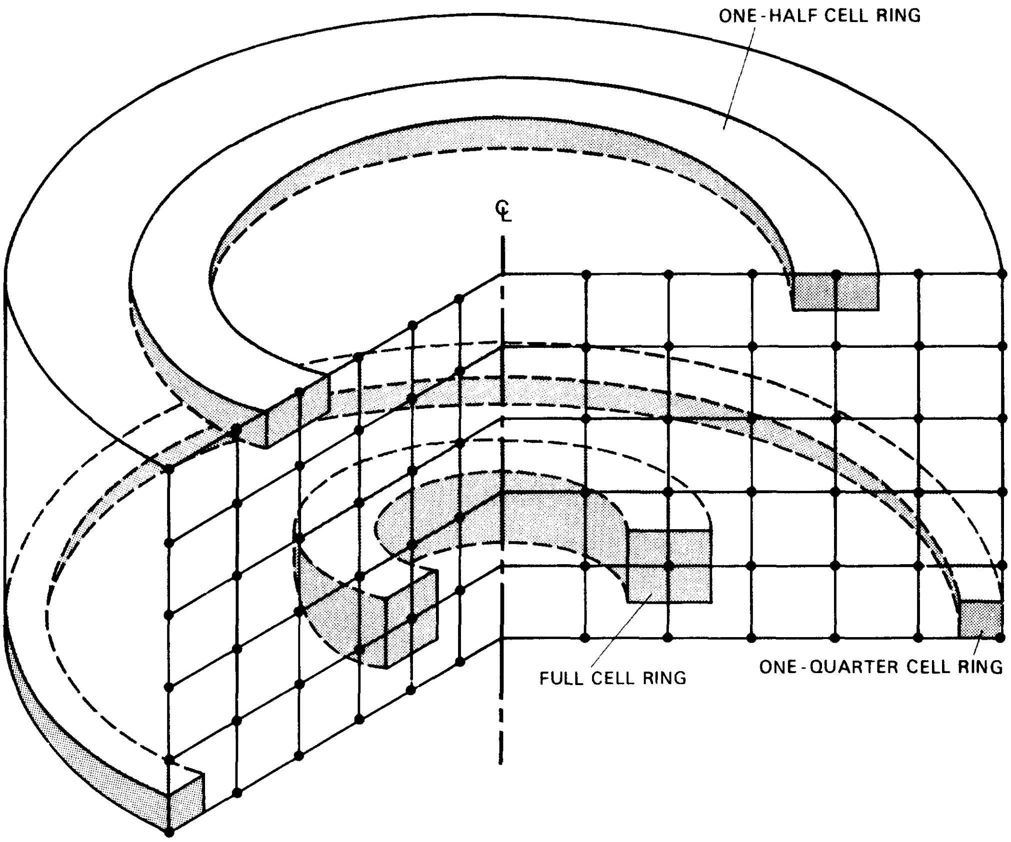


Figure 3.2.--Sketch of finite-difference spatial discretization for a cylindrical-coordinate system.

same porous-medium properties. The one restriction is that zones need to be convex. In other words, they need to be rectangular prisms. One zone may not border another zone on more than one side. Sometimes multiple, adjacent zones will have to be specified that have the same properties in order to adhere to this restriction of convex shape. The fourth subdivision is the subdomain that is the intersection or common volume of an element with a cell. A cell may have as many as eight subdomains, if it is an interior cell, or as few as one subdomain, if it is a corner cell. The finite-difference equations are

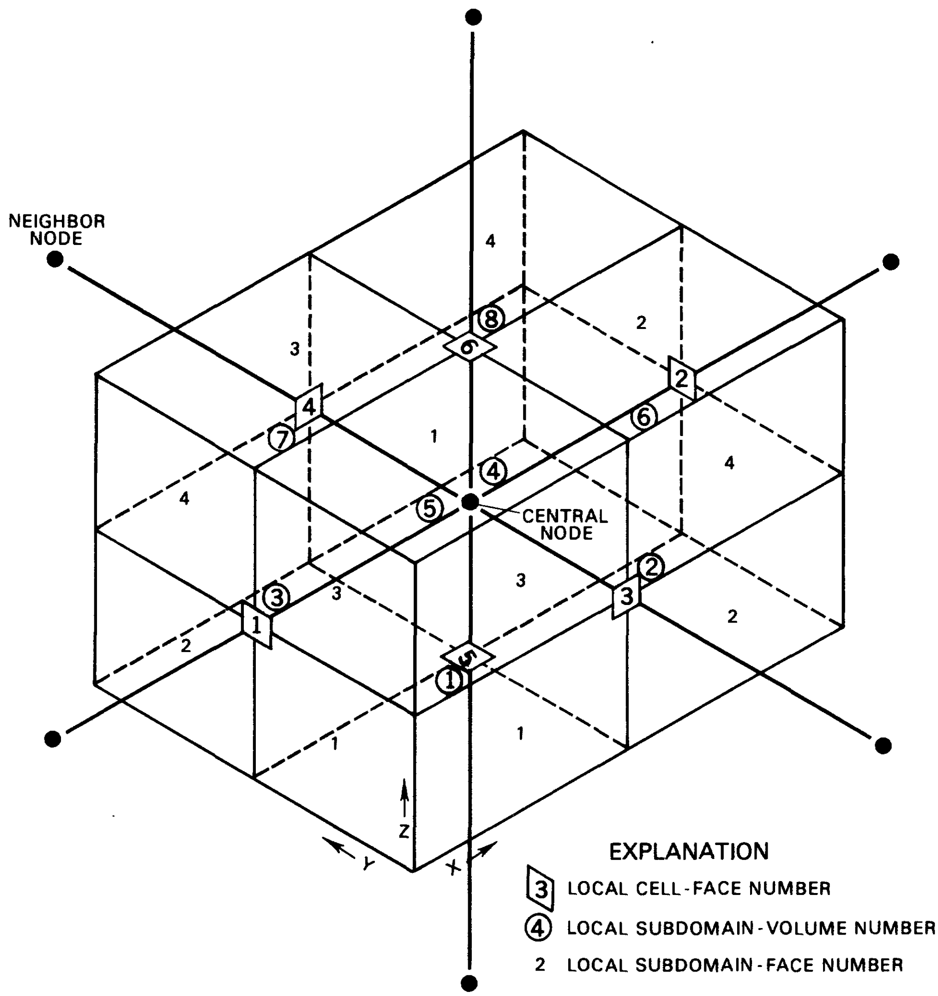


Figure 3.3.--Sketch of a node with its cell volume showing the cell faces, the subdomain volumes, and the subdomain faces.

assembled by adding the contributions of each subdomain in turn to the equation for a given cell. The primary reason for introducing the concepts of elements and zones for assigning porous-medium properties is so that porous-medium properties can be defined easily at the cell boundaries without the need for harmonic-mean calculations.

A numbering scheme local to the cell is used during coefficient calculation and assembly. Figure 3.3 shows the subdomain local numbering from one to eight. The six faces are numbered as shown in figure 3.3 and each face is subdivided into four subdomain faces with numbers for the visible faces as shown. For the cylindrical system, the corresponding volumes and faces are numbered in figure 3.4.

A common alternative method for constructing the mesh is to specify the locations of the planes that form the cell walls. Their intersections form the cells; then the node points are located in the center of each cell. This is called a cell-centered or block-centered grid. One advantage of this grid is that fewer cells are required to span a given simulation region, because fractional cells do not appear at the boundaries.

The point-distributed grid was selected for this simulator, because the finite-difference spatial approximations to the dispersive terms in the flow and transport equations are consistent and convergent for the point-distributed grid under conditions of variable-grid spacing; whereas, these approximations are not necessarily consistent and convergent for the cell-centered grid. As shown by Settari and Aziz (1972, 1974), the local truncation error for the cell-centered grid has a term that does not necessarily vanish as the grid spacing is refined. A second reason for selecting the point-distributed grid is that the presence of nodes on the boundary surfaces simplifies the treatment of certain boundary conditions. It is common to approximate spatially distributed, aquifer properties as uniform zones. A disadvantage of the point-distributed grid is that it is difficult to locate the cell boundaries, so that they coincide with the zoned-property boundaries. This difficulty can be avoided by making the properties uniform over an element rather than a cell. This will be described in the parameter-discretization section.

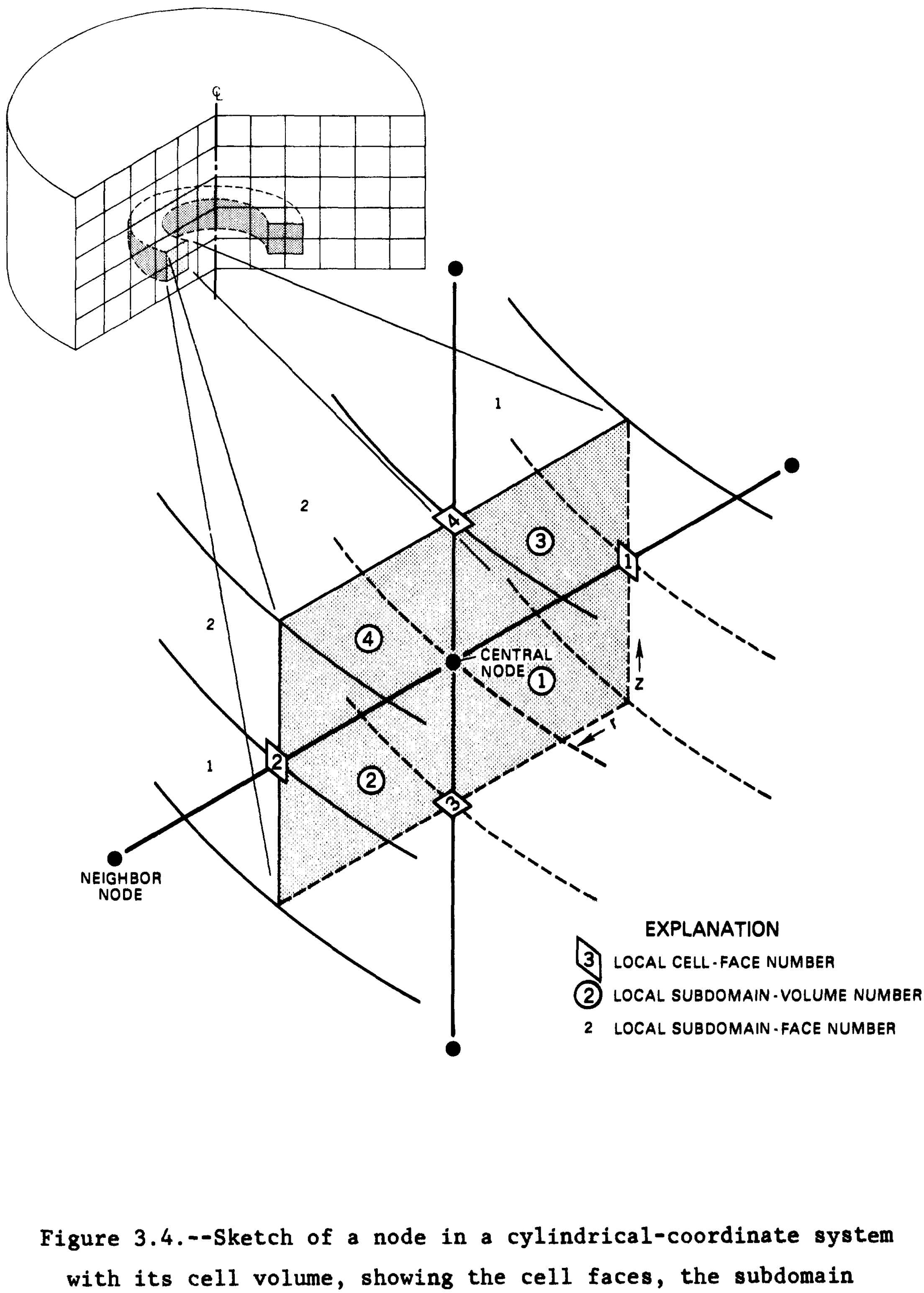


Figure 3.4.--Sketch of a node in a cylindrical-coordinate system with its cell volume, showing the cell faces, the subdomain volumes, and the subdomain faces.

A cartesian-coordinate grid for a region is shown in figure 3.1. Note that the basic cell is a rectangular prism; thus, region boundaries that are not parallel to a coordinate axis must be approximated by a staircase-like pattern of cell boundaries. No provision for boundary faces that are diagonally oriented to the coordinate axes exists in the present version of the HST3D code. It can also be seen that the entire simulation region needs to be contained in a large rectangular prism. The nodal dimensions of this prism are the maximum number of nodes along the three coordinate axes, N_x, N_y, and N_z. Approximation of diagonal boundaries by a staircase-like pattern will cause a set of cells to be within the large prism that are excluded from the simulation region. These cells will be referred to as excluded cells.

The method used for spatial discretization is a subdomain weighted-residual method with approximating functions that are piecewise linear for the dependent variables. The unknown parameters in the approximating functions are the nodal values of the dependent variables. The residuals are the errors in the governing equations that result from using approximating functions to the exact solutions, and equations for the unknown parameters require that the average residual over each cell is zero (Crandall, 1956, p. 149; Finlayson, 1972, p. 7-9, 137, 142). The partial-differential equations are discretized in space by integrating them over each cell volume. Then the divergence theorem of Gauss (Karamcheti, 1967, p. 73) is used to transform the volume integrals of the divergence terms into surface integrals of a normal derivative. The spatial derivatives in the surface integrals are approximated by central or upstream differences. The volume integrals are approximated easily, using the mean value theorem of integral calculus, because all fluid and porous medium properties are assumed to be constant throughout the subdomain volumes of a cell. For the integral of the time derivative, we assume that the time derivative evaluated at the node approximates the spatial average of the derivative over the volume of the cell. Thus, the capacitance-coefficient matrix of the temporal derivative terms is diagonalized. This method for spatial discretization is conceptually similar to the integrated-finite-difference method presented by Narasimhan and Witherspoon (1976).

The porous-matrix hydraulic, thermal-transport, and solute-transport properties are discretized on an element basis, with a set of elements forming a zone of constant properties. The dependent variables that are properties of the fluid are discretized on a cell basis. Boundary-condition fluxes and source flow rates also are discretized on a cell basis.

To illustrate the discretization of the flow, heat, and solute equations, we shall use a general transport equation. The procedure follows that of Varga (1962, sec. 6.3), Spanier (1967, p. 218-222), Cooley (1974, p. 10-13) or Roache (1976, p. 23-28) extended to include spatial first-derivative terms, to three dimensions for cartesian coordinates, and to handling dispersive tensors that are not necessarily diagonal. The restriction exists that all cell-boundary planes need to be perpendicular to a coordinate direction. The general transport equation has the form of a parabolic, partial-differential equation:

$$\frac{\partial}{\partial t}(A(\underline{x},t)\ u(\underline{x},t)) = \nabla\cdot\underline{\underline{B}}(\underline{x},t)\cdot\nabla u(\underline{x},t)-\nabla\cdot\underline{C}(\underline{x},t)u(\underline{x},t) \qquad (3.1.1a)$$

$$+ D(\underline{x},t)u(\underline{x},t) + \sum_{s=1}^{N_s} E_s(t)\delta(\underline{x}-\underline{x}_s)\ ;$$

where

$\underline{x}$ is the vector of position, (x,y,z), (m);

A is the capacitance coefficient (appropriate units);

$\underline{\underline{B}}$ is the tensor of diffusion or dispersion of rank 3 (appropriate units);

$\underline{C}$ is the vector of interstitial velocity (m/s);

D is the source factor for chemical reaction (appropriate units);

E_s is the source-term intensity (appropriate units);

N_s is the number of source terms;

u is the dependent variable (appropriate units); and

$\delta(\underline{x}-\underline{x}_s)$ is the delta function for a point source at $\underline{x}=\underline{x}_s$ (-).

Initial condition is:

at $\quad t = 0;\ u = u^0(\underline{x})$; (3.1.1b)

Boundary conditions are:

specified value:

$$u = u_B(\underline{x},t), \text{ for } \underline{x} \text{ on } S^1 ; \tag{3.1.1c}$$

specified flux:

$$-\underline{\underline{B}}(\underline{x},t)\ \frac{\partial u}{\partial n} + \underline{C}(\underline{x},t)\cdot\underline{n}u = \underline{J}(\underline{x},t)\cdot\underline{n} \text{ for } \underline{x} \text{ on } S^2 ; \tag{3.1.1d}$$

where

u^0 is the initial distribution of u;

u_B is the boundary distribution of u;

$\frac{\partial}{\partial n}$ is the derivative in the direction of the outward normal at the boundary;

$\underline{J}\cdot\underline{n}$ is the specified total flux normal to the boundary surface; and

$\underline{C}\cdot\underline{n}$ is the advective flux normal to the boundary surface.

3.1.1 Cartesian Coordinates

Integration of equation 3.1.1a over the cell volume associated with a mesh point, m, at a given location with indices i, j, k, gives:

$$\int_{V_m} \frac{\partial}{\partial t}\ Au\ dV = \int_{V_m} \nabla\cdot(\underline{\underline{B}}\cdot\nabla u)dV - \int_{V_m} \nabla\cdot\underline{C}udV \tag{3.1.1.1}$$

$$+ \int_{V_m} \underline{D}\ udV + \sum_{s=1}^{N_{sm}} \int_{V_m} E\delta(\underline{x}-\underline{x}_s)\ dV ;$$

where

V_m is the volume of cell m.

Now, using the previously stated assumption about the integral of the time derivative:

$$\int_{V_m} \frac{\partial}{\partial t} AudV = \frac{\partial}{\partial t} \int_{V_m} AudV \; . \tag{3.1.1.2}$$

Use of the Gauss divergence theorem on the dispersive and advective terms yields:

$$\int_{V_m} \nabla\cdot(\underline{\underline{B}}\cdot\nabla u)dV = \int_{S_m} (\underline{\underline{B}}\nabla u)\cdot\underline{n}dS \; ; \tag{3.1.1.3}$$

and

$$\int_{V_m} \nabla\cdot(\underline{C}u)dV = \int_{S_m} (\underline{C}u)\cdot\underline{n}dS \; ; \tag{3.1.1.4}$$

where

S_m is the boundary of cell m; and

$\underline{n}$ is the outward unit normal vector to the boundary.

Then equation 3.1.1.1 becomes:

$$\frac{\partial}{\partial t} \int_{V_m} AudV = \int_{S_m} [\underline{\underline{B}}\cdot\nabla u - \underline{C}u]\cdot\underline{n}dS + \int_{V_m} \underline{D}udV +$$

$$\int_{V_m} E_{s(m)} \delta(\underline{x}-\underline{x}_{s(m)})dV \; . \tag{3.1.1.5}$$

We have consolidated all the line sources within cell m into a single equivalent line source, thus eliminating the summation. Since considerable arbitrariness exists in selecting the finite-difference approximation of equation 3.1.1.5, we shall choose finite differences that preserve the conservation of u for each cell.

Following Varga, 1962, p. 253, or Cooley, 1974, p. 16, we approximate the rate of change of u in the cell using the mean-value theorem giving:

$$\frac{\partial}{\partial t} \int_{V_m} A u dV \cong \frac{\partial}{\partial t} \left(u(\underline{x}_m, t) \int_{V_m} A dV\right) ; \qquad (3.1.1.6)$$

where

$\underline{x}_m$ is the vector of the node point location (m).

This approximation diagonalizes the coefficient matrix of the temporal-derivative terms. The value of the dependent variable at the node is taken to represent the average value over the cell. Now each cell may consist of up to eight subdomains, as shown in figure 3.3, and each subdomain may have different spatial properties. Thus, the integral of A in equation 3.1.1.6 is actually:

$$\sum_{s=1}^{8} \int_{V_{ms}} A dV = \sum_{s=1}^{8} A_{ms} V_{ms} ; \qquad (3.1.1.7)$$

where

A_{ms} is the value of A in subdomain s of cell m; and

V_{ms} is the volume of subdomain s of cell m (m^3).

The dispersive-flux term is approximated, recognizing that the surface of cell m is composed of six faces, and that each face belongs to four elements, each of which may have different spatial properties (fig. 3.3). Thus:

$$\int_{S_m} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS = \sum_{p=1}^{6} \sum_{q=1}^{4} \int_{S_{mpq}} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS \; ; \qquad (3.1.1.8)$$

where

S_{mpq} is the part of the cell surface that belongs to face p in element q (m^2).

A typical integral over a cell face is of the form, for p = 2, as an example:

$$\int_{S_{m2}} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS = \sum_{q=1}^{4} \int_{S_{m2q}} [B_{xx} \frac{\partial u}{\partial x} + B_{xy} \frac{\partial u}{\partial y} + B_{xz} \frac{\partial u}{\partial z}]dS \; ; \qquad (3.1.1.9)$$

where

$B_{ij}(t)$ are the tensor components of $\underline{\underline{B}}$ for a face whose outward normal points in the ith direction, i=x,y, or z.

A sample subdomain volume for subdomain s=1 in figure 3.3 is:

$$V_1 = \tfrac{1}{2}(x_i - x_{i-1}) \; \tfrac{1}{2}(y_j - y_{j-1}) \; \tfrac{1}{2}(z_k - z_{k-1}) \qquad (3.1.1.10)$$

A sample cell-face area belonging to face p=2 and zone q=1 in figure 3.3 is:

$$S_{m21} = \tfrac{1}{2}\ (y_j - y_{j-1})\ \ \tfrac{1}{2}\ (z_k - z_{k-1}) \qquad (3.1.1.11)$$

Equation 3.1.1.9 is based on the fact that each cell face has a normal vector that is alined with one of the cartesian-coordinate directions. Note that B_{ij} is assumed to be spatially constant over the element q. The partial derivatives are approximated by central differences across each face. Thus, for the face midway between x_i and x_{i+1}, denoted by p=2, the outward normal is in the positive x-direction. An integral over this cell face becomes:

$$\int_{S_{m2}} (\underline{\underline{B}}\cdot\nabla)\cdot\underline{n}dS \cong \sum_{q=1}^{4} [B_{xx}(q)\frac{\partial u}{\partial x}\Big|_{i+\frac{1}{2},j,k} S_{m2q} + B_{xy}(q)\frac{\partial u}{\partial y}\Big|_{q} S_{m2q}$$

$$+B_{xz}(q)\frac{\partial u}{\partial z}\Big|_{q} S_{m2q}] \ ; \qquad (3.1.1.12)$$

where

$\frac{\partial u}{\partial x}\Big|_{i+\frac{1}{2},j,k}$ is the gradient of u in the x-direction across the p=2 face at y_j, z_k;

$\frac{\partial u}{\partial y}\Big|_{q}$ is the gradient of u in the y-direction for the subface in the qth element; and

$\frac{\partial u}{\partial z}\Big|_{q}$ is the gradient of u in the z-direction for the subface in the qth element.

Now

$$\left.\frac{\partial u}{\partial x}\right|_{i+\frac{1}{2},j,k} \cong \frac{u_{i+1,j,k} - u_{i,j,k}}{x_{i+1} - x_i} ; \tag{3.1.1.13}$$

where

$u_{i,j,k}$ is the value of u at node x_i, y_j, z_k.

The approximation for $\frac{\partial u}{\partial y}$ depends on q. For example, for the element bounded by the planes at x_i and x_{i+1}, y_j and y_{j+1}; and z_k and z_{k+1} ; denoted by p=2, q=4 in figure 3.3, we have:

$$\left.\frac{\partial u}{\partial y}\right|_{q=4} \cong \frac{u_{i+\frac{1}{2},j+1,k} - u_{i+\frac{1}{2},j,k},}{y_{j+1} - y_j} \tag{3.1.1.14}$$

where

$$u_{i+\frac{1}{2},j,k} = \tfrac{1}{2}(u_{i+1,j,k} + u_{i,j,k}). \tag{3.1.1.15}$$

Similarly,

$$\left.\frac{\partial u}{\partial z}\right|_{q=4} \cong \frac{u_{i+\frac{1}{2},j,k+1} - u_{i+\frac{1}{2},j,k}}{z_{k+1} - z_k} . \tag{3.1.1.16}$$

The advective-transport term is treated in a similar fashion, but it is somewhat simpler. First, we break it into a sum over the faces and zones:

$$\int_{S_m} (\underline{C}u)\cdot\underline{n}dS = \sum_{p=1}^{6} \sum_{q=1}^{4} \int_{S_{mpq}} (\underline{C}u)\cdot\underline{n}dS . \tag{3.1.1.17}$$

A typical integral over a cell face, p=2, is:

$$\int_{S_{m2}} (\underline{C}u)\cdot\underline{n}dS = \sum_{q=1}^{4} \int_{S_{m2q}} C_x udS \; ; \qquad (3.1.1.18)$$

where

C_x is the vector component of $\underline{C}$ for a face whose outward normal points in the x-direction.

Now if the integral is approximated, for the same example face as above, p=2, by:

$$\int_{S_{m2}} C_x udS \cong \sum_{q=1}^{4} C_x(m,2,q)\tfrac{1}{2}\,(u_{i+1,j,k} + u_{i,j,k})S_{m2q} \; ; \qquad (3.1.1.19)$$

where C_x (m,2,q) is the value of C_x on the face p=2 in element q.

This will lead to a central difference for the advective term of equation 3.1.1.5. If, instead, the following approximation is used:

$$\int_{S_{m2}} C_x udS \;\cong\; \sum_{q=1}^{4} C_x(m,2,q)\; u_{i,j,k}S_{m2q}, \text{ for } C_x > 0 \; ; \qquad (3.1.1.20a)$$

or

$$\cong \sum_{q=1}^{4} C_x(m,2,q)\; u_{i+1,j,k}S_{m2q}, \text{ for } C_x < 0 \; ; \qquad (3.1.1.20b)$$

this will lead to an upstream difference for the advective term. Central differencing may produce oscillations in the solution, whereas upstream differencing cannot (Price and others, 1966; Roache 1976, p. 161-165). But the penalty to eliminate oscillation resulting from spatial differencing is the addition of artificial dispersion (Roache, 1976, p. 64-66; Lantz, 1970), which can be regarded as smearing out of steep concentration or temperature gradients caused by the numerical method rather than the dispersive mixing term.

An approximation has the transportive property, if a disturbance in the field of property u, is advected only in the direction of the velocity. Recall that C represents the velocity in these equations. The central approximation of equation 3.1.1.19 does not have the transportive property, whereas the upstream approximation of equations 3.1.1.20a and 3.1.1.20b does. However, not all upstream approximations have the transportive property (Roache, 1976, p. 69). While the transportive property is desirable on physical grounds, the grid spacing must be limited to avoid excessive artificial dispersion caused by the numerical method. The criteria for avoidance will be presented in a later section. The numerical implementation of the heat- and solute-transport simulator offers the choice of central or upstream differencing for the advective terms. If upstream differencing is selected, the user must determine the grid spacing that limits numerical dispersion to an acceptable amount.

The source term in equation 3.1.1.5 that is linearly proportional to the value of the dependent variable, u, is averaged throughout the cell volume to obtain the finite-difference approximation. The mean-value theorem is used to approximate the integral, with $\underline{x}_m$ being the node-point location. The volume integral is split into the contributions from the eight subdomains. Thus:

$$\int_{V_m} \mathcal{D}u \, dV \cong u(\underline{x}_m,t) \sum_{s=1}^{8} \int_{V_s} \mathcal{D}(s)dV \; ; \qquad (3.1.1.21a)$$

$$\cong u_{i,j,k} \sum_{s=1}^{8} \mathcal{D}_{ms} V_{ms} \; ; \qquad (3.1.1.21b)$$

where

D_{ms} is the value of D in subdomain s of cell m.

The source term at the end of equation 3.1.1.5 is assumed to be a line source in the z-direction of constant intensity, that fully penetrates the cell at $x=x_s$; $y=y_s$. Discretization is achieved by carrying out the integration. Thus:

$$\int_{V_m} E_{sm}\delta(x-x_s)\delta(y-y_s)dV = E_m \quad . \qquad (3.1.1.22)$$

This shows that a line source becomes distributed throughout the cell volume, and the precise location is lost in the finite-difference equation.

Combining equations 3.1.1.6 through 3.1.1.22 gives the finite-difference approximation to equation 3.1.1.1a for an interior node or cell. It is of the form:

$$\begin{aligned}
\frac{\partial}{\partial t}(a_{28}u_{i,j,k}) = {} & a_1\, u_{i-1,j-1,k-1} + a_2\, u_{i,j-1,k-1} + a_3\, u_{i+1,j-1,k-1} \\
& + a_4\, u_{i-1,j,k-1} + a_5\, u_{i,j,k-1} + a_6\, u_{i+1,j,k-1} \\
& + a_7\, u_{i-1,j+1,k-1} + a_8\, u_{i,j+1,k-1} + a_9\, u_{i+1,j+1,k-1} \\
& + a_{10}\, u_{i-1,j-1,k} + a_{11}\, u_{i,j-1,k} + a_{12}\, u_{i+1,j-1,k} \\
& + a_{13}\, u_{i-1,j,k} + a_{14}\, u_{i,j,k} + a_{15}\, u_{i+1,j,k} \\
& + a_{16}\, u_{i-1,j+1,k} + a_{17}\, u_{i,j+1,k} + a_{18}\, u_{i+1,j+1,k} \\
& + a_{19}\, u_{i-1,j-1,k+1} + a_{20}\, u_{i,j-1,k+1} + a_{21}\, u_{i+1,j-1,k+1} \\
& + a_{22}\, u_{i-1,j,k+1} + a_{23}\, u_{i,j,k+1} + a_{24}\, u_{i+1,j,k+1}
\end{aligned}$$

$$+ a_{25} u_{i-1,j+1,k+1} + a_{26} u_{i,j+1,k+1} + a_{27} u_{i+1,j+1,k+1}$$

$$+ a_o \, . \tag{3.1.1.23}$$

It is important to observe that the dependent variable for each interior node is related to 26 other nodal values of that variable, through the finite-difference equation in space. The six nearest neighbors to a given node appear in the terms with coefficients a_5, a_{11}, a_{13}, a_{15}, a_{17}, and a_{23}. The central node is in the term with a_{14}. All of the other terms result from the cross-dispersive flux integrals of equations 3.1.1.8 and 3.1.1.9. Thus it will be advantageous to reduce the bandwidth of the final finite-difference equations by treating the cross-derivative dispersive-flux terms in an approximate manner. This will be covered in section 3.2.

Boundary cells with specified flux are handled similarly to interior cells. With the point-distributed grid, nodes will be located on boundary faces, edges, and corners. The cells associated with these boundary nodes will not have all eight subdomains (fig. 3.5). For example, a lateral boundary cell will have only four subdomains, while a corner boundary cell will have only one. The volume integrations over the cell are carried out as before, with the appropriate reduction in the number of subdomains. Flux-boundary conditions enter the finite-difference approximations through the surface integrations.

Consider a side node where part of the regional boundary is an x-plane; that is, the outward normal to the regional-boundary surface points in the positive x-direction. The associated half-cell for the node consists of four subdomains shown in figure 3.5. The discretization of equation 3.1.1.5 proceeds as follows. The temporal term for the rate-of-change-of-u becomes:

$$\frac{\partial}{\partial t} \int_{V_m} AudV \cong \frac{\partial}{\partial t} u_{i,j,k} \sum_{s=1}^{4} A_{ms} V_{ms} \, . \tag{3.1.1.24}$$

The dispersive- and advective-flux terms are still given by equations 3.1.1.8 and 3.1.1.17, but now one of the faces, denoted by p=2 in figure 3.5, is a boundary face for the region. Note that its outward normal points in the positive x-direction. Using equation 3.1.1.1d for the flux-boundary condition, the integral over this face becomes:

$$\int_{S_{m2}} [\underline{\underline{B}}\cdot\nabla u-\underline{C}u]\cdot\underline{n}dS = - \int_{S_{m2}} \underline{J}\cdot\underline{n}dS \; ; \qquad (3.1.1.25a)$$

$$= - \sum_{q=1}^{4} J_{xq}S_{m2q} \; ; \qquad (3.1.1.25b)$$

where

J_{xq} is the component of vector $\underline{J}$ in the x-direction in the qth element.

Normally J_x is constant over the entire cell face. Thus, the specified-flux boundary conditions has been incorporated in the finite-difference equation as a source term.

The distributed-source and line-source terms simply are adjusted to account for the reduced cell volume. Thus equation 3.1.1.22 is unchanged, but equation 3.1.1.21b becomes:

$$\int_{V_m} DudV \cong u_{i,j,k} \sum_{s=1}^{4} D_{ms}V_{ms} \; . \qquad (3.1.1.26)$$

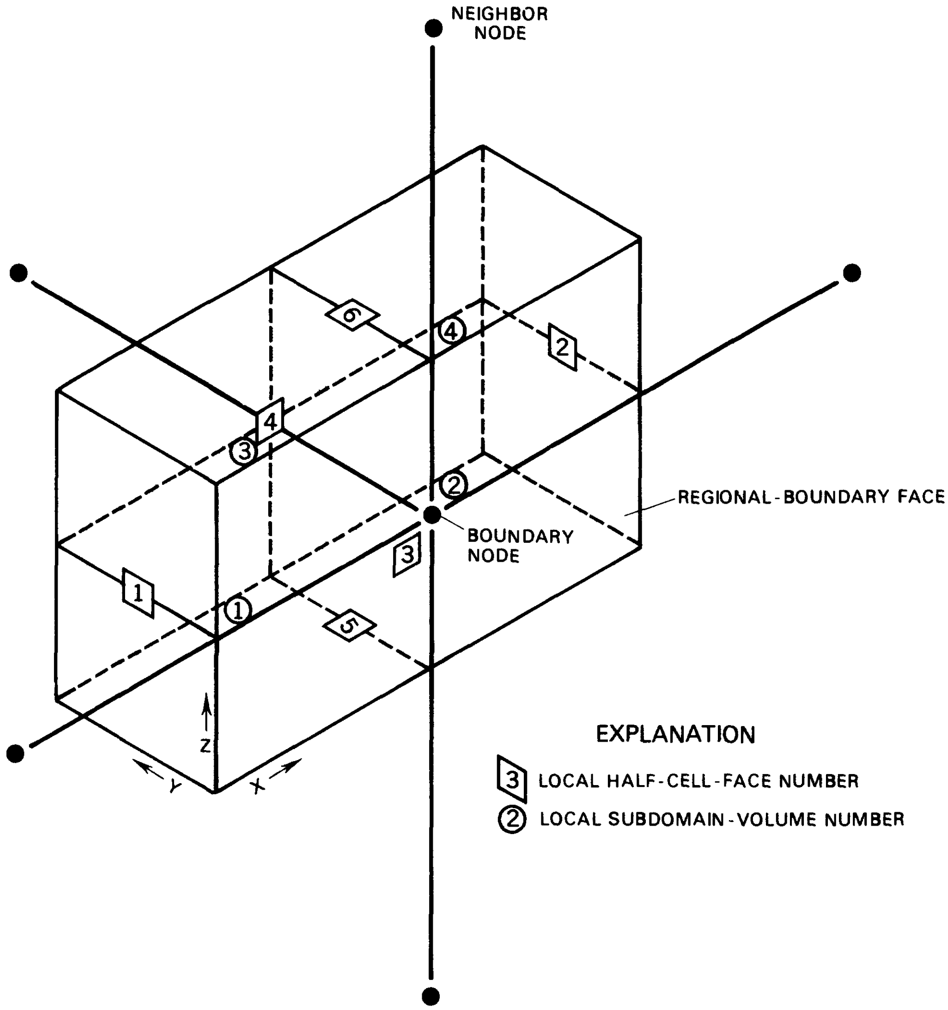


Figure 3.5.--Sketch of a boundary node with its half-cell volume showing the cell faces and the subdomain volumes.

In the flow equation, there is no advective term to cancel the corresponding term in the specified-flux boundary-condition equation. Inside the region, this means $\underline{C} = 0$; but some of the boundary conditions have $\underline{C} \neq 0$. Then equation 3.1.1.9 takes the form:

$$\int_{S_{m2}} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS = \int_{S_{m2}} [(\underline{C}u)\cdot\underline{n} - \underline{J}\cdot\underline{n}]dS \; ; \qquad (3.1.1.27a)$$

which discretizes to:

$$\int_{S_{m2}} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS \cong \sum_{q=1}^{4} [C_x u_{i,j,k} S_q - J_{xq} S_q] \; . \qquad (3.1.1.27b)$$

No central or upstream approximation for u is necessary, because the boundary face contains the node point.

3.1.2 Cylindrical Coordinates

The discretization of equation 3.1.1a in cylindrical coordinates is analogous to what has just been presented. We make the assumption of cylindrical symmetry, so no angular dependence exists. No line source terms can be present so E=0. A cell volume becomes a ring bounded by $r_{i-\frac{1}{2}}$ and $r_{i+\frac{1}{2}}$ and $z_{k-\frac{1}{2}}$ and $z_{k+\frac{1}{2}}$, where $r_{i+\frac{1}{2}}$ is the radius of the cell wall between r_i and r_{i+1} (fig. 3.4).

An option is provided for automatic placement of the radial-grid lines between the interior and exterior radius. With this option the grid lines are spaced according to:

$$\frac{r_{i+1}}{r_i} = \left[\frac{r_{N_r}}{r_1}\right]^{1/(N_r-1)} \; ; \qquad (3.1.2.1)$$

where

N_r is the number of grid points (lines) in the radial direction;

r_{N_r} is the exterior radius (m); and

r_1 is the interior radius (m).

This equation gives logarithmic-node spacing in the radial direction. The cell boundaries for the cylindrical faces are chosen to be at the logarithmic mean radii; for example:

$$r_{i+\frac{1}{2}} = \frac{r_{i+1} - r_i}{\ell n(r_{i+1}/r_i)}; \qquad (3.1.2.2)$$

for both the automatic and user-specified radial-grid distribution. A logarithmic-grid spacing will make the pressure drop uniform between adjacent grid points for steady radial flow in a homogenous medium (Aziz and Settari, 1979, p. 88). The discharge flux at $r_{i+\frac{1}{2}}$ matches the analytical solution under these conditions.

Each cell ring is composed of four subdomain rings (shown in fig. 3.5). Thus, the temporal rate-of-change of u in the cell is approximated by:

$$\frac{\partial}{\partial t} \int_{V_m} AudV \cong \frac{\partial}{\partial t} u_{ik} \sum_{s=1}^{4} A_{ms} V_{ms} ; \qquad (3.1.2.3)$$

where a sample subdomain volume is, for subdomain s=1:

$$V_1 = \pi \, (r_i^2 - r_{i-\frac{1}{2}}^2) \, \tfrac{1}{2} \, (z_k - z_{k-1}) \; . \qquad (3.1.2.4)$$

The surface of cell m is composed for four faces, and each face belongs to two elements. Each element in the z-direction may have different porous-medium properties, but these properties must be constant in the r-direction.

Equation 3.1.1.8 for the dispersive-flux term becomes:

$$\int_{S_m} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS = \sum_{p=1}^{4}\sum_{q=1}^{2}\int_{S_{mpq}} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS \ ; \tag{3.1.2.5}$$

and typical integral over a face of a cell surface, p=2, is

$$\int_{S_{m2}} (\underline{\underline{B}}\cdot\nabla u)\cdot\underline{n}dS = \sum_{q=1}^{2}\int_{S_{m2q}} [B_{rr}\frac{\partial u}{\partial r} + B_{rz}\frac{\partial u}{\partial z}]dS \ ; \tag{3.1.2.6a}$$

$$= \sum_{q=1}^{2} [B_{rr}(q)\frac{\partial u}{\partial r}\bigg|_{i+\frac{1}{2},k} S_{m2q} + B_{rz}(q)\frac{\partial u}{\partial z}\bigg|_{q} S_{m2q}] \ . \tag{3.1.2.6b}$$

Equation 3.1.1.13, with r taking the place of x, is used to approximate $\frac{\partial u}{\partial r}\big|_{i+\frac{1}{2},k}$, and equations 3.1.1.14 and 3.1.1.15 are used to approximate $\frac{\partial u}{\partial z}\big|_{q}$.

Representative cell face areas are, for the face p=1:

$$S_{m_{11}} = 2\pi\, r_{i-\frac{1}{2}}(z_k - z_{k-1}) \ ; \tag{3.1.2.7}$$

and, for the face p=3:

$$S_{m_{31}} = \pi(r_i^2 - r_{i-\frac{1}{2}}{}^2) \ ; \tag{3.1.2.8a}$$

$$S_{m_{32}} = \pi(r_{i+\frac{1}{2}}^2 - r_i^2) \ . \tag{3.1.2.8b}$$

The advective term is treated as in the cartesian-coordinate case, with appropriate reduction in the number of faces and zones. Again, central or upstream differencing in space can be selected. The distributed source term proportional to u is also handled as in equations 3.1.1.21a and 3.1.1.21b with appropriate reduction in the number of subdomains from eight to four.

Finally, the finite-difference approximation to equation 3.1.1.1a for cylindrical coordinates with angular symmetry takes the form:

$$\begin{aligned}\frac{\partial}{\partial t}(a_{10}'u_{i,k}) = {} & a_1'u_{i-1,k-1} + a_2'u_{i,k-1} + a_3'u_{i+1,k-1} \\ & + a_4'u_{i-1,k} + a_5'u_{i,k} + a_6'u_{i+1,k} \\ & + a_7'u_{i-1,k+1} + a_8'u_{i,k+1} + a_9'u_{i+1,k+1} .\end{aligned} \tag{3.1.2.9}$$

Note that no a_0' term exists because annular-ring sources normally are not encountered. The terms a_2', a_4', a_6', and a_8' are contributions from the closest neighbors to the central node point appearing in the a_5' term. The other terms arise from the cross-derivative dispersive-flux integrals, and they may be treated in an approximate fashion to reduce the matrix-band width. The specified-flux boundary conditions are discretized in the same manner as for the cartesian-coordinate system, with appropriate reduction in the number of subdomains and surface faces.

3.1.3 Temporal Discretization

To approximate the time derivative, two options are offered. The first is centered-in-time differencing, commonly known as the Crank-Nicholson method. The time derivative is approximated by the finite difference:

$$\frac{\partial}{\partial t} (a_m u_m) \cong \frac{(a^m u_m)^{n+1} - (a^m{}_m)^n}{t^{n+1} - t^n} ; \qquad (3.1.3.1)$$

where t_n is the time at level n.

The right-hand-side, F, of the equation concerned is evaluated as follows:

$$F \cong \tfrac{1}{2}(F^{n+1} + F^n) ; \qquad (3.1.3.2)$$

where

F^n is the spatial-difference function evaluated at time n.

The other option is backward-in-time differencing, which has the form:

$$F \cong F^{n+1} \qquad (3.1.3.3)$$

As with the advective spatial differencing, central-in-time differencing has the potential for causing oscillations in the solution (Price and others, 1966; Smith and others, 1977), whereas backwards-in-time differ-encing does not. However, backwards-in-time differencing does introduce numerical dispersion that must be kept under control by limiting the time-step size (Lantz, 1970; Price and others, 1966; Smith and others, 1977; Briggs and Dixon, 1968).

Equations 3.1.3.1 through 3.1.3.3 for the time discretization can be combined into a general form as:

$$\frac{(a_m u_m)^{n+1} - (a_m u_m)^n}{t^{n+1} - t^n} = \theta F^{n+1} + (1-\theta)F^n; \tag{3.1.3.4}$$

where $\theta = 1$ gives the fully implicit or backward-in-time (BT) differencing, and $\theta = \frac{1}{2}$ gives the Crank-Nicholson or centered-in-time (CT) differencing.

The next step is to express the difference equation in residual form by writing:

$$u^{n+1} = u^n + \delta u \; ; \tag{3.1.3.5}$$

where

δu is the temporal change in u.

Equation 3.1.3.5 is inserted in equation 3.1.3.4 and the following expansions of the temporal-difference terms are used. These are consistent differencing expansions that correspond to the differentials of products. For terms of the form $(a_i u)^{n+1}$ we have:

$$(a_i u)^{n+1} - (a_i u)^n = a_i^{n+1} \delta u + u^n \delta a_i \; ; \tag{3.1.3.6}$$

for terms of the form $(a_i a_j u)^{n+1}$, we have:

$$(a_i a_j u)^{n+1} - (a_i a_j u)^n = a_i^{n+1} a_j^{n+1} \delta u + a_i^{n+1} u^n \delta a_j + a_j^n u^n \delta a_i . \tag{3.1.3.7}$$

3.1.4 Finite-Difference Flow and Transport Equations

Combining equations 3.1.3.4, 3.1.3.6, and 3.1.3.7 with 3.1.1.23 or 3.1.2.9, we obtain the form of the general finite-difference equation for an interior node. The rather large number of terms makes presentation of the general equation impractical. It is more instructive to present the discretized flow, and heat and solute-transport equations individually, showing the x-direction terms only. The additional dispersive and convective terms for the y and z-directions follow the same pattern as their counterparts in the x-direction.

The finite-difference approximation to the flow equation (2.3.1a) is, for an interior node m:

$$C_{33}\delta p_m + C_{32}\delta T_m + C_{31}\delta w_m = \theta \mathcal{T}_{Fi+\frac{1}{2}} (\delta p_{i+1} - \delta p_i) - \theta \mathcal{T}_{Fi-\frac{1}{2}}(\delta p_i - \delta p_{i-1})$$

$$+ \mathcal{T}_{Fi+\frac{1}{2}}(p^n_{i+1} - p^n_i + \rho^n_{i+\frac{1}{2}} g \, (x_{i+1} - x_i))$$

$$- \mathcal{T}_{Fi-\frac{1}{2}}(p^n_i - p^n_{i-1} + \rho^n_{i-\frac{1}{2}} g \, (x_i - x_{i-1}))$$

$$+ Q^n_m \rho^* + \theta \frac{\partial Q^n_m}{\partial p} \rho^* \, \delta p_i$$

$$+ \text{ y and z direction difference terms}; \qquad (3.1.4.1a)$$

where

$$C_{33} = [\rho^{n+1}_m \sum_{s=1}^{8} \alpha_{bs} V_s + \rho_o \beta_p \sum_{s=1}^{8} \varepsilon^n_s V_{ms}]/\delta t \; ; \qquad (3.1.4.1b)$$

$$C_{32} = [\rho_o \beta_T \sum_{s=1}^{8} \varepsilon^n_s V_{ms}]/\delta t \; ; \qquad (3.1.4.1c)$$

$$C_{31} = [\rho_o \beta_w \sum_{s=1}^{8} \varepsilon^n_s V_{ms}]/\delta t \; ; \qquad (3.1.4.1d)$$

$$T_{Fi+\frac{1}{2}} = \frac{\rho^n_{i+\frac{1}{2}}}{\mu^n_{i+\frac{1}{2}}(x_{i+1}-x_i)} \sum_{q=1}^{4} k_{m2q}S_{m2q} \; ; \qquad (3.1.4.1e)$$

$$T_{Fi-\frac{1}{2}} = \frac{\rho^n_{i-\frac{1}{2}}}{\mu^n_{i-\frac{1}{2}}(x_i-x_{i-1})} \sum_{q=1}^{4} k_{m1q}S_{m1q} \; ; \qquad (3.1.4.1f)$$

where

C_{ij} are the capacitance factors (various);

T_{Fi} are the conductance terms for flow (m-s); and

Q^n_m is the volumetric source flow rate for cell m (m^3/s).

The flow-conductance factors, $\tilde{T}_{Fi}$, (m^3), are defined as:

$$\tilde{T}_{Fi+\frac{1}{2}} = \sum_{q=1}^{4} k_{m2q}S_{m2q} \,/\, (x_{i+1}-x_i). \qquad (3.1.4.2)$$

These factors contain the spatial information and are constants.

In equation 3.1.4.1a, the source-sink flow rate has been made semi-implicit in time by including a term that accounts for changes in flow rate with changes in cell pressure. It is semi-implicit, because only the flow rate that contributes to the equation for cell m is treated implicitly. The effect of a change in pressure in cell m on the source-sink flow rates for other cells coupled to the cell through a well bore are not included; this approach avoids enlarging the bandwidth of the system equation matrix, equation 3.6.1a.

The finite-difference approximation to the heat-transport equation (2.3.1b) is, for an interior node, m:

$$C_{23}\delta p_m + C_{22}\delta T_m + C_{21}\delta w_m = \theta \mathcal{T}_{Hi+\frac{1}{2}}(\delta T_{i+1}-\delta T_i) - \theta \mathcal{T}_{Hi-\frac{1}{2}}(\delta T_i-\delta T_{i-1})$$

$$+ \mathcal{T}_{Hi+\frac{1}{2}}(T^n_{i+1}-T^n_i) - \mathcal{T}_{Hi-\frac{1}{2}}(T^n_i-T^n_{i-1})$$

$$- \theta S^{n+1}_{xi+\frac{1}{2}}\, c_f \delta\, T_{i+\frac{1}{2}} - \theta \delta S_{xi+\frac{1}{2}} c_f T^n_{i+\frac{1}{2}}$$

$$+ \theta S^{n+1}_{xi-\frac{1}{2}}\, c_f \delta\, T_{i-\frac{1}{2}} + \theta \delta S_{xi-\frac{1}{2}}\, c_f T^n_{i-\frac{1}{2}}$$

$$- S^n_{xi+\frac{1}{2}}\, c_f\, T^n_{i+\frac{1}{2}} + S^n_{xi-\frac{1}{2}}\, c_f\, T^n_{i-\frac{1}{2}}$$

$$+ \mathcal{T}_{Hxy\ i+\frac{1}{2}}\,(T^n_{i+1,j+1,k} + T^n_{i,j+1,k} - T^n_{i+1,j-1,k} - T^n_{i,j-1,k})$$

$$+ \mathcal{T}_{Hxz\ i+\frac{1}{2}}\,(T^n_{i+1,j,k+1} + T^n_{i,j,k+1} - T^n_{i+1,j,k-1} - T^n_{i,j,k-1})$$

$$- \mathcal{T}_{Hxy\ i-\frac{1}{2}}\,(T^n_{i,j+1,k} + T^n_{i-1,j+1,k} - T^n_{i,j-1,k} - T^n_{i-1,j-1,k})$$

$$- \mathcal{T}_{Hxz\ i-\frac{1}{2}}\,(T^n_{i,j,k+1} + T^n_{i-1,j,k+1} - T^n_{i,j,k-1} - T^n_{i-1,j,k-1})$$

$$+ Q_m^n \rho^* c_f T_m^{*n} + \theta \frac{\partial Q_m^n}{\partial p} \rho^* \delta p_m T_m^{*n} + \theta Q_m^n \rho^* c_f \delta T_m^*$$

$$+ \theta \frac{\partial Q_m^n}{\partial p} \rho^* c_f \delta p_m \delta T_m^*$$

+ y and z direction dispersive, cross-dispersive

and advective flux terms; (3.1.4.3a)

where

$$C_{23} = [\rho_o \beta_p H_m^{n+1} \sum_{s=1}^{8} \varepsilon_s^n V_s + \rho_m^{n+1} H_m^{n+1} \sum_{s=1}^{8} \alpha_{bs} V_{ms}$$

$$- T_m^{n+1} \sum_{s=1}^{8} (\rho_s c_s)_s \alpha_{bs} V_{ms}]/\delta t \; ; \tag{3.1.4.3b}$$

$$C_{22} = [\rho_m^{n+1} c_f \sum_{s=1}^{8} \varepsilon_s^n V_s + \sum_{s=1}^{8} (1-\varepsilon_s^n)(\rho_s c_s)_s V_{ms} +$$

$$\rho_o \beta_T H_m^n \sum_{s=1}^{8} \varepsilon_s^n V_{ms}]/\delta t \; ; \tag{3.1.4.3c}$$

$$C_{21} = [\rho_o \beta_w H_m^n \sum_{s=1}^{8} \varepsilon_s^n V_{ms}]/\delta t \; ; \tag{3.1.4.3d}$$

$$T_{Hi+\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n D_{Hxx}(2,q)S_{m2q} + K_f \sum_{q=1}^{4} \varepsilon_q^n S_{m2q} +$$

$$\sum_{q=1}^{4} (1-\varepsilon_q^n)K_{sq}S_{m2q}]/(x_{i+1}-x_i) \; ; \qquad (3.1.4.3e)$$

$$T_{Hi-\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n D_{Hxx}(1,q)S_{m1q} + K_f \sum_{q=1}^{4} \varepsilon_q^n S_{m1q} +$$

$$\sum_{q=1}^{4} (1-\varepsilon_q^n)K_{sq}S_{m1q}]/(x_i-x_{i-1}) \; ; \qquad (3.1.4.3f)$$

$$S_{xi+\frac{1}{2}}^n = (\rho v_x)_{i+\frac{1}{2}}^n \sum_{q=1}^{4} \varepsilon_q^n S_{m2q} \; ; \qquad (3.1.4.3g)$$

$$S_{xi-\frac{1}{2}}^n = (\rho v_x)_{i-\frac{1}{2}}^n \sum_{q=1}^{4} \varepsilon_q^n S_{m1q} \; ; \qquad (3.1.4.3h)$$

$$\delta S_{xi+\frac{1}{2}} = \rho_{i+\frac{1}{2}}^n \delta v_{xi+\frac{1}{2}} \sum_{q=1}^{4} \varepsilon_q^n S_{m2q} \; ; \qquad (3.1.4.3i)$$

$$\delta S_{xi-\frac{1}{2}} = \rho_{i-\frac{1}{2}}^n \delta v_{xi-\frac{1}{2}} \sum_{q=1}^{4} \varepsilon_q^n S_{m1q} \; ; \qquad (3.1.4.3j)$$

$$T_{Hxy\ i+\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n D_{Hxy}(2,q)S_{m2q}]/(y_{j+1}-y_j) \; ; \qquad (3.1.4.3k)$$

$$T_{Hxz\ i+\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n D_{Hxz}(2,q)S_{m2q}]/(z_{k+1}-z_k)\ ; \qquad (3.1.4.3\ell)$$

$$H = H(T_{oH}) + \overline{c_f}\ (T-T_{oH}); \qquad (3.1.4.3m)$$

where T_{Hi} are the thermal conductance terms (W/°C).

In equation 3.1.4.3a, the same semi-implicit treatment of the source-sink flow rate has been incorporated as in equation 3.1.4.1a.

The central-or upstream-weighted value for the variables v_x, δv_x, T and δT is given by the general form:

$$u_{i+\frac{1}{2}} = (1-\sigma)\ u_i + \sigma\ u_{i+1} \qquad (3.1.4.4)$$

where

σ is the spatial weighting coefficient.

Central weighting is obtained with $\sigma = \frac{1}{2}$; upstream weighting is obtained with $\sigma = 0$ for a positive v_x.

The finite-difference approximation to the solute-transport equation (2.3.1c) is, for an interior node, m:

$$C_{13}\delta p_m + C_{12}\delta T_m + C_{11}\delta w_m = \theta T_{Si+\frac{1}{2}}(\delta w_{i+1} - \delta w_i) - \theta T_{Si-\frac{1}{2}}(\delta w_i - \delta w_{i-1})$$

$$+ T_{Si+\frac{1}{2}}(w^n_{i+1} - w^n_i) - T_{Si-\frac{1}{2}}(w^n_i - w^n_{i-1})$$

$$-\theta S^{n+1}_{xi+\frac{1}{2}}\, \delta w_{i+\frac{1}{2}} - \theta \delta S_{xi+\frac{1}{2}} w^n_{i+\frac{1}{2}}$$

$$+\theta S^{n+1}_{xi-\frac{1}{2}}\, \delta w_{i-\frac{1}{2}} + \theta\, \delta S_{xi-\frac{1}{2}} w^n_{i-\frac{1}{2}}$$

$$- S^n_{xi+\frac{1}{2}}\, w^n_{i+\frac{1}{2}} + S^n_{xi-\frac{1}{2}}\, w^n_{i-\frac{1}{2}}$$

$$+ T_{Sxy\ i+\frac{1}{2}}(w^n_{i+1,j+1,k} + w^n_{i,j+1,k} - w^n_{i+1,j-1,k} - w^n_{i,j-1,k})$$

$$+ T_{Sxz\ i+\frac{1}{2}}(w^n_{i+1,j,k+1} + w^n_{i,j,k+1} - w^n_{i+1,j,k-1} - w^n_{i,j,k-1})$$

$$- T_{Sxy\ i-\frac{1}{2}}(w^n_{i,j+1,k} + w^n_{i-1,j+1,k} - w^n_{i,j-1,k} - w^n_{i-1,j-1,k})$$

$$- T_{Sxz\ i-\frac{1}{2}}(w^n_{i,j,k+1} + w^n_{i-1,j,k+1} - w^n_{i,j,k-1} - w^n_{i-1,j,k-1})$$

$$- \lambda M^n_m\, w^n_m - \theta\lambda(w^{n+1}_m\, \delta M_m + M^n_m\, \delta w_m)$$

$$+ Q^n_m\, \rho^* w^{*n}_m + \theta \frac{\partial Q^n_m}{\partial p}\, \delta p_m \rho^* w^{*n}_m + \theta Q^n_m\, \delta(\rho^* w^*)_m$$

$$+ \theta \frac{\partial Q^n_m}{\partial p}\, \delta p_m\, \delta(\rho^* w^*)_m$$

+ y and z direction dispersive, cross-dispersive and advective-flux terms; (3.1.4.5a)

where

$$C_{13} = [\rho_o \beta_p \, w_m^n \sum_{s=1}^{8} K_s^n \, V_s + \rho_m^{n+1} \, w_m^{n+1} \sum_{s=1}^{8} \alpha_{bs} V_{ms}]/\delta t \qquad (3.1.4.5b)$$

$$C_{12} = [\rho_o \beta_T \, w_m^n \sum_{s=1}^{8} K_s^n \, V_{ms}]/\delta t \; ; \qquad (3.1.4.5c)$$

$$C_{11} = [\rho_o \beta_w w_m^n + \rho_m^{n+1}] \sum_{s=1}^{8} K_s^n \, V_{ms}/\delta t \; ; \qquad (3.1.4.5d)$$

$$T_{Si+\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n \, D_{Sxx}(2,q) S_{m2q} + D_m \sum_{q=1}^{4} \varepsilon_q^n \, S_{m2q}] \; \frac{\rho_{i+\frac{1}{2}}^n}{x_{i+1}-x_i} \; ; \qquad (3.1.4.5e)$$

$$T_{Si-\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n \, D_{Sxx}(1,q) S_{m1q} + D_m \sum_{q=1}^{4} \varepsilon_q^n \, S_{m1q}] \; \frac{\rho_{i-\frac{1}{2}}^n}{x_i-x_{i-i}} \; ; \qquad (3.1.4.5f)$$

$$T_{Sxy\; i+\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n \; D_{Sxy}(2,q) S_{m2q}] \; \frac{\rho_{i+\frac{1}{2}}^n}{y_{j+1}-y_j} \; ; \qquad (3.1.4.5g)$$

$$T_{Sxz\; i+\frac{1}{2}} = [\sum_{q=1}^{4} \varepsilon_q^n \; D_{Sxz}(2,q) S_{m2q}] \; \frac{\rho_{i+\frac{1}{2}}^n}{z_{k+1}-z_k} \; ; \qquad (3.1.4.5h)$$

$$M_m^n = \rho_m^n \sum_{s=1}^{8} K_s^n \, V_s \; ; \qquad (3.1.4.5i)$$

$$\delta M_m = \rho_m^{n+1} \sum_{s=1}^{8} \alpha_{bs} V_s \delta p_m$$

$$+ \sum_{s=1}^{8} (K_s^n \, V_s)[\rho_o \beta_p \delta p_m + \rho_o \beta_T \delta T_m + \rho_o \beta_w \delta w_m] \; ; \qquad (3.1.4.5j)$$

$$K_s^n = \varepsilon_s^n + (\rho_b K_d)_s \; ; \qquad (3.1.4.5k)$$

where

M_m^n is the mass of fluid plus the effective additional fluid mass from sorption in cell m at time level n (kg);

T_{Si} are the conductance terms for solute transport (kg/s); and

K_s is the augmented porosity factor for subdomain s(-).

In equation 3.1.4.5a, the same semi-implicit treatment of the source-sink flow rate has been incorporated as in equation 3.1.4.1a. In equations 3.1.4.1a-f, 3.1.4.3a-ℓ, and 3.1.4.5a-k, subscripts pertaining to the y and z directions have been omitted for clarity, unless necessary. The source density, ρ^*, temperature, T^*, and mass fraction, w^*, are specified functions of time and source location. When the source-flow rate is negative, so that it becomes a sink, the density, temperature, and mass fraction become those of the cell. In the abbreviated subscript notation, u_m and u_i become identical for a given variable, u. Note that the cross-dispersive flux terms have been evaluated explicitly, that is, at time n, to limit the number of elements in the coefficient matrix of the unknowns, $\underline{\underline{A}}$, to a maximum of seven for each equation. The coefficients $\mathcal{T}_i$, $\mathcal{S}_i$, and $\mathcal{M}_i$ are evaluated at time n.

The preceding flow and transport equations are valid for confined flow. The forms of the capacitance terms that contain the porous-medium bulk compressibility are based on a slightly compressible porous matrix and a cell volume that deforms slightly in space. The coefficients C_{ij}, and the cell facial areas S_{mpq} are modified for the case of unconfined flow, as will be shown in section 3.4.6.

The permeability tensor in the flow equation is a diagonal matrix in the numerical implementation, because the coordinate directions are chosen to be along the principal directions of this tensor. These directions are assumed not to change with position in the simulation region. The finite-element discretization technique must be used for the more general situation of spatially variable, anisotropic, permeability directions.

In summary, the properties and variables that are spatially discretized on a cell-by-cell basis include pressure, temperature, solute-mass fraction, density, viscosity, enthalpy, and specified fluxes. Porous-matrix properties that are discretized on an element-by-element or zonal basis include porosity, permeability, thermal conductivity, heat capacity, bulk compressibility, bulk density, equilibrium-distribution coefficient, longitudinal dispersivity, and transverse dispersivity. Well-completion intervals also are designated on a zonal basis.

3.1.5 Numerical Dispersion and Oscillation Criteria

For guidance in selecting the spatial and temporal discretization method, the following results have been obtained by Lantz (1970), Roache (1976, p. 19, 48), Smith and others (1977) and Price and others (1966), expressing the truncation errors that give rise to numerical dispersion and criteria for avoiding oscillations in the solution. They were derived for the one-dimensional form of equation 3.1.1a with constant coefficients and no source terms; that is:

$$A \frac{\partial u}{\partial t} = B \frac{\partial^2 u}{\partial x^2} - C \frac{\partial u}{\partial x} \ . \tag{3.1.5.1}$$

Similar analyses can be performed for the more general equation:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(B \frac{\partial u}{\partial x}\right) - C \frac{\partial u}{\partial x} + Du + E \ ; \tag{3.1.5.2}$$

where B, C, D, and E are functions of x and t, and A is positive.

The truncation errors and oscillation criteria for both equation forms are given in table 3.1. The maximum values of A, B, and C should be used in the variable coefficient case, equation 3.1.5.2. All of the methods are stable in the sense that errors do not grow without bound. However, oscillations in space and time may persist without growth or decay. The oscillation criterion for the centered-in-time differencing was presented by Keller (1960, p. 140) and Briggs and Dixon (1968). They are sufficient conditions; thus, they may be conservative. Alternate conditions appear in Price and others (1966) but they require knowledge of the maximum or minimum eigenvalue of the spatial-discretization matrix that cannot be expressed analytically. An important thing to note from table 3.1 is that it is possible for oscillations in the solution to arise from both spatial and temporal discretization. For the flow equation with no advective term, oscillations from temporal discretization are still possible. For the flow equation in cylindrical coordinates, an advective-type term appears so oscillations can also be caused by spatial discretization. If a source term that depends on u appears in equation 3.1.5.2, the oscillation criteria for the centered-in-time discretization are modified as shown.

When using the backwards-in-space (upstream) or backward-in-time differencing, one needs to check that the truncation-error terms that cause numerical dispersion do not become large relative to the physical-dispersion coefficient. Mathematically, for a dispersion coefficient given by equation 2.2.6.1.2; one needs to adhere to the following criteria:

$$\frac{\Delta x}{2} << \alpha_L \; ; \qquad (3.1.5.3)$$

and

$$\frac{C\delta t}{2} << \alpha_L \; ; \qquad (3.1.5.4)$$

where

$$\Delta x = x_{i+1} - x_i \; ; \text{ and} \qquad (3.1.5.5a)$$

$$\delta t = t^{n+1} - t^n \; . \qquad (3.1.5.5b)$$

Note that these results are from a one-dimensional analysis with constant coefficients, but they give guidance for grid and time-step selection. Table 3.1 shows that, in the case of variable coefficients, additional truncation-error terms occur with backwards-in-time differencing, that can give rise to numerical-dispersion errors.

Table 3.1.--*Truncation errors and oscillation criteria for one-dimensional parabolic equations*

[BS, backward-in-space; BT, backward-in-time; CS, centered-in-space; CT, centered-in-time;

$$u_{xx} = \frac{\partial^2 u}{\partial x^2} \; ; \; u_x = \frac{\partial u}{\partial x} \quad .]$$

Discretization Method	Truncation Error	Oscillation Criterion
Equation 3.1.5.1		
BS	$\frac{C\Delta x u}{2}{}_{xx}$	---
BT	$\frac{C^2\Delta t u}{2}{}_{xx}$	---
CS	$0(\Delta x^2)$	$\Delta x \leq \frac{2B}{\lvert C\rvert}$
CT	$0(\Delta t^2)$	$\frac{\Delta t}{(\Delta x)^2} \leq \frac{A}{B}$
Equation 3.1.5.2		
BS	$\frac{C\Delta x u}{2}{}_{xx}$	---
BT	$\frac{C^2\Delta t u}{2}{}_{xx} + B_t \Delta t u_{xx} + BDu_{xx}$ $+ 3B_x C u_{xx} + 2B_x^2 \; u_{xx}$	---
CS	$0(\Delta x^2)$	$\Delta x \leq \frac{2B}{\lvert C\rvert}$
CT	$0(\Delta t^2)$	$\Delta t \leq \mathrm{MIN} \left(\frac{1}{\frac{B}{\Delta x^2} - \frac{D}{2}}, \frac{2}{D}\right); D>0$ or $\Delta t \leq \frac{1}{\frac{B}{\Delta x^2} - \frac{D}{2}}; D \leq 0$

3.1.6 Automatic Time-Step Algorithm

Manual time-step selection can be difficult, when many source terms and boundary conditions change considerably with time. In general, the more rapidly the conditions change, the smaller the time steps will need to be for an accurate solution. Therefore, the heat- and solute-transport simulator has an automatic time-step option that uses an empirical algorithm (INTERCOMP Resource Development and Engineering, Inc., 1976). The user specifies the maximum values of change in pressure, temperature, and mass fraction considered acceptable as well as the maximum and minimum time step allowed. Then, at the beginning of each time step, the following adjustments are made, depending on the conditions:

$$\text{if: } |\delta u_{max}| > \delta u^s_{max} \; ; \; \delta t = \tfrac{1}{2}\delta t_o \left(1 + \frac{\delta u^s_{max}}{|\delta u_{max}|}\right) ; \qquad (3.1.4.1)$$

otherwise, if:

$$0 < |\delta u_{max}| < \delta u^s_{max} \; ; \; \delta t = \delta t_o \left(0.2 + 0.8 \frac{\delta u^s_{max}}{|\delta u_{max}|}\right) ; \qquad (3.1.4.2)$$

otherwise, if:

$$\delta u_{max} = 0; \; \delta t = 1.5\delta t_o \; ; \qquad (3.1.4.3)$$

where

u is pressure, temperature, or mass fraction;

δu^s_{max} is the specified maximum change in u;

δt is the new time step;

δt_o is the previous time step; and

$|\delta u_{max}|$ is the absolute value of the maximum-calculated change in u over the previous time step.

The new time step is selected to be the minimum of the three that were calculated on the basis of change in the pressure, temperature, and mass fraction. The time step is constrained to a user-specified range, and the maximum increase in δt is limited to a factor of 1.5. This algorithm tends to increase the time step such that the maximum acceptable change in pressure, temperature, or mass fraction is achieved as the simulation progresses. The minimum required time step, set by the user, is maintained for the first two steps after boundary-condition changes occur or after the automatic time-step algorithm is invoked.

3.1.7 Discretization Guidelines

No complete set of discretization rules exists that will guarantee an accurate solution discretization with a minimum number of nodes and time steps, even for the case of constant coefficients. However, the following empirical guidelines should be useful.

1. If using the backward-in-space or backward-in-time differencing, make some estimates of the truncation error, using parameter values at their limits expected for the simulation. Thus, verify that the grid-spacing and time-step selection do not introduce excessive numerical dispersion.

2. If using centered-in-space and centered-in-time differencing, print results every time step for a short simulation period, 5-10 time steps. Examine the results for spatial and temporal oscillations that are caused by the time or space discretization being too coarse.

3. Check on spatial-discretization error by refining the mesh. However, this often is impractical for large regions. A check on temporal-discretization error is relatively easy to make by refining the time-step length for a short simulation.

4. At each change of boundary condition or source flow rates, reduce the time step until the abrupt changes have had time to propagate into the region. The automatic time-step algorithm does this.

5. To adequately represent a sharp solute-concentration or temperature front, span it with at least 4-5 nodes. A large number of nodes may be required if a sharp front moves through much of the region over the simulation time. Compromises often will have to be made. An advantage of the centered-in-space differencing is that oscillations will reveal when the grid is too coarse relative to the gradients of solute concentration or temperature.

6. Well flows that highly stress the aquifer require a small time step after a change in flow rate, to control errors from explicit flow-rate allocation or explicit well-datum pressure calculation.

7. Sometimes, the global-balance summary table will indicate that the time step is too large by exhibiting large residuals, particularly if the density and viscosity variations are large.

8. To check for unusual results that could indicate discretization error, print out all of the results some of the time, and some of the results all of the time.

3.2 PROPERTY FUNCTIONS AND TRANSPORT COEFFICIENTS

Numerical implementation of the fluid-density function is simply the evaluation of equation 2.2.1.1b or 2.2.1.3a. Fluid viscosity is obtained by evaluation of equation 2.2.2.1 and equation 2.2.2.2 if necessary. The enthalpy of pure water at the selected reference values of pressure and temperature, $H(p_{oH},T_{oH},0)$ is obtained by a two-step interpolation. First, the enthalpy of saturated fluid at the given temperature is calculated by linear interpolation in the table of saturated enthalpy as a function of temperature; then, adjustment to the given pressure is made by bilinear interpolation in the table of enthalpy deviation from saturation as a function of pressure and temperature. This procedure is given by equation 2.2.3.1a. A sequential search is made for each interpolation, since the number of pressure or temperature entries is 32 or less in both tables. Equation 2.2.3.1c is used for all subsequent fluid-enthalpy calculations. It is possible that simulation of wide variations in pressure and temperature could require a table look-up for all enthalpy calculations, and the algorithm in the program code would need to be modified.

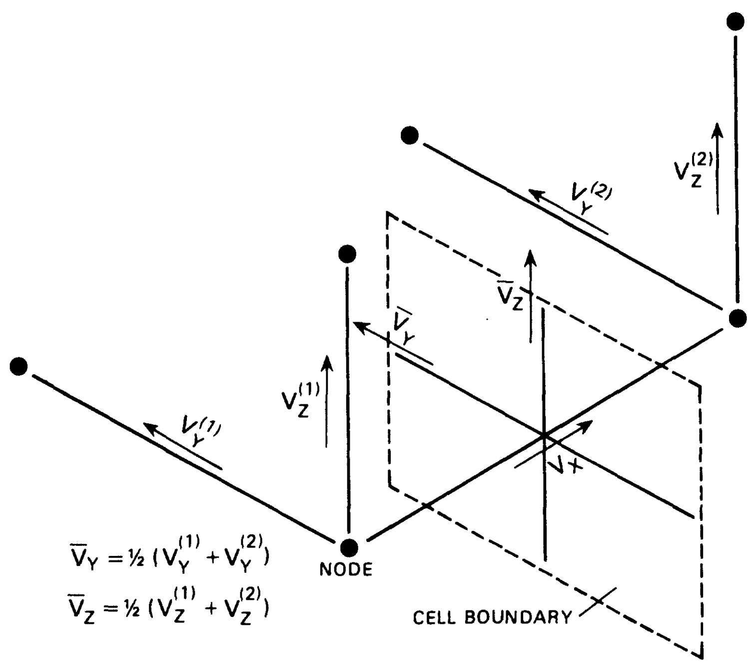


Figure 3.6.--Sketch of velocity vectors used for the dispersion-coefficient calculation for a given cell.

The hydrodynamic-dispersion coefficient is calculated by equation 2.2.6.1.1 with equations 2.2.6.1.2 and 2.2.6.1.3. A separate value of $D_{sij}(p,q)$ is associated with each element, q, of each cell face, p. Interstitial velocities are obtained from the pressure and elevation differences across the face for the velocities normal to the face. Velocities parallel to the face are determined by averaging velocities from each side of the face. An x-face, with the y and z velocities interpolated to get the effective values on the subface appears in figure 3.6. Average values are used, since the face lies midway between x_i and x_{i+1}.

The thermo-hydrodynamic-dispersion tensor is calculated by equations 2.2.6.2.1 and 2.2.6.2.2. The porosity and the thermal conductivities are defined by zones and the interstitial velocities are obtained the same as for the hydrodynamic dispersion.

Two methods are available in the HST3D simulator for computation of the cross-derivative dispersive-flux terms. The most rigorous treatment of the cross-derivative terms involves explicit calculation. They are lagged one iteration in the solution cycle of the flow, heat, and solute equations. The cross-derivative dispersive fluxes are recalculated for each iteration based on the conditions existing at the end of the previous iteration and then they are incorporated into the right-hand-side vector. Therefore at least two iterations in the solution cycle are required at each time step. This full treatment requires storage of the nine dispersion-coefficient terms for thermal and solute dispersion. An approximate empirical treatment of the cross-derivative dispersion terms is available also, that consists of lumping the cross-derivative dispersion coefficients into the diagonal dispersion-coefficient terms. The three augmented dispersion coefficients for thermal and solute dispersion are the only coefficients stored, and extra iterations are not required, because the cross-derivative dispersive fluxes are not computed.

3.3 SOURCE OR SINK TERMS--THE WELL MODEL

In the present version of the HST3D program, only one well can exist in a particular cell. Multiple wells in a cell must be represented by an equivalent single well, or the spatial grid must be refined to separate them. This restriction includes wells that are located in the same areal cell that are completed in different vertical intervals.

Recall that a cell may contain up to four zones of different porous-media properties over a given areal plane. If a well is completed in a cell with multiple zones, the effective ambient permeability is taken to be that of the lowest zone number. This is because no algorithm presently exists to calculate the effective ambient permeability for a well in areally heterogeneous porous media.

3.3.1 The Well-Bore Model

The volumetric flow rate per unit length of well bore is given by equation 2.4.1.1. Discretization for a given cell, m, is achieved by choosing the average pressure to be the cell pressure, and multiplying by the length of well bore in that cell. Since the well bore is usually screened over the more permeable zones of the formation region, the screened intervals are specified by zones or sets of elements rather than by cells. The upper and lower parts of a screened interval will be one-half of the cell thickness in length, unless the cell in question is an upper or a lower boundary cell for the region. Thus:

$$Q_{w\ell} = \frac{(p_{w\ell}-p_m)\ [W_I(\ell_1)L_{\ell 1} + W_I(\ell_2)L_{\ell 2}]}{\mu_{m(\ell)}} \qquad (3.3.1.1a)$$

In the case of an unconfined aquifer with a well screened through the free surface, the screened length, L_ℓ, is adjusted as the saturated thickness varies in time.

where

$$L_{\ell 1} = \tfrac{1}{2}(z_\ell - z_{\ell-1}) \; ; \tag{3.3.1.1b}$$

$$L_{\ell 2} = \tfrac{1}{2}(z_{\ell+1} - z_\ell) \; ; \tag{3.3.1.1c}$$

and where

$Q_{w\ell}$ is the volumetric-flow rate from the well to the aquifer in cell m at well-bore level ℓ (m^3/s);

p_m is the pressure at node m (Pa);

$p_{w\ell}$ is the pressure in the well bore at elevation of node m (Pa);

$m(\ell)$ is the cell number associated with the well-bore level ℓ;

$L_{\ell 1}$ is the length of well bore in the lower half of cell $m(\ell)$ (m); and

$L_{\ell 2}$ is the length of well bore in the upper half of cell $m(\ell)$ (m).

Equations 3.3.1.1b and 3.3.1.1c are valid for the z-coordinate directed vertically upward.

For wells drilled at an angle θ_w to the vertical:

$$L_{\ell 1} = \frac{\tfrac{1}{2}(z_\ell - z_{\ell-1})}{\cos \theta_w} \; ; \tag{3.3.1.2a}$$

$$L_{\ell 2} = \frac{\tfrac{1}{2}(z_{\ell+1} - z_\ell)}{\cos \theta_w} \tag{3.3.1.2b}$$

The well indices may be different in the upper and lower halves of the cell, because the porous-medium zone boundaries pass through planes of node points. The two-term sum in equation 3.3.1.1a accounts for this. When the cell is at the upper or lower boundary of the region, or at the ends of the screened interval for the well, the appropriate term in equation 3.3.1.1a becomes zero.

For notational convenience, we define:

$$M_{w\ell}L_{\ell} = M_{w\ell 1}L_{\ell 1} + M_{w\ell 2}L_{\ell 2} \; ; \qquad (3.3.1.3)$$

where M_w is the well mobility defined by equation 2.4.1.5.

Flow-rate allocation by mobility is obtained by discretizing equation 2.4.1.5 to give:

$$Q_{w\ell} = \frac{Q_w \, M_{w\ell} \, L_{\ell}}{\sum_{\ell=\ell_L}^{\ell_U} M_{w\ell} \, L_{\ell}} \; ; \qquad (3.3.1.4)$$

where

ℓ_L is the index of the bottom level of the well screen; and
ℓ_U is the index of the top level of the well screen.

If the screened interval is not continuous from ℓ_L to ℓ_U, the length, $L_{\ell j}$ is set to zero over the appropriate subintervals. For an observation well, the dependent variable data in the aquifer are taken from the cell at location ℓ_U.

Flow-rate allocation by the product of mobility and pressure difference is obtained by discretization of equations 2.4.1.9 using equation 2.4.1.8. The pressure at the well datum is given by (Thomas, 1982, p. 156):

$$p_{wd} = \frac{\sum_{\ell=\hat{\ell}_L}^{\hat{\ell}_U} M_{w\ell} L_\ell (p_m + \rho_w g z_\ell) - Q_w}{\sum_{\ell=\hat{\ell}_L}^{\hat{\ell}_U} M_{w\ell} L_\ell} \; ; \qquad (3.3.1.5)$$

and the flow rate from the well to the aquifer at each layer is given by:

$$Q_{w\ell} = M_{w\ell} L_\ell [p_{wd} + \rho_w g(z_{wd} - z_\ell) - p_m] \; ; \qquad (3.3.1.6)$$

where z_ℓ is the elevation of the well node at level ℓ (m).

Similar expressions were derived by Bennett and others (1982) for constant-density fluids.

For simulations with a well completed in more than one layer, and explicit calculation of the well-datum pressure, a large well index or mobility can cause computational instabilities (Chapplear and Williamson, 1981). A large flow rate will be allocated to a layer with large mobility, and the cell pressure can become nearly equal to the well-bore pressure. This will make the flow-rate allocation small during the next time step, and an oscillation may develop. To avoid a severe time-step limitation, a semi-implicit, well flow-rate allocation can be used. It is available as a calculation option in the HST3D program. Equation 3.3.1.6 becomes:

$$Q_{w\ell}^{n+1} = M_{w\ell} L_\ell (p_{wd}^n + \rho_w^n \, g(z_{wd} - z_\ell) - p_m^n) - M_{w\ell} L_\ell \delta p_m \; . \qquad (3.3.1.7)$$

This gives an implicit coefficient that is included in the matrix element for node m in the finite-difference equations. Note that well-datum pressure still is treated explicitly. This can put a restriction on the time step for stability particularly when the aquifer is being stressed heavily. Also the total flow rate in the well will not be maintained over the time step. Therefore, iterations are necessary.

A fully implicit approach would eliminate the iterations, but would introduce additional coefficients in the flow equation for all the cells that were communicating with the given well. The band-width of the finite-difference flow equations, would be increased, thus making the two matrix-solution techniques, much more difficult to implement. However, Bennett and others (1982) employ the fully implicit approach with a compatible matrix-solution technique.

A compromise algorithm was developed starting from equation 3.3.1.5 expressed as a well constraint to maintain specified well-flow rate:

$$\sum_{\ell=\hat{\ell}_L}^{\hat{\ell}_U} M_{w\ell} L_\ell \delta p_m - \sum_{\ell=\hat{\ell}_L}^{\hat{\ell}_U} M_{w\ell} L_\ell \delta p_{wd} = 0. \tag{3.3.1.8}$$

Equation 3.3.1.8 is written for each well in the region.

The matrix representation of the flow- and well-constraint equations being solved simultaneously is bordered as shown by:

$$\begin{bmatrix} \underline{\underline{A}} & \underline{\underline{W}}_2 \\ \underline{\underline{W}}_1 & \underline{\underline{W}}_3 \end{bmatrix} \begin{bmatrix} \underline{\delta p} \\ \underline{\delta p}_{wd} \end{bmatrix} = \begin{bmatrix} \underline{b}_1 \\ \underline{0} \end{bmatrix}; \tag{3.3.1.9}$$

where

$\underline{\underline{A}}$ is the coefficient matrix from the discretized flow equation;

$\underline{\underline{W}}_1$ is the coefficient matrix linking the pressures in each of the cells which communicate with a well;

$\underline{\underline{W}}_2$ is the coefficient matrix linking the well-datum pressures to the flow equation through the source term;

$\underline{\underline{W}}_3$ is the coefficient matrix (diagonal) for the well-datum pressures in the well constraint equation (eq. 3.3.1.8); and

$\underline{b}_1$ is the vector of known quantities from the discretized flow equation.

The two equations are solved iteratively at each time step for $\underline{\delta p}$ and $\underline{\delta p}_{wd}$. The well-datum pressures are lagged one iteration in the solution of the flow equations. The initial value for $\underline{\delta p}_{wd}$ is taken to be $\underline{0}$. The iterations are terminated when the maximum fractional change in $\underline{\delta p}_{wd}$ is less than 0.001. Usually, only two or three iterations are required for convergence. This algorithm has the advantage that the sparse structure of the matrix $\underline{\underline{A}}$ is preserved, so that the implemented matrix-equation solvers can be employed. At the first time step, equation 3.3.1.4 is used to calculate the flow rates at each layer for each well. Equation 3.3.1.6 is used thereafter.

A reversal of flow between the well and the aquifer at any layer communicating with a well is allowed. However, difficulties arise if there is a reversal of flow within the well bore. An algorithm to compute a realistic density profile in the well bore under flow-reversal conditions has not yet been developed; therefore, the following algorithm is currently used in HST3D to compute heat and solute flow rates in a well bore.

For a production well, heat and solute balance calculations are done from the bottom to the top of the well-screen interval. If injection occurs at a given layer, the density, temperature, and solute concentration injected are based on the current values coming up the well bore from below. Any fluid flowing down the well bore to that layer is neglected. Density, temperature, and solute concentration values based on well-datum conditions are used if there is no upward flow in the well bore below the given injection layer. This algorithm is suitable for producing wells which leak into the aquifer but have net upward flow along the entire well bore. It may be a poor approximation if there are large density, temperature, or solute concentration variations within the well and flow reversals occur in the well bore.

For an injection well, no account is taken of the effect of producing layers on the density, temperature, or solute concentration in the well bore. The conditions at the well-datum level are used for all injection layers. Clearly, this approximation is valid only for injection wells with slight invasion from producing layers and no flow reversals in the well bore. A more realistic algorithm will require a complex, iterative calculation.

In the present version of the HST3D simulator, when a production or injection well becomes inactive, by having its flow rate set to zero, no circulation of fluid from one aquifer-discretization layer to another is computed. Removing this restriction would require the algorithm, described previously, to handle flow reversals in the well bore.

For the case of a single well in the cylindrical-coordinate system, equation 2.4.1.12, for the well bore, is discretized in space and time in the same manner as the system flow equation. Flow-rate allocation by mobility and pressure gradient or specified pressure at the well datum are the options available. The augmented-flow equation 2.4.1.13, is discretized in space and time in the manner that led to equation 3.1.4.1a. At a node along the well screen below the top of the screen, $k<\hat{\ell}_U$, the equation is:

$$
\begin{aligned}
C_{33}\delta p_m + C_{32}\delta T_m + C_{31}\delta w_m = {} & \theta(T_{Fk+\frac{1}{2}} + T_{wFk+\frac{1}{2}})(\delta p_{k+1} - \delta p_k) \\
& -\theta\,(T_{Fk-\frac{1}{2}} + T_{wFk-\frac{1}{2}})(\delta p_k - \delta p_{k-1}) \\
& +(T_{Fk+\frac{1}{2}} + T_{wFk+\frac{1}{2}})\,(p^n_{k+1} - p^n_k) + T_{Fk+\frac{1}{2}}\,\rho^n_{k+\frac{1}{2}}g(z_{k+1} - z_k) \\
& \quad +T_{wFk+\frac{1}{2}}\,\rho^n_{wk+\frac{1}{2}}\,g(z_{k+1} - z_k) \\
& -(T_{Fk-\frac{1}{2}} + T_{wFk-\frac{1}{2}})\,(p^n_k - p^n_{k-1}) - T_{Fk-\frac{1}{2}}\,\rho^n_{k-\frac{1}{2}}g(z_k - z_{k-1}) \\
& \quad -T_{wFk-\frac{1}{2}}\,\rho^n_{wk-\frac{1}{2}}\,g(z_k - z_{k-1}) \\
& + \text{r-direction dispersive terms;} \qquad (3.3.1.10a)
\end{aligned}
$$

where

$$T_{wFk+\frac{1}{2}} = \frac{4\pi r_w^{\ 3}}{v_w f_w(z_{k+1} - z_k)} \ ; \qquad (3.3.1.10b)$$

$$T_{wFk-\frac{1}{2}} = \frac{4\pi r_w^{\ 3}}{v_w f_w(z_k - z_{k-1})} \ ; \qquad (3.3.1.10c)$$

and where

T_{wF} are the conductances for flow at the well bore (m-s).

For the node at the top of the well-screen interval, $k=\ell_U$, the discretized, augmented flow equation (2.4.1.13) becomes:

$$C_{33}\delta p_m + C_{32}\delta T_m + C_{31}\delta w_m = -\theta(T_{F\ell_U-\frac{1}{2}} + T_{wF\ell_U-\frac{1}{2}})(\delta p_{\ell_U} - \delta p_{\ell_U-1})$$

$$-(T_{F\ell_U-\frac{1}{2}} + T_{wF\ell_U-\frac{1}{2}})(p^n_{\ell_U} - p^n_{\ell_U-1}) - T_{F\ell_U-\frac{1}{2}}\rho^n_{\ell_U-\frac{1}{2}}g(z_{\ell_U} - z_{\ell_U-1})$$

$$+\rho^* Q_w - T_{wF\ell_U-\frac{1}{2}}\ \rho^n_{w\ell_U-\frac{1}{2}}\ g(z_{\ell_U} - z_{\ell_U-1})$$

$$+ \text{ r-direction dispersive terms}; \qquad (3.3.1.11)$$

where

Q_w is the specified volumetric flow rate of the well (positive is injection to the aquifer) (m^3/s).

In the case of specified pressure at the well datum, equation 3.3.1.11 is replaced by:

$$p_{\ell_U} = p_{wd} \qquad (3.3.1.12)$$

The well-bore velocity and friction factor are calculated explicitly at the beginning of the time step. Since the friction factor is a weak function of velocity, this causes no instabilities. Evaluating the well-conductance factors explicitly is consistent with the treatment of the aquifer-conductance factors.

3.3.2 **The Well-Riser Model**

The well-riser calculation is done by numerically solving equation 2.4.2.11. These ordinary differential equations are integrated using the midpoint method with rational-function extrapolation, developed by Bulirsch and Stoer (1966) and presented by Gear (1971, p. 96).

The following algorithm is applied to the well-riser calculations:

$$p_r^* = p_{rk} + \frac{\Delta \ell}{2} \; F(p_{rk}, T_{rk}, \ell_k) \; ; \tag{3.3.2.1a}$$

$$T_r^* = T_{rk} + \frac{\Delta \ell}{2} \, G(p_{rk}, T_{rk}, \ell_k) \; ; \tag{3.3.2.1b}$$

$$p_{rk+1} = p_{rk} + \Delta \ell F(p_r^*, T_r^*, \ell_k + \frac{\Delta \ell}{2}) \; ; \tag{3.3.2.1c}$$

$$T_{rk+1} = T_{rk} + \Delta \ell G(p_r^*, T_r^*, \ell_k + \frac{\Delta \ell}{2}) \; ; \tag{3.3.2.1d}$$

where

$$\Delta \ell = \ell_{k+1} - \ell_k . \tag{3.3.2.1e}$$

Boundary conditions are:

$$\text{at} \;\; k = 0 \; ; \;\; p_{rk} = p_r^0 \; ; \; T_{rk} = T_r^0 \; ; \; \rho_r = \rho_r^0 \tag{3.3.2.2a,b,c}$$

The pressure at the well datum used in evaluating equation 3.3.2.2a for production conditions is explicitly calculated at time plane n. Equations 3.3.2.1a-d are integrated over the length of the well riser, L_r, yielding the desired quantities $p_r(L_r)$ and $T_r(L_r)$. The functions *F* and *G* are evaluated by the right-hand-side of equation 2.4.2.11 with the following equations used for calculating density and velocity:

$$\rho_r = \rho_r^0 + \rho_r^0 \beta_p(p_r - p_r^0) - \rho_r^0\beta_T(T_r - T_r^0) \; ; \qquad (3.3.2.3a)$$

$$\rho_{rk}v_{rk} = \rho_r^0 v_r^0 = \frac{\rho_r^0 Q_w}{\pi r_r^2} \; . \qquad (3.3.2.3b)$$

The midpoint method of integration is a second-order method, which means that the error in $p_r(L_r)$ and $T_r(L_r)$ decreases as $(\Delta\ell)^2$. The extrapolation procedure improves the accuracy of the numerical integration by estimating results for $p_r(L_r)$ and $T_r(L_r)$ that would be obtained if the step length, $\Delta\ell$, were reduced to zero. Pressure and temperature at the end of the well riser are expressed by power-series expansions as a function of step length along the riser:

$$p_r(L_r,\Delta\ell) = p_r(L_r) + \sum_{i=1}^{n} d_{pi} \, \Delta\ell^{\,2i} \qquad (3.3.2.4a)$$

$$T_r(L_r,\Delta\ell) = T\,(L_r) + \sum_{i=1}^{n} d_{Ti} \, \Delta\ell^{\,2i} \qquad (3.3.2.4b)$$

where

d_{pi} are the coefficients in the series expansion for pressure (Pa/m); and (3.3.2.4c)

d_{Ti} are the coefficients in the series expansion for temperature (°C/m). (3.3.2.4d)

Equations 3.3.2.4a and 3.3.2.4b can be written in vector form by defining:

$$\underline{Y}(L_r,\Delta\ell) = \begin{bmatrix} p_r(L_r,\Delta\ell) \\ T_r(L_r,\Delta\ell) \end{bmatrix} \tag{3.3.2.5a}$$

$$\underline{Y}_i = \begin{bmatrix} d_{pi} \\ d_{Ti} \end{bmatrix} \tag{3.3.2.5b}$$

so that:

$$\underline{Y}(L_r,\Delta\ell) = \underline{Y}(L_r) + \sum_{i=1}^{m} \underline{Y}_i \,\Delta\ell^{2i} \tag{3.3.2.6}$$

The right-hand-side of equation 3.3.2.6 is approximated by a rational function, $\underline{R}_m$, that is, a quotient of two polynomials. The coefficients of the rational function are determined so that:

$$\underline{R}_m(L_r,\Delta\ell_j) = \underline{Y}(L_r,\Delta\ell_j) \ ; \ j = 0,1\ldots m \tag{3.3.2.7}$$

where

$\Delta\ell_j$ is the spatial-step length for the jth integration from 0 to L (m).

Then, the desired solution, $\underline{Y}(L)$, is related to the approximating rational function by:

$$\underline{Y}(L_r) = R_m(L_r,0) \tag{3.3.2.8}$$

The algorithm is formed by defining $R_m^j\ (\Delta\ell)$ as the rational approximation which agrees with $\underline{Y}(L_r,\Delta\ell)$ at $\Delta\ell = \Delta\ell_j,\ \Delta\ell_{j+1},\ \ldots,\ \Delta\ell_{j+m}$, where $\Delta\ell_j > \Delta\ell_{j+1}$, and defining $R_m^j(0) = R_m^j$. Then the R_m^j give better approximations to $\underline{Y}(L_r)$ as j and(or) m increase. The extrapolation procedure is initiated by integrating equations (3.3.2.1 a-d) for a sequence of step lengths, L/2, L/4, L/6, L/8, ... to obtain values for $R_o^{\,o}$, $R_o^{\,1}$, $R_o^{\,2}$, ... Values of R_m^j for increasing j and m are calculated by the recurrence relation given in Gear, 1971, p. 95. The procedure is terminated when two successive approximations, R_k^{m-k} and R_k^{m-k+1}, are sufficiently close. The tolerance estimate for the fractional error can be set by the user with a default value of 10^{-3}.

Depending on the rate of convergence, the step size may be increased or decreased for successive well-riser calculations. Sixth-order polynomials are the maximum order used for the rational approximation with a maximum of 10 different step sizes.

3.4 BOUNDARY CONDITIONS

All boundary conditions are specified on a cell-by-cell rather than on a zone-by-zone basis. The default-boundary condition is that of no dispersive or advective flux through the boundary faces of the cell. For a cell with three boundary faces, up to three different types of flux-boundary conditions can be applied, each to a different face. For example, a specified flux, an aquifer-influence function, and a leakage-boundary condition could be applied to the faces of a corner cell.

3.4.1 Specified Pressure, Temperature, and Solute-Mass Fraction

Specified-value boundary conditions are incorporated by replacing the flow and transport equations for those nodes, by equations of the form of equation 3.1.1c defining the specified values. These nodes could be removed from the set of simultaneous equations to be solved, by incorporating the known boundary values into the remaining equations; that has not been done in the present version of the HST3D simulator. For boundary conditions that change discontinuously with time, the value at time t^n is taken to be the limit of the value at $t^n - \delta t$, as $\delta t \rightarrow 0$; that is, the jump in the boundary-condition value takes place after the time of change. This means that the effective value of a boundary condition over a time interval when a change occurs is the average value under centered-in-time differencing and the later value under backward-in-time differencing.

It should be noted that an initial hydrostatic-pressure boundary condition over depth will not be maintained under conditions of variable-density flow. Specification of hydrostatic-pressure boundary conditions over depth using a uniform initial density can cause disconcertingly large vertical flows to occur, when realistic fluid compressibility effects are incorporated during the simulation. Even if the compressibility is very small, the boundary pressure values need to be specified to four or five significant digits to avoid vertical flows caused by roundoff error.

Since a specified-value boundary condition removes the equation for the corresponding variable (pressure, temperature, or mass fraction) from the set to be solved, some constraints do exist on what boundary conditions can be specified for a cell that has more than one boundary face. For example, if the pressure is specified, then the ability to specify a fluid flux, an aquifer-influence function, or a leakage boundary condition on the other boundary faces is lost.

3.4.2 Specified-Flux Boundary Conditions

Discretization of the flow equations and transport equations causes the specified-flux boundary conditions to be incorporated as source terms in the finite-difference equations, as described by equation 3.1.1.25a and b. The specified fluxes are input as vector components at each of the respective boundary faces. Thus they are described on a cell-face basis, not by zone boundary. Fluid fluxes are input as volume fluxes; heat fluxes are input as energy fluxes; solute fluxes are input as mass fluxes.

Recall that a boundary cell can have up to three boundary faces, each with an outward normal vector pointing in one of the coordinate directions. The flux-vector components can specify flux only through a face whose normal is parallel to the vector component. Thus, the number of specified-flux vector components must be less than or equal to the number of boundary faces for a given cell. If the normal and the vector component point in opposite directions, flux is added to the boundary cell; if they point in the same direction, flux is withdrawn.

A persistent numerical error can arise in the case where only specified-flux boundary conditions are employed for the entire region, because of the occurance of a zero eigenvalue for the discretized equation (Mitchell, 1969, p. 39-44). Errors generated by discontinuous changes in the boundary conditions with time or by discontinuities between the initial conditions and the boundary conditions will persist. If a specified-value boundary condition or flux-dependent-on-value boundary condition is applied over some part of the boundary, this problem vanishes, because the zero eigenvalue disappears.

A one-dimensional analysis shows that the integral form of derivation used for the specified-flux boundary conditions gives a discretization error of order $\Delta t \Delta x$. Thus, the finite-difference equations are only first-order accurate at the boundary cells in terms of specified flux.

3.4.3 Leakage-Boundary Conditions

Leakage-boundary conditions are transformed into source-sink terms in a similar fashion to specified-flux conditions. They also are incorporated on a cell basis rather than on a zone basis. Equations 2.5.3.1.1a-c and 2.5.3.2.1a-c, when applied on a discrete grid, become for boundary cell, m:

$$Q_{Lm} = \frac{k_{Lm}}{\mu_{Lm} b_{Lm}} \; [(\rho_e \phi_e)_m - (p_m^n + \rho_m^n g z_m) \tag{3.4.3.1a}$$

$$- (\rho_e - \rho_m^n) g (z_e + z_m)/2] \; S_{BLm}$$

$$- \frac{k_{Lm}}{\mu_{Lm} b_{Lm}} \; S_{BLm} \delta p_m \; ;$$

$$Q_{Rm} = \gamma_R Q_{Lm} \; ; \tag{3.4.3.1b}$$

where

Q_{Lm} is the volumetric flow rate at a leakage boundary (m^3/s);

Q_{Rm} is the volumetric flow rate at a river-leakage boundary (m^3/s); and

S_{BLm} is the part of the boundary cell surface that is a leakage boundary (m^2).

The leakage-flow rate, of equation 3.4.3.1a, has an explicit term for the right-hand-side of the discretized system-flow equation, 3.1.4.1a, and an implicit factor for the left-hand-side.

3.4.4 **Aquifer-Influence-Function Boundary Conditions**

3.4.4.1 Pot-Aquifer-Influence Function

The aquifer-influence-function boundary conditions for a pot aquifer are discretized by writing equation 2.5.4.1.1 for each cell face over which the pot-aquifer boundary condition applies. Let there be M_A pot-aquifer boundary condition cells. Then:

$$Q_{Am} = [\alpha_{be} + \varepsilon_e \beta_{pe}) \frac{\delta p_{Bm}}{\delta t} V_{em}, \quad m = 1, M_A \; ; \qquad (3.4.4.1.1)$$

where

$\frac{\delta p_{Bm}}{\delta t}$ is the rate of pressure change at the boundary of the inner region for cell m (Pa/s);

V_{em} is the volume of the outer-aquifer region that influences boundary cell m (m^3); and

Q_{Am} is the volumetric flow rate across the boundary face for cell m between the inner- and outer-aquifer regions; (positive is into the inner region), (m^3/s).

The volume of outer aquifer that influences boundary cell m usually is taken to be the permeability-weighted fractional area of the boundary face:

$$V_{em} = \frac{V_e \sum_{q=1}^{4} k_{mpq} S_{Ampq}}{\sum_{m=1}^{M_A} \sum_{p=1}^{n_p} \sum_{q=1}^{4} k_{mpq} S_{Ampq}} \; ; \qquad (3.4.4.1.2)$$

where

S_{Ampq} is the area of the aquifer-influence function boundary face for cell m, face p, subdomain q (m^2); and

k_{mpq} is the permeability for cell m, face p, subdomain q (m^2).

This aquifer-influence-function flow rate gives only an implicit coefficient for the left-hand-side of the flow equation, 3.1.4.1a.

3.4.4.2 Transient-Flow Aquifer-Influence Function

The transient-flow, aquifer-influence function is discretized by writing equations 2.5.4.2.6a-b for each cell at which this boundary condition applies. Thus, a different pressure history may occur at each boundary node. The aquifer-influence-function flow rate must be suitably apportioned among the boundary cell faces. The method used for HST3D is to make the fraction of the total flow rate that is apportioned to a given boundary cell the same as the ratio of that boundary-cell facial area to the total boundary facial area between the inner-and outer-aquifer regions. For cases where the inner-aquifer region is strongly heterogeneous, apportionment by the product of hydraulic conductivity and facial area, using equation 3.4.4.1.2, would be more realistic. This would require modification to the program code.

The derivation of the flow rate given by equation 2.5.4.2.6a also was based upon a uniform pressure plus gravitational potential over the approximating cylindrical interface. A finite-difference flow simulation in the inner-aquifer region will yield a nonuniform distribution of pressures at the boundary nodes, except in special cases. In the numerical implementation of this aquifer-influence-function calculation, the pressure at each interfacial boundary node is taken to be the value computed by the discretized simulation calculation. This introduces an additional approximation, because any lateral or vertical flow in the outer-aquifer region, induced by the nonuniform pressure plus gravitational potential distribution over the interfacial boundary, is neglected.

Another approximation used is that the boundary between the inner- and outer-aquifer regions is represented by a cylindrical interface (fig. 2.5a); whereas, the actual boundary is a set of rectangular faces for the finite-difference discretization in cartesian coordinates of a three-dimensional, inner-aquifer region. In contrast, a two-dimensional, cylindrical gridding for the inner-aquifer region would have an exact cylindrical boundary. For a cartesian-coordinate system with the x-y axes horizontal, the equivalent radius, r_I, for the approximate interfacial boundary is calculated, so that the rectangular area and the equivalent circular area are the same; that is:

$$\pi r_I^2 = (x_{Nx}-x_1)(y_{Ny}-y_1) \ . \qquad (3.4.4.2.1)$$

Equation 3.4.4.2.1 will be a poor approximation for long, slender rectangular areas in the x-y plane. For boundaries between the inner- and outer-aquifer regions that do not completely surround the inner aquifer laterally, the apportionment factor γ_{Am}, must contain an angle-of-influence factor, f_θ. Then:

$$\gamma_{Am} = f_\theta S_{Ampq} \ ; \qquad (3.4.4.2.2)$$

where

S_{Ampq} is the area of the aquifer-influence function boundary face for cell m, face p, subdomain q (m^2); and

f_θ is the angle-of-influence factor for the simulation region (-).

This factor is the fraction of a full circle that the boundary between the inner- and outer-aquifer regions subtends. For example, an outer-aquifer region that surrounds half of the inner-aquifer simulation region (fig. 2.5b) would have an angle-of-influence factor of ½.

The Carter-Tracy approximation is used to avoid successive recomputation of the convolution integral that gives the flow rate at the transient aquifer-influence-function boundary. After discretization of time, the expression for the flow rate across the boundary between the inner-aquifer region and the outer-aquifer region is [Kipp (1986)]:

$$Q_{Am}^{n+1} = \gamma_{Am} \frac{2\pi k_e b_e}{\mu_e} \left[\frac{p_m^n - p_m^o - \left.\frac{dP_U'}{dt}\right|^{n+1} \frac{W_m^n}{2\pi r_I^2(\alpha_{be} + \varepsilon_e \beta_{pe}) b_e} + \delta p_m}{P_U'^{\,n+1} - \left.\frac{dP_U'}{dt}\right|^{n+1} t'^{\,n}} \right]. \tag{3.4.4.2.3}$$

The Carter-Tracy approximation is based on representing the continuous flow rate of equation 2.5.4.2.6a by a discontinuous sequence of constant flow rates so that the cumulative net inflow from the start of the simulation to the given time is the same for the convolution integral and the current constant-flow rate. This approximation is exact for constant-flow rates. Therefore, slowly varying flow rates are more accurately handled by the Carter-Tracy approximation than rapidly varying ones. The major disadvantage of the Carter-Tracy approximation is the inaccuracy of the computation of the discretized flow rate in the case where boundary-flow rates vary with time. Effects on the computed-flow rate appear as a significant time lag and smoothing of transients. Errors can be serious for boundary-flow rates whose variations are large relative to the average value.

Note that the computer storage requirements for this calculation are only the cumulative flow, W_m^n, the pressure at the end of the nth time step, p_m^n, and the flow-rate allocation factor, γ_{Am}, for each node on the boundary between the inner- and outer-aquifer regions. The values of p_m^n are from the flow simulation, so the additional storage amounts to only two times the number of AIF boundary nodes.

Equation 3.4.4.2.3 is of the form:

$$Q_{Am}^{n+1} = a_1(t') + a_2(t')\ \delta p_m \ ; \tag{3.4.4.2.4}$$

where a_1 is the known flow rate term added to the right-hand side of the discretized flow equation for node m; and a_2 is the implicit term added to the left-hand side factor.

The values of $P_U'(t')$ and $\frac{dP_U'}{dt'}$ are obtained usually from tables by interpolation. Values of the dimensionless pressure function in response to a unit-withdrawal flow rate at the AIF boundary have been tabulated by Van Everdingen and Hurst (1949) for the infinite cylindrical region and for regions with a finite outer-boundary radius.

However, it is much more convenient to use the approximate analytical representation developed by Fanchi (1985). He employed linear regression analysis to obtain the following equation that approximates the Van Everdingen and Hurst (1949) aquifer influence functions with very small errors. In the notation of this report;

$$P_U'(t') = b_0 + b_1\ t' + b_2\ \ell n(t') + b_3\ \ell n^2(t') \tag{3.4.4.2.5}$$

Table 3.2 adapted from Fanchi (1985) contains values of the b_i coefficients for several cases. The first line gives the coefficients for the analytical approximation to equation 2.5.4.2.5b for the case of an infinite outer-aquifer region. The subsequent lines are for various values of R for the case of a finite outer-aquifer region with no flow at the exterior boundary, where R is the ratio of exterior to interior radius for the outer-aquifer region.

As with the previous boundary conditions, the heat- and solute-advective transport rates across the AIF boundary are calculated by multiplying the flow rate by the appropriate density, enthalpy, and mass-fraction values, depending on the direction of flow.

Table 3.2--*Coefficients for the analytical approximations to the Van Everdingen and Hurst aquifer-influence functions*

R	b_0	b_1	b_2	b_3
∞	-0.82092	3.68×10^{-4}	-0.28908	-0.02882
1.5	-0.10371	-1.66657	0.04579	0.01023
2.0	-0.30210	-0.68178	0.01599	0.01356
3.0	-0.51243	-0.29317	-0.01534	0.06732
4.0	-0.63656	-0.16101	-0.15812	0.09104
5.0	-0.65106	-0.10414	-0.30953	0.11258
6.0	-0.63367	-0.06940	-0.41750	0.11137
8.0	-0.40132	-0.04104	-0.69592	0.14350
10.0	-0.14386	-0.02649	-0.89646	0.15502

3.4.5 Heat-Conduction Boundary Condition

The heat-conduction boundary condition is a flux-type boundary condition, with the heat flux dependent on the thermal parameters and the thermal history at the boundary, and in the conducting medium, which lies outside the simulation region. Equation 2.5.5.7 gives the heat flux that is applied on a cell basis to the source term in the heat-transport equation. Finite-difference approximations to equations 2.5.5.3a-c and 2.5.5.4a-c are solved at each time step for each heat-conduction, boundary-condition cell. Central-differencing in time is used giving a tridiagonal-matrix equation to be solved numerically. The user specifies the spatial mesh extending out from the boundary into the conducting medium. Up to 10 nodes are allowed with variable spacing. The first node must be on the boundary. The Thomas algorithm (Varga, 1962, p. 195) is used to obtain the numerical solutions for T_e and T_U. Then the heat flux for cell m is given by:

$$Q_{HCm} = - K_{em} \left[\frac{T_e(z_{n2}) - T_e(z_{n1})}{z_{n2} - z_{n1}} + \frac{T_U(z_{n2}) - T_U(z_{n1})}{z_{n2} - z_{n1}} \right] \delta T_{Bm} \quad S_{BHCm}; \quad (3.4.5.1)$$

where

z_{n1} and z_{n2} are the first two nodes moving in the outward normal direction from boundary cell m (m).

Node z_{n1} is coincident with the boundary node of the simulation region, so the boundary temperature is:

$$T_{Bm} = T_m \ . \quad (3.4.5.2)$$

Equation 3.4.5.1 is of the form:

$$Q_{HCm} = a_1(t) + a_{2_m}(t)\ \delta T \ ; \quad (3.4.5.3)$$

where the $a_1(t)$ term goes into the source term for the cell heat-transport equation, and the $a_2(t)$ term goes into the thermal-coefficient matrix for cell m. An initial temperature profile can be specified for the heat-conduction region. However, the same profile is used for all cells with a heat-conduction boundary condition.

3.4.6 Unconfined-Aquifer, Free-Surface Boundary Condition

The unconfined-aquifer, free-surface boundary condition is implemented by modifying the pressure-coefficient terms in the discretized equations for flow and solute-transport, and adjusting the fluid volume or saturated thickness of the uppermost layer of cells in the simulation region. A free surface is not allowed when the heat-transport equation is being solved, because no satisfactory method has been developed to handle the conductive-heat flux through the free-surface boundary that moves with time through the unsaturated porous medium.

The location of the free surface within an upper-boundary cell is established by linearly extrapolating the nodal pressure to the elevation of zero (atmospheric) pressure, using the fluid density in that cell. The fraction of the cell thickness that is saturated is given by:

$$f_{FS} = 1 + \frac{p_m}{\rho_m g \frac{1}{2}(z_{Nz} - z_{Nz-1})} \; ; \qquad (3.4.6.1)$$

where

f_{FS} is the fraction of the cell thickness that is saturated (-); and

p_m is the pressure at node m (Pa);

z_{Nz} is the elevation of the upper boundary (m); and

z_{Nz-1} is the elevation of the next layer of nodes down from the upper boundary (m).

This equation is valid for coordinate systems with the z-axis pointing vertically upward. Remember that the pressure value is relative to atmospheric pressure, so that atmospheric pressure does not appear explicitly in equation 3.4.6.1. This fraction is allowed to range from zero to two; that is, the free surface is allowed to rise above the upper boundary of the simulation region to a distance that is equal to the upper-layer half-cell thickness. This rise is effectively the same as using full cells in the vertical direction for the upper layer when a free-surface boundary condition is being used. The extra cell height allows for a greater variation of the free surface than is possible with the normal cell height in the upper layer.

When designing the grid for a free-surface boundary problem, the uppermost layer of cells must be made thick enough to accommodate the maximum variations in the free-surface location. With the present algorithm, the free surface may not drop below the lower boundary of the uppermost layer of cells. No conversion to confined flow conditions is made if it does rise above the extended upper boundary. These restrictions can be burdensome if a large drawdown cone is created by a well pumping an unconfined aquifer, because the uppermost layer of cells may need to be so thick that vertical gradients are represented poorly. The present version of the HST3D program is suitable for simulation of only modest drawdowns relative to aquifer thickness for unconfined conditions.

If desired by the user, a message may be printed when the free surface rises above the extended upper boundary of the region or falls below the bottom of the uppermost layer of cells, or cells in a lower layer become partially saturated.

To obtain the appropriate coefficients for the flow equation (3.1.4.1a) at the free-surface boundary cells, we evaluate the terms of equation 3.1.1.8, this time, including the saturation fraction of equation 3.4.6.1, using equation 3.1.3.5, and assuming that the porosity is constant, the fluid compressibility is zero and isothermal conditions exist, to obtain:

$$C_{33} = [\sum_{s=1}^{8} \varepsilon_s V_s]/[\tfrac{1}{8}g(z_{Nz} - z_{Nz-1})\delta t] \; ; \qquad (3.4.6.2a)$$

and the coefficient C_{31} remains unchanged from equation 3.1.4.1d.

Using the same procedure as for equations 3.4.6.2a and 3.4.6.2b, the corresponding terms for the solute-transport equation (3.1.4.4a) are:

$$C_{11} = \sum_{s=1}^{8} (\varepsilon_s + \rho_b K_d)_s \; V_s[\rho_m^{n+1} \; f_{Fs}^{n+1} + \; w_m^n \; \rho_o \beta_w]/\delta t \; ; \qquad (3.4.6.3a)$$

and

$$C_{13} = [\sum_{s=1}^{8} (\varepsilon_s+\rho_b K_d)_s V_s \frac{w_m^n}{\frac{1}{2}g(z_{Nz} - z_{Nz-1})}]/\delta t \ . \qquad (3.4.6.3b)$$

Additional terms arise from the source-sink term in the solute equation that are functions of solute-mass fraction. They form part of the C_{11} and C_{13} coefficients. It has also been assumed that all of the solute in a cell is either in the fluid phase or is sorbed on the saturated part of the porous medium. No account is taken of solute that might sorb onto the porous medium and be left behind, when the free surface falls. This simplification is consistent with this approximate treatment of a free-surface boundary condition. The other terms in the flow and transport equations have the saturated fraction parameter included, as necessary, for the cell facial-area terms involving the x and y directions. No additional contributions to the C_{ij} terms occur, because the dispersive and advective coefficients are evaluated at time t^n only.

The case of a free-surface boundary with accretion of fluid by infiltration is also handled in an approximate fashion. The fluid flux is specified at the upper boundary of the cell, and the associated temperature and mass fraction determine the amount of heat and solute that enter through the free surface.

3.5 INITIAL CONDITIONS

The numerical implementation of the initial conditions is straightforward. Values of pressure, temperature and mass fraction are set to the initial value distributions for each node in the simulation region given in the form of equation 3.1.1b, that is:

$$\text{at } t=0, \quad u_{ijk} = u^0_{ijk}. \qquad (3.5.1)$$

The specified distributions can vary on a node-by-node basis or be zones of constant conditions. Available options for the initial pressure distribution include hydrostatic equilibrium, a water-table surface, or a pressure field specified node by node. The water-table surface is specified for the upper layer of cells only. These initial distributions are based on the initial temperature and solute-mass fraction distributions.

The hydrostatic-equilibrium pressure distribution takes the fluid compressibility into account. The calculation proceeds from the bottom of the region upward or from the top of the region downward depending on the elevation of the specified initial pressure.

The water-table elevation surface is specified for the upper layer of nodes only. Hydrostatic equilibrium is assumed to compute the pressure distribution elsewhere in the simulation region. When specifying the initial pressure field on a node-by-node basis, it is permissible to include nodes that are outside the simulation region. This is for ease of data input by rectangular zones or by ascending node number.

3.6 EQUATION SOLUTION

Equations 3.1.4.1a, 3.1.4.2a and 3.1.4.4a can be written in matrix form as:

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \delta w_m \\ \delta T_m \\ \delta p_m \end{bmatrix} = \begin{bmatrix} \underline{E}_{11} & \underline{0} & \underline{E}_{13} \\ \underline{0} & \underline{E}_{23} & \underline{E}_{23} \\ \underline{0} & \underline{0} & \underline{E}_{33} \end{bmatrix} \begin{bmatrix} \underline{\delta w}_m \\ \underline{\delta T}_m \\ \underline{\delta p}_m \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{3.6.1a}$$

$$= \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix} \quad ; \tag{3.6.1b}$$

where

$\underline{E}_{ij}$ are the coefficient vectors in the discretized equations;

δp_m is the change in pressure for node m (Pa);

$\underline{\delta p}_m$ is the change-in-pressure vector for node m (Pa);

δT_m is the change in temperature for node m (°C);

$\underline{\delta T}_m$ is the change-in-temperature vector for node m (°C);

δw_m is the change in mass fraction for node m (-);

$\underline{\delta w}_m$ is the change-in-mass-fraction vector for node m (-); and

F_i are the known terms at time n in the discretized system equations.

The vectors of the changes in the dependent variables contain the values for each node connected to node m plus node m itself. The $\underline{E}$ vectors in equation 3.6.1a have seven components each that correspond to node m and its six neighbors in the three coordinate directions. The known terms F_i are those that do not contain δw for i=1, δT for i=2, or δp for i=3. The terms $\underline{E}_i$ and F_i can be functions of pressure, temperature, and mass fraction at time n which gives explicit linking of the three equations. Implicit linking is through the $\underline{C}$ matrix on the left-hand side. The equations are written in reverse order to what has been done previously, with 1 referring to the solute-transport equation, 2 referring to the heat-transport equation, and 3 referring to the flow equation. Equation 3.6.1b is written for each node in the simulation region, giving a set of 3M simultaneous equations to be solved for the unknown vectors, δp, δT and δw, where M is the total number of nodes in the region.

3.6.1 Modification of the Flow and Transport Equations

To avoid storing a 3M × 3M matrix and vectors of length 3M, a sequential solution scheme has been developed by Coats and others (1974) and was used by INTERCOMP Resource and Development and Engineering, Inc. (1976). This algorithm consists of solving a modified flow equation then a modified heat-transport equation, then the solute-transport equation in turn for each time step. The modified equations are obtained by a partial Gauss reduction of equation 3.6.1b transforming the capacitance matrix, $\underline{C}$, into upper-triangular form. Thus we have:

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ 0 & C_{22}' & C_{23}' \\ 0 & 0 & C_{33}' \end{bmatrix} \begin{bmatrix} \delta w_m \\ \delta T_m \\ \delta p_m \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ C_{24} & 1 & 0 \\ C_{34} & C_{35} & 1 \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix} ; \quad (3.6.1.1a)$$

where

$$C_{22}' = C_{22} - \frac{C_{12}C_{21}}{C_{11}} ; \quad (3.6.1.1b)$$

$$C_{23}' = C_{23} - \frac{C_{13}C_{21}}{C_{11}} ; \quad (3.6.1.1c)$$

$$C_{24} = - \frac{C_{21}}{C_{11}} ; \quad (3.6.1.1d)$$

$$C_{33}' = C_{33} - \frac{C_{13}C_{31}}{C_{11}} - \frac{(C_{23}C_{11} - C_{13}C_{21})(C_{32}C_{11}-C_{12}C_{31})}{C_{11}(C_{22}C_{11} - C_{12}C_{21})} ; \quad (3.6.1.1e)$$

$$C_{34} = - \frac{C_{31}}{C_{11}} + \frac{C_{21}(C_{32}C_{11} - C_{12}C_{31})}{C_{11}(C_{22}C_{11} - C_{12}C_{21})} ; \quad (3.6.1.1f)$$

$$C_{35} = - \frac{C_{32}C_{11} - C_{12}C_{31}}{C_{22}C_{11} - C_{12}C_{21}} \quad (3.6.1.1g)$$

3.6.2 Sequential Solution

The finite-difference approximations (equation 3.6.1.1a) to the ground-water flow, heat transport, and solute transport equations are solved sequentially. First, the flow equations are solved for the pressures. Then the pressures are used to update the coefficients in the heat equations and back substitute for the δp terms on the left-hand side. Second, the heat equations are solved for the temperatures. The temperatures and pressures are used in the solute equation to update the coefficients and to back substitute for the δp and δT terms on the left-hand side. Finally the solute equations are solved for the mass fractions, which completes an iteration. Thus only M equations at one time are being solved with a coefficient matrix that is $M \times M$ in size. M is the total number of active nodes in the simulation region. Actually the storage requirement is reduced by taking advantage of the matrix sparsity as will be described later. The solution cycle is repeated until convergence is achieved, that is, when the fractional change in fluid density in each cell is less than a tolerance value. The default value is 0.001 fractional change in density. Since changes in both temperature and mass fraction cause changes in density, a secondary tolerance criterion is set to determine whether the heat equation or the solute equation or both equations must be solved again. The secondary tolerance is 0.0005 fractional change in density. If the maximum density change due to temperature changes after solution of the heat equation or due to mass-fraction changes after solution of the solute equation is less than the secondary tolerance, then that equation is excluded from the iterative cycle for the remainder of the calculations for that time step. However, a final solution of the excluded equation is performed after convergence of the iterative cycle for the two remaining equations. In practice, convergence is usually achieved within three iterations for a given time step. Lack of convergence may indicate that the time step is too large.

Equation 3.6.1.1a is transformed into:

$$\underline{\underline{A}}\ \underline{\delta u} = \underline{b} \ ; \qquad (3.6.2.1a)$$

where

$$\underline{\delta u} = \begin{bmatrix} \underline{\delta w} \\ \underline{\delta T} \\ \underline{\delta p} \end{bmatrix}; \qquad (3.6.2.1b)$$

by shifting all the terms on the right-hand side that contain δp, δT or δw to the left-hand side. Equation 3.6.2.1a is a linear-matrix equation for the region that can be solved using techniques to be described in section 3.7.

3.7 MATRIX SOLVERS

The linearized flow and transport finite-difference equations are solved in turn by one of the solution algorithms for linear, sparse-matrix equations. For the present version of the simulator, two such algorithms are available. One is a direct equation solver that uses Gauss elimination, after reordering the equations for a savings in computation time and computer-storage requirements. Alternating diagonal planes are used for the reordering. This method is referred to as the D4 solution technique; it was developed by Price and Coats (1974).

The other sparse-matrix equation solver uses the two-line, successive-overrelaxation method. This is one of a class of block-iterative methods described by Varga (1962, p. 199-200). In this solver, two lines of nodes along a selected coordinate direction are solved together by direct elimination. One iteration sweep consists of solving for the nodal values for each pair of lines, plus the odd left-over line, if necessary. Overrelaxation is used to speed convergence, and the optimum overrelaxation factor is estimated, using the eigenvalue estimation technique of Varga (1962, p. 284-288) at the beginning, and then, every n time steps, as specified by the user. In the process of estimating the optimum-overrelaxation factor, the solution is tested in all three coordinate directions, and the direction with the best-conditioned iterative matrix is selected. It may be different for each of the three equations.

The form of the equations to be solved is the same as equation 3.6.2.1a, but now the matrix $\underline{\underline{A}}$ is a sub-matrix of the original one, containing the coefficients of only the flow, or heat transport, or solute transport equations. Matrix $\underline{\underline{A}}$ has the banded structure under the original nodal numbering scheme, where the index i is incremented first; the index j is incremented second; and the index k is incremented third. Figure 3.7 shows the structure of the matrix $\underline{\underline{A}}$, with the bandwidths given by N_x N_y and N_z where N_i is the number of nodes in the ith direction. The rectangular prism of nodes also is shown in figure 3.7, which encompases the entire simulation region.

3.7.1 The Alternating Diagonal, D4, Direct-Equation Solver

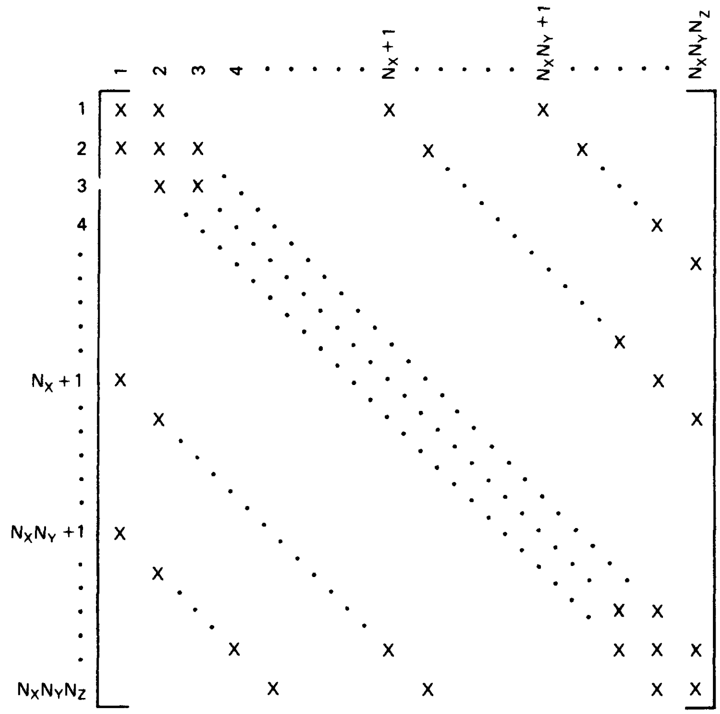
The alternating diagonal reordering scheme was developed by Price and Coats (1974). In three dimensions, the equations are grouped by diagonal planes of nodes. A plane is defined by a fixed sum of the nodal subscripts, designated by the index, m, so:

$$i + j + k = m; \; m = 3,4\ldots, M \; ; \tag{3.7.1.1a}$$

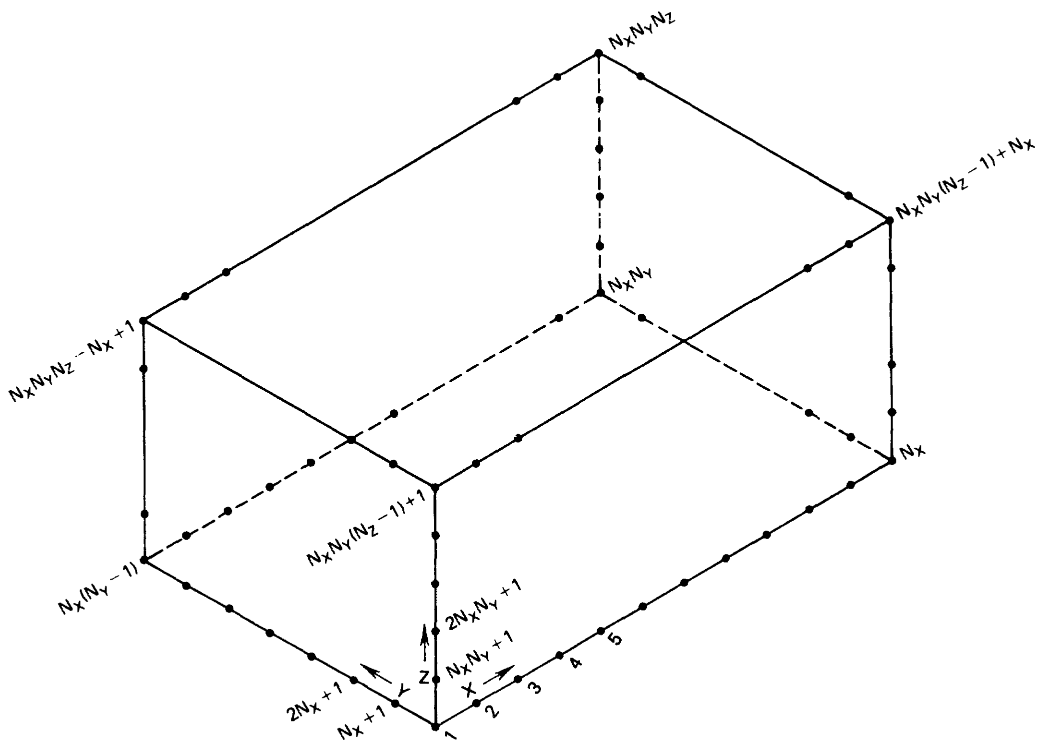
where $$M = N_x + N_y + N_z \; . \tag{3.7.1.1b}$$

If M is even, then the order of plane-index selection for reordering of the node numbers should be m = 3,5,7,...M-1,4,6,8...M. If M is odd, then the order should be m = 3,5,7,...M,4,6,8,...M-1. For each plane index, m, the points should be numbered in order of decreasing k, decreasing j, and increasing i, assuming $N_x > N_y > N_z$. Any excluded cells are skipped during the node renumbering. The matrix $\underline{\underline{A}}$ under D4 ordering takes the form shown in figure 3.8. This matrix can be partitioned as shown, so

$$\left[\begin{array}{c|c} \underline{\underline{A}}_1 & \underline{\underline{A}}_2 \\ \hline \underline{\underline{A}}_3 & \underline{\underline{A}}_4 \end{array}\right] \left[\begin{array}{c} \underline{\delta u}_1 \\ \hline \underline{\delta u}_2 \end{array}\right] = \left[\begin{array}{c} \underline{b}_1 \\ \hline \underline{b}_2 \end{array}\right] ; \tag{3.7.1.2}$$



MATRIX STRUCTURE



RECTANGULAR PRISM OF NODES

Figure 3.7.--Sketch of the matrix structure and the rectangular prism of nodes of the flow or heat-transport, or solute-transport equation in finite-difference form.

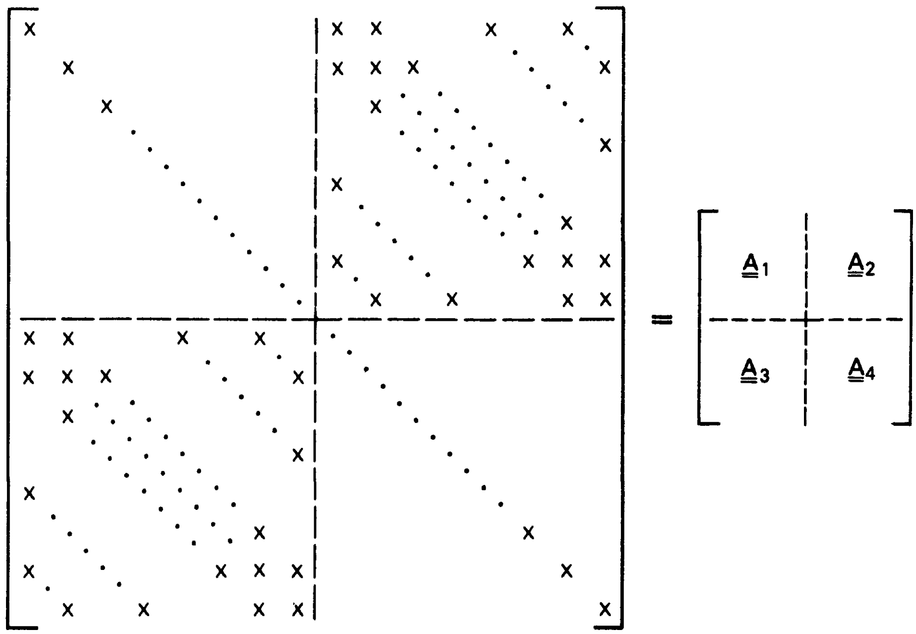


Figure 3.8.--Sketch of the matrix structure with the D4 alternating-diagonal-plane, node-renumbering scheme.

where

$\underline{\underline{A}}_1$ and $\underline{\underline{A}}_4$ are diagonal and $\underline{\underline{A}}_2$ and $\underline{\underline{A}}_3$ are sparse matrices.

Forward elimination gives:

$$\left[\begin{array}{c|c} \underline{\underline{A}}_1 & \underline{\underline{A}}_2 \\ \hline \underline{\underline{0}} & \underline{\underline{A}}'_4 \end{array}\right] \left[\begin{array}{c} \underline{\delta u}_1 \\ \hline \underline{\delta u}_2 \end{array}\right] = \left[\begin{array}{c} \underline{b}_1 \\ \hline \underline{b}'_2 \end{array}\right] \quad ; \qquad (3.7.1.3)$$

where

$\underline{\underline{A}}_4'$ is a band matrix with a maximum bandwidth which is the same as for the original matrix $\underline{\underline{A}}$.

The solution for $\underline{\delta u_2}$ is obtained by standard Gaussian elimination. Back substitution is used to compute $\underline{\delta u_1}$ by:

$$\underline{\delta u_1} = \underline{\underline{A}}_1^{-1}\, \underline{b}_1 - \underline{\underline{A}}_1^{-1}\underline{\underline{A}}_2\underline{\delta u_2}. \tag{3.7.1.4}$$

The work required with the D4 reordering is from 17 to 50 percent, and the storage is from 33 to 50 percent of that using standard ordering. When the D4 ordering is selected with direct Gaussian elimination, the renumbering is entirely transparent to the user. The storage requirement for the $\underline{\underline{A}}_4$ matrix is minimized by employing variable bandwidth storage (Jennings, 1977, p. 97). The matrix is stored by rows, with the length of each row being sufficient to accommodate the fill-in that occurs during elimination. Two pointer arrays are necessary: one that contains the indices of the diagonal elements, and the other that gives the bandwidth to the right of the diagonal. The next row begins in the location just after that specified by the diagonal-element location, plus the right-side bandwidth.

3.7.2 The Two-Line, Successive-Overrelaxation Technique

This iterative matrix-equation solution technique, abbreviated L2SOR, is a block successive-overrelaxation algorithm as described by Varga (1962, sec. 6.4) or Jennings (1977, p. 202). Equation 3.6.2.1a for only the flow or heat transport or solute transport equation may be written with a partitioned $\underline{\underline{A}}$ matrix (fig. 3.9) as:

$$
\begin{bmatrix}
\underline{\underline{D}}_1 & \underline{\underline{U}}_1 & & & 0 \\
\underline{\underline{L}}_2 & \underline{\underline{D}}_2 & \underline{\underline{U}}_2 & & \\
& \underline{\underline{L}}_3 & \ddots & \ddots & \\
& & \ddots & \ddots & \underline{\underline{U}}_{L-1} \\
0 & & & \underline{\underline{L}}_L & \underline{\underline{D}}_L
\end{bmatrix}
\begin{bmatrix}
\delta\underline{u}_1 \\ \vdots \\ \delta\underline{u}_L
\end{bmatrix}
=
\begin{bmatrix}
\underline{b}_1 \\ \vdots \\ \underline{b}_L
\end{bmatrix}
\quad ; \tag{3.7.2.1}
$$

where the sparse submatrices $\underline{\underline{D}}_k$ are penta-diagonal and the submatrices $\underline{\underline{L}}_k$ and $\underline{\underline{U}}_k$ are diagonal (fig. 3.9). There are L pairs of mesh lines with the last set containing only the odd remaining line, if one exists. These matrices are of size $2N_x$ X $2N_x$, where the nodes have been numbered in the normal way, and the two lines have been selected to be in the x-direction. Some rearrangement would be necessary, if the y or z-directions were selected, but the basic structure would be the same. Then the iterative technique is expressed by the following equations:

$$
[\underline{\underline{D}}_\ell + \omega\underline{\underline{L}}_\ell]\delta\underline{u}_\ell^{\ \nu+1} = [-\omega\underline{\underline{U}}_\ell + (1-\omega)\underline{\underline{D}}_\ell]\delta\underline{u}_\ell^{\ \nu} + \omega\underline{b}_\ell; \ \ell = 1, 2 \ldots L \ ; \tag{3.7.2.2}
$$

where

ν is the iteration counter; and

ω is the overrelaxation factor.

Direct elimination is used to solve equation 3.7.2.2, after a renumbering is performed transverse to the direction of the two-lines. This renumbering compresses the bandwidth so no fill-in occurs. The iterations could be terminated when a vector-difference norm is less than a specified tolerance (Jennings, 1977, p. 184). That is, when:

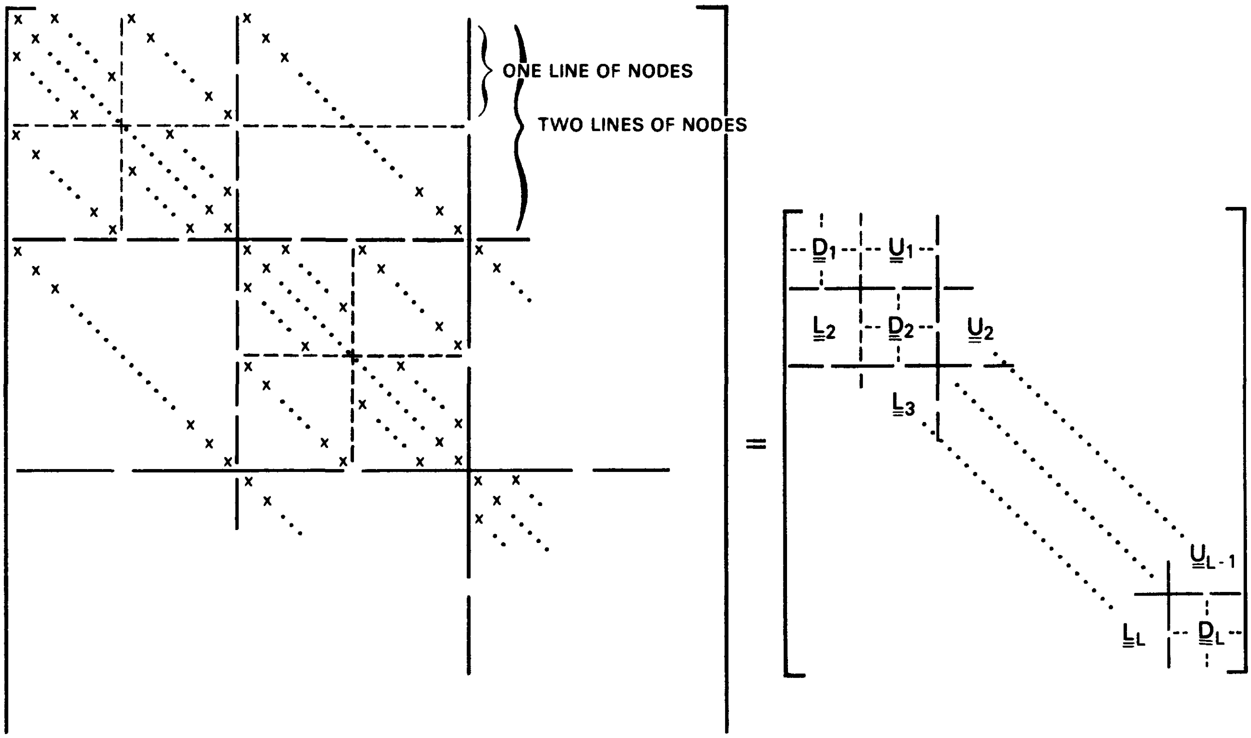


Figure 3.9--Sketch of the matrix structure with the two-line, successive-overrelaxation, node-renumbering scheme.

$$\max_m \frac{\left|\delta u_m^{\nu+1} - \delta u_m^{\nu}\right|}{\left|\delta u_m^{\nu+1}\right|} \leq \varepsilon_{SOR} \; ; \; m = 1, 2 \ldots M \; ; \qquad (3.7.2.3)$$

where

ε_{SOR} is the specified tolerance; and

m is the node number.

An alternate termination criterion is from Remson and others (1971, p. 185). Properties of the iteration matrix are taken into account by terminating the iteration when:

$$\frac{R(L_\omega)}{1-R(L_\omega)} \quad \frac{\left|\delta u_m^{\nu+1} - \delta u_m^{\nu}\right|}{\left|\delta u_m^{\nu+1}\right|} \leq \varepsilon_{SOR} ; \; m = 1, 2 \ldots M \; ; \qquad (3.7.2.4a)$$

where

$R(L_\omega)$ is the spectral radius of the SOR iteration matrix.

Now from Varga (1962, p. 111):

$$R(L_\omega) = \omega_{opt} - 1, \tag{3.7.2.4b}$$

where

ω_{opt} is the optimum overrelaxation factor.

This alternate criterion is used in the HST3D simulator. The tolerance, ε_{SOR}, is set to 1×10^{-5} by default, but it may be changed by the user.

The optimum overrelaxation factor is determined from estimates on the eigenvalues of the associated block Gauss-Seidel matrix. This is a combination of the power method and Perron-Frobenius theory of nonnegative matrices, presented by Varga (1962, p. 283-288). The algorithm is as follows: A two-line, Gauss-Seidel solution of the matrix equation:

$$\underline{\underline{A}}\ \underline{\delta u} = \underline{0} ; \tag{3.7.2.5a}$$

is performed with a starting vector of unit components, that is,

$$\underline{\delta u}_\ell^{\nu+1} = [\underline{\underline{D}}_\ell + \underline{\underline{L}}_\ell]^{-1} [-\underline{\underline{U}}_\ell]\underline{\delta u}_\ell^{\nu} ; \ \ell = 1, 2 \ldots L \tag{3.7.2.5b}$$

$$\text{with } \ \underline{\delta u}_\ell^{\circ} = [\underline{1}] . \tag{3.7.2.5c}$$

Then the minimum and maximum estimates of the dominant eigenvalue for $\underline{\underline{A}}$ are:

$$\lambda_{min}^{\nu} = \min_i \frac{\delta u_i^{\nu+1}}{\delta u_i^{\nu}} ; \tag{3.7.2.6a}$$

and $$\lambda_{max}^{\nu} = \max_i \frac{\delta u_i^{\nu+1}}{\delta u_i^{\nu}} ; \qquad (3.7.2.6b)$$

which satisfy

$$\lambda_{min}^{\nu} < \lambda_{min}^{\nu+1} < R(L_1) < \lambda_{max}^{\nu+1} < \lambda_{max}^{\nu}; \qquad (3.7.2.6c)$$

where

$R(L_1)$ is the spectral radius of the Gauss-Seidel iteration matrix, $[\underline{\underline{D}}+\underline{\underline{L}}]^{-1}[-\underline{\underline{U}}]$. Now, since:

$$\omega_{opt} = \frac{2}{1 + (1-R(L_1))^{\frac{1}{2}}} ; \qquad (3.7.2.7)$$

let

$$\omega_{min}^{\nu} = \frac{2}{1 + (1 - \lambda_{min}^{\nu})^{\frac{1}{2}}} ; \qquad (3.7.2.8a)$$

and

$$\omega_{max}^{\nu} = \frac{2}{1 + (1 - \lambda_{max}^{\nu})^{\frac{1}{2}}} . \qquad (3.7.2.8b)$$

Then it can be shown that for the matrices, $\underline{\underline{A}}$, that arise from the finite-difference equations, which are diagonally dominant and irreducible (Varga, 1962, p. 177):

$$\omega_{min}^{\nu} < \omega_{min}^{\nu+1} < \omega_{opt} < \omega_{max}^{\nu+1} < \omega_{max}^{\nu} ; \qquad (3.7.2.9)$$

where

ω_{opt} is the optimum overrelaxation factor.

For $\underline{\underline{A}}$ nonsymmetric, the mesh spacing must be sufficiently small to ensure diagonal dominance. One requirement for irreducibility is that equations for nodes with specified value boundary conditions are eliminated from the set that forms $\underline{\underline{A}}$. This elimination is not done in the present version of HST3D, and this sometimes causes difficulties in calculating ω_{opt}, because a reducible matrix can have a maximum eigenvalue that is zero. The eigenvalues of the Gauss-Seidel iteration matrix must be real and nonnegative for the optimum-overrelaxation-factor calculation formula, equation 3.7.2.7, to apply.

The iterations are terminated when:

$$\frac{\omega_{max} - \omega_{min}}{2 - \omega_{max}} \leq 0.2 \ . \qquad (3.7.2.10)$$

This algorithm provides an estimate of the optimum overrelaxation factor with rigorous upper and lower bounds before the iterative solution scheme is begun. Since the matrix $\underline{\underline{A}}$ changes with time, recomputation of the estimate of ω_{opt} is performed every n time steps, as set by the user (default is 5). The power method algorithm converges when the dominant eigenvalue of $\underline{\underline{A}}$ is real, and it converges more quickly the smaller the ratio of the second dominant eigenvalue to the dominant eigenvalue. The convergence rate to ω_{opt} can be discouragingly slow when this ratio is near one.

3.7.3 Choosing the Equation Solver

The choice of equation solver depends on the size of the problem and the rate of convergence of the iterative method. The minimum half bandwidth of matrix $\underline{\underline{A}}$ is the product of the two smaller numbers of nodes, N_x, N_y, and N_z. For a half bandwidth of 50, the work of the direct D4 solver is equivalent to about 68 iterations of the L2SOR solver (Price and Coats, 1974). The other consideration is the greater storage requirement of the D4 method, because of the pointer arrays required and the partial fill-in of the $\underline{\underline{A}}_4'$ matrix that occurs. The L2SOR method causes no extra fill-in of nonzero elements in the $\underline{\underline{A}}$ matrix; however, rapid convergence rates are highly dependent on calculation of an accurate optimum overrelaxation factor. It is difficult to give any general rule for selection of matrix solver. Storage requirements probably will determine the best choice. The HST3D program writes out the storage requirements for both methods at the beginning of the simulation.

3.8 GLOBAL-BALANCE CALCULATIONS

Global balances for fluid mass, enthalpy, and solute mass are calculated at the end of each time step. Cumulative totals as well as increments over each time step are computed. The primary use of the global-balance calculations is to aid in the interpretation of the magnitudes of transport that are occurring in the simulated system, and their distribution among the various types of boundary conditions and sources. Each of the system equations represents a mass or energy balance over each cell. By summing over the cells, we obtain the global-balance equations that relate the total change of mass or energy to the net boundary flow rates and the net injection through wells. The solute balance includes sorption and disappearance by reaction. The global-balance equations are integrated over the current time step to obtain the incremental changes.

The fluid-flow, heat-flow, and solute-flow rates at specified-value boundary nodes are obtained by evaluating the appropriate system equation at each specified-boundary-value cell and computing the flow of fluid, heat, or solute across the region boundary for that cell, necessary to satisfy the fluid-balance, heat-balance, or solute-balance equation. Heat and solute fluxes that result from cross-derivative terms are neglected for the heat-balance and solute-balance equations for specified temperature and specified mass-fraction cells. The temporal differencing method is taken into account, also. This means that, for example, if the specified boundary pressure changes over a time step, and centered-in-time differencing is used, then the pressure at the beginning of the time step is effective over the first half of that time step, and the pressure at the end of the time step is effective over the second half. Therefore, the flow rate induced by the new specified pressure is effective for only half the time step. The appropriate equation for the boundary-flow rate calculation is the flow equation for specified-pressure boundary cells, the heat-transport equation for specified-temperature boundary cells, and the solute-transport equation for specified-mass fraction boundary cells. Thus, the cell-balance equations are satisfied exactly for specified-value boundary cells.

The fluid, heat, or solute residual is defined as the change in the amount of the quantity present, minus the net flow of that quantity into the region. Thus, a positive residual means that there is an excess of that quantity present over what would be expected based on the net flows over the time interval. The various flows, amounts present, and residuals are printed in tabular form. Fractional residuals are defined as a ratio of the residual to the larger of the inflow, outflow, or accumulation. The utility of the fractional residuals is not great. It is more informative to look at the residuals relative to the various flows and accumulations in the region.

A mass and energy balance with a small residual is necessary but not sufficient for an accurate numerical simulation. Because the system equations are a balance for each cell, and the method used for calculating the flows at specified-value boundary condition cells insures that the residual will be zero for those cells, and because the fluxes between the cells are conservative, errors in the global-balance equations will result from the following causes: (1) The approximate solution from use of the iterative-matrix equation solver; (2) the degree of convergence on density of the iteration on the solution cycle of the three system equations; (3) the explicit treatment of the cross-dispersive flux terms; (4) the explicit and iterative treatment of well flow rates; and (5) roundoff error in special cases, such as wide variation in parameter magnitudes. Errors caused by discretization in time or space will not be revealed by these global-balance calculations. However, the inaccuracies resulting from too-long a time step under conditions of significantly nonlinear parameters will be evident. Significant nonlinearity can be caused by large variations in the density and viscosity fields, for example.

4. COMPUTER CODE DESCRIPTION

4.1 CODE ORGANIZATION

The HST3D computer code is written in FORTRAN 77. Only code conforming to American National Standards Institute (ANSI) standards (American National Standards Institute, 1978) has been used to maximize program portability. The present version of the program code consists of a main routine and 49 subroutines. The program length is approximately 12,000 lines of code. Many FORTRAN statements occupy multiple lines. The following is a list of the routines and a brief description of their function:

HST3D - Main routine that drives the program execution. The basic steps are: (1) Read, error check, initialize, and write output for space allocation; (2) read, error check, initialize, and write output for static or time-invariant information; (3) read, error check, initialize and write output for transient information; (4) start the time-step calculations by calculating flow- and transport-equation coefficients, applying the boundary conditions, calculating the source-sink well terms; (5) perform the assembly and solution of the three equations in turn, iterating to convergence; (6) perform the summary calculations; (7) write the output information for the time plane; and (8) dump restart data to a disc file if desired. Then return to step 4 and continue until the time for a change in boundary conditions or source terms occurs. At this time return to step 3. Continue until the simulation is finished. Then (9), plot dependent variables as a function of time if desired; and (10) close files and terminate program execution.

The following subroutines are described in their order of execution:

READ1 - Reads the data pertaining to allocation of computer storage space for the problem.

ERROR1 - Checks for errors in the data read by READ1.

INIT1 - Initializes the spatial-allocation data, including the pointers for the two variable-partitioned arrays. If necessary, sets the inch-pound-to-metric conversion factors and their inverses, and sets the unit labels.

WRITE1 - Writes to a disc file information about the storage allocation and array-size requirements.

READ2 - Reads the data pertaining to all the time invariant or static information, including fluid and porous-media properties, grid geometry, equation-solution method, and desired output.

ERROR2 - Checks the data read by READ2 for errors.

INIT2 - Initializes the calculated static data for the simulation.

WRITE2 - Writes the static data to a disc file.

READ3 - Reads the transient data, including boundary condition and source-sink information, plus time-step, calculation, and printout information.

INIT3 - Initializes the calculated transient data.

ERROR3 - Checks the transient data read by READ3 for errors.

WRITE3 - Writes the transient data to a disc file.

COEFF - Calculates the coefficients for the flow-, heat- and solute-transport equations, and adjusts the automatic time step if selected. These coefficients include conductances, dispersion coefficients, and interstitial velocities.

WRITE4 - Writes the coefficients to a disc file as desired.

APLYBC - Applies the boundary conditions to the set of equations.

WELLSS - Calculates and applies the well source-sink terms.

ITER - Assembles and solves the three equations iteratively for a given time plane.

SUMCAL - Performs the summary calculations at the end of a time step. This includes flow, heat, and solute-mass balances, and flow rates.

WRITE5 - Writes to a disc file the desired information at the end of a time step. This may include pressure, temperature, and mass-fraction distributions, interstitial velocities, fluid viscosities, fluid densities, and summary tables of flow rates and balances of flow, heat, and solute mass.

DUMP - Dumps restart information to a disc file, if desired.

PLOTOC - Creates character-string plots of selected dependent variables (pressure temperature, or solute-mass fraction) as a function of time at the end of the simulation, and plots observed data if desired.

CLOSE - Closes files, deletes unused files, prints the total simulation time and the number of time planes, the number of restart and map records written, and any error messages.

The following subroutines are listed in approximate order of execution. Some are called from several routines, and some are optional.

ORDER - Determines the node numbering for the D4 reordering scheme for the direct-matrix equation solver.

REWI - Reads, error checks, echo writes, and initializes array elements that are input as zones of constant values over a rectangular prism of cells, or are input as node by node distributions.

REWI3 - The same as REWI, but for parameters that occur in sets of three, such as vectors or principal components of tensors.

IREWI - The same as REWI, but for parameters that are integers.

VSINIT - Calculates parameters for the viscosity function.

ETOM1 - Converts static data from inch-pound units to metric units.

ETOM2 - Converts transient data from inch-pound units to metric units.

PRNTAR - Writes one-dimensional or two-dimensional arrays in tabular form to a disc file.

ZONPLT - Creates two-dimensional maps on the printer of the porous-media zones contained in the simulation region.

INTERP - Performs one-dimensional or two-dimensional linear or bilinear tabular interpolation, as required.

VISCOS - Calculates fluid viscosity as a function of temperature and solute-mass fraction.

TOFEP - Determines temperature as a function of enthalpy and pressure by tabular interpolation.

WELRIS - Performs the pressure and temperature calculation up or down the well-riser pipe, using simultaneous solution of the two ordinary differential equations.

ASEMBL - Assembles the coefficients of the modified flow, modified heat-transport, and solute-transport equations at each time step.

CALCC - Calculates the elements of the change in fluid-mass, change in heat, or change in solute-mass matrix for a given cell at each iteration at each time step.

CRSDSP - Calculates the components of the cross-dispersion tensor, that are evaluated explicitly in the transport equations at each iteration.

D4DES - Solves the matrix equation, using alternating-diagonal reordering and a direct Gaussian elimination algorithm.

SOR2L - Solves the matrix equation, using a two-line, successive-overrelaxation algorithm, with the lines oriented in a selected coordinate direction.

L2SOR - Invokes the two-line, successive-overrelaxation solver for each coordinate direction to estimate the optimum overrelaxation parameter, and to solve the equations.

MAP2D - Creates two-dimensional contour maps on the printer of selected variables with contour intervals divided into zones.

PLOT - Creates plots of pressure, temperature, or mass fraction versus time.

ERRPRT - Writes the error messages for a given simulation to a disc file.

SBCFLO - Calculates the flow rates at specified-value boundary-condition cells for the global balances.

WBBAL - Calculates flow rates for each well of fluid, heat, and solute for the summary calculations.

WBCFLO - Calculates the flow rates at the well-bore boundary for a single well in the cylindrical coordinate system.

BSODE - Integrates the coupled ordinary differential equations for pressure and temperature up or down the well-riser casing, using the Bulirsch-Stoer algorithm for rational polynomial extrapolation.

WFDYDZ - Calculates derivatives of pressure and temperature along the well-riser casing for the Bulirsch-Stoer integration algorithm.

If errors occur, the error checking that is in progress continues to completion, but then, information to that point in the calculations is written out, and the simulation is aborted.

A chart showing the main sequence of subroutine execution, the time step and transient data loops and the linkage between the subroutines appears in figure 4.1. The primary subroutines are on the left and the secondary and utility subroutines are to the right. The sequence of execution is from top to bottom of the leftmost column. Some utility subroutines are listed more than once for graphical clarity.

4.2 MEMORY ALLOCATION, ARRAY-SIZE REQUIREMENTS, AND SUBPROGRAM COMMUNICATION

A semi-dynamic method for array-storage allocation is employed in the simulator program (Akin and Stoddart, 1975, p. 114). Instead of dimensioning each of the arrays at some maximum value, a different approach is taken. The various arrays whose size depends on the number of nodes in the region, or the number of cells with a given type of boundary condition, or the number of wells, or the type of equation solver selected, are all contained in two large arrays; one array for real variables and one array for integer variables.

These two large arrays are partitioned into the required subarrays, based on the storage-allocation information provided. Pointer variables are used to indicate the location of the first element of each subarray in its large array. Thus, the variable-length subarrays are passed to the subroutines and functions though the calling arguments using the pointer variables. This means that some subprograms have a large number of arguments. The advantage of this approach is that the lengths of the two large, variably partitioned arrays, VPA and IVPA, are set during compilation of the main routine, and only the main routine must be recompiled if the lengths are to be changed. The compiled length of the VPA array is contained in the variable ILVPA and the compiled length of the IVPA array is contained in the variable ILIVPA.

A rough estimate of the sizes required for the two large arrays is given by the following equations. For the large variably partitioned integer array,

$$ILI = 8\ NXYZ + 6\ NPMZ + 5\ NWEL + NBC; \tag{4.2.1}$$

where

ILI is the estimated length of the large variably partitioned integer array;

NXYZ is the number of nodes in the region;

NPMZ is the number of porous medium zones;

NWEL is the number of wells; and

NBC is the number of boundary condition nodes.

For the large real array:

$$ILR = 70\ NXYZ + 12\ NPMZ + 60\ NWEL + 60\ NBC; \tag{4.2.2}$$

where

ILR is the estimated length of the large variably partitioned integer array.

Equations 4.2.1 and 4.2.2 are based on solving all three equations and they overestimate the storage requirements in the interest of simplifying the estimate calculation. During the storage-allocation step, the actual required lengths of the large real and integer arrays are calculated and printed. Execution is aborted if insufficient space has been set during compilation. If redimensioning is necessary, the VPA and IVPA array sizes are redimensioned, and the variable ILVPA and is set to the compiled dimension of the VPA array, and the variable ILIVPA is set to the compiled dimension of the VPA array in the main HST3D program.

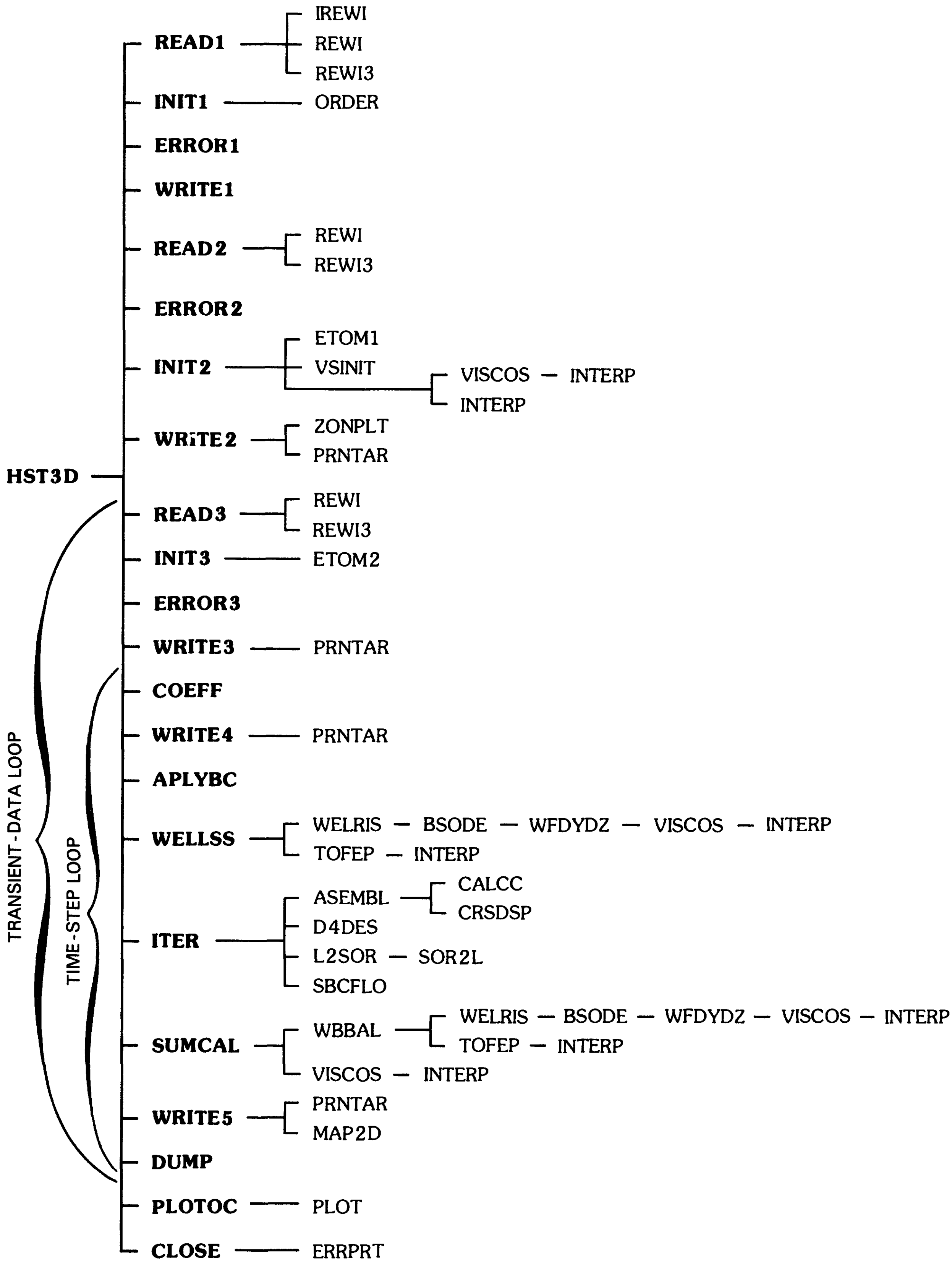


Figure 4.1.--Connection chart for the HST3D main program and subprogram showing routine hierarchy and calculation loops.

Some of the subarrays share storage space if they are used sequentially and the contents do not need to be retained for the duration of the simulation. Table 4.1 shows the partitioning of the large integer array; table 4.2 shows the partitioning of the large real array, with the shared storage indicated by subarrays on the same horizontal line. The size of each subarray and conditions for its inclusion (if optional) also are given. A significant savings in space is obtained by not including arrays that deal exclusively with either heat or solute transport in cases when transport of only one or the other is being simulated.

All other parameters and variables are passed through common blocks. Named common blocks are used, with the name being an abbreviation of the subprogram where that particular common block is first used with a suffix letter. There are many common blocks, because the variables are sorted into usage groups, so that each subprogram has access to only the common variables that it needs (as nearly as practical). All common blocks appear in the main program for easy reference and for static-storage allocation on some computer systems.

Another programming convention used is that, passing of subarrays through one subprogram to another is done by making the entire large array available to the calling subprogram along with the necessary pointer variables. This reduces the number of arguments for the calling subprogram.

Table 4.1.--*Space allocation within the large, variably partitioned, integer array, IVPA*

[--, no space sharing or no conditions; b.c., boundary condition]

Subarray	Dimension	Other arrays that share the space	Conditions for inclusion of optional subarrays
IBC	NXYZ	--	--
I1Z	NPMZ	--	--
I2Z	NPMZ	--	--
J1Z	NPMZ	--	--
J2Z	NPMZ	--	--
K1Z	NPMZ	--	--
K2Z	NPMZ	--	--
LPRNT	NXYZ	LBW,INZONE,JWEL,IBCMAP	--
IW	NWT	--	Wells present
JW	NWT	--	Wells present
LCTOPW	NWT	--	Wells present
LCBOTW	NWT	--	Wells present
WQMETH	NWT	--	Wells present
MAIFC	NAIFC	--	Aquifer influence function b.c.
MFBC	NFBC	--	Specified flux b.c.
MHCBC	NHCBC	--	Heat conduction b.c.
MLBC	NLBC	--	Leakage b.c.
MSBC	NPTCBC	--	Specified value b.c.
CI	6(NXYZ/2+1)	--	D4 matrix solver
IDIAG	NXYZ/2+1	--	D4 matrix solver
RBW	NXYZ/2+1	--	D4 matrix solver
ID4NO	NXYZ	--	D4 matrix solver

Table 4.2.--*Space allocation within the large, variably partitioned real array, VPA*

[--, no space sharing or no conditions; b.c., mean boundary condition]

Subarray	Dimension	Other arrays that share the space	Conditions for inclusion of optional subarrays
X	NX, NR	RR	--
Y	NY	--	Cartesian coordinates
Z	NZ	--	--
RM	NR-1	--	Cylindrical coordinates
ARX	NXYZ	TO	--
ARY	NXYZ	--	Cartesian coordinates
ARZ	NXYZ	--	--
POROS	NPMZ	--	--
ABPM	NPMZ	--	--
KXX	NPMZ	--	--
KYY	NPMZ	--	--
KZZ	NPMZ	--	--
RCPPM	NPMZ	--	Heat transport
KTHX	NPMZ	--	Heat transport
KTHY	NPMZ	--	Heat transport
KTHZ	NPMZ	--	Heat transport
ALPHL	NPMZ	--	Heat or solute transport
ALPHT	NPMZ	--	Heat or solute transport
DBKD	NPMZ	--	Solute transport
PV	NXYZ	POS	--
PMCV	NXYZ	--	--
PVK	NXYZ	--	Solute transport
PMCVK	NXYZ	--	Solute transport
PMHV	NXYZ	TOS	Heat transport
PMCHV	NXYZ	--	Heat transport
P	NXYZ	POW	--
DP	NXYZ	HWT,PNP	--
T	NXYZ	TOW	Heat transport
DT	NXYZ	UTBC,TNP,TQFLX	Heat transport
C	NXYZ	COW	Solute transport
DC	NXYZ	UCBC,CNP,CQFLX	Solute transport
DEN	NXYZ	TC	--
VIS	NXYZ	--	--
EH	NXYZ	--	Heat transport
TX	NXYZ	--	--
TY	NXYZ	--	Cartesian coordinates
TZ	NXYZ	PCW	--
TFX	NXYZ	HDPRNT,UPHILB	--
TFY	NXYZ	--	Cartesian coordinates
TFZ	NXYZ	UVAIFC,UDENLB,AMAP	--
THX	NXYZ	UDTHHC,QHFX,KTXPM	Heat transport
THXY	NXYZ	--	Heat transport, cartesian coordinates, full dispersion tensor

Table 4.2.--*Space allocation within the large, variably partitioned real array, VPA*--Continued

Subarray	Dimension	Other arrays that share the space	Conditions for inclusion of optional subarrays
THXZ	NXYZ	--	Heat transport, full dispersion tensor
THY	NXYZ	QHFY,KTYPM	Heat transport, cartesian coordinates
THYX	NXYZ	--	Heat transport, full dispersion tensor, cartesian coordinates
THYZ	NXYZ	--	Heat transport, full dispersion tensor, cartesian coordinates
THZ	NXYZ	QHFZ,UKHCBC,KTZPM	Heat transport
THZX	NXYZ	--	Heat transport, full dispersion tensor
THZY	NXYZ	--	Heat transport, full dispersion tensor, cartesian coordinates
TSX	NXYZ	QSFX	Solute transport
TSXY	NXYZ	--	Solute transport, cartesian coordinates, full dispersion tensor
TSXZ	NXYZ	--	Solute transport, full dispersion tensor
TSY	NXYZ	QSFY	Solute transport, cartesian coordinates
TSYX	NXYZ	--	Solute transport, cartesian coordinates, full dispersion tensor
TSYZ	NXYZ	--	Solute transport, cartesian coordinates, full dispersion tensor
TSZ	NXYZ	QSFZ	Solute transport
TSZX	NXYZ	--	Solute transport, full dispersion tensor
TSZY	NXYZ	--	Solute transport, full dispersion tensor, cartesian coordinates
SXX	NXYZ	QFFX,ARXBC,PCS	--
SYY	NXYZ	QFFY,ARYBC	Cartesian coordinates
SZZ	NXYZ	QFFZ,ARZBC	--
RF	NXYZ	APRNT,UVISLB	--
RH	NXYZ	--	Heat transport
RH1	NXYZ	--	Heat transport
RS	NXYZ	CCW	Solute transport
RS1	NXYZ	--	Solute transport
URR1	NXYZ	--	Solute transport

Table 4.2.--*Space allocation within the large, variably partitioned real array, VPA*--Continued

Subarray	Dimension	Other arrays that share the space	Conditions for inclusion of optional subarrays
RHS	NXYZ	FRAC,UZELB,TCS	Solute transport
CC24	NXYZ	UBBLB	--
CC34	NXYZ	--	--
CC35	NXYZ	TCW	--
VA	7·NXYZ	VXX,AA1,MOBW,UDENBC,UKLB	--
		VYY,AA2	--
		VZZ,AA3	--
		AA4	--
WI	NWEL·NZ	--	Wells present
QWLYR	NWEL·NZ	WCF	Wells present
QHLYR	NWEL·NZ	--	Wells present, heat transport
QSLYR	NWEL·NZ	--	Wells present, solute transport
DQWDPL	NWEL·NZ	--	Wells present
WRANGL	NWEL	--	Wells present
WBOD	NWEL	--	Wells present
WRISL	NWEL	--	Wells present
WRID	NWEL	--	Wells present
WRRUF	NWEL	--	Wells present
KTHWR	NWEL	--	Wells present, heat transport
DTHWR	NWEL	--	Wells present, heat transport
KTHAWR	NWEL	--	Wells present, heat transport
HTCWR	NWEL	--	Wells present, heat transport
TABWR	NWEL	--	Wells present, heat transport
TATWR	NWEL	--	Wells present, heat transport
TWKT	NWEL	--	Wells present, heat transport
TWSUR	NWEL	--	Wells present, heat transport
EHWKT	NWEL	--	Wells present, heat transport
EHWSUR	NWEL	--	Wells present, heat transport
PWKTS	NWEL	--	Wells present
PWKT	NWEL	--	Wells present
DPWKT	NWEL	--	Wells present
PWSURS	NWEL	--	Wells present
PWSUR	NWEL	--	Wells present

Table 4.2.--*Space allocation within the large, variably partitioned real array, VPA*--Continued

Subarray	Dimension	Other arrays that share the space	Conditions for inclusion of optional subarrays
CWKT	NWEL	--	Wells present, solute transport
QWV	NWEL	--	Wells present
QWM	NWEL	--	Wells present
QHW	NWEL	--	Wells present, heat transport
QSW	NWEL	--	Wells present, solute transport
WFICUM	NWEL	--	Wells present
WFPCUM	NWEL	--	Wells present
WHICUM	NWEL	--	Wells present, heat transport
WHPCUM	NWEL	--	Wells present, heat transport
WSICUM	NWEL	--	Wells present, solute transport
WSPCUM	NWEL	--	Wells present, solute transport
VASBC	7·NPTCBC	--	Specified-value b.c.
RHSBC	NPTCBC	--	Specified-value b.c.
QFSBC	NPTCBC	--	Specified-value b.c.
PSBC	NPTCBC	--	Specified-value b.c.
QHSBC	NPTCBC	--	Specified-value b.c. and heat transport
TSBC	NPTCBC	--	Specified-value b.c. and heat transport
QSSBC	NPTCBC	--	Specified-value b.c. and solute transport
CSBC	NPTCBC	--	Specified-value b.c. and solute transport
ARXFBC	NFBC	--	Specified-flux b.c.
ARYFBC	NFBC	--	Specified-flux b.c., cartesian coordinates
ARZFBC	NFBC	--	Specified-flux b.c.
QFFBC	NFBC	QFBCV	Specified-flux b.c.
DENFBC	NFBC	--	Specified-flux b.c.
QHFBC	NFBC	--	Specified-flux b.c. and heat transport
TFLX	NFBC	--	Specified-flux b.c. and heat transport
QSFBC	NFBC	--	Specified-flux b.c. and solute transport
CFLX	NFBC	--	Specified-flux b.c. and solute transport
ALBC	NLBC	--	Leakage b.c.

Table 4.2.--*Space allocation within the large, variably partitioned real array, VPA*--Continued

Subarray	Dimension	Other arrays that share the space	Conditions for inclusion of optional subarrays
BLBC	NLBC	--	Leakage b.c.
KLBC	NLBC	--	Leakage b.c.
ZELBC	NLBC	--	Leakage b.c.
BBLBC	NLBC	--	Leakage b.c.
HLBC	NLBC	PHILBC	Leakage b.c.
DENLBC	NLBC	--	Leakage b.c.
VISLBC	NLBC	--	Leakage b.c.
TLBC	NLBC	--	Leakage b.c. and heat transport
CLBC	NLBC	--	Leakage b.c. and solute transport
QFLBC	NLBC	--	Leakage b.c.
QHLBC	NLBC	--	Leakage b.c. and heat transport
QSLBC	NLBC	--	Leakage b.c. and solute transport
AAIF	NAIFC	--	Aquifer-influence-function b.c.
BAIF	NAIFC	--	Aquifer-influence-function b.c.
VAIF	NAIFC	--	Aquifer-influence-function b.c.
WCAIF	NAIFC	--	Aquifer-influence-function b.c.
DENOAR	NAIFC	--	Aquifer-influence-function b.c.
PAIF	NAIFC	--	Aquifer-influence-function b.c.
TAIF	NAIFC	--	Aquifer-influence-function b.c. and heat transport
CAIF	NAIFC	--	Aquifer-influence-function b.c. and solute transport
QFAIF	NAIFC	--	Aquifer-influence-function b.c.
QHAIF	NAIFC	--	Aquifer-influence-function b.c. and heat transport
QSAIF	NAIFC	--	Aquifer-influence-function b.c. and solute transport
ZHCBC	NHCN	--	Heat-conduction b.c.
A1HC	NHCN	--	Heat-conduction b.c.
A2HC	NHCN	--	Heat-conduction b.c.
A3HC	NHCN	--	Heat-conduction b.c.
KARHC	NHCBC	--	Heat-conduction b.c.

Table 4.2.--*Space allocation within the large, variably partitioned real array, VPA*--Continued

Subarray	Dimension	Other arrays that share the space	Conditions for inclusion of optional subarrays
DTHHC	NHCBC	--	Heat-conduction b.c.
QHCBC	NHCBC	--	Heat-conduction b.c.
DQHCDT	NHCBC	--	Heat-conduction b.c.
THCBC	NHCBC·NHCN	--	Heat-conduction b.c.
TPHCBC	NHCBC·NHCN	--	Heat-conduction b.c.
A4	MAXNAL	D	D4 direct solution method
	(D4 SOLVER)	TM1	or two-line, successive-
	OR	TM2	overrelaxation
	7·NXYZ	TP1	iterative-solution
	(L2SOR SOLVER)	TP2	method
		XX	
		XXN	

4.3 FILE USAGE

The heat- and solute-transport program uses several sequential-access files on disc. The following list gives the FORTRAN unit number and a description of each file:

FILE 5 Input data with comments stripped out.

FILE 6 Output data in character form for the line printer or video screen.

FILE 7 Plot data for dependent variables versus time.

FILE 8 Output data for a restart run.

FILE 9 Input data for a restart run.

FILE 10 Input data with comment lines

FILE 11 Echo of the input data as it is read.

These files must be declared or established on the computer system before program execution. The present version of the program also requires pre-opening of these files before starting execution. File names are assigned when the files are opened, using the job-control language of the computer employed.

4.4 INITIALIZATION OF VARIABLES

In general, all real and integer variables are set to zero and logical variables are set to false by explicitly coded instructions at the beginning of program execution. Certain input parameters are set to default values if no data or data values of zero are input. These are indicated in the input-file description. Two tables are initialized for the enthalpy of pure water as a function of temperature and pressure. These tables are used for the heat-transport calculation.

4.5 PROGRAM EXECUTION

The program is designed to be executed by the job-control language (JCL) or command-procedure language (CPL) of the computer employed. The data-input file for a given program must have been created previously. The job-control language statements should declare and open the FORTRAN files listed above, assign file names, then start program execution. At the end of the simulation, the JCL or CPL statements should close all files and delete scratch files. Unused files are deleted by the program in the CLOSE subroutine. Execution of the command-procedure, control-statement file is usually done interactively at a terminal, so file names can be provided by the user.

4.6 RESTART OPTION

The program has the option for restarting from an intermediate or ending time of the simulation. The user may specify that restart information is to be written to the restart file every nth time step (record 3.8.3 of the input file). Then, the ability to continue in a later run is created. This is useful; for example, if boundary conditions or sources need to be changed, based on the results of the simulation, or if a steady-state flow field needs to be established, before a heat or solute source is introduced. The other use of the restart option is to specify that restart information is to be written at periodic time-step intervals. This is insurance against computer-system failure, so that long simulation runs will not have to be repeated entirely.

The information written to disc for restart consists of the two large, variably partitioned arrays, and those common blocks that contain parameters and variables that are active during the simulation time-step loop. This information can be written only at the end of a time step. To restart the simulation, a small amount of data is needed in the input file, including the first three lines of the READ1 record group and the transient-boundary condition and fluid-source information of the READ3 record group. After the restart-data file is read, execution continues at the beginning of the subroutine (READ3) that reads the transient information, shown in figure 4.1. Only the transient data that are to be changed from the last values of the previous simulation need to be read at the beginning of a restart. From this point onward in the simulation, there is no difference between a restart run and one that has just gone through a change in transient-boundary conditions. In particular, time is measured relative to the beginning of the original simulation period.

One consequence of the way restarts are implemented is the restriction that the number and type of boundary condition and source nodes cannot be changed from the values for the original simulation run. To do so will cause a fatal error resulting from a file length conflict. At the beginning of a restart, the program will attempt to find the restart file whose time-plane value agrees within 0.1 percent of the specified time value for restarting. Failure to locate the specified restart file causes a fatal error.

5. THE DATA-INPUT FILE

The data-input file has two general characteristics: (1) It is free format for ease of preparation at a computer terminal; and (2) it may be freely commented for rapid identification of the data items. The free format is supported by FORTRAN 77, and is sometimes referred to as list-directed input.

5.1. LIST DIRECTED INPUT

The data values may be located in any position of the record, provided they correspond in number and type with the input list. Data are separated by commas or blanks, with multiple blanks allowed. Character strings must be enclosed in quotation marks (apostrophies on some computers). There is a third delimiter in addition to a comma and a blank: the slash, /. A slash terminates a record and any remaining items in the input list are left unchanged from their previous values. On some computer systems, there must be a space before the slash. A data item within an input list may be left unchanged by separating the preceding item from the subsequent item with two commas; in other words, making no entry for that item.

The list-directed input of a record continues until the end-of-record slash is encountered, or the input list is satisfied. If the input list is not satisfied at the end of a data line, the program will continue to read additional data lines, until the list is satisfied, or the end of the file is reached. Having an insufficient number of data items in the input file is a common error. It usually results from a misinterpretation of the amount or type of data required by a given program option.

List-directed input also allows for a repeat count. Data in the form:

$$n*d; \qquad (5.1.1)$$

where

n is the repeat count integer; and

d is the data item;

causes n consecutive values of d to be input.

5.2. PREPARING THE DATA-INPUT FILE

To simplify the preparation of the data-input file, a file is available containing only comments. This file is presented as table 5.1. These comments identify all the input-variable names, indicate the logical ones, show the conditions for the optional items, and give the default values, where used. The user actually can enter the data between the lines of a copy of this file, using section 5.2.2 as a guide. This data-input form should make it easier to create and modify the data-input file. Only the variable names used in the program are given. For definitions see section 5.2.2 or 11.1. Optional input records are indicated by (O) followed by the logical variables that must be true, or the numerical conditions that must be met for inclusion of a given input record. The numbers in brackets after a variable give the record where that variable is read. The following section contains a line-by-line presentation of the data-file form with an explanation of all the variables and options.

Many published sources exist of fluid and porous-medium properties, and transport parameters such as compressibilities, heat capacities, and equilibrium distribution coefficients. Typical examples are Perry and others (1963), Clark (1966), Weast and others (1964), and Mercer and others (1982).

5.2.1. General Information

Comment lines are numbered in the format C.N1.N2.N3 where C denotes a comment record; N1 is the read-group number; and N2.N3 is the record or line number. The read-group number denotes which subroutine, READ1, READ2, READ3, or PLOTOC, reads that record for the values 1 through 4 respectively. The record number identifies a line of input data, with the two component number enabling a logical group structure to be assigned. A suffix letter associated with N2 or N3 indicates that one of the numbered set with that letter must be selected exclusively. Optional data records are indicated to the right of the record list of variables, and the conditions under which these data are required are explained. If a particular input record is not needed, it is manditory that it be omitted. A record number in square brackets following a variable indicates the record where that variable is first set.

Input by i,j,k range means that an array of data is to be specified for a rectangular prism of nodes. The extent of this prism is defined by the ranges of the i,j, and k indices. One may specify a plane, or a line, or a point instead of a prism by making the appropriate indices equal. The value to be entered into the array locations pertaining to the nodes contained with the prism is specified next. There is the capability to replace, multiply or add to existing data within the prism. The specified value fills the specified array locations or operates on the existing values in those locations. In some cases, three arrays are affected for the nodes within the specified prism. There is a fourth option for data input which allows for a sequence of different values to be entered into consecutive node locations within the prism. The data will be loaded in order of increasing node number, that is, in order of increasing i, then increasing j, and then increasing k. With this input scheme, subregions of uniform parameters within the simulation region are easily defined. A parameter distribution that varies continuously in space is less convenient to define. It is permissible to use rectangular prisms that include cells that are outside the simulation region.

Table 5.1.--*Data-input form*

```
C.....HST DATA-INPUT FORM
C...   NOTES:
C...   INPUT LINES ARE DENOTED BY C.N1.N2.N3 WHERE
C...        N1 IS THE READ GROUP NUMBER, N2.N3 IS THE RECORD NUMBER
C...        A LETTER INDICATES AN EXCLUSIVE RECORD SELECTION MUST BE MADE
C...          I.E. A OR B OR C
C...   (O) - OPTIONAL DATA WITH CONDITIONS FOR REQUIREMENT
C...     A RECORD NUMBER IN SQUARE BRACKETS IS THE RECORD WHERE THAT PARAMETER IS FIRST SET
C.....INPUT BY I,J,K RANGE FORMAT IS;
C.0.1.. I1,I2,J1,J2,K1,K2
C.0.2..  VAR1,IMOD1,[VAR2,IMOD2,VAR3,IMOD3]
C...  USE AS MANY OF LINE 0.1 & 0.2 SETS AS NECESSARY
C...  END WITH LINE 0.3
C.0.3.. 0 /  THE SPACE IS REQUIRED
C...   {NNN} - INDICATES THAT THE DEFAULT NUMBER, NNN, IS USED IF A ZERO IS ENTERED FOR THAT VARIABLE
C...   (T/F) - INDICATES A LOGICAL VARIABLE
C...   [I] - INDICATES AN INTEGER VARIABLE
C--------------------------------------------------------------------------------------------
C.....START OF THE DATA FILE
C.....DIMENSIONING DATA - READ1
C.1.1 .. TITLE LINE 1
C.1.2 .. TITLE LINE 2
C.1.3 .. RESTRT(T/F),TIMRST
C.....IF RESTRT IS TRUE, PROCEED TO READ3 GROUP [3.1]
C.1.4 .. HEAT,SOLUTE,EEUNIT,CYLIND,SCALMF; ALL (T/F)
C.1.5 .. NX,NY,NZ,NHCN
C.1.6 .. NPTCBC,NFBC,NAIFC,NLBC,NHCBC,NWEL
C.1.7 .. NPMZ
C.1.8 .. SLMETH[I],LCROSD(T/F)
C.1.9 .. IBC BY I,J,K RANGE {0.1-0.3} ,WITH NO IMOD PARAMETER, FOR EXCLUDED CELLS
C.1.10 .. RDECHO(T/F)
C------------------------------------------------------------------------------
C.....STATIC DATA - READ2
C.....OUTPUT INFORMATION
C.2.1 .. PRTRE(T/F)
C.....COORDINATE GEOMETRY INFORMATION
C..... RECTANGULAR COORDINATES
C.2.2A.1 .. UNIGRX,UNIGRY,UNIGRZ; ALL (T/F); (O) - NOT CYLIND [1.4]
C.2.2A.2A .. X(1),X(NX);(O) - UNIGRX [2.2A.1]
C.2.2A.2B .. X(I);(O) - NOT UNIGRX [2.2A.1]
C.2.2A.3A .. Y(1),Y(NY);(O) - UNIGRY [2.2A.1]
C.2.2A.3B .. Y(J);(O) - NOT UNIGRY [2.2A.1]
C.2.2A.4A .. Z(1),Z(NZ);(O) - UNIGRZ [2.2A.1]
C.2.2A.4B .. Z(K);(O) - NOT UNIGRZ [2.2A.1]
C.....CYLINDRICAL COORDINATES
```

Table 5.1.--*Data-input form*--Continued

```
C.2.2B.1A .. R(1),R(NR),ARGRID(T/F);(0) - CYLIND [1.4]
C.2.2B.1B .. R(I);(0) - NOT ARGRID [2.2B.1A];(0) - CYLIND [1.4]
C.2.2B.2 .. UNIGRZ(T/F);(0) - CYLIND [1.4]
C.2.2B.3A .. Z(1),Z(NZ);(0) - UNIGRZ [2.2B.3A],CYLIND [1.4]
C.2.2B.3B .. Z(K);(0) - NOT UNIGRZ [2.2B.3A],CYLIND [1.4]
C.2.3.1 .. TILT(T/F);(0) - NOT CYLIND [1.4]
C.2.3.2 .. THETXZ,THETYZ,THETZZ;(0) - TILT [2.3.1] AND NOT CYLIND [1.4]
C.....FLUID PROPERTY INFORMATION
C.2.4.1 .. BP
C.2.4.2 .. P0,T0,W0,DENF0
C.2.4.3 .. W1,DENF1;(0) - SOLUTE [1.4]
C.2.5.1 .. NOTV0,TVF0(I),VISTF0(I),I=1 TO NOTV0;(0) - HEAT [1.4] OR HEAT [1.4] AND SOLUTE [1.4] OR .NOT.HEAT AND .NOT.SOLUTE [1.4]
C.2.5.2 .. NOTV1,TVF1(I),VISTF1(I),I=1 TO NOTV1;(0) -  SOLUTE [1.4] AND HEAT [1.4]
C.2.5.3 .. NOCV,TRVIS,CVIS(I),VISCTR(I),I=1 TO NOCV;(0) - SOLUTE [1.4]
C.....REFERENCE CONDITION INFORMATION
C.2.6.1 .. PAATM
C.2.6.2 .. P0H,T0H
C.....FLUID THERMAL PROPERTY INFORMATION
C.2.7 .. CPF,KTHF,BT;(0) - HEAT [1.4]
C.....SOLUTE INFORMATION
C.2.8 .. DM,DECLAM;(0) - SOLUTE [1.4]
C.....POROUS MEDIA ZONE INFORMATION
C.2.9.1 .. IPMZ,I1Z(IPMZ),I2Z(IPMZ),J1Z(IPMZ),J2Z(IPMZ),K1Z(IPMZ),K2Z(IPMZ)
C.....USE AS MANY 2.9.1 LINES AS NECESSARY
C.2.9.2 .. END WITH 0 /
C.....POROUS MEDIA PROPERTY INFORMATION
C.2.10.1 .. KXX(IPMZ),KYY(IPMZ),KZZ(IPMZ),IPMZ=1 TO NPMZ [1.7]
C.2.10.2 .. POROS(IPMZ),IPMZ=1 TO NPMZ [1.7]
C.2.10.3 .. ABPM(IPMZ),IPMZ=1 TO NPMZ [1.7]
C.....POROUS MEDIA THERMAL PROPERTY INFORMATION
C.2.11.1 .. RCPPM(IPMZ),IPMZ=1 TO NPMZ [1.7];(0) - HEAT [1.4]
C.2.11.2 .. KTXPM(IPMZ),KTYPM(IPMZ),KTZPM(IPMZ),IPMZ=1 TO NPMZ [1.7];(0) - HEAT [1.4]
C.....POROUS MEDIA SOLUTE AND THERMAL DISPERSION INFORMATION
C.2.12 .. ALPHL(IPMZ),ALPHT(IPMZ),IPMZ=1 TO NPMZ [1.7];(0) - SOLUTE [1.4] OR HEAT [1.4]
C.....POROUS MEDIA SOLUTE PROPERTY INFORMATION
C.2.13 .. DBKD(IPMZ),IPMZ=1 TO NPMZ [1.7];(0) - SOLUTE [1.4]
C.....SOURCE-SINK WELL INFORMATION
C.2.14.1 .. RDWDEF(T/F);(0) - NWEL [1.6] > 0
C.2.14.2 .. IMPQW(T/F);(0) - NWEL [1.6] > 0 AND NOT CYLIND [1.4]
C.2.14.3. .. IWEL,IW,JW,LCBOTW,LCTOPW,WBOD,WQMETH[I];(0) - RDWDEF [2.14.1],
C.2.14.4 .. WCF(L);L = 1 TO NZ (EXCLUSIVE) BY ELEMENT
C.2.14.5 .. WRISL,WRID,WRRUF,WRANGL;(0) - RDWDEF [2.14.1] AND WRCALC(WQMETH [2.14.3] >30)
C.2.14.6 .. HTCWR,DTHAWR,KTHAWR,KTHWR,TABWR,TATWR;(0) - RDWDEF [2.14.1]  WRCALC(WQMETH [2.14.3] >30) AND HEAT [1.4]
C.....USE AS MANY 2.14.3-6 LINES AS NECESSARY
C.2.14.7 .. END WITH 0 /
```

Table 5.1.--*Data-input form*--Continued

```
C.2.14.8 .. MXITQW{14},TOLDPW{6.E-3},TOLFPW{.001},TOLQW{.001},DAMWRC{2.},DZMIN{.01},EPSWR{.001};(0) - RDWDEF [2.14.1]
C.....          AND WRCALC(WQMETH[2.14.3] >30)
C.....BOUNDARY CONDITION INFORMATION
C.....     SPECIFIED VALUE B.C.
C.2.15 .. IBC BY I,J,K RANGE {0.1-0.3} WITH NO IMOD PARAMETER,;(0) - NPTCBC [1.6] > 0
C.....     SPECIFIED FLUX B.C.
C.2.16 .. IBC BY I,J,K RANGE {0.1-0.3} WITH NO IMOD PARAMETER,;(0) - NFBC [1.6] > 0
C.....     AQUIFER AND RIVER LEAKAGE B.C.
C.2.17.1 .. IBC BY I,J,K RANGE {0.1-0.3} WITH NO IMOD PARAMETER;(0) - NLBC [1.6] > 0
C.2.17.2 .. KLBC,BBLBC,ZELBC BY I,J,K RANGE {0.1-0.3};(0) - NLBC [1.6] > 0
C.....          RIVER LEAKAGE B.C.
C.2.17.3 .. I1,I2,J1,J2,KRBC,BBRBC,ZERBC;(0) - NLBC [1.6] > 0
C.2.17.4 .. END WITH 0 /
C.....     AQUIFER INFLUENCE FUNCTIONS
C.2.18.1 .. IBC BY I,J,K RANGE {0.1-0.3} WITH NO IMOD PARAMETER;(0) - NAIFC [1.6] >0
C.2.18.2 .. UVAIFC BY I,J,K RANGE {0.1-0.3};(0) - NAIFC [1.6] > 0
C.2.18.3 .. IAIF;(0) - NAIFC [1.6] > 0
C.....TRANSIENT, CARTER-TRACY A.I.F.
C.2.18.4 .. KOAR,ABOAR,VISOAR,POROAR,BOAR,RIOAR,ANGOAR;(0) - IAIF [2.18.3] = 2
C.....     HEAT CONDUCTION B.C.
C.2.19.1 .. IBC BY I,J,K RANGE {0.1-0.3} ,WITH NO IMOD PARAMETER ,FOR HCBC NODES;(0) - HEAT [1.4] AND NHCBC [1.6] > 0
C.2.19.2 .. ZHCBC(K);(0) - HEAT [1.4] AND NHCBC [1.6] > 0
C.2.19.3 .. UDTHHC BY I,J,K RANGE {0.1-0.3} FOR HCBC NODES;(0) - HEAT [1.4]  AND NHCBC [1.6] > 0
C.2.19.4 .. UKHCBC BY I,J,K RANGE {0.1-0.3} FOR HCBC NODES;(0) - HEAT [1.4] AND NHCBC [1.6] >0
C.....FREE SURFACE B.C.
C.2.20 .. FRESUR(T/F),PRTCCM(T/F)
C.....INITIAL CONDITION INFORMATION
C.2.21.1 .. ICHYDP,ICT,ICC; ALL (T/F);IF NOT.HEAT, ICT = F, IF NOT.SOLUTE, ICC = F
C.2.21.2 .. ICHWT(T/F);(0) - FRESUR [2.20]
C.2.21.3A .. ZPINIT,PINIT;(0) - ICHYDP [2.21.1] AND NOT ICHWT [2.21.2]
C.2.21.3B .. P BY I,J,K RANGE {0.1-0.3};(0) - NOT ICHYDP [2.21.1] AND NOT ICHWT [2.21.2]
C.2.21.3C .. HWT BY I,J,K RANGE {0.1-0.3};(0) - FRESUR [2.20] AND ICHWT [2.21.2]
C.2.21.4A .. NZTPRO,ZT(I),TVD(I),I=1,NZTPRO;(0) - HEAT [1.4] AND NOT ICT [2.21.1], LIMIT OF 10
C.2.21.4B .. T BY I,J,K RANGE {0.1-0.3};(0) - HEAT [1.4] AND ICT [2.21.1]
C.2.21.5 .. NZTPHC, ZTHC(I),TVZHC(I);(0) - HEAT [1.4] AND NHCBC [1.6] > 0,LIMIT OF 5
C.2.21.6 .. C BY I,J,K RANGE {0.1-0.3};(0) - SOLUTE [1.4] AND ICC [2.21.1]
C.....CALCULATION INFORMATION
C.2.22.1 .. FDSMTH,FDTMTH
C.2.22.2 .. TOLDEN{.001},MAXITN{5}
C.2.22.3 .. NTSOPT{5},EPSSOR{.00001},EPSOMG{.2},MAXIT1{50},MAXIT2{100};(0) - SLMETH [1.8] = 2
C.....OUTPUT INFORMATION
C.2.23.1 .. PRTPMP,PRTFP,PRTIC,PRTBC,PRTSLM,PRTWEL; ALL (T/F)
C.2.23.2 .. IPRPTC,PRTDV(T/F);(0) - PRTIC [2.23.1]
C.2.23.3 .. ORENPR[I];(0) - NOT CYLIND [1.4]
C.2.23.4 .. PLTZON(T/F);(0) - PRTPMP [2.23.1]
```

Table 5.1.--*Data-input form*--Continued

```
C.2.23.5 .. OCPLOT(T/F)
C-----------------------------------------------------------------------------
C..... TRANSIENT DATA - READ3
C.3.1 .. THRU(T/F)
C.....IF THRU IS TRUE PROCEED TO RECORD 3.99
C.....THE FOLLOWING IS FOR NOT THRU
C.....SOURCE-SINK WELL INFORMATION
C.3.2.1 .. RDWFLO(T/F),RDWHD(T/F);(0) - NWEL [1.6] > 0
C.3.2.2 .. IWEL,QWV,PWSUR,PWKT,TWSRKT,CWKT;(0) - RDWFLO [3.2.1] OR RDWHD [3.2.1]
C.....USE AS MANY 3.2.2 LINES AS NECESSARY
C.3.2.3 .. END WITH 0 /
C.....BOUNDARY CONDITION INFORMATION
C.....     SPECIFIED VALUE B.C.
C.3.3.1 .. RDSPBC,RDSTBC,RDSCBC,ALL(T/F);(0) - NOT CYLIND [1.4] AND NPTCBC [1.6] > 0
C.3.3.2 .. PNP B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDSPBC [3.3.1]
C.3.3.3 .. TSBC BY I,J,K RANGE {0.1-0.3}; (0) - RDSPBC [3.3.1] AND HEAT [1.4]
C.3.3.4 .. CSBC BY I,J,K RANGE {0.1-0.3}; (0) - RDSPBC [3.3.1] AND SOLUTE [1.4]
C.3.3.5 .. TNP B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDSTBC [3.3.1] AND HEAT [1.4]
C.3.3.6 .. CNP B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDSCBC [3.3.1] AND SOLUTE [1.4]
C.....     SPECIFIED FLUX
C.3.4.1 .. RDFLXQ,RDFLXH,RDFLXS,ALL(T/F);(0) - NFBC [1.6] > 0
C.3.4.2 .. QFFX,QFFY,QFFZ B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDFLXQ [3.4.1]
C.3.4.3 .. UDENBC  BY I,J,K RANGE {0.1-0.3};(0) - RDFLXQ [3.4.1]
C.3.4.4 .. TFLX B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDFLXQ [3.4.1] AND HEAT [1.4]
C.3.4.5 .. CFLX B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDFLXQ [3.4.1] AND SOLUTE [1.4]
C.3.4.6 .. QHFX,QHFY,QHFZ B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDFLXH [3.4.5]
C.3.4.7 .. QSFX,QSFY,QSFZ B.C. BY I,J,K RANGE {0.1-0.3};(0) - RDFLXS [3.4.1]
C.....     LEAKAGE BOUNDARY
C.3.5.1 .. RDLBC(T/F);(0) - NLBC [1.6] > 0
C.3.5.2 .. PHILBC,DENLBC,VISLBC BY I,J,K RANGE {0.1-0.3};(0) - RDLBC [3.5.1]
C.3.5.3 .. TLBC BY I,J,K RANGE {0.1-0.3};(0) - RDLBC [3.5.1] AND HEAT [1.4]
C.3.5.4 .. CLBC BY I,J,K RANGE {0.1-0.3};(0) - RDLBC [3.5.1] AND SOLUTE [1.4]
C.....     RIVER LEAKAGE
C.3.5.5 .. I1,I2,J1,J2,HRBC,DENRBC,VISRBC,TRBC,CRBC;(0) - RDLBC [3.5.1]
C.....USE AS MANY 3.5.5 LINES AS NECESSARY
C.3.5.6 .. END WITH 0 /
C.....     A.I.F. B.C.
C.3.6.1 .. RDAIF(T/F); (0) - NAIFC [1.6] > 0
C.3.6.2 .. DENOAR BY I,J,K RANGE {0.1-0.3};(0) - RDAIF [3.6.1]
C.3.6.3 .. TAIF BY I,J,K RANGE {0.1-0.3};(0) - RDAIF [3.6.1] AND HEAT [1.4]
C.3.6.4 .. CAIF BY I,J,K RANGE {0.1-0.3};(0) - RDAIF [3.6.1] AND SOLUTE [1.4]
C.....CALCULATION INFORMATION
C.3.7.1 .. RDCALC(T/F)
C.3.7.2 .. AUTOTS(T/F);(0) - RDCALC [3.7.1]
C.3.7.3.A .. DELTIM;(0) - RDCALC [3.7.1] AND NOT AUTOTS [3.7.2]
```

Table 5.1.--*Data-input form*--Continued

```
C.3.7.3.B .. DPTAS{5E4},DTTAS{5.},DCTAS{.25},DTIMMN{1.E4},DTIMMX{1.E7};(0) - RDCALC [3.7.1] AND AUTOTS [3.7.2]
C.3.7.4 .. TIMCHG
C.....OUTPUT INFORMATION
C.3.8.1 .. PRIVEL,PRIDV,PRISLM,PRIKD,PRIPTC,PRIGFB,PRIWEL,PRIBCF; ALL [I]
C.3.8.2 .. IPRPTC;(0) - IF PRIPTC [3.8.1] NOT = 0
C.3.8.3 .. CHKPTD(T/F),NTSCHK,SAVLDO(T/F)
C.....CONTOUR MAP INFORMATION
C.3.9.1 .. RDMPDT,PRTMPD; ALL (T/F)
C.3.9.2 .. MAPPTC,PRIMAP[I];(0) - RDMPDT [3.9.1]
C.3.9.3 .. YPOSUP(T/F),ZPOSUP(T/F),LENAX,LENAY,LENAZ;(0) - RDMPDT [3.9.1]
C.3.9.4 .. IMAP1{1},IMAP2{NX},JMAP1{1},JMAP2{NY},KMAP1{1},KMAP2{NZ},AMIN,AMAX,NMPZON{5}:(0) - RDMPDT [3.9.1]
C.....ONE OF THE 3.9.4 LINES REQUIRED FOR EACH DEPENDENT VARIABLE
C.....    TO BE MAPPED
C.....END OF FIRST SET OF TRANSIENT INFORMATION
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.....READ SETS OF READ3 DATA AT EACH TIMCHG UNTIL THRU  (LINES 3.N1.N2)
C.....END OF CALCULATION LINES FOLLOW, THRU=.TRUE.
C.3.99.1 .. THRU
T
C.....TEMPORAL PLOT INFORMATION
C.3.99.2 .. PLOTWP,PLOTWT,PLOTWC; ALL (T/F)
C.....PLOT INFORMATION; (0) - PLOTWP [3.99] OR PLOTWT [3.99] OR PLOTWC [3.99]
C.4.1 ..  IWEL,RDPLTP(T/F)
C.4.2 ..  IDLAB
C.4.3 ..  NTHPTO,NTHPTC,PWMIN,PWMAX,PSMIN,PSMAX,TWMIN,TWMAX,TSMIN,TSMAX,CMIN,CMAX; (0) - RDPLTP [4.1]
C.4.4 ..  TO,POW,POS,TOW,TOS,COW
C.....USE AS MANY 4.4 LINES AS NECESSARY
C.4.5 ..  END WITH -1. /
C.....READ DATA FOR ADDITIONAL WELLS,  4.1-4.5 LINES
C.4.6 ..  END WITH 0 /
C.....END OF DATA FILE
```

A set of record pairs is needed where the first is:

I.1 I1, I2, J1, J2, K1, K2;

where Ii, Jj, Kk are inclusive node-index ranges in the i,j, and k directions, and the second is:

I.2 VAR1, IMOD1, [VAR2, IMOD2, VAR3, IMOD3];

where VARi is the value of the ith variable;

IMODi is the modification code for the ith variable;

1 means insert the data into the variable location;

2 means multiply the existing value of the variable by VARi;

3 means add VARi to the existing value of the variable;

4 means read in values of VARi on a node-by-node basis <u>in the order of increasing i, then j, then k</u>. In this case, the value of VARi in record I.2 is not used and needs to be given a value of zero. Note that this type of input is only suitable for zones of cells with a continuous sequence of nodes in the i-direction. Zones of cells whose i-direction index is a discontinuous sequence are more easily treated by using one of the first three types of modification code.

The brackets indicate that some input lists need three values of VAR and IMOD, while others need only one. As many lines of type I.1 and I.2 are used as necessary, then the following line is entered to terminate the input:

I.3 0 /.

The generic dimensions for the various parameters are indicated, and the units must be selected from table 5.2. unless the conversion factors in the HST3D program are modified. The units for output will match the units for input. The few exceptions, where the units are not combinations of the fundamental units, are indicated. The conversion factors from inch-pound units to metric units are contained in the following subprograms: READ1, ETOM1, ETOM2. They are specified in PARAMETER statements and the variable names are in the form, CNVxxx. The conversion factors from metric units to inch-pound units are defined in the INIT1 subprogram, and the variable names are in the form CNVxxI. The unit labels are contained in variables named UNITxx and are defined in the INIT1 subprogram also. By changing the appropriate conversion factors, and unit labels the user can employ the most convenient units for a particular simulation series.

The notation (T/F) indicates a logical variable. All index ranges are inclusive unless stated otherwise. All pressure values are relative to atmospheric unless stated otherwise.

Table 5.2.--*International System and inch-pound units used in the heat- and solute-transport simulator*

Quantity (generic units)	International System units	Inch-pound units
Mass (M)	kilogram (kg)	pound (lb)
Length (L)	meter (m)	foot (ft)
Time (t)	second (s)	day (d)
Temperature (T)	degree Celsius (°C)	degree Fahrenheit (°F)
Energy (E)	Joule (J) or watt-second (W-s)	British Thermal Unit (BTU)
Fluid Volumetric Flow Rate (L^3/t)	liter per second (ℓ/s)	cubic foot per day (ft^3/d)
Density (M/L^3)	kilogram per cubic meter (kg/m^3)	pound per cubic foot (lb/ft^3)[1]
Velocity (L/t)	meter per second (m/s)	foot per day (ft/d)
Pressure (F/L^2)	Pascal (Pa)	pound per square inch (psi)
Viscosity (M/Lt)	kilogram per meter-second (kg/m-s)	centipoise (cP)[2]
Diffusivity (L^2/t)	square meter per second (m^2/s)	square foot per day (ft^2/d)
Permeability (L^2)	square meter (m^2)	square foot (ft^2)
Thermal Conductivity (E/L-t-T) or (F/t-T)	Watt per meter-degree Celsius (W/m-°C)	British Thermal Unit per foot-hour-degree Fahrenheit (BTU/ft-h-°F)
Specific Heat Capacity (FL/M-T or E/M-T)	Joule per kilogram-degree Celsius (J/kg-°C)	British Thermal Unit per pound-degree Fahrenheit (BTU/lb-°F)

Table 5.2.--*International System and inch-pound units used in the heat- and solute-transport simulator*--Continued

Quantity (generic units)	International system units	Inch-pound units
Heat-Transfer Coefficient ($E/t\text{-}L^2\text{-}T$)	Watt per square meter-degree Celsius ($W/m^2\text{-}^\circ C$)	British Thermal Unit per hour-square foot-degree Fahrenheit ($BTU/h\text{-}ft^2\text{-}^\circ F$)
Volumetric flux ($L^3/L^2\text{-}t$)	cubic meter per square meter-second ($m^3/m^2\text{-}s$)	cubic foot per square foot-day ($ft^3/ft^2\text{-}d$)
Heat flux ($E/L^2\text{-}t$)	Watt per square meter (W/m^2)	British Thermal Unit per square foot-hour ($BTU/ft^2\text{-}h$)
Mass flux ($M/L^2\text{-}t$)	kilogram per square meter-second ($kg/m^2\text{-}s$)	pound per square foot-day ($lb/ft^2\text{-}d$)

[1]A weight density rather than a mass density.
[2]Not Inch-Pound but common usage.

5.2.2. Input Record Descriptions

The following order generally has been observed for data input: (1) Fundamental and dimensioning information, (2) spatial geometry and mesh information, (3) fluid properties, (4) porous medium properties, (5) source information, (6) boundary condition information, (7) initial condition information, (8) calculation parameters, (9) output specifications. Items 5, 6, 8 and 9 have transient data associated with them, and data are input in the item-order given. The static data are read only once while the transient data are read at each time a change in the data is to occur. Only the data that are being changed need to be entered, because any unmodified data will remain the same over the next time interval of simulation. Each input record number identifies a particular record in the input-data form listed in table 5.1.

Dimensioning data for space allocation - READ1

1.1 Title line 1.

A character string of up to 80 characters. It is wise to start in column two in case the string begins with the letter 'c'.

1.2 Title line 2.

A character string of up to 80 characters. It is wise to start in column two in case the string begins with the letter 'c'.

1.3 RESTRT (T/F), TIMRST.

RESTRT - True if this run is to be a restart of a previous simulation.

TIMRST - Time within the range of a previous simulation from which this restart is to continue (t). A search is made of the restart-record file for data pertaining to the time value specified or within 0.1 percent of that time. Enter zero if this is not a restart of a previous simulation.

If this is a restart run, proceed to the READ3 group for input of transient data.

1.4 HEAT, SOLUTE, EEUNIT, CYLIND, SCALMF; all (T/F).

HEAT - True if heat transport is to be simulated.

SOLUTE - True if solute transport is to be simulated.

EEUNIT - True if the input data are in inch-pound units; otherwise, metric units are assumed. The conversion factors are set for the inch-pound and metric units given in table 5.1 and appear after the table of contents also. The program uses metric units internally.

CYLIND - True if cylindrical coordinates with no angular dependence are to be used; otherwise, cartesian, x,y,z, coordinates are used.

SCALMF - True if a scaled-mass fraction determined by equation 2.2.1.2 is to be used for data input and output; otherwise, the unscaled mass fraction will be used.

1.5 NX, NY, NZ, NHCN.

NX - The number of nodes in the x-direction for cartesian coordinates, or r-direction for cylindrical coordinates.

NY - The number of nodes in the y-direction. Unused for cylindrical coordinates, but a space in the input record must be included.

NZ - Number of nodes in the z-direction.

NHCN - Number of nodes for the heat-conduction boundary condition. This may be included only in a heat-transport simulation. These nodes are used for the finite-difference solution of equations 2.5.5.3a-d and 2.5.5.4a-d.

1.6 NPTCBC, NFBC, NAIFC, NLBC, NHCBC, NWEL.

NPTCBC - Number of cells (nodes) with a specified pressure, temperature, and(or) mass-fraction boundary condition.

NFBC - Number of cells with a specified-flux boundary condition; flow, heat, and(or) solute.

NAIFC - Number of cell faces with an aquifer-influence-boundary condition.

NLBC - Number of cell faces with a leakage-boundary condition.

NHCBC - Number of cell faces with a heat-conduction boundary condition.

NWEL - Number of wells in the simulation region.

These values must be the <u>exact counts</u> for the simulation. A cell with more than one face on the region boundary may be counted more than once for the above counts. A specified-value boundary-condition cell may <u>not</u> have any other type of boundary condition.

1.7 NPMZ.

NPMZ - Number of porous medium zones in the region. Porous medium hydraulic, thermal-, and solute-transport properties vary by zone.

1.8 SLMETH, LCROSD (T/F).

SLMETH - Solution method (integer):
Enter 1 to select the direct, D4, matrix-equation solver; or
Enter 2 to select the iterative, two-line, successive-overrelaxation, matrix-equation solver.

LCROSD - True if the off-diagonal cross-dispersive terms in the heat and solute equations are to be lumped into the diagonal terms to give amplified coefficients. Otherwise, the coefficients in the dispersion tensor are computed explicitly in time. Lumping the coefficients reduces storage requirements and the number of iterative cycles of the flow, heat, and solute equations, but it is an empirical simplification.

1.9 IBC by i,j,k range for excluded cells.

IBC - Index of cells excluded from the simulation region. For space allocation, the location of the excluded cells must be known. Excluded cells are denoted by IBC = -1.

1.10 RDECHO (T/F).

RDECHO - True if a file is to be written that echos the input data as they are read that is used for locating data-input errors.

Static data - READ2.

The following data are invariant throughout the simulation.

2.1 PRTRE (T/F).

PRTRE - True if a read-echo printout of data input by i,j,k range is desired.

The following 12 records describe the gridding of the simulation region.

2.2A.1 UNIGRX, UNIGRY, UNIGRZ; All (T/F); optional, used for cartesian-coordinate systems.

UNIGRi - True if the grid in the ith direction is uniform.

2.2A.2A X(1),X(NX); Optional, used if uniform grid in the x-direction.

X(1) - Location of the first node point in the x-direction (L).
X(NX) - Location of the last node point in the x-direction (L).

2.2A.2B X(I), I=1 to NX; Optional, used for nonuniform grid in the x-direction.

2.2A.3A Y(1), Y(NY); Optional, used if uniform grid in the y-direction.

Y(1) - Location of the first node point in the y-direction (L).
Y(NY) - Location of the last node point in the y-direction (L).

2.2A.3B Y(J) J=1 to NY; Optional, used for nonuniform grid in the y-direction.

2.2A.4A Z(1), Z(NZ); Optional, used for uniform grid in the z-direction.

Z(1) - Location of the first node point in the z-direction (L).
Z(NZ) - Location of the last node point in the z-direction (L).

2.2A.4B Z(K), K=1 to NZ; Optional, used for nonuniform grid in the z-direction.

2.2B.1A R(1), R(NR), ARGRID (T/F); Optional, used for cylindrical-coordinate systems.

R(1) - Interior radius of cylindrical-coordinate system (L). Required by cylindrical-coordinate system.

R(NR) - Exterior radius of cylindrical-coordinate system (L). Required if a cylindrical-coordinate system.

ARGRID - True if automatic gridding or node location in the r-direction is desired. A logarithmic spacing in R will be used according to equation 3.1.2.1.

2.2B.1B R(I), I = 1 to NX; Optional, used for user-specified radial gridding for a cylindrical-coordinate system.

R(I) - Array of node locations along the r-axis (L). Required by cylindrical-coordinate system if automatic gridding is not selected.

2.2B.2 UNIGRZ (T/F).

2.2B.3A Z(1), Z(NZ); Optional, used if uniform grid in the z-direction for cylindrical coordinates.

2.2B.3B Z(K), K = 1 to NZ; Optional, used for nonuniform grid in the z-direction for cylindrical coordinates.

Z(K) - Array of node locations along the z-axis (L).

The following two records describe a tilted-coordinate system.

2.3.1 TILT (T/F); Optional, required only if a cartesian-coordinate system is used.

TILT - True if a tilted-coordinate system is desired with the z-axis not in the vertical upward direction.

No tilt may be specified if a free-surface boundary condition is to be employed (Record 2.20).

2.3.2 THETXZ, THETYZ, THETZZ; Optional, required only if a tilted-coordinate system is being used.

THETXZ - Angle that the x-axis makes with the vertical upward direction (DEG).
THETYZ - Angle that the y-axis makes with the vertical upward direction (DEG).
THETZZ - Angle that the z-axis makes with the vertical upward direction (DEG).

Fluid properties

2.4.1 BP.

BP - Compressibility of the fluid $(F/L^2)^{-1}$.

Fluid density data

2.4.2 PRDEN, TRDEN, WO, DENFO.

PRDEN - Reference pressure for density (relative to atmospheric) (eq. 2.2.1.1b) (F/L^2).
TRDEN - Reference temperature for density (eq. 2.2.1.1b) (T).
WO - Reference mass fraction for density (eq. 2.2.1.1b) and minimum-mass fraction for scaling (eq. 2.2.1.2) and $\rho_o\beta_w$ term (eq. 2.2.1.1c) (-). Needs to be zero if solute decay takes place. Should be zero if solute transport is not being simulated.
DENFO - Fluid density at the minimum solute mass fraction (eq. 2.2.1.1c or eq. 2.2.1.3b) (M/L^3). (kg/m^3) or (lb/ft^3).

2.4.3 W1, DENF1; Optional, required only if a solute transport is being simulated.

W1 - Maximum-mass fraction for scaling (eq. 2.2.1.2) and $\rho_o\beta_w$ term (eq. 2.2.1.1c) (-). Should be equal to or greater than the maximum mass-fraction value specified by any boundary condition or initial condition.

DENF1 - Fluid density at the maximum solute mass fraction (eq. 2.2.1.1c or eq. 2.2.1.3b) (M/L^3) (kg/m^3) or (lb/ft^3).

If solute transport is being simulated and a scaled-mass fraction has been selected for input and output (SCALMF in record 1.4), W0 and W1 are used to perform the scaling according to equation 2.2.1.2 and slope calculation of equation 2.2.1.3b.

Fluid-viscosity data

2.5.1 NOTV0, (TVF0(I), VISTF0(I), I = 1 to NOTV0); Optional, required if only flow; or flow and heat transport; or flow, heat and solute transport are being simulated. Not used if only flow and solute transport are being simulated.

NOTV0 - Number of viscosity versus temperature points for fluid at minimum or reference solute-mass fraction (minimum of one point).

TVF0 - Array of temperature points (T).

VISTF0 - Array of viscosity points at minimum solute-mass fraction, W0, (M/Lt) (kg/m-s) or (cP).

2.5.2 NOTV1 (TVF1(I), VISTF1(I) I = 1 to NOTV1); Optional, required only if heat and solute transport are being simulated.

NOTV1 - Number of viscosity versus temperature points for fluid at maximum solute mass fraction (minimum of one point).

TVF1 - Array of temperature points (T).

VISTF1 - Array of viscosity points at maximum solute-mass fraction, W1, (M/Lt), (kg/m-s) or (cP).

2.5.3 NOCV, TRVIS, (CVIS(I), VISCTR(I), I = 1 to NOCV); Optional, required only if solute transport is being simulated.

NOCV - Number of viscosity versus mass-fraction points (minimum of two points).

TRVIS - Reference temperature for viscosity versus mass-fraction data (T).

CVIS - Array of mass-fraction (or scaled-mass-fraction) points (-).

VISCTR - Array of viscosity points for fluid at reference temperature (M/Lt), (kg/m-s) or (cP).

If only solute transport is being simulated, only record number 2.5.3 is needed from this group.

Reference condition information

2.6.1 PAATM.

PAATM - Atmospheric absolute-pressure value used to relate gage pressure to absolute pressure (F/L^2).

If zero is entered, standard atmospheric pressure of 1.01325×10^5 Pa is used.

2.6.2 POH, TOH.

POH - Reference pressure (relative to atmospheric) for enthalpy variations, P_{oH}, (F/L^2). This value should be within the range of pressure to be simulated.

TOH - Reference temperature for enthalpy variations or the constant temperature for isothermal simulations, T_{oH} (T). This value should be within the range of temperature to be simulated.

These values are used in equation 2.2.3.1c.

Fluid thermal properties

2.7 CPF, KTHF, BT; Optional, required only if heat transport is being simulated.

CPF - Fluid heat capacity at constant pressure (FL/M-T). An average value for the range of solute concentration and temperature to be simulated should be used (eq. 2.2.3.1b and eq. 2.2.3.1c).

KHTF - Fluid thermal conductivity (F/t-T). An average value for the range of solute concentration and temperature to be simulated should be used.

BT - Fluid coefficient of thermal expansion (T^{-1}). An average value for the range of solute concentration and temperature to be simulated should be used.

Solute information

2.8.1 DM, DECLAM; Optional, required only if solute transport is being simulated.

DM - Effective molecular diffusivity for the solute in the porous media (L^2/t).

DECLAM - Solute-decay-rate constant (t^{-1}).

The following two records describe the zonation of the simulation region.

2.9.1 IPMZ, I1Z(IPMZ) I2Z(IPMZ),J1Z(IPMZ),J2Z(IPMZ),K1Z(IPMZ),K2Z(IPMZ).

IMPZ - Porous-medium zone number.

Records of this form define the zones within the simulation region and assign zone numbers. These zones are used to assign values to the porous-medium properties. The ranges of the indices in the I, J, and K directions define the rectangular prism for a given zone. The zones must be convex and non-overlapping as explained in section 3.1 and the entire simulation region must be covered by the set of zones. No zones may be defined that include elements outside the simulation region. For cylindrical coordinates, J1Z and J2Z must equal 1. The subscript IPMZ identifies data that are input by zone number in subsequent records. The number of 2.9.1 records must equal NPMZ.

2.9.2 End the input with 0 / .

Porous-media properties

2.10.1 KXX(IPMZ), KYY(IPMZ), KZZ(IPMZ); IPMZ = 1 to NPMZ.

KXX - Permeability in the x-direction or r-direction for zone IPMZ (L^2).
KYY - Permeability in the y-direction for zone IPMZ (L^2). Not used for cylindrical coordinates, but a zero or blank space must be indicated in the input record.
KZZ - Permeability in the z-direction for zone IPMZ (L^2).

2.10.2 POROS(IPMZ), IPMZ = 1 to NPMZ.

POROS - Porosity of the medium in zone IPMZ (-).

2.10.3 ABPM(IPMZ), IPMZ = 1 to NPMZ.

ABPM - Porous-medium bulk vertical compressibility in zone IPMZ $(F/L^2)^{-1}$. This compressibility is determined on a fixed mass of porous medium undergoing vertical compression.

Porous-media thermal properties

2.11.1 RCPPM(IPMZ), IPMZ = 1 to NPMZ; Optional, required only if heat transport is being simulated.

RCPPM - Heat capacity of the porous-medium solid phase per unit volume for zone IPMZ (F/L^2-T).

2.11.2 KTXPM(IPMZ), KTYPM(IPMZ), KTZPM(IPMZ), IPMZ = 1 to NPMZ; Optional, required only if heat transport is being simulated.

KTXPM - Thermal conductivity of the porous medium in the x-direction for zone IPMZ (F/t-T).

KTYPM - Thermal conductivity of the porous medium in the y-direction for zone IPMZ (F/t-T). Not used if cylindrical coordinates, but a zero or a blank space in the input file must be denoted.

KTZPM - Thermal conductivity of the porous medium in the z-direction for zone IPMZ (F/t-T).

Solute and thermal dispersion information

2.12 ALPHL(IPMZ), ALPHT(IPMZ), IPMZ = 1 to NPMZ; Optional, required only if heat or solute transport is being simulated.

ALPHL - Longitudinal dispersivity for zone IPMZ (L).

ALPHT - Transverse dispersivity for zone IPMZ (L).

Solute sorption information

2.13 DBKD(IPMZ) IPMZ = 1 to NPMZ; Optional, required only if solute transport is being simulated.

DBKD - Dimensionless linear-equilibrium-distribution coefficient. This is the porous-medium bulk density times the distribution coefficient (-).

Well-model information

2.14.1 RDWDEF (T/F); Optional, required only if there are wells in the simulation region.

RDWDEF - True if well definition data are to be read.

2.14.2 IMPQW (T/F); Optional, required only if there are wells in the region and cartesian coordinates are being used.

IMPQW - True if semi-implicit well-flow calculations are to be made. Otherwise, well-flow rates are calculated explicitly at the beginning of the time step.

Record lines 2.14.3-6 are repeated as often as necessary for each well in the simulation. Each well must be defined at this point, whether or not it is active at the start of the simulation.

2.14.3 IWEL, IW, JW, LCBOTW, LCTOPW, WBOD, WQMETH; Optional, required only if there are wells in the simulation region and well-definition data are to be read.

IWEL - Well number.

IW - Cell number in x-direction of well location.

JW - Cell number in y-direction of well location. Not used in cylindrical coordinates, but a zero or a blank entry must be included.

LCBOTW - Cell number in z-direction of lowermost completion layer for the well.

LCTOPW - Cell number in z-direction of uppermost completion layer for the well. This is the location at which the dependent-variable data are taken if this is an observation well.

WBOD - Well bore outside diameter. This is the drilled diameter or diameter of the screen or perforated casing (L).

WQMETH - Index for well-flow calculation method (integer).

10 - Specified well-flow rate with allocation by mobility and pressure difference.

11 - Specified well-flow rate with allocation by mobility.

20 - Specified pressure at well datum with allocation by mobility and pressure difference.

30 - Specified well-flow rate with a limiting pressure at well datum. Flow-rate allocation by mobility and pressure difference.

40 - Specified surface pressure with allocation by mobility and pressure difference. Well-riser calculations will be performed.

50 - Specified surface-flow rate with limiting surface pressure. Allocation by mobility and pressure difference. Well-riser calculations will be performed.

0 - Observation well or abandoned well.

2.14.4 WCF(L), L = 1 to NZ-1; Optional, required only if there are wells in the simulation region and well-definition data are to be read.

WCF - Well-completion-factor array on an element-by-element basis (-). Element L goes from the node at z-level L to the node at z-level L+1. An element completion factor of one means the geometric-mean of the horizontal permeability for that element will be used in the well index. An element completion factor of zero means the well is cased off from the aquifer in that element. A reduced permeability around the well bore can be approximately represented by specifying a completion factor less than one. Note that this is not the same as a well skin factor, which is not included in the present version of the program. If WCF is zero for the element between LCBOTW and LCBOTW+1 or for the element between LCTOPW-1 and LCTOPW, then LCBOTW or LCTOPW should be adjusted as necessary to range from the bottom to the top completion layer that communicate with the aquifer. The well-completion factor also can be used to compute an approximate effective permeability for a well that is completed in a cell that contains multiple zones of different permeability.

2.14.5 WRISL, WRID, WRRUF, WRANGL; Optional, required only if well-definition data are to be read and well-riser calculations are to be made (WQMETH is 40 or 50).

WRISL - Well riser-pipe length (L).

WRID - Well riser-pipe inside diameter (L).

WRRUF - Well riser-pipe roughness factor (L) (see table 2.1).

WRANGL - Well riser-pipe angle with vertical direction (DEG).

2.14.6 HTCWR, DTHAWR, KTHAWR, KTHWR, TABWR, TATWR; Optional, required only for heat-transport simulations, and if well-riser calculations are to be done.

HTCWR - Heat-transfer coefficient from the fluid to the well-riser pipe (E/tL^2T).

DTHAWR - Thermal diffusivity of the medium adjacent to the well-riser pipe (L^2/t).

KTHAWR - Thermal conductivity of the medium adjacent to the well-riser pipe (F/t-T).

KTHWR - Thermal conductivity of the well-riser pipe (F/t-T).

TABWR - Ambient temperature at the bottom of the well-riser pipe (T).

TATWR - Ambient temperature at the top of the well-riser pipe (T).

2.14.7 End this data set with 0 /.

2.14.8 MXITQW, TOLDPW, TOLFPW, TOLQW, DAMWRC, DZMIN, EPSWR; Optional, required only if wells are in the region, and if well-riser calculations are to be done.

MXITQW - Maximum number of iterations allowed for the well-flow rate allocation calculation. Default of 20.

TOLDPW - Tolerance on the change in well-riser pressure for the well-riser iterative calculation (F/L^2). Default of 6×10^{-3} Pa. This is the primary convergence test.

TOLFPW - Tolerance on the fractional change in well-riser pressure for the well-riser iterative calculation. Default of 0.001. This is the secondary convergence test.

TOLQW - Tolerance on the fractional change in well flow rate because of temperature and mass-fraction changes for the source term in the flow equation. Default of 0.001. This is the tertiary convergence test.

DAMWRC - Damping factor for well-pressure adjustment during the iterations for allocating flow rates. Default of 2.

DZMIN - Minimum value of step length along the well riser (L). Default of 1 percent of riser length.

EPSWR - Fractional tolerance for the integration of the pressure and temperature equations along the well riser. Default of 0.001.

Boundary-condition information

Specified value

2.15 IBC by i,j,k range; Optional, required only if specified value boundary-condition cells are present.

IBC - Index of boundary-condition type. This is in the form $n_1n_2n_3$ where n_1 refers to pressure; n_2 refers to temperature; and n_3 refers to mass fraction. The value for n_i is set to 1 to indicate that a specified value boundary condition for variable i exists at that cell.

Remember that a specified value for cell removes that cell from the calculation. Therefore, no other boundary conditions at that cell can be specified for the equation concerned.

Specified flux

2.16 IBC by i,j,k range; Optional, required only if specified flux boundary-condition cells are present.

IBC - Index of boundary-condition type. This is in the form $n_100n_4n_5n_6$ where n_1=1,2,3, meaning that the flux is through the x,y, or z boundary face respectively. Values for n_4, n_5, and n_6 are set to

2 to indicate that a specified flux boundary condition, for a chosen equation, is required at that cell, where n_4 refers to the flow equation, n_5 refers to the heat-transport equation, and n_6 refers to the solute-transport equation.

Aquifer leakage

2.17.1 IBC by i,j,k range for aquifer-leakage boundary cells; Optional, required only if leakage boundary conditions are being used.

IBC - Index of boundary-condition type. It is of the form $n_1$00300 for aquifer leakage, where n_1 indicates the direction of the normal to the leakage boundary face. Values for n_1 are 1 for the x-direction; 2 for the y-direction; and 3 for the z-direction. The number 3 in the hundreds place denotes an aquifer-leakage boundary condition.

2.17.2 KLBC, BBLBC, ZELBC by i,j,k range; Optional, required only if leakage boundary conditions are being used.

KLBC - Permeability of confining layer for aquifer-leakage boundary condition (L^2). Appears in equations 2.5.3.1a and 3.4.3.1a.

BBLBC - Confining-layer thickness for leakage boundary condition (L).

ZELBC - Elevation of the far side of the confining layer away from the simulation region (L).

For leakage across lateral boundaries of the simulation region, ZELBC is automatically set equal to the elevation of the corresponding boundary node.

River leakage

2.17.3 I1,I2,J1,J2, KRBC, BBRBC, ZERBC; Optional, required only if the river-leakage boundary condition is being used.

I1,I2,J1,J2 - Node or cell number ranges in the x and y directions for a river-leakage boundary-condition segment. River segments are lines in the x-, y-, or diagonal direction.

KRBC - Permeability multiplied by effective-riverbed-area factor for river-leakage boundary-condition (L^2). The effective-riverbed-area factor is the ratio of the riverbed area to the boundary-face area for a given cell.

BBRBC - Thickness of the confining layer that forms the riverbed (L).

ZERBC - Elevation of the top of the confining layer that forms the riverbed defined in figure 2.4 (L).

Use as many 2.17.3 records as necessary to describe the river.

2.17.4 End this data set (records 2.17.3) with 0 /.

Aquifer-influence functions

2.18.1 IBC by i,j,k range for aquifer-influence-function boundary cells; Optional required only if aquifer-influence-function boundary conditions are being used.

IBC - Index of boundary-condition type. It is in the form $n_1$00400 for aquifer-influence functions, where n_1 indicates the direction of the normal to the influence-function-boundary face. Values for n_1 are 1 for the x-direction; 2 for the y-direction; and 3 for the z-direction. The number 4 in the hundreds place denotes an aquifer-influence-function boundary condition.

2.18.2 UVAIFC by i,j,k range; Optional, required only if aquifer-influence functions are used.

UVAIFC - Temporary storage for input of user-specified factors for aquifer-influence-function spatial allocation.

These factors are defined on a zonal basis, so the factor for all boundary zones which include a common cell for which an aquifer-influence-function boundary condition applies must be the same. If default weighting of the aquifer-influence functions in proportion to their boundary-cell facial area is desired, no values for UVAIBC need to be entered (except the closing 0 /).

2.18.3 IAIF; Optional, required only if aquifer-influence functions are being used.

IAIF - Index of aquifer-influence function.

1 - Pot aquifer for outer-aquifer region.

2 - Transient-aquifer-influence function with calculation using the Carter-Tracy approximation.

2.18.4 KOAR, ABOAR, VISOAR, POROAR, BOAR, RIOAR, ANGOAR; Optional, required only if transient-aquifer-influence functions are being used.

KOAR - Permeability for the outer-aquifer region (L^2).

ABOAR - Porous-medium bulk vertical compressibility for the outer-aquifer region $(F/L^2)^{-1}$.

VISOAR - Viscosity of the fluid in the outer-aquifer region (M/Lt), (kg/m-s) or (cP).

POROAR - Porosity for the outer-aquifer region (-).

BOAR - Total thickness of the outer-aquifer region (L).

RIOAR - Radius of the equivalent cylinder that contains the inner-aquifer region (L). Usually determined by equation 3.4.4.2.1.

ANGOAR - Angle of influence of the outer-aquifer region (DEG.). This is the angle subtended by the part of the equivalent cylindrical boundary that is subject to flux determined by the aquifer-influence function.

The following three records describe the gridding and parameters for heat-conduction boundary conditions.

2.19.1 IBC by i,j,k range for heat-conduction boundary-condition nodes; Optional, required only if a heat simulation, and if heat-conduction boundary condition cells are to be used.

IBC - Index of boundary-condition type, denoted by a number in the form $n_1$00040 where n_1 indicates the direction of the outward normal to the heat-conduction boundary-condition cell face. The values for n_1 are: 1 for the x-direction; 2 for the y-direction; and 3 for the z-direction. The signs of the normal is disregarded. The 4 indicates that this cell has a heat-conduction boundary condition on one of its faces. Only one face of a given cell can be assigned a heat-conduction boundary condition.

2.19.2 ZHCBC(K) K = 1 to NHCN; Optional, required if heat simulation is being done, and there are heat-conduction boundaries (NHCBC > 0).

ZHCBC - Array of node distances along the outward pointing normal from a heat-conduction boundary surface (L). The first value must be zero. All heat-conduction boundary-condition cells use the same nodal distribution.

2.19.3 UDTHHC by i,j,k range for heat-conduction boundary-condition cells; Optional, required only if a heat simulation, and if there are heat-conduction boundary-condition cells.

UDTHHC - Temporary storage for input of thermal diffusivity of the heat-conducting medium outside the simulation region as a function of boundary-cell location (L^2/t).

2.19.4 UKHCBC by i,j,k range for heat-conduction boundary-condition cells; Optional, required only if a heat simulation, and if there are heat-conduction boundary-condition cells.

UKHCBC - Temporary storage for input of thermal conductivity of the heat-conducting medium outside the simulation region as a function of boundary-cell location (E/L-t-T).

Free-surface boundary condition

2.20 FRESUR (T/F), PRTCCM (T/F).

FRESUR - True if the region is unconfined, so that a free-surface boundary exists.

PRTCCM - True if a message is to be printed when a free surface rises above the top of a cell or falls below the bottom of a cell, or if a cell below the uppermost layer becomes unsaturated.

Initial conditions

2.21.1 ICHYDP (T/F), ICT (T/F), ICC (T/F).

ICHYDP - True if initial condition of hydrostatic pressure distribution is to be specified.

ICT - True if an initial-condition temperature distribution is to be specified.

ICC - True if an initial-condition mass-fraction distribution is to be specified.

2.21.2 ICHWT (T/F); Optional, required only if a free-surface boundary exists.

ICHWT - True if an initial-condition water-table-elevation distribution is to be input.

2.21.3A ZPINIT, PINIT; Optional, required only if an initial-condition hydrostatic-pressure distribution is being specified.

ZPINIT - Elevation of the initial-condition pressure (L).

PINIT - Pressure for hydrostatic, initial-condition distribution (F/L^2).

2.21.3B P by i,j,k range; Optional, required only if a non-hydrostatic pressure distribution is being specified as an initial condition.

P - Pressure distribution for the initial condition (F/L^2).

2.21.3C HWT by i,j,k range; Optional, required only if desired in conjunction with a free-surface boundary condition.

HWT - Water-table-elevation distribution for the initial condition (L). Specified for the upper layer of cells only.

2.21.4A NZTPRO (ZT(I), TVD(I); I = 1 to NZTPRO); Optional, required only if a heat simulation is being done.

NZTPRO - Number of points in the temperature-versus-depth profile for initial-condition temperature distribution. Limit of 10.

ZT - Array of locations along the z-axis for initial-temperature distribution (L). These locations must span the entire z-axis range of the region.

TVD - Array of initial temperatures along the z-axis (T).

2.21.4B T by i,j,k range; Optional, required only if ICT is true.

T - Temperature distribution for the initial condition (T).

2.21.5 NZTPHC, ZTHC(I), TVZHC(I), I = 1 to NZTPHC; Optional, required only if a heat-transport simulation is being done, and if there are heat-conduction boundary conditions.

NZTPHC - Number of points in the outward normal direction to the heat-conduction boundary-condition surfaces for initial-condition-temperature profile. Limit of 5.

ZTHC - Array of node locations in the outward normal direction for initial-condition-temperature profile for heat-conduction boundary-condition cell faces (L). The first value must be zero, and these nodes must span the mesh defined by ZHCBC in record 2.19.2.

TVZHC - Array for the initial-condition temperature-profile values for heat-conduction boundary-condition cell faces (T).

The same initial-condition profile is used for each heat-conduction boundary condition cell.

2.21.6 C by i,j,k range; Optional, required only if ICC is true.

C - Mass-fraction (or scaled mass fraction) distribution for the initial condition (-).

Calculation information

2.22.1 FDSMTH, FDTMTH.

FDSMTH - Factor for spatial-discretization method.
- 0.5 - centered-in-space differencing used for advective terms.
- 0 - upstream differencing in space used for advective terms.

FDTMTH - Factor for temporal-discretization method.
- 0.5 - centered-in-time or Crank-Nicholson differencing used.
- 1. - backward-in-time or fully-implicit differencing used.

2.22.2 TOLDEN, MAXITN.

TOLDEN - Tolerance in fractional change in density for convergence over a solution cycle of flow, heat, and solute equations at a given time plane. Default set at 0.001.

MAXITN - Maximum number of iterations allowed for a cycle of pressure, temperature, and mass-fraction solutions allowed at a given time plane. Default set at 5.

2.22.3 NTSOPT, EPSSOR, EPSOMG, MAXIT1, MAXIT2; Optional, required only if the two-line, successive-overrelaxation method for solving the system matrix equations is selected.

NTSOPT - Number of time steps between recalculations of the optimum-overrelaxation parameter. Default set at 5.

EPSSOR - Tolerance for the two-line, successive-overrelaxation iterative solution of the matrix equations at each time plane. Default set at 1×10^{-5}. The maximum fractional change in any of the values of the dependent variable must be less than or equal to this tolerance times $(2-\omega_{opt})$.

EPSOMG - Tolerance on the fractional change in the overrelaxation parameter during the iterative calculation to determine the optimum value. Default set to 0.2.

MAXIT1 - Maximum number of iterations allowed for the calculation of the optimum overrelaxation parameter. Default set at 50.

MAXIT2 - Maximum number of iterations allowed for the solution of the matrix equations. Default set at 100.

Output of static data

2.23.1 PRTPMP (T/F), PRTFP (T/F), PRTIC (T/F), PRTBC (T/F), PRTSLM (T/F), PRTWEL (T/F).

PRTPMP - True if a printout of porous-media properties is desired.

PRTFP - True if a printout of fluid properties is desired.

PRTIC - True if a printout of initial conditions is desired.

PRTBC - True if a printout of static boundary-condition information is desired.

PRTSLM - True if a printout of solution-method information is desired.

PRTWEL - True if a printout of static-well bore information is desired.

2.23.2 IPRPTC, PRTDV (T/F); Optional, required only if initial-condition printouts of the dependent variables are desired.

IPRPTC - Index of printout for initial-condition information. It is of the form $n_1n_2n_3$, where n_i is set to 1 for printout of the ith variable, otherwise n_i is set to 0. The variables are n_1 for pressure; n_2 for temperature; and n_3 for mass fraction. In addition, n_1 is set to 2 for both pressure and potentiometric head to be printed for isothermal cases; n_2 is set to 2 for both temperature and fluid enthalpy to be printed.

PRTDV - True if a printout of the density and viscosity arrays is desired.

2.23.3 ORENPR; Optional, required only for a cartesian-coordinate system.

ORENPR - Index for orientation of the array printouts (integer);

12 - Means x-y printouts for each plane along the z-axis, areal layers.

13 - Means x-z (or r-z) printouts for each plane along the y-axis, vertical slices.

A negative value means the y or z-axis is positive down the page.

2.23.4 PLTZON (T/F); Optional, required only if printout of porous-media properties has been requested.

PLTZON - True if a line-printer plot of the porous-media property zones is desired.

2.23.5 OCPLOT (T/F).

OCPLOT - True if plots of observed and calculated values of the dependent variables are to be plotted versus time at the end of the simulation.

Transient data - READ3

Groups of transient data are read by subroutine READ3; one at the beginning and others during the simulation, as necessary, whenever sources, boundary conditions, calculation parameters, or output options are to be changed. Only the parameters that are to be changed need to be input. The remaining parameters will keep their previous values.

3.1 THRU (T/F).

THRU - True if the simulation is through, and the closing procedures can begin. Proceed to record 3.99 if the simulation is finished.

Well information

3.2.1 RDWFLO (T/F), RDWHD (T/F); Optional, required only if there are wells in the simulation region.

RDWFLO - True if well-flow-rate data is to be read at this time.
RDWHD - True if well-head data is to be read at this time.

3.2.2 IWEL, QWV, PWSUR, PWKT, TWSRKT, CWKT; Optional, required only if well-flow or well-head data are to be read at this time.

IWEL - Well number.
QWV - Volumetric flow rate for this well (L^3/t), (ℓ/s) or (ft^3/d).
PWSUR - Pressure at the land surface for this well (F/L^2). Used when surface conditions are specified and the well-riser calculation is to be done.
PWKT - Pressure at the well datum for this well (F/L^2). Used when well-datum conditions are specified, and no well-riser calculation is to be done.
TWSRKT - Fluid temperature at the land surface or well datum for this well (T). Used when surface conditions are specified for an injection well, and used for the well-datum value, when well-datum conditions are specified.

CWKT - Mass fraction (or scaled-mass fraction) at the well datum for this well (-). Surface and well-datum concentrations are equal, so this variable also is used to specify surface conditions for an injection well.

As many records of type 3.2.2 are used as necessary to define conditions at all the wells. Data do not have to be input for any well that does not have its conditions changed at this time.

3.2.3 End this data set with 0 /.

Boundary-condition information

Specified value

3.3.1 RDSPBC (T/F), RDSTBC (T/F), RDSCBC (T/F); Optional, required only if there are specified-pressure, temperature or mass-fraction boundary-condition cells.

RDSPBC - True if specified-pressure boundary-condition data are to be read at this time.

RDSTBC - True if specified-temperature boundary-condition data are to be read at this time.

RDSCBC - True if specified mass-fraction boundary-condition data are to be read at this time.

3.3.2 PNP by i,j,k range; Optional, required only if specified-pressure boundary-condition values are to be input.

PNP - Pressure at specified-pressure boundary-condition nodes (F/L^2).

3.3.3 TSBC by i,j,k range; Optional, required only if specified-pressure boundary-condition values are to be read, and if a heat-transport simulation is being done.

TSBC - Temperature associated with a specified-pressure boundary condition node (T). If inflow occurs, this temperature will determine the heat-inflow rate.

3.3.4 CSBC by i,j,k range; Optional, required only if specified-pressure boundary-condition values are to be read, and if solute transport is being simulated.

CSBC - Mass fraction (or scaled-mass fraction) associated with a specified-pressure boundary-condition node (-). If inflow occurs, this mass fraction will determine the solute-inflow rate.

3.3.5 TNP by i,j,k range; Optional, required only if specified-temperature boundary-condition data are to be read.

TNP - Temperature at specified-temperature boundary-condition nodes (T).

3.3.6 CNP by i,j,k range; Optional, required only if specified mass-fraction values are to be input.

CNP - Mass fraction (or scaled-mass fraction) for specified mass-fraction boundary-condition nodes.

Specified flux

3.4.1 RDFLXQ (T/F), RDFLXH (T/F), RDFLXS (T/F); Optional, required only if specified-flux boundary conditions exist.

RDFLXQ - True if specified fluid-flux values are to be read at this time.

RDFLXH - True if specified heat-flux values are to be read at this time.

RDFLXS - True if specified solute-flux values are to be read at this time.

3.4.2 QFFX, QFFY, QFFZ by i,j,k range; Optional, required only if specified fluid-flux values are to be read at this time.

QFFX, QFFY, QFFZ - Components of the fluid-flux vector for a boundary cell in the x,y, and z-coordinate directions, respectively (L^3/L^2t).

3.4.3 UDENBC by i,j,k range; Optional, required only if specified fluid-fluxes values are to be read at this time.

UDENBC - Density associated with specified-fluid flux (M/L^3); (kg/m^3) or (lb/ft^3). If inflow, this density determines the mass flux.

3.4.4 TFLX by i,j,k range; Optional, required only if specified fluid fluxes are to be read at this time, and if heat transport is being simulated.

TFLX - Temperature associated with specified fluid flux (T). If inflow, this temperature determines the heat flux.

3.4.5 CFLX by i,j,k range; Optional, required only if specified fluid fluxes are to be read at this time, and if solute transport is being simulated.

CFLX - Mass fraction (or scaled mass fraction) associated with specified fluid flux (-). If inflow, this mass fraction determines the solute flux.

3.4.6 QHFX, QHFY, QHFZ by i,j,k range; Optional, required only if specified heat-flux values are to be read at this time.

QHFX, QHFY, QHFZ - Components of the specified heat-flux vector for a boundary cell in the x,y and z-coordinate directions respectively. Heat flux should be specified only through faces where there is no fluid flux (E/L^2-t).

3.4.7 QSFX, QSFY, QSFZ by i,j,k range; Optional, required only if specified solute-flux values are to be read at this time.

QSFX, QSFY, QSFZ - Components of the specified solute flux for a boundary cell in the x,y and z coordinate directions respectively. Solute flux should be specified only through faces where there is no fluid flux (M/L^2-t).

Leakage boundary conditions

3.5.1 RDLBC(T/F); Optional, required only if leakage boundary-condition cells are employed.

RDLBC - True if leakage boundary-condition data are to be read this time.

3.5.2 PHILBC, DENLBC, VISLBC by i,j,k range; Optional, required only if leakage boundary-condition data are to be read at this time.

PHILBC - Potential energy per unit mass of fluid (eq. 2.5.3.1.1b) on the other side of the aquitard from the simulation region (E/M).

DENLBC - Density of the fluid on the other side of the aquitard (M/L^3), (kg/m^3) or (lb/ft^3).

VISLBC - Viscosity of the fluid on the other side of the aquitard (M/L-t); (kg/m-s) or (cP).

3.5.3 TLBC by i,j,k range; Optional, required only if leakage boundary-condition data are to be read at this time, and if heat transport is being simulated.

TLBC - Temperature of the fluid on the other side of the aquitard (T).

3.5.4 CLBC by i,j,k range; Optional, required only if leakage boundary-condition data are to be read at this time, and if solute transport is being simulated.

CLBC - Solute-mass fraction (or scaled-mass fraction) on the other side of the aquitard (-).

River leakage

3.5.5 I1,I2,J1,J2, HRBC, DENRBC, VISRBC, TRBC, CRBC; Optional, required only if river-leakage boundary-condition data are to be read at this time.

I1,I2,J1,J2 - Node or cell number ranges in the x and y directions for a river-leakage boundary-condition segment. They should correspond to the segments used to define the river in data record 2.22.4.

HRBC - Potentiometric head in the river (L).

DENRBC - Density of the river fluid (M/L^3); (kg/m^3) or (lb/ft^3).

VISRBC - Viscosity of the river fluid (M/L-t); (kg/m-s) or (cP).

TRBC - Temperature of the river fluid (T).

CRBC - Solute-mass fraction (or scaled-mass fraction) of the river fluid (-).

As many records of type 3.5.5 are used as necessary to include all the cells at which a river-leakage boundary condition exists.

3.5.6 End this data set (record 3.5.5) with 0 /.

Aquifer influence functions

3.6.1 RDAIF (T/F); Optional, required only if aquifer-influence-function boundary-condition cells are employed.

RDAIF - True if aquifer-influence-function boundary-condition data are to be read at this time.

3.6.2 DENOAR by i,j,k range; Optional, required only if aquifer-influence-function cells are employed.

DENOAR - Density of the fluid in the outer-aquifer region (M/L^3); (kg/m^3) or (lb/ft^3).

3.6.3 TAIF by i,j,k range; Optional, required only if aquifer-influence-function boundary condition cells are employed, and if heat transport is being simulated.

TAIF - Temperature of the fluid in the outer-aquifer region associated with a given aquifer-influence-function cell (T).

3.6.4 CAIF by i,j,k range; Optional, required only if aquifer-influence-function cells are employed, and if solute transport is being simulated.

CAIF - Mass fraction of solute in the outer-aquifer region associated with a given aquifer-influence-function cell (-).

Calculation information

The following data pertains to time-step control and the time when new transient data will be read.

3.7.1 RDCALC (T/F).

RDCALC - True if calculation information is to be read at this time.

3.7.2 AUTOTS (T/F); Optional, required only if calculation information is to be read at this time.

AUTOTS - True if automatic time-step adjustment is desired for the next interval of simulation time.

3.7.3A DELTIM; Optional, required only if automatic time-step calculation is not being used and calculation information is being read at this time.

DELTIM - Time-step length (t).

3.7.3B DPTAS, DTTAS, DCTAS, DTIMMN, DTIMMX; Optional, required only if automatic time-step calculation is being used and calculation data are being read at this time.

DPTAS - Maximum change in pressure allowed for setting the time step automatically (F/L^2). Default set at 5×10^4 Pa.

DTTAS - Maximum change in temperature allowed for setting the time step automatically (T). Default set at 5 °C.

DCTAS - Maximum change in mass fraction (or scaled-mass fraction) allowed for setting the time step automatically (-). Default set at 0.25 (scaled).

DTIMMN - Minimum time step required (t). This time step will be used for the first two steps after a change in boundary conditions, that is, at TIMCHG. Default set at 10^4 s.

DTIMMX - Maximum time step allowed (t). Default set at 10^7 s.

3.7.4 TIMCHG.

TIMCHG - Time at which new transient data will be read or at which the simulation will be terminated (t).

Output information

3.8.1 PRIVEL, PRIDV, PRISLM, PRIKD, PRIPTC, PRIGFB, PRIWEL, PRIBCF.

PRIVEL - Printout interval (integer) for velocity arrays. These are interstitial velocities at the cell boundaries.

PRIDV - Printout interval (integer) for fluid density and fluid-viscosity arrays.

PRISLM - Printout interval (integer) for solution-method information, number of iterations, maximum changes in dependent variables, and so forth.

PRIKD - Printout interval (integer) for conductance and dispersion-coefficient arrays.

PRIPTC - Printout interval (integer) for pressure, temperature, and mass-fraction arrays.

PRIGFB - Printout interval (integer) for flow-balance information for the region.

PRIWEL - Printout interval (integer) for well information.

PRIBCF - Printout interval (integer) for specified-value boundary-condition flow rates.

For all of the above printout intervals:

0 - Means no printout of this information.

n - Means that printout will occur every nth time step and at the end of the simulation.

-1 - Means that printout will occur only at the time at which new transient data will be read and at the end of the simulation.

3.8.2 IPRPTC; Optional, required only if dependent-variable printouts are desired.

IPRPTC - Index for printout of pressure, temperature and mass-fraction arrays. It is of the form $n_1n_2n_3$ where n_1 is for the pressure; n_2 is for the temperature; and n_3 is for the mass-fraction array. The n_i are set to 1 if printout is desired for the ith variable. n_1 is set to 2 if both pressures and potentiometric heads are to be printed for isothermal cases. n_2 is set to 2 if both temperatures and fluid enthalpies are to be printed.

3.8.3 CHKPTD (T/F), NTSCHK, SAVLDO (T/F).

CHKPTD - True if check-point dumps are to be made for possible restarts of the simulation.

NTSCHK - Number of time steps between successive check-point dumps. If set to -1, a dump will occur only at the times when new transient data are read and at the end of the simulation.

SAVLDO - True if only the last check-point dump is to be saved.

The following four records are for the generation of contour maps on the line printer.

3.9.1 RDMPDT (T/F), PRTMPD (T/F).

RDMPDT - True if control data for map generation are to be read at this time.

PRTMPD - True if control data for map generation are to be written to the output file.

3.9.2 MAPPTC, PRIMAP; Optional, required only if contour-map-control data are to be read at this time.

MAPPTC - Index for a zoned contour map. It is in the form $n_1n_2n_3$, where n_1 is for pressure; n_2 is for temperature; n_3 is for mass fraction. The n_i are set to 1 if a contour map is desired for the ith dependent variable.

PRIMAP - Printout interval (integer) for contour maps. Number of time steps between map generations.

- 0 - means no contour maps.
- n - means contour maps at every nth time step.
- -1 - means contour maps at the time when new transient data will be read and at the end of the simulation.

3.9.3 YPOSUP (T/F), ZPOSUP (T/F), LENAX, LENAY, LENAZ; Optional, required only if contour-map-control data are to be read at this time.

YPOSUP - True if the y-axis is positive upward on the page for this contour-map set.

ZPOSUP - True if the z-axis is positive upward on the page for this contour-map set.

LENAX - Length of the x-axis on the page for this contour-map set (in).

LENAY - Length of the y-axis on the page for this contour-map set (in).

LENAZ - Length of the z-axis on the page for this contour-map set (in).

3.9.4 IMAP1, IMAP2, JMAP1, JMAP2, KMAP1, KMAP2, AMIN, AMAX, NMPZON; Optional, required only if contour-map-control data are to be read at this time.

IMAP1, IMAP2 - Range of node numbers along the x-axis for a contour map. Default set from 1 to NX.

JMAP1, JMAP2 - Range of node numbers along the y-axis for a contour map. Set to 1,1 for cylindrical coordinates. Default set from 1 to NY.

KMAP1, KMAP2 - Range of node numbers along the z-axis for a contour map. Default set from 1 to NZ.

AMIN, AMAX - Range of the dependent variable for a contour map (appropriate units). If a pair of null entries, automatic scaling of the range will be performed of that dependent variable.

One pair of the indices iMAP1 and iMAP2 may be set equal to produce a contour map for just one plane.

NMPZON - Number of zones into which the contour map will be divided. Default set at 5. Limit of 32.

Up to three records of type 3.9.4 may be needed, depending on which combinations of pressure, temperature, and mass fraction are selected for mapping. The record order is: (1) Pressure-map data, (2) temperature-map data, and (3) solute-mass-fraction map data.

This ends the transient data set that is read at a given time. At simulation time equal to TIMCHG, another transient-data set will be read, until the simulation is finished. At that time, THRU is read as true in the following record.

3.99.1 THRU (T/F).

THRU - Set to true at this point to signify the end of the simulation.

3.99.2 PLOTWP(T/F), PLOTWT(T/F) PLOTWC(T/F).

PLOTWP - True if observed and(or) calculated well pressures are to be plotted versus time.

PLOTWT - True if observed and(or) calculated well temperatures are to be plotted versus time.

PLOTWC - True if observed and(or) calculated well mass fractions are to be plotted versus time.

Temporal-plot information

The following data records of type 4.N are required only if character-string plots of variables versus time are desired at selected wells.

4.1 IWEL, RDPLTP (T/F); Optional, required only if temporal plots are to be made, and new plot-control parameters are to be set for subsequent plots (RDPLTP is true).

IWEL - Well number. This number must agree with the number associated with the calculated data.

RDPLTP - True if new plotting-control parameters are to be read at this time for subsequent plots.

4.2 IDLAB; Optional, required only if temporal plots are to be made, and new plot-control parameters are to be read for subsequent plots (RDPLTP is true).

IDLAB - Identification label for this well's plots. A character string of up to 80 characters. Space over at least one character position from the left.

4.3 NTHPTO, NTHPTC, PWMIN, PWMAX, PSMIN, PSMAX, TWMIN, TWMAX, TSMIN, TSMAX, CMIN, CMAX; Optional, required only if temporal plots are to be made, and new plot-control parameters are to be read for subsequent plots (RDPLTP is true).

NTHPTO - Index for plotting the first, then every nth observed data point versus time. Default set to one. A blank may be entered if no observed data are to be plotted.

NTHPTC - Index for plotting the first then every nth calculated value versus time. Default set to one.

PWMIN, PWMAX - Minimum and maximum values of pressure at the well datum that set the axis range for the temporal plots (F/L^2).

PSMIN, PSMAX - Minimum and maximum values of pressure at the land surface that set the axis range for the temporal plots (F/L^2).

TWMIN, TWMAX - Minimum and maximum values of temperature at the well datum that set the axis range for the temporal plots (T).

TSMIN, TSMAX - Minimum and maximum values of temperature at the land surface that set the axis range for the temporal plots (T).

CMIN, CMAX - Minimum and maximum values of solute-mass fraction (scaled-mass fraction) at the well datum and the land surface that set the range for the temporal plots (-).

The pressure, temperature and solute-mass fraction ranges for the plots can be specified by the user or established automatically. The latter option is invoked by entering zeros for the maximum and minimum values.

4.4 TO, POW, POS, TOW, TOS, COW; Optional, required only if there are observed data.

TO - Time of observation (t).
POW - Pressure observed at the well-datum level (F/L^2).
POS - Pressure observed at the land surface in the well (F/L^2).
TOW - Temperature observed at the well-datum level (T).
TOS - Temperature observed at the land surface in the well (T).
COW - Mass fraction (or scaled-mass fraction) observed in the well at the well datum or the land surface (-).

It is assumed that the observed data are in the same units that will be used for output of the calculated data. As many records of type 4.4 are used as necessary to enter all the observed well data. There may be wells for which only calculated data are available; for these wells, no records of type 4.4 will be read.

4.5 End this data set with -1. / .

Indicates the end of the observed data set for this well.

As many records of type 4.1-4.4 are used as necessary for all of the wells for which observed or calculated data are being plotted.

4.6 End this data set with 0 / .

Indicates the end of the temporal-plot information and the observed data for all the wells.

This ends the input-data-file description. For quick reference, a list of the definitions for the various program-control options is provided in table 5.3.

Table 5.3.--*Option lists for program-control variables*

Variable	Option definitions
SLMETH	1 - Selects the direct, D4, equation solver.
	2 - Selects the iterative, two-line, successive-overrelaxation equation solver.

In the following, n_1 denotes outward normal direction to the boundary face: 1 is the x-direction; 2 is the y-direction; and 3 is the z-direction.

Variable	Option definitions
IBC	-1 - Cell is excluded from the simulation region.
	100 - Specified-pressure boundary-condition node.
	010 - Specified-temperature boundary-condition node.
	001 - Specified-solute-concentration boundary-condition node.
	$n_1$00200 - Specified-fluid-flux boundary-condition cell.
	$n_1$00020 - Specified-diffusive-heat-flux boundary-condition cell.
	$n_1$00002 - Specified-diffusive-solute-flux boundary-condition cell.
	$n_1$00300 - Leakage boundary-condition cell.
	$n_1$00400 - Aquifer-influence-function boundary-condition cell.
	$n_1$00040 - Heat-conduction boundary-condition cell.
IAIF	1 - Pot aquifer for outer region.
	2 - Transient-aquifer-influence function with calculation using the Carter-Tracy approximation.
WQMETH	10 - Specified well-flow rate with allocation by mobility and pressure difference.
	11 - Specified well-flow rate with allocation by mobility.
	20 - Specified pressure at well datum with allocation by mobility and pressure difference.
	30 - Specified well-flow rate with a limiting pressure at well datum. Flow-rate allocation by mobility and pressure difference.
	40 - Specified surface pressure with allocation by mobility and pressure difference. Well-riser calculations will be performed.
	50 - Specified surface-flow rate with limiting surface pressure. Allocation by mobility and pressure difference. Well-riser calculations will be performed.
	0 - Observation well or abandoned well.
FDSMTH	0.5 - Centered-in-space differencing for advective terms.
	0.0 - Upstream differencing for advective terms.
FDTMTH	0.5 - Centered-in-time differencing.
	1.0 - Backward-in-time or fully implicity differencing.

Table 5.3.--*Option lists for program-control variables*--Continued

Variable	Option definitions
IPRPTC	1xx - Printout of pressure field. 2xx - Printout of pressure and potentiometric-head fields. x1x - Printout of temperature field. x2x - Printout of temperature- and fluid-enthalpy fields. xx1 - Printout of solute-concentration field.
ORENPR	12 - Printouts of arrays by areal (x-y) layers. 13 - Printouts of arrays by vertical x-z or (r-z) slices. A negative value means the y or z-axis is to be positive down the page.
PRIxxx	0 - No printout. n - Printout every nth time step and at the end of the simulation. -1 - Printout only at the time of new transient data being read and at the end of the simulation.
MAPPTC	1xx - Pressure-contour maps desired to be produced. x1x - Temperature-contour maps to be produced. xx1 - Solute-concentration contour maps to be produced.

6. OUTPUT DESCRIPTION

Various types of output result from running the HST3D program. Most of the output is to disc files to be displayed on a video screen or routed to a line printer. These files are written in ASCII format. The two exceptions are the optional check-point/restart dumps written in binary format to a disc file, and the calculated dependent-variable data for the wells that periodically are written, also in binary format, to a disc file for the temporal plots that can be made at the end of the simulation.

Output is generated at several stages during the simulation. Some information, such as the heading, title, array-partitioning data, and problem-geometrical information, is printed always. The heading contains the program version number which will change when major modifications or corrections are made. The units employed for the ouput are the same as those used for the input data, either metric or inch-pound as specified in record 1.4. Table 5.2 and the input-record descriptions (section 5.2.2) give the inch-pound and the metric units employed. For easier reading, variables are identified in the output by descriptors rather than program-variable names. Much of the output is optional, and the numbers of the records containing the control variables in the data-input-form list of table 5.1 are indicated. The writing of a file that echos each record of input data, as it is read, is optional (record 1.10). The static data that may be printed include porous-media properties, fluid properties, initial-condition distributions, boundary-condition information, solution-method information, well information (record 2.23.1), and density and viscosity distributions (record 2.23.2). The selection of which of the dependent variables (pressure, temperature or mass fraction) will have initial conditions printed is made in record 2.23.2.

Print intervals can be selected individually for information that is printed at the end of a time step. The information printed may include the velocity distribution, the density and viscosity distributions, the solution method information, the conductance and dispersion-coefficient distributions, the dependent-variable distributions, the regional fluid-flow, heat-flow and solute-flow rates, the regional cumulative-flow results, and the specified-value boundary-condition flow rates (record 3.8.1). The selection of of the dependent variables which will be printed is determined in record 3.8.2.

Contour maps of pressure, temperature, and mass fraction can be produced on the line printer; they are zoned into intervals and may cover subregions of the simulation as specified by the user (record 3.9.4). The contour-mapping routine produces character-string plots. Alternating zones of symbols and blanks are used to make perception easier. The user can make a programming change (set variable, ZEBRA, to false) to cause the symbol-filled zones to be adjacent to each other. Contour intervals are automatically calculated to be a multiple of 2, 5, or 10. The lower and upper limits can be chosen by the user or determined from the range of the data to be contoured. In the former case, values below the specified-lower limit are contoured with a zone of minus signs and values above the specified-upper limit are contoured with a zone of plus signs. The contour zones contain their lower-boundary values; the upper-boundary values belong to the next zone above with the exception of the highest zone of the map which does contain its upper-boundary value. The maps are either areal or vertical slices along nodal planes of the three-dimensional region (record 2.23.3), with the orientation sepcified in record 3.9.3. The size of the maps on the paper is chosen by the user (record 3.9.3). An echo printout of the mapping specifications can be requested (record 3.9.1). Bilinear interpolation is used to locate the contour-interval boundaries; cells excluded from the simulation region are indicated by X's. If multiple pages are used for the contour maps, no printing is done across the paper folds. Thus, separation and alinement of the various pages is necessary to eliminate gaps.

Temporal plots of selected variables are also in character-string format. The plots that may be produced at the end of the simulation include well-datum pressure, well-surface pressure, well-datum temperature, well-surface temperature, and well solute-mass fraction (or scaled-mass fraction) (record 3.99.2). For observation wells, the well-datum value is taken to be the value in the aquifer cell at the well-datum level. Observed (record 4.4) and calculated data of the same type are plotted together for comparison purposes. The time axis runs down, and the dependent-variable axis runs across the page.

A limit of 500 lines is set. If the time series to be plotted of any calculated variable contains more than three times the total number of node points in the region, or the series of any observed variable contains more than two times the number of node points in the region, array-storage problems will occur and program execution will be terminated. These problems may be avoided by plotting the first point followed by only every nth point thereafter (record 4.3). The user may specify the ranges of the variables to be plotted. However automatic scaling of the plot is available using the minimum and maximum values of the variables. Axis subdivisions that are a multiple of 2, 5 or 10 are produced. The present version of the HST3D code contains no provision for producing line plots on pen-plotting devices or video screens.

7. COMPUTER-SYSTEM CONSIDERATIONS

The heat- and solute-transport simulation program was developed initially on a Control Data Cyber 170/720 computer[1], and finally on a Prime 9950 computer. The Cyber computer has a very fast arithmetic central-processing unit relative to the Prime, while the Prime has virtual storage that the Cyber does not have. Therefore, the programming philosophies needed to create the optimum code for execution of large, long-running simulations are in direct opposition for these two machines. Specifically, the Cyber, with its fast arithmetic, but limitations on storage, is most efficiently used with a code that minimizes storage requirements. This is accomplished to a certain extent by recalculating some quantities each time they are needed, rather than storing them. On the other hand, the virtual storage of the Prime means that storage space is not a limiting factor; but, the slower arithmetic means that the running time for large, long simulations may become inconveniently long. This implies that the most efficient code for the Prime will use more storage than the Cyber and never compute a quantity more than once.

The present version of the heat- and solute-transport code is not optimal for either type of machine, but it tends to be oriented toward the Prime. Further optimization of the program will require timing tests. The storage requirement on the Prime computer for the executable-code module is about 1.1 megabytes, exclusive of the variably partitioned arrays, when compiled with the interactive-debug option and no optimization.

The language used for this program is FORTRAN-77, although some FORTRAN-IV coding still exists. An attempt has been made to use only the ANSI standard FORTRAN-77 for maximum portability (American National Standards Institute, 1978).

Double-precision arithmetic has been used for all real variables. Separate variably partitioned arrays were defined for real and integer vari-

---

[1]Use of brand names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

ables, so that the real variables could be made double precision. FORTRAN-77 intrinsic function names were used in their generic form, so that no changes had to be made for use with double-precision variables.

Although the standards for FORTRAN-77 have been followed as closely as possible, there are always problems of portability to different computer systems. Development experiences revealed some differences between the Prime and Cyber computers that affect program portability.

There are no alternate entry points into any of the subroutines, to avoid computer incompatibilities. The Prime uses dynamic storage of variables and of compiled code during program execution. However, all variables in common blocks are automatically made static. Static data items retain their values between subprogram references, while dynamic data items in a subprogram lose their values upon return from that subprogram. By including all common blocks in the main program, potential problems with computer systems that do not make common block variables automatically static should be avoided. Variables are explicitly initialized where necessary, so no reliance is made on system-default initialization.

The Cyber computer FORTRAN compiler allows only 63 arguments in a subprogram-argument list. The Prime compiler allows 256. There are a few subroutines in the heat- and solute-transport program that have more than 63 arguments. To reduce the number to within the limit for the Cyber computer, some subarrays would need to be eliminated from the argument list, necessitating some recoding. The eliminated subarrays would be passed by passing the entire variably partitioned arrays along with appropriate pointer indices.

The Cyber requires the BLOCKDATA subprogram to be contained in the same file as an executable subprogram; whereas, the Prime allows it to be compiled from its own separate file. The version of the code discussed in this documentation has the main program, and each subprogram, including BLOCKDATA, contained in a separate file.

8. COMPUTER-CODE VERIFICATION

Verification of a computer program is the process of ensuring that the code performs the intended calculations correctly. This is in contrast to computer-model validation, which is the demonstration that a particular model with a particular set of parameters adequately describes a given physical situation. Program verification is accomplished by running various test problems for which an analytical solution is known, or for which numerical results from another verified program are available.

8.1. SUMMARY OF VERIFICATION TEST PROBLEMS

Several sets of test problems have been used for verification and are summarized hereinafter. Verification is a continuing process, as many combinations of program options could be tested.

8.1.1. One-Dimensional Flow

Test problem set 1 tested the ability of HST3D to simulate compressible flow and was based on the analog between confined ground-water flow and heat conduction. A thermal-conduction problem was taken from Carnahan and others (1969, p. 443). The physical situation was that of one-dimensional confined flow of a compressible fluid when a unit-step increase of pressure was applied at both ends of the region. The dimensions of the simulation region were 1.0 meter in each direction. Eleven equally spaced nodes were used to discretize the region in the x-direction, with two nodes each in the y- and z-directions. The initial condition was hydrostatic equilibrium. The porosity was set to 1.0 so no porous medium was present. The parameters were set to the following values:

Porosity, 1.0

Permeabilities

x-direction, $1.\times10^{-8}$ m^2; y- and z-directions $1.\times10^{-20}$ m^2

Fluid compressibility, 1×10^{-5} Pa^{-1}

Fluid density (at reference conditions), 1,000 kg/m^3

Fluid viscosity (at reference conditions), 0.001 kg/m-s.

A fixed time step of $1.25\text{x}10^{-2}$ was used and the results compared at 2.5×10^{-2}s. Backwards-in-time differencing was used with the direct method for matrix solution.

Variations on the basic problem included using centered-in-time differencing, reduction of the time step length by a factor of 10, using the iterative solver for the matrix equation, and entering data in inch-pound units.

The results agreed to five significant digits with the numerical solution of Carnahan and others (1969, p. 446-447). Representative values for pressure increase at 2.5×10^{-2}s are shown in table 8.1. The results were symmetric about the mid-point of the region in the x-direction as expected.

Table 8.1.--*Representative values for the pressure increase from the HST3D simulator and the results of Carnahan and others (1969)*
[Values at time of 2.5×10^{-2} seconds]

Distance along column (meters)	Change in pressure calculated by HST3D (Pa)	Change in pressure calculated by Carnahan (Pa)
0	1.00000	1.00000
0.2	0.32471	0.32471
0.4	0.10104	0.10104
0.5	0.07965	0.07965

8.1.2. **Flow to a Well**

Flow to a single well in a cylindrical-coordinate system was the basis for test problem set 2, providing another test of the flow-simulation part of HST3D. Both confined and unconfined conditions were simulated. The confined-flow problem was taken from Lohman (1972, p. 19) and the unconfined-flow problem was example 1 from Boulton (1954). A specified flow rate from the well was used for both problems.

A cylindrical region of 2,000 ft external radius and 100 ft thickness was used for the confined-flow problem. The theoretical results of Lohman were based on the this solution for an infinite region, so comparisons were restricted to time values sufficiently small so that the outer boundary did not affect the flow field. Twenty-one nodes were used in the radial direction logarithmically spaced by the automatic discretization algorithm, except for those at 200 and 400 ft. Two nodes defined the vertical discretization. The well flow rate was allocated by mobility and the upper, lower and outer boundaries were impermeable. The initial condition was that of hydrostatic equilibrium. A time step of 30 s was used for a duration of 600 s. The parameters used for the confined problem were the following:

Porosity, 0.20
Fluid compressibility, 3.33×10^{-6} psi^{-1}
Porous-medium compressibility, 3.94×10^{-6} psi^{-1}
Permeability, 5.31×10^{-10} ft^2 (hydraulic conductivity 137 ft/d)
Well radius, 0.1 ft
Well flow rate, 96,000 ft^3/d
Fluid density at reference conditions, 62.4 lb/ft^3
Fluid viscosity at reference conditions, 1 cP

The calculated fluid drawdown was compared with the results of Lohman (1972, p. 19) for several locations at six values of time. The drawdown values agreed to within 0.01 to 0.1 ft (table 8.2). The differences were due mostly to spatial-discretization error, since they were reduced by 30 to 50 percent by doubling the number of nodes in the radial direction.

A cylindrical region of 2,000 ft radius and 800 ft thickness was used for the unconfined-flow problem. The theoretical solution of Boulton (1954) was used for comparison. This solution was based on a linearized free-surface boundary condition valid for small values of drawdown and included a correction for the fact that a line sink of constant intensity represented the well flow.

Sixteen nodes were distributed in the radial direction, logarithmically spaced, except at 160 ft, and nine nodes were distributed in the vertical direction. Advantage was taken of the fact that the free surface can rise above the upper plane of nodes. The well flow rate was allocated by mobility. The lower and outer boundaries were impermeable while the upper boundary was a free surface. The initial condition was that of hydrostatic equilibrium. The automatic-time-step algorithm was used to simulate from 10 s to 3×10^5 s.

Table 8.2.--*Comparison of drawdown values calculated by HST3D with those of Lohman (1972, p. 19)*

Time (second)	Drawdown (feet)			
	radius (feet)			
	200		400	
	HST3D	Lohman	HST3D	Lohman
60	0.56	0.66	0.15	0.16
120	0.91	0.99	0.33	0.38
240	1.29	1.36	0.61	0.67
300	1.42	1.49	0.72	0.77
480	1.67	1.75	0.94	0.99
600	1.79	1.86	1.05	1.12

The parameters for the unconfined-flow problem were the following:

Porosity, 0.15

Fluid compressibility, 1×10^{-15} psi^{-1}

Porous-medium compressibility, 1×10^{-15} psi^{-1}

Permeability, 6.7×10^{-12} ft^2 (hydraulic conductivity 2×10^{-5} ft/s)

Well radius, 0.1 ft

Well flow rate, 1 ft^3/s

Fluid density at reference conditions, 62.4 lb/ft^3

Fluid viscosity at reference conditions, 1 cP

The calculated drawdown of fluid was compared with the result of Boulton's (1954) example 1. His only reported value was at a time of 3.47 days and a radius of 160 ft. The drawdown was 2.13 ft compared to the HST3D result of 2.20 ft. Use of five nodes in the vertical direction reduced the drawdown calculated by HST3D to 1.96 ft. This indicated the effect of spatial-discretization error, particularly when vertical flow is important.

The agreement is very good considering that the HST3D simulator does not take the kinematic boundary condition at the free surface into account. Therefore greater discrepancies should appear as the well bore is approached.

8.1.3. **One-Dimensional Flow**

Three cases of ground-water flow for which analytic solutions are available (Bear, 1972, p. 301, 367, 380) formed test problem set 3. They were one-dimensional, confined flow; one-dimensional, unconfined flow; and one-dimensional, unconfined flow with influx from precipitation. The simulation region was 400 × 400 × 100 meters in the x-, y- and z-directions respectively. The flow field was horizontal in the y-direction for all cases. Specified-pressure boundary conditions were used on the inlet and outlet boundaries with impermeable lateral and bottom boundaries.

The parameters employed were the following:

Porosity, 0.15
Fluid compressibility, 5×10^{-15} Pa^{-1}
Porous-medium compressibility, 8.8×10^{-14} Pa^{-1}
Permeability, 1.18×10^{-11} m^2
Fluid density at reference conditions, 1,000 kg/m^3
Fluid viscosity at reference conditions, 0.001 kg/m-s

Spatial discretization was accomplished using five equally spaced nodes in the x- and y-directions and two nodes in the z-direction. All cases were run to steady-state with a time step of 86,400 s.

Case 1: Confined flow

The upper boundary surface was made impermeable and the other boundary conditions for this case were specified potentiometric heads of 200 m along the y = 0 boundary and 100 m along the y = 400 m boundary. The analytical solution was a linear potentiometric-head variation between the two boundaries

(Bear, 1972, p. 301). The results at steady-state agreed with the analytical solution to five significant digits. Global-balance results were verified by hand calculation.

Case 2: Unconfined flow

The upper boundary was defined as a free surface and the region dimension in the z-direction was extended to 200 m. The permeabilities in the x-, y-, and z-directions were modified to 1.18×10^{-9} m^2, 1.18×10^{-10} m^2 and 1.18×10^{-5} m^2 respectively. The high permeability in the z-direction was to make the Dupuit approximation of hydrostatic equilibrium in the vertical direction valid. The specified potentiometric heads were 200 m along the y = 0 m boundary and 150 m along the y = 400 m boundary.

The simulation was run to 172,800 s, which was essentially steady-state. The results for potentiometric head were compared with the analytical solution (Bear, 1972, p. 367) based on the Dupuit approximation. Agreement to five significant digits was obtained at the three interior-node locations along the y-axis. The global-balance results were verified by hand calculation.

Better agreement with the analytical solution was obtained for this test problem than for the unconfined flow to a well in test problem set 2. This improved agreement was attributed to the fact that the analytical solution for this problem was based on the Dupuit assumption of negligible vertical flow and a high vertical-permeability value was used in the HST3D simulation to achieve hydrostatic equilibrium. The unconfined case of test problem set 2 had significant vertical flow. Because the HST3D simulation does not attempt to satisfy the non-linear, kinematic, free-surface boundary condition, described in sections 2.5.6 and 3.4.6, the poor results obtained in cases of significant vertical flow at the free surface were not surprising.

Case 3: Unconfined flow with precipitation recharge

Case 2 was modified by the addition of areal recharge at a uniform rate and distribution. The region dimension in the z-direction was extended to 275 m. The areal-recharge flux was set to $-1,157\times10^{-3}$ m^3/m^2s with the negative sign denoting flux in the negative z-direction.

The simulation was run to 172,800 s, which was essentially steady-state. The results for potentiometric head were compared with the analytical solution from Bear (1972, p. 380). This solution also was based on the Dupuit approximation. Agreement to five significant digits was obtained at the three interior-node locations along the y-axis. The global-balance results were verified by hand calculation.

8.1.4. **One-Dimensional Solute Transport**

Flow with solute transport in a one-dimensional column was the basis of test problem set 4. A steady-state flow field was established by specifying an initial-pressure gradient along the column. The boundary condition at the column inlet was a specified scaled-solute concentration of a unit-step at time zero. The column length was 160 m discretized by 21 equally distributed nodes in the x-direction. The y- and z-directions were 1 m with two nodes in each direction. The following parameter values were chosen:

Porosity, 0.5
Fluid compressibility, 1×10^{-10} Pa^{-1}
Porous-medium compressibility, 1×10^{-10} Pa^{-1}
Permeability, 1×10^{-10} m^2
Fluid density at reference conditions, 1,000 kg/m^3 (independent of solute concentration)
Fluid viscosity at reference conditions, 0.001 kg/m-s
Interstitial velocity along the column, 2.7778×10^{-4} m/s
Longitudinal dispersivity, 10 m
Molecular diffusivity, 1×10^{-10} m^2/s

The initial solute concentration in the column was zero. A time-step length of 720 s was used for a total simulation time of 7,200 s. The HST3D options tested included the spatial and temporal differencing methods and the two different equation solvers. A second case with a column that was 4-meters wide was also tested.

The results were compared to a one-dimensional finite-difference transport program (Grove and Stollenwerk, 1984), and to an analytical solution (Ogata and Banks, 1961). The calculated solute-mass fraction values agreed with the finite-difference program results to four significant digits. Differences between the HST3D results and the analytical solution were as much as 0.035 units of scaled-mass fraction. The results at the end of the simulation period appear in table 8.3.

Table 8.3.--*Scaled solute-concentration values calculated by HST3D compared to the one-dimensional finite-difference solution of Grove and Stollenwerk (1984) and the analytical solution of Ogata and Banks (1961)*

[Values at time of 7,200 seconds; CSCT, centered-in-space and centered-in-time differencing; BSBT, backward-in-space and backward-in-time differencing]

		Scaled solute concentration (-) Distance along the column (meters)				
		0	8	16	24	32
HST3D	CSCT	1.0000	0.31665	0.05939	0.007843	0.000801
	BSBT	1.0000	0.37500	0.09414	0.01824	0.00295
One-dimensional finite-difference solution	CSCT	1.0000	0.31666	0.05939	0.007843	0.000801
Analytical solution		1.0000	0.29808	0.02439	0.000469	0.000002

The discrepancies are attributed to spatial- and temporal-discretization errors, because reducing the spatial and temporal steps by factors of eight and five respectively, reduced the maximum difference to 0.003 units of scaled-mass fraction. The differences between the two differencing schemes were the result of numerical-dispersion errors. The simulation was not run long enough for the difference between the analytic and numerical boundary condition at the far end of the column to affect the solution. Flow and solute global-balance residuals were at least eleven orders of magnitude smaller than the net amounts entering the region.

8.1.5. One-Dimensional Heat Conduction

Test problem set 5 involved heat transport without fluid flow. A heat-conduction problem was taken from Carnahan and others (1969, p. 443). The physical situation was that of one-dimensional heat conduction when a unit-step increase of temperature was applied at both ends of a column. The dimensions of the region were 1 m in each direction. Eleven equally spaced nodes were used to discretize the region in the x-direction, with two nodes each in the y- and z-directions. The initial condition was a uniform temperature of 1 °C. The parameters were set to the following values:

Porosity, 1.0 (no porous medium present)
Fluid compressibility, 5×10^{-6} Pa^{-1}
Permeability, 1×10^{-8} m^2
Fluid density, 1,000 kg/m^3
Fluid viscosity, 0.001 kg/m-s (independent of temperature)
Fluid thermal expansion factor, 0. $°C^{-1}$
Fluid heat capacity, 1.0 J/kg °C
Fluid thermal conductivity, 1.0 W/m-°C

A time step of 0.0125 s was used for a total simulation time of 0.0250 s. The options tested included backwards-in-time differencing, direct and iterative solvers of the matrix equations, and inch-pound and metric units for data entry and output. The results agreed to five significant digits with those of Carnahan and others (1969, p. 446, 447) and matched the numerical values for scaled-pressure rise given in table 8.1. Heat-balance residuals were 12 orders of magnitude less than the amount of heat that entered the region.

8.1.6. One-Dimensional Heat Transport

Test problem set 6 was heat transport with fluid flow and was the analog to problem set 4. A steady-state flow field was established by specifying an initial-pressure gradient along the column. The boundary condition at the column inlet was a specified scaled-temperature of a unit-step at time zero. The dimensional-temperature step was 10 °C. The column length was 160 m discretized by 21 equally distributed nodes in the x-direction. The y- and z-directions were 1 m with two nodes in each direction. The following parameter values were used:

Porosity, 0.5
Fluid compressibility, 1×10^{-10} Pa^{-1}
Porous-medium compressibility, 1×10^{-10} Pa^{-1}
Permeability, 1×10^{-10} m^2
Fluid density at reference conditions, 1,000 kg/m^3 (independent of temperature)
Fluid viscosity at reference conditions, 0.001 kg/m-s (independent of temperature)
Interstitial velocity along the column, 2.7778×10^{-2} m/s
Longitudinal dispersivity, 10 m
Porous-medium product of density and heat capacity, 800 J/m^3-°C
Porous-medium thermal conductivity, 1.8 W/m-°C
Fluid heat capacity, 4,200 J/kg-°C
Fluid thermal conductivity, 0.6 W/m-°C
Fluid thermal expansion factor, 0. $°C^{-1}$

The initial temperature in the column was 10 °C. A time step length of 1076.5 s was used for a total simulation time of 10,765 s. Centered-in-space and centered-in-time differencing were used for discretization.

The results were compared to a one-dimensional finite-difference transport program (Grove and Stollenwerk, 1984), and to an analytical solution (Ogata and Banks, 1961). The scaled temperature values matched the analogous solute-transport problem of set 4 as expected. Numerical differences in the fourth significant digit were attributed to the fact that the thermal-dispersion coefficient was slightly larger than the solute-dispersion coefficient. The results at the end of the simulation period appear in table 8.4.

Table 8.4.--*Scaled temperature values calculated by HST3D compared to the one-dimensional finite-difference solution of Grove and Stollenwerk (1984) and the analytical solution of Ogata and Banks (1961)*

[Values at time of 10,765 seconds]

	Scaled temperature (-) Distance along the column (meters)					
	0	8	16	24	32	56
HST3D	1.0000	0.31670	0.05941	0.007846	0.000802	0.000005
One-dimensional finite-difference solution	1.0000	0.31671	0.05941	0.007847	0.000802	0.000000
Analytical solution	1.0000	0.29815	0.02441	0.000470	0.000002	0.000000

The time at which the temperature profile essentially matched the solute-concentration profile is a factor of about 1.5 later. This is because the thermal-storage coefficient, which includes the porous-matrix solid phase as well as the fluid phase, is greater than the solute-storage coefficient involving only the fluid phase. The effect of spatial- and temporal-discretization errors can be seen by comparing the results to the analytical solution. The simulation was not run long enough for the difference between the analytic and the numerical boundary condition at the far end of the column to affect the solution. Flow and heat global-balance residuals were at least eight orders of magnitude smaller than the net amounts entering the region.

8.1.7 Thermal Injection in a Cylindrical Coordinate System

Simulation of the injection of hot water at 60 °C into an aquifer initially at 20 °C for 90 days, followed by production for an equivalent time period, formed test problem set 7. A cylindrical-coordinate system was used with a fully penetrating well. Both density and viscosity were taken to be functions of temperature. Results from this test problem were compared to the results obtained by Voss (1984) p. 207-212 using his SUTRA finite-element transport-simulation program. Options tested included the approximate, augmented-diagonal treatment of the cross-derivative dispersion fluxes, the explicit evaluation of the cross-derivative dispersion fluxes, central and upstream differencing for the advective terms, and equal and unequal longitudinal and transverse dispersivities.

A list of the parameters employed will not be presented here, because this test problem also is given as an example for the user. The complete set of parameters and other data that define the problem is presented in section 8.2.2.

The region had an interior radius of 1 m, exterior radius of 226 m, and a thickness of 30 m. At the upper and lower surfaces, the boundary conditions were no fluid flow and no heat flow. At the exterior radius, a hydrostatic pressure was specified with any fluid entering the region having a temperature of 20 °C. The initial condition was hydrostatic equilibrium of the fluid at 20 °C.

The results for the temperature field at 30, 90, 120, 150, 180 days were compared to those of Voss (1984, p. 207-212) for his particular case of options. The profiles were in general agreement with some deviation for the withdrawal phase. The warmer water was extracted more rapidly in the HST3D simulation. This can be attributed to the fact that Voss used a line sink of constant intensity per unit length to represent the well whereas the HST3D simulator allocated the flow from each layer in proportion to the local-fluid mobility. The increased mobility of the warmer fluid caused increased flow from the upper parts of the well bore and decreased flow from the lower. The total-flow rate was held constant. The finite-difference methods used by the HST3D simulator showed some spatial oscillation of about 3 °C. Use of upstream differencing removed this problem at the cost of increased dispersion which reduced the sharpness of the temperature front considerably.

8.2. TWO EXAMPLE PROBLEMS

Two example problems are presented for the purpose of giving the new HST3D program user some experience in learning to run a successful simulation. One involves solute transport and the other involves heat transport. These examples also will aid in adapting the HST3D program to run on computer systems other than the PRIME. The problem descriptions, data files and selected parts of output are included for comparison. Several ways are available to input some of the data so an exact match with the data files presented is not necessary to execute these examples correctly.

8.2.1. Solute Transport with Variable Density and Variable Viscosity

The first example problem is based upon displacement of a fluid of one density and viscosity by another fluid of different density and viscosity. The density and viscosity differences are caused by different amounts of dissolved solute. The system is isothermal. The region is a square slice of porous medium, that is oriented vertically, so that gravitational effects will occur in the flow field. The dimensions are 2 m in the x-direction by 0.2 m in the y-direction by 2 m in the z-direction. The z-direction is oriented vertically upward. Except at the inlet and outlet corners, the boundaries are confining. The porous medium is homogeneous and isotropic.

The parameters to be used are as follows:

Permeability, 1×10^{-8} m^2
Porosity, 0.10
Density of fluid initially present, 800 kg/m^3

Viscosity of fluid initially present, 1.3×10^{-3} kg/m-s

Scaled-solute-mass fraction of fluid initially present, 0.

Temperature of region (isothermal), 20 °C

Density of injected fluid, 1,000 kg/m^3

Viscosity of injected fluid, 7×10^{-4} kg/m-s

Scaled-solute-mass fraction of injected fluid, 1.0

Longitudinal dispersivity, 0.1 m

Transverse dispersivity, 0.1 m

Fluid compressibility, 0. Pa^{-1}

Porous-medium compressibility, 0. Pa^{-1}

The molecular diffusivity of the solute is neglected. Absolute mass fractions of solute are needed for solute-mass balance calculations. They may be chosen as 0.0 for the fluid initially in the region and 0.005 for the injected fluid. The initial condition is one of hydrostatic equilibrium with a pressure of 0.0 Pa at an elevation of 2 m. The injection location is at the lower left-hand corner of the region. The injection boundary is maintained at a scale-solute concentration of 1.0 with an injection pressure of 25,000 Pa. The outlet is at the upper right-hand corner of the region and is open to the atmosphere. The region is illustrated in figure 8.1.

Construct a numerical model of this system with nodal dimensions 11 × 2 × 11 and observe the migration of the fluid containing solute from the lower left-hand corner to the upper right-hand corner for a total simulation time of 10 s. Use a time-step length of 0.2 s. Use the approximate method for calculating the cross-derivative dispersive terms. Print out results at 10 s with a contour plot of solute concentration. Use a contour interval of 0.2.

A listing of the data file that will run example problem 1 is given in table 8.5. The input-data form (table 5.1) was used to construct this file, but comments pertaining to unnecessary data items have been eliminated for brevity.

The output file for this problem is contained in table 8.6. The header shows the release number for the version of the program. The problem title and information relating to dimensioning requirements is next. Then follows the static data. Read-echo printouts were selected for data input by i,j,k range. The conductance factors are constant for this simulation. The numbering sequence for boundary-condition cells is primarily for debugging purposes. In this problem, the approximate method for handling the cross-derivative dispersion terms using amplified-diagonal values was chosen.

The next section of the printout contains the transient data including both input parameters and output variables at selected time steps. For this example, the input data for boundary conditions, calculation information, and mapping data are printed. The printout interval was set to print at the end of the simulation only at time step 50. No cross-derivative dispersive conductances appear, because the approximate method was selected for handling cross-dispersive fluxes. The output at the end of the time step, and end of the simulation in this case, includes some calculation information, pressure and solute-mass fraction, density and viscosity, the global-balance summary, boundary-condition flow rates, a contour map of solute-mass fraction, and the velocity field. The contour map was made for only one plane of cells because the other is identical by symmetry. The velocity field is that which would be used for the next time step, if one were to be calculated. In the global-balance summary, we can see that the flow-balance residual is about 5 orders of magnitude less than the other amounts. The solute-balance residual is about 5 orders of magnitude less than the amount of inflow and amount of change. Similar results exist for the cumulative amounts. The map of the solute-mass fraction field shows the asymmetry that is caused by the denser injected fluid tending to stay in the bottom part of the region. The same case was run under conditions of constant density and viscosity and the results were symmetric about the diagonal as expected.

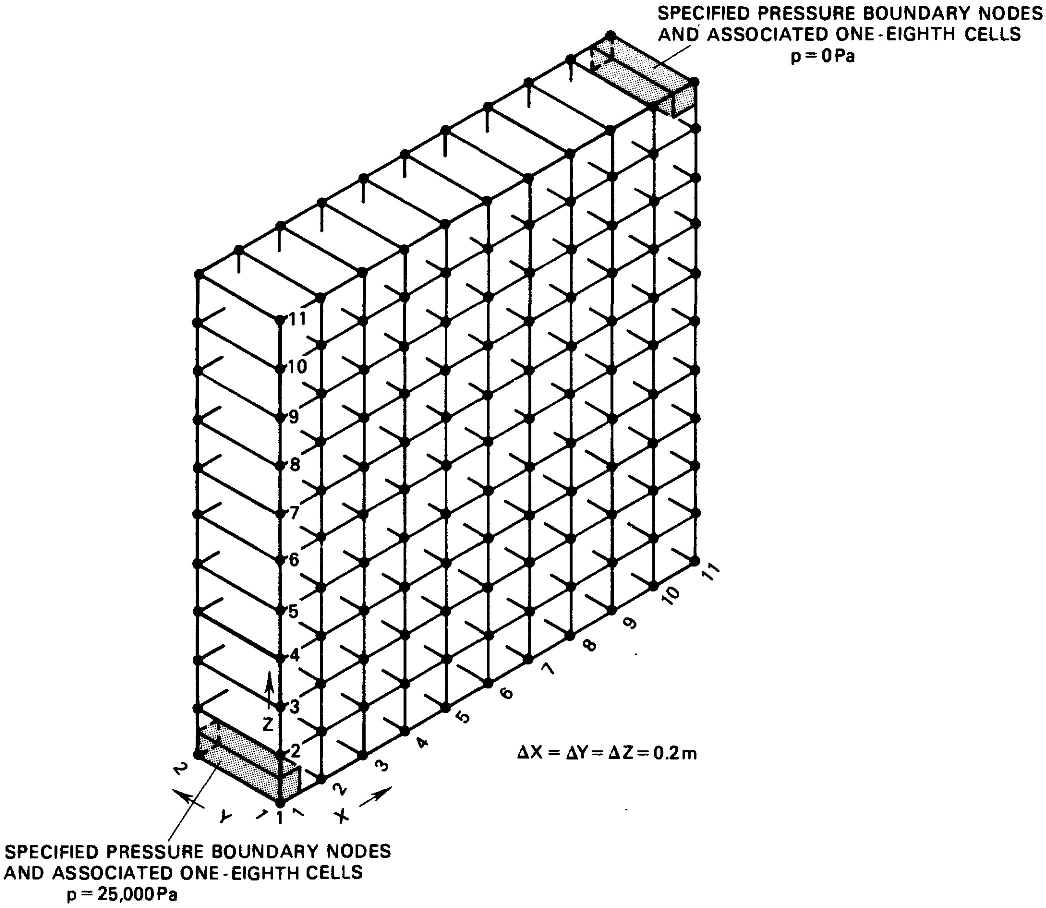


Figure 8.1.--Sketch of the grid with boundary conditions for example problem 1.

Table 8.5.--*Input-data file for example problem 1*

```
C.....HST DATA-INPUT FORM
C...   NOTES:
C...   INPUT LINES ARE DENOTED BY C.N1.N2.N3 WHERE
C...        N1 IS THE READ GROUP NUMBER, N2.N3 IS THE RECORD NUMBER
C...        A LETTER INDICATES AN EXCLUSIVE RECORD SELECTION MUST BE MADE
C...          I.E. A OR B OR C
C...   (O) - OPTIONAL DATA WITH CONDITIONS FOR REQUIREMENT
C...     A RECORD NUMBER IN SQUARE BRACKETS IS THE RECORD WHERE THAT PARAMETER IS FIRST SET
C.....INPUT BY I,J,K RANGE FORMAT IS;
C.0.1.. I1,I2,J1,J2,K1,K2
C.0.2..  VAR1,IMOD1,[VAR2,IMOD2,VAR3,IMOD3]
C...  USE AS MANY OF LINE 0.1 & 0.2 SETS AS NECESSARY
C...  END WITH LINE 0.3
C.0.3.. 0 /  THE SPACE IS REQUIRED
C...   {NNN} - INDICATES THAT THE DEFAULT NUMBER, NNN, IS USED
C...     IF A ZERO IS ENTERED FOR THAT VARIABLE
C...   (T/F) - INDICATES A LOGICAL VARIABLE
C...   [I] - INDICATES AN INTEGER VARIABLE
C-----------------------------------------------------------------------------------------------
C.....START OF THE DATA FILE
C.....DIMENSIONING DATA - READ1
C.1.1 .. TITLE LINE 1
     EXAMPLE #1 SOLUTE TRANSPORT WITH VARIABLE DENSITY AND VISCOSITY
C.1.2 .. TITLE LINE 2
          DIAGONAL FLOW IN X-Z PLANE
C.1.3 .. RESTRT(T/F),TIMRST
F /
C.1.4 .. HEAT,SOLUTE,EEUNIT,CYLIND,SCALMF ; ALL (T/F)
F T F F T
C.1.5 .. NX,NY,NZ,NHCN
11 2 11 0
C.1.6 .. NPTCBC,NFBC,NAIFC,NLBC,NHCBC,NWEL
4 5*0
C.1.7 .. NPMZ
1
C.1.8 .. SLMETH[I],LCROSD(T/F)
1 T
C.1.9 .. IBC BY I,J,K RANGE {0.1-0.3} ,WITH NO IMOD PARAMETER ,FOR EXCLUDED CELLS
0 /
C.1.10 .. RDECHO(T/F)
F
C-------------------------------------------------------------------------------
C.....STATIC DATA - READ2
C.....OUTPUT INFORMATION
C.2.1 .. PRTRE(T/F)
```

Table 8.5.--*Input-data file for example problem 1*--Continued

```
F
C.....COORDINATE GEOMETRY INFORMATION
C..... RECTANGULAR COORDINATES
C.2.2A.1 .. UNIGRX,UNIGRY,UNIGRZ; ALL (T/F) ; (0) - NOT CYLIND [1.4]
3*T
C.2.2A.2A .. X(1),X(NX) ;(0) - UNIGRX [2.2A.1]
0. 2.
C.2.2A.2B .. X(I) ;(0) - NOT UNIGRX [2.2A.1]
C.2.2A.3A .. Y(1),Y(NY) ;(0) - UNIGRY [2.2A.1]
0. .2
C.2.2A.3B .. Y(J) ;(0) - NOT UNIGRY [2.2A.1]
C.2.2A.4A .. Z(1),Z(NZ) ;(0) - UNIGRZ [2.2A.1]
0. 2.
C.2.2A.4B .. Z(K) ;(0) - NOT UNIGRZ [2.2A.1]
C.2.2B.3B .. Z(K) ;(0) - NOT UNIGRZ [2.2B.3A],CYLIND [1.4]
C.2.3.1 .. TILT(T/F) ;(0) - NOT CYLIND [1.4]
F
C.2.3.2 .. THETXZ,THETYZ,THETZZ;(0) - TILT [2.3.1] AND NOT CYLIND [1.4]
C.....FLUID PROPERTY INFORMATION
C.2.4.1 .. BP
0.
C.2.4.2 .. PRDEN,TRDEN,WO,DENFO
0. 20. 0. 800.
C.2.4.3 .. W1,DENF1 ;(0) - SOLUTE [1.4]
.005 1000.
C.2.5.1 .. NOTVO,TVFO(I),VISTFO(I) ,I=1 TO NOTVO;(0) - HEAT [1.4] OR HEAT [1.4] AND SOLUTE [1.4] OR .NOT.HEAT AND .NOT.SOLUTE [1.4]
C.2.5.2 .. NOTV1,TVF1(I),VISTF1(I) ,I=1 TO NOTV1;(0) -  SOLUTE [1.4] AND HEAT [1.4]
C.2.5.3 .. NOCV,TRVIS,CVIS(I),VISCTR(I) ,I=1 TO NOCV;(0) - SOLUTE [1.4]
2 20. 0. .0013 1. .0007
C.....REFERENCE CONDITION INFORMATION
C.2.6.1 .. PAATM
0.
C.2.6.2 .. PO,TO
0. 20.
C.....SOLUTE INFORMATION
C.2.8 .. DM,DECLAM ;(0) - SOLUTE [1.4]
0. 0.
C.....POROUS MEDIA ZONE INFORMATION
C.2.9.1 .. IPMZ,I1Z(IPMZ),I2Z(IPMZ),J1Z(IPMZ),J2Z(IPMZ),K1Z(IPMZ),K2Z(IPMZ)
1 1 11 1 2 1 11
C.....USE AS MANY 2.9.1 LINES AS NECESSARY
C.2.9.2 .. END WITH 0 /
0 /
C.....POROUS MEDIA PROPERTY INFORMATION
C.2.10.1 .. KXX(IPMZ),KYY(IPMZ),KZZ(IPMZ),IPMZ=1 TO NPMZ [1.7]
```

Table 8.5.--*Input-data file for example problem 1*--Continued

```
3*1.E-8
C.2.10.2 .. POROS(IPMZ),IPMZ=1 TO NPMZ [1.7]
.1
C.2.10.3 .. ABPM(IPMZ),IPMZ=1 TO NPMZ [1.7]
0.
C.....POROUS MEDIA SOLUTE AND THERMAL DISPERSION INFORMATION
C.2.12 .. ALPHL(IPMZ),ALPHT(IPMZ) ,IPMZ=1 TO NPMZ [1.7] ;(0) - SOLUTE [1.4] OR HEAT [1.4]
.1 .1
C.....POROUS MEDIA SOLUTE PROPERTY INFORMATION
C.2.13 .. DBKD(IPMZ) ,IPMZ=1 TO NPMZ [1.7] ;(0) - SOLUTE [1.4]
0.
C.....BOUNDARY CONDITION INFORMATION
C.....     SPECIFIED VALUE B.C.
C.2.15 .. IBC BY I,J,K RANGE {0.1-0.3} WITH NO IMOD PARAMETER, ;(0) - NPTCBC [1.6] > 0
1 1 1 2 1 1
101
11 11 1 2 11 11
100
0 /
C.....FREE SURFACE B.C.
C.2.20 .. FRESUR(T/F),PRTCCM(T/F)
F /
C.....INITIAL CONDITION INFORMATION
C.2.21.1 .. ICHYDP,ICT,ICC; ALL (T/F);IF NOT.HEAT, ICT = F, IF NOT.SOLUTE, ICC = F
T F T
C.2.21.2 .. ICHWT(T/F) ;(0) - FRESUR [2.20]
C.2.21.3A .. ZPINIT,PINIT ;(0) - ICHYDP [2.21.1] AND NOT ICHWT [2.21.2]
2. 0.
C.2.21.6 .. C BY I,J,K RANGE {0.1-0.3} ;(0) - SOLUTE [1.4] AND ICC [2.21.1]
1 11 1 2 1 11
0. 1
0 /
C.....CALCULATION INFORMATION
C.2.22.1 .. FDSMTH,FDTMTH
.5 .5
C.2.22.2 .. TOLDEN{.001},MAXITN{5}
0. 0
C.....OUTPUT INFORMATION
C.2.23.1 .. PRTPMP,PRTFP,PRTIC,PRTBC,PRTSLM,PRTWEL; ALL (T/F)
T T T T T T
C.2.23.2 .. IPRPTC,PRTDV(T/F) ;(0) - PRTIC [2.23.1]
101 T
C.2.23.3 .. ORENPR[I];(0) - NOT CYLIND [1.4]
13
C.2.23.4 .. PLTZON(T/F)
```

Table 8.5.--*Input-data file for example problem 1*--Continued

```
F
C.2.23.5 .. OCPLOT(T/F)
F
C-----------------------------------------------------------------------------------------
C..... TRANSIENT DATA - READ3
C.3.1 .. THRU(T/F)
F
C.....IF THRU IS TRUE PROCEED TO RECORD 3.99
C.....THE FOLLOWING IS FOR NOT THRU
C.....BOUNDARY CONDITION INFORMATION
C.....     SPECIFIED VALUE B.C.
C.3.3.1 .. RDSPBC,RDSTBC,RDSCBC,ALL(T/F);(0) - NOT CYLIND [1.4] AND NPTCBC [1.6] > 0 OR CYLIND AND NPTCBC > 1
 T F T
C.3.3.2 .. PNP B.C. BY I,J,K RANGE {0.1-0.3} ;(0) - RDSPBC [3.3.1]
1 1 1 2 1 1
25000. 1
11 11 1 2 11 11
0. 1
0 /
C.3.3.4 .. CSBC BY I,J,K RANGE {0.1-0.3}; (0) - RDSPBC [3.3.1] AND SOLUTE [1.4]
0 /
C.3.3.6 .. CNP B.C. BY I,J,K RANGE {0.1-0.3} ;(0) - RDSCBC [3.3.1] AND SOLUTE [1.4]
1 1 1 2 1 1
1. 1
0 /
C.....CALCULATION INFORMATION
C.3.7.1 .. RDCALC(T/F)
T
C.3.7.2 .. AUTOTS(T/F) ;(0) - RDCALC [3.7.1]
F
C.3.7.3.A .. DELTIM ;(0) - RDCALC [3.7.1] AND NOT AUTOTS [3.7.2]
.2
C.3.7.4 .. TIMCHG
10.
C.....OUTPUT INFORMATION
C.3.8.1 .. PRIVEL,PRIDV,PRISLM,PRIKD,PRIPTC,PRIGFB,PRIWEL,PRIBCF ; ALL [I]
8*-1
C.3.8.2 .. IPRPTC ;(0) - IF PRIPTC [3.8.1] NOT = 0
101
C.3.8.3 .. CHKPTD(T/F),NTSCHK,SAVLDO(T/F)
F /
C.....CONTOUR MAP INFORMATION
C.3.9.1 .. RDMPDT,PRTMPD; ALL (T/F)
T T
C.3.9.2 .. MAPPTC,PRIMAP[I] ;(0) - RDMPDT [3.9.1]
```

Table 8.5.--*Input-data file for example problem 1*--Continued

```
001 -1
C.3.9.3 .. YPOSUP(T/F),ZPOSUP(T/F),LENAX,LENAY,LENAZ ;(0) - RDMPDT [3.9.1]
F T 10. 0. 10.
C.3.9.4 .. IMAP1{1},IMAP2{NX},JMAP1{1},JMAP2{NY},KMAP1{1},KMAP2{NZ},AMIN,AMAX,NMPZON{5}:(0) - RDMPDT [3.9.1]
1 11 1 1 1 11 0. 1. 10
C.....ONE OF THE 3.9.4 LINES REQUIRED FOR EACH DEPENDENT VARIABLE
C.....     TO BE MAPPED
C.....END OF FIRST SET OF TRANSIENT INFORMATION
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.....READ SETS OF READ3 DATA AT EACH TIMCHG UNTIL THRU  (LINES 3.N1.N2)
C.....END OF CALCULATION LINES FOLLOW, THRU=.TRUE.
C.3.99.1 .. THRU
T
C.....TEMPORAL PLOT INFORMATION
C.3.99.2 .. PLOTWP,PLOTWT,PLOTWC; ALL (T/F)
3*F
C.....END OF DATA FILE
```

Table 8.6.--*Output file for example problem 1*

```
************************************************
*                                              *
*   THREE DIMENSIONAL FLOW, HEAT AND SOLUTE    *
*  TRANSPORT SIMULATOR - (HST3D):RELEASE - 1.0 *
*                                              *
************************************************



 EXAMPLE #1 SOLUTE TRANSPORT WITH VARIABLE DENSITY AND VISCOSITY
      DIAGONAL FLOW IN X-Z PLANE

          *** FUNDAMENTAL INFORMATION ***
CARTESIAN COORDINATES
ISOTHERMAL SIMULATION
SOLUTE TRANSPORT SIMULATION
INPUT DATA IS EXPECTED IN METRIC UNITS
SOLUTE CONCENTRATION IS EXPRESSED AS SCALED MASS FRACTION WITH RANGE (0-1)

          *** PROBLEM DIMENSIONING INFORMATION ***
NUMBER OF NODES IN X-DIRECTION ............................................. NX ...   11
NUMBER OF NODES IN Y-DIRECTION ............................................. NY ...    2
NUMBER OF NODES IN Z-DIRECTION ............................................. NZ ...   11
NUMBER OF POROUS MEDIA ZONES ............................................... NPMZ .    1
NUMBER OF SPECIFIED PRESSURE, TEMPERATURE OR MASS FRACTION B.C. ........... NPTCBC    4
NUMBER OF SPECIFIED FLUX B.C. CELLS (FLOW, HEAT OR SOLUTE).................. NFBC .    0
NUMBER OF HEAT CONDUCTION B.C. CELLS ....................................... NHCBC     0
NUMBER OF NODES OUTSIDE REGION FOR EACH HEAT CONDUCTION B.C. CELL ......... NHCN      0
NUMBER OF AQUIFER INFLUENCE FUNCTION CELLS ................................. NAIFC     0
NUMBER OF LEAKAGE CELLS .................................................... NLBC .    0
NUMBER OF WELLS ............................................................ NWEL .    0

DIRECT D4 SOLVER IS SELECTED
ABBREVIATED DIAGONAL CROSS-DISPERSIVITY COEFFICIENT STORAGE ALLOCATED
THE A4 ARRAY IN D4DES IS DIMENSIONED ....................................    3799  ELEMENTS
THE TOTAL STORAGE REQUIRED BY THE DIRECT METHOD IS .......................    5009  ELEMENTS
THE TOTAL STORAGE REQUIRED BY THE ITERATIVE METHOD IS ....................    1694  ELEMENTS
TOTAL LENGTH OF LABELED COMMON BLOCKS ....................................    6681  BYTES
                                                                  REQUIRED          COMPILED
LENGTH OF VARIABLE LENGTH REAL ARRAY  (VPA ARRAY) ..........    13076  ELEMENTS   250000  ELEMENTS
LENGTH OF VARIABLE LENGTH INTEGER ARRAY (IVPA ARRAY) .......     1779  ELEMENTS    20000  ELEMENTS
---------------------------------------------------------------------------------------------------
```

Table 8.6.--*Output file for example problem 1*--Continued

```
                          *** TIME INVARIANT OR STATIC DATA ***

                          X-DIRECTION NODE COORDINATES    ( M)

  1         2         3         4         5         6         7         8         9         10
   0.00      0.20      0.40      0.60      0.80      1.00      1.20      1.40      1.60      1.80

 11
   2.00


                          Y-DIRECTION NODE COORDINATES   ( M)

  1         2
   0.00      0.20


                          Z-DIRECTION NODE COORDINATES   ( M)

  1         2         3         4         5         6         7         8         9         10
   0.00      0.20      0.40      0.60      0.80      1.00      1.20      1.40      1.60      1.80

 11
   2.00


                          Z-AXIS IS POSITIVE VERTICALLY UPWARD

                                   ** AQUIFER PROPERTIES  **  (READ ECHO)
          REGION                POROUS MEDIUM       M.C.=MODIFICATION CODE
     I1  I2  J1  J2  K1  K2      ZONE INDEX
.................................................................................
      1  11   1   2   1  11           1
```

Table 8.6.--*Output file for example problem 1*--Continued

```
                  *** POROUS MEDIA PROPERTIES ***

                  X-DIRECTION PERMEABILITIES   ( M**2)

     1
1.0000E-08


                  Y-DIRECTION PERMEABILITIES   ( M**2)

     1
1.0000E-08


                  Z-DIRECTION PERMEABILITIES   ( M**2)

     1
1.0000E-08


                  POROSITY (-)

     1
0.1000


             *** INTERMEDIATE COMPUTED DATA ***

             X-DIRECTION CONDUCTANCE FACTOR BETWEEN X(I) AND X(I+1)   ( M**3)

        VERTICAL SLICES

                                J =  1

     1      2      3      4      5      6      7      8      9      10
```

Table 8.6.--*Output file for example problem 1*--Continued

11	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10
10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
9	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
8	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
7	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
6	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
5	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
4	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
3	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
2	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
1	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10

J = 2

	1	2	3	4	5	6	7	8	9	10
11	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10
10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
9	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
8	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
7	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
6	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
5	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
4	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
3	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
2	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
1	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10	5.0000E-10

Y-DIRECTION CONDUCTANCE FACTOR BETWEEN Y(J) AND Y(J+1) (M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
10	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
9	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
8	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
7	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
6	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
5	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
4	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
3	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09

Table 8.6.--*Output file for example problem 1*--Continued

2	1.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09	2.0000E-09
1	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09

	11
11	5.0000E-10
10	1.0000E-09
9	1.0000E-09
8	1.0000E-09
7	1.0000E-09
6	1.0000E-09
5	1.0000E-09
4	1.0000E-09
3	1.0000E-09
2	1.0000E-09
1	5.0000E-10

Z-DIRECTION CONDUCTANCE FACTOR BETWEEN Z(K) AND Z(K+1) (M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
10	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
9	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
8	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
7	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
6	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
5	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
4	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
3	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
2	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
1	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09

	11
10	5.0000E-10
9	5.0000E-10
8	5.0000E-10
7	5.0000E-10

Table 8.6.--*Output file for example problem 1*--Continued

6	5.0000E-10
5	5.0000E-10
4	5.0000E-10
3	5.0000E-10
2	5.0000E-10
1	5.0000E-10

J = 2

	1	2	3	4	5	6	7	8	9	10
10	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
9	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
8	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
7	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
6	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
5	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
4	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
3	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
2	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09
1	5.0000E-10	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09	1.0000E-09

	11
10	5.0000E-10
9	5.0000E-10
8	5.0000E-10
7	5.0000E-10
6	5.0000E-10
5	5.0000E-10
4	5.0000E-10
3	5.0000E-10
2	5.0000E-10
1	5.0000E-10

*** PROPERTIES BY POROUS MEDIUM ZONE ***

POROUS MEDIUM VERTICAL COMPRESSIBILITY (1/ PA)

1
0.0000

Table 8.6.--*Output file for example problem 1*--Continued

```
                      LONGITUDINAL DISPERSIVITY   ( M)

      1
 0.1000


                      TRANSVERSE DISPERSIVITY   ( M)

      1
 0.1000


                      DENSITY - DISTRIBUTION COEFFICIENT PRODUCT  (-)

      1
 0.0000


                      MOLECULAR DIFFUSIVITY-TORTUOSITY PRODUCT ................... DM ... 0.000      ( M**2/S)
                      SOLUTE LINEAR DECAY RATE CONSTANT .......................... DECLAM 0.000      (1/S)

                      SCALE FACTORS FOR SCALED MASS FRACTION...................... W0 ...   0.00000
                                                                                   W1 ...   0.00500

                      ATMOSPHERIC PRESSURE (ABSOLUTE) ............................ PAATM   101325.0  ( PA)
                      REFERENCE PRESSURE FOR ENTHALPY ............................ POH ..       0.0  ( PA)

                      ISOTHERMAL AQUIFER TEMPERATURE ............................. TOH ..      20.0  (DEG.C)


                                *** FLUID PROPERTIES ***
                                       PHYSICAL
                      FLUID COMPRESSIBILITY ...................................... BP ...   0.00E-01  (1/ PA)

                      REFERENCE PRESSURE FOR DENSITY ............................. P0 ..         0.0  ( PA)
                      REFERENCE TEMPERATURE FOR DENSITY .......................... T0 ..        20.0  (DEG.C)
                      FLUID DENSITY AT SOLUTE SCALED MASS FRACTION OF ZERO ....... DENF0   800.0       (KG/ M**3)
                      FLUID DENSITY AT SOLUTE SCALED MASS FRACTION OF ONE ........ DENF1   1000.       (KG/ M**3)

                                VISCOSITY-CONCENTRATION DATA TABLE AT    20.0 DEG.C
                                                  SCALED MASS FRACTION          VISCOSITY (KG/ M-S)
```

Table 8.6.--*Output file for example problem 1*--Continued

```
                                      0.00                  1.300E-03
                                      1.00                  7.000E-04

               *** INITIAL CONDITIONS ***

          INITIAL AQUIFER FLUID PRESSURE FOR HYDROSTATIC I.C. ........ PINIT.         0.0  ( PA)
          ELEVATION OF INITIAL PRESSURE .............................. ZPINIT         2.0  ( M)

          INITIAL PRESSURE DISTRIBUTION  ( PA)

     VERTICAL SLICES

                                  J =  1
```

	1	2	3	4	5	6	7	8	9	10
11	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10
10	1569.	1569.	1569.	1569.	1569.	1569.	1569.	1569.	1569.	1569.
9	3138.	3138.	3138.	3138.	3138.	3138.	3138.	3138.	3138.	3138.
8	4707.	4707.	4707.	4707.	4707.	4707.	4707.	4707.	4707.	4707.
7	6276.	6276.	6276.	6276.	6276.	6276.	6276.	6276.	6276.	6276.
6	7846.	7846.	7846.	7846.	7846.	7846.	7846.	7846.	7846.	7846.
5	9415.	9415.	9415.	9415.	9415.	9415.	9415.	9415.	9415.	9415.
4	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04
3	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04
2	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04
1	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04

	11
11	5.5746E-10
10	1569.
9	3138.
8	4707.
7	6276.
6	7846.
5	9415.
4	1.0984E+04
3	1.2553E+04
2	1.4122E+04
1	1.5691E+04

J = 2

1	2	3	4	5	6	7	8	9	10

Table 8.6.--*Output file for example problem 1*--Continued

11	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10	5.5746E-10
10	1569.	1569.	1569.	1569.	1569.	1569.	1569.	1569.	1569.	1569.
9	3138.	3138.	3138.	3138.	3138.	3138.	3138.	3138.	3138.	3138.
8	4707.	4707.	4707.	4707.	4707.	4707.	4707.	4707.	4707.	4707.
7	6276.	6276.	6276.	6276.	6276.	6276.	6276.	6276.	6276.	6276.
6	7846.	7846.	7846.	7846.	7846.	7846.	7846.	7846.	7846.	7846.
5	9415.	9415.	9415.	9415.	9415.	9415.	9415.	9415.	9415.	9415.
4	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04	1.0984E+04
3	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04	1.2553E+04
2	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04	1.4122E+04
1	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04	1.5691E+04

	11
11	5.5746E-10
10	1569.
9	3138.
8	4707.
7	6276.
6	7846.
5	9415.
4	1.0984E+04
3	1.2553E+04
2	1.4122E+04
1	1.5691E+04

INITIAL SCALED MASS FRACTIONS (-)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 8.6.--*Output file for example problem 1*--Continued

2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

	11
11	0.0000
10	0.0000
9	0.0000
8	0.0000
7	0.0000
6	0.0000
5	0.0000
4	0.0000
3	0.0000
2	0.0000
1	0.0000

J = 2

	1	2	3	4	5	6	7	8	9	10
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

	11
11	0.0000
10	0.0000
9	0.0000
8	0.0000
7	0.0000
6	0.0000
5	0.0000
4	0.0000
3	0.0000

Table 8.6.--*Output file for example problem 1*--Continued

2	0.0000
1	0.0000

INITIAL PORE VOLUME PER CELL (M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	1.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04
10	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
9	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
8	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
7	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
6	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
5	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
4	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
3	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
2	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
1	1.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04

	11
11	1.0000E-04
10	2.0000E-04
9	2.0000E-04
8	2.0000E-04
7	2.0000E-04
6	2.0000E-04
5	2.0000E-04
4	2.0000E-04
3	2.0000E-04
2	2.0000E-04
1	1.0000E-04

J = 2

	1	2	3	4	5	6	7	8	9	10
11	1.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04
10	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
9	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04

Table 8.6.--*Output file for example problem 1*--Continued

8	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
7	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
6	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
5	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
4	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
3	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
2	2.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04	4.0000E-04
1	1.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04	2.0000E-04

	11
11	1.0000E-04
10	2.0000E-04
9	2.0000E-04
8	2.0000E-04
7	2.0000E-04
6	2.0000E-04
5	2.0000E-04
4	2.0000E-04
3	2.0000E-04
2	2.0000E-04
1	1.0000E-04

INITIAL FLUID DENSITY IN CELL (KG/ M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
10	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
9	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
8	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
7	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
6	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
5	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
4	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
3	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
2	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
1	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00

Table 8.6.--*Output file for example problem 1*--Continued

	11
11	800.00
10	800.00
9	800.00
8	800.00
7	800.00
6	800.00
5	800.00
4	800.00
3	800.00
2	800.00
1	800.00

J = 2

	1	2	3	4	5	6	7	8	9	10
11	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
10	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
9	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
8	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
7	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
6	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
5	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
4	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
3	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
2	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00
1	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00	800.00

	11
11	800.00
10	800.00
9	800.00
8	800.00
7	800.00
6	800.00
5	800.00
4	800.00
3	800.00
2	800.00
1	800.00

Table 8.6.--*Output file for example problem 1*--Continued

INITIAL FLUID VISCOSITY IN CELL (KG/ M-S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
10	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
9	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
8	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
7	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
6	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
5	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
4	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
3	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
2	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
1	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03

	11
11	1.3000E-03
10	1.3000E-03
9	1.3000E-03
8	1.3000E-03
7	1.3000E-03
6	1.3000E-03
5	1.3000E-03
4	1.3000E-03
3	1.3000E-03
2	1.3000E-03
1	1.3000E-03

J = 2

	1	2	3	4	5	6	7	8	9	10
11	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
10	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
9	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
8	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
7	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
6	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03

Table 8.6.--*Output file for example problem 1*--Continued

5	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
4	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
3	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
2	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03
1	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03	1.3000E-03

	11
11	1.3000E-03
10	1.3000E-03
9	1.3000E-03
8	1.3000E-03
7	1.3000E-03
6	1.3000E-03
5	1.3000E-03
4	1.3000E-03
3	1.3000E-03
2	1.3000E-03
1	1.3000E-03

INITIAL FLUID IN REGION .... 6.400000E+01 (KG) ; 8.000000E-02 (M**3)

INITIAL SOLUTE IN REGION .... 0.000000E-01 (KG)

INDEX NUMBERS FOR SPECIFIED P,T OR C NODES

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1	1.									

Table 8.6.--*Output file for example problem 1*--Continued

```
           11
  11            3.
  10
   9
   8
   7
   6
   5
   4
   3
   2
   1

                                          J =  2

            1         2         3         4         5         6         7         8         9        10
  11
  10
   9
   8
   7
   6
   5
   4
   3
   2
   1            2.

           11
  11            4.
  10
   9
   8
   7
   6
   5
   4
   3
   2
   1
```

Table 8.6.--*Output file for example problem 1*--Continued

```
                 *** CALCULATION INFORMATION ***

             TOLERANCE FOR P,T,C ITERATION (FRACTIONAL DENSITY CHANGE) ... TOLDEN  0.0010
             MAXIMUM NUMBER OF ITERATIONS ALLOWED ON P,T,C EQUATIONS ..... MAXITN    5
             CENTERED-IN-TIME (CRANK-NICHOLSON) DIFFERENCING FOR TEMPORAL DERIVATIVE
             CENTERED-IN-SPACE DIFFERENCING FOR CONVECTIVE TERMS
             THE CROSS-DERIVATIVE HEAT AND SOLUTE FLUX TERMS WILL BE APPROXIMATED
                  BY AMPLIFYING THE DIAGONAL COEFFICIENTS OF THE DISPERSION TENSOR

------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------

                 *** TRANSIENT DATA ***

             SPECIFIED BOUNDARY PRESSURES ( PA)

        VERTICAL SLICES

                                        J =  1

        1           2           3           4           5           6           7           8           9          10
11
10
 9
 8
 7
 6
 5
 4
 3
 2
 1  2.5000E+04


       11
11  0.0000
10
 9
 8
```

Table 8.6.--*Output file for example problem 1*--Continued

```
  7
  6
  5
  4
  3
  2
  1

                                    J =  2

        1          2          3          4          5          6          7          8          9         10
 11
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1  2.5000E+04


       11
 11  0.0000
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1


                 ASSOCIATED BOUNDARY SCALED MASS FRACTIONS FOR INFLOW (-)

            VERTICAL SLICES

                                    J =  1
```

Table 8.6.--*Output file for example problem 1*--Continued

	1	2	3	4	5	6	7	8	9	10
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

	11
11	0.00000
10	
9	
8	
7	
6	
5	
4	
3	
2	
1	

J = 2

	1	2	3	4	5	6	7	8	9	10
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

Table 8.6.--*Output file for example problem 1*--Continued

```
        11
  11    0.00000
  10
   9
   8
   7
   6
   5
   4
   3
   2
   1


                SPECIFIED BOUNDARY SCALED MASS FRACTIONS (-)

            VERTICAL SLICES

                              J =  1

         1      2      3      4      5      6      7      8      9     10
  11
  10
   9
   8
   7
   6
   5
   4
   3
   2
   1    1.00000


        11
  11
  10
   9
   8
   7
   6
   5
```

Table 8.6.--*Output file for example problem 1*--Continued

```
    4
    3
    2
    1

                                        J =  2

         1        2        3        4        5        6        7        8        9       10
   11
   10
    9
    8
    7
    6
    5
    4
    3
    2
    1   1.00000


        11
   11
   10
    9
    8
    7
    6
    5
    4
    3
    2
    1


                    *** CALCULATION INFORMATION ***

          FIXED TIME STEP LENGTH ........................................ DELTIM  0.200     (S);  2.315E-06  (D)
          TIME AT WHICH NEXT SET OF TIME VARYING
               PARAMETERS WILL BE READ .................................. TIMCHG   10.0     (S);  1.157E-04  (D)

                              *** MAPPING DATA ***
```

Table 8.6.--*Output file for example problem 1*--Continued

```
-----------------------------------------------------------------------------------------
        LENGTH OF X-AXIS ..............10.0  (IN.)
        LENGTH OF Y-AXIS .............. 0.0  (IN.)
        LENGTH OF Z-AXIS ..............10.0  (IN.)
        Y-AXIS IS POSITIVE DOWNWARD
        Z-AXIS IS POSITIVE UPWARD
-----------------------------------------------------------------------------------------
    MAP NO.  I1   I2   J1   J2   K1   K2       MINIMUM VALUE    MAXIMUM VALUE   NUMBER OF
                                                OF VARIABLE      OF VARIABLE     ZONES
-----------------------------------------------------------------------------------------
          3    1   11    1    1    1   11          0.00             1.0           10

---------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------

                    *** START OF SIMULATION TIME STEP NO.   50 ***

PREPARING TO CALCULATE FOR TIME .....    10.0      (S); 1.157E-04  (D)
PROPERTIES EVALUATED AT TIME ........    9.80      (S)

                    *** INTERMEDIATE COMPUTED DATA ***

    X-DIRECTION - FLUID CONDUCTANCE BETWEEN X(I) AND X(I+1) (KG/S- PA)

VERTICAL SLICES
```

J = 1

	1	2	3	4	5	6	7	8	9	10
11	3.0804E-04	3.0826E-04	3.0857E-04	3.0894E-04	3.0937E-04	3.0985E-04	3.1036E-04	3.1082E-04	3.1111E-04	3.1129E-04
10	6.2114E-04	6.2172E-04	6.2264E-04	6.2367E-04	6.2469E-04	6.2556E-04	6.2606E-04	6.2586E-04	6.2487E-04	6.2402E-04
9	6.5091E-04	6.5097E-04	6.5093E-04	6.5053E-04	6.4952E-04	6.4747E-04	6.4389E-04	6.3852E-04	6.3228E-04	6.2790E-04
8	7.4652E-04	7.4200E-04	7.3384E-04	7.2310E-04	7.1038E-04	6.9561E-04	6.7854E-04	6.6008E-04	6.4332E-04	6.3308E-04
7	9.5025E-04	9.3328E-04	9.0359E-04	8.6606E-04	8.2429E-04	7.8020E-04	7.3532E-04	6.9304E-04	6.5929E-04	6.4045E-04
6	1.2159E-03	1.1878E-03	1.1374E-03	1.0703E-03	9.9128E-04	9.0534E-04	8.1901E-04	7.4129E-04	6.8256E-04	6.5118E-04
5	1.3823E-03	1.3628E-03	1.3229E-03	1.2602E-03	1.1718E-03	1.0584E-03	9.3039E-04	8.0881E-04	7.1586E-04	6.6631E-04
4	1.4234E-03	1.4171E-03	1.4001E-03	1.3648E-03	1.3018E-03	1.2004E-03	1.0580E-03	8.9748E-04	7.6207E-04	6.8671E-04
3	1.4281E-03	1.4267E-03	1.4212E-03	1.4050E-03	1.3675E-03	1.2950E-03	1.1729E-03	1.0010E-03	8.2402E-04	7.1332E-04
2	1.4285E-03	1.4282E-03	1.4263E-03	1.4182E-03	1.3948E-03	1.3425E-03	1.2464E-03	1.0949E-03	8.9966E-04	7.4831E-04
1	7.1428E-04	7.1421E-04	7.1361E-04	7.1064E-04	7.0108E-04	6.7855E-04	6.3621E-04	5.6935E-04	4.8063E-04	3.9647E-04

J = 2

	1	2	3	4	5	6	7	8	9	10
11	3.0804E-04	3.0826E-04	3.0857E-04	3.0894E-04	3.0937E-04	3.0985E-04	3.1036E-04	3.1082E-04	3.1111E-04	3.1129E-04

Table 8.6.--*Output file for example problem 1*--Continued

10	6.2114E-04	6.2172E-04	6.2264E-04	6.2367E-04	6.2469E-04	6.2556E-04	6.2606E-04	6.2586E-04	6.2487E-04	6.2402E-04
9	6.5091E-04	6.5097E-04	6.5093E-04	6.5053E-04	6.4952E-04	6.4747E-04	6.4389E-04	6.3852E-04	6.3228E-04	6.2790E-04
8	7.4652E-04	7.4200E-04	7.3384E-04	7.2310E-04	7.1038E-04	6.9561E-04	6.7854E-04	6.6008E-04	6.4332E-04	6.3308E-04
7	9.5025E-04	9.3328E-04	9.0359E-04	8.6606E-04	8.2429E-04	7.8020E-04	7.3532E-04	6.9304E-04	6.5929E-04	6.4045E-04
6	1.2159E-03	1.1878E-03	1.1374E-03	1.0703E-03	9.9128E-04	9.0534E-04	8.1901E-04	7.4129E-04	6.8256E-04	6.5118E-04
5	1.3823E-03	1.3628E-03	1.3229E-03	1.2602E-03	1.1718E-03	1.0584E-03	9.3039E-04	8.0881E-04	7.1586E-04	6.6631E-04
4	1.4234E-03	1.4171E-03	1.4001E-03	1.3648E-03	1.3018E-03	1.2004E-03	1.0580E-03	8.9748E-04	7.6207E-04	6.8671E-04
3	1.4281E-03	1.4267E-03	1.4212E-03	1.4050E-03	1.3675E-03	1.2950E-03	1.1729E-03	1.0010E-03	8.2402E-04	7.1332E-04
2	1.4285E-03	1.4282E-03	1.4263E-03	1.4182E-03	1.3948E-03	1.3425E-03	1.2464E-03	1.0949E-03	8.9966E-04	7.4831E-04
1	7.1428E-04	7.1421E-04	7.1361E-04	7.1064E-04	7.0108E-04	6.7855E-04	6.3621E-04	5.6935E-04	4.8063E-04	3.9647E-04

Y-DIRECTION - FLUID CONDUCTANCE BETWEEN Y(J) AND Y(J+1) (KG/S- PA)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	3.0796E-04	6.1625E-04	6.1680E-04	6.1749E-04	6.1829E-04	6.1921E-04	6.2021E-04	6.2123E-04	6.2203E-04	6.2242E-04
10	6.2098E-04	1.2426E-03	1.2443E-03	1.2463E-03	1.2484E-03	1.2504E-03	1.2519E-03	1.2523E-03	1.2511E-03	1.2484E-03
9	6.5089E-04	1.3019E-03	1.3020E-03	1.3017E-03	1.3004E-03	1.2976E-03	1.2923E-03	1.2833E-03	1.2708E-03	1.2584E-03
8	7.4773E-04	1.4906E-03	1.4774E-03	1.4581E-03	1.4345E-03	1.4072E-03	1.3755E-03	1.3391E-03	1.3016E-03	1.2719E-03
7	9.5489E-04	1.8913E-03	1.8423E-03	1.7730E-03	1.6926E-03	1.6062E-03	1.5164E-03	1.4268E-03	1.3470E-03	1.2910E-03
6	1.2236E-03	2.4166E-03	2.3356E-03	2.2163E-03	2.0687E-03	1.9016E-03	1.7260E-03	1.5565E-03	1.4137E-03	1.3189E-03
5	1.3873E-03	2.7548E-03	2.6967E-03	2.5962E-03	2.4477E-03	2.2461E-03	1.9982E-03	1.7368E-03	1.5100E-03	1.3591E-03
4	1.4247E-03	2.8442E-03	2.8243E-03	2.7765E-03	2.6839E-03	2.5267E-03	2.2842E-03	1.9656E-03	1.6452E-03	1.4156E-03
3	1.4283E-03	2.8558E-03	2.8512E-03	2.8339E-03	2.7865E-03	2.6848E-03	2.5000E-03	2.2053E-03	1.8253E-03	1.4948E-03
2	1.4285E-03	2.8570E-03	2.8559E-03	2.8492E-03	2.8238E-03	2.7560E-03	2.6166E-03	2.3774E-03	2.0233E-03	1.6099E-03
1	7.1429E-04	1.4285E-03	1.4283E-03	1.4261E-03	1.4165E-03	1.3881E-03	1.3271E-03	1.2210E-03	1.0644E-03	8.7219E-04

	11
11	3.1137E-04
10	6.2382E-04
9	6.2662E-04
8	6.3023E-04
7	6.3544E-04
6	6.4303E-04
5	6.5342E-04
4	6.6651E-04
3	6.8144E-04
2	6.9727E-04

Table 8.6.--*Output file for example problem 1*--Continued

```
 1  3.6187E-04
```

Z-DIRECTION - FLUID CONDUCTANCE BETWEEN Z(K) AND Z(K+1) (KG/S- PA)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
10	3.0922E-04	6.1877E-04	6.1946E-04	6.2031E-04	6.2123E-04	6.2219E-04	6.2307E-04	6.2369E-04	6.2378E-04	6.2332E-04
9	3.1784E-04	6.3587E-04	6.3634E-04	6.3678E-04	6.3702E-04	6.3684E-04	6.3593E-04	6.3385E-04	6.3044E-04	6.2669E-04
8	3.4845E-04	6.9582E-04	6.9287E-04	6.8835E-04	6.8255E-04	6.7542E-04	6.6647E-04	6.5538E-04	6.4305E-04	6.3255E-04
7	4.2104E-04	8.3681E-04	8.2261E-04	8.0218E-04	7.7789E-04	7.5099E-04	7.2173E-04	6.9097E-04	6.6202E-04	6.4070E-04
6	5.3846E-04	1.0651E-03	1.0337E-03	9.8823E-04	9.3338E-04	8.7242E-04	8.0813E-04	7.4480E-04	6.8987E-04	6.5243E-04
5	6.5080E-04	1.2887E-03	1.2532E-03	1.1975E-03	1.1232E-03	1.0316E-03	9.2739E-04	8.2151E-04	7.3034E-04	6.6941E-04
4	7.0289E-04	1.3995E-03	1.3797E-03	1.3420E-03	1.2809E-03	1.1901E-03	1.0671E-03	9.2300E-04	7.8775E-04	6.9348E-04
3	7.1324E-04	1.4250E-03	1.4188E-03	1.4025E-03	1.3672E-03	1.3020E-03	1.1942E-03	1.0402E-03	8.6591E-04	7.2720E-04
2	7.1421E-04	1.4282E-03	1.4268E-03	1.4208E-03	1.4025E-03	1.3600E-03	1.2786E-03	1.1445E-03	9.6028E-04	7.7539E-04
1	7.1428E-04	1.4285E-03	1.4281E-03	1.4254E-03	1.4142E-03	1.3830E-03	1.3177E-03	1.2047E-03	1.0375E-03	8.3758E-04

	11
10	3.1164E-04
9	3.1261E-04
8	3.1421E-04
7	3.1641E-04
6	3.1961E-04
5	3.2410E-04
4	3.2996E-04
3	3.3696E-04
2	3.4465E-04
1	3.5517E-04

J = 2

	1	2	3	4	5	6	7	8	9	10
10	3.0922E-04	6.1877E-04	6.1946E-04	6.2031E-04	6.2123E-04	6.2219E-04	6.2307E-04	6.2369E-04	6.2378E-04	6.2332E-04
9	3.1784E-04	6.3587E-04	6.3634E-04	6.3678E-04	6.3702E-04	6.3684E-04	6.3593E-04	6.3385E-04	6.3044E-04	6.2669E-04
8	3.4845E-04	6.9582E-04	6.9287E-04	6.8835E-04	6.8255E-04	6.7542E-04	6.6647E-04	6.5538E-04	6.4305E-04	6.3255E-04
7	4.2104E-04	8.3681E-04	8.2261E-04	8.0218E-04	7.7789E-04	7.5099E-04	7.2173E-04	6.9097E-04	6.6202E-04	6.4070E-04
6	5.3846E-04	1.0651E-03	1.0337E-03	9.8823E-04	9.3338E-04	8.7242E-04	8.0813E-04	7.4480E-04	6.8987E-04	6.5243E-04
5	6.5080E-04	1.2887E-03	1.2532E-03	1.1975E-03	1.1232E-03	1.0316E-03	9.2739E-04	8.2151E-04	7.3034E-04	6.6941E-04

Table 8.6.--*Output file for example problem 1*--Continued

4	7.0289E-04	1.3995E-03	1.3797E-03	1.3420E-03	1.2809E-03	1.1901E-03	1.0671E-03	9.2300E-04	7.8775E-04	6.9348E-04
3	7.1324E-04	1.4250E-03	1.4188E-03	1.4025E-03	1.3672E-03	1.3020E-03	1.1942E-03	1.0402E-03	8.6591E-04	7.2720E-04
2	7.1421E-04	1.4282E-03	1.4268E-03	1.4208E-03	1.4025E-03	1.3600E-03	1.2786E-03	1.1445E-03	9.6028E-04	7.7539E-04
1	7.1428E-04	1.4285E-03	1.4281E-03	1.4254E-03	1.4142E-03	1.3830E-03	1.3177E-03	1.2047E-03	1.0375E-03	8.3758E-04

	11
10	3.1164E-04
9	3.1261E-04
8	3.1421E-04
7	3.1641E-04
6	3.1961E-04
5	3.2410E-04
4	3.2996E-04
3	3.3696E-04
2	3.4465E-04
1	3.5517E-04

X-DIRECTION - SOLUTE DISPERSIVE CONDUCTANCE BETWEEN X(I) AND X(I+1) (KG/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	2.1266E-03	6.4845E-03	1.1234E-02	1.6811E-02	2.3920E-02	3.3821E-02	4.9145E-02	7.6615E-02	0.1374	0.3114
10	9.3831E-03	1.5388E-02	2.3889E-02	3.4464E-02	4.8098E-02	6.6772E-02	9.4014E-02	0.1364	0.2079	0.3609
9	1.7300E-02	2.1333E-02	2.8260E-02	3.7634E-02	4.9986E-02	6.6585E-02	8.9262E-02	0.1202	0.1611	0.2091
8	2.6750E-02	3.0016E-02	3.5810E-02	4.3767E-02	5.4269E-02	6.8151E-02	8.6094E-02	0.1079	0.1314	0.1508
7	4.0235E-02	4.3271E-02	4.7976E-02	5.3709E-02	6.1087E-02	7.1188E-02	8.4360E-02	9.9377E-02	0.1135	0.1230
6	6.1574E-02	6.4471E-02	6.7502E-02	6.9310E-02	7.1014E-02	7.5069E-02	8.2881E-02	9.2927E-02	0.1020	0.1073
5	9.4466E-02	9.5241E-02	9.4987E-02	9.1958E-02	8.6465E-02	8.1458E-02	8.1237E-02	8.6403E-02	9.2651E-02	9.6338E-02
4	0.1449	0.1366	0.1275	0.1181	0.1070	9.4150E-02	8.3015E-02	7.9424E-02	8.2387E-02	8.5477E-02
3	0.2283	0.1960	0.1659	0.1451	0.1288	0.1126	9.4428E-02	7.7743E-02	7.0227E-02	7.0709E-02
2	0.4451	0.2723	0.2096	0.1702	0.1487	0.1326	0.1153	9.2307E-02	6.5203E-02	4.7800E-02
1	0.4916	0.1597	0.1205	9.2608E-02	8.1138E-02	7.5409E-02	7.0572E-02	6.2207E-02	4.5608E-02	1.7645E-02

J = 2

	1	2	3	4	5	6	7	8	9	10
11	2.1266E-03	6.4845E-03	1.1234E-02	1.6811E-02	2.3920E-02	3.3821E-02	4.9145E-02	7.6615E-02	0.1374	0.3114
10	9.3831E-03	1.5388E-02	2.3889E-02	3.4464E-02	4.8098E-02	6.6772E-02	9.4014E-02	0.1364	0.2079	0.3609

Table 8.6.--*Output file for example problem 1*--Continued

9	1.7300E-02	2.1333E-02	2.8260E-02	3.7634E-02	4.9986E-02	6.6585E-02	8.9262E-02	0.1202	0.1611	0.2091
8	2.6750E-02	3.0016E-02	3.5810E-02	4.3767E-02	5.4269E-02	6.8151E-02	8.6094E-02	0.1079	0.1314	0.1508
7	4.0235E-02	4.3271E-02	4.7976E-02	5.3709E-02	6.1087E-02	7.1188E-02	8.4360E-02	9.9377E-02	0.1135	0.1230
6	6.1574E-02	6.4471E-02	6.7502E-02	6.9310E-02	7.1014E-02	7.5069E-02	8.2881E-02	9.2927E-02	0.1020	0.1073
5	9.4466E-02	9.5241E-02	9.4987E-02	9.1958E-02	8.6465E-02	8.1458E-02	8.1237E-02	8.6403E-02	9.2651E-02	9.6338E-02
4	0.1449	0.1366	0.1275	0.1181	0.1070	9.4150E-02	8.3015E-02	7.9424E-02	8.2387E-02	8.5477E-02
3	0.2283	0.1960	0.1659	0.1451	0.1288	0.1126	9.4428E-02	7.7743E-02	7.0227E-02	7.0709E-02
2	0.4451	0.2723	0.2096	0.1702	0.1487	0.1326	0.1153	9.2307E-02	6.5203E-02	4.7800E-02
1	0.4916	0.1597	0.1205	9.2608E-02	8.1138E-02	7.5409E-02	7.0572E-02	6.2207E-02	4.5608E-02	1.7645E-02

Y-DIRECTION - SOLUTE DISPERSIVE CONDUCTANCE BETWEEN Y(J) AND Y(J+1) (KG/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	8.9629E-14	8.6104E-03	1.7718E-02	2.8044E-02	4.0729E-02	5.7738E-02	8.2963E-02	0.1258	0.2141	0.4488
10	8.3242E-03	2.3854E-02	3.8923E-02	5.8074E-02	8.2195E-02	0.1142	0.1593	0.2260	0.3274	0.4848
9	1.6646E-02	3.7577E-02	4.8806E-02	6.5119E-02	8.6613E-02	0.1150	0.1531	0.2042	0.2713	0.3596
8	2.6217E-02	5.5782E-02	6.4874E-02	7.8492E-02	9.6639E-02	0.1205	0.1518	0.1909	0.2366	0.2846
7	3.9872E-02	8.2763E-02	9.0316E-02	0.1004	0.1132	0.1303	0.1537	0.1825	0.2131	0.2398
6	6.1793E-02	0.1258	0.1309	0.1353	0.1383	0.1437	0.1560	0.1753	0.1960	0.2118
5	9.6675E-02	0.1900	0.1881	0.1847	0.1762	0.1652	0.1599	0.1666	0.1802	0.1911
4	0.1531	0.2812	0.2589	0.2417	0.2229	0.1991	0.1741	0.1597	0.1620	0.1699
3	0.2499	0.4194	0.3483	0.3052	0.2719	0.2404	0.2056	0.1693	0.1454	0.1418
2	0.6396	0.5467	0.4702	0.3757	0.3181	0.2814	0.2481	0.2072	0.1551	0.1085
1	0.0000	0.6513	0.2802	0.2132	0.1739	0.1568	0.1463	0.1331	0.1079	6.2509E-02

	11
11	0.0000
10	0.4048
9	0.2215
8	0.1541
7	0.1238
6	0.1072
5	9.5650E-02
4	8.4157E-02
3	6.8402E-02
2	4.3339E-02
1	0.0000

Table 8.6.--*Output file for example problem 1*--Continued

Z-DIRECTION - SOLUTE DISPERSIVE CONDUCTANCE BETWEEN Z(K) AND Z(K+1) (KG/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
10	2.1119E-03	9.5084E-03	1.8033E-02	2.8037E-02	4.0362E-02	5.6722E-02	8.0326E-02	0.1180	0.1880	0.3531
9	6.2519E-03	1.5187E-02	2.1671E-02	3.0550E-02	4.1994E-02	5.7195E-02	7.8294E-02	0.1087	0.1532	0.2158
8	1.0557E-02	2.3064E-02	2.8270E-02	3.5924E-02	4.6032E-02	5.9359E-02	7.7066E-02	0.1001	0.1284	0.1590
7	1.5982E-02	3.3803E-02	3.8482E-02	4.4930E-02	5.3074E-02	6.3667E-02	7.7490E-02	9.4400E-02	0.1127	0.1290
6	2.4132E-02	5.0224E-02	5.4470E-02	5.8951E-02	6.3553E-02	6.9706E-02	7.8804E-02	9.0508E-02	0.1025	0.1118
5	3.7475E-02	7.6146E-02	7.8616E-02	7.9678E-02	7.8757E-02	7.7858E-02	8.0190E-02	8.6772E-02	9.4708E-02	0.1005
4	5.8807E-02	0.1146	0.1112	0.1066	9.9591E-02	9.0746E-02	8.3505E-02	8.2480E-02	8.6751E-02	9.0914E-02
3	9.4083E-02	0.1711	0.1532	0.1377	0.1237	0.1094	9.3965E-02	8.1496E-02	7.7688E-02	7.9561E-02
2	0.1557	0.2634	0.2095	0.1713	0.1475	0.1301	0.1127	9.2613E-02	7.3620E-02	6.4279E-02
1	0.4838	0.4533	0.2580	0.2000	0.1661	0.1485	0.1349	0.1177	9.0843E-02	5.4444E-02

	11
10	0.3147
9	0.1467
8	9.1353E-02
7	6.8849E-02
6	5.7832E-02
5	5.1310E-02
4	4.6200E-02
3	4.0449E-02
2	3.1819E-02
1	1.6073E-02

J = 2

	1	2	3	4	5	6	7	8	9	10
10	2.1119E-03	9.5084E-03	1.8033E-02	2.8037E-02	4.0362E-02	5.6722E-02	8.0326E-02	0.1180	0.1880	0.3531
9	6.2519E-03	1.5187E-02	2.1671E-02	3.0550E-02	4.1994E-02	5.7195E-02	7.8294E-02	0.1087	0 1532	0.2158
8	1.0557E-02	2.3064E-02	2.8270E-02	3.5924E-02	4.6032E-02	5.9359E-02	7.7066E-02	0.1001	0.1284	0.1590
7	1.5982E-02	3.3803E-02	3.8482E-02	4.4930E-02	5.3074E-02	6.3667E-02	7.7490E-02	9.4400E-02	0.1127	0.1290
6	2.4132E-02	5.0224E-02	5.4470E-02	5.8951E-02	6.3553E-02	6.9706E-02	7.8804E-02	9.0508E-02	0.1025	0.1118
5	3.7475E-02	7.6146E-02	7.8616E-02	7.9678E-02	7.8757E-02	7.7858E-02	8.0190E-02	8.6772E-02	9.4708E-02	0.1005
4	5.8807E-02	0.1146	0.1112	0.1066	9.9591E-02	9.0746E-02	8.3505E-02	8.2480E-02	8.6751E-02	9.0914E-02

3	9.4083E-02	0.1711	0.1532	0.1377	0.1237	0.1094	9.3965E-02	8.1496E-02	7.7688E-02	7.9561E-02
2	0.1557	0.2634	0.2095	0.1713	0.1475	0.1301	0.1127	9.2613E-02	7.3620E-02	6.4279E-02
1	0.4838	0.4533	0.2580	0.2000	0.1661	0.1485	0.1349	0.1177	9.0843E-02	5.4444E-02

	11
10	0.3147
9	0.1467
8	9.1353E-02
7	6.8849E-02
6	5.7832E-02
5	5.1310E-02
4	4.6200E-02
3	4.0449E-02
2	3.1819E-02
1	1.6073E-02

X-DIRECTION - INTERSTITIAL PORE VELOCITY BETWEEN X(I) AND X(I+1) (M/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	5.3147E-03	1.6203E-02	2.8062E-02	4.1978E-02	5.9706E-02	8.4383E-02	0.1226	0.1910	0.3425	0.7758
10	5.1193E-03	1.5614E-02	2.6965E-02	4.0079E-02	5.6353E-02	7.7922E-02	0.1080	0.1502	0.2024	0.2096
9	4.8136E-03	1.4572E-02	2.4790E-02	3.6012E-02	4.9015E-02	6.4614E-02	8.2953E-02	0.1013	0.1079	6.9241E-02
8	4.9878E-03	1.4765E-02	2.4096E-02	3.3120E-02	4.2236E-02	5.1673E-02	6.0621E-02	6.5697E-02	5.8997E-02	2.8473E-02
7	6.9004E-03	1.9542E-02	2.9470E-02	3.6502E-02	4.1330E-02	4.4676E-02	4.6368E-02	4.4429E-02	3.4949E-02	1.4342E-02
6	1.1938E-02	3.2313E-02	4.5418E-02	5.1154E-02	5.1235E-02	4.7783E-02	4.2351E-02	3.4980E-02	2.4216E-02	8.9491E-03
5	2.0321E-02	5.2459E-02	7.0599E-02	7.6288E-02	7.2688E-02	6.2804E-02	4.9649E-02	3.5753E-02	2.1836E-02	7.4139E-03
4	3.5123E-02	8.1111E-02	0.1016	0.1056	9.9527E-02	8.6266E-02	6.7538E-02	4.6358E-02	2.6284E-02	8.3966E-03
3	7.2376E-02	0.1316	0.1430	0.1368	0.1258	0.1114	9.2011E-02	6.6718E-02	3.8457E-02	1.2019E-02
2	0.1688	0.2351	0.1988	0.1674	0.1485	0.1340	0.1183	9.5473E-02	6.1827E-02	2.0481E-02
1	0.9832	0.3194	0.2411	0.1854	0.1630	0.1527	0.1452	0.1316	0.1008	4.1060E-02

J = 2

	1	2	3	4	5	6	7	8	9	10
11	5.3147E-03	1.6203E-02	2.8062E-02	4.1978E-02	5.9706E-02	8.4383E-02	0.1226	0.1910	0.3425	0.7758
10	5.1193E-03	1.5614E-02	2.6965E-02	4.0079E-02	5.6353E-02	7.7922E-02	0.1080	0.1502	0.2024	0.2096
9	4.8136E-03	1.4572E-02	2.4790E-02	3.6012E-02	4.9015E-02	6.4614E-02	8.2953E-02	0.1013	0.1079	6.9241E-02

Table 8.6.--*Output file for example problem 1*--Continued

8	4.9878E-03	1.4765E-02	2.4096E-02	3.3120E-02	4.2236E-02	5.1673E-02	6.0621E-02	6.5697E-02	5.8997E-02	2.8473E-02
7	6.9004E-03	1.9542E-02	2.9470E-02	3.6502E-02	4.1330E-02	4.4676E-02	4.6368E-02	4.4429E-02	3.4949E-02	1.4342E-02
6	1.1938E-02	3.2313E-02	4.5418E-02	5.1154E-02	5.1235E-02	4.7783E-02	4.2351E-02	3.4980E-02	2.4216E-02	8.9491E-03
5	2.0321E-02	5.2459E-02	7.0599E-02	7.6288E-02	7.2688E-02	6.2804E-02	4.9649E-02	3.5753E-02	2.1836E-02	7.4139E-03
4	3.5123E-02	8.1111E-02	0.1016	0.1056	9.9527E-02	8.6266E-02	6.7538E-02	4.6358E-02	2.6284E-02	8.3966E-03
3	7.2376E-02	0.1316	0.1430	0.1368	0.1258	0.1114	9.2011E-02	6.6718E-02	3.8457E-02	1.2019E-02
2	0.1688	0.2351	0.1988	0.1674	0.1485	0.1340	0.1183	9.5473E-02	6.1827E-02	2.0481E-02
1	0.9832	0.3194	0.2411	0.1854	0.1630	0.1527	0.1452	0.1316	0.1008	4.1060E-02

Y-DIRECTION - INTERSTITIAL PORE VELOCITY BETWEEN Y(J) AND Y(J+1) (M/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	-2.2402E-13	1.3446E-13	-2.0182E-13	2.0198E-13	-1.7970E-13	5.6216E-14	-7.8794E-14	5.6347E-14	-4.5119E-14	1.6927E-14
10	1.8026E-13	-1.3525E-13	2.7075E-13	-2.0330E-13	9.0463E-14	-1.5849E-13	4.5321E-14	-9.0667E-14	4.5300E-14	0.0000
9	-1.3981E-13	3.7285E-13	-3.0297E-13	3.4952E-13	-1.8628E-13	1.6274E-13	-2.3181E-13	1.3839E-13	-6.8714E-14	2.0470E-13
8	2.5748E-13	0.0000	3.0628E-13	-5.0562E-14	1.4991E-13	-1.4785E-13	2.9089E-13	-9.5114E-14	9.3202E-14	-9.1674E-14
7	0.0000	6.1123E-14	-2.3982E-13	1.1658E-13	-2.2538E-13	3.7965E-13	-3.1211E-13	0.0000	0.0000	4.1696E-13
6	0.0000	-2.1995E-13	2.8588E-13	-3.4366E-13	3.9182E-13	-5.5232E-13	2.2862E-13	-1.0602E-13	0.0000	-9.4088E-14
5	-4.0666E-13	3.2358E-13	-5.5724E-13	7.7364E-14	-7.4020E-14	1.3884E-13	-2.5461E-13	5.7417E-14	-3.1115E-13	1.9227E-13
4	3.3193E-13	-6.6295E-13	4.9458E-13	-1.6275E-13	3.1729E-13	-4.5482E-13	1.4059E-13	-6.2883E-14	1.1038E-13	-2.9697E-13
3	0.0000	8.3124E-13	-1.6604E-13	4.9585E-13	-1.6319E-13	9.5209E-13	-4.5120E-13	6.8479E-13	-1.1910E-13	3.0887E-13
2	1.6631E-13	-1.6630E-13	0.0000	0.0000	4.9451E-13	-8.0921E-13	-3.1128E-13	-2.8970E-13	1.2848E-13	-1.0865E-13
1	0.0000	4.9892E-13	0.0000	1.6609E-13	-4.9572E-13	0.0000	-4.7197E-13	0.0000	-1.3341E-13	3.4561E-13

	11
11	0.0000
10	3.3909E-14
9	2.2678E-14
8	0.0000
7	-2.7487E-13
6	1.8481E-13
5	-4.6734E-14
4	3.3182E-13
3	0.0000
2	2.9379E-13
1	0.0000

Table 8.6.--*Output file for example problem 1*--Continued

Z-DIRECTION - INTERSTITIAL PORE VELOCITY BETWEEN Z(K) AND Z(K+1) (M/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
10	5.2721E-03	5.4081E-03	5.9016E-03	6.9400E-03	8.8591E-03	1.2351E-02	1.9123E-02	3.4269E-02	7.5822E-02	0.2168
9	1.5484E-02	1.5907E-02	1.7325E-02	2.0207E-02	2.5363E-02	3.4215E-02	4.9494E-02	7.6856E-02	0.1282	0.2243
8	2.5468E-02	2.6055E-02	2.8004E-02	3.1947E-02	3.8890E-02	5.0277E-02	6.8156E-02	9.5312E-02	0.1348	0.1858
7	3.6571E-02	3.6981E-02	3.8489E-02	4.2033E-02	4.8865E-02	6.0284E-02	7.7367E-02	0.1004	0.1282	0.1557
6	5.1752E-02	5.1034E-02	4.9809E-02	5.0282E-02	5.4568E-02	6.4093E-02	7.9208E-02	9.8604E-02	0.1191	0.1358
5	7.6668E-02	7.2393E-02	6.3791E-02	5.6736E-02	5.5122E-02	6.0807E-02	7.3757E-02	9.1384E-02	0.1089	0.1217
4	0.1181	0.1050	8.2061E-02	6.2301E-02	5.1263E-02	5.0606E-02	6.0307E-02	7.7522E-02	9.5784E-02	0.1087
3	0.1882	0.1513	0.1024	6.5566E-02	4.4378E-02	3.6571E-02	4.0916E-02	5.6056E-02	7.6291E-02	9.2439E-02
2	0.3115	0.2152	0.1120	5.7847E-02	3.2313E-02	2.1230E-02	2.0668E-02	3.0104E-02	4.8174E-02	6.7575E-02
1	0.9677	0.1533	6.9135E-02	2.7564E-02	1.2346E-02	5.8178E-03	3.9359E-03	6.3626E-03	1.4191E-02	2.7546E-02

	11
10	0.7839
9	0.3651
8	0.2270
7	0.1707
6	0.1430
5	0.1264
4	0.1132
3	9.8536E-02
2	7.7023E-02
1	3.8579E-02

J = 2

	1	2	3	4	5	6	7	8	9	10
10	5.2721E-03	5.4081E-03	5.9016E-03	6.9400E-03	8.8591E-03	1.2351E-02	1.9123E-02	3.4269E-02	7.5822E-02	0.2168
9	1.5484E-02	1.5907E-02	1.7325E-02	2.0207E-02	2.5363E-02	3.4215E-02	4.9494E-02	7.6856E-02	0.1282	0.2243
8	2.5468E-02	2.6055E-02	2.8004E-02	3.1947E-02	3.8890E-02	5.0277E-02	6.8156E-02	9.5312E-02	0.1348	0.1858
7	3.6571E-02	3.6981E-02	3.8489E-02	4.2033E-02	4.8865E-02	6.0284E-02	7.7367E-02	0.1004	0.1282	0.1557
6	5.1752E-02	5.1034E-02	4.9809E-02	5.0282E-02	5.4568E-02	6.4093E-02	7.9208E-02	9.8604E-02	0.1191	0.1358
5	7.6668E-02	7.2393E-02	6.3791E-02	5.6736E-02	5.5122E-02	6.0807E-02	7.3757E-02	9.1384E-02	0.1089	0.1217
4	0.1181	0.1050	8.2061E-02	6.2301E-02	5.1263E-02	5.0606E-02	6.0307E-02	7.7522E-02	9.5784E-02	0.1087
3	0.1882	0.1513	0.1024	6.5566E-02	4.4378E-02	3.6571E-02	4.0916E-02	5.6056E-02	7.6291E-02	9.2439E-02

Table 8.6.--*Output file for example problem 1*--Continued

```
 2 0.3115      0.2152      0.1120      5.7847E-02  3.2313E-02  2.1230E-02  2.0668E-02  3.0104E-02  4.8174E-02  6.7575E-02
 1 0.9677      0.1533      6.9135E-02  2.7564E-02  1.2346E-02  5.8178E-03  3.9359E-03  6.3626E-03  1.4191E-02  2.7546E-02


      11
10 0.7839
 9 0.3651
 8 0.2270
 7 0.1707
 6 0.1430
 5 0.1264
 4 0.1132
 3 9.8536E-02
 2 7.7023E-02
 1 3.8579E-02


                    *** OUTPUT AT END OF TIME STEP NO.    50 ***

             TIME ...................................................  10.0     (S);     1.157E-04 (D)

          CURRENT TIME STEP LENGTH .................................  0.200    (S);     2.315E-06 (D)

          NO. OF P,T,C LOOP ITERATIONS USED .........................  2
          MAXIMUM CHANGE IN PRESSURE ................................  4.6795E+02 ( PA)   AT LOCATION ( 2, 2, 1)
          MAXIMUM CHANGE IN SCALED MASS FRACTION ....................  2.1016E-02 (-)     AT LOCATION (10, 2, 1)

             PRESSURE    ( PA)

       VERTICAL SLICES

                                       J =  1

       1           2           3           4           5           6           7           8           9           10
11 4586.2      4571.1      4525.2      4445.8      4327.7      4160.5      3925.9      3587.1      3062.0      2123.5
10 6172.7      6158.3      6114.2      6038.3      5925.9      5768.9      5553.2      5256.2      4844.6      4291.3
 9 7800.5      7787.3      7747.1      7678.9      7580.0      7446.2      7270.5      7045.8      6771.4      6478.4
 8 9494.4      9481.8      9444.3      9382.7      9297.4      9187.9      9053.1      8893.9      8719.4      8560.8
 7 11292.      11278.      11236.      11171.      11089.      10992.      10885.      10769.      10655.      10563.
 6 13217.      13196.      13139.      13058.      12961.      12859.      12757.      12661.      12576.      12514.
 5 15253.      15221.      15139.      15028.      14905.      14781.      14665.      14565.      14485.      14432.
 4 17378.      17323.      17201.      17050.      16893.      16740.      16599.      16478.      16385.      16325.
```

Table 8.6.--*Output file for example problem 1*--Continued

3	19619.	19496.	19298.	19093.	18896.	18713.	18544.	18395.	18273.	18192.
2	22135.	21737.	21407.	21131.	20894.	20682.	20487.	20305.	20144.	20023.
1	25000.	24092.	23468.	23130.	22867.	22637.	22417.	22199.	21985.	21797.

	11
11	0.00000
10	3718.6
9	6289.9
8	8483.6
7	10524.
6	12490.
5	14413.
4	16305.
3	18164.
2	19977.
1	21709.

J = 2

	1	2	3	4	5	6	7	8	9	10
11	4586.2	4571.1	4525.2	4445.8	4327.7	4160.5	3925.9	3587.1	3062.0	2123.5
10	6172.7	6158.3	6114.2	6038.3	5925.9	5768.9	5553.2	5256.2	4844.6	4291.3
9	7800.5	7787.3	7747.1	7678.9	7580.0	7446.2	7270.5	7045.8	6771.4	6478.4
8	9494.4	9481.8	9444.3	9382.7	9297.4	9187.9	9053.1	8893.9	8719.4	8560.8
7	11292.	11278.	11236.	11171.	11089.	10992.	10885.	10769.	10655.	10563.
6	13217.	13196.	13139.	13058.	12961.	12859.	12757.	12661.	12576.	12514.
5	15253.	15221.	15139.	15028.	14905.	14781.	14665.	14565.	14485.	14432.
4	17378.	17323.	17201.	17050.	16893.	16740.	16599.	16478.	16385.	16325.
3	19619.	19496.	19298.	19093.	18896.	18713.	18544.	18395.	18273.	18192.
2	22135.	21737.	21407.	21131.	20894.	20682.	20487.	20305.	20144.	20023.
1	25000.	24092.	23468.	23130.	22867.	22637.	22417.	22199.	21985.	21797.

	11
11	0.00000
10	3718.6
9	6289.9
8	8483.6
7	10524.
6	12490.
5	14413.
4	16305.

Table 8.6.--*Output file for example problem 1*--Continued

3	18164.
2	19977.
1	21709.

SOLUTE SCALED MASS FRACTION (-)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	0.00111	0.00179	0.00293	0.00435	0.00603	0.00798	0.01016	0.01238	0.01420	0.01515
10	0.01133	0.01199	0.01367	0.01577	0.01798	0.02012	0.02186	0.02258	0.02151	0.01895
9	0.06912	0.06924	0.06952	0.06946	0.06861	0.06639	0.06180	0.05359	0.04161	0.02927
8	0.23719	0.23339	0.22294	0.20745	0.18837	0.16590	0.13891	0.10664	0.07184	0.04297
7	0.53015	0.51866	0.48762	0.44239	0.38761	0.32582	0.25755	0.18448	0.11436	0.06186
6	0.82292	0.80818	0.76825	0.70692	0.62622	0.52739	0.41289	0.28967	0.17356	0.08869
5	0.96737	0.95923	0.93480	0.89107	0.82304	0.72342	0.58674	0.42091	0.25350	0.12594
4	0.99701	0.99501	0.98714	0.96779	0.92904	0.85975	0.74331	0.56766	0.35659	0.17596
3	0.99978	0.99948	0.99770	0.99088	0.97181	0.92938	0.84746	0.70243	0.48055	0.24235
2	0.99999	0.99994	0.99951	0.99693	0.98686	0.95926	0.89977	0.78922	0.60195	0.33220
1	1.00000	0.99999	0.99978	0.99813	0.99050	0.96757	0.91610	0.81989	0.66059	0.42791

	11
11	0.01601
10	0.01823
9	0.02413
8	0.03158
7	0.04223
6	0.05746
5	0.07785
4	0.10288
3	0.13074
2	0.15963
1	0.20642

J = 2

	1	2	3	4	5	6	7	8	9	10
11	0.00111	0.00179	0.00293	0.00435	0.00603	0.00798	0.01016	0.01238	0.01420	0.01515
10	0.01133	0.01199	0.01367	0.01577	0.01798	0.02012	0.02186	0.02258	0.02151	0.01895

Table 8.6.--*Output file for example problem 1*--Continued

9	0.06912	0.06924	0.06952	0.06946	0.06861	0.06639	0.06180	0.05359	0.04161	0.02927
8	0.23719	0.23339	0.22294	0.20745	0.18837	0.16590	0.13891	0.10664	0.07184	0.04297
7	0.53015	0.51866	0.48762	0.44239	0.38761	0.32582	0.25755	0.18448	0.11436	0.06186
6	0.82292	0.80818	0.76825	0.70692	0.62622	0.52739	0.41289	0.28967	0.17356	0.08869
5	0.96737	0.95923	0.93480	0.89107	0.82304	0.72342	0.58674	0.42091	0.25350	0.12594
4	0.99701	0.99501	0.98714	0.96779	0.92904	0.85975	0.74331	0.56766	0.35659	0.17596
3	0.99978	0.99948	0.99770	0.99088	0.97181	0.92938	0.84746	0.70243	0.48055	0.24235
2	0.99999	0.99994	0.99951	0.99693	0.98686	0.95926	0.89977	0.78922	0.60195	0.33220
1	1.00000	0.99999	0.99978	0.99813	0.99050	0.96757	0.91610	0.81989	0.66059	0.42791

	11
11	0.01601
10	0.01823
9	0.02413
8	0.03158
7	0.04223
6	0.05746
5	0.07785
4	0.10288
3	0.13074
2	0.15963
1	0.20642

DENSITY (KG/ M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	800.22	800.36	800.59	800.87	801.21	801.60	802.03	802.48	802.84	803.03
10	802.27	802.40	802.73	803.15	803.60	804.02	804.37	804.52	804.30	803.79
9	813.82	813.85	813.90	813.89	813.72	813.28	812.36	810.72	808.32	805.85
8	847.44	846.68	844.59	841.49	837.67	833.18	827.78	821.33	814.37	808.59
7	906.03	903.73	897.52	888.48	877.52	865.16	851.51	836.90	822.87	812.37
6	964.58	961.64	953.65	941.38	925.24	905.48	882.58	857.93	834.71	817.74
5	993.47	991.85	986.96	978.21	964.61	944.68	917.35	884.18	850.70	825.19
4	999.40	999.00	997.43	993.56	985.81	971.95	948.66	913.53	871.32	835.19
3	999.96	999.90	999.54	998.18	994.36	985.88	969.49	940.49	896.11	848.47
2	1000.0	999.99	999.90	999.39	997.37	991.85	979.95	957.84	920.39	866.44
1	1000.0	1000.0	999.96	999.63	998.10	993.51	983.22	963.98	932.12	885.58

Table 8.6.--*Output file for example problem 1*--Continued

	11
11	803.20
10	803.65
9	804.83
8	806.32
7	808.45
6	811.49
5	815.57
4	820.58
3	826.15
2	831.93
1	841.28

J = 2

	1	2	3	4	5	6	7	8	9	10
11	800.22	800.36	800.59	800.87	801.21	801.60	802.03	802.48	802.84	803.03
10	802.27	802.40	802.73	803.15	803.60	804.02	804.37	804.52	804.30	803.79
9	813.82	813.85	813.90	813.89	813.72	813.28	812.36	810.72	808.32	805.85
8	847.44	846.68	844.59	841.49	837.67	833.18	827.78	821.33	814.37	808.59
7	906.03	903.73	897.52	888.48	877.52	865.16	851.51	836.90	822.87	812.37
6	964.58	961.64	953.65	941.38	925.24	905.48	882.58	857.93	834.71	817.74
5	993.47	991.85	986.96	978.21	964.61	944.68	917.35	884.18	850.70	825.19
4	999.40	999.00	997.43	993.56	985.81	971.95	948.66	913.53	871.32	835.19
3	999.96	999.90	999.54	998.18	994.36	985.88	969.49	940.49	896.11	848.47
2	1000.0	999.99	999.90	999.39	997.37	991.85	979.95	957.84	920.39	866.44
1	1000.0	1000.0	999.96	999.63	998.10	993.51	983.22	963.98	932.12	885.58

	11
11	803.20
10	803.65
9	804.83
8	806.32
7	808.45
6	811.49
5	815.57
4	820.58
3	826.15
2	831.93
1	841.28

Table 8.6.--*Output file for example problem 1*--Continued

VISCOSITY (KG/ M-S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	1.2991E-03	1.2986E-03	1.2976E-03	1.2965E-03	1.2952E-03	1.2936E-03	1.2919E-03	1.2901E-03	1.2886E-03	1.2879E-03
10	1.2909E-03	1.2904E-03	1.2890E-03	1.2874E-03	1.2856E-03	1.2839E-03	1.2825E-03	1.2820E-03	1.2828E-03	1.2848E-03
9	1.2455E-03	1.2455E-03	1.2452E-03	1.2453E-03	1.2459E-03	1.2477E-03	1.2512E-03	1.2576E-03	1.2669E-03	1.2767E-03
8	1.1225E-03	1.1251E-03	1.1324E-03	1.1433E-03	1.1569E-03	1.1731E-03	1.1929E-03	1.2170E-03	1.2435E-03	1.2659E-03
7	9.3630E-04	9.4299E-04	9.6128E-04	9.8857E-04	1.0227E-03	1.0625E-03	1.1084E-03	1.1597E-03	1.2111E-03	1.2512E-03
6	7.8110E-04	7.8826E-04	8.0799E-04	8.3925E-04	8.8224E-04	9.3790E-04	1.0068E-03	1.0866E-03	1.1676E-03	1.2305E-03
5	7.1428E-04	7.1789E-04	7.2883E-04	7.4883E-04	7.8104E-04	8.3072E-04	9.0407E-04	1.0018E-03	1.1112E-03	1.2025E-03
4	7.0129E-04	7.0216E-04	7.0560E-04	7.1410E-04	7.3144E-04	7.6349E-04	8.2055E-04	9.1481E-04	1.0425E-03	1.1658E-03
3	7.0009E-04	7.0023E-04	7.0100E-04	7.0396E-04	7.1232E-04	7.3128E-04	7.6932E-04	8.4159E-04	9.6550E-04	1.1189E-03
2	7.0001E-04	7.0003E-04	7.0021E-04	7.0133E-04	7.0572E-04	7.1788E-04	7.4481E-04	7.9756E-04	8.9559E-04	1.0584E-03
1	7.0000E-04	7.0001E-04	7.0010E-04	7.0081E-04	7.0413E-04	7.1420E-04	7.3732E-04	7.8256E-04	8.6367E-04	9.9747E-04

	11
11	1.2872E-03
10	1.2854E-03
9	1.2807E-03
8	1.2748E-03
7	1.2665E-03
6	1.2546E-03
5	1.2388E-03
4	1.2198E-03
3	1.1989E-03
2	1.1777E-03
1	1.1441E-03

J = 2

	1	2	3	4	5	6	7	8	9	10
11	1.2991E-03	1.2986E-03	1.2976E-03	1.2965E-03	1.2952E-03	1.2936E-03	1.2919E-03	1.2901E-03	1.2886E-03	1.2879E-03
10	1.2909E-03	1.2904E-03	1.2890E-03	1.2874E-03	1.2856E-03	1.2839E-03	1.2825E-03	1.2820E-03	1.2828E-03	1.2848E-03
9	1.2455E-03	1.2455E-03	1.2452E-03	1.2453E-03	1.2459E-03	1.2477E-03	1.2512E-03	1.2576E-03	1.2669E-03	1.2767E-03
8	1.1225E-03	1.1251E-03	1.1324E-03	1.1433E-03	1.1569E-03	1.1731E-03	1.1929E-03	1.2170E-03	1.2435E-03	1.2659E-03
7	9.3630E-04	9.4299E-04	9.6128E-04	9.8857E-04	1.0227E-03	1.0625E-03	1.1084E-03	1.1597E-03	1.2111E-03	1.2512E-03

Table 8.6.--*Output file for example problem 1*--Continued

6	7.8110E-04	7.8826E-04	8.0799E-04	8.3925E-04	8.8224E-04	9.3790E-04	1.0068E-03	1.0866E-03	1.1676E-03	1.2305E-03
5	7.1428E-04	7.1789E-04	7.2883E-04	7.4883E-04	7.8104E-04	8.3072E-04	9.0407E-04	1.0018E-03	1.1112E-03	1.2025E-03
4	7.0129E-04	7.0216E-04	7.0560E-04	7.1410E-04	7.3144E-04	7.6349E-04	8.2055E-04	9.1481E-04	1.0425E-03	1.1658E-03
3	7.0009E-04	7.0023E-04	7.0100E-04	7.0396E-04	7.1232E-04	7.3128E-04	7.6932E-04	8.4159E-04	9.6550E-04	1.1189E-03
2	7.0001E-04	7.0003E-04	7.0021E-04	7.0133E-04	7.0572E-04	7.1788E-04	7.4481E-04	7.9756E-04	8.9559E-04	1.0584E-03
1	7.0000E-04	7.0001E-04	7.0010E-04	7.0081E-04	7.0413E-04	7.1420E-04	7.3732E-04	7.8256E-04	8.6367E-04	9.9747E-04

	11
11	1.2872E-03
10	1.2854E-03
9	1.2807E-03
8	1.2748E-03
7	1.2665E-03
6	1.2546E-03
5	1.2388E-03
4	1.2198E-03
3	1.1989E-03
2	1.1777E-03
1	1.1441E-03

```
*** GLOBAL FLOW BALANCE SUMMARY ***
   CURRENT TIME STEP                              RATES                  AMOUNTS

FLUID INFLOW ...............................  3.245287E+00  (KG/S)    6.490574E-01  (KG)
FLUID OUTFLOW ..............................  2.580780E+00  (KG/S)    5.161559E-01  (KG)
CHANGE IN FLUID IN REGION ..................  6.645022E-01  (KG/S)    1.329004E-01  (KG)
RESIDUAL IMBALANCE ......................... -5.047900E-06  (KG/S)   -1.009580E-06  (KG)
FRACTIONAL IMBALANCE .......................                            0.0000

SOLUTE INFLOW ..............................  1.622645E-02  (KG/S)    3.245291E-03  (KG)
SOLUTE OUTFLOW .............................  1.915943E-04  (KG/S)    3.831887E-05  (KG)
CHANGE IN SOLUTE IN REGION .................  1.603472E-02  (KG/S)    3.206945E-03  (KG)

RESIDUAL IMBALANCE ......................... -1.366906E-07  (KG/S)   -2.733812E-08  (KG)
FRACTIONAL IMBALANCE .......................                            0.0000

   CUMULATIVE SUMMARY
                                                 AMOUNTS

FLUID INFLOW ...............................  3.325127E+01  (KG)
FLUID OUTFLOW ..............................  2.626598E+01  (KG)
```

Table 8.6.--*Output file for example problem 1*--Continued

```
CHANGE IN FLUID IN REGION ................................  6.985026E+00  (KG)
FLUID IN REGION ..........................................  7.098503E+01  (KG)
RESIDUAL IMBALANCE ....................................... -2.603850E-04  (KG)
FRACTIONAL IMBALANCE .....................................      0.0000

SOLUTE INFLOW ............................................  1.665237E-01  (KG)
SOLUTE OUTFLOW ...........................................  2.253053E-04  (KG)
CHANGE IN SOLUTE IN REGION ...............................  1.663889E-01  (KG)
SOLUTE IN REGION .........................................  1.663889E-01  (KG)
RESIDUAL IMBALANCE .......................................  9.057561E-05  (KG)
FRACTIONAL IMBALANCE .....................................      0.0005

CUMULATIVE SPECIFIED P CELL FLUID NET INFLOW ..............  6.985286E+00  (KG)
CUMULATIVE FLUX B.C. FLUID NET INFLOW .....................  0.000000E-01  (KG)
CUMULATIVE LEAKAGE B.C. FLUID NET INFLOW ..................  0.000000E-01  (KG)
CUMULATIVE AQUIFER INFLUENCE FLUID NET INFLOW .............  0.000000E-01  (KG)


CUMULATIVE SPECIFIED C CELL OR ASSOCIATED WITH
     SPECIFIED P CELL SOLUTE NET INFLOW....................  1.662984E-01  (KG)
CUMULATIVE FLUX B.C. SOLUTE NET INFLOW ....................  0.000000E-01  (KG)
CUMULATIVE LEAKAGE B.C. SOLUTE NET INFLOW .................  0.000000E-01  (KG)
CUMULATIVE AQUIFER INFLUENCE SOLUTE NET INFLOW ............  0.000000E-01  (KG)


               SPECIFIED PRESSURE, TEMPERATURE, OR MASS FRACTION B.C. FLOW RATES
                               POSITIVE IS INTO THE REGION
FLUID   (KG/S)

VERTICAL SLICES

                                      J =  1

        1         2         3         4         5         6         7         8         9        10
 11
 10
  9
  8
  7
  6
  5
  4
  3
  2
```

Table 8.6.--*Output file for example problem 1*--Continued

```
  1   1.623


       11
 11  -1.290
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1

                                    J =  2

        1        2        3        4        5        6        7        8        9       10
 11
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1   1.623


       11
 11  -1.290
 10
  9
  8
  7
  6
  5
  4
  3
  2
```

Table 8.6.--*Output file for example problem 1*--Continued

```
1


               VERTICAL SLICES

               ASSOCIATED SOLUTE   (KG/S)

               VERTICAL SLICES

                                         J =  1

       1         2         3         4         5         6         7         8         9        10
 11
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1


      11
 11 -9.5797E-05
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1

                                         J =  2

       1         2         3         4         5         6         7         8         9        10
 11
```

Table 8.6.--*Output file for example problem 1*--Continued

```
10
 9
 8
 7
 6
 5
 4
 3
 2
 1

       11
11 -9.5797E-05
10
 9
 8
 7
 6
 5
 4
 3
 2
 1

             SOLUTE   (KG/S)

             VERTICAL SLICES

                                         J =  1

       1          2          3          4          5          6          7          8          9         10
11
10
 9
 8
 7
 6
 5
 4
 3
 2
 1  8.1132E-03
```

Table 8.6.--*Output file for example problem 1*--Continued

	11
11	
10	
9	
8	
7	
6	
5	
4	
3	
2	
1	

J = 2

	1	2	3	4	5	6	7	8	9	10
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1	8.1132E-03									

	11
11	
10	
9	
8	
7	
6	
5	
4	
3	
2	
1	

Table 8.6.--*Output file for example problem 1*--Continued

VERTICAL PLANE AT ROW NO. 1

VERTICAL CROSS-SECTION OF SIMULATION REGION -   MASS FRACTIONS   PAGE 1

```
*         *         *         *         *         *         *         *         *         *         *





*                                                                                                   *





*                                                                                                   *

11111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111
111111111111111111111111111111111111111111111111111111111111111111
                  1111111111111111111111111111111111111111111111111111
*                                 11111111111111111111111111111111111111                            *
                                                111111111111111111111111111
2222222222222222222222222                              11111111111111111111111
2222222222222222222222222222222222222                       1111111111111111111
                    2222222222222222222222222                   11111111111111111
                               2222222222222222222                1111111111111111
*3333333333333333                     2222222222222222              111111111111111                 *
3333333333333333333333333333                 2222222222222             11111111111111
               33333333333333333333               222222222222           1111111111111
                          333333333333333            22222222222          1111111111111
44444444444444                   3333333333333           222222222          111111111111
444444444444444444444444               33333333333         222222222         111111111111
*            444444444444444444            3333333333         22222222        11111111111           *
                   4444444444444444           3333333333        2222222        11111111111
                         44444444444444           33333333        2222222        1111111111
                              4444444444444          3333333       2222222        1111111111
55555555555555                     44444444444         3333333       2222222       11111111111
```

Table 8.6.--*Output file for example problem 1*--Continued

```
5555555555555555555555                    44444444444      3333333      2222222       11111111111
*555555555555555555555555555                  444444444      333333      222222       11111111111    *
5555555555555555555555555555555                 444444444      333333     222222       11111111111
555555555555555555555555555555555                444444444      333333     222222       11111111111
555555555555555555555555555555555555              44444444      333333     222222      111111111111
555555555555555555555555555555555555555             44444444     33333      22222      111111111111
555555555555555555555555555555555555555555            4444444     333333     22222      11111111111
*55555555555555555555555555555555555555555555           4444444    33333    222222      11111111111*
5555555555555555555555555555555555555555555555          4444444    33333    222222     11111111111
555555555555555555555555555555555555555555555555         4444444    33333    22222     11111111111
55555555555555555555555555555555555555555555555555        444444    33333    22222      111111111
555555555555555555555555555555555555555555555555555        4444444    33333    22222      11111111
55555555555555555555555555555555555555555555555555555       444444    33333    2222      11111111
*55555555555555555555555555555555555555555555555555555      4444444    33333    2222       111111*
5555555555555555555555555555555555555555555555555555555      444444    33333    2222       111111
55555555555555555555555555555555555555555555555555555555     4444444    3333    2222        11111
555555555555555555555555555555555555555555555555555555555     4444444    3333    2222        1111
5555555555555555555555555555555555555555555555555555555555     4444444    3333    2222        1111
55555555555555555555555555555555555555555555555555555555555     444444    3333   22222        111
*55555555555555555555555555555555555555555555555555555555555     44444     333    2222        11*
5555555555555555555555555555555555555555555555555555555555555     444444    3333    2222        11
5555555555555555555555555555555555555555555555555555555555555     444444    3333   22222        11
5555555555555555555555555555555555555555555555555555555555555      444444    3333   22222        1
55555555555555555555555555555555555555555555555555555555555555      444444    3333   22222        1
55555555555555555555555555555555555555555555555555555555555555      4444444    3333   22222       1
*555555555*555555555*555555555*555555555*555555555*555555555*5      * 444444  *  3333   * 2222    *
```

```
              MAP LEGEND
HORIZONTAL GRID NODE RANGE, FROM    1 TO 11
VERTICAL GRID NODE RANGE, FROM    1 TO 11
DEPENDENT VARIABLE RANGE    MAP CHARACTER

   0.000E-01 -  1.000E-01
   1.000E-01 -  2.000E-01        1
   2.000E-01 -  3.000E-01
   3.000E-01 -  4.000E-01        2
   4.000E-01 -  5.000E-01
   5.000E-01 -  6.000E-01        3
   6.000E-01 -  7.000E-01
   7.000E-01 -  8.000E-01        4
   8.000E-01 -  9.000E-01
   9.000E-01 -  1.000E+00        5
```

Table 8.6.--*Output file for example problem 1*--Continued

GRID NODE LOCATIONS ACROSS THE PAGE

0.000 0.200 0.400 0.600 0.800 1.00 1.20 1.40 1.60 1.80 2.00

GRID NODE LOCATIONS ALONG THE PAGE

AXIS IS POSITIVE UP THE PAGE

0.000 0.200 0.400 0.600 0.800 1.00 1.20 1.40 1.60 1.80 2.00

X-DIRECTION - INTERSTITIAL PORE VELOCITY BETWEEN X(I) AND X(I+1) (M/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	5.8050E-03	1.7689E-02	3.0581E-02	4.5596E-02	6.4559E-02	9.0775E-02	0.1312	0.2036	0.3643	0.8247
10	5.6043E-03	1.7088E-02	2.9465E-02	4.3659E-02	6.1105E-02	8.4037E-02	0.1158	0.1605	0.2155	0.2228
9	5.3289E-03	1.6128E-02	2.7400E-02	3.9668E-02	5.3692E-02	7.0286E-02	8.9582E-02	0.1087	0.1152	7.3718E-02
8	5.6099E-03	1.6607E-02	2.7078E-02	3.7094E-02	4.6992E-02	5.6947E-02	6.6094E-02	7.0920E-02	6.3208E-02	3.0376E-02
7	7.7525E-03	2.1933E-02	3.3082E-02	4.0965E-02	4.6238E-02	4.9597E-02	5.0885E-02	4.8175E-02	3.7551E-02	1.5336E-02
6	1.3214E-02	3.5479E-02	4.9678E-02	5.5950E-02	5.6113E-02	5.2287E-02	4.6079E-02	3.7726E-02	2.5927E-02	9.5482E-03
5	2.2395E-02	5.6518E-02	7.5087E-02	8.0709E-02	7.6885E-02	6.6560E-02	5.2617E-02	3.7758E-02	2.2978E-02	7.7931E-03
4	3.9561E-02	8.6821E-02	0.1060	0.1089	0.1023	8.8744E-02	6.9581E-02	4.7723E-02	2.6993E-02	8.6132E-03
3	8.7745E-02	0.1410	0.1461	0.1387	0.1271	0.1123	9.2878E-02	6.7429E-02	3.8829E-02	1.2105E-02
2	0.2840	0.2362	0.1963	0.1687	0.1487	0.1337	0.1178	9.5323E-02	6.1982E-02	2.0557E-02
1	0.6489	0.4450	0.2414	0.1873	0.1625	0.1512	0.1434	0.1303	0.1006	4.1525E-02

J = 2

	1	2	3	4	5	6	7	8	9	10
11	5.8050E-03	1.7689E-02	3.0581E-02	4.5596E-02	6.4559E-02	9.0775E-02	0.1312	0.2036	0.3643	0.8247
10	5.6043E-03	1.7088E-02	2.9465E-02	4.3659E-02	6.1105E-02	8.4037E-02	0.1158	0.1605	0.2155	0.2228
9	5.3289E-03	1.6128E-02	2.7400E-02	3.9668E-02	5.3692E-02	7.0286E-02	8.9582E-02	0.1087	0.1152	7.3718E-02
8	5.6099E-03	1.6607E-02	2.7078E-02	3.7094E-02	4.6992E-02	5.6947E-02	6.6094E-02	7.0920E-02	6.3208E-02	3.0376E-02
7	7.7525E-03	2.1933E-02	3.3082E-02	4.0965E-02	4.6238E-02	4.9597E-02	5.0885E-02	4.8175E-02	3.7551E-02	1.5336E-02
6	1.3214E-02	3.5479E-02	4.9678E-02	5.5950E-02	5.6113E-02	5.2287E-02	4.6079E-02	3.7726E-02	2.5927E-02	9.5482E-03
5	2.2395E-02	5.6518E-02	7.5087E-02	8.0709E-02	7.6885E-02	6.6560E-02	5.2617E-02	3.7758E-02	2.2978E-02	7.7931E-03
4	3.9561E-02	8.6821E-02	0.1060	0.1089	0.1023	8.8744E-02	6.9581E-02	4.7723E-02	2.6993E-02	8.6132E-03
3	8.7745E-02	0.1410	0.1461	0.1387	0.1271	0.1123	9.2878E-02	6.7429E-02	3.8829E-02	1.2105E-02
2	0.2840	0.2362	0.1963	0.1687	0.1487	0.1337	0.1178	9.5323E-02	6.1982E-02	2.0557E-02
1	0.6489	0.4450	0.2414	0.1873	0.1625	0.1512	0.1434	0.1303	0.1006	4.1525E-02

Table 8.6.--*Output file for example problem 1*--Continued

Y-DIRECTION - INTERSTITIAL PORE VELOCITY BETWEEN Y(J) AND Y(J+1) (M/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	2.2403E-13	-1.3447E-13	2.0185E-13	-2.2448E-13	1.7977E-13	-6.7495E-14	9.0115E-14	-6.7679E-14	5.6463E-14	-2.2599E-14
10	-1.8036E-13	1.5788E-13	-2.9351E-13	2.0346E-13	-1.1319E-13	1.8135E-13	-4.5385E-14	9.0811E-14	-6.8063E-14	2.2652E-14
9	1.4020E-13	-3.7389E-13	3.0384E-13	-3.7394E-13	1.8687E-13	-1.8661E-13	2.5587E-13	-1.3886E-13	6.8915E-14	-2.2797E-13
8	-2.5928E-13	5.1735E-14	-3.0841E-13	5.0911E-14	-1.5094E-13	1.4885E-13	-3.4157E-13	9.5661E-14	-9.3623E-14	9.1965E-14
7	0.0000	-6.1727E-14	2.4221E-13	-1.1776E-13	2.8459E-13	-4.3825E-13	3.1509E-13	0.0000	0.0000	-4.1871E-13
6	0.0000	2.2153E-13	-2.8816E-13	3.4678E-13	-3.9586E-13	5.5855E-13	-2.3126E-13	1.0714E-13	0.0000	1.4191E-13
5	4.0746E-13	-3.2432E-13	5.5905E-13	-7.7731E-14	7.4526E-14	-1.4014E-13	2.5754E-13	-5.8103E-14	3.1430E-13	-1.9362E-13
4	-3.3200E-13	8.2898E-13	-4.9497E-13	1.6302E-13	-3.1832E-13	6.0991E-13	-1.4187E-13	0.0000	-1.1167E-13	3.4950E-13
3	0.0000	-8.3127E-13	1.6607E-13	-4.9611E-13	1.6343E-13	-9.5516E-13	4.5397E-13	-6.9164E-13	1.2058E-13	-3.1214E-13
2	-1.6631E-13	1.6630E-13	0.0000	0.0000	-4.9488E-13	8.1083E-13	3.1261E-13	2.9193E-13	-1.2999E-13	1.1000E-13
1	0.0000	-4.9892E-13	0.0000	-1.6612E-13	6.6133E-13	0.0000	4.7367E-13	0.0000	1.3479E-13	-3.5013E-13

	11
11	0.0000
10	-3.3963E-14
9	-2.2724E-14
8	0.0000
7	2.7577E-13
6	-1.8559E-13
5	4.6986E-14
4	-3.3404E-13
3	0.0000
2	-2.9655E-13
1	0.0000

Z-DIRECTION - INTERSTITIAL PORE VELOCITY BETWEEN Z(K) AND Z(K+1) (M/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
10	5.7866E-03	5.9248E-03	6.4297E-03	7.4921E-03	9.4663E-03	1.3093E-02	2.0197E-02	3.6207E-02	8.0299E-02	0.2301
9	1.6924E-02	1.7337E-02	1.8738E-02	2.1623E-02	2.6865E-02	3.6005E-02	5.1998E-02	8.0921E-02	0.1354	0.2376

Table 8.6.--*Output file for example problem 1*--Continued

8	2.7306E-02	2.7862E-02	2.9744E-02	3.3654E-02	4.0699E-02	5.2475E-02	7.1243E-02	0.1000	0.1420	0.1963
7	3.7916E-02	3.8247E-02	3.9625E-02	4.3149E-02	5.0198E-02	6.2180E-02	8.0286E-02	0.1049	0.1345	0.1638
6	5.2533E-02	5.1521E-02	4.9879E-02	5.0235E-02	5.4842E-02	6.5113E-02	8.1361E-02	0.1022	0.1241	0.1420
5	7.7592E-02	7.2491E-02	6.2906E-02	5.5513E-02	5.4215E-02	6.0676E-02	7.4727E-02	9.3692E-02	0.1126	0.1263
4	0.1207	0.1052	8.0368E-02	6.0270E-02	4.9664E-02	4.9680E-02	6.0153E-02	7.8395E-02	9.7907E-02	0.1118
3	0.1994	0.1516	9.8869E-02	6.3029E-02	4.2766E-02	3.5583E-02	4.0303E-02	5.5857E-02	7.6984E-02	9.4281E-02
2	0.3963	0.2001	0.1053	5.6562E-02	3.1188E-02	2.0501E-02	2.0146E-02	2.9606E-02	4.7945E-02	6.8290E-02
1	0.6455	0.2806	7.1863E-02	2.7406E-02	1.1437E-02	5.0341E-03	3.4014E-03	5.9634E-03	1.3839E-02	2.7596E-02

	11
10	0.8329
9	0.3873
8	0.2401
7	0.1798
6	0.1497
5	0.1313
4	0.1168
3	0.1009
2	7.8480E-02
1	3.9222E-02

J = 2

	1	2	3	4	5	6	7	8	9	10
10	5.7866E-03	5.9248E-03	6.4297E-03	7.4921E-03	9.4663E-03	1.3093E-02	2.0197E-02	3.6207E-02	8.0299E-02	0.2301
9	1.6924E-02	1.7337E-02	1.8738E-02	2.1623E-02	2.6865E-02	3.6005E-02	5.1998E-02	8.0921E-02	0.1354	0.2376
8	2.7306E-02	2.7862E-02	2.9744E-02	3.3654E-02	4.0699E-02	5.2475E-02	7.1243E-02	0.1000	0.1420	0.1963
7	3.7916E-02	3.8247E-02	3.9625E-02	4.3149E-02	5.0198E-02	6.2180E-02	8.0286E-02	0.1049	0.1345	0.1638
6	5.2533E-02	5.1521E-02	4.9879E-02	5.0235E-02	5.4842E-02	6.5113E-02	8.1361E-02	0.1022	0.1241	0.1420
5	7.7592E-02	7.2491E-02	6.2906E-02	5.5513E-02	5.4215E-02	6.0676E-02	7.4727E-02	9.3692E-02	0.1126	0.1263
4	0.1207	0.1052	8.0368E-02	6.0270E-02	4.9664E-02	4.9680E-02	6.0153E-02	7.8395E-02	9.7907E-02	0.1118
3	0.1994	0.1516	9.8869E-02	6.3029E-02	4.2766E-02	3.5583E-02	4.0303E-02	5.5857E-02	7.6984E-02	9.4281E-02
2	0.3963	0.2001	0.1053	5.6562E-02	3.1188E-02	2.0501E-02	2.0146E-02	2.9606E-02	4.7945E-02	6.8290E-02
1	0.6455	0.2806	7.1863E-02	2.7406E-02	1.1437E-02	5.0341E-03	3.4014E-03	5.9634E-03	1.3839E-02	2.7596E-02

	11
10	0.8329
9	0.3873
8	0.2401
7	0.1798
6	0.1497

Table 8.6.--*Output file for example problem 1*--Continued

```
5  0.1313
4  0.1168
3  0.1009
2  7.8480E-02
1  3.9222E-02


                   ***** JOB COMPLETED *****
LAST TIME PLANE CALCULATED .........................................     10.      (S)
LAST TIME PLANE INDEX ..............................................       50
```

8.2.2. Heat Transport with Variable Density and Variable Viscosity

The second example problem is based upon thermal injection of a hot fluid through a well followed by production for an equivalent time period. The cylindrical coordinate system is used. Injection is for 90 days, followed by 90 days of withdrawal. Temperature of the injected fluid is 60 °C; flow rate is 203 ℓ/s for both injection and withdrawal. Initial temperature in the aquifer is 20 °C. The other parameters are:

Well radius, 1 m
Region outer radius, 246 m
Aquifer thickness, 30 m
Permeability, 1.02×10^{-10} m^2
Porosity, 0.35
Fluid compressibility, 0. Pa^{-1}
Porous matrix compressibility, 0. Pa^{-1}
Fluid heat capacity, 4,182 J/kg-°C
Porous-medium heat capacity, 840 J/kg-°C
Fluid density at 20 °C, 1,000 kg/m^3
Porous-medium density of the solid, 2,650 kg/m^3
Thermal conductivity of the fluid, 0.6 W/m-°C
Thermal conductivity of the porous medium, 3.5 W/m-°C
Coefficient of thermal expansion of the fluid, 0.375 $°C^{-1}$
Fluid viscosity at 20 °C, 0.001 kg/m-s
Longitudinal dispersivity, 4 m
Transverse dispersivity, 1 m

The well flow rate is to be allocated by mobility explicitly in time. The upper and lower boundaries are impermeable and thermally insulated. The boundary condition at the outer radius is specified as hydrostatic pressure with a specified temperature of 20 °C. The boundary condition pressures and temperatures do not vary with time.

Construct a numerical model of this system, and observe the movement of the injected hot water and subsequent withdrawal, for a total simulation time of 180 days. Automatic spacing of 26 nodes in the radial direction and uniform spacing of 11 nodes in the vertical direction are to be used. Use backwards-in-time differencing with a fixed time step of 3 days. The cross-derivative dispersion terms should be calculated explicitly. Print results at 30 day intervals including contour maps of temperature. Use a contour range of 20 to 60 °C, with a contour interval of 5 °C.

The data file that will run example 2 is given in table 8.7. Again, the comments that do not pertain to this problem have been eliminated for brevity.

Table 8.8 contains selections from the output file for this problem. Only key results are presented with highlights of the output described. The initial heat in the region includes both the enthalpy of the fluid and of the porous matrix. The effective ambient permeability at the well bore is defined element by element, thus, 10 values occur. The well-flow rate is calculated explicitly at the beginning of each time step. The temperature field at the end of the injection period shows the less-dense fluid rising up over the cooler, resident fluid. Some numerical overshoot or spatial oscillation is apparent, with temperatures up to 62.5 °C appearing. This gives a zone of plus signs on the contour map of temperature. At the value of time step selected, only two cycles for solution of the pressure and temperature equations were needed, which is the minimum necessary for the explicit calculation of the cross-derivative dispersive-flux terms. During the production part of the cycle, we can see the preferential withdrawal of the warmer water from the upper part of the region, caused by the enhanced mobility of the water at higher temperature. An additional zone, shown as minus signs on the contour maps, shows that the temperature is below the lower limit of 20 °C selected for these plots. Again, this is a spatial-oscillation effect, caused by the coarseness of the grid in conjunction with central differencing for the advective terms. The global-balance summary shows that, at the end of 90 days of withdrawal, about 86 percent of the heat has been recovered. Only about 0.1 percent of the heat left the region through the boundary at the exterior radius. The fluid-withdrawal and heat-withdrawal rates are shown on a per-layer basis at the end of the simulation. The well is producing water at about 38 °C at this time.

Table 8.7.--*Input-data file for example problem 2*

```
C.....HST DATA-INPUT FORM
C...   NOTES:
C...   INPUT LINES ARE DENOTED BY C.N1.N2.N3 WHERE
C...        N1 IS THE READ GROUP NUMBER, N2.N3 IS THE RECORD NUMBER
C...        A LETTER INDICATES AN EXCLUSIVE RECORD SELECTION MUST BE MADE
C...          I.E. A OR B OR C
C...   (O) - OPTIONAL DATA WITH CONDITIONS FOR REQUIREMENT
C...     A RECORD NUMBER IN SQUARE BRACKETS IS THE RECORD WHERE THAT PARAMETER IS FIRST SET
C.....INPUT BY I,J,K RANGE FORMAT IS;
C.0.1.. I1,I2,J1,J2,K1,K2
C.0.2..  VAR1,IMOD1,[VAR2,IMOD2,VAR3,IMOD3]
C...  USE AS MANY OF LINE 0.1 & 0.2 SETS AS NECESSARY
C...  END WITH LINE 0.3
C.0.3.. 0 /  THE SPACE IS REQUIRED
C...    {NNN} - INDICATES THAT THE DEFAULT NUMBER, NNN, IS USED IF A ZERO IS ENTERED FOR THAT VARIABLE
C...   (T/F) - INDICATES A LOGICAL VARIABLE
C...    [I] - INDICATES AN INTEGER VARIABLE
C-----------------------------------------------------------------------------------------------
C.....START OF THE DATA FILE
C.....DIMENSIONING DATA - READ1
C.1.1 .. TITLE LINE 1
 EXAMPLE #2 THERMAL INJECTION AND WITHDRAWAL, CYLINDRICAL SYSTEM,
C.1.2 .. TITLE LINE 2
     VARIABLE DENSITY AND VISCOSITY
C.1.3 .. RESTRT(T/F),TIMRST
F /
C.1.4 .. HEAT,SOLUTE,EEUNIT,CYLIND,SCALMF; ALL (T/F)
T F F T F
C.1.5 .. NX,NY,NZ,NHCN
26 1 11 0
C.1.6 .. NPTCBC,NFBC,NAIFC,NLBC,NHCBC,NWEL
11 0 0 0 0 1
C.1.7 .. NPMZ
1
C.1.8 .. SLMETH[I],LCROSD(T/F)
1 F
C.1.9 .. IBC BY I,J,K RANGE {0.1-0.3} ,WITH NO IMOD PARAMETER, FOR EXCLUDED CELLS
0 /
C.1.10 .. RDECHO(T/F)
F
C---------------------------------------------------------------------------------
C.....STATIC DATA - READ2
C.....OUTPUT INFORMATION
C.2.1 .. PRTRE(T/F)
F
```

Table 8.7.--*Input-data file for example problem 2*--Continued

```
C.....COORDINATE GEOMETRY INFORMATION
C.....CYLINDRICAL COORDINATES
C.2.2B.1A .. R(1),R(NR),ARGRID(T/F);(0) - CYLIND [1.4]
1. 246. T
C.2.2B.1B .. R(I);(0) - NOT ARGRID [2.2B.1A];(0) - CYLIND [1.4]
C.2.2B.2 .. UNIGRZ(T/F);(0) - CYLIND [1.4]
T
C.2.2B.3A .. Z(1),Z(NZ);(0) - UNIGRZ [2.2B.3A],CYLIND [1.4]
0. 30.
C.2.2B.3B .. Z(K);(0) - NOT UNIGRZ [2.2B.3A],CYLIND [1.4]
C.....FLUID PROPERTY INFORMATION
C.2.4.1 .. BP
0.
C.2.4.2 .. P0,T0,W0,DENF0
0. 20. 0. 1000.
C.2.5.1 .. NOTVO,TVFO(I),VISTFO(I),I=1 TO NOTVO;(0) - HEAT [1.4] OR HEAT [1.4] AND SOLUTE [1.4] OR .NOT.HEAT AND .NOT.SOLUTE [1.4]
2 20. .001 60. 4.62E-4
C.....REFERENCE CONDITION INFORMATION
C.2.6.1 .. PAATM
0.
C.2.6.2 .. P0H,T0H
0. 20.
C.....FLUID THERMAL PROPERTY INFORMATION
C.2.7 .. CPF,KTHF,BT;(0) - HEAT [1.4]
4182. .6 3.75E-4
C.....POROUS MEDIA ZONE INFORMATION
C.2.9.1 .. IPMZ,I1Z(IPMZ),I2Z(IPMZ),J1Z(IPMZ),J2Z(IPMZ),K1Z(IPMZ),K2Z(IPMZ)
1 1 26 1 1 1 11
C.....USE AS MANY 2.9.1 LINES AS NECESSARY
C.2.9.2 .. END WITH 0 /
0 /
C.....POROUS MEDIA PROPERTY INFORMATION
C.2.10.1 .. KXX(IPMZ),KYY(IPMZ),KZZ(IPMZ),IPMZ=1 TO NPMZ [1.7]
1.02E-10,,1.02E-10
C.2.10.2 .. POROS(IPMZ),IPMZ=1 TO NPMZ [1.7]
.35
C.2.10.3 .. ABPM(IPMZ),IPMZ=1 TO NPMZ [1.7]
0.
C.....POROUS MEDIA THERMAL PROPERTY INFORMATION
C.2.11.1 .. RCPPM(IPMZ),IPMZ=1 TO NPMZ [1.7];(0) - HEAT [1.4]
2.226E6
C.2.11.2 .. KTXPM(IPMZ),KTYPM(IPMZ),KTZPM(IPMZ),IPMZ=1 TO NPMZ [1.7];(0) - HEAT [1.4]
3*3.5
C.....POROUS MEDIA SOLUTE AND THERMAL DISPERSION INFORMATION
C.2.12 .. ALPHL(IPMZ),ALPHT(IPMZ),IPMZ=1 TO NPMZ [1.7];(0) - SOLUTE [1.4] OR HEAT [1.4]
```

Table 8.7.--*Input-data file for example problem 2*--Continued

```
4. 1.
C.....SOURCE-SINK WELL INFORMATION
C.2.14.1 .. RDWDEF(T/F);(0) - NWEL [1.6] > 0
T
C.2.14.3. .. IWEL,IW,JW,LCBOTW,LCTOPW,WBOD,WQMETH[I];(0) - RDWDEF [2.14.1],
1 1 1 1 11 2. 11
C.2.14.4 .. WCF(L);L = 1 TO NZ (EXCLUSIVE) BY ELEMENT
10*1.
C.....USE AS MANY 2.14.3-6 LINES AS NECESSARY
C.2.14.7 .. END WITH 0 /
0 /
C.....BOUNDARY CONDITION INFORMATION
C.....    SPECIFIED VALUE B.C.
C.2.15 .. IBC BY I,J,K RANGE {0.1-0.3} WITH NO IMOD PARAMETER,;(0) - NPTCBC [1.6] > 0
26 26 1 1 1 11
100
0 /
C.....FREE SURFACE B.C.
C.2.20 .. FRESUR(T/F),PRTCCM(T/F)
F /
C.....INITIAL CONDITION INFORMATION
C.2.21.1 .. ICHYDP,ICT,ICC; ALL (T/F);IF NOT.HEAT, ICT = F, IF NOT.SOLUTE, ICC = F
T T F
C.2.21.3A .. ZPINIT,PINIT;(0) - ICHYDP [2.21.1] AND NOT ICHWT [2.21.2]
30. 0.
C.2.21.4B .. T BY I,J,K RANGE {0.1-0.3};(0) - HEAT [1.4] AND ICT [2.21.1]
1 26 1 1 1 11
20. 1
0 /
C.....CALCULATION INFORMATION
C.2.22.1 .. FDSMTH,FDTMTH
.5 1.
C.2.22.2 .. TOLDEN{.001},MAXITN{5}
0. 10
C.....OUTPUT INFORMATION
C.2.23.1 .. PRTPMP,PRTFP,PRTIC,PRTBC,PRTSLM,PRTWEL; ALL (T/F)
6*T
C.2.23.2 .. IPRPTC,PRTDV(T/F);(0) - PRTIC [2.23.1]
110 F
C.2.23.4 .. PLTZON(T/F);(0) - PRTPMP [2.23.1]
F
C.2.23.5 .. OCPLOT(T/F)
F
C-----------------------------------------------------------------------------------------
C..... TRANSIENT DATA - READ3
```

Table 8.7.--*Input-data file for example problem 2*--Continued

```
C.3.1 .. THRU(T/F)
F
C.....IF THRU IS TRUE PROCEED TO RECORD 3.99
C.....THE FOLLOWING IS FOR NOT THRU
C.....SOURCE-SINK WELL INFORMATION
C.3.2.1 .. RDWFLO(T/F),RDWHD(T/F);(O) - NWEL [1.6] > 0
T F
C.3.2.2 .. IWEL,QWV,PWSUR,PWKT,TWSRKT,CWKT;(O) - RDWFLO [3.2.1] OR RDWHD [3.2.1]
1 203. 0. 0. 60. 0.
C.....USE AS MANY 3.2.2 LINES AS NECESSARY
C.3.2.3 .. END WITH 0 /
0 /
C.....BOUNDARY CONDITION INFORMATION
C.....    SPECIFIED VALUE B.C.
C.3.3.1 .. RDSPBC,RDSTBC,RDSCBC,ALL(T/F);(O) - NOT CYLIND [1.4] AND NPTCBC [1.6] > 0
T F F
C.3.3.2 .. PNP B.C. BY I,J,K RANGE {0.1-0.3};(O) - RDSPBC [3.3.1]
26 26 1 1 11 11
0. 1
26 26 1 1 10 10
2.9421E4 1
26 26 1 1 9 9
5.8842E4 1
26 26 1 1 8 8
8.8263E4 1
26 26 1 1 7 7
1.17684E5 1
26 26 1 1 6 6
1.47105E5 1
26 26 1 1 5 5
1.76526E5 1
26 26 1 1 4 4
2.05947E5 1
26 26 1 1 3 3
2.35368E5 1
26 26 1 1 2 2
2.64789E5 1
26 26 1 1 1 1
2.9421E5 1
0 /
C.3.3.3 .. TSBC BY I,J,K RANGE {0.1-0.3}; (O) - RDSPBC [3.3.1] AND HEAT [1.4]
26 26 1 1 1 11
20. 1
0 /
C.....CALCULATION INFORMATION
```

Table 8.7.--*Input-data file for example problem 2*--Continued

```
C.3.7.1 .. RDCALC(T/F)
T
C.3.7.2 .. AUTOTS(T/F);(0) - RDCALC [3.7.1]
F
C.3.7.3.A .. DELTIM;(0) - RDCALC [3.7.1] AND NOT AUTOTS [3.7.2]
2.592E5
C.3.7.4 .. TIMCHG
7.7760E6
C.....OUTPUT INFORMATION
C.3.8.1 .. PRIVEL,PRIDV,PRISLM,PRIKD,PRIPTC,PRIGFB,PRIWEL,PRIBCF; ALL [I]
0 0 10 0 10 -1 0 30
C.3.8.2 .. IPRPTC;(0) - IF PRIPTC [3.8.1] NOT = 0
010
C.3.8.3 .. CHKPTD(T/F),NTSCHK,SAVLDO(T/F)
F /
C.....CONTOUR MAP INFORMATION
C.3.9.1 .. RDMPDT,PRTMPD; ALL (T/F)
T T
C.3.9.2 .. MAPPTC,PRIMAP[I];(0) - RDMPDT [3.9.1]
010 10
C.3.9.3 .. YPOSUP(T/F),ZPOSUP(T/F),LENAX,LENAY,LENAZ;(0) - RDMPDT [3.9.1]
F T 12. 0. 6.
C.3.9.4 .. IMAP1{1},IMAP2{NX},JMAP1{1},JMAP2{NY},KMAP1{1},KMAP2{NZ},AMIN,AMAX,NMPZON{5}:(0) - RDMPDT [3.9.1]
0 0 0 0 0 0 20. 60. 8
C.....ONE OF THE 3.9.4 LINES REQUIRED FOR EACH DEPENDENT VARIABLE
C.....     TO BE MAPPED
C.....END OF FIRST SET OF TRANSIENT INFORMATION
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.....READ SETS OF READ3 DATA AT EACH TIMCHG UNTIL THRU  (LINES 3.N1.N2)
C..... TRANSIENT DATA - READ3
C.3.1 .. THRU(T/F)
F
C.....IF THRU IS TRUE PROCEED TO RECORD 3.99
C.....THE FOLLOWING IS FOR NOT THRU
C.....SOURCE-SINK WELL INFORMATION
C.3.2.1 .. RDWFLO(T/F),RDWHD(T/F);(0) - NWEL [1.6] > 0
T F
C.3.2.2 .. IWEL,QWV,PWSUR,PWKT,TWSRKT,CWKT;(0) - RDWFLO [3.2.1] OR RDWHD [3.2.1]
1 -203. 0. 0. 60. 0.
C.....USE AS MANY 3.2.2 LINES AS NECESSARY
C.3.2.3 .. END WITH 0 /
0 /
C.....BOUNDARY CONDITION INFORMATION
C.....     SPECIFIED VALUE B.C.
C.3.3.1 .. RDSPBC,RDSTBC,RDSCBC,ALL(T/F);(0) - NOT CYLIND [1.4] AND NPTCBC [1.6] > 0
```

Table 8.7.--*Input-data file for example problem 2*--Continued

```
F F F
C.....CALCULATION INFORMATION
C.3.7.1 .. RDCALC(T/F)
T
C.3.7.2 .. AUTOTS(T/F);(0) - RDCALC [3.7.1]
F
C.3.7.3.A .. DELTIM;(0) - RDCALC [3.7.1] AND NOT AUTOTS [3.7.2]
2.592E5
C.3.7.4 .. TIMCHG
1.5552E7
C.....OUTPUT INFORMATION
C.3.8.1 .. PRIVEL,PRIDV,PRISLM,PRIKD,PRIPTC,PRIGFB,PRIWEL,PRIBCF; ALL [I]
0 0 10 0 10 -1 -1 30
C.3.8.2 .. IPRPTC;(0) - IF PRIPTC [3.8.1] NOT = 0
010
C.3.8.3 .. CHKPTD(T/F),NTSCHK,SAVLDO(T/F)
F /
C.....CONTOUR MAP INFORMATION
C.3.9.1 .. RDMPDT,PRTMPD; ALL (T/F)
T T
C.3.9.2 .. MAPPTC,PRIMAP[I];(0) - RDMPDT [3.9.1]
010 10
C.3.9.3 .. YPOSUP(T/F),ZPOSUP(T/F),LENAX,LENAY,LENAZ;(0) - RDMPDT [3.9.1]
F T 12. 0. 6.
C.3.9.4 .. IMAP1{1},IMAP2{NX},JMAP1{1},JMAP2{NY},KMAP1{1},KMAP2{NZ},AMIN,AMAX,NMPZON{5}:(0) - RDMPDT [3.9.1]
0 0 0 0 0 0 20. 60. 8
C.....ONE OF THE 3.9.4 LINES REQUIRED FOR EACH DEPENDENT VARIABLE
C.....     TO BE MAPPED
C.....END OF SECOND SET OF TRANSIENT INFORMATION
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.....READ SETS OF READ3 DATA AT EACH TIMCHG UNTIL THRU  (LINES 3.N1.N2)
C.....END OF CALCULATION LINES FOLLOW, THRU=.TRUE.
C.3.99.1 .. THRU
T
C.....TEMPORAL PLOT INFORMATION
C.3.99.2 .. PLOTWP,PLOTWT,PLOTWC; ALL (T/F)
3*F
C.....END OF DATA FILE
```

Table 8.8.--*Selections from the output file for example problem 2*

```
**************************************************
*                                                *
*    THREE DIMENSIONAL FLOW, HEAT AND SOLUTE     *
*  TRANSPORT SIMULATOR - (HST3D):RELEASE - 1.0   *
*                                                *
**************************************************


EXAMPLE #2 THERMAL INJECTION AND WITHDRAWAL, CYLINDRICAL SYSTEM,
    VARIABLE DENSITY AND VISCOSITY

          *** FUNDAMENTAL INFORMATION ***
   CYLINDRICAL COORDINATES
   HEAT TRANSPORT SIMULATION
   NO SOLUTE TRANSPORT SIMULATON
   WELLBORE MODEL MAY INCLUDE : FLUID PRESSURE DROP
             HEAT BALANCE UP THE RISER PIPE
   INPUT DATA IS EXPECTED IN METRIC UNITS

          *** PROBLEM DIMENSIONING INFORMATION ***
   NUMBER OF NODES IN X-DIRECTION ........................................... NX ...   26
   NUMBER OF NODES IN Y-DIRECTION ........................................... NY ...    1
   NUMBER OF NODES IN Z-DIRECTION ........................................... NZ ...   11
   NUMBER OF POROUS MEDIA ZONES ............................................. NPMZ .    1
   NUMBER OF SPECIFIED PRESSURE, TEMPERATURE OR MASS FRACTION B.C. ........... NPTCBC   11
   NUMBER OF SPECIFIED FLUX B.C. CELLS (FLOW, HEAT OR SOLUTE)................. NFBC .    0
   NUMBER OF HEAT CONDUCTION B.C. CELLS ..................................... NHCBC     0
   NUMBER OF NODES OUTSIDE REGION FOR EACH HEAT CONDUCTION B.C. CELL ......... NHCN      0
   NUMBER OF AQUIFER INFLUENCE FUNCTION CELLS ............................... NAIFC     0
   NUMBER OF LEAKAGE CELLS .................................................. NLBC .    0
   NUMBER OF WELLS .......................................................... NWEL .    1

   DIRECT D4 SOLVER IS SELECTED
   FULL CROSS-DISPERSIVITY COEFFICIENT STORAGE ALLOCATED
   THE A4 ARRAY IN D4DES IS DIMENSIONED ...................................   2917  ELEMENTS
   THE TOTAL STORAGE REQUIRED BY THE DIRECT METHOD IS .......................   4347  ELEMENTS
   THE TOTAL STORAGE REQUIRED BY THE ITERATIVE METHOD IS ....................   2002  ELEMENTS
   TOTAL LENGTH OF LABELED COMMON BLOCKS ..................................   6681  BYTES
                                                              REQUIRED        COMPILED
   LENGTH OF VARIABLE LENGTH REAL ARRAY  (VPA ARRAY) ..........  13876  ELEMENTS  250000  ELEMENTS
   LENGTH OF VARIABLE LENGTH INTEGER ARRAY (IVPA ARRAY) .......   2099  ELEMENTS   20000  ELEMENTS
-----------------------------------------------------------------------------------------------
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
               *** TIME INVARIANT OR STATIC DATA ***
               *** CYLINDRICAL (R-Z) COORDINATE DATA ***
          AQUIFER INTERIOR RADIUS .............. RINT .      1.000  ( M)
          AQUIFER EXTERIOR RADIUS .............. REXT .     246.0   ( M)

               R-DIRECTION NODE COORDINATES   ( M)

  1         2         3         4         5         6         7         8         9         10
    1.00      1.25      1.55      1.94      2.41      3.01      3.75      4.67      5.82      7.26

 11        12        13        14        15        16        17        18        19        20
    9.04     11.27     14.05     17.51     21.82     27.20     33.90     42.25     52.66     65.63

 21        22        23        24        25        26
   81.80    101.95    127.06    158.37    197.38    246.00


               R-COORDINATE CELL BOUNDARY LOCATIONS (BETWEEN NODE(I) AND NODE(I+1))   ( M)

  1         2         3         4         5         6         7         8         9         10
    1.12      1.39      1.74      2.17      2.70      3.36      4.19      5.23      6.51      8.12

 11        12        13        14        15        16        17        18        19        20
   10.12     12.61     15.72     19.59     24.41     30.43     37.92     47.26     58.91     73.42

 21        22        23        24        25
   91.50    114.05    142.14    177.16    220.80


               Z-DIRECTION NODE COORDINATES   ( M)
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
   1         2         3         4         5         6         7         8         9        10
    0.00      3.00      6.00      9.00     12.00     15.00     18.00     21.00     24.00     27.00

  11
   30.00


                    Z-AXIS IS POSITIVE VERTICALLY UPWARD

                              ** AQUIFER PROPERTIES  **  (READ ECHO)
         REGION                POROUS MEDIUM      M.C.=MODIFICATION CODE
    I1  I2  J1  J2  K1  K2     ZONE INDEX
 ...................................................................................
     1  26   1   1   1  11          1

                    *** POROUS MEDIA PROPERTIES ***


                    X-DIRECTION PERMEABILITIES   ( M**2)

       1
 1.0200E-10


                    Z-DIRECTION PERMEABILITIES   ( M**2)

       1
 1.0200E-10


                    POROSITY (-)

       1
 0.3500
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

*** INTERMEDIATE COMPUTED DATA ***

R-DIRECTION CONDUCTANCE FACTOR BETWEEN R(I) AND R(I+1) (M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09
10	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
9	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
8	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
7	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
6	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
5	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
4	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
3	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
2	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
1	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09

	11	12	13	14	15	16	17	18	19	20
11	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09
10	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
9	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
8	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
7	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
6	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
5	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
4	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
3	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
2	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
1	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09

	21	22	23	24	25
11	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09
10	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
9	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
8	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09

Table 8.8.--*Selections from the output file for example problem 2*--Continued

7	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
6	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
5	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
4	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
3	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
2	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09	8.7309E-09
1	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09	4.3654E-09

Z-DIRECTION CONDUCTANCE FACTOR BETWEEN Z(K) AND Z(K+1) (M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
10	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
9	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
8	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
7	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
6	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
5	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
4	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
3	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
2	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09
1	2.6852E-11	7.3967E-11	1.1490E-10	1.7848E-10	2.7724E-10	4.3066E-10	6.6898E-10	1.0392E-09	1.6142E-09	2.5075E-09

	11	12	13	14	15	16	17	18	19	20
10	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
9	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
8	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
7	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
6	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
5	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
4	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
3	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
2	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07
1	3.8950E-09	6.0504E-09	9.3985E-09	1.4599E-08	2.2678E-08	3.5228E-08	5.4721E-08	8.5003E-08	1.3204E-07	2.0511E-07

21	22	23	24	25	26

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
10  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 9  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 8  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 7  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 6  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 5  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 4  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 3  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 2  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06
 1  3.1861E-07  4.9492E-07  7.6879E-07  1.1942E-06  1.8551E-06  1.2566E-06

                          *** PROPERTIES BY POROUS MEDIUM ZONE ***

                      POROUS MEDIUM VERTICAL COMPRESSIBILITY  (1/ PA)

      1
  0.0000


                      DENSITY-HEAT CAPACITY PRODUCT   (  J/ M**3-DEG.C)

      1
  2.2260E+06


                      THERMAL CONDUCTIVITY IN X-DIRECTION   (    W/ M-DEG.C)

      1
   3.500

      1
   3.500


                      THERMAL CONDUCTIVITY IN Z-DIRECTION   (    W/ M-DEG.C)

      1
   3.500
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
                LONGITUDINAL DISPERSIVITY   ( M)

     1
 4.000


                TRANSVERSE DISPERSIVITY   ( M)

     1
 1.000


                ATMOSPHERIC PRESSURE (ABSOLUTE) ............................ PAATM   101325.0  ( PA)
                REFERENCE PRESSURE FOR ENTHALPY ............................ POH ..       0.0  ( PA)

                REFERENCE TEMPERATURE FOR ENTHALPY ......................... TOH ..      20.0  (DEG.C)


                          *** FLUID PROPERTIES ***
                                    PHYSICAL
                FLUID COMPRESSIBILITY ...................................... BP ...   0.00E-01  (1/ PA)

                REFERENCE PRESSURE FOR DENSITY ............................. PO ..        0.0  ( PA)
                REFERENCE TEMPERATURE FOR DENSITY .......................... TO ..       20.0  (DEG.C)
                FLUID DENSITY AT SOLUTE     MASS    FRACTION OF ZERO ....... DENFO   1000.      (KG/ M**3)

                                    THERMAL
                FLUID COEFFICIENT OF THERMAL EXPANSION ..................... BT ...   3.75E-04  (1/DEG.C)
                FLUID HEAT CAPACITY ........................................ CPF ..  4.182E+03  (  J/KG-DEG.C)
                FLUID THERMAL CONDUCTIVITY ................................. KTHF .  0.600      (    W/ M-DEG.C)
                FLUID SPECIFIC ENTHALPY AT REFERENCE CONDITIONS............. EHO ..  8.333E+04  (  J/KG)

                                    VISCOSITY-TEMPERATURE DATA TABLE
                                            TEMPERATURE (DEG.C)     VISCOSITY (KG/ M-S)

                FLUID AT SOLUTE MINIMUM MASS FRACTION
                                              20.0                   1.000E-03
                                              60.0                   4.620E-04

                          *** INITIAL CONDITIONS ***
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

INITIAL AQUIFER FLUID PRESSURE FOR HYDROSTATIC I.C. .... PINIT. 0.0 (PA)
ELEVATION OF INITIAL PRESSURE .... ZPINIT 30.0 (M)

INITIAL PRESSURE DISTRIBUTION (PA)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04
9	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04
8	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04
7	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05
6	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05
5	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05
4	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05
3	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05
2	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05
1	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05

	11	12	13	14	15	16	17	18	19	20
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04
9	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04
8	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04
7	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05
6	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05
5	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05
4	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05
3	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05
2	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05
1	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05

.

	21	22	23	24	25	26
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04	2.9421E+04
9	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04	5.8842E+04
8	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04	8.8263E+04

Table 8.8.--*Selections from the output file for example problem 2*--Continued

7	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05	1.1768E+05
6	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05	1.4711E+05
5	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05	1.7653E+05
4	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05	2.0595E+05
3	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05	2.3537E+05
2	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05	2.6479E+05
1	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05	2.9421E+05

INITIAL TEMPERATURES (DEG.C)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
10	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
9	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
8	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
7	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
6	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
5	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
4	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
3	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
2	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0

	11	12	13	14	15	16	17	18	19	20
11	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
10	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
9	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
8	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
7	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
6	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
5	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
4	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
3	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
2	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0

Table 8.8.--*Selections from the output file for example problem 2*--Continued

	21	22	23	24	25	26
11	20.0	20.0	20.0	20.0	20.0	20.0
10	20.0	20.0	20.0	20.0	20.0	20.0
9	20.0	20.0	20.0	20.0	20.0	20.0
8	20.0	20.0	20.0	20.0	20.0	20.0
7	20.0	20.0	20.0	20.0	20.0	20.0
6	20.0	20.0	20.0	20.0	20.0	20.0
5	20.0	20.0	20.0	20.0	20.0	20.0
4	20.0	20.0	20.0	20.0	20.0	20.0
3	20.0	20.0	20.0	20.0	20.0	20.0
2	20.0	20.0	20.0	20.0	20.0	20.0
1	20.0	20.0	20.0	20.0	20.0	20.0

INITIAL PORE VOLUME PER CELL RING (M**3)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	0.4146	1.142	1.774	2.756	4.281	6.650	10.33	16.05	24.93	38.72
10	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
9	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
8	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
7	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
6	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
5	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
4	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
3	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
2	0.8292	2.284	3.548	5.512	8.562	13.30	20.66	32.09	49.85	77.44
1	0.4146	1.142	1.774	2.756	4.281	6.650	10.33	16.05	24.93	38.72

	11	12	13	14	15	16	17	18	19	20
11	60.14	93.43	145.1	225.4	350.2	544.0	845.0	1313.	2039.	3167.
10	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
9	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
8	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
7	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
6	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
5	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.

Table 8.8.--*Selections from the output file for example problem 2*--Continued

4	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
3	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
2	120.3	186.9	290.2	450.9	700.4	1088.	1690.	2625.	4078.	6334.
1	60.14	93.43	145.1	225.4	350.2	544.0	845.0	1313.	2039.	3167.

	21	22	23	24	25	26
11	4920.	7642.	1.1871E+04	1.8440E+04	2.8644E+04	1.9404E+04
10	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
9	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
8	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
7	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
6	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
5	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
4	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
3	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
2	9839.	1.5284E+04	2.3742E+04	3.6880E+04	5.7288E+04	3.8808E+04
1	4920.	7642.	1.1871E+04	1.8440E+04	2.8644E+04	1.9404E+04

```
INITIAL FLUID IN REGION ..................................  1.996192E+09  (KG) ;  1.996192E+06  ( M**3)

INITIAL HEAT IN REGION (FLUID & POROUS MEDIUM).............  3.313955E+14  (  J)

                         *** WELL DATA ***
          .....................................................................
          WELL          PERFORATIONS    CALCULATION        WELL DIAMETER
          NO.    I    J   K1   K2          TYPE               ( M)
          .....................................................................
           1     1    1    1   11           11              2.000E+00
WELL NO.   1    SPECIFIED FLOW RATE                         ALLOCATION BY MOBILITY

          ELEMENT LEVEL    EFFECTIVE AMBIENT     LAYER    WELL FLOW FACTOR
              NO.          PERMEABILITY           NO.
                             ( M**2)                          ( M**3)
          .....................................................................

          WELL NO.   1

                                                  11          1.530E-10
           10              1.020E-10              10          3.060E-10
            9              1.020E-10               9          3.060E-10
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

8	1.020E-10	8	3.060E-10
7	1.020E-10	7	3.060E-10
6	1.020E-10	6	3.060E-10
5	1.020E-10	5	3.060E-10
4	1.020E-10	4	3.060E-10
3	1.020E-10	3	3.060E-10
2	1.020E-10	2	3.060E-10
1	1.020E-10	1	1.530E-10

EXPLICIT WELL FLOW RATE AT EACH LAYER

INDEX NUMBERS FOR SPECIFIED P,T OR C NODES

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

	11	12	13	14	15	16	17	18	19	20
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
        21          22          23          24          25          26
 11                                                                      11.
 10                                                                      10.
  9                                                                       9.
  8                                                                       8.
  7                                                                       7.
  6                                                                       6.
  5                                                                       5.
  4                                                                       4.
  3                                                                       3.
  2                                                                       2.
  1                                                                       1.


                          *** CALCULATION INFORMATION ***

                 TOLERANCE FOR P,T,C ITERATION (FRACTIONAL DENSITY CHANGE) ... TOLDEN  0.0010
                 MAXIMUM NUMBER OF ITERATIONS ALLOWED ON P,T,C EQUATIONS ..... MAXITN   10
                 BACKWARDS-IN-TIME (IMPLICIT)  DIFFERENCING FOR TEMPORAL DERIVATIVE
                 CENTERED-IN-SPACE DIFFERENCING FOR CONVECTIVE TERMS
                 THE CROSS-DERIVATIVE HEAT AND SOLUTE FLUX TERMS WILL BE CALCULATED EXPLICITLY

----------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------


                          *** TRANSIENT DATA ***


                 SPECIFIED BOUNDARY PRESSURES ( PA)

            VERTICAL SLICES

                                        J =  1

         1           2           3           4           5           6           7           8           9          10
 11
 10
  9
  8
  7
  6
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

5										
4										
3										
2										
1										

	11	12	13	14	15	16	17	18	19	20
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

	21	22	23	24	25	26
11						0.0000
10						2.9421E+04
9						5.8842E+04
8						8.8263E+04
7						1.1768E+05
6						1.4710E+05
5						1.7653E+05
4						2.0595E+05
3						2.3537E+05
2						2.6479E+05
1						2.9421E+05

ASSOCIATED BOUNDARY TEMPERATURES FOR INFLOW (DEG.C)

VERTICAL SLICES

J = 1

1	2	3	4	5	6	7	8	9	10

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
 11
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1

          11          12          13          14          15          16          17          18          19          20
 11
 10
  9
  8
  7
  6
  5
  4
  3
  2
  1

          21          22          23          24          25          26
 11                                                                      20.0
 10                                                                      20.0
  9                                                                      20.0
  8                                                                      20.0
  7                                                                      20.0
  6                                                                      20.0
  5                                                                      20.0
  4                                                                      20.0
  3                                                                      20.0
  2                                                                      20.0
  1                                                                      20.0
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
                    *** TRANSIENT WELL DATA ***
WELL      FLOW RATE     SURFACE      WELL DATUM     PRESSURE     INJECTION          INJECTION
NO.       ( M**3/S)    PRESSURE     PRESSURE       LIMITED?    TEMPERATURE        MASS    FRACTION
                         ( PA)           ( PA)                     (DEG.C)            (-)
.............................................................................................
 1       0.2030                                        NO          60.00


                    *** CALCULATION INFORMATION ***

FIXED TIME STEP LENGTH ...................................... DELTIM  2.592E+05  (S);   3.00      (D)
TIME AT WHICH NEXT SET OF TIME VARYING
     PARAMETERS WILL BE READ ................................ TIMCHG  7.776E+06  (S);   90.0      (D)
                              *** MAPPING DATA ***
--------------------------------------------------------------------------------------------
          LENGTH OF X-AXIS ...............12.0  (IN.)
          LENGTH OF Y-AXIS ............... 0.0  (IN.)
          LENGTH OF Z-AXIS ............... 6.0  (IN.)
          Z-AXIS IS POSITIVE UPWARD
--------------------------------------------------------------------------------------------
     MAP NO.   I1   I2   J1   J2   K1   K2        MINIMUM VALUE     MAXIMUM VALUE   NUMBER OF
                                                   OF VARIABLE       OF VARIABLE     ZONES
--------------------------------------------------------------------------------------------
          2     1   26    1    1    1   11             20.               60.            8

------------------------------------------------------------------------------------------------------------


                    *** OUTPUT AT END OF TIME STEP NO.    30 ***

          TIME ................................................  7.776E+06 (S);      90.0     (D)

     CURRENT TIME STEP LENGTH .................................  2.592E+05 (S);      3.00     (D)

     NO. OF P,T,C LOOP ITERATIONS USED ........................  2
     MAXIMUM CHANGE IN PRESSURE ............................... -1.4460E+02 ( PA)   AT LOCATION (22, 1, 1)
     MAXIMUM CHANGE IN TEMPERATURE ............................  1.8286E+00 (DEG.C) AT LOCATION (24, 1,11)

          TEMPERATURE   (DEG.C)

VERTICAL SLICES

                              J =  1
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

	1	2	3	4	5	6	7	8	9	10
11	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
10	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
9	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
8	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
7	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
6	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
5	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
4	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
3	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
2	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00
1	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00

	11	12	13	14	15	16	17	18	19	20
11	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.02	59.98
10	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.02	59.96
9	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.03	59.91
8	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.04	59.84
7	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.05	59.75
6	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.05	59.68
5	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.05	59.63
4	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.05	59.62
3	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.04	59.65
2	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.03	59.69
1	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.02	59.71

	21	22	23	24	25	26
11	59.55	62.16	62.44	44.19	25.49	21.09
10	59.61	62.21	61.63	43.03	25.17	21.02
9	59.80	62.28	59.93	41.20	24.66	20.91
8	60.12	62.19	57.52	38.90	24.05	20.79
7	60.55	61.78	54.52	36.34	23.41	20.66
6	61.01	60.94	51.07	33.70	22.80	20.54
5	61.42	59.64	47.30	31.16	22.25	20.44
4	61.69	57.90	43.42	28.88	21.79	20.35
3	61.77	55.79	39.70	27.01	21.43	20.29
2	61.67	53.53	36.57	25.69	21.19	20.24
1	61.49	51.54	34.69	25.11	21.10	20.23

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
                    *** GLOBAL FLOW BALANCE SUMMARY ***
                         CURRENT TIME STEP                       RATES                 AMOUNTS

FLUID INFLOW ............................................... 1.999550E+02  (KG/S)    5.182833E+07  (KG)
FLUID OUTFLOW .............................................. 2.014491E+02  (KG/S)    5.221562E+07  (KG)
CHANGE IN FLUID IN REGION ..................................-1.493657E+00  (KG/S)   -3.871558E+05  (KG)
RESIDUAL IMBALANCE ......................................... 4.905415E-04  (KG/S)    1.271484E+02  (KG)
FRACTIONAL IMBALANCE .......................................                             0.0000

HEAT INFLOW ................................................ 5.011149E+07  (  J/S)    1.298890E+13  (  J)
HEAT OUTFLOW ............................................... 1.732711E+07  (  J/S)    4.491186E+12  (  J)
CHANGE IN HEAT IN REGION ................................... 3.278493E+07  (  J/S)    8.497854E+12  (  J)
RESIDUAL IMBALANCE ......................................... 5.465375E+02  (  J/S)    1.416625E+08  (  J)
FRACTIONAL IMBALANCE .......................................                             0.0000

                         CUMULATIVE SUMMARY
                                                               AMOUNTS

FLUID INFLOW ............................................... 1.554850E+09  (KG)
FLUID OUTFLOW .............................................. 1.566620E+09  (KG)
CHANGE IN FLUID IN REGION ..................................-1.176905E+07  (KG)
FLUID IN REGION ............................................ 1.984422E+09  (KG)
RESIDUAL IMBALANCE ......................................... 8.228376E+02  (KG)
FRACTIONAL IMBALANCE .......................................     0.0000

HEAT  INFLOW ............................................... 3.896669E+14  (  J)
HEAT  OUTFLOW .............................................. 1.312503E+14  (  J)
CHANGE IN HEAT IN REGION ................................... 2.584199E+14  (  J)
HEAT IN REGION ............................................. 5.898154E+14  (  J)
RESIDUAL IMBALANCE ......................................... 3.271183E+09  (  J)
FRACTIONAL IMBALANCE .......................................     0.0000

CUMULATIVE SPECIFIED P CELL FLUID NET INFLOW ...............-1.566620E+09  (KG)
CUMULATIVE FLUX B.C. FLUID NET INFLOW ...................... 0.000000E-01  (KG)
CUMULATIVE LEAKAGE B.C. FLUID NET INFLOW ................... 0.000000E-01  (KG)
CUMULATIVE AQUIFER INFLUENCE FLUID NET INFLOW .............. 0.000000E-01  (KG)


CUMULATIVE SPECIFIED T CELL OR ASSOCIATED WITH
     SPECIFIED P CELL HEAT NET INFLOW.......................-1.312503E+14  (  J)
CUMULATIVE FLUX B.C. HEAT NET INFLOW ....................... 0.000000E-01  (  J)
CUMULATIVE LEAKAGE B.C. HEAT NET INFLOW .................... 0.000000E-01  (  J)
CUMULATIVE AQUIFER INFLUENCE HEAT NET INFLOW ............... 0.000000E-01  (  J)
CUMULATIVE HEAT CONDUCTION B.C. HEAT NET INFLOW ............ 0.000000E-01  (  J)
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

SPECIFIED PRESSURE, TEMPERATURE, OR MASS FRACTION B.C. FLOW RATES
POSITIVE IS INTO THE REGION

FLUID (KG/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

	11	12	13	14	15	16	17	18	19	20
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

	21	22	23	24	25	26
11						-23.71
10						-21.04
9						-20.08
8						-19.41

Table 8.8.--*Selections from the output file for example problem 2*--Continued

7	-18.92
6	-18.58
5	-18.36
4	-18.26
3	-18.26
2	-18.36
1	-6.482

ASSOCIATED HEAT (J/S)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

	11	12	13	14	15	16	17	18	19	20
11										
10										
9										
8										
7										
6										
5										
4										
3										
2										
1										

Table 8.8.--*Selections from the output file for example problem 2*--Continued

	21	22	23	24	25	26
11						-2.0832E+06
10						-1.8430E+06
9						-1.7498E+06
8						-1.6810E+06
7						-1.6288E+06
6						-1.5904E+06
5						-1.5640E+06
4						-1.5484E+06
3						-1.5434E+06
2						-1.5489E+06
1						-5.4633E+05

VERTICAL SLICES

HEAT (J/S)

VERTICAL SLICES

VERTICAL PLANE AT ROW NO. 1

```
VERTICAL CROSS-SECTION OF SIMULATION REGION -     TEMPERATURES                    PAGE 1

****4**4*4*4*444*444*4444*44444*4444444*4++++++++*+++++++++++*++4444    33333*    22222     1111*11     *
44444444444444444444444444444444444444444++++++++++++++++++++++44444    33333     22222     1111111
44444444444444444444444444444444444444444++++++++++++++++++++++44444    3333     22222      111111
44444444444444444444444444444444444444444++++++++++++++++++++++4444     3333     22222      111111
*4444444444444444444444444444444444444444++++++++++++++++++++++4444     3333     22222     1111111      *
44444444444444444444444444444444444444444+++++++++++++++++++++44444    3333    222222      111111
44444444444444444444444444444444444444444+++++++++++++++++++++4444     3333     22222      11111
44444444444444444444444444444444444444444+++++++++++++++++++++4444    33333    222222      11111
*444444444444444444444444444444444444444+++++++++++++++++++++44444    3333     222222     111111        *
4444444444444444444444444444444444444444++++++++++++++++++4444444    33333    222222     111111
4444444444444444444444444444444444444444++++++++++++++++44444444    3333    222222     1111111
*44444444444444444444444+44444444444++++++++++++++++++444444444    3333     22222     111111          *
444444444444444444444444+44444444444++++++++++++++++++444444444   33333    222222     1111111
444444444444444444444444+4444444444++++++++++++++++++444444444   3333    222222      1111111
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
4444444444444444444444444+444444444++++++++++++++++++444444444    3333    22222       1111111
*444444444444444444444444+44444444++++++++++++++++++444444444    33333   22222       1111111         *
4444444444444444444444444+44444444++++++++++++++++++4444444     3333   22222       11111111
444444444444444444444444++44444444+++++++++++++++++4444444     3333    2222       11111111
*44444444444444444444444++44444444+++++++++++++++++444444    33333   2222       111111111          *
444444444444444444444444++44444444++++++++++++++++444444     3333   22222      111111111
444444444444444444444444++4444444+++++++++++++++++44444     33333  22222      1111111111
444444444444444444444444++4444444++++++++++++++++444444    33333   2222      1111111111
*44444444444444444444444++4444444++++++++++++++4444444    33333   22222     11111111111            *
444444444444444444444444++4444444+++++++++++++4444444    33333   22222     11111111111
4444444444444444444444444+4444444++++++++++++44444444   33333   22222     11111111111
4444444444444444444444444+4444444+++++++++++44444444   3333   222222    111111111111
*444444444444444444444444+4444444+++++++++++44444444   3333   222222    111111111111               *
4444444444444444444444444+4444444++++++++++44444444   3333   222222    11111111111
4444444444444444444444444+4444444++++++++++44444444   333   222222    11111111111
*44444444444444444444444444444444+++++++++44444444   3333  222222    1111111111                    *
444444444444444444444444444444444+++++++++44444444  3333   222222   11111111111
444444444444444444444444444444444+++++++++4444444  3333  222222   1111111111
444444444444444444444444444444444+++++++++444444  3333  22222   1111111111
*44444444444444444444444444444444+++++++++444444  3333  22222   111111111                          *
444444444444444444444444444444444++++++++444444  333  2222   111111111
444444444444444444444444444444444++++++++444444 3333  2222   111111111
****4**4*4*4*444*444*4444*44444*4++++++*+44444  * 333  222*    11111111*                *          *
```

MAP LEGEND
HORIZONTAL GRID NODE RANGE, FROM   1 TO 26
VERTICAL GRID NODE RANGE, FROM   1 TO 11
DEPENDENT VARIABLE RANGE   MAP CHARACTER

2.000E+01 - 2.500E+01	
2.500E+01 - 3.000E+01	1
3.000E+01 - 3.500E+01	
3.500E+01 - 4.000E+01	2
4.000E+01 - 4.500E+01	
4.500E+01 - 5.000E+01	3
5.000E+01 - 5.500E+01	
5.500E+01 - 6.000E+01	4

GRID NODE LOCATIONS ACROSS THE PAGE

1.00	1.25	1.55	1.94	2.41	3.01	3.75	4.67	5.82	7.26	9.04	11.3
14.0	17.5	21.8	27.2	33.9	42.3	52.7	65.6	81.8	102.	127.	158.
197.	246.										

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
       GRID NODE LOCATIONS ALONG THE PAGE

       AXIS IS POSITIVE UP THE PAGE

0.000    3.00    6.00    9.00    12.0    15.0    18.0    21.0    24.0    27.0    30.0
    --------------------------------------------------------------------------------------------


                         *** TRANSIENT DATA ***


                         *** TRANSIENT WELL DATA ***
            WELL     FLOW RATE    SURFACE    WELL DATUM   PRESSURE    INJECTION        INJECTION
            NO.      ( M**3/S)   PRESSURE    PRESSURE     LIMITED?   TEMPERATURE     MASS   FRACTION
                                   ( PA)        ( PA)                   (DEG.C)         (-)
            ...........................................................................................
             1     -0.2030                                   NO

                         *** CALCULATION INFORMATION ***

            FIXED TIME STEP LENGTH .................................... DELTIM  2.592E+05  (S);   3.00     (D)
            TIME AT WHICH NEXT SET OF TIME VARYING
                 PARAMETERS WILL BE READ .............................. TIMCHG  1.555E+07  (S);   180.     (D)
                                   *** MAPPING DATA ***
            -------------------------------------------------------------------------------------------
                      LENGTH OF X-AXIS ..............12.0  (IN.)
                      LENGTH OF Y-AXIS .............. 0.0  (IN.)
                      LENGTH OF Z-AXIS .............. 6.0  (IN.)
                      Z-AXIS IS POSITIVE UPWARD
            -------------------------------------------------------------------------------------------
                 MAP NO.   I1   I2   J1   J2   K1   K2      MINIMUM VALUE     MAXIMUM VALUE   NUMBER OF
                                                             OF VARIABLE       OF VARIABLE     ZONES
            -------------------------------------------------------------------------------------------
                      2    1   26    1    1    1   11            20.               60.            8

    --------------------------------------------------------------------------------------------


                         *** OUTPUT AT END OF TIME STEP NO.    40 ***

                      TIME .............................................. 1.037E+07 (S);     120.     (D)
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
CURRENT TIME STEP LENGTH .................................. 2.592E+05 (S);     3.00     (D)

NO. OF P,T,C LOOP ITERATIONS USED ......................... 2
MAXIMUM CHANGE IN PRESSURE ................................ -3.3222E+02 ( PA)   AT LOCATION ( 1, 1,11)
MAXIMUM CHANGE IN TEMPERATURE ............................. -1.4874E+00 (DEG.C) AT LOCATION (23, 1,11)
```

TEMPERATURE (DEG.C)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	59.22	59.22	59.22	59.22	59.22	59.22	59.22	59.23	59.24	59.24
10	59.17	59.17	59.17	59.17	59.17	59.17	59.17	59.18	59.18	59.19
9	59.02	59.02	59.02	59.02	59.02	59.02	59.02	59.02	59.03	59.04
8	58.77	58.77	58.77	58.77	58.78	58.78	58.78	58.78	58.78	58.79
7	58.46	58.46	58.46	58.46	58.46	58.46	58.46	58.46	58.46	58.46
6	58.09	58.09	58.09	58.09	58.09	58.09	58.09	58.09	58.09	58.08
5	57.70	57.70	57.70	57.70	57.70	57.70	57.70	57.70	57.69	57.68
4	57.33	57.33	57.33	57.33	57.33	57.33	57.32	57.32	57.31	57.29
3	57.02	57.02	57.02	57.02	57.02	57.02	57.01	57.00	56.99	56.97
2	56.81	56.81	56.81	56.81	56.81	56.80	56.80	56.79	56.77	56.74
1	56.73	56.73	56.73	56.73	56.73	56.73	56.72	56.71	56.69	56.66

	11	12	13	14	15	16	17	18	19	20
11	59.26	59.27	59.28	59.29	59.30	59.29	59.26	59.18	59.01	58.62
10	59.20	59.21	59.23	59.23	59.24	59.22	59.18	59.09	58.88	58.43
9	59.04	59.05	59.06	59.06	59.05	59.02	58.95	58.81	58.52	57.91
8	58.79	58.79	58.79	58.78	58.75	58.69	58.58	58.36	57.94	57.09
7	58.46	58.45	58.44	58.41	58.36	58.26	58.08	57.76	57.17	56.02
6	58.07	58.06	58.03	57.98	57.89	57.74	57.49	57.05	56.25	54.75
5	57.66	57.63	57.59	57.51	57.39	57.19	56.85	56.27	55.24	53.36
4	57.27	57.23	57.16	57.06	56.90	56.63	56.20	55.48	54.22	51.95
3	56.93	56.88	56.80	56.67	56.47	56.15	55.63	54.77	53.29	50.66
2	56.70	56.64	56.54	56.40	56.17	55.80	55.21	54.24	52.56	49.65
1	56.62	56.55	56.45	56.29	56.04	55.65	55.02	53.97	52.18	49.09

	21	22	23	24	25	26
11	57.71	55.38	48.87	33.85	22.08	20.07
10	57.38	54.67	47.43	32.84	21.94	20.00

Table 8.8.--*Selections from the output file for example problem 2*--Continued

9	56.54	53.20	45.16	31.41	21.70	19.98
8	55.27	51.14	42.46	29.83	21.45	19.97
7	53.65	48.70	39.62	28.27	21.20	19.98
6	51.79	46.05	36.82	26.82	20.99	19.98
5	49.79	43.37	34.22	25.55	20.81	19.98
4	47.82	40.88	31.96	24.50	20.67	19.98
3	46.06	38.78	30.18	23.71	20.58	19.98
2	44.71	37.29	29.03	23.23	20.51	19.98
1	43.97	36.58	28.57	23.05	20.49	19.98

VERTICAL PLANE AT ROW NO. 1

```
         VERTICAL CROSS-SECTION OF SIMULATION REGION -     TEMPERATURES                                 PAGE 1

****4**4*4*4*444*444*4444*44444*4444444*444444444*        33*3333    22222 *      11111111    *                        *
44444444444444444444444444444444444444444444444444        333333     22222       111111111
44444444444444444444444444444444444444444444444444       3333333    222222       11111111
4444444444444444444444444444444444444444444444444        3333333    22222        11111111
*444444444444444444444444444444444444444444444444       3333333    22222       111111111                         *
44444444444444444444444444444444444444444444444        33333333   222222       111111111                         -
4444444444444444444444444444444444444444444444         3333333    22222      1111111111                          -
444444444444444444444444444444444444444444444         3333333   222222       111111111                           -
*4444444444444444444444444444444444444444444         33333333   22222      1111111111                            *
4444444444444444444444444444444444444444444         3333333   222222     11111111111                             -
44444444444444444444444444444444444444444          3333333   222222     11111111111                              -
*444444444444444444444444444444444444444          3333333   222222     11111111111                               *
444444444444444444444444444444444444444          3333333   222222     11111111111                                -
4444444444444444444444444444444444444          3333333   2222222    11111111111                                  -
444444444444444444444444444444444444          3333333   2222222    11111111111                                   -
*4444444444444444444444444444444444          3333333   2222222    11111111111                                    *
444444444444444444444444444444444          33333333   2222222    11111111111                                     -
44444444444444444444444444444444          33333333   2222222    111111111111                                     -
*444444444444444444444444444444          33333333   2222222    11111111111                                       *
44444444444444444444444444444          33333333   222222     11111111111                                         -
4444444444444444444444444444          33333333   222222     11111111111                                          -
444444444444444444444444444          33333333   222222     11111111111                                           -
*4444444444444444444444444          33333333   222222     11111111111                                            *
4444444444444444444444444          33333333   2222222    11111111111                                             -
444444444444444444444444          33333333   2222222    1111111111                                               -
44444444444444444444444          33333333    222222     1111111111                                               -
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
*44444444444444444444          333333333      222222         1111111111                                   *
44444444444444444444          333333333      2222222        11111111111                                   -
44444444444444444444         333333333      2222222       111111111111                                    -
*4444444444444444444         33333333      2222222       111111111111                                     *
444444444444444444          33333333      2222222       1111111111111                                     -
44444444444444444           33333333      2222222       1111111111111                                     -
44444444444444444          333333333     2222222       111111111111                                       -
*444444444444444           33333333      2222222      1111111111111                                       *
4444444444444444          333333333     2222222       111111111111                                        -
4444444444444444          33333333     2222222        111111111111                                        -
****4**4*4*4*444*    *    *  3*333333 *    2222*22      11*1111111111    *             *                 *



                     MAP LEGEND
            HORIZONTAL GRID NODE RANGE, FROM   1 TO 26
            VERTICAL GRID NODE RANGE, FROM   1 TO 11
            DEPENDENT VARIABLE RANGE   MAP CHARACTER

               2.000E+01 -  2.500E+01
               2.500E+01 -  3.000E+01        1
               3.000E+01 -  3.500E+01
               3.500E+01 -  4.000E+01        2
               4.000E+01 -  4.500E+01
               4.500E+01 -  5.000E+01        3
               5.000E+01 -  5.500E+01
               5.500E+01 -  6.000E+01        4

        GRID NODE LOCATIONS ACROSS THE PAGE
 1.00     1.25     1.55     1.94     2.41     3.01     3.75     4.67     5.82     7.26     9.04     11.3
 14.0     17.5     21.8     27.2     33.9     42.3     52.7     65.6     81.8     102.     127.     158.
 197.     246.

        GRID NODE LOCATIONS ALONG THE PAGE

        AXIS IS POSITIVE UP THE PAGE

0.000     3.00     6.00     9.00     12.0     15.0     18.0     21.0     24.0     27.0     30.0


                        *** OUTPUT AT END OF TIME STEP NO.    60 ***

                   TIME ..................................................  1.555E+07 (S);     180.    (D)

              CURRENT TIME STEP LENGTH ...................................  2.592E+05 (S);     3.00    (D)
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
NO. OF P,T,C LOOP ITERATIONS USED .........................  2
MAXIMUM CHANGE IN PRESSURE ................................  -7.1829E+02 ( PA)   AT LOCATION ( 1, 1,11)
MAXIMUM CHANGE IN TEMPERATURE .............................  -1.0762E+00 (DEG.C) AT LOCATION (15, 1,11)
```

TEMPERATURE (DEG.C)

VERTICAL SLICES

J = 1

	1	2	3	4	5	6	7	8	9	10
11	39.59	39.59	39.59	39.59	39.59	39.59	39.60	39.61	39.62	39.63
10	39.48	39.48	39.48	39.48	39.49	39.49	39.49	39.50	39.51	39.52
9	39.18	39.18	39.18	39.18	39.18	39.18	39.19	39.19	39.20	39.21
8	38.70	38.70	38.70	38.71	38.71	38.71	38.71	38.71	38.72	38.72
7	38.11	38.11	38.11	38.11	38.11	38.11	38.11	38.11	38.11	38.10
6	37.45	37.45	37.45	37.45	37.45	37.45	37.44	37.44	37.43	37.42
5	36.79	36.79	36.79	36.79	36.78	36.78	36.78	36.77	36.76	36.74
4	36.19	36.19	36.19	36.19	36.19	36.18	36.18	36.17	36.15	36.12
3	35.72	35.72	35.72	35.71	35.71	35.71	35.70	35.69	35.67	35.64
2	35.41	35.41	35.41	35.41	35.41	35.40	35.39	35.38	35.36	35.32
1	35.31	35.31	35.31	35.30	35.30	35.30	35.29	35.27	35.25	35.21

	11	12	13	14	15	16	17	18	19	20
11	39.65	39.66	39.66	39.64	39.59	39.46	39.21	38.76	37.99	36.72
10	39.54	39.55	39.55	39.52	39.46	39.32	39.06	38.60	37.82	36.52
9	39.22	39.22	39.21	39.17	39.09	38.93	38.65	38.15	37.32	35.98
8	38.72	38.71	38.68	38.63	38.53	38.34	38.01	37.46	36.58	35.18
7	38.09	38.07	38.03	37.95	37.82	37.59	37.22	36.62	35.67	34.21
6	37.40	37.36	37.30	37.20	37.04	36.77	36.35	35.70	34.69	33.18
5	36.71	36.66	36.58	36.45	36.26	35.96	35.50	34.79	33.73	32.18
4	36.08	36.02	35.93	35.78	35.56	35.23	34.73	33.99	32.89	31.31
3	35.59	35.52	35.41	35.25	35.01	34.66	34.13	33.35	32.22	30.62
2	35.27	35.19	35.08	34.91	34.66	34.29	33.74	32.95	31.80	30.19
1	35.16	35.08	34.96	34.79	34.54	34.16	33.61	32.81	31.65	30.04

	21	22	23	24	25	26
11	34.68	31.54	27.13	22.21	19.60	19.94
10	34.46	31.32	26.96	22.14	19.60	19.93
9	33.88	30.75	26.51	21.97	19.62	19.92

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
  8     33.04     29.94     25.89     21.75     19.65     19.93
  7     32.04     29.00     25.20     21.51     19.68     19.95
  6     30.99     28.04     24.52     21.27     19.72     19.96
  5     30.00     27.15     23.90     21.06     19.76     19.96
  4     29.14     26.39     23.39     20.89     19.80     19.97
  3     28.47     25.81     23.00     20.77     19.82     19.97
  2     28.05     25.45     22.76     20.69     19.84     19.97
  1     27.90     25.32     22.67     20.67     19.85     19.97


                         *** GLOBAL FLOW BALANCE SUMMARY ***
                              CURRENT TIME STEP                          RATES                    AMOUNTS

         FLUID INFLOW ................................................ 2.022665E+02  (KG/S)     5.242748E+07  (KG)
         FLUID OUTFLOW ............................................... 2.015989E+02  (KG/S)     5.225444E+07  (KG)
         CHANGE IN FLUID IN REGION ................................... 6.675322E-01  (KG/S)     1.730243E+05  (KG)
         RESIDUAL IMBALANCE ..........................................-5.618930E-05  (KG/S)    -1.456427E+01  (KG)
         FRACTIONAL IMBALANCE ........................................                              0.0000

         HEAT INFLOW ................................................. 1.685565E+07  (  J/S)     4.368984E+12  (  J)
         HEAT OUTFLOW ................................................ 3.154910E+07  (  J/S)     8.177527E+12  (  J)
         CHANGE IN HEAT IN REGION ....................................-1.469188E+07  (  J/S)    -3.808136E+12  (  J)
         RESIDUAL IMBALANCE .......................................... 1.570947E+03  (  J/S)     4.071895E+08  (  J)
         FRACTIONAL IMBALANCE ........................................                              0.0000

                              CUMULATIVE SUMMARY
                                                                        AMOUNTS

         FLUID INFLOW ................................................ 3.126293E+09  (KG)
         FLUID OUTFLOW ............................................... 3.128713E+09  (KG)
         CHANGE IN FLUID IN REGION ...................................-2.415865E+06  (KG)
         FLUID IN REGION ............................................. 1.993776E+09  (KG)
         RESIDUAL IMBALANCE .......................................... 3.698014E+03  (KG)
         FRACTIONAL IMBALANCE ........................................   0.0000

         HEAT  INFLOW ................................................ 5.206214E+14  (  J)
         HEAT  OUTFLOW ............................................... 4.673636E+14  (  J)
         CHANGE IN HEAT IN REGION .................................... 5.326113E+13  (  J)
         HEAT IN REGION .............................................. 3.846566E+14  (  J)
         RESIDUAL IMBALANCE .......................................... 3.332536E+09  (  J)
         FRACTIONAL IMBALANCE ........................................   0.0000

         CUMULATIVE SPECIFIED P CELL FLUID NET INFLOW ................ 2.131608E+06  (KG)
         CUMULATIVE FLUX B.C. FLUID NET INFLOW ....................... 0.000000E-01  (KG)
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
CUMULATIVE LEAKAGE B.C. FLUID NET INFLOW ..................  0.000000E-01  (KG)
CUMULATIVE AQUIFER INFLUENCE FLUID NET INFLOW .............  0.000000E-01  (KG)


CUMULATIVE SPECIFIED T CELL OR ASSOCIATED WITH
     SPECIFIED P CELL HEAT NET INFLOW...................... -5.301546E+11  (  J)
CUMULATIVE FLUX B.C. HEAT NET INFLOW ......................  0.000000E-01  (  J)
CUMULATIVE LEAKAGE B.C. HEAT NET INFLOW ...................  0.000000E-01  (  J)
CUMULATIVE AQUIFER INFLUENCE HEAT NET INFLOW ..............  0.000000E-01  (  J)
CUMULATIVE HEAT CONDUCTION B.C. HEAT NET INFLOW ...........  0.000000E-01  (  J)


                               *** WELL SUMMARY ***

                  FLOW RATES (POSITIVE IS INJECTION)   CUMULATIVE PRODUCTION          CUMULATIVE INJECTION
WELL LOCATION      FLUID     HEAT      SOLUTE         FLUID     HEAT      SOLUTE         FLUID     HEAT      SOLUTE
NO. I  J   K      (KG/S)   (  J/S)    (KG/S)          (KG)     (  J)     (KG)           (KG)     (  J)     (KG)
------------------------------------------------------------------------------------------------------------------
 1  1  1  1-11     -202.    -3.155E+07                1.56E+09  3.36E+14                1.55E+09  3.90E+14
------------------------------------------------------------------------------------------------------------------
TOTAL - PRODUCTION  202.     3.155E+07                1.56E+09  3.36E+14
      - INJECTION  0.000     0.000                                                      1.55E+09  3.90E+14

                  THE FOLLOWING PARAMETERS WERE IN EFFECT DURING THE TIME STEP JUST COMPLETED
WELL   TOP COMPLETION LAYER   WELL DATUM       WELL HEAD      WELL DATUM      WELL HEAD            MASS
NO.       CELL PRESSURE        PRESSURE        PRESSURE       TEMPERATURE     TEMPERATURE         FRACTION
            ( PA)               ( PA)          ( PA)           (DEG.C)         (DEG.C)             (-)
------------------------------------------------------------------------------------------------------------------
 1         -4.1870E+04        -4.1870E+04                        38.4             0.0

          PER LAYER FLUID PRODUCTION/INJECTION RATES- (KG/S) (INJECTION IS POSITIVE)
------------------------------------------------------------------------------------------------------------------

LAYER NO.                WELL NUMBER
           1
11      -10.5
10      -21.0
 9      -20.9
 8      -20.7
 7      -20.4
 6      -20.2
 5      -19.9
 4      -19.6
 3      -19.4
 2      -19.3
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
 1     -9.64

        PER LAYER HEAT PRODUCTION/INJECTION RATES- (  J/S) (INJECTION IS POSITIVE)
```

LAYER NO.	WELL NUMBER 1
11	-1.739E+06
10	-3.461E+06
9	-3.414E+06
8	-3.340E+06
7	-3.249E+06
6	-3.150E+06
5	-3.053E+06
4	-2.967E+06
3	-2.899E+06
2	-2.856E+06
1	-1.421E+06

```
                 VERTICAL CROSS-SECTION OF SIMULATION REGION -     TEMPERATURES                    PAGE 1

****2**2*2*2*222*222*2222*22222*222222 *         *    1111111*111111         *                --*------------------------*
22222222222222222222222222222222222222                11111111111111                          ----------------------------
22222222222222222222222222222222222222               111111111111111                          ----------------------------
22222222222222222222222222222222222222               111111111111111                          ----------------------------
*222222222222222222222222222222222222                111111111111111                          ---------------------------*
2222222222222222222222222222222222222                111111111111111                          ----------------------------
222222222222222222222222222222222222                111111111111111                           ----------------------------
222222222222222222222222222222222222                111111111111111                          -----------------------------
*2222222222222222222222222222222222                 111111111111111                          ----------------------------*
2222222222222222222222222222222222                 111111111111111                           -----------------------------
222222222222222222222222222222222                 1111111111111111                           -----------------------------
*2222222222222222222222222222222                 1111111111111111                            ----------------------------*
2222222222222222222222222222222                  111111111111111                             -----------------------------
222222222222222222222222222222                  1111111111111111                             -----------------------------
22222222222222222222222222222                  1111111111111111                              -----------------------------
*222222222222222222222222222                  1111111111111111                               ----------------------------*
222222222222222222222222222                  1111111111111111                                -----------------------------
22222222222222222222222222                  11111111111111111                                -----------------------------
*22222222222222222222222                   11111111111111111                                 ----------------------------*
22222222222222222222222                   11111111111111111                                  -----------------------------
2222222222222222222222                   111111111111111111                                  -----------------------------
222222222222222222222                   111111111111111111                                   -----------------------------
*222222222222222222                     11111111111111111                                    ----------------------------*
```

Table 8.8.--*Selections from the output file for example problem 2*--Continued

```
222222222222222222                     111111111111111111                                    ----------------------------
22222222222222222                     111111111111111111                                     ----------------------------
2222222222222222                     1111111111111111111                                     ----------------------------
*2222222222222                      1111111111111111111                                      ---------------------------*
2222222222222                       111111111111111111                                       ----------------------------
222222222222                       1111111111111111111                                       ----------------------------
*2222222222                       1111111111111111111                                        ---------------------------*
2222222222                       11111111111111111111                                        ----------------------------
222222222                        1111111111111111111                                         ----------------------------
22222222                         1111111111111111111                                         ----------------------------
*222222                         11111111111111111111                                         ---------------------------*
2222222                         1111111111111111111                                          ----------------------------
2222222                         1111111111111111111                                          ----------------------------
****2** * * *   *   *    *     *1111111*111111111*1           *               *              ---*-----------------------*
```

```
                        MAP LEGEND
              HORIZONTAL GRID NODE RANGE, FROM   1 TO 26
              VERTICAL GRID NODE RANGE, FROM   1 TO 11
              DEPENDENT VARIABLE RANGE   MAP CHARACTER

                 2.000E+01 -  2.500E+01
                 2.500E+01 -  3.000E+01       1
                 3.000E+01 -  3.500E+01
                 3.500E+01 -  4.000E+01       2
                 4.000E+01 -  4.500E+01
                 4.500E+01 -  5.000E+01       3
                 5.000E+01 -  5.500E+01
                 5.500E+01 -  6.000E+01       4

        GRID NODE LOCATIONS ACROSS THE PAGE
 1.00     1.25     1.55     1.94     2.41     3.01     3.75     4.67     5.82     7.26     9.04     11.3
 14.0     17.5     21.8     27.2     33.9     42.3     52.7     65.6     81.8     102.     127.     158.
 197.     246.

        GRID NODE LOCATIONS ALONG THE PAGE
        AXIS IS POSITIVE UP THE PAGE
0.000     3.00     6.00     9.00     12.0     15.0     18.0     21.0     24.0     27.0     30.0

                           ***** JOB COMPLETED *****
             LAST TIME PLANE CALCULATED ....................................  1.56E+07  (S)
             LAST TIME PLANE INDEX .........................................     60
```

9. NOTATION

9.1 ROMAN

a_i coefficients of the various u_{ijk} in the spatially discretized equations.

a'_i coefficients of the various u_{ik} in the spatially discretized equations for the cylindrical-coordinate system.

a_1 implicit term for aquifer-influence-function or heat-conduction boundary conditions (appropriate units).

a_2 implicit term coefficient for aquifer-influence-function or heat-conduction boundary conditions (appropriate units).

$\underline{\underline{A}}$ matrix of coefficients for the $\underline{\delta u}$ vector for the finite-difference equations (eq. 3.6.2.1a).

$\underline{\underline{A}}_i$ Banded, sparse submatrices of the matrix $\underline{\underline{A}}$ for the finite-difference equations.

A capacitance coefficient (appropriate units).

A_{ms} value of capacitance coefficient within subdomain s of cell m.

$\underline{b}$ vector of known terms on the right-hand side for the finite-difference equations.

b_e thickness of an outer-aquifer region (m).

b_{HC} effective thickness of a heat-conducting medium exterior to the region (m).

b_L thickness of an aquitard layer (m).

b_R thickness of a riverbed sediment (m).

b_i coefficient in the approximation to the transient aquifer-influence function given by equation 3.4.4.2.5.

B coefficient of dispersion or diffusion (appropriate units).

B_0, B_1 parameters for the temperature-dependence of viscosity (°C).

$\underline{\underline{B}}$ tensor of diffusion or dispersion of rank 3 (appropriate units).

B_{ij} tensor components of $\underline{\underline{B}}$.

$\overline{c_f}$ average heat capacity of the fluid phase at constant pressure (J/kg-°C).

c_{fo} heat capacity of pure water at constant pressure (J/kg-°C)

c_s heat capacity of the solid phase (porous matrix) at constant pressure (J/kg-°C).

c_{se} heat capacity of the exterior heat-conducting medium at co... pressure (J/kg-°C).

C_{ij} coefficient in the capacitance matrix for the discretized equations (appropriate units).

$\underline{C}$ vector of interstitial velocity (m/s).

C_i (m,p,q) vector components of $\underline{C}$ in the ith direction, for the cell m, face p, in element q (m/s).

d input data item.

d_{pi} coefficient in the series expansion for pressure (eq. 3.3.2.4c) (Pa/m).

d_{Ti} coefficient in the series expansion for temperature (eq. 3.3.2.4d) (°C/m).

$\underline{d}$ coefficient vector for series expansion (eq. 3.3.2.5b).

$\underline{\underline{D}}_i$ submatrices on the diagonal of matrix $\underline{\underline{A}}$.

D_m effective-molecular-diffusivity coefficient of the solute (m^2/s).

D_{Hrm} thermal diffusivity of the medium surrounding a well riser (m^2/s).

$\underline{\underline{D}}_H$ thermo-mechanical dispersion tensor from flow mechanisms (W/m-°C).

D_{He} thermal diffusivity for an exterior medium at a heat-conduction boundary (m^2/s).

D_{Hij} thermo-mechanical-dispersion-tensor component (W/m-°C).

$\underline{\underline{D}}_S$ mechanical-dispersion-coefficient tensor (m^2/s).

D_{Sij} mechanical-dispersion-tensor component (m^2/s).

D^*_{Sij} hydrodynamic-dispersion coefficient (m^2/s).

D^*_{Hij} thermo-hydrodynamic-dispersion coefficient (W/m-°C).

$\mathcal{D}$ coefficient for a source proportional to the dependent variable (appropriate units).

$\mathcal{D}_{ms}$ value of $\mathcal{D}$ in cell m for subdomain s.

$\underline{E}_i$ coefficient vector for a node; i = 1 for flow equation, i = 2 for heat-transport equation, i = 3 for solute-transport equation.

E source-term intensity (appropriate units).

$E_{s(m)}$ source-term intensity for source s in cell m (appropriate units).

E_m value of source term in cell m.

f_{FS} fraction of cell thickness that is saturated for a free-surface boundary condition (-).

f_r head-loss friction factor for a well riser (-).

f_w head-loss friction factor for a well bore (-).

f_θ angle-of-influence factor for aquifer-influence-function boundary condition (-).

F_{CJ} heat-flux function (Carslaw and Jaeger) to an infinite medium from a constant-temperature cylindrical source (-).

F_i vector component of the known terms at time level n in the three discretized system equations for a given node, i = 1,2,3.

F spatial finite-difference function.

F^n spatial finite-difference function evaluated at time level n.

F_{CJ}^S approximation to F_{CJ} function for small dimensionless time.

F_{CJ}^L approximation to F_{CJ} function for large dimensionless time.

$\mathcal{F}$ function for the well-riser calculation (eq. 2.4.2.11).

g gravitational constant (m/s^2).

g_p component of gravitational acceleration in the direction normal to the face p.

G function for the well-riser calculation (eq. 2.4.2.11).

h_r heat-transfer coefficient from the fluid to a riser pipe (W/m^2-°C).

H specific enthalpy of the fluid phase (J/kg).

H_B specific enthalpy of fluid at a region boundary (J/kg).

H_e specific enthalpy of fluid in an outer aquifer (J/kg).

H_r specific enthalpy of fluid in a well riser (J/kg).

H_s specific enthalpy of the solid phase (J/kg).

$\underline{J}$ vector of specified total flux at a boundary (appropriate units).

J_i component of specified flux at a boundary in the ith direction.

J_{iq} component of specified flux at a boundary in the ith direction in element q.

$\underline{\underline{k}}$ porous-medium permeability tensor (m^2).

k_e permeability of an outer-aquifer region (m^2).

k_L permeability of an aquitard layer (m^2).

k_R permeability of a riverbed sediment (m^2).

k_w mean permeability around a well between r_w and r_e (m^2).

k_x permeability in the x-direction (m^2).

k_y permeability in the y-direction (m^2).

k_{pq} permeability in the direction normal to face p for subdomain q (m^2).

$k_{wm(\ell)}$ average radial permeability around the well in cell m at level ℓ (m^2).

k_{mpq} permeability for cell m, face p, subdomain q (m^2).

K_d equilibrium-distribution coefficient (m^3/kg).

K_e thermal conductivity of a medium exterior to the simulation region (W/m-°C).

K_f thermal conductivity of the fluid (W/m-°C).

K_r thermal conductivity of a riser pipe (W/m-°C).

K_{re} thermal conductivity of the medium surrounding a riser pipe (W/m-°C).

K_s thermal conductivity of the solid phase (porous matrix) (W/m-°C).

$\mathcal{K}_s$ Augmented porosity factor for subdomain s (eq. 3.1.4.4k) (-).

ℓ distance along the well bore or well riser (m).

ℓ_L lower end of a well-screen interval (m).

$\hat{\ell}_L$ index of the bottom level of a well screen.

ℓ_U upper end of a well-screen interval (m).

$\hat{\ell}_U$ index of the top level of a well screen.

L_r length of a well riser (m).

$L_{\ell 1}$ length of a well bore in the lower half of cell m at level ℓ (m).

$L_{\ell 2}$ length of a well bore in the upper half of cell m at level ℓ (m).

$\underline{\underline{L}}_i$ submatrices below the diagonal of matrix $\underline{\underline{A}}$.

m cell number with coordinate indices i,j,k.

m(i-1) cell number with coordinate indices i-1,j,k.

m(ℓ) cell number associated with well bore level ℓ.

M total number of nodes in the simulation region.

M_A number of pot-aquifer boundary-condition cells.

M_w well mobility per unit length of well bore (m^3/s-m-Pa).

$M_{w\ell}$ well mobility per unit length of well bore at level ℓ (m^3/s-m-Pa).

$\mathcal{M}_m^n$ mass of fluid plus effective additional fluid mass from sorption in cell m (kg).

n data repeat factor.

$\underline{n}$ vector in the outward normal direction to the boundary.

N_r number of grid points in the r-direction.

N_x number of grid points in the x-direction.

N_y number of grid points in the y-direction.

N_z number of grid points in the z-direction.

N_s number of line sources in the region.

p fluid pressure relative to atmospheric pressure (Pa).

$\hat{p}$ absolute pressure (Pa).

p_{av} average pressure around a well between r_w and r_e (Pa).

p_B pressure at a boundary (Pa).

p'_B dimensionless pressure at an aquifer-influence-function boundary (-) (eq. 2.5.4.2.6b).

p_e pressure in the outer-aquifer region (Pa).

p_e^0 initial specified pressure in an outer-aquifer region (Pa).

p_{inj} pressure of injected fluid (Pa).

p_m pressure at node m (Pa).

p_o pressure at a reference state for density (Pa).

p_{oH} pressure at a reference state for enthalpy (Pa).

$\hat{p}_{oH}$ absolute pressure at a reference state for enthalpy (Pa).

p_r pressure averaged across a riser cross section (Pa).

p_{rk} pressure averaged across a riser cross section at location ℓ_k (Pa).

p_r^* intermediate value of pressure averaged over a riser cross section (Pa).

$\hat{p}_{sat}$ absolute pressure of saturated water at zero degrees Celsius (Pa).

p_w pressure at the well bore (Pa).

p_{wd} pressure at a well datum (Pa).

$p_{w\ell}$ pressure in the well bore at level ℓ (Pa).

p^0 initial pressure distribution (Pa).

P'_U dimensionless pressure response to a unit-step withdrawal flow rate (-).

q fluid-source, flow-rate intensity (m^3/m^3-s).

q_{Fi}^B specified fluid-flux-vector components (m^3/m^2-s).

q_{Fn} normal component of fluid flux (m^3/m^2-s).

q_{fw} fluid flux from a well (m^3/m^2-s).

q_H heat-source-rate intensity (W/m^3).

q_{HC} heat flux at the heat-conduction boundary (W/m^2).

q_{Hi}^B specified heat-flux-vector components (W/m^2).

q_{HL} heat flux across a leakage boundary (W/m^2).

q_{Hn} normal component of heat flux (W/m^2).

q_L fluid flux across a leakage boundary (m^3/m^2-s).

q_R fluid flux across a river-leakage boundary (m^3/m^2-s).

q_{Rmax} limit on the fluid flux from a river to the aquifer (m^3/m^2-s).

q_{Si}^B specified solute-flux-vector components (kg/m^2-s).

q_{SL} solute flux across a leakage boundary (kg/m^2-s).

q_{Sn} normal component of solute flux (kg/m^2-s).

q_w volumetric flow rate per unit length of well bore; (positive is from the well to the aquifer) (m^3/m-s).

Q fluid-mass-source flow rate (kg/s).

Q'_A dimensionless flow rate per unit thickness at an aquifer-influence-function boundary (-).

Q_A constant specified flow rate at an aquifer-influence-function boundary (m^3/s).

Q_{Am} volumetric flow rate across the boundary for cell m between the inner- and outer-aquifer regions; (positive is into the inner region) (m^3/s).

Q_B specified flow rate per unit thickness at the aquifer-influence-function boundary of the region (m^3/m-s).

Q_{Fr} mass flow rate in a well riser (kg/s).

Q_{HCm} heat-flow rate across a heat-conduction boundary at cell m (W).

Q_{Hr} heat transfered per unit mass per unit length to the fluid in a riser (J/kg-m).

Q_{Lm} volumetric-flow rate across a leakage boundary at cell m (m^3/s).

Q_m^n volumetric-flow rate of a source term for cell m at time level n including specified-flux boundary condition and line sources (m^3/s).

Q'_U dimensionless flow-rate response to a unit step change of pressure.

Q_{Rm} volumetric-flow rate across a river-leakage boundary at cell m (m^3/s).

Q_w volumetric-flow rate from a well to the aquifer (m^3/s).

$Q_{w\ell}$ volumetric-flow rate from a well to the aquifer at well bore level ℓ (m^3/s).

r distance along the r-coordinate direction (m).

r_e radius of influence of a well (m).

r_E exterior radius of an outer-aquifer region (m).

r_1 interior radius of the region, cylindrical coordinates (m).

$r_{i+\frac{1}{2}}$ radius of the cell boundary between the node at r_i and the node at r_{i+1} (m).

r_I interior radius of outer-aquifer region (m).

r_{N_r} exterior radius of the region, cylindrical coordinates (m).

r_r inner radius of a well-riser pipe (m).

r_w well-bore radius (m).

R ratio of exterior to interior radius for an outer-aquifer region for the aquifer-influence-function boundary condition (-).

R_{fs} transfer of solute from fluid to solid phase per unit mass of solid phase (kg/s-kg).

R_i right-hand side terms for the ith equation, i = 1,2,3 for pressure, temperature, and mass fraction respectively.

$\underline{R}_m$ rational-function vector that approximates the right-hand-side of equation 3.3.2.6.

$R(L_1)$ spectral radius of the Gauss-Seidel iteration matrix.

S_o specific storage for a confined aquifer (eq. 2.2.5.5) (m^{-1}).

$S_{A\ell pq}$ area of the aquifer-influence-function boundary face for cell ℓ, face p, element q (m^2).

S_m boundary surface of cell m (m^2).

S_{mp} part of the boundary surface of cell m that belongs to face p (m^2).

S_{mpq} part of the boundary surface of cell m that belongs to face p in element q (m^2).

S_{Ampq} part of the boundary surface of cell m that is an aquifer-influence-function boundary (m^2).

S_{BLm} part of the boundary surface of cell m that is a leakage boundary (m^2).

S_{BHCm} part of the boundary surface of cell m that is a heat-conduction boundary (m^2).

S_u^1 parts of the region boundary where specified-value boundary conditions are applied (Dirchlet boundary conditions); u = p, T, or w.

S_u^2 parts of the region boundary where specified-flux boundary conditions are applied (Neumann boundary conditions); u = p, T, or w.

S^3 part of the boundary that is an aquifer-leakage boundary.

S^4 part of the boundary that is a river-leakage boundary.

S^5 part of the region boundary that is a heat-conduction boundary.

S^6 part of the region boundary that is a free-surface boundary.

t time (s).

t^n time at level n (s).

t' dimensionless time defined by equation 2.5.4.2.5c (-).

T temperature of the fluid and porous medium (°C).

T_a ambient temperature of the medium adjacent to a well riser (°C).

T_B temperature at a boundary (°C).

T_e temperature in the medium exterior to a heat-conduction boundary (°C).

T_o temperature at a reference state for density (°C).

T_{oH} temperature at a reference state for enthalpy (°C).

T_{ov} temperature at a reference state for fluid viscosity (°C).

T_r temperature of the fluid averaged across a riser cross section (°C).

T_1 temperature solution to the first heat-conduction problem (eq. 2.5.5.3a-d) (°C).

T_2 temperature solution to the second heat-conduction problem (eq. 2.5.5.4a-d) (°C).

T^* temperature of a fluid source (°C).

T^0 initial-temperature distribution (°C).

T_e^0 initial-temperature profile in the exterior medium (°C).

T_{inj} temperature of injected fluid (°C).

T_U unit-step temperature-response solution to heat-conduction problem (eq. 2.5.5.6b) (°C).

T_{wd} temperature of the fluid at a well datum (°C).

$\hat{T}$ absolute temperature (K).

$\mathcal{T}_{Fi}$ conductance for flow (kg/s-Pa) or (m-s).

$\mathcal{T}_{Hi}$ conductance for heat transport (W/°C).

$\mathcal{T}_{Si}$ conductance for solute transport (kg/s).

$\mathcal{T}_{wF}$ conductance for flow at a well bore in the cylindrical coordinate system (m-s).

u generic dependent variable (appropriate units).

u_B boundary-condition distribution of u.

u^0 initial-condition distribution of u.

u_{ijk}^n value of u at x_i, y_j, z_k at time t^n.

u_{ik}^n value of u at x_i, z_k in cylindrical-coordinate system at time t^n.

U_T overall heat-transfer coefficient for the fluid, riser-pipe, and surrounding medium (W/m²-°C).

$\underline{\underline{U}}_i$ submatrices of matrix $\underline{\underline{A}}$ above the diagonal.

$\underline{v}$ interstitial-velocity vector (m/s).

v magnitude of the interstitial velocity (m/s).

v_i interstitial-velocity component in the ith direction (m/s).

v_r velocity averaged across a riser cross section at a given z-level (m/s).

v_w velocity averaged across a well-bore cross section at a given z-level (m/s).

V region of the aquifer for simulation.

V_b bulk or total volume of a fixed mass of porous medium (m³).

V_e volume of an outer-aquifer region (m³).

V_ℓ volume of an outer-aquifer region that influences boundary cell ℓ (m^3).

V_m volume of cell m (m^3).

V_{ms} volume of subdomain s in cell m (m^3).

w mass fraction of solute in the fluid phase (-).

w_B mass fraction of solute at a boundary (-).

w_e mass fraction of solute in an outer aquifer outside a leakage boundary (-).

w_{min} minimum solute-mass fraction for scaling (-).

w_{max} maximum solute-mass fraction for scaling (-).

w_o mass fraction of solute at a reference state for density (-).

w' scaled-mass fraction of solute defined by equation 2.2.1.2 (-).

w^0 mass fraction of solute initial distribution (-).

w^* mass fraction of solute in the fluid source (-).

$\bar{w}$ mass fraction of solute on the solid phase (-).

W cumulative net flow from an outer- to an inner-aquifer region (m^3).

W_I well index per unit length of well bore defined by equation 2.4.1.2 (m^2).

$\underline{x}_m$ position vector of node point m (m).

x distance along the x-coordinate direction (m).

$\underline{x}$ vector of position in the simulation region including the boundary (m).

$\underline{x}_{s(m)}$ location vector of line source s in cell m (m).

y distance along the y-coordinate direction (m).

$\underline{Y}$ Pressure and temperature vector in a well riser.

z distance along the z-coordinate or vertical direction (m).

z_B elevation of the region boundary (m).

z_e elevation of the top of an aquitard layer (m).

z_ℓ elevation of the node in a well at level ℓ (m).

z_{LS} elevation of the land surface (m).

z_m elevation of the node of cell m (m).

z_n coordinate in the outward-normal direction to the boundary (m).

z_R elevation of a river bottom (m).

z_{RFS} elevation of the water surface of a river (m).

z_{wd} elevation of a well datum (m).

9.2 GREEK

α_b bulk compressibility of the porous medium (Pa^{-1}).

α_{be} bulk compressibility of the porous medium in an outer region (Pa^{-1}).

α_L dispersivity longitudinal to the flow direction (m).

α_T dispersivity transverse to the flow direction (m).

α_r rock compressibility (Pa^{-1}).

γ_{Am} apportionment factor for aquifer-influence-function flow rate (-).

γ_R fraction of riverbed area per unit area of aquifer boundary (-).

β_p fluid compressibility (Pa^{-1}).

β_T fluid coefficient of thermal expansion ($°C^{-1}$).

β_w slope of fluid density as a function of mass fraction divided by the reference fluid density (-).

β_{pe} fluid compressibility in an outer region (Pa^{-1}).

β_w' slope of fluid density as a function of scaled-mass fraction divided by the reference fluid density (-).

ε effective porosity (-).

ε_e effective porosity of an outer-aquifer region (-).

λ linear decay constant (s^{-1}).

λ_{min} minimum estimate of spectral radius or maximum eigenvalue (eq. 3.7.2.6a).

λ_{max} maximum estimate of spectral radius or maximum eigenvalue (eq. 3.7.2.6b).

μ viscosity of the fluid (kg/m-s).

μ_e viscosity of the fluid in an outer-aquifer region (kg/m-s).

μ_L viscosity of fluid in an aquitard layer (kg/m-s).

$\mu(T_{ov},w')$ viscosity at the reference temperature (kg/m-s).

μ_R viscosity of the fluid in a riverbed (kg/m-s).

$\bar{\mu}_p^n$ average viscosity at cell face p at time t^n (kg/m-s).

μ_0 viscosity at the minimum mass fraction (kg/m-s).

μ_1 viscosity at the maximum mass fraction (kg/m-s).

μ_r viscosity of the fluid in a riser pipe (kg/m-s).

ν counter index for successive-overrelaxation iteration.

Φ_e potential energy per unit mass of fluid in the aquifer outside a leakage boundary (Nt-m/kg).

ρ fluid density (kg/m^3).

ρ_b bulk density of the porous medium (mass/unit volume of dry porous medium) (kg/m^3).

ρ_B fluid density at the region boundary (kg/m^3).

ρ_e fluid density outside a leakage boundary (kg/m^3).

ρ_o fluid density at a reference pressure, temperature, and mass fraction (kg/m^3).

ρ_r fluid density averaged across a well-riser cross section (kg/m^3).

ρ_R fluid density of the river (kg/m^3).

ρ_s density of the solid phase of the porous matrix (mass/unit volume of solid phase) (kg/m^3).

ρ_{se} density of the solid phase of the porous matrix exterior to the simulation region (mass/unit volume of solid phase) (kg/m^3).

ρ_w average fluid density in a well bore (kg/m^3).

$\bar{\rho}_p^n$ average density at cell face p at time t^n (kg/m^3).

ρ^* density of a fluid source (kg/m^3).

σ spatial weighting coefficient (eq. 3.1.4.4).

τ dimensionless time for heat transfer from a defined well riser by equation 2.4.2.6c (-).

θ factor for time differencing (eq. 3.1.3.4).

θ_r angle between the well riser and the z-coordinate (deg).

θ_w angle between the well bore and the z-coordinate (deg).

ω overrelaxation factor (eq. 3.7.2.2).

ω_{opt} optimum overrelaxation factor (eq. 3.7.2.7).

9.3 MATHEMATICAL OPERATORS AND SPECIAL FUNCTIONS

$\underline{\delta u}$ vector of dependent variables for the finite-difference equations.

δu temporal change in u.

Δu spatial change in u.

δu_{max} maximum temporal change in u.

δu_{max}^{s} specified maximum temporal change in u.

$\bar{u}$ average value of u.

Δr_r wall thickness of a well-riser pipe (m).

δt current time-step length.

δt_o previous time-step length.

∇ del operator; $\hat{e}_x \frac{\partial}{\partial x} + \hat{e}_y \frac{\partial}{\partial y} + \hat{e}_z \frac{\partial}{\partial z}$ in cartesian coordinates;

$\hat{e}_r \frac{\partial}{\partial r} + \hat{e}_\theta \frac{1}{r} \frac{\partial}{\partial \theta} + \hat{e}_z \frac{\partial}{\partial z}$ in cylindrical coordinates.

$|\ |$ absolute value.

$\frac{\partial}{\partial n}$ derivative in the outward-normal direction at a boundary.

δ_{ij} Kronecker delta function; =1 for i=j, = 0 for i≠j.

$\delta(\underline{x}-\underline{x}_s)$ delta function for the point source in a cell at $\underline{x}_s$.

$\underline{\underline{I}}$ identity matrix of rank 3.

J_n Bessel function of the first kind, order n.

Y_n Bessel function of the second kind, order n.

γ Euler's constant $\simeq$ 0.577.

$\overline{\frac{\partial p_b}{\partial t}}$ spatial average of the rate of pressure change at the aquifer-influence-function boundary (Pa/s).

$\overline{\frac{\partial p_e}{\partial t}}$ spatial average of the rate of pressure change in an outer-aquifer region (Pa/s)

$\overline{\frac{\partial p_{b\ell}}{\partial t}}$ average rate of pressure change at the boundary of an inner region (Pa/s)

$\left.\frac{\partial Q_m}{\partial p}\right|_n$ implicit term for fluid line sources at the nth time level (kg/s-Pa).

10. REFERENCES

Akin, J.E., and Stoddart, W.T.C., 1975, Computer codes for implementing the finite element method: University of Tennessee, Department of Engineering Science and Mechanics Report 75-1, 202 p.

American National Standards Institute, 1978, ANSI X3.9-1978 programming language FORTRAN: New York, 85 p.

Anderson, M.P., 1979, Using models to simulate the movement of contaminants through groundwater flow systems: Critical Reviews in Environmental Control, v. 9, no. 2, p. 97-156.

Aronofsky, J.S., 1952, Mobility rates--It's influence on flood patterns during water encroachment: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 195, p. 15-24.

Aziz, Khalid, and Settari, Antonin, 1979, Petroleum reservoir simulation: London, England, Applied Science, 476 p.

Bear, Jacob, 1961, On the tensor form of dispersion: Journal of Geophysical Research, v. 66, no. 4, p. 1185-1197.

_____1972, Dynamics of Fluids in Porous Media: New York, American Elsevier, 764 p.

Bennett, G.D., Kontis, A.L., and Larson, S.P., 1982, Representation of multi-aquifer well effects in three-dimensional groundwater flow simulation: Ground Water, v. 20, no. 3, p. 334-341.

Bird, R.B., Stewart, W.E., and Lightfoot, E.N., 1960, Transport phenomena: New York, John Wiley, 162 p.

Boulton, N.S., 1954, The drawdown of the water-table under non-steady conditions near a pumped well in an unconfined formation: Proceedings of the Institution of Civil Engineers, v. 3, pt. 3, p. 564-579.

Briggs, J.E., and Dixon, T.N., 1968, Some practical considerations in the numerical solution of two-dimensional reservoir problems: Society of Petroleum Engineers Journal, v. 8, no. 6, p. 185-194.

Bulirsch, R., and Stoer, J., 1966, Numerical treatment of ordinary differential equations by extrapolation methods: Numerische Mathematik, v. 8, no. 1, p. 1-13.

Carnahan, Brice, Luther, H.A., and Wilkes, J.O., 1969, Applied numerical methods: New York, John Wiley, 604 p.

Carslaw, H.S., and Jaeger, J.C., 1959, Conduction of heat in solids (2nd ed.): Oxford, England, Clarendon Press, 510 p.

Carter, R.D., and Tracy, G.W., 1960, An improved method for calculating water influx: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 219, p. 415-417.

Chappelear, J.E., and Williamson, A.S., 1981, Representing wells in numerical reservoir simulation; Part 2--Implementation: Society of Petroleum Engineers Journal v. 21, no. 3, p. 339-344.

Clark, S.P., Jr., ed., 1966, Handbook of physical constants: New York, Geological Society of America, Inc., 416 p.

Coats, K.H., George, W.D., and Marcum, B.E., 1974, Three-dimensional simulation of steamflooding: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 257, p. 573-592.

Cooley, R.L., 1974, Finite element solutions for the equations of ground-water flow: Reno, Nevada, Desert Research Institute, Technical Report Series H-W, Hydrology and Water Resources Publication No. 18, 134 p.

Craft, B.C., and Hawkins, M.F., 1959, Applied petroleum reservoir engineering: Englewood Cliffs, New Jersey, Prentice Hall, 917 p.

Crandall, S.H., 1956, Engineering analysis--A survey of numerical procedures: New York, McGraw Hill, 417 p.

Dougherty, E.L., 1963, Mathematical model of an unstable miscible displacement: Society of Petroleum Engineers Journal, v. 3, no. 2, p. 155-163.

Eagleson, P.S., 1970, Dynamic hydrology: New York, McGraw Hill, 462 p.

Fanchi, J.R., 1985, Analytical representation of the Van Everdingen-Hurst aquifer influence functions for reservoir simulation: Society of Petroleum Engineers Journal, v. 25, no. 3, p. 405-406.

Faust, C.R., and Mercer, J.W., 1977, Theoretical analysis of fluid flow and energy transport in hydrothermal systems: U.S. Geological Survey Open-File Report 77-60, 85 p.

Finlayson, B.A., 1972, The method of weighted residuals and variational principles: New York, Academic Press, 412 p.

Freeze, R.A., and Cherry, J.A., 1979, Groundwater: Englewood Cliffs, New Jersey, Prentice Hall, 604 p.

Gear, C.W., 1971, Numerical initial value problems in ordinary differential equations: Englewood Cliffs, New Jersey, Prentice Hall, 253 p.

Grove, D.B., and Stollenwerk, K.G., 1984, Computer model of one-dimensional equilibrium controlled sorption processes: U.S. Geological Survey Water-Resources Investigations Report 84-4059, 58 p.

Hildebrand, F.B., 1956, Introduction to numerical analysis: New York, McGraw Hill, 511 p.

Hougen, O.A., Watson, K.M., and Ragatz, R.A., 1959, Chemical process principles, Part 2--Thermodynamics (2nd ed.): New York, John Wiley and Sons, 567 p.

Hurst, W., 1958, The simplification of the material balance formulas by the Laplace transformation to flow problems in reservoirs: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 213, p. 292-303.

Huyakorn, P.S., and Pinder, G.F., 1983, Computational methods in subsurface flow: New York, Academic Press, 473 p.

INTERA Environmental Consultants, Inc., 1979, Revision of the documentation for a model for calculating effects of liquid waste disposal in deep saline aquifers: U.S. Geological Survey Water-Resources Investigations Report 79-96, 73 p.

INTERCOMP Resource Development and Engineering, Inc., 1976, A model for calculating effects of liquid waste disposal in deep saline aquifers, Part I--Development and Part II--Documentation: U.S. Geological Survey Water-Resources Investigations Report 76-61, 253 p.

Jacob, C.E., and Lohman, S.W., 1952, Non-steady flow to a well of constant drawdown in an extensive aquifer: American Geophysical Union Transactions v. 33, p. 559-569.

Jennings, Alan, 1977, Matrix computation for engineers and scientists, New York, John Wiley, 330 p.

Karamcheti, Krishnamurty, 1967, Vector analysis and cartesian tensors: San Francisco, Holden-Day, 255 p.

Keller, H.B., 1960, The numerical solution of parabolic partial differential equations, *in* Ralston, Anthony, and Wilf, H.S., eds., Mathematical methods for digital computers: New York, John Wiley, v. 1, 293 p.

Keenan, J.H., Keyes, F.G., Hill, P.G., and Moore, J.G., 1969, Steam tables: New York, John Wiley, 162 p.

Kipp, K.L., Jr., 1986, Adaptation of the Carter-Tracy water influx calculation to ground-water flow simulation: Water Resources Research, v. 22, no. 3, p. 423-428.

Konikow, L.F, and Grove, D.B., 1977, Derivation of equations describing solute transport in ground water: U.S. Geological Survey Water-Resources Investigations Report 77-19, 30 p.

Lantz, R.B., 1970, Quantitative evaluation of numerical diffusion (truncation error): Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 251, p. 315-320.

Lohman, S.W., 1972, Groundwater hydraulics: U.S. Geological Survey Professional Paper 708, 70 p.

McAdams, W.H., 1954, Heat transmission (3rd ed.): New York, McGraw Hill, 744 p.

Mercer, J.W., Thomas, S.D., and Ross, Benjamin, 1982, Parameters and variables appearing in repository siting models: Washington, D.C., U.S. Nuclear Regulatory Commission, Division of Waste Management, Office of Nuclear Material Safety and Safeguards, report NUREG/CR-3066, 244 p.

Mitchell, A.R., 1969, Computational methods in partial differential equations: New York, John Wiley, 255 p.

Narasimhan, T.N., and Witherspoon, P.A., 1976, An integrated finite difference method for analyzing fluid flow in porous media: Water Resources Research, v. 12, no. 1, p. 57-64.

Ogata, Akio, and Banks, R.B., 1961, A solution of the differential equation of longitudinal dispersion in porous media: U.S. Geological Survey Professional Paper 411-A, 7 p.

Perry, J.H., Chilton, C.H., and Kirkpatrick, S.D., 1963, Chemical engineer's handbook (4th ed.): New York, McGraw Hill, 1,254 p.

Price, H.S., Varga, R.S., and Warren, J.E., 1966, Application of oscillation matrices to diffusion-convection equations: Journal of Mathematics and Physics, v. 45, no. 3, p. 301-311.

Price, H.S., and Coats, K.H., 1974, Direct methods in reservoir simulation: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 257, p. 295-308.

Prickett, T.A., and Lonnquist, C.G., 1971, Selected digital computer techniques for groundwater resource evaluation: Urbana, State of Illinois, Department of Registration and Education, Illinois State Water Survey ISWS-71-BUL-55, 62 p.

Ramey, H.H., Jr., 1962, Wellbore heat transmission: Journal of Petroleum Technology, v. 14, no. 4, p. 427-435.

Remson, Irwin, Hornberger, G.M., and Moltz, F.J., 1971, Numerical methods in subsurface hydrology: New York, Wiley-Interscience, 389 p.

Ritchie, R.H., and Sakakura, A.Y., 1956, Asymptotic expansions of solutions of the heat conduction equation in internally bounded cylindrical geometry: Journal of Applied Physics, v. 27, no. 12, p. 1453-1459.

Roache, P.J., 1976, Computational fluid dynamics: Albuquerque, New Mexico, Hermosa Publishers, 446 p.

Saffman, P.G., and Taylor, G.I., 1958, The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid: London, England, Proceedings of the Royal Society of London, Series A-Mathematical and Physical Sciences, v. 245, no. 1242, p. 312-329.

Scheidegger, A.E., 1960, Growth of instabilities of displacement fronts in porous media--A discussion of the Muskat-Aronofsky model: Physics of Fluids, v. 3, no. 2, p. 153-162.

_____1961, General theory of dispersion in porous media: Journal of Geophysical Research, v. 66, no. 10, p. 3273-3278.

Scheidegger, A.E., and Johnson, E.F., 1963, The statistical behavior of instabilities in displacement processes in porous media: Canadian Journal of Physics, v. 39, no. 9, p. 1,253-1,263.

Settari, Antonin, and Aziz, Khalid, 1972, Use of irregular grid in reservoir simulation: Society of Petroleum Engineers Journal, v. 12, no. 2, p. 103-114.

_____1974, Use of irregular grid in cylindrical coordinates: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 257, p. 396-412.

Shames, I.H., 1962, Mechanics of fluids: New York, McGraw Hill, 555 p.

Smith, I.M., Siemieniuch, J.L., and Gladwell, I., 1977, Evaluation of Norsett methods for integrating differential equations in time: International Journal for Numerical and Analytical Methods in Geomechanics, v. 1, no. 1, p. 57-74.

Sneddon, I.N., 1951, Fourier transforms: New York, McGraw Hill, 542 p.

Spanier, Jerome, 1967, Alternating direction methods applied to heat conduction problems, *in* Ralston, Anthony, and Wilf, H.S., eds., Mathematical methods for digital computers: New York, John Wiley, v. II, 287 p.

Thomas, G.W., 1982, Principles of hydrocarbon reservoir simulation (2nd ed.): Boston, International Human Resources Development Corp., 208 p.

Tychonov, A.N., and Samarski, A.A., 1964, Partial differential equations of mathematical physics: Translated by S. Radding, San Franscisco, Holden-Day, v. I, 380 p.

Van Everdingen, A.F., and Hurst, William, 1949, The application of the Laplace transformation to flow problems in reservoirs: Transactions of the Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, v. 186, p. 305-324.

Van Poolen, H.K., Breitenback, E.A., and Thurnau, D.H., 1968, Treatment of individual wells and grids in reservoir modelling: Society of Petroleum Engineers Journal, v. 8, no. 4, p. 341-346.

Van Wylen, G.J., 1959, Thermodynamics: New York, John Wiley, 492 p.

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

Voss, C.I., SUTRA, 1984, Saturated-unsaturated transport, A finite-element simulation model for saturated-unsaturated, fluid-density-dependent ground-water flow with energy transport or chemically-reactive single-species solute transport: U.S. Geological Survey Water-Resources Investigations Report 84-4369, 429 p.

Weast, R.C., Selpy, S.M., and Hodgman, C.D., eds., 1964, Handbook of chemistry and physics (45th ed.): Cleveland, Chemical Rubber Co., 1,468 p.

Williamson, A.S., and Chappelear, J.E., 1981, Representing wells in numerical reservoir simulation, Part 1--Theory: Society of Petroleum Engineers Journal, v. 21, no. 3, p. 323-338.

Wooding, R.A., 1959, Stability of a viscous liquid in a vertical tube containing porous material: London, England, Proceedings of the Royal Society of London, Series A - Mathematical and Physical Sciences, v. 252, no. 1268, p. 120-134.

11. SUPPLEMENTAL DATA

Because of the excessive length (about 12,000 lines), no program listing is provided for the HST3D code. However, listings of the major program variables and a cross-reference table are provided to aid the user in error tracing and program modification.

11.1 HST3D PROGRAM VARIABLE LIST WITH DEFINITIONS

The following program-variable list (table 11.1) contains all of the major variable names with brief definitions. Minor variables including loop indices, array subscripts, and temporary results, have not been included. Most temporary variables begin with the letter U with the remainder of the name based on the corresponding major variable. Variably partitioned array pointers begin with the letter I.

Table 11.1.--*Definition list for selected HST3D program variables*

A1HC	- HEAT CONDUCTION B.C. GEOMETRIC FACTOR
A2HC	- HEAT CONDUCTION B.C. GEOMETRIC FACTOR
A3HC	- HEAT CONDUCTION B.C. GEOMETRIC FACTOR
A4	- SUB-MATRIX OF SYSTEM EQUATIONS FOR D4 SOLVER
AAIF	- AQUIFER INFLUENCE FUNCTION B.C. CONSTANT TERM
ABOAR	- POROUS MEDIUM COMPRESSIBILITY FOR OUTER AQUIFER REGION
ABPM	- POROUS MEDIUM COMPRESSIBILITY
ALBC	- LEAKAGE B.C. CONSTANT TERM
ALPHL	- LONGITUDINAL DISPERSIVITY
ALPHT	- TRANSVERSE DISPERSIVITY
AMAX	- UPPER LIMIT OF VARIABLE FOR CONTOUR MAP
AMIN	- LOWER LIMIT OF VARIABLE FOR CONTOUR MAP
ANGOAR	- ANGLE OF INFLUENCE FOR CARTER-TRACY AQUIFER INFLUENCE FUNCTION B.C.
APLOT	- OUTPUT ARRAY FOR CONTOUR MAP
APRNT	- OUTPUT LINE FOR ARRAY PRINTING
ARGRID	- TRUE IF AUTOMATIC RADIAL NODE LOCATIONS TO BE SET
ARRAY	- OUTPUT ARRAY FOR PRINTED TABLES
ARWB	- CROSS-SECTIONAL AREA OF WELL BORE
ARX	- CELL-FACE AREA IN Y-Z PLANE
ARXBC	- CELL-FACE AREA IN Y-Z PLANE FOR BOUNDARY CONDITIONS
ARXFBC	- CELL-FACE AREA IN Y-Z PLANE FOR FLUX B.C
ARY	- CELL-FACE AREA IN X-Z PLANE
ARYBC	- CELL-FACE AREA IN X-Z PLANE FOR BOUNDARY CONDITIONS
ARYFBC	- CELL-FACE AREA IN X-Z PLANE FOR FLUX B.C
ARZ	- CELL-FACE AREA IN X-Y PLANE
ARZBC	- CELL-FACE AREA IN X-Y PLANE FOR BOUNDARY CONDITIONS
ARZFBC	- CELL-FACE AREA IN X-Y PLANE FOR FLUX B.C
AST	- '*'
AUTOTS	- TRUE IF AUTOMATIC TIME STEPPING IS DESIRED
BAIF	- AQUIFER INFLUENCE FUNCTION B.C. TEMPORAL COEFFICIENT
BBLBC	- THICKNESS OF CONFINING LAYER OR RIVER BED
BLANK	- ' '
BLANKL	- '
BLBC	- LEAKAGE B.C. TEMPORAL COEFFICIENT
BOAR	- THICKNESS OF OUTER AQUIFER REGION FOR A.I.F.B.C.
BP	- FLUID COMPRESSIBILITY
BT	- FLUID COEFFICIENT OF THERMAL EXPANSION
BV	- VISCOSITY FUNCTION THERMAL PARAMETER
C	- SOLUTE MASS FRACTION
C00	- SOLUTE CONCENTRATION AT THE END OF A WELL RISER, INITIAL VALUE
C11	- C11 THROUGH C33 ARE COEFFICIENTS IN THE CAPACITANCE MATRIX
C12	
C13	
C21	
C22	

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
C23
C24    - C24,C34, AND C35 ARE COEFFICIENTS OF THE TRANSPORT TERMS
C31
C32
C33
C34
C35
CAIF   - SOLUTE MASS FRACTION IN OUTER AQUIFER REGION
CC24   - CC24,CC34,CC35 ARE CONDUCTANCE COFACTOR ARRAYS
CC34
CC35
CCLBL  - LABEL FOR SOLUTE MASS FRACTION IN A WELL FOR TEMPORAL PLOTS
CCW    - CALCULATED SOLUTE MASS FRACTION IN A WELL
CFLX   - SOLUTE MASS FRACTION FOR INFLOW AT A SPECIFIED FLUX B.C.
CHAPRT - CHARACTER ARRAY FOR PRINTOUT
CHARC  - 'C'
CHARS  - NUMERIC CHARACTERS THAT ILLUSTRATE THE CONTOUR-MAP ZONES
CHKPTD - TRUE IF CHECK POINT DUMPS ARE DESIRED
CI     - NODAL CONNECTION INDEX FOR D4 NUMBERING SCHEME
CIBC   - CHARACTER REPRESENTATION OF IBC
CICALL - CHARACTER REPRESENTATION OF ICALL
CLBC   - SOLUTE MASS FRACTION FOR INFLOW AT LEAKAGE B.C.
CLINE  - CHARACTER LINE FOR CONTOUR MAP
CMX    - SOLUTE CONCENTRATION AT CELL FACE BETWEEN X(I-1) AND X(I)
CMY    - SOLUTE CONCENTRATION AT CELL FACE BETWEEN Y(J-1) AND Y(J)
CMZ    - SOLUTE CONCENTRATION AT CELL FACE BETWEEN Z(K-1) AND Z(K)
CNP    - SOLUTE MASS FRACTION FOR SPECIFIED VALUE B.C.
CNV    - CONVERSION FACTOR
CNVD   - CONVERSION FACTOR FOR DENSITY (ENGLISH TO METRIC)
CNVDFI - CONVERSION FACTOR FOR DIFFUSIVITY (METRIC TO ENGLISH)
CNVDI  - CONVECTOR FOR DENSITY (METRIC TO ENGLISH)
CNVE   - CONVERSION FACTOR FOR ENERGY (ENGLISH TO METRIC)
CNVEPI - CONVERSION FACTOR FOR PRESSURE ENERGY (METRIC TO ENGLISH)
CNVFF  - CONVERSION FACTOR FOR FLUID FLUX (ENGLISH TO METRIC)
CNVGZ  - CONVERSION FACTOR FOR GRAVITY TIMES ELEVATION (ENGLISH TO METRIC)
CNVHC  - CONVERSION FACTOR FOR HEAT CAPACITY (ENGLISH TO METRIC)
CNVHCI - CONVERSION FACTOR FOR HEAT CAPACITY (METRIC TO ENGLISH)
CNVHF  - CONVERSION FACTOR FOR HEAT FLUX (ENGLISH TO METRIC)
CNVHFI - CONVERSION FACTOR FOR HEAT FLUX (METRIC TO ENGLISH)
CNVHI  - CONVERISON FACTOR FOR THERMAL ENERGY (METRIC TO ENGLISH)
CNVHTC - CONVERSION FACTOR FOR HEAT TRANSFER COEFFICIENT (ENGLISH TO METRIC)
CNVHTI - CONVERSION FACTOR FOR HEAT TRANSFER COEFFICIENT (METRIC TO ENGLISH)
CNVL   - CONVERSION FACTOR FOR LENGTH (ENGLISH TO METRIC)
CNVL2  - CONVERSION FACTOR FOR LENGTH SQUARED (ENGLISH TO METRIC)
CNVL2I - CONVERSION FACTOR FOR LENGTH SQUARED (METRIC TO ENGLISH)
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

CNVL3 - CONVERISON FACTOR FOR LENGTH CUBED (ENGLISH TO METRIC)
CNVL3I - CONVERSION FACTOR FOR LENGTH CUBED (METRIC TO ENGLISH)
CNVLI - CONVERSION FACTOR FOR LENGTH (METRIC TO ENGLISH)
CNVMFI - CONVERSION FACTOR FOR MASS FLOW RATE (METRIC TO ENGLISH)
CNVMI - CONVERSION FACTOR FOR MASS (METRIC TO ENGLISH)
CNVP - CONVERSION FACTOR FOR PRESSURE (ENGLISH TO METRIC)
CNVPI - CONVERSION FACTOR FOR PRESSURE (METRIC TO ENGLISH)
CNVSF - CONVERSION FACTOR FOR SOLUTE FLUX (ENGLISH TO METRIC)
CNVT1 - FIRST CONVERSION FACTOR FOR TEMPERATURE (ENGLISH TO METRIC)
CNVT1I - FIRST CONVERSION FACTOR FOR TEMPERATURE (METRIC TO ENGLISH)
CNVT2 - SECOND CONVERSION FACTOR FOR TEMPERATURE (ENGLISH TO METRIC)
CNVT2I - SECOND CONVERSION FACTOR FOR TEMPERATURE (METRIC TO ENGLISH)
CNVTC - CONVERSION FACTOR FOR THERMAL CONDUCTIVITY (ENGLISH TO METRIC)
CNVTCI - CONVERSION FACTOR FOR THERMAL CONDUCTIVITY (METRIC TO ENGLISH)
CNVTM - CONVERSION FACTOR FOR TIME (DAYS TO SECONDS)
CNVTMI - CONVERSION FACTOR FOR TIME (SECONDS TO DAYS)
CNVUUI - CONVERSION FACTOR FOR FLUID CONDUCTANCE (METRIC TO ENGLISH)
CNVVF - CONVERSION FACTOR FOR VOLUMETRIC FLOW RATE (ENGLISH TO METRIC)
CNVVLI - CONVERSION FACTOR FOR VELOCITY (METRIC TO ENGLISH)
CNVVSI - CONVERSION FACTOR FOR VISCOSITY (KG/M-S TO CP)
COLS - NUMBER OF COLUMNS REQUIRED FOR CONTOUR MAPS
COMOPT - TRUE IF COMPUTATION OF AN OPTIMUM OVER-RELAXATION FACTOR IS TO BE DONE
CONLBL - LABEL FOR SOLUTE CONCENTRATION PRINTOUT
CONVC - TRUE IF SOLUTE CALCULATION HAS CONVERGED FOR A GIVEN TIME STEP
CONVP - TRUE IF PRESSURE CALCULATION HAS CONVERGED FOR A GIVEN TIME STEP
CONVRG - TRUE IF WELL-RISER CALCULATION HAS CONVERGED
CONVT - TRUE IF TEMPERATURE CALCULATION HAS CONVERGED FOR A GIVEN TIME STEP
COW - SOLUTE MASS FRACTION OBSERVED VALUES IN A WELL
CPAR - CHARACTER REPRESENTATION OF INPUT PARAMETER
CPARI - INITIAL CHARACTER REPRESENTATION OF INPUT PARAMETER
CPF - HEAT CAPACITY OF FLUID AT MINIMUM SOLUTE CONCENTRATION
CPX - SOLUTE CONCENTRATION AT CELL FACE BETWEEN X(I) AND X(I+1)
CPY - SOLUTE CONCENTRATION AT CELL FACE BETWEEN Y(J) AND Y(J+1)
CPZ - SOLUTE CONCENTRATION AT CELL FACE BETWEEN Z(K) AND Z(K+1)
CSBC - SOLUTE MASS FRACTION FOR INFLOW AT A SPECIFIED PRESSURE B.C.
CVIS - SOLUTE CONCENTRATION DATA FOR CONCENTRATION-VISCOSITY TABLE
CWKT - SOLUTE MASS FRACTION AT WELL DATUM LEVEL
CYLIND - TRUE IF A CYLINDRICAL COORDINATE SYSTEM IS BEING USED
D - DIAGONAL ELEMENTS OF THE SYSTEM EQUATION MATRIX
DAMWRC - DAMPING FACTOR FOR WELL RISER CALCULATION
DASHES - '-----------------------------------'
DBKD - POROUS MEDIUM BULK DENSITY TIMES LINEAR EQUILIBRIUM SORPTION COEFFICIENT
DC - CHANGE IN SOLUTE MASS FRACTION
DCMAX - MAXIMUM CHANGE IN SOLUTE MASS FRACTION
DCTAS - SPECIFIED MAXIMUM CHANGE IN SOLUTE MASS FRACTION FOR AUTOMATIC TIME STEP ADJUSTMENT

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
DDNMAX - MAXIMUM CHANGE IN DENSITY
DDV    - CHANGE IN DEPENDENT VARIABLE FOR FLOW AT SPECIFIED VALUE B.C.
DECLAM - SOLUTE DECAY FACTOR
DEHIR  - CHANGE IN ENTHALPY IN THE REGION
DELTIM - CURRENT TIME STEP
DELX   - NODE SPACING IN X-DIRECTION
DELY   - NODE SPACING IN Y-DIRECTION
DELZ   - NODE SPACING IN Z-DIRECTION
DEN    - FLUID DENSITY
DENO   - FLUID DENSITY AT REFERENCE CONDITIONS
DENC   - COEFFICIENT FOR FLUID DENSITY VARIATION WITH SOLUTE MASS FRACTION
DENCHC - DENSITY CHANGE DUE TO SOLUTE CONCENTRATON CHANGES
DENCHT - DENSITY CHANGE DUE TO TEMPERATURE CHANGES
DENFO  - FLUID DENSITY AT MINIMUM SOLUTE MASS FRACTION
DENF1  - FLUID DENSITY AT MAXIMUM SOLUTE MASS FRACTION
DENFBC - FLUID DENSITY FOR INFLOW AT SPECIFIED FLUX B.C.
DENGL  - CHANGE IN PRESSURE ESTIMATE OVER WELL RISER LENGTH
DENLBC - FLUID DENSITY FOR INFLOW AT LEAKAGE B.C.
DENN   - FLUID DENSITY AT TIME LEVEL N
DENNP  - FLUID DENSITY AT TIME LEVEL N+1
DENOAR - FLUID DENSITY IN OUTER AQUIFER REGION FOR A.I.F.B.C.
DENP   - COEFFICIENT FOR FLUID DENSITY VARIATION WITH PRESSURE
DENT   - COEFFICIENT FOR FLUID DENSITY VARIATION WITH TEMPERATURE
DENW   - FLUID DENSITY IN A WELL BORE
DENWKT - FLUID DENSITY IN A WELL BORE AT THE WELL DATUM
DENWRK - FLUID DENSITY IN A WELL RISER AT LEVEL K
DET    - DETERMINANT OF THE WELL RISER EQUATIONS
DFIR   - CHANGE IN FLUID IN THE REGION
DM     - MOLECULAR DIFFUSIVITY OF THE SOLUTE
DOTS   - '....................'
DP     - CHANGE IN PRESSURE
DPMAX  - MAXIMUM CHANGE IN PRESSURE
DPTAS  - SPECIFIED MAXIMUM CHANGE IN PRESSURE FOR AUTOMATIC TIME STEP ADJUSTMENT
DPUDT  - DERIVATIVE OF THE TRANSIENT AQUIFER INFLUENCE FUNCTION WITH TIME
DPWKT  - CHANGE IN PRESSURE AT A WELL DATUM
DQFDP  - DERIVATIVE OF FLOW RATE WITH PRESSURE
DQHBC  - CHANGE IN HEAT FLOW FOR ANY B.C.
DQHCDT - DERIVATIVE OF HEAT FLOW RATE WITH TEMPERATURE FOR HEAT CONDUCTION B.C.
DQHDP  - DERIVATIVE OF HEAT FLOW WITH PRESSURE
DQHDT  - DERIVATIVE OF HEAT FLOW WITH TEMPERATURE
DQHWDP - DERIVATIVE OF WELL HEAT FLOW WITH PRESSURE
DQHWDT - DERIVATIVE OF WELL HEAT FLOW WITH TEMPERATURE
DQSBC  - CHANGE IN SOLUTE FLOW RATE FOR ANY B.C.
DQSDC  - DERIVATIVE OF SOLUTE FLOW RATE WITH CONCENTRATION
DQSDP  - DERIVATIVE OF SOLUTE FLOW RATE WITH PRESSURE
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

DQSWDC - DERIVATIVE OF WELL SOLUTE FLOW WITH CONCENTRATION
DQSWDP - DERIVATIVE OF WELL SOLUTE FLOW WITH PRESSURE
DQWDP - DERIVATIVE OF WELL FLOW RATE WITH PRESSURE
DQWDPL - DERIVATIVE OF WELL FLOW RATE AT A LAYER WITH PRESSURE
DQWLYR - CHANGE IN WELL FLOW RATE AT A LAYER
DSIR - CHANGE IN SOLUTE IN THE REGION
DT - CHANGE IN TEMPERATURE
DTADZW - DERIVATIVE OF AMBIENT TEMPERATURE AT A WELL RISER WITH DISTANCE IN THE Z-DIRECTION
DTHAWR - THERMAL DIFFUSIVITY OF THE MEDIUM SURROUNDING A WELL RISER
DTHHC - THERMAL DIFFUSIVITY OF THE EXTERNAL MEDIUM AT A HEAT CONDUCTION BOUNDARY
DTIMMN - SPECIFIED MINIMUM TIME STEP FOR AUTOMATIC TIME STEP ADJUSTMENT
DTIMMX - SPECIFIED MAXIMUM TIME STEP FOR AUTOMATIC TIME STEP ADJUSTMENT
DTMAX - MAXIMUM CHANGE IN TEMPERATURE
DTTAS - SPECIFIED MAXIMUM CHANGE IN TEMPERATURE FOR AUTOMATIC TIME STEP ADJUSTMENT
DX - X-COORDINATE SPACING FOR PLOT OF POROUS MEDIA ZONES
DY - Y-COORDINATE SPACING FOR PLOT OF POROUS MEDIA ZONES
DYY - CHANGE IN DEPENDENT VARIABLE VECTOR FOR WELL RISER INTEGRATION
DZ - CHANGE IN Z-COORDINATE VALUE
DZMIN - SPECIFIED MINIMUM CHANGE IN Z-COORDINATE FOR WELL RISER INTEGRATION
EEUNIT - TRUE IF ENGLISH ENGINEERING UNITS ARE BEING USED FOR DATA INPUT AND OUTPUT
EH - FLUID ENTHALPY
EH0 - FLUID ENTHALPY AT REFERENCE CONDITIONS
EH00 - ENTHALPY AT THE END OF A WELL RISER, INITIAL VALUE
EHDT - FLUID ENTHALPY TABLE FOR DEVIATIONS FROM SATURATED CONDITIONS WITH PRESSURE AND TEMPERATURE
EHIR - ENTHALPY IN THE REGION
EHIR0 - INITIAL ENTHALPY IN THE REGION
EHIRN - ENTHALPY IN THE REGION AT TIME LEVEL N
EHMX - ENTHALPY AT CELL FACE BETWEEN X(I-1) AND X(I)
EHMY - ENTHALPY AT CELL FACE BETWEEN Y(J-1) AND Y(J)
EHMZ - ENTHALPY AT CELL FACE BETWEEN Z(K-1) AND Z(K)
EHPX - ENTHALPY AT CELL FACE BETWEEN X(I) AND X(I+1)
EHPY - ENTHALPY AT CELL FACE BETWEEN Y(J) AND Y(J+1)
EHPZ - ENTHALPY AT CELL FACE BETWEEN Z(K) AND Z(K+1)
EHST - FLUID ENTHALPY TABLE AT SATURATION AS A FUNCTION OF TEMPERATURE
EHWEND - ENTHALPY AT THE END OF THE WELL RISER
EHWKT - ENTHALPY IN A WELL AT THE WELL DATUM
EHWSUR - ENTHALPY IN A WELL AT THE LAND SURFACE
EOD - WELL-RISER PIPE ROUGHNESS DIVIDED BY PIPE INSIDE DIAMETER
EPS - TOLERANCE FACTOR FOR TEMPERATURE CALCULATED FROM ENTHALPY AND PRESSURE
EPSFAC - TOLERANCE FACTOR FOR TWO-LINE SUCCESSIVE OVERRELAXATION CALCULATION
EPSOMG - TOLERANCE FACTOR FOR CONVERGENCE OF OPTIMUM OVER-RELAXATION PARAMETER CALCULATION
EPSSOR - TOLERANCE FOR CONVERGENCE OF TWO-LINE SUCCESSIVE-OVER-RELAXATION EQUATION SOLVER
EPSWR - TOLERANCE FOR CONVERGENCE OF WELL RISER CALCULATION
ERREX - TRUE IF PROGRAM ABORT DUE TO ERROR CONDITIONS
ERREXE - TRUE IF PROGRAM ABORT DUE TO EXECUTION ERRORS

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
ERREXI - TRUE IF PROGRAM ABORT DUE TO INPUT ERRORS
EVMAX  - MAXIMUM EIGENVALUE ESTIMATE
EVMIN  - MINIMUM EIGENVALUE ESTIMATE
EX     - 'X'
EXTRAP - EXTRAPOLATED RESULTS OF THE WELL-RISER INTEGRATION
F1AIF  - FACTOR FOR AQUIFER INFLUENCE FUNCTION B.C.
F2AIF  - FACTOR FOR AQUIFER INFLUENCE FUNCTION B.C.
FCJ    - FUNCTION FOR WELL RISER HEAT TRANSFER USING CARSLAW-JAEGER SOLUTION
FDDP   - FRACTIONAL CHANGE IN WELL-DATUM PRESSURE
FDSMTH - FACTOR FOR SPATIAL DIFFERENCING METHOD
FDTMTH - FACTOR FOR TEMPORAL DIFFERENCING METHOD
FFPHL  - FRICTIONAL HEAD LOSS IN WELL RISER
FIR    - FLUID IN THE REGION
FIRO   - INITIAL FLUID IN THE REGION
FIRN   - FLUID IN THE REGION AT TIME LEVEL N
FIRVO  - INITIAL VOLUMETRIC AMOUNT OF FLUID IN THE REGION
FLOREV - TRUE IF FLOW REVERSAL OCCURS AT A GIVEN LEVEL IN A WELL
FLOW   - TRUE IF A WELL IS PRODUCING OR INJECTING AT AT LEAST ONE LEVEL
FMT    - CHARACTER STRING FOR PRINT FORMAT
FPR3   - 4*PI*R**3
FRAC   - FRACTION OF CELL THAT IS FILLED FOR UNCONFINED FLOW
FRACN  - FRACTION OF CELL FILLED AT TIME LEVEL N
FRACNP - FRACTION OF CELL FILLED AT TIME LEVEL N+1
FRESUR - TRUE IF A FREE SURFACE BOUNDARY IS ALLOWED
FRFAC  - FRICTION FACTOR FOR FLOW IN A WELL RISER PIPE
FRFLM  - FRICTION FACTOR IN WELL BORE AT BOUNDARY BETWEEN LEVEL L-1 AND L
FRFLP  - FRICTION FACTOR IN WELL BORE AT BOUNDARY BETWEEN LEVEL L AND L+1
FS     - FUNCTION TABLE FOR INTERPOLATION
FSCON  - TRUE IF A FREE SURFACE RISES ABOVE THE TOP OF A CELL AND CONVERTS TO CONFINED CONDITIONS
FSLOW  - TRUE IF FREE SURFACE FALLS BELOW BOTTOM OF CELL
FTDAIF - FACTOR FOR DIMENSIONLESS TIME FOR A.I.F.B.C.
GCOSTH - COMPONENT OF GRAVITY ALONG A WELL RISER
GRAV   - GRAVITATIONAL CONSTANT
GX     - COMPONENT OF GRAVITY IN THE X-DIRECTION
GY     - COMPONENT OF GRAVITY IN THE Y-DIRECTION
GZ     - COMPONENT OF GRAVITY IN THE Z-DIRECTION
HDPRNT - POTENTIOMETRIC HEAD FOR PRINTOUT
HEAT   - TRUE IF HEAT TRANSPORT IS BEING SIMULATED
HTCU   - OVERALL HEAT TRANSFER COEFFICIENT FOR WELL RISER CALCULATON
HTCWR  - HEAT TRANSFER COEFFICIENT FROM THE FLUID TO A WELL RISER PIPE
HWT    - ELEVATION OF THE WATER TABLE
    THE UNDEFINED 'I' VARIABLES ARE POINTERS FOR THE LARGE VARIABLY PARTITIONED ARRAYS
I12X   - '12X' FORMAT
I1Z    - ZONE INDEX
I2Z    - ZONE INDEX
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
IBC    - INDEX OF BOUNDARY CONDITION TYPE FOR A CELL
IBCMAP - INDEX OF BOUNDARY CONDITION TYPE FOR CONTOUR MAP
ICALL  - INDEX OF CALL TO ARRAY-INPUT ROUTINE
ICC    - TRUE IF INITIAL CONDITION SOLUTE MASS FRACTION VALUES TO BE READ
ICHWT  - TRUE IF INITIAL CONDITION WATER-TABLE ELEVATION TO BE READ
ICHYDP - TRUE IF INITIAL CONDITION OF HYDROSTATIC PRESSURE DISTRIBUTION
ICMAX  - INDEX IN X-DIRECTION OF CELL WITH MAXIMUM CHANGE IN SOLUTE CONCENTRATION
ICON   - INDEX OF CONNECTED NODES
ICT    - TRUE IF INITIAL CONDITION TEMPERATURE DISTRIBUTION TO BE READ
ICXM   - INDEX FOR VA ARRAY OF CONNECTING NODE IN NEGATIVE X-DIRECTION
ICXP   - INDEX FOR VA ARRAY OF CONNECTING NODE IN POSITIVE X-DIRECTION
ICYM   - INDEX FOR VA ARRAY OF CONNECTING NODE IN NEGATIVE Y-DIRECTION
ICYP   - INDEX FOR VA ARRAY OF CONNECTING NODE IN POSITIVE Y-DIRECTION
ICZM   - INDEX FOR VA ARRAY OF CONNECTING NODE IN NEGATIVE Z-DIRECTION
ICZP   - INDEX FOR VA ARRAY OF CONNECTING NODE IN POSITIVE Z-DIRECTION
ID4    - INDEX FOR D4 NODE RENUMBERING
ID4CON - INDEX OF CONNECTED NODES FOR D4 NODE RENUMBERING
ID4NO  - INDEX OF NEW NODE NUMBERS FOR D4 RENUMBERING SCHEME
IDIAG  - INDEX OF DIAGONAL ELEMENTS IN THE A4 MATRIX
IDIR   - INDEX OF DIRECTION FOR TWO-LINE-SOR SOLVER
IDNUM  - IDENTIFICATION NUMBER FOR TEMPORAL PLOT
IDRCON - INDEX OF DIRECTION TO NEIGHBORING CONNECTED NODE
IE     - INDEX OF ERROR NUMBER
IE1    - INDEX LIMIT OF ERROR NUMBER
IE2    - INDEX LIMIT OF ERROR NUMBER
IENDIV - INDEX OF LAST LOCATION IN THE IVPA ARRAY
IENDVV - INDEX OF LAST LOCATION IN THE VPA ARRAY
IEQ    - INDEX OF EQUATION BEING SOLVED
IER    - INDEX OF ERROR NUMBER
IERR   - LOGICAL FLAGS FOR ERROR MESSAGES
IF12   - 'F12' FORMAT
IFACE  - INDEX OF FACE OF APPLIED B.C.
IFMT   - INDEX OF FORMAT TYPE
IFORM  - CHARACTER STRING CONTAINING FORMAT STATEMENT
IG12   - 'G12' FORMAT
II12   - 'I12' FORMAT
II5    - 'I5' FORMAT
ILAVEL - TOTAL LENGTH OF ALL LABELED COMMON BLOCKS
ILBL   - INDEX OF LABEL CHARACTERS FOR TEMPORAL PLOT
ILH    - INDEX OF ELEMENT IN EQUATION MATRIX FOR D4 SOLVER
ILIVPA - LENGTH OF IVPA ARRAY AS DIMENSIONED
ILNXI  - (LOG(XI))**-1
ILVPA  - LENGTH OF VPA ARRAY AS DIMENSIONED
IMAP1  - IMAP1, IMAP2, JMAP1, JMAP2, KMAP1, KMAP2 ARE NODE RANGES FOR MAP OUTPUT
IMAP2
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
IMOD   - INDEX OF MODIFICATION FOR ARRAY INPUT
IMPPTC - INDEX FOR CONTOUR MAP OF DEPENDENT VARIABLE ARRAYS
IMPQW  - TRUE IF SEMI-IMPLICIT WELL FLOW CALCULATION IS DESIRED
INZONE - TRUE IF A CELL IS CONTAINED WITHIN A DEFINED POROUS MEDIA ZONE
IPAGE  - PAGE NUMBER FOR CONTOUR MAP
IPAR   - INTEGER PARAMETER VALUE FOR ARRAY INPUT
IPARI  - INITIAL INTEGER PARAMETER VALUE FOR ARRAY INPUT
IPLOT  - INDEX ARRAY IDENTIFYING EXCLUDED CELLS FOR CONTOUR MAPS
IPMAX  - INDEX IN X-DIRECTION OF CELL WITH MAXIMUM CHANGE IN PRESSURE
IPMZ   - INDEX OF POROUS MEDIUM ZONE
IPMZ1  - INDEX OF FIRST ZONE FOR ZONE PLOT
IPMZ2  - INDEX OF LAST ZONE FOR ZONE PLOT
IPRPTC - INDEX OF PRINTING PRESSURE, TEMPERATURE, OR SOLUTE CONCENTRAION ARRAYS
IROW   - INDEX OF EQUATION NUMBER FOR D4 SOLVER
ISIGN  - INDEX OF DIRECTION FOR THE WELL RISER CALCULATION. POSITIVE IS UPWARDS.
ISORD  - INDEX OF OPTIMUM DIRECTION FOR TWO-LINE SOR SOLVER
ISUM   - SUM OF I,J, AND K INDICES FORMING A DIAGONAL PLANE
ITIME  - TIME STEP NUMBER
ITMAX  - INDEX IN X-DIRECTION OF CELL WITH MAXIMUM CHANGE IN TEMPERATURE
ITNO   - ITERATION NUMBER IN TWO-LINE SOR SOLVER
ITNOC  - ITERATIONS USED FOR SOLUTE CALCULATION BY TWO-LINE SOR SOLVER
ITNOP  - ITERATIONS USED FOR PRESSURE CALCULATION BY TWO-LINE SOR SOLVER
ITNOT  - ITERATIONS USED FOR TEMPERATURE CALCULATION BY TWO-LINE-SOR SOLVER
ITRN   - ITERATION COUNT FOR SEQUENCE OF P,T AND C EQUATION SOLUTION CYCLES
ITRN1  - ITERATION COUNT FOR WELLBORE CALCULATION
ITRN2  - ITERATION COUNT FOR WELLBORE CALCULATION
ITRNDN - ITERATION COUNT FOR WELLBORE DENSITY CALCULATION
ITRNI  - ITERATION COUNT FOR P,T OR P,C SOLUTION CYCLE
ITRNP  - ITERATION COUNT FOR P AND WELL BORE SOLUTION CYCLE
IUH    - INDEX OF EQUATION IN UPPER HALF OF MATRIX FOR D4 SOLVER
IVPA   - INTEGER VARIABLY PARTITIONED ARRAY
IW     - INDEX OF CELL NUMBER IN X-DIRECTION FOR A WELL
IWEL   - INDEX OF WELL NUMBER
J1Z    - ZONE INDEX
J2Z    - ZONE INDEX
JCMAX  - INDEX IN Y-DIRECTION OF CELL WITH MAXIMUM CHANGE IN SOLUTE CONCENTRATION
JMAP1
JMAP2
JPMAX  - INDEX IN Y-DIRECTION OF CELL WITH MAXIMUM CHANGE IN PRESSURE
JTMAX  - INDEX IN Y-DIRECTION OF CELL WITH MAXIMUM CHANGE IN TEMPERATURE
JW     - INDEX OF CELL NUMBER IN Y-DIRECTION FOR A WELL
K1Z    - ZONE INDEX
K2Z    - ZONE INDEX
KARHC  - THERMAL CONDUCTANCE FACTOR FOR HEAT CONDUCTION B.C.
KCMAX  - INDEX IN Z-DIRECTION OF CELL WITH MAXIMUM CHANGE IN SOLUTE CONCENTRATION
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
KLBC   - CONDUCTANCE FACTOR FOR LEAKAGE B.C.
KMAP1
KMAP2
KO     - WELL NUMBER INDEX FOR OBSERVED DATA
KOAR   - PERMEABILITY OF OUTER AQUIFER REGION FOR A.I.F.B.C.
KPMAX  - INDEX IN Z-DIRECTION OF CELL WITH MAXIMUM CHANGE IN PRESSURE
KTHAWR - THERMAL CONDUCTIVITY OF AMBIENT MEDIUM AT WELL RISER
KTHF   - THERMAL CONDUCTIVITY OF FLUID
KTHWR  - THERMAL CONDUCTIVITY OF WELL RISER PIPE
KTHX   - THERMAL CONDUCTIVITY OF FLUID AND POROUS MEDIUM IN X-DIRECTION
KTHXPM - THERMAL CONDUCTIVITY OF POROUS MEDIUM IN X-DIRECTION
KTHY   - THERMAL CONDUCTIVITY OF FLUID AND POROUS MEDIUM IN Y-DIRECTION
KTHYPM - THERMAL CONDUCTIVITY OF POROUS MEDIUM IN Y-DIRECTION
KTHZ   - THERMAL CONDUCTIVITY OF FLUID AND POROUS MEDIUM IN Z-DIRECTION
KTHZPM - THERMAL CONDUCTIVITY OF POROUS MEDIUM IN Z-DIRECTION
KTMAX  - INDEX IN Z-DIRECTION OF CELL WITH MAXIMUM CHANGE IN TEMPERATURE
KXX    - PERMEABILITY IN X-DIRECTION
KYY    - PERMEABILITY IN Y-DIRECTION
KZZ    - PERMEABILITY IN Z-DIRECTION
LABEL  - CHARACTER STRING FOR LABEL
LBLDIR - LABEL FOR OPTIMUM DIRECTION FOR TWO-LINE SOR SOLVER
LBLEQ  - LABEL FOR SYSTEM EQUATION NUMBER
LBW    - LEFT SIDE BAND WIDTH OF A4 ARRAY FOR EACH EQUATION
LCBOTW - INDEX OF BOTTOM LAYER OF COMPLETION OF A WELL
LCROSD - TRUE IF THE CROSS-DERIVATIVE DISPERSION COEFFICIENTS ARE TO BE LUMPED INTO THE DIAGONAL TERMS
LCTOPW - INDEX OF TOP LAYER OF COMPLETION OF A WELL
LDASH  - '----------------------------------------'
LDOTS  - '.....................................................'
LENAX  - LENGTH OF THE X-AXIS FOR A CONTOUR MAP
LENAY  - LENGTH OF THE Y-AXIS FOR A CONTOUR MAP
LENAZ  - LENGTH OF THE Z-AXIS FOR A CONTOUR MAP
LGREN  - LOGARITHM OF THE REYNOLDS NUMBER
LIMAGE - LINE IMAGE FOR DATA INPUT
LIMIT  - LABEL FOR WELL-BORE CALCULATION CONSTRAINTS
LINE   - CHARACTER STRING FOR TEMPORAL PLOT
LINLIM - LIMIT TO THE NUMBER OF PRINTER LINES FOR TEMPORAL PLOT
LOCAIF - INDEX FOR LOCATION OF THE AQUIFER INFLUENCE FUNCTION B.C.
LPRNT  - INDEX FOR INCLUSION OF A NODE IN AN ARRAY PRINTOUT
LTD    - LOGARITHM OF DIMENSIONLESS TIME FOR A.I.F.
M      - CELL NUMBER
MA     - CELL NUMBER FOR ASSEMBLY
MAIFC  - CELL NUMBERS FOR AQUIFER INFLUENCE FUNCTION B.C.
MAPPTC - INDEX FOR OUTPUT OF ZONED CONTOUR MAPS OF PRESSURE, TEMPERATURE, OR SOLUTE CONCENTRATION
MAXIT  - MAXIMUM NUMBER OF ITERATONS FOR TWO-LINE SOR SOLVER
MAXIT1 - MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR CALCULATION OF OPTIMUM OVER-RELAXATION FACTOR
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

MAXIT2 - MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR TWO-LINE SOR SOLUTION
MAXITN - MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR THE SOLUTION CYCLE OF THE THREE SYSTEM EQUATIONS
MAXORD - MAXIMUM ORDER OF POLYNOMIAL ALLOWED FOR INTEGRATION OF THE WELL-RISER EQUATIONS
MAXPTS - MAXIMUM NUMBER OF SPATIAL STEPS ALLOWED FOR WELL-RISER CALCULATION
METH - INDEX FOR EXTRAPOLATION METHOD SELECTION FOR WELL-RISER CALCULATION
MFBC - CELL NUMBERS FOR SPECIFIED FLUX B.C.
MFLBL - LABEL FOR MASS FRACTION OR SCALED MASS FRACTION
MHCBC - CELL NUMBERS FOR HEAT CONDUCTION B.C.
MIJKM - CELL NUMBER AT I,J,K-1
MIJKP - CELL NUMBER AT I,J,K+1
MIJMK - CELL NUMBER AT I,J-1,K
MIJMKM - CELL NUMBER AT I,J-1,K-1
MIJMKP - CELL NUMBER AT I,J-1,K+1
MIJPK - CELL NUMBER AT I,J+1,K
MIJPKM - CELL NUMBER AT I,J+1,K-1
MIJPKP - CELL NUMBER AT I,J+1,K+1
MIMJK - CELL NUMBER AT I-1,J,K
MIMJKM - CELL NUMBER AT I-1,J,K-1
MIMJKP - CELL NUMBER AT I-1,J,K+1
MIMJMK - CELL NUMBER AT I-1,J-1,K
MIMJPK - CELL NUMBER AT I-1,J+1,K
MINUS - '-'
MIPJK - CELL NUMBER AT I+1,J,K
MIPJKM - CELL NUMBER AT I+1,J,K-1
MIPJKP - CELL NUMBER AT I+1,J,K+1
MIPJMK - CELL NUMBER AT I+1,J-1,K
MIPJPK - CELL NUMBER AT I+1,J+1,K
MKT - CELL NUMBER AT A WELL-DATUM LEVEL
MLBC - CELL NUMBERS FOR LEAKAGE B.C.
MOBW - MOBILITY FOR AN AQUIFER LAYER AT A WELL BORE
MSBC - CELL NUMBERS FOR SPECIFIED VALUE B.C.
MT - CELL NUMBER AT TOP OF REGION
MTJP1 - CELL NUMBER AT TOP OF REGION AT I,J+1
MXITQW - MAXIMUM NUMBER OF ITERATIONS ALLOWED FOR WELL FLOW RATE CALCULATION
NAC - COUNT OF ACTIVE CELLS
NAIFC - NUMBER OF AQUIFER INFLUENCE FUNCTION B.C. CELLS
NCHARS - NUMBER OF CHARACTERS
NCPR - NUMBER OF CHARACTERS TO BE PRINTED FOR ZONE PLOT
NDIM - NUMBER OF DIMENSIONS FOR TABULAR INTERPOLATION
NEHST - NUMBER OF ENTRIES IN THE SATURATED ENTHALPY VS. TEMPERATURE TABLE
NFBC - NUMBER OF SPECIFIED FLUX B.C. CELLS
NFC - COUNT OF SPECIFIED FLUX CELLS
NGRIDX - COLUMN NUMBER CORRESPONDING TO X-GRID POINT
NGRIDY - LINE NUMBER CORRESPONDING TO Y-GRID POINT
NHC - COUNT OF HEAT CONDUCTION BOUNDARY CELLS

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

NHCBC - NUMBER OF HEAT CONDUCTION B.C. CELLS
NHCN - NUMBER OF NODES EXTERNAL TO THE REGION FOR THE HEAT CONDUCTION B.C. CALCULATION
NLBC - NUMBER OF LEAKAGE B.C. CELLS
NLC - COUNT OF LEAKAGE B.C. CELLS
NMAPR - NUMBER OF MAP RECORDS WRITTEN TO DISC
NMPZON - NUMBER OF ZONES FOR CONTOUR MAPS
NNOPPR - NUMBER OF NODES TO BE PRINTED FOR ARRAY OUTPUT
NNPR - NUMBER OF NODES TO HAVE PRINTED VALUES
NO - NUMBER OF OBSERVED DATA POINTS
NOCV - NUMBER OF DATA POINTS FOR VISCOSITY VS. SOLUTE CONCENTRATION
NOTV - NUMBER OF DATA POINTS FOR VISCOSITY VS. TEMPERATURE
NOTV0 - NUMBER OF DATA POINTS FOR VISCOSITY VS. TEMPERATURE AT MINIMUM SOLUTE CONCENTRATION
NOTV1 - NUMBER OF DATA POINTS FOR VISCOSITY VS. TEMPERATURE AT MAXIMUM SOLUTE CONCENTRATION
NPAGES - NUMBER OF PAGES FOR A CONTOUR MAP
NPEHDT - NUMBER OF POINTS IN THE ENTHALPY DEVIATION TABLE ALONG THE PRESSURE COORDINATE
NPMZ - NUMBER OF ZONES OF POROUS MEDIA PROPERTIES
NPOSNS - NUMBER OF CHARACTER POSITIONS IN THE X-DIRECTION FOR A CONTOUR MAP ELEMENT
NPTAIF - NUMBER OF POINTS IN THE TABLE OF TRANSIENT AQUIFER INFLUENCE FUNCTION VS. TIME
NPTCBC - NUMBER OF SPECIFIED VALUE B.C. CELLS
NPTSA4 - NUMBER OF POINTS IN THE A4 MATRIX FOR THE D4 SOLVER
NPTSD4 - NUMBER OF POINTS FOR THE D4 SOLVER
NPTSUH - NUMBER OF POINTS IN THE UPPER HALF OF THE EQUATION MATRIX FOR THE D4 SOLVER
NR - NUMBER OF NODES IN THE R-DIRECTION
NRSTTP - NUMBER OF RESTART RECORD TIME PLANES WRITTEN TO DISC
NSC - NUMBER OF SPECIFIED VALUE B.C. CELLS
NSHUT - NUMBER OF WELLS SHUT IN
NSTD4 - NUMBER OF STORAGE LOCATIONS REQUIRED BY THE D4 SOLVER
NSTSOR - NUMBER OF STORAGE LOCATIONS REQUIRED BY THE TWO-LINE SOR SOLVER
NTEHDT - NUMBER OF POINTS ALONG THE TEMPERATURE COORDINATE IN THE ENTHALPY DEVIATION TABLE
NTHPTC - INDEX FOR SPECIFYING THAT EVERY NTH CALCULATED P, T OR C POINT IS TO BE PLOTTED
NTHPTO - INDEX FOR SPECIFYING THAT EVERY NTH OBSERVED P, T OR C POINT IS TO BE PLOTTED
NTSCHK - NUMBER OF TIME STEPS BETWEEN CHECKPOINT DUMPS
NTSOPT - NUMBER OF TIME STEPS BETWEEN RECALCULATION OF THE OPTIMUM OVER-RELAXATION PARAMETER
NVST - NUMBER OF POINTS IN THE TABLE OF GENERALIZED VISCOSITY VS. TEMPERATURE
NWEL - NUMBER OF WELLS IN THE REGION
NX - NUMBER OF NODES IN THE X-DIRECTION
NXPR - NUMBER OF VALUES IN X-DIRECTION FOR ARRAY PRINTOUT
NXY - NUMBER OF NODES IN AN X-Y PLANE
NXYZ - NUMBER OF NODES IN THE SIMULATION REGION, NX*NY*NZ
NY - NUMBER OF NODES IN THE Y-DIRECTION
NYPR - NUMBER OF VALUES IN Y-DIRECTION FOR ARRAY PRINTOUT
NZ - NUMBER OF NODES IN THE Z-DIRECTION
NZPR - NUMBER OF VALUES IN Z-DIRECTION FOR ARRAY PRINTOUT
NZTPHC - NUMBER OF NODES FOR THE INITIAL CONDITION TEMPERATURE PROFILE OUTSIDE THE REGION FOR HEAT CONDUCTION B.C.
NZTPRO - NUMBER OF NODES FOR THE INITIAL CONDITION TEMPERATURE PROFILE IN THE Z-DIRECTION

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

OCPLOT - TRUE IF PLOTS OF OBSERVED AND CALCULATED VALUES OF THE DEPENDENT VARIABLES ARE DESIRED
OH - '0'
OMEGA - OVER-RELAXATION FACTOR FOR TWO-LINE SOR SOLVER
OMOPT - OPTIMUM OVER-RELAXATION FACTOR FOR TWO-LINE SOR SOLVER
ORENPR - ORIENTATION OF THE ARRAY PRINTOUTS, AREAL OR VERTICAL SLICE
ORFR - ORIENTATION OF THE ARRAY PRINTOUTS FOR SATURATED FRACTION OF CELL
P - PRESSURE
P0 - REFERENCE PRESSURE FOR DENSITY
P00 - PRESSURE AT THE END OF A WELL RISER, INITIAL VALUE
P0H - REFERENCE PRESSURE FOR ENTHALPY
P1 - '.1' FORMAT
P2 - '.2' FORMAT
P4 - '.4' FORMAT
P5 - '.5' FORMAT
PAATM - ATMOSPHERIC PRESSURE IN ABSOLUTE UNITS
PAEHDT - PRESSURE VALUES (ABSOLUTE) FOR ENTHALPY DEVIATION TABLE
PAIF - PRESSURE OUTSIDE THE REGION FOR A.I.F.B.C.
PAR - PARAMETER VALUE FOR ARRAY INPUT
PARA - PARAMETER VALUE FOR ARRAY INPUT
PARB - PARAMETER VALUE FOR ARRAY INPUT
PARC - PARAMETER VALUE FOR ARRAY INPUT
PCS - PRESSURE CALCULATED IN A WELL AT LAND SURFACE FOR TEMPORAL PLOT
PCW - PRESSURE CALCULATED AT A WELL DATUM FOR TEMPORAL PLOT
PFSLOW - TRUE IF A MESSAGE IS TO BE PRINTED WHEN THE FREE SURFACE FALLS BELOW A CELL BOUNDARY
PHILBC - POTENTIOMETRIC HEAD ON THE OTHER SIDE OF AN AQUITARD FOR A LEAKAGE B.C.
PI - '3.14159...'
PINIT - PRESSURE INITIAL CONDITION AT A GIVEN ELEVATION
PLBL - LABEL FOR PRESSURE ARRAY PRINTOUTS
PLOTWC - TRUE IF TEMPORAL PLOTS OF SOLUTE CONCENTRATIONS IN WELLS ARE DESIRED
PLOTWP - TRUE IF TEMPORAL PLOTS OF PRESSURES IN WELLS ARE DESIRED
PLOTWT - TRUE IF TEMPORAL PLOTS OF TEMPERATURES IN WELLS ARE DESIRED
PLTZON - TRUE IF A PLOT OF THE POROUS MEDIA ZONES IS DESIRED
PLUS - '+'
PMCHV - POROUS MEDIUM COMPRESSIBILITY COEFFICIENT IN HEAT EQUATION
PMCV - POROUS MEDIUM COMPRESSIBILITY COEFFICIENT IN FLOW EQUATION
PMHV - POROUS MEDIUM THERMAL COEFFICIENT IN HEAT EQUATION, CURRENT VALUE
PNP - PRESSURE FOR SPECIFIED VALUE B.C.
POROAR - POROSITY OF THE OUTER AQUIFER REGION FOR A.I.F.B.C.
POROS - POROSITY
POS - PRESSURE OBSERVED IN A WELL AT THE LAND SURFACE FOR TEMPORAL PLOT
POSUP - TRUE IF THE VERTICAL AXIS OF THE CONTOUR MAP IS POSITIVE UPWARD
POW - PRESSURES OBSERVED IN A WELL AT THE WELL DATUM FOR TEMPORAL PLOT
PRBCF - TRUE IF PRINTOUT OF BOUNDARY CONDITION FLOW RATES IS DESIRED
PRDV - TRUE IF PRINTOUTS OF DENSITY AND VISCOSITY FIELDS ARE DESIRED
PRGFB - TRUE IF PRINTOUT OF GLOBAL FLOW BALANCE IS DESIRED

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

PRIBCF - PRINTOUT INTERVAL FOR BOUNDARY CONDITION FLOW RATES
PRIDV - PRINTOUT INTERVAL FOR DENSITY AND VISCOSITY FIELDS
PRIGFB - PRINTOUT INTERVAL FOR GLOBAL FLOW BALANCE
PRIKD - PRINTOUT INTERVAL FOR CONDUCTANCES AND DISPERSION COEFFICIENTS
PRIMAP - PRINTOUT INTERVAL FOR ZONED CONTOUR MAPS
PRIPTC - PRINTOUT INTERVAL FOR PRESSURE, TEMPERATURE, AND SOLUTE MASS FRACTION FIELDS
PRISLM - PRINTOUT INTERVAL FOR SOLUTION METHOD INFORMATION
PRIVEL - PRINTOUT INTERVAL FOR VELOCITY FIELD
PRIWEL - PRINTOUT INTERVAL FOR WELL INFORMATION
PRKD - TRUE IF PRINTOUTS OF CONDUCTANCES AND DISPERSION COEFFICIENTS ARE DESIRED
PRPTC - TRUE IF PRINTOUTS OF DEPENDENT VARIABLES P,T OR C ARE DESIRED
PRSLM - TRUE IF PRINTOUT OF SOLUTION METHOD INFORMATION IS DESIRED
PRTBC - TRUE IF PRINTOUT OF BOUNDARY CONDITION FLOW RATES IS DESIRED
PRTCCM - TRUE IF PRINTOUT OF MESSAGE OF FREE-SURFACE B.C. BECOMING CONFINED IS DESIRED
PRTCHR - CHARACTER STRING FOR PRINTOUT OF CONTOUR MAPS
PRTDV - TRUE IF PRINTOUT OF DENSITY AND VISCOSITY FIELDS IS DESIRED
PRTFP - TRUE IF PRINTOUT OF FLUID PARAMETERS IS DESIRED
PRTIC - TRUE IF PRINTOUT OF INITIAL CONDITIONS IS DESIRED
PRTMPD - TRUE IF PRINTOUT OF CONTOUR MAP PARAMETERS IS DESIRED
PRTPMP - TRUE IF PRINTOUT OF POROUS MEDIA PROPERTIES IS DESIRED
PRTRE - TRUE IF PRINTOUT OF READ ECHO FOR ARRAY INPUT IS DESIRED
PRTSLM - TRUE IF PRINTOUT OF SOLUTION METHOD PARAMETERS IS DESIRED
PRTWEL - TRUE IF PRINTOUT OF WELL BORE INFORMATION IS DESIRED
PRVEL - TRUE IF PRINTOUT OF VELOCITY FIELD IS DESIRED
PRWEL - TRUE OF PRINTOUT OF WELL BORE INFORMATION IS DESIRED
PSBC - PRESSURE FOR SPECIFIED VALUE B.C.
PSLBL - LABEL FOR WELL PRESSURE AT LAND SURFACE FOR TEMPORAL PLOTS
PSMAX - MAXIMUM PRESSURE AT LAND SURFACE FOR SCALING TEMPORAL PLOTS
PSMIN - MINIMUM PRESSURE AT LAND SURFACE FOR SCALING TEMPORAL PLOTS
PU - TRANSIENT AQUIFER INFLUENCE FUNCTION DIMENSIONLESS PRESSURE RESPONSE TO A UNIT DIMENSIONLESS WITHDRAWAL FLOW-RATE CHANGE
PV - PORE VOLUME, CURRENT VALUE
PVK - PORE VOLUME WEIGHTED SORPTION PARAMETER, CURRENT VALUE
PWCELL - PRESSURE IN THE CELL AT THE UPPERMOST COMPLETION LEVEL OF A WELL
PWKT - PRESSURE IN A WELL AT THE WELL DATUM
PWKTS - SPECIFIED LIMITING PRESSURE IN A WELL AT THE WELL DATUM
PWLBL - LABEL FOR WELL PRESSURE AT WELL DATUM FOR TEMPORAL PLOTS
PWMAX - MAXIMUM PRESSURE AT WELL DATUM FOR SCALING TEMPORAL PLOTS
PWMIN - MINIMUM PRESSURE AT WELL DATUM FOR SCALING TEMPORAL PLOT
PWREND - PRESSURE AT THE END OF A WELL RISER PIPE
PWRK - PRESSURE IN A WELL RISER AT A GIVEN LEVEL
PWSUR - PRESSURE IN A WELL AT THE LAND SURFACE
PWSURS - SPECIFIED LIMITING PRESSURE IN A WELL AT THE LAND SURFACE
QDVSBC - FLOW RATE FOR THE DEPENDENT VARIABLE AT A SECIFIED VALUE B.C. CELL
QFAC - BI-LINEAR INTERPOLATION FACTOR FOR CONTOUR MAPS
QFAIF - VOLUMETRIC FLOW RATE FOR A.I.F.B.C.

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

QFBC - FLUID FLOW RATE AT ANY B.C. CELL
QFBCV - VOLUMETRIC FLUID FLOW RATE FOR SPECIFIED FLUX B.C.
QFFX - FLUID FLUX IN X-DIRECTION
QFFY - FLUID FLUX IN Y-DIRECTION
QFFZ - FLUID FLUX IN Z-DIRECTION
QFLBC - FLUID MASS FLOW RATE FOR LEAKAGE B.C.
QFSBC - VOLUMETRIC FLUID FLOW RATE FOR SPECIFIED VALUE B.C.
QHAIF - HEAT FLOW RATE FOR A.I.F.B.C.
QHBC - HEAT FLOW RATE AT ANY B.C. CELL
QHCBC - HEAT FLOW RATE FOR HEAT CONDUCTION B.C.
QHFAC - HEAT FLOW RATE FACTOR FOR WELL RISER
QHFBC - HEAT FLOW RATE FOR SPECIFIED FLUX B.C.
QHFX - SPECIFIED HEAT FLUX IN THE X-DIRECTION
QHFY - SPECIFIED HEAT FLUX IN THE Y-DIRECTION
QHFZ - SPECIFIED HEAT FLUX IN THE Z-DIRECTION
QHLBC - HEAT FLOW RATE AT LEAKAGE B.C.
QHLYR - HEAT FLOW RATE FROM A WELL TO AN AQUIFER LAYER
QHSBC - HEAT FLOW RATE AT A SPECIFIED VALUE B.C.
QHSBC - HEAT FLOW RATE AT A SPECIFIED PRESSURE B.C.
QHW - HEAT FLOW RATE FROM A WELL TO THE AQUIFER
QHWRK - HEAT FLOW RATE IN A WELL RISER AT A GIVEN LEVEL
QLIM - LIMITING FLOW RATE FOR RIVER LEAKAGE
QN - FLOW RATE AT TIME LEVEL N
QNP - FLOW RATE AT TIME LEVEL N+1
QSAIF - SOLUTE FLOW RATE AT AN A.I.F.B.C.
QSBC - SOLUTE FLOW RATE AT ANY B.C. CELL
QSFBC - SOLUTE FLOW RATE AT A SPECIFIED FLUX B.C.
QSFX - SPECIFIED SOLUTE FLUX IN THE X-DIRECTION
QSFY - SPECIFIED SOLUTE FLUX IN THE Y-DIRECTION
QSFZ - SPECIFIED SOLUTE FLUX IN THE Z-DIRECTION
QSLBC - SOLUTE FLOW RATE AT A LEAKAGE B.C.
QSLYR - SOLUTE FLOW RATE FROM A WELL TO AN AQUIFER LAYER
QSSBC - SOLUTE FLOW RATE FOR A SPECIFIED VALUE B.C.
QSW - SOLUTE FLOW RATE FROM A WELL TO THE AQUIFER
QWAV - AVERAGE FLOW RATE FROM A WELL TO AN AQUIFER LAYER OVER A TIME STEP
QWLYR - FLUID FLOW RATE FROM A WELL TO AN AQUIFER LAYER
QWM - FLUID MASS FLOW RATE FROM A WELL TO THE AQUIFER
QWN - FLUID FLOW RATE FROM A WELL TO THE AQUIFER AT TIME LEVEL N
QWNP - FLUID FLOW RATE OF A WELL AT TIME LEVEL N+1
QWR - FLUID FLOW RATE IN A WELL RISER
QWV - VOLUMETRIC FLUID FLOW RATE FROM A WELL TO THE AQUIFER
RBW - RIGHT-HAND BAND WIDTH FOR EACH EQUATION OF THE A4 MATRIX
RCPPM - HEAT CAPACITY OF THE POROUS MEDIUM PER UNIT VOLUME
RDAIF - TRUE IF AQUIFER INFLUENCE B.C. INFORMATION IS TO BE READ
RDCALC - TRUE IF CALCULATION INFORMATION IS TO BE READ

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
RDECHO - TRUE IF A READ-ECHO FILE IS TO BE WRITTEN
RDFLXH - TRUE IF HEAT FLUX DATA ARE TO BE READ
RDFLXQ - TRUE IF FLUID FLUX DATA ARE TO BE READ
RDFLXS - TRUE IF SOLUTE FLUX DATA ARE TO BE READ
RDLBC  - TRUE IF LEAKAGE B.C. DATA ARE TO BE READ
RDMPDT - TRUE IF CONTOUR MAP INFORMATION IS TO BE READ
RDPLTP - TRUE IF INFORMATION FOR PRESSURE PLOTS IS TO BE READ
RDSCBC - TRUE IF SPECIFIED SOLUTE CONCENTRATION B.C. DATA ARE TO BE READ
RDSPBC - TRUE IF SPECIFIED PRESSURE B.C. DATA ARE TO BE READ
RDSTBC - TRUE IF SPECIFIED TEMPERATURE B.C. DATA ARE TO BE READ
RDVAIF - TRUE IF A.I.F.B.C. GEOMETRIC FACTORS ARE TO BE READ
RDWDEF - TRUE IF WELL DEFINITION INFORMATION IS TO BE READ
RDWFLO - TRUE IF WELL FLOW RATE DATA ARE TO BE READ
RDWHD  - TRUE IF WELL-HEAD DATA AT LAND SURFACE ARE TO BE READ
REN    - REYNOLDS NUMBER FOR FLOW IN A WELL RISER OR WELL BORE
RESTRT - TRUE IF THIS IS A RESTART RUN
RF     - RIGHT-HAND-SIDE VECTOR FOR THE FLOW EQUATION
RH     - RIGHT-HAND-SIDE VECTOR FOR THE HEAT EQUATION
RH1    - RIGHT-HAND-SIDE VECTOR AUGMENTED WITH CROSS-DISPERSIVE FLUX TERMS
RHS    - RIGHT-HAND-SIDE VECTOR FOR THE SYSTEM EQUATIONS
RHSSBC - RIGHT-HAND-SIDE VECTOR FOR THE SPECIFIED VALUE B.C. NODES
RHSW   - RIGHT-HAND-SIDE VECTOR FOR THE WELL-BORE NODES FOR CYLINDRICAL COORDINATE SYSTEM
RIOAR  - RADIUS OF THE APPROXIMATE BOUNDARY BETWEEN THE INNER AND OUTER AQUIFER REGIONS FOR A.I.F.B.C.
RM     - RADII OF CELL BOUNDARIES IN THE R-DIRECTION FOR THE CYLINDRICAL COORDINATE SYSTEM
RORW2  - EQUIVALENT CELL RADIUS DIVIDED BY THE WELL RADIUS, QUANTITY SQUARED
RPRN   - '('
RS     - RIGHT-HAND-SIDE VECTOR FOR THE SOLUTE EQUATION
RS1    - RIGHT-HAND-SIDE VECTOR AUGMENTED WITH CROSS-DISPERSIVE FLUX TERMS
SAVLDO - TRUE IF ONLY THE LAST RESTART OR CHECKPOINT DUMP IS TO BE SAVED
SCALMF - TRUE IF A SCALED MASS FRACTION IS TO BE USED FOR INPUT AND OUTPUT
SDECAY - AMOUNT OF SOLUTE DECAYED DURING A TIME STEP
SHRES  - RESIDUAL ERROR IN THE HEAT EQUATION FOR CURRENT TIME STEP
SHRESF - FRACTIONAL RESIDUAL ERROR IN THE HEAT EQUATION FOR THE CURENT TIME STEP
SIR    - SOLUTE IN THE REGION
SIRO   - INITIAL SOLUTE IN THE REGION
SIRN   - SOLUTE IN THE REGION AT TIME LEVEL N
SLMETH - INDEX FOR SELECTION OF EQUATION SOLUTION METHOD
SMCALC - TRUE IF AN ABBREVIATED CAPACITANCE-COEFFICIENT CALCULATION IS TO BE DONE
SOLUTE - TRUE IF SOLUTE TRANSPORT IS BEING SIMULATED
SOLVE  - TRUE IF THE TWO-LINE-SOR SOLVER IS TO SOLVE THE EQUATIONS
SPRAD  - SPECTRAL RADIUS OF THE EQUATION COEFFICIENT MATRIX
SRES   - RESIDUAL ERROR IN THE FLOW EQUATION FOR THE CURRENT TIME STEP
SRESF  - FRACTIONAL RESIDUAL ERROR IN THE FLOW EQUATION FOR THE CURRENT TIME STEP
SSRES  - RESIDUAL ERROR IN THE SOLUTE EQUATION FOR THE CURRENT TIME STEP
SSRESF - FRACTIONAL RESIDUAL ERROR IN THE SOLUTE EQUATION FOR THE CURRENT TIME STEP
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
STOTFI - FLUID INPUT TO THE REGION DURING THE CURRENT TIME STEP
STOTFP - FLUID PRODUCED FROM THE REGION DURING THE CURRENT TIME STEP
STOTHI - HEAT INPUT TO THE REGION DURING THE CURRENT TIME STEP
STOTHP - HEAT PRODUCED FROM THE REGION DURING THE CURRENT TIME STEP
STOTSI - SOLUTE INPUT TO THE REGION DURING THE CURRENT TIME STEP
STOTSP - SOLUTE PRODUCED FROM THE REGION DURING THE CURRENT TIME STEP
SUMMOB - SUM OF ALL WELL MOBILITIES FOR A GIVEN WELL
SUMWI  - SUM OF ALL WELL INDICES FOR A GIVEN WELL
SVBC   - TRUE IF A SPECIFIED VALUE B.C. IS AT THIS CELL
SXX    - FLUID MASS FLOW RATE IN THE X-DIRECTION
SXXM   - FLUID MASS FLOW RATE IN X-DIRECTION BETWEEN CELL AT X(I-1) AND X(I)
SXXP   - FLUID MASS FLOW RATE IN X-DIRECTION BETWEEN CELL AT X(I) AND X(I+1)
SYMID  - CHARACTER SYMBOL FOR CONTOUR MAP BOUNDARIES AND NODES
SYY    - FLUID MASS FLOW RATE IN THE Y-DIRECTION
SYYM   - FLUID MASS FLOW RATE IN Y-DIRECTION BETWEEN CELL AT Y(J-1) AND Y(J)
SYYP   - FLUID MASS FLOW RATE IN Y-DIRECTION BETWEEN CELL AT Y(J) AND Y(J+1)
SZZ    - FLUID MASS FLOW RATE IN THE Z-DIRECTION
SZZM   - FLUID MASS FLOW RATE IN Z-DIRECTION BETWEEN CELL AT Z(K-1) AND Z(K)
SZZP   - FLUID MASS FLOW RATE IN Z-DIRECTION BETWEEN CELL AT Z(K) AND Z(K+1)
SZZW   - FLUID MASS FLOW RATE UP THE WELL BORE
T      - TEMPERATURE
T0     - REFERENCE TEMPERATURE FOR DENSITY
T0     - REFERENCE TEMPERATURE AT WHICH DENSITY DATA ARE GIVEN
T00    - TEMPERATURE AT THE END OF A WELL RISER, INITIAL VALUE
T0H    - REFERENCE TEMPERATURE FOR ENTHALPY AND VISCOSITY
TABWR  - TEMPERATURE IN THE AMBIENT MEDIUM AT THE BOTTOM OF A WELL RISER
TAIF   - TEMPERATURE IN THE OUTER AQUIFER REGION FOR A.I.F.B.C.
TAMBK  - TEMPERATURE IN THE AMBIENT MEDIUM AT A GIVEN LEVEL ALONG THE WELL RISER
TATWR  - TEMPERATURE IN THE AMBIENT MEDIUM AT THE TOP OF A WELL RISER
TC     - TIME VALUES FOR CALCULATED VARIABLES AT A WELL
TCS    - TEMPERATURE CALCULATED IN A WELL AT LAND SURFACE FOR TEMPORAL PLOT
TCW    - TEMPERATURE CALCULATED AT A WELL DATUM FOR TEMPORAL PLOT
TDEHIR - CUMULATIVE CHANGE IN ENTHALPY IN THE REGION
TDFIR  - CUMULATIVE CHANGE IN FLUID IN THE REGION
TDSIR  - CUMULATIVE CHANGE IN SOLUTE IN THE REGION
TDX    - DISPERSIVE CONDUCTANCE IN X-DIRECTION
TDXY   - CROSS-DISPERSIVE CONDUCTANCE
TDXZ   - CROSS-DISPERSIVE CONDUCTANCE
TDY    - DISPERSIVE CONDUCTANCE IN Y-DIRECTION
TDYX   - CROSS-DISPERSIVE CONDUCTANCE
TDYZ   - CROSS-DISPERSIVE CONDUCTANCE
TDZ    - DISPERSIVE CONDUCTANCE IN Z-DIRECTION
TDZX   - CROSS-DISPERSIVE CONDUCTANCE
TDZY   - CROSS-DISPERSIVE CONDUCTANCE
TEHDT  - TEMPERATURE VALUES FOR THE TABLE OF ENTHALPY DEVIATION
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

TEHST - TEMPERATURE VALUES FOR THE TABLE OF ENTHALPY AT SATURATION
TFLX - TEMPERATURE FOR INFLOW AT A SPECIFIED FLUX B.C.
TFRES - CUMULATIVE RESIDUAL ERROR IN THE FLOW EQUATION
TFRESF - FRACTIONAL CUMULATIVE RESIDUAL ERROR IN THE FLOW EQUATION
TFW - CONDUCTANCE IN WELL BORE FOR CYLINDRICAL SYSTEM
TFX - FLUID CONDUCTANCE IN X-DIRECTION
TFY - FLUID CONDUCTANCE IN Y-DIRECTION
TFZ - FLUID CONDUCTANCE IN Z-DIRECTION
THCBC - TEMPERATURE PROFILE FOR FIRST PROBLEM FOR HEAT CONDUCTION B.C.
THETXZ - ANGLE BETWEEN X-AXIS AND GRAVITATIONAL VECTOR
THETYZ - ANGLE BETWEEN Y-AXIS AND GRAVITATIONAL VECTOR
THETZZ - ANGLE BETWEEN Z-AXIS AND GRAVITATIONAL VECTOR
THRES - CUMULATIVE RESIDUAL ERROR IN THE HEAT EQUATION
THRESF - FRACTIONAL CUMULATIVE RESIDUAL ERROR IN THE HEAT EQUATION
THRU - TRUE IF THIS IS THE END OF THE SIMULATION
THX - THERMAL CONDUCTANCE IN X-DIRECTION
THXY - CROSS-DISPERSIVE THERMAL CONDUCTANCE
THXZ - CROSS-DISPERSIVE THERMAL CONDUCTANCE
THY - THERMAL CONDUCTANCE IN Y-DIRECTION
THYX - CROSS-DISPERSIVE THERMAL CONDUCTANCE
THYZ - CROSS-DISPERSIVE THERMAL CONDUCTANCE
THZ - THERMAL CONDUCTANCE IN Z-DIRECTION
THZX - CROSS-DISPERSIVE THERMAL CONDUCTANCE
THZY - CROSS-DISPERSIVE THERMAL CONDUCTANCE
TILT - TRUE IF THE COORDINATE SYSTEM IS TILTED SO THAT THE GRAVITATIONAL VECTOR DOES NOT POINT IN THE NEGATIVE Z-DIRECTION
TIMCHG - TIME AT WHICH NEW TRANSIENT DATA ARE TO BE READ
TIMDAY - TIME IN DAYS
TIME - TIME VALUE
TIMED - DIMENSIONLESS TIME
TIMEDN - DIMENSIONLESS TIME AT TIME LEVEL N
TIMRST - TIME AT WHICH A RESTART RECORD IS TO BE WRITTEN
TITLE - TITLE FOR THE SIMULATION RUN
TITLEO - TITLE FOR THE ORIGINAL SIMULATION THAT IS BEING RESTARTED
TLBC - TEMPERATURE AT THE OTHER SIDE OF AN CONFINING LAYER FOR A LEAKAGE B.C.
TLBL - LABEL FOR TEMPERATURE FOR ARRAY PRINTOUTS
TM1 - CONDUCTANCE COEFFICIENT VECTOR FOR TWO-LINE SOR SOLVER
TM2 - CONDUCTANCE COEFFICIENT VECTOR FOR TWO-LINE SOR SOLVER
TMLBL - LABEL FOR TIME FOR TEMPORAL PLOTS
TNP - TEMPERATURE FOR SPECIFIED VALUE B.C.
TO - TIME VALUES FOR OBSERVED VARIABLES IN A WELL FOR TEMPORAL PLOTS
TOL - TOLERANCE FOR AN ITERATIVE CALCULATION
TOLDEN - TOLERANCE ON DENSITY CHANGES FOR CONVERGENCE OF THE SOLUTION CYCLE OF THE THREE SYSTEM EQUATIONS
TOLDNC - TOLERANCE ON DENSITY CHANGES DUE TO SOLUTE CONCENTRATION CHANGES FOR THE SOLUTION CYCLE OF THE FLOW AND SOLUTE EQUATIONS
TOLDNT - TOLERANCE ON DENSITY CHANGES DUE TO TEMPERATURE CHANGES FOR THE SOLUTION CYCLE OF THE FLOW AND HEAT EQUATIONS
TOLDPW - TOLERANCE ON PRESSURE CHANGES FOR THE WELL BORE CALCULATION

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
TOLFPW - TOLERANCE ON FRACTIONAL PRESSURE CHANGES FOR THE WELL-BORE CALCULATION
TOLQW  - TOLERANCE ON FLOW RATE CHANGES FOR THE WELL-BORE CALCULATION
TOS    - TEMPERATURES OBSERVED IN A WELL AT THE LAND SURFACE FOR TEMPORAL PLOT
TOTFI  - CUMULATIVE FLUID INPUT TO THE REGION
TOTFP  - CUMULATIVE FLUID PRODUCED FROM THE REGION
TOTHI  - CUMULATIVE HEAT INPUT TO THE REGION
TOTHP  - CUMULATIVE HEAT PRODUCED FROM THE REGION
TOTSI  - CUMULATIVE SOLUTE INPUT TO THE REGION
TOTSP  - CUMULATIVE SOLUTE PRODUCED FROM THE REGION
TOTWFI - CUMULATIVE FLUID INJECTED BY THE WELLS
TOTWFP - CUMULATIVE FLUID PRODUCED BY THE WELLS
TOTWHI - CUMULATIVE HEAT INJECTED BY THE WELLS
TOTWHP - CUMULATIVE HEAT PRODUCED BY THE WELLS
TOTWSI - CUMULATIVE SOLUTE INJECTED BY THE WELLS
TOTWSP - CUMULATIVE SOLUTE PRODUCED BY THE WELLS
TOW    - TEMPERATURES OBSERVED IN A WELL AT THE WELL DATUM FOR TEMPORAL PLOT
TP1    - CONDUCTANCE COEFFICIENT VECTOR FOR TWO-LINE SOR SOLVER
TP2    - CONDUCTANCE COEFFICIENT VECTOR FOR TWO-LINE SOR SOLVER
TPHCBC - TEMPERATURE PROFILE FOR SECOND PROBLEM FOR HEAT CONDUCTION B.C.
TQFAIF - CUMULATIVE FLUID INPUT FROM A.I.F.B.C.
TQFFBC - CUMULATIVE FLUID INPUT FROM SPECIFIED FLUX B.C.
TQFINJ - TOTAL FLUID INJECTION RATE FOR ALL WELLS
TQFLBC - CUMULATIVE FLUID INPUT FROM LEAKAGE B.C.
TQFPRO - TOTAL FLUID PRODUCTION RATE FROM ALL WELLS
TQFSBC - CUMULATIVE FLUID INPUT FROM SPECIFIED VALUE B.C.
TQHAIF - CUMULATIVE HEAT INPUT FROM A.I.F.B.C.
TQHFBC - CUMULATIVE HEAT INPUT FROM SPECIFIED FLUX B.C.
TQHHBC - CUMULATIVE HEAT INPUT FROM HEAT CONDUCTION B.C.
TQHINJ - TOTAL HEAT INJECTION RATE FROM ALL WELLS
TQHLBC - CUMULATIVE HEAT INPUT FROM LEAKAGE B.C.
TQHPRO - TOTAL HEAT PRODUCTION RATE FROM ALL WELLS
TQHSBC - CUMULATIVE HEAT INPUT FROM SPECIFIED VALUE B.C.
TQSAIF - CUMULATIVE SOLUTE INPUT FROM A.I.F.B.C.
TQSFBC - CUMULATIVE SOLUTE INPUT FROM SPECIFIED FLUX B.C.
TQSINJ - TOTAL SOLUTE INJECTION RATE FROM ALL WELLS
TQSLBC - CUMULATIVE SOLUTE INPUT FROM LEAKAGE B.C.
TQSPRO - TOTAL SOLUTE PRODUCTION RATE FROM ALL WELLS
TQSSBC - CUMULATIVE SOLUTE INPUT FROM SPECIFIED VALUE B.C.
TRVIS  - REFERENCE TEMPERATURE AT WHICH VISCOSITY DATA ARE GIVEN
TSBC   - TEMPERATURE FOR INFLOW AT A SPECIFIED PRESSURE B.C.
TSLBL  - LABEL FOR WELL TEMPERATURE AT LAND SURFACE FOR TEMPORAL PLOTS
TSMAX  - MAXIMUM TEMPERATURE AT LAND SURFACE FOR SCALING TEMPORAL PLOTS
TSMIN  - MINIMUM TEMPERATURE AT LAND SURFACE FOR SCALING TEMPORAL PLOTS
TSRES  - CUMULATIVE RESIDUAL ERROR IN THE SOLUTE EQUATION
TSRESF - FRACTIONAL CUMULATIVE ERROR IN THE SOLUTE EQUATION
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

```
TSX    - SOLUTE CONDUCTANCE IN X-DIRECTION
TSXY   - CROSS-DISPERSIVE SOLUTE CONDUCTANCE
TSXZ   - CROSS-DISPERSIVE SOLUTE CONDUCTANCE
TSY    - SOLUTE CONDUCTANCE IN Y-DIRECTION
TSYX   - CROSS-DISPERSIVE SOLUTE CONDUCTANCE
TSYZ   - CROSS-DISPERSIVE SOLUTE CONDUCTANCE
TSZ    - SOLUTE CONDUCTANCE IN Z-DIRECTION
TSZX   - CROSS-DISPERSIVE SOLUTE CONDUCTANCE
TSZY   - CROSS-DISPERSIVE SOLUTE CONDUCTANCE
TVD    - TEMPERATURE VS. DEPTH VALUES FOR INITIAL CONDITION
TVF0   - TEMPERATURE VALUES FOR VISCOSITY VS. TEMPERATURE DATA AT MINIMUM SOLUTE CONCENTRATION
TVF1   - TEMPERATURE VALUES FOR VISCOSITY VS. TEMPERATURE DATA AT MAXIMUM SOLUTE CONCENTRATION
TVZHC  - TEMPERATURE VS. DISTANCE OUTWARD FROM HEAT CONDUCTION B.C. FOR INITIAL CONDITION
TWKT   - TEMPERATURE IN A WELL AT THE WELL DATUM
TWLBL  - LABEL FOR WELL TEMPERATURE AT WELL DATUM FOR TEMPORAL PLOTS
TWMAX  - MAXIMUM TEMPERATURE AT WELL DATUM FOR SCALING TEMPORAL PLOTS
TWMIN  - MINIMUM TEMPERATURE AT WELL DATUM FOR SCALING TEMPORAL PLOTS
TWOPI  - '6.28...'
TWREND - TEMPERATURE AT THE END OF A WELL RISER PIPE
TWRK   - TEMPERATURE IN A WELL RISER AT A GIVEN LEVEL
TWSRKT - TEMPERATURE AT LAND SURFACE OR AT WELL DATUM
TWSUR  - TEMPERATURE IN A WELL AT THE LAND SURFACE
TX     - FLUID CONDUCTANCE FACTOR IN THE X-DIRECTION
TY     - FLUID CONDUCTANCE FACTOR IN THE Y-DIRECTION
TZ     - FLUID CONDUCTANCE FACTOR IN THE Z-DIRECTION
    THE UNDEFINED 'U' VARIABLES ARE USED FOR TEMPORARY STORAGE
UCROSC - CROSS-DISPERSIVE SOLUTE FLUX
UCROST - CROSS-DISPERSIVE HEAT FLUX
UNIGRX - TRUE IF A UNIFORM GRID IN THE X-DIRECTION IS DESIRED
UNIGRY - TRUE IF A UNIFORM GRID IN THE Y-DIRECTION IS DESIRED
UNIGRZ - TRUE IF A UNIFORM GRID IN THE Z-DIRECTION IS DESIRED
UNITEP - UNIT FOR ENERGY PER UNIT MASS
UNITH  - UNIT FOR HEAT
UNITHF - UNIT FOR HEAT FLOW RATE
UNITL  - UNIT FOR LENGTH
UNITM  - UNIT FOR MASS
UNITP  - UNIT FOR PRESSURE
UNITT  - UNIT FOR TEMPERATURE
UNITTM - UNIT FOR TIME
UNITVS - UNIT FOR VISCOSITY
VA     - SYSTEM EQUATION COEFFICIENT MATRIX
VAIFC  - GEOMETRIC FACTOR FOR A.I.F.B.C.
VAR    - VARIABLE FOR ARRAY INPUT
VASBC  - SYSTEM EQUATION COEFFICIENT MATRIX FOR NODES WITH SPECIFIED B.C.
VAW    - SYSTEM EQUATION COEFFICIENT MATRIX FOR NODES ALONG WELL BORE IN CYLINDRICAL COORDINATES
```

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

VDATA - VISCOSITY DATA
VELWRK - VELOCITY OF FLOW IN A WELL RISER AT A GIVEN LEVEL
VIS - VISCOSITY
VISCTR - VISCOSITY VS. SOLUTE CONCENTRATION DATA AT REFERENCE TEMPERATURE
VISLBC - VISCOSITY AT OTHER SIDE OF CONFINING LAYER FOR LEAKAGE B.C.
VISOAR - VISCOSITY IN OUTER AQUIFER REGION FOR A.I.F.B.C.
VISTF0 - VISCOSITY VS. TEMPERATURE DATA AT MINIMUM SOLUTE CONCENTRATION
VISTF1 - VISCOSITY VS. TEMPERATURE DATA AT MAXIMUM SOLUTE CONCENTRATION
VPA - VARIABLY PARTITIONED ARRAY
VREF - REFERENCE VISCOSITY
VSTLOG - LOGARITHM OF VISCOSITY
VXX - INTERSTITIAL VELOCITY IN THE X-DIRECTION
VYY - INTERSTITIAL VELOCITY IN THE Y-DIRECTION
VZZ - INTERSTITIAL VELOCITY IN THE Z-DIRECTION
W0 - MINIMUM MASS FRACTION FOR SCALING
W1 - MAXIMUM MASS FRACTION FOR SCALING
WBOD - WELL-BORE OUTER DIAMETER
WCAIF - CUMULATIVE INFLOW AT AQUIFER INFLUENCE FUNCTION B.C.
WCF - WELL COMPLETION FACTOR FOR A GIVEN LAYER
WCLBL1 - LABEL DESCRIBING WELL CALCULATION TYPE
WCLBL2 - LABEL DESCRIBING WELL-FLOW ALLOCATION METHOD
WFICUM - CUMULATIVE FLUID INJECTED BY A WELL
WFPCUM - CUMULATIVE FLUID PRODUCED BY A WELL
WHICUM - CUMULATIVE HEAT INJECTED BY A WELL
WHPCUM - CUMULATIVE HEAT PRODUCED BY A WELL
WI - WELL INDEX
WIDLBL - WELL IDENTIFICATION LABEL FOR TEMPORAL PLOT
WQMETH - INDEX OF WELL-FLOW CALCULATION METHOD
WRANGL - WELL-RISER ANGLE WITH THE GRAVITATIONAL VECTOR
WRCALC - TRUE IF WELL-RISER CALCULATIONS ARE TO BE PERFORMED
WRID - WELL-RISER PIPE INSIDE DIAMETER
WRIDT - WELL-RISER PIPE INSIDE DIAMETER FOR A GIVEN WELL
WRISL - WELL-RISER PIPE LENGTH
WRRUF - WELL-RISER PIPE ROUGHNESS PARAMETER
WSICUM - CUMULATIVE SOLUTE INJECTED BY A WELL
WSPCUM - CUMULATIVE SOLUTE PRODUCED BY A WELL
WT - WEIGHT FACTOR FOR SPATIAL DISCRETIZATION
X - X-COORDINATE NODE LOCATIONS
XC - X-COORDINATE LOCATION OF CALCULATED DATA FOR TEMPORAL PLOT
XI - DIMENSIONLESS TIME PARAMETER FOR HEAT TRANSFER FROM WELL RISER
XLBL - LABEL FOR X-AXIS FOR TEMPORAL PLOTS
X0 - X-COORDINATE LOCATION OF OBSERVED DATA FOR TEMPORAL PLOTS
XX - SOLUTION VECTOR FROM TWO-LINE SOR SOLVER
XXN - NEW SOLUTION VECTOR FROM TWO-LINE SOR SOLVER
Y - Y-COORDINATE NODE LOCATIONS

Table 11.1.--*Definition list for selected HST3D program variables*--Continued

YC - Y-COORDINATE LOCATIONS OF CALCULATED DATA FOR TEMPORAL PLOTS
YLBL - LABEL FOR Y-AXIS FOR TEMPORAL PLOTS
YO - Y-COORDINATE LOCATION OF OBSERVED DATA FOR TEMPORAL PLOTS
YPOSUP - TRUE IF THE Y-AXIS IS POSITIVE UPWARD FOR MAPS
YY - DEPENDENT VARIABLE VECTOR FOR WELL RISER INTEGRATION
Z - Z-COORDINATE NODE LOCATIONS
ZCHARS - CHARACTER ARRAY FOR ZEBRA-STRIPED CONTOUR MAPS
ZEBRA - TRUE IF ZONED CONTOUR MAPS ARE TO BE ZEBRA STRIPED WITH ALTERNATING SYMBOL AND BLANK ZONES
ZELBC - ELEVATION OF THE OUTER SURFACE OF A CONFINING LAYER FOR A LEAKAGE B.C.
ZHCBC - NODE LOCATIONS OUTSIDE THE REGION IN A NORMAL DIRECTION TO A HEAT CONDUCTION BOUNDARY FACE FOR HEAT CONDUCTION B.C.
ZPINIT - Z-COORDINATE LOCATION OF SPECIFIED INITIAL PRESSURE FOR HYDROSTATIC INITIAL CONDITIONS
ZPOSUP - TRUE IF THE Z-AXIS IS POSITIVE UPWARD FOR MAPS
ZT - Z-COORDINATE LOCATIONS OF TEMPERATURE PROFILE DATA FOR INITIAL CONDITIONS
ZTHC - LOCATIONS ALONG THE OUTWARD NORMAL TO HEAT-CONDUCTION BOUNDARY FACES OF TEMPERATURE PROFILE DATA FOR INITIAL CONDITIONS
ZWK - LOCATION ALONG A WELL-RISER PIPE
ZWKT - Z-COORDINATE LOCATION OF A WELL DATUM

11.2 CROSS-REFERENCE LIST OF VARIABLES

The following cross-reference list (table 11.2) shows in which subprograms each variable appears. Actual line numbers within each subprogram are not provided; however, every FORTRAN compiler can provide a local cross-reference map for a given subprogram. The first column contains the variable names and the second column lists the subprograms that employ each variable. The third column gives the variable type and the dimension, if an array. Some variables are scalars in some subprograms and arrays in others; however, the definitions are the same.

Table 11.2--*Cross-reference list of variables*

Variable name	Referencing programs	Variable type
A	MAP2D	REAL*8 DIMENSION(*,*)
A0	MAP2D	REAL*8
A1	BSODE	REAL*8
	INTERP	REAL*8
	MAP2D	REAL*8
A1HC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
A2	BSODE	REAL*8
	INTERP	REAL*8
A2HC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
A3HC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
A4	D4DES	REAL*8 DIMENSION(*)
AA1	APLYBC	REAL*8 DIMENSION(*)
AA2	APLYBC	REAL*8 DIMENSION(*)
AA3	APLYBC	REAL*8 DIMENSION(*)
AA4	APLYBC	REAL*8 DIMENSION(*)
AAIF	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
ABOAR	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
ABPM	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
ACHR	MAP2D	REAL*8
ALBC	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
ALLOUT	ERROR2	LOGICAL*4
ALPHL	COEFF	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
AMAX	HST3D	REAL*8 DIMENSION(3)
	ETOM2	REAL*8 DIMENSION(3)
	INIT3	REAL*8 DIMENSION(3)
	MAP2D	REAL*8
	READ3	REAL*8 DIMENSION(3)
	WRITE3	REAL*8 DIMENSION(3)
	WRITE5	REAL*8 DIMENSION(3)
AMAXN	MAP2D	REAL*8
AMIN	HST3D	REAL*8 DIMENSION(3)
	ETOM2	REAL*8 DIMENSION(3)
	INIT3	REAL*8 DIMENSION(3)
	MAP2D	REAL*8
	READ3	REAL*8 DIMENSION(3)
	WRITE3	REAL*8 DIMENSION(3)
	WRITE5	REAL*8 DIMENSION(3)
AMINN	MAP2D	REAL*8
AMN	MAP2D	REAL*8
AMX	MAP2D	REAL*8
ANGOAR	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8

Variable name	Referencing programs	Variable type
APLOT	MAP2D	REAL*8 DIMENSION(*,*)
APLOT1	MAP2D	REAL*8
APLOT2	MAP2D	REAL*8
APLOT3	MAP2D	REAL*8
APLOT4	MAP2D	REAL*8
APRNT	PRNTAR	REAL*8 DIMENSION(10)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
ARGRID	HST3D	LOGICAL*4
	INIT2	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
ARRAY	PRNTAR	REAL*8 DIMENSION(*)
ARWB	WELLSS	REAL*8
ARX	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
ARXBC	INIT2	REAL*8 DIMENSION(*)
ARXFBC	INIT2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
ARY	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
ARYBC	INIT2	REAL*8 DIMENSION(*)
ARYFBC	INIT2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
ARZ	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
ARZBC	INIT2	REAL*8 DIMENSION(*)
ARZFBC	INIT2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
AST	PLOT	CHARACTER*1
	ZONPLT	CHARACTER*1
AUTOTS	HST3D	LOGICAL*4
	COEFF	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
B0	BSODE	REAL*8
B1	BSODE	REAL*8
	WFDYDZ	REAL*8
B2	WFDYDZ	REAL*8

Variable name	Refer-encing programs	Variable type
BAIF	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
BBAIF	HST3D	REAL*8 DIMENSION(0:3)
	APLYBC	REAL*8 DIMENSION(0:3)
	BLOCKDATA	REAL*8 DIMENSION(0:3)
	DUMP	REAL*8 DIMENSION(0:3)
	READ1	REAL*8 DIMENSION(0:3)
	READ2	REAL*8 DIMENSION(0:3)
	WRITE2	REAL*8 DIMENSION(0:3)
BBLBC	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
BETA	APLYBC	REAL*8 DIMENSION(10)
BLANK	MAP2D	CHARACTER*1
	PLOT	CHARACTER*1
	PRNTAR	CHARACTER*1
	WRITE3	CHARACTER*11
	ZONPLT	CHARACTER*1
BLANKL	PLOT	CHARACTER*51
BLBC	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
BOAR	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
BP	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
BT	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
BV	VSINIT	REAL*8
BV0	HST3D	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	VISCOS	REAL*8
BV1	HST3D	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	VISCOS	REAL*8
C	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	CALCC	REAL*8
	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	VISCOS	REAL*8
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
	ZONPLT	REAL*8
C00	HST3D	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
C1	ZONPLT	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
C11	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	WFDYDZ	REAL*8
C12	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	WFDYDZ	REAL*8
C13	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
C2	ZONPLT	REAL*8
C21	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	WFDYDZ	REAL*8
C22	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	WFDYDZ	REAL*8
C23	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
C24	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
C31	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
C32	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
C33	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
C34	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
C35	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
CAIF	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
CC24	ASEMBL	REAL*8 DIMENSION(*)
CC34	ASEMBL	REAL*8 DIMENSION(*)
CC35	ASEMBL	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
CCLBL	PLOTOC	CHARACTER*50
CCW	PLOTOC	REAL*8 DIMENSION(*)
CFLX	APLYBC	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
CHAPRT	WRITE5	CHARACTER*12 DIMENSION(10)
CHARC	READ1	CHARACTER*1
CHARS	MAP2D	CHARACTER*1 DIMENSION(0:31)
CHK1	WRITE5	CHARACTER*2
CHKPTD	HST3D	LOGICAL*4
	CLOSE	LOGICAL*4
	DUMP	LOGICAL*4
	READ3	LOGICAL*4
CHU3	WRITE5	CHARACTER*12
CHU4	WRITE5	CHARACTER*8
CHU5	WRITE5	CHARACTER*8
CHU6	WRITE5	CHARACTER*8
CI	D4DES	INTEGER*4 DIMENSION(6,*)
	ORDER	INTEGER*4 DIMENSION(6,*)

Variable name	Referencing programs	Variable type
CIBC	APLYBC	CHARACTER*9
	ASEMBL	CHARACTER*9
	ERROR2	CHARACTER*9
	ERROR3	CHARACTER*9
	ETOM1	CHARACTER*9
	INIT2	CHARACTER*9
	INIT3	CHARACTER*9
	SBCFLO	CHARACTER*9
	SUMCAL	CHARACTER*9
	WBCFLO	CHARACTER*9
	WRITE2	CHARACTER*9
	WRITE3	CHARACTER*9
	WRITE5	CHARACTER*9
CIBC1	SOR2L	CHARACTER*9
CIBC2	SOR2L	CHARACTER*9
CICALL	IREWI	CHARACTER*1
CLBC	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
CLINE	MAP2D	REAL*8
CMAX	PLOTOC	REAL*8
CMIN	PLOTOC	REAL*8
CMX	ASEMBL	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
CMY	ASEMBL	REAL*8
CMZ	ASEMBL	REAL*8
CN3	PRNTAR	CHARACTER*2
CNP	CALCC	REAL*8
	READ3	REAL*8 DIMENSION(*)
CNV	ETOM1	REAL*8
	PRNTAR	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE5	REAL*8
CNVD	ETOM1	REAL*8
	ETOM2	REAL*8
	READ3	REAL*8
CNVDFI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8

Variable name	Refer-encing programs	Variable type
CNVDI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVE	ETOM2	REAL*8
CNVEPI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	READ1	REAL*8
	WRITE3	REAL*8
CNVFF	ETOM2	REAL*8
	READ3	REAL*8
CNVGZ	READ3	REAL*8
CNVHC	ETOM1	REAL*8
	ETOM2	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
CNVHCI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVHF	ETOM2	REAL*8
	READ3	REAL*8
CNVHFI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVHI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8

Variable name	Referencing programs	Variable type
CNVHTC	ETOM1	REAL*8
	ETOM2	REAL*8
CNVHTI	WRITE2	REAL*8
CNVL	ETOM1	REAL*8
	READ3	REAL*8
CNVL2	ETOM1	REAL*8
	ETOM2	REAL*8
	READ3	REAL*8
CNVL2I	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVL3	ETOM1	REAL*8
	ETOM2	REAL*8
CNVL3I	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
CNVLI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVMFI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVMI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVP	ETOM1	REAL*8
	ETOM2	REAL*8
	READ3	REAL*8
CNVPI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVSF	ETOM2	REAL*8
	READ3	REAL*8
CNVT1	ETOM1	REAL*8
	ETOM2	REAL*8
	READ3	REAL*8
CNVT1I	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
CNVT2	ETOM1	REAL*8
	ETOM2	REAL*8
	READ3	REAL*8
CNVT2I	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVTCI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVTHC	ETOM1	REAL*8
	ETOM2	REAL*8
CNVTM	ETOM1	REAL*8
	ETOM2	REAL*8
	READ1	REAL*8
CNVTMI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVUUI	WRITE4	REAL*8
CNVVF	ETOM1	REAL*8
	READ3	REAL*8
CNVVLI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
CNVVSI	HST3D	REAL*8
	DUMP	REAL*8
	INIT1	REAL*8
	PLOTOC	REAL*8
	READ1	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
COLS	MAP2D	REAL*8
COMOPT	ITER	LOGICAL*4
	L2SOR	LOGICAL*4
CONLBL	WRITE5	CHARACTER*20
CONVC	HST3D	LOGICAL*4
	ITER	LOGICAL*4
CONVP	HST3D	LOGICAL*4
	ITER	LOGICAL*4
CONVRG	BSODE	LOGICAL*4
CONVT	HST3D	LOGICAL*4
	ITER	LOGICAL*4
COW	PLOTOC	REAL*8 DIMENSION(*)
CPAR	IREWI	CHARACTER*9
CPARI	IREWI	CHARACTER*6

Variable name	Referencing programs	Variable type
CPF	HST3D	REAL*8
	APLYBC	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	COEFF	REAL*8
	CRSDSP	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
CPX	ASEMBL	REAL*8
CPY	ASEMBL	REAL*8
CPZ	ASEMBL	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
CSBC	ASEMBL	REAL*8 DIMENSION(*)	CYLIND	HST3D	LOGICAL*4
	INIT3	REAL*8 DIMENSION(*)		ASEMBL	LOGICAL*4
	SUMCAL	REAL*8 DIMENSION(*)		CLOSE	LOGICAL*4
	WRITE3	REAL*8 DIMENSION(*)		COEFF	LOGICAL*4
				DUMP	LOGICAL*4
CVIS	HST3D	REAL*8 DIMENSION(10)		ERROR2	LOGICAL*4
	DUMP	REAL*8 DIMENSION(10)		ERROR3	LOGICAL*4
	ETOM1	REAL*8 DIMENSION(10)		ETOM1	LOGICAL*4
	INIT2	REAL*8 DIMENSION(10)		INIT1	LOGICAL*4
	READ1	REAL*8 DIMENSION(10)		INIT2	LOGICAL*4
	READ2	REAL*8 DIMENSION(10)		INIT3	LOGICAL*4
	VISCOS	REAL*8 DIMENSION(10)		IREWI	LOGICAL*4
	WRITE2	REAL*8 DIMENSION(10)		ITER	LOGICAL*4
				L2SOR	LOGICAL*4
CWKT	ASEMBL	REAL*8 DIMENSION(*)		PLOTOC	LOGICAL*4
	READ3	REAL*8 DIMENSION(*)		READ1	LOGICAL*4
	WBBAL	REAL*8 DIMENSION(*)		READ2	LOGICAL*4
	WELLSS	REAL*8 DIMENSION(*)		READ3	LOGICAL*4
	WRITE3	REAL*8 DIMENSION(*)		REWI	LOGICAL*4
	WRITE5	REAL*8 DIMENSION(*)		REWI3	LOGICAL*4
				SUMCAL	LOGICAL*4
				WBBAL	LOGICAL*4
				WELLSS	LOGICAL*4
				WRITE1	LOGICAL*4
				WRITE2	LOGICAL*4
				WRITE3	LOGICAL*4
				WRITE4	LOGICAL*4
				WRITE5	LOGICAL*4
				ZONPLT	LOGICAL*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
D	SOR2L	REAL*8 DIMENSION(*)
DAMWRC	HST3D	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8
DASHES	WRITE5	CHARACTER*50
DBKD	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
DC	ASEMBL	REAL*8 DIMENSION(*)
	CALCC	REAL*8
	CRSDSP	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
DCMAX	HST3D	REAL*8
	COEFF	REAL*8
	ITER	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
DCTAS	HST3D	REAL*8
	COEFF	REAL*8
	ERROR3	REAL*8
	ETOM2	REAL*8
	INIT3	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
DDNMAX	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
DDV	SBCFLO	REAL*8 DIMENSION(*)
	WBCFLO	REAL*8 DIMENSION(*)
DECLAM	HST3D	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	COEFF	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	INIT3	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	READ3	REAL*8
	SUMCAL	REAL*8
	VISCOS	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
DEHIR	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
DELA	MAP2D	REAL*8
DELAN	MAP2D	REAL*8
DELAP	MAP2D	REAL*8
DELTIM	HST3D	REAL*8
	APLYBC	REAL*8
	CALCC	REAL*8
	COEFF	REAL*8
	DUMP	REAL*8
	ETOM2	REAL*8
	INIT2	REAL*8
	INIT3	REAL*8
	READ1	REAL*8
	READ3	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
DELX	INIT2	REAL*8
DELY	INIT2	REAL*8
	PLOT	REAL*8
DELZ	INIT2	REAL*8
DEN	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
DENO	HST3D	REAL*8
	CALCC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WFDYDZ	REAL*8
DENC	HST3D	REAL*8
	CALCC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WFDYDZ	REAL*8
DENCHC	ITER	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
DENCHT	ITER	REAL*8
DENF0	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
DENF1	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
DENFBC	APLYBC	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
DENGL	WELLSS	REAL*8
DENLBC	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
DENMAX	SUMCAL	REAL*8
DENN	CALCC	REAL*8
DENNP	CALCC	REAL*8
DENOAR	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	ETOM1	REAL*8
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
DENP	HST3D	REAL*8
	CALCC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WFDYDZ	REAL*8
DENT	HST3D	REAL*8
	CALCC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WFDYDZ	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
DENW	WELLSS	REAL*8 DIMENSION(*)
DENWKT	ASEMBL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
DENWRK	HST3D	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
DET	WFDYDZ	REAL*8
DFIR	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
DM	HST3D	REAL*8
	ASEMBL	REAL*8
	COEFF	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	INIT3	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	READ3	REAL*8
	SUMCAL	REAL*8
	VISCOS	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE5	REAL*8
DNTEST	WELLSS	REAL*8
DOTS	CLOSE	CHARACTER*50
	REWI	CHARACTER*50
	REWI3	CHARACTER*50
	WRITE2	CHARACTER*50
	WRITE3	CHARACTER*50
	WRITE5	CHARACTER*50
DP	ASEMBL	REAL*8 DIMENSION(*)
	CALCC	REAL*8
	ITER	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
DPMAX	HST3D	REAL*8
	COEFF	REAL*8
	ITER	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
DPTAS	HST3D	REAL*8
	COEFF	REAL*8
	ERROR3	REAL*8
	ETOM2	REAL*8
	INIT3	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
DPUDT	APLYBC	REAL*8
DPWKT	ASEMBL	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
DQFDP	ASEMBL	REAL*8
DQHBC	ASEMBL	REAL*8
	SUMCAL	REAL*8
DQHCDT	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
DQHDP	ASEMBL	REAL*8
DQHDT	ASEMBL	REAL*8
DQHWDP	ASEMBL	REAL*8
DQHWDT	ASEMBL	REAL*8
DQSBC	ASEMBL	REAL*8
	SUMCAL	REAL*8
DQSDC	ASEMBL	REAL*8
DQSDP	ASEMBL	REAL*8
DQSWDC	ASEMBL	REAL*8
DQSWDP	ASEMBL	REAL*8
DQWDP	ASEMBL	REAL*8
DQWDPL	ASEMBL	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)

Variable name	Referencing programs	Variable type
DQWLYR	ASEMBL	REAL*8
DSIR	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
DSXXM	ASEMBL	REAL*8
DSXXP	ASEMBL	REAL*8
DSYYM	ASEMBL	REAL*8
DSYYP	ASEMBL	REAL*8
DSZZM	ASEMBL	REAL*8
DSZZP	ASEMBL	REAL*8
DT	ASEMBL	REAL*8 DIMENSION(*)
	CALCC	REAL*8
	CRSDSP	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
DTADZW	HST3D	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
DTHAWR	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)
DTHHC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
DTIMMN	HST3D	REAL*8
	COEFF	REAL*8
	ERROR3	REAL*8
	ETOM2	REAL*8
	INIT3	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
DTIMMX	HST3D	REAL*8
	COEFF	REAL*8
	ERROR3	REAL*8
	ETOM2	REAL*8
	INIT3	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
DTMAX	HST3D	REAL*8
	COEFF	REAL*8
	ITER	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
DTTAS	HST3D	REAL*8
	COEFF	REAL*8
	ERROR3	REAL*8
	ETOM2	REAL*8
	INIT3	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
DX	ZONPLT	REAL*8
DXMIN	PLOT	REAL*8
DXPRNT	PLOT	REAL*8
DY	ZONPLT	REAL*8
DYY	BSODE	REAL*8 DIMENSION(2)
	WELRIS	REAL*8 DIMENSION(2)
	WFDYDZ	REAL*8 DIMENSION(2)
DYYN	BSODE	REAL*8 DIMENSION(2)
DZ	ASEMBL	REAL*8
	BSODE	REAL*8
	CALCC	REAL*8
DZCHNG	BSODE	REAL*8
DZMIN	HST3D	REAL*8
	BSODE	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WELRIS	REAL*8
DZW	WELRIS	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type	Variable name	Refer-encing programs	Variable type
EEUNIT	HST3D	LOGICAL*4	EH0	HST3D	REAL*8
	ASEMBL	LOGICAL*4		APLYBC	REAL*8
	CLOSE	LOGICAL*4		ASEMBL	REAL*8
	COEFF	LOGICAL*4		DUMP	REAL*8
	DUMP	LOGICAL*4		INIT2	REAL*8
	ERROR2	LOGICAL*4		READ1	REAL*8
	ETOM1	LOGICAL*4		SUMCAL	REAL*8
	INIT1	LOGICAL*4		WBBAL	REAL*8
	INIT2	LOGICAL*4		WELLSS	REAL*8
	INIT3	LOGICAL*4		WRITE2	REAL*8
	IREWI	LOGICAL*4		WRITE5	REAL*8
	PLOTOC	LOGICAL*4			
	READ1	LOGICAL*4	EH00	HST3D	REAL*8
	READ2	LOGICAL*4		WBBAL	REAL*8
	READ3	LOGICAL*4		WELLSS	REAL*8
	REWI	LOGICAL*4		WELRIS	REAL*8
	REWI3	LOGICAL*4			
	SUMCAL	LOGICAL*4			
	WELLSS	LOGICAL*4	EHDT	HST3D	REAL*8 DIMENSION(14,10)
	WRITE1	LOGICAL*4		BLOCKDATA	REAL*8 DIMENSION(14,10)
	WRITE2	LOGICAL*4		DUMP	REAL*8 DIMENSION(14,10)
	WRITE3	LOGICAL*4		INIT2	REAL*8 DIMENSION(14,10)
	WRITE4	LOGICAL*4		READ1	REAL*8 DIMENSION(14,10)
	WRITE5	LOGICAL*4		TOFEP	REAL*8 DIMENSION(14,10)
EH	APLYBC	REAL*8 DIMENSION(*)	EHI	TOFEP	REAL*8
	ASEMBL	REAL*8 DIMENSION(*)			
	COEFF	REAL*8 DIMENSION(*)	EHIR	HST3D	REAL*8
	INIT2	REAL*8 DIMENSION(*)		SUMCAL	REAL*8
	SUMCAL	REAL*8 DIMENSION(*)		WRITE5	REAL*8
	TOFEP	REAL*8			
	WBBAL	REAL*8 DIMENSION(*)			
	WELLSS	REAL*8 DIMENSION(*)			
	WRITE2	REAL*8 DIMENSION(*)			
	WRITE5	REAL*8 DIMENSION(*)			

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
EHIRO	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
EHIRN	SUMCAL	REAL*8
EHMX	ASEMBL	REAL*8
EHMY	ASEMBL	REAL*8
EHMZ	ASEMBL	REAL*8
EHPX	ASEMBL	REAL*8
EHPY	ASEMBL	REAL*8
EHPZ	ASEMBL	REAL*8
EHST	HST3D	REAL*8 DIMENSION(32)
	BLOCKDATA	REAL*8 DIMENSION(32)
	DUMP	REAL*8 DIMENSION(32)
	INIT2	REAL*8 DIMENSION(32)
	READ1	REAL*8 DIMENSION(32)
	TOFEP	REAL*8 DIMENSION(32)
EHWEND	HST3D	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
EHWKT	ASEMBL	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
EHWSUR	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
ELBL	PLOT	CHARACTER*1
ELO	TOFEP	REAL*8
ENTH	TOFEP	REAL*8
EOD	HST3D	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
EODF	WELLSS	REAL*8
	WFDYDZ	REAL*8
EPS	TOFEP	REAL*8
EPSFAC	SOR2L	REAL*8
EPSOMG	HST3D	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	SOR2L	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
EPSSOR	HST3D	REAL*8	ERREXI	HST3D	LOGICAL*4
	DUMP	REAL*8		CLOSE	LOGICAL*4
	INIT2	REAL*8		INIT2	LOGICAL*4
	READ1	REAL*8		WRITE2	LOGICAL*4
	READ2	REAL*8	EVEN	SOR2L	LOGICAL*4
	SOR2L	REAL*8	EVMAX	SOR2L	REAL*8
	WRITE2	REAL*8	EVMIN	SOR2L	REAL*8
EPSWR	HST3D	REAL*8	EX	PLOT	CHARACTER*1
	BSODE	REAL*8	EXTRAP	BSODE	REAL*8 DIMENSION(2,11)
	DUMP	REAL*8	F10	ZONPLT	CHARACTER*20
	INIT2	REAL*8	F12	ZONPLT	CHARACTER*20
	READ1	REAL*8	F1AIF	HST3D	REAL*8
	READ2	REAL*8		APLYBC	REAL*8
	WELRIS	REAL*8		DUMP	REAL*8
ERREX	ERROR1	LOGICAL*4		INIT2	REAL*8
	ERROR2	LOGICAL*4		READ1	REAL*8
	ERROR3	LOGICAL*4	F2AIF	HST3D	REAL*8
	INTERP	LOGICAL*4		APLYBC	REAL*8
	L2SOR	LOGICAL*4		DUMP	REAL*8
	READ1	LOGICAL*4		INIT2	REAL*8
	SOR2L	LOGICAL*4		READ1	REAL*8
	VISCOS	LOGICAL*4	F6	ZONPLT	CHARACTER*20
	VSINIT	LOGICAL*4	F7	ZONPLT	CHARACTER*20
ERREXE	HST3D	LOGICAL*4	F8	ZONPLT	CHARACTER*20
	CLOSE	LOGICAL*4			
	ITER	LOGICAL*4			
	SUMCAL	LOGICAL*4			
	TOFEP	LOGICAL*4			
	WBBAL	LOGICAL*4			
	WELLSS	LOGICAL*4			
	WELRIS	LOGICAL*4			
	WFDYDZ	LOGICAL*4			
	WRITE5	LOGICAL*4			

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
FCJ	WELRIS	REAL*8
FDDP	ITER	REAL*8
FDSMTH	HST3D	REAL*8
	APLYBC	REAL*8
	ASEMBL	REAL*8
	COEFF	REAL*8
	DUMP	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	SBCFLO	REAL*8
	SUMCAL	REAL*8
	WBCFLO	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
FDTMTH	HST3D	REAL*8
	APLYBC	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	DUMP	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	SBCFLO	REAL*8
	SUMCAL	REAL*8
	WBCFLO	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
FFPHL	WFDYDZ	REAL*8
FIR	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
FIRO	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
FIRN	SUMCAL	REAL*8
FIRVO	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
FLO	ZONPLT	REAL*8
FL1	ZONPLT	REAL*8
FL2	ZONPLT	REAL*8
FL3	ZONPLT	REAL*8
FL4	ZONPLT	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
FL5	ZONPLT	REAL*8
FL6	ZONPLT	REAL*8
FLOREV	WBBAL	LOGICAL*4
	WELLSS	LOGICAL*4
FLOW	WELLSS	LOGICAL*4
FMAX	BSODE	REAL*8
FMT	ZONPLT	CHARACTER*20
FMTL	ZONPLT	CHARACTER*20
FPR3	WELLSS	REAL*8
FRAC	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
FRACN	CALCC	REAL*8
FRACNP	CALCC	REAL*8
FRESUR	HST3D	LOGICAL*4
	ASEMBL	LOGICAL*4
	CALCC	LOGICAL*4
	CLOSE	LOGICAL*4
	COEFF	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR2	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	INIT3	LOGICAL*4
	IREWI	LOGICAL*4
	PLOTOC	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	READ3	LOGICAL*4
	REWI	LOGICAL*4
	SUMCAL	LOGICAL*4
	WELLSS	LOGICAL*4
	WRITE1	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
	WRITE5	LOGICAL*4
FRFAC	WFDYDZ	REAL*8
FRFLM	WELLSS	REAL*8
FRFLP	WELLSS	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
FS	INTERP	REAL*8 DIMENSION(*,*)
FSCON	INIT2	LOGICAL*4
	SUMCAL	LOGICAL*4
FSLOW	INIT2	LOGICAL*4
	SUMCAL	LOGICAL*4
FTDAIF	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
FTXYDM	CRSDSP	REAL*8
FTXYDP	CRSDSP	REAL*8
FTXZDM	CRSDSP	REAL*8
FTXZDP	CRSDSP	REAL*8
FTYXDM	CRSDSP	REAL*8
FTYXDP	CRSDSP	REAL*8
FTYZDM	CRSDSP	REAL*8
FTYZDP	CRSDSP	REAL*8
FTZXDM	CRSDSP	REAL*8
FTZXDP	CRSDSP	REAL*8
FTZYDM	CRSDSP	REAL*8
FTZYDP	CRSDSP	REAL*8
GAMMA	APLYBC	REAL*8 DIMENSION(10)
GCOSTH	HST3D	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
GRAV	INIT2	REAL*8
GX	HST3D	REAL*8
	APLYBC	REAL*8
	ASEMBL	REAL*8
	CALCC	REAL*8
	COEFF	REAL*8
	DUMP	REAL*8
	ERROR3	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ3	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type	Variable name	Refer-encing programs	Variable type
GY	HST3D	REAL*8	HEAT	HST3D	LOGICAL*4
	APLYBC	REAL*8		APLYBC	LOGICAL*4
	ASEMBL	REAL*8		ASEMBL	LOGICAL*4
	CALCC	REAL*8		CALCC	LOGICAL*4
	COEFF	REAL*8		COEFF	LOGICAL*4
	DUMP	REAL*8		CRSDSP	LOGICAL*4
	ERROR3	REAL*8		DUMP	LOGICAL*4
	INIT2	REAL*8		ERROR2	LOGICAL*4
	READ1	REAL*8		ERROR3	LOGICAL*4
	READ3	REAL*8		ETOM1	LOGICAL*4
	SUMCAL	REAL*8		ETOM2	LOGICAL*4
	WELLSS	REAL*8		INIT1	LOGICAL*4
	WELRIS	REAL*8		INIT2	LOGICAL*4
				INIT3	LOGICAL*4
GZ	HST3D	REAL*8		ITER	LOGICAL*4
	APLYBC	REAL*8		PLOTOC	LOGICAL*4
	ASEMBL	REAL*8		READ1	LOGICAL*4
	CALCC	REAL*8		READ2	LOGICAL*4
	COEFF	REAL*8		READ3	LOGICAL*4
	DUMP	REAL*8		SUMCAL	LOGICAL*4
	ERROR3	REAL*8		VISCOS	LOGICAL*4
	INIT2	REAL*8		WBBAL	LOGICAL*4
	READ1	REAL*8		WELLSS	LOGICAL*4
	READ3	REAL*8		WRITE1	LOGICAL*4
	SUMCAL	REAL*8		WRITE2	LOGICAL*4
	WELLSS	REAL*8		WRITE3	LOGICAL*4
	WELRIS	REAL*8		WRITE4	LOGICAL*4
				WRITE5	LOGICAL*4
HDPRNT	INIT2	REAL*8 DIMENSION(*)			
	SUMCAL	REAL*8 DIMENSION(*)	HTCU	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)			
	WRITE5	REAL*8 DIMENSION(*)			

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
HTCWR	ETOM1	REAL*8 DIMENSION(*)	I	HST3D	INTEGER*4
	READ2	REAL*8 DIMENSION(*)		APLYBC	INTEGER*4
	WELRIS	REAL*8		ASEMBL	INTEGER*4
	WRITE2	REAL*8 DIMENSION(*)		BSODE	INTEGER*4
				COEFF	INTEGER*4
HWT	ERROR2	REAL*8 DIMENSION(*)		CRSDSP	INTEGER*4
	ETOM1	REAL*8 DIMENSION(*)		D4DES	INTEGER*4
	INIT2	REAL*8 DIMENSION(*)		DUMP	INTEGER*4
	READ2	REAL*8 DIMENSION(*)		ERROR1	INTEGER*4
				ERROR2	INTEGER*4
				ERROR3	INTEGER*4
				ETOM1	INTEGER*4
				INIT1	INTEGER*4
				INIT2	INTEGER*4
				INIT3	INTEGER*4
				INTERP	INTEGER*4
				IREWI	INTEGER*4
				MAP2D	INTEGER*4
				ORDER	INTEGER*4
				PLOT	INTEGER*4
				PLOTOC	INTEGER*4
				PRNTAR	INTEGER*4
				READ1	INTEGER*4
				READ2	INTEGER*4
				READ3	INTEGER*4
				REWI	INTEGER*4
				REWI3	INTEGER*4
				SUMCAL	INTEGER*4
				TOFEP	INTEGER*4
				VSINIT	INTEGER*4
				WELLSS	INTEGER*4
				WRITE2	INTEGER*4
				WRITE3	INTEGER*4
				WRITE5	INTEGER*4
				ZONPLT	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
I1	DUMP	INTEGER*4
	IREWI	INTEGER*4
	MAP2D	INTEGER*4
	ORDER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	SOR2L	INTEGER*4
	ZONPLT	INTEGER*4
I11Z	ZONPLT	INTEGER*4 DIMENSION(*)
I12X	PRNTAR	CHARACTER*4
I12Z	ZONPLT	INTEGER*4 DIMENSION(*)
I1Z	COEFF	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
I2	DUMP	INTEGER*4
	IREWI	INTEGER*4
	MAP2D	INTEGER*4
	ORDER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	SOR2L	INTEGER*4
	ZONPLT	INTEGER*4
I21Z	ZONPLT	INTEGER*4 DIMENSION(*)
I22Z	ZONPLT	INTEGER*4 DIMENSION(*)
I2Z	COEFF	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
I3	ORDER	INTEGER*4
	SOR2L	INTEGER*4
	ZONPLT	INTEGER*4
I31Z	ZONPLT	INTEGER*4 DIMENSION(*)
I32Z	ZONPLT	INTEGER*4 DIMENSION(*)
IA1HC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IA2HC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IA3HC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IA4	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IA4B	D4DES	INTEGER*4
IAA1	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IAA2	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IAA3	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IAA4	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IAAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4

Variable name	Referencing programs	Variable type
IABPM	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IAIF	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4
IALBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IALFL	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
IALFT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IAMAP	HST3D	INTEGER*4
IAPRT	HST3D	INTEGER*4
IARHC	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IARX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IARXBC	HST3D	INTEGER*4
IARY	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IARYBC	HST3D	INTEGER*4
IARZ	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IARZBC	HST3D	INTEGER*4
IAXFBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IAYFBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IAZFBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IB	SOR2L	INTEGER*4
IBAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IBBLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IBC	APLYBC	INTEGER*4 DIMENSION(*)
	ASEMBL	INTEGER*4 DIMENSION(*)
	COEFF	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	ERROR3	INTEGER*4 DIMENSION(*)
	ETOM1	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	INIT3	INTEGER*4 DIMENSION(*)
	ITER	INTEGER*4 DIMENSION(*)
	ORDER	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	SBCFLO	INTEGER*4 DIMENSION(*)
	SOR2L	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)
	WBBAL	INTEGER*4 DIMENSION(*)
	WELLSS	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE3	INTEGER*4 DIMENSION(*)
	WRITE4	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
IBCMAP	WRITE5	INTEGER*4 DIMENSION(*)
IBLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IC	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	ETOM1	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	READ1	INTEGER*4
	SOR2L	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
	ZONPLT	INTEGER*4
ICAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICALL	IREWI	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
ICC	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR2	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
ICC24	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICC34	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICC35	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICCW	HST3D	INTEGER*4
ICFLX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICH	ZONPLT	INTEGER*4
ICHWT	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR2	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
ICHYDP	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR2	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
ICI	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICMAX	HST3D	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE5	INTEGER*4
ICNP	HST3D	INTEGER*4
ICON	D4DES	INTEGER*4
ICOW	HST3D	INTEGER*4
ICQFLX	HST3D	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
ICSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ICT	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR2	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
ICWKT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
ICXM	ASEMBL	INTEGER*4
ICXP	ASEMBL	INTEGER*4
ICYM	ASEMBL	INTEGER*4
ICYP	ASEMBL	INTEGER*4
ICZM	ASEMBL	INTEGER*4
ICZP	ASEMBL	INTEGER*4
ID	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
ID4	ORDER	INTEGER*4
ID4CON	D4DES	INTEGER*4 DIMENSION(6)
ID4NO	ASEMBL	INTEGER*4 DIMENSION(*)
	D4DES	INTEGER*4 DIMENSION(*)
	ORDER	INTEGER*4 DIMENSION(*)
	SBCFLO	INTEGER*4 DIMENSION(*)
	WBCFLO	INTEGER*4 DIMENSION(*)
IDBKD	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IDC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IDEN	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IDENW	HST3D	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IDIAG	D4DES	INTEGER*4 DIMENSION(*)
	ORDER	INTEGER*4 DIMENSION(*)
IDIR	L2SOR	INTEGER*4
	ORDER	INTEGER*4
	SOR2L	INTEGER*4
IDNFBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IDNLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IDNOAR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IDNUM	PLOT	INTEGER*4
IDP	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IDPWKT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IDQHDT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IDQWDP	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IDRCON	D4DES	INTEGER*4 DIMENSION(6)
IDT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IDTHHC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IDTHWR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IE	ERRPRT	INTEGER*4
IE1	ERRPRT	INTEGER*4
IE2	ERRPRT	INTEGER*4
IEH	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IEHSUR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IEHWKT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IENDIV	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
IENDVV	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
IEQ	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	CALCC	INTEGER*4
	ITER	INTEGER*4
	L2SOR	INTEGER*4
	SBCFLO	INTEGER*4
	SOR2L	INTEGER*4
IER	IREWI	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IERR	HST3D	LOGICAL*4 DIMENSION(200)
	ERROR1	LOGICAL*4 DIMENSION(200)
	ERROR2	LOGICAL*4 DIMENSION(200)
	ERROR3	LOGICAL*4 DIMENSION(200)
	ERRPRT	LOGICAL*4 DIMENSION(200)
	INIT2	LOGICAL*4 DIMENSION(200)
	IREWI	LOGICAL*4 DIMENSION(200)
	ITER	LOGICAL*4 DIMENSION(200)
	READ1	LOGICAL*4 DIMENSION(200)
	READ2	LOGICAL*4 DIMENSION(200)
	READ3	LOGICAL*4 DIMENSION(200)
	REWI	LOGICAL*4 DIMENSION(200)
	REWI3	LOGICAL*4 DIMENSION(200)
	SOR2L	LOGICAL*4 DIMENSION(200)
	SUMCAL	LOGICAL*4 DIMENSION(200)
IF12	PRNTAR	CHARACTER*4
IFACE	IREWI	INTEGER*4
IFMT	PRNTAR	INTEGER*4
	WRITE2	INTEGER*4
IFORM	PRNTAR	CHARACTER*84
IFRAC	HST3D	INTEGER*4
IG12	PRNTAR	CHARACTER*6
IHDPRT	HST3D	INTEGER*4
IHTCR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IHWT	HST3D	INTEGER*4
II	ASEMBL	INTEGER*4
	MAP2D	INTEGER*4
II1	SOR2L	INTEGER*4
II12	PRNTAR	CHARACTER*4
II1P1	SOR2L	INTEGER*4
II1Z	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
II2Z	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
II3	SOR2L	INTEGER*4
II32	ZONPLT	INTEGER*4
II5	PRNTAR	CHARACTER*6

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
IIBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
IIBCMP	HST3D	INTEGER*4
IID4	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IIDAG	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IIFMT	PRNTAR	INTEGER*4
IINZON	HST3D	INTEGER*4
IIR2	PRNTAR	INTEGER*4
IIW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IJ1Z	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IJ2Z	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IJW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IJWEL	HST3D	INTEGER*4
IK1Z	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IK2Z	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IKARHC	HST3D	INTEGER*4
IKLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IKTAWR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IKTHX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IKTHY	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IKTHZ	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IKTWR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IKTXPM	HST3D	INTEGER*4
IKTYPM	HST3D	INTEGER*4
IKTZPM	HST3D	INTEGER*4
IKXX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IKYY	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IKZZ	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ILAVFL	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
ILBL	PLOT	INTEGER*4
ILBW	INIT1	INTEGER*4
ILCBW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
ILCTW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
ILH	D4DES	INTEGER*4
ILIVPA	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
ILNXI	WELRIS	REAL*8
ILPRT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ILVPA	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
IMAIFC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IMAP	READ3	INTEGER*4
IMAP1	HST3D	INTEGER*4 DIMENSION(3)
	ERROR3	INTEGER*4 DIMENSION(3)
	ETOM2	INTEGER*4 DIMENSION(3)
	INIT3	INTEGER*4 DIMENSION(3)
	READ3	INTEGER*4 DIMENSION(3)
	WRITE3	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)
IMAP2	HST3D	INTEGER*4 DIMENSION(3)
	ERROR3	INTEGER*4 DIMENSION(3)
	ETOM2	INTEGER*4 DIMENSION(3)
	INIT3	INTEGER*4 DIMENSION(3)
	READ3	INTEGER*4 DIMENSION(3)
	WRITE3	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)
IMAX	ORDER	INTEGER*4
IMFBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IMHCBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IMIN	ORDER	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IMLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IMM	INIT2	INTEGER*4
IMOBW	HST3D	INTEGER*4
IMOD	REWI	INTEGER*4
	REWI3	INTEGER*4 DIMENSION(3)
IMPPTC	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
IMPQW	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	ITER	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WBBAL	LOGICAL*4
	WELLSS	LOGICAL*4
	WELRIS	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE5	LOGICAL*4
IMSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4

Variable name	Referencing programs	Variable type
INCI	READ2	INTEGER*4
	READ3	INTEGER*4
INCJ	READ2	INTEGER*4
	READ3	INTEGER*4
INCJ1	SOR2L	INTEGER*4
INCJ2	SOR2L	INTEGER*4
INCJ3	SOR2L	INTEGER*4
INZONE	ERROR2	DIMENSION(*) LOGICAL*4
IOR	PRNTAR	INTEGER*4
IOUT	MAP2D	INTEGER*4
IP	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE5	INTEGER*4
IPAGE	MAP2D	INTEGER*4
IPAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type	Variable name	Refer-encing programs	Variable type
IPAR	IREWI	INTEGER*4 DIMENSION(*)	IPMHV	HST3D	INTEGER*4
				ASEMBL	INTEGER*4
IPARI	IREWI	INTEGER*4		DUMP	INTEGER*4
				INIT1	INTEGER*4
IPCS	HST3D	INTEGER*4		READ1	INTEGER*4
IPCW	HST3D	INTEGER*4	IPMZ	COEFF	INTEGER*4
				ERROR2	INTEGER*4
IPHILB	HST3D	INTEGER*4		ETOM1	INTEGER*4
	DUMP	INTEGER*4		INIT2	INTEGER*4
	INIT1	INTEGER*4		READ2	INTEGER*4
	READ1	INTEGER*4		ZONPLT	INTEGER*4
IPLOT	MAP2D	INTEGER*4 DIMENSION(*,*)	IPMZ1	ZONPLT	INTEGER*4
IPMAX	HST3D	INTEGER*4	IPMZ2	ZONPLT	INTEGER*4
	SUMCAL	INTEGER*4			
	WRITE5	INTEGER*4	IPNP	HST3D	INTEGER*4
IPMCHV	HST3D	INTEGER*4	IPOR	HST3D	INTEGER*4
	ASEMBL	INTEGER*4		DUMP	INTEGER*4
	DUMP	INTEGER*4		INIT1	INTEGER*4
	INIT1	INTEGER*4		READ1	INTEGER*4
	READ1	INTEGER*4			
			IPOS	HST3D	INTEGER*4
IPMCV	HST3D	INTEGER*4			
	ASEMBL	INTEGER*4	IPOW	HST3D	INTEGER*4
	DUMP	INTEGER*4			
	INIT1	INTEGER*4			
	READ1	INTEGER*4			
IPMCVK	HST3D	INTEGER*4			
	ASEMBL	INTEGER*4			
	DUMP	INTEGER*4			
	INIT1	INTEGER*4			
	READ1	INTEGER*4			

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IPRPTC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE4	INTEGER*4
	WRITE5	INTEGER*4
IPSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IPT	PLOTOC	INTEGER*4
IPV	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IPVK	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IPWKT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IPWKTS	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IPWSRS	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IPWSUR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IQFAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
IQFBCV	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IQFFX	HST3D	INTEGER*4
IQFFY	HST3D	INTEGER*4
IQFFZ	HST3D	INTEGER*4
IQFLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IQFSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IQHAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IQHCBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IQHFBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IQHFX	HST3D	INTEGER*4
IQHFY	HST3D	INTEGER*4
IQHFZ	HST3D	INTEGER*4
IQHLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IQHLYR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IQHSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IQHW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IQSAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IQSFBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IQSFX	HST3D	INTEGER*4
IQSFY	HST3D	INTEGER*4
IQSFZ	HST3D	INTEGER*4
IQSLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IQSLYR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IQSSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IQSW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IQWLYR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IQWM	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IQWV	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IR2	PRNTAR	INTEGER*4
IR3	PRNTAR	INTEGER*4
IR3LBL	PRNTAR	CHARACTER*1
IR3P	PRNTAR	INTEGER*4
IRBW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IRCPPM	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IRF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IRH	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IRH1	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IRHS	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IRHSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IRHSW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IRM	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IROW	D4DES	INTEGER*4
IRS	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IRS1	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ISIGN	HST3D	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
	WELRIS	INTEGER*4
ISORD	HST3D	INTEGER*4 DIMENSION(3)
	DUMP	INTEGER*4 DIMENSION(3)
	L2SOR	INTEGER*4 DIMENSION(3)
	READ1	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
ISUM	ORDER	INTEGER*4
1SXX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ISYY	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ISZZ	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ITBWR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
ITC	HST3D	INTEGER*4
ITCS	HST3D	INTEGER*4
ITCW	HST3D	INTEGER*4
ITFLX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ITFW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
ITFX	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type	Variable name	Refer-encing programs	Variable type
ITFY	HST3D	INTEGER*4	ITHY	HST3D	INTEGER*4
	ASEMBL	INTEGER*4		ASEMBL	INTEGER*4
	DUMP	INTEGER*4		DUMP	INTEGER*4
	INIT1	INTEGER*4		INIT1	INTEGER*4
	READ1	INTEGER*4		READ1	INTEGER*4
ITFZ	HST3D	INTEGER*4	ITHYX	HST3D	INTEGER*4
	ASEMBL	INTEGER*4		ASEMBL	INTEGER*4
	DUMP	INTEGER*4		DUMP	INTEGER*4
	INIT1	INTEGER*4		INIT1	INTEGER*4
	READ1	INTEGER*4		READ1	INTEGER*4
ITHCBC	HST3D	INTEGER*4	ITHYZ	HST3D	INTEGER*4
	DUMP	INTEGER*4		ASEMBL	INTEGER*4
	INIT1	INTEGER*4		DUMP	INTEGER*4
	READ1	INTEGER*4		INIT1	INTEGER*4
				READ1	INTEGER*4
ITHX	HST3D	INTEGER*4	ITHZ	HST3D	INTEGER*4
	ASEMBL	INTEGER*4		ASEMBL	INTEGER*4
	DUMP	INTEGER*4		DUMP	INTEGER*4
	INIT1	INTEGER*4		INIT1	INTEGER*4
	READ1	INTEGER*4		READ1	INTEGER*4
ITHXY	HST3D	INTEGER*4	ITHZX	HST3D	INTEGER*4
	ASEMBL	INTEGER*4		ASEMBL	INTEGER*4
	DUMP	INTEGER*4		DUMP	INTEGER*4
	INIT1	INTEGER*4		INIT1	INTEGER*4
	READ1	INTEGER*4		READ1	INTEGER*4
ITHXZ	HST3D	INTEGER*4	ITHZY	HST3D	INTEGER*4
	ASEMBL	INTEGER*4		ASEMBL	INTEGER*4
	DUMP	INTEGER*4		DUMP	INTEGER*4
	INIT1	INTEGER*4		INIT1	INTEGER*4
	READ1	INTEGER*4		READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
ITIME	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	CALCC	INTEGER*4
	CLOSE	INTEGER*4
	COEFF	INTEGER*4
	DUMP	INTEGER*4
	ERROR3	INTEGER*4
	ETOM2	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	READ3	INTEGER*4
	SUMCAL	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE4	INTEGER*4
	WRITE5	INTEGER*4
ITLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ITM1	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
ITM2	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
ITMAX	HST3D	INTEGER*4
	SUMCAL	INTEGER*4
	TOFEP	INTEGER*4
	WRITE5	INTEGER*4
ITNO	SOR2L	INTEGER*4
ITNOC	HST3D	INTEGER*4
	SOR2L	INTEGER*4
	WRITE5	INTEGER*4
ITNOP	HST3D	INTEGER*4
	SOR2L	INTEGER*4
	WRITE5	INTEGER*4
ITNOT	HST3D	INTEGER*4
	SOR2L	INTEGER*4
	WRITE5	INTEGER*4
ITNP	HST3D	INTEGER*4
ITO	HST3D	INTEGER*4
ITOS	HST3D	INTEGER*4
ITOW	HST3D	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
ITP1	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
ITP2	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
ITPHBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITQFLX	HST3D	INTEGER*4
ITRN	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	ITER	INTEGER*4
	WRITE5	INTEGER*4
ITRN1	WELLSS	INTEGER*4
ITRN2	WELLSS	INTEGER*4
ITRNDN	WELLSS	INTEGER*4
ITRNI	ITER	INTEGER*4
ITRNP	ITER	INTEGER*4
ITSBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
ITSX	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITSXY	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITSXZ	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITSY	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITSYX	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
ITSYZ	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITSZ	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITSZX	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITSZY	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITTWR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
ITWKT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
ITWSUR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
ITX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITY	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
ITZ	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IUBBLB	HST3D	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
IUCBC	HST3D	INTEGER*4
IUDNBC	HST3D	INTEGER*4
IUDNLB	HST3D	INTEGER*4
IUDTHC	HST3D	INTEGER*4
IUH	D4DES	INTEGER*4
IUKHBC	HST3D	INTEGER*4
IUKLB	HST3D	INTEGER*4
IUPHIL	HST3D	INTEGER*4
IUTBC	HST3D	INTEGER*4
IUVAIF	HST3D	INTEGER*4
IUVISL	HST3D	INTEGER*4
IUZELB	HST3D	INTEGER*4
IV	REWI3	INTEGER*4
IVA	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
IVAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IVASBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IVAW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IVIS	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IVPA	HST3D	INTEGER*4 DIMENSION(*)
	DUMP	INTEGER*4 DIMENSION(*)
	INIT1	INTEGER*4 DIMENSION(*)
	ITER	INTEGER*4 DIMENSION(*)
	L2SOR	INTEGER*4 DIMENSION(*)
	READ1	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IVSLBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IVXX	HST3D	INTEGER*4
IVYY	HST3D	INTEGER*4
IVZZ	HST3D	INTEGER*4
IW	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	ITER	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WBBAL	INTEGER*4 DIMENSION(*)
	WELLSS	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
	ZONPLT	INTEGER*4
IW1P	WRITE5	INTEGER*4
IW2P	WRITE5	INTEGER*4
IWAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IWCF	HST3D	INTEGER*4

Variable name	Referencing programs	Variable type
IWEL	ASEMBL	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	ITER	INTEGER*4
	PLOTOC	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	SUMCAL	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
	WELRIS	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
IWFICU	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IWFPCU	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IWHICU	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
IWHPCU	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IWI	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IWID	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IWOD	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IWPP	WRITE5	INTEGER*4
IWQ1	WRITE2	INTEGER*4
IWQ2	WRITE2	INTEGER*4
IWQMTH	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	SUMCAL	INTEGER*4
IWRANG	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IWRSL	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
IWRUF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
IWSICU	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
IWSPCU	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IX1M	SOR2L	INTEGER*4
IX1P	SOR2L	INTEGER*4
IX2M	SOR2L	INTEGER*4
IX2P	SOR2L	INTEGER*4
IX3M	SOR2L	INTEGER*4
IX3P	SOR2L	INTEGER*4
IXX	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
IXXN	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	L2SOR	INTEGER*4
	READ1	INTEGER*4
IY	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
IZ	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IZELBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
IZHCBC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type	Variable name	Refer-encing programs	Variable type
J	ASEMBL	INTEGER*4	J1Z	COEFF	INTEGER*4 DIMENSION(*)
	BSODE	INTEGER*4		ERROR2	INTEGER*4 DIMENSION(*)
	COEFF	INTEGER*4		INIT2	INTEGER*4 DIMENSION(*)
	CRSDSP	INTEGER*4		READ2	INTEGER*4 DIMENSION(*)
	D4DES	INTEGER*4		WRITE2	INTEGER*4 DIMENSION(*)
	DUMP	INTEGER*4			
	ERROR2	INTEGER*4	J2	D4DES	INTEGER*4
	ETOM1	INTEGER*4		IREWI	INTEGER*4
	INIT2	INTEGER*4		MAP2D	INTEGER*4
	INTERP	INTEGER*4		READ2	INTEGER*4
	IREWI	INTEGER*4		READ3	INTEGER*4
	MAP2D	INTEGER*4		REWI	INTEGER*4
	ORDER	INTEGER*4		REWI3	INTEGER*4
	READ1	INTEGER*4		ZONPLT	INTEGER*4
	READ2	INTEGER*4			
	READ3	INTEGER*4	J2Z	COEFF	INTEGER*4 DIMENSION(*)
	REWI	INTEGER*4		ERROR2	INTEGER*4 DIMENSION(*)
	REWI3	INTEGER*4		INIT2	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4		READ2	INTEGER*4 DIMENSION(*)
	VSINIT	INTEGER*4		WRITE2	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4			
	ZONPLT	INTEGER*4	J4	D4DES	INTEGER*4
J1	D4DES	INTEGER*4	JBOT	MAP2D	INTEGER*4
	IREWI	INTEGER*4			
	MAP2D	INTEGER*4	JC	ERROR2	INTEGER*4
	READ2	INTEGER*4		INIT2	INTEGER*4
	READ3	INTEGER*4		MAP2D	INTEGER*4
	REWI	INTEGER*4			
	REWI3	INTEGER*4	JCMAX	HST3D	INTEGER*4
	ZONPLT	INTEGER*4		SUMCAL	INTEGER*4
				WRITE5	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
JHVSV	BSODE	INTEGER*4
JHVSV1	BSODE	INTEGER*4
JI	ERROR2	INTEGER*4
JIFMT	PRNTAR	INTEGER*4
JINC	MAP2D	INTEGER*4
JJ	D4DES	INTEGER*4
JJ2	D4DES	INTEGER*4
JL	PLOT	INTEGER*4
JMAP1	HST3D	INTEGER*4 DIMENSION(3)
	ERROR3	INTEGER*4 DIMENSION(3)
	ETOM2	INTEGER*4 DIMENSION(3)
	INIT3	INTEGER*4 DIMENSION(3)
	READ3	INTEGER*4 DIMENSION(3)
	WRITE3	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)
JMAP2	HST3D	INTEGER*4 DIMENSION(3)
	ERROR3	INTEGER*4 DIMENSION(3)
	ETOM2	INTEGER*4 DIMENSION(3)
	INIT3	INTEGER*4 DIMENSION(3)
	READ3	INTEGER*4 DIMENSION(3)
	WRITE3	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)
JODD	BSODE	INTEGER*4
JPMAX	HST3D	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE5	INTEGER*4
JPRPTC	WRITE2	INTEGER*4
	WRITE5	INTEGER*4
JPT	PLOTOC	INTEGER*4
JSTART	BSODE	INTEGER*4
	WELRIS	INTEGER*4
JTIME	HST3D	INTEGER*4
	COEFF	INTEGER*4
	INIT3	INTEGER*4
JTMAX	HST3D	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE5	INTEGER*4
JTOP	MAP2D	INTEGER*4
JW	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	ITER	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WBBAL	INTEGER*4 DIMENSION(*)
	WELLSS	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
JWELL	WELLSS	INTEGER*4 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
K	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	BSODE	INTEGER*4
	CALCC	INTEGER*4
	COEFF	INTEGER*4
	CRSDSP	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	IREWI	INTEGER*4
	ITER	INTEGER*4
	MAP2D	INTEGER*4
	ORDER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	SUMCAL	INTEGER*4
	VSINIT	INTEGER*4
	WBBAL	INTEGER*4
	WBCFLO	INTEGER*4
	WELLSS	INTEGER*4
	WELRIS	INTEGER*4
	WRITE2	INTEGER*4
	WRITE5	INTEGER*4
K1	INIT2	INTEGER*4
	IREWI	INTEGER*4
	MAP2D	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	WRITE5	INTEGER*4
K1Z	COEFF	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
K2	IREWI	INTEGER*4
	MAP2D	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
K2Z	COEFF	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
KARHC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
KC	PLOT	INTEGER*4
KCMAX	HST3D	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE5	INTEGER*4
KF	INIT2	INTEGER*4
KFLAG	BSODE	INTEGER*4
	WELRIS	INTEGER*4
KINC	INIT2	INTEGER*4
KK	MAP2D	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
KL	INIT2	INTEGER*4
KLBC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
KMAP1	HST3D	INTEGER*4 DIMENSION(3)
	ERROR3	INTEGER*4 DIMENSION(3)
	ETOM2	INTEGER*4 DIMENSION(3)
	INIT3	INTEGER*4 DIMENSION(3)
	READ3	INTEGER*4 DIMENSION(3)
	WRITE3	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)
KMAP2	HST3D	INTEGER*4 DIMENSION(3)
	ERROR3	INTEGER*4 DIMENSION(3)
	ETOM2	INTEGER*4 DIMENSION(3)
	INIT3	INTEGER*4 DIMENSION(3)
	READ3	INTEGER*4 DIMENSION(3)
	WRITE3	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)
KMAX	WELRIS	INTEGER*4
KO	PLOT	INTEGER*4
KOAR	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8

Variable name	Referencing programs	Variable type
KPMAX	HST3D	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE5	INTEGER*4
KTHAWR	ETOM1	INTEGER*4 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
KTHF	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
KTHWR	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
KTHX	COEFF	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
KTHXPM	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
KTHY	COEFF	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
KTHYPM	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
KTHZ	COEFF	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
KTHZPM	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
KTMAX	HST3D	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE5	INTEGER*4
KWEL	PLOTOC	INTEGER*4
KXX	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
KYY	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
KZZ	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)

Variable name	Referencing programs	Variable type
L	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	BSODE	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	ORDER	INTEGER*4
	PLOT	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	SBCFLO	INTEGER*4
	SUMCAL	INTEGER*4
	WBCFLO	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
	ZONPLT	INTEGER*4
L1	ERROR2	INTEGER*4
	INIT2	INTEGER*4
	ZONPLT	INTEGER*4
L2	ERROR3	INTEGER*4
	INIT2	INTEGER*4
	SOR2L	INTEGER*4
	ZONPLT	INTEGER*4
L3	D4DES	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
LABEL	REWI	CHARACTER*13 DIMENSION(5)
	REWI3	CHARACTER*10 DIMENSION(3)
	WRITE2	CHARACTER*20
	ZONPLT	CHARACTER*4
LBLDIR	SOR2L	CHARACTER*1 DIMENSION(3)
	WRITE5	CHARACTER*1 DIMENSION(3)
LBLEQ	L2SOR	CHARACTER*10 DIMENSION(3)
	SOR2L	CHARACTER*10 DIMENSION(3)
LBW	ORDER	INTEGER*4 DIMENSION(*)
LCBOTW	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	ITER	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WBBAL	INTEGER*4 DIMENSION(*)
	WBCFLO	INTEGER*4 DIMENSION(*)
	WELLSS	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)

Variable name	Referencing programs	Variable type
LCROSD	HST3D	LOGICAL*4
	ASEMBL	LOGICAL*4
	COEFF	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR1	LOGICAL*4
	INIT1	LOGICAL*4
	INIT2	LOGICAL*4
	ITER	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	SBCFLO	LOGICAL*4
	WBCFLO	LOGICAL*4
	WRITE1	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE4	LOGICAL*4
	WRITE5	LOGICAL*4
LCTOPW	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	ITER	INTEGER*4 DIMENSION(*)
	READ2	INTEGER*4 DIMENSION(*)
	WBBAL	INTEGER*4 DIMENSION(*)
	WBCFLO	INTEGER*4 DIMENSION(*)
	WELLSS	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
LDASH	WRITE5	CHARACTER*150
LDOTS	CLOSE	CHARACTER*130
	REWI	CHARACTER*130
	REWI3	CHARACTER*120
	WRITE2	CHARACTER*130
	WRITE3	CHARACTER*130
	WRITE5	CHARACTER*150
LENAX	HST3D	REAL*8
	MAP2D	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
	WRITE5	REAL*8
	ZONPLT	REAL*8
LENAY	HST3D	REAL*8
	MAP2D	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
	WRITE5	REAL*8
LENAZ	HST3D	REAL*8
	READ3	REAL*8
	WRITE3	REAL*8
	WRITE5	REAL*8
LGREN	WELLSS	REAL*8
	WFDYDZ	REAL*8
LIMAGE	READ1	CHARACTER*80
LIMIT	WRITE3	CHARACTER*4
LINE	PLOT	CHARACTER*101

Variable name	Referencing programs	Variable type
LINLIM	PLOT	INTEGER*4
LL	APLYBC	INTEGER*4
	INIT2	INTEGER*4
LLL	APLYBC	INTEGER*4
LN	MAP2D	INTEGER*4
LOCAIF	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4
LPRNT	PRNTAR	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE3	INTEGER*4 DIMENSION(*)
	WRITE4	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
LRBC	ETOM1	INTEGER*4
LTD	APLYBC	REAL*8
LXL	MAP2D	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
LXR	MAP2D	INTEGER*4
M	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	BSODE	INTEGER*4
	COEFF	INTEGER*4
	CRSDSP	INTEGER*4
	D4DES	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	ETOM1	INTEGER*4
	ETOM2	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	IREWI	INTEGER*4
	ITER	INTEGER*4
	L2SOR	INTEGER*4
	ORDER	INTEGER*4
	PRNTAR	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	SBCFLO	INTEGER*4
	SOR2L	INTEGER*4
	SUMCAL	INTEGER*4
	WBBAL	INTEGER*4
	WBCFLO	INTEGER*4
	WELLSS	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE4	INTEGER*4
	WRITE5	INTEGER*4
M1	ORDER	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	SOR2L	INTEGER*4
	WELLSS	INTEGER*4
M2	BSODE	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	SOR2L	INTEGER*4
M3	SUMCAL	INTEGER*4
MA	ASEMBL	INTEGER*4
	SBCFLO	INTEGER*4
	WBCFLO	INTEGER*4
MAIFC	APLYBC	INTEGER*4 DIMENSION(*)
	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR3	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	INIT3	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
MAPPTC	HST3D	INTEGER*4
	ERROR3	INTEGER*4
	READ3	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
MAXDXX	SOR2L	REAL*8
MAXIT	SOR2L	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
MAXIT1	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	SOR2L	INTEGER*4
MAXIT2	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	SOR2L	INTEGER*4
MAXITN	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4
MAXORD	HST3D	INTEGER*4
	BSODE	INTEGER*4
	WELRIS	INTEGER*4
MAXPTS	HST3D	INTEGER*4
	BSODE	INTEGER*4
	WELRIS	INTEGER*4
MAXXX	SOR2L	REAL*8
MD4	D4DES	INTEGER*4
	ORDER	INTEGER*4

Variable name	Referencing programs	Variable type
METH	HST3D	INTEGER*4
	BSODE	INTEGER*4
	WELRIS	INTEGER*4
MFBC	APLYBC	INTEGER*4 DIMENSION(*)
	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR3	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	INIT3	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE3	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
MFLBL	PLOTOC	CHARACTER*12
	WRITE2	CHARACTER*12
	WRITE3	CHARACTER*12
	WRITE5	CHARACTER*12
MHCBC	APLYBC	INTEGER*4 DIMENSION(*)
	ASEMBL	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
MIJKM	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	COEFF	INTEGER*4
	CRSDSP	INTEGER*4
	INIT2	INTEGER*4
	SBCFLO	INTEGER*4
	WBBAL	INTEGER*4
	WBCFLO	INTEGER*4
	WELLSS	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
MIJKP	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	COEFF	INTEGER*4
	CRSDSP	INTEGER*4
	SBCFLO	INTEGER*4
	WBBAL	INTEGER*4
	WBCFLO	INTEGER*4
	WELLSS	INTEGER*4
MIJMK	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	COEFF	INTEGER*4
	CRSDSP	INTEGER*4
	INIT2	INTEGER*4
	SBCFLO	INTEGER*4
	WBCFLO	INTEGER*4
MIJMKM	CRSDSP	INTEGER*4
MIJMKP	CRSDSP	INTEGER*4
MIJPK	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	COEFF	INTEGER*4
	CRSDSP	INTEGER*4
	SBCFLO	INTEGER*4
	WBCFLO	INTEGER*4
MIJPKM	CRSDSP	INTEGER*4
MIJPKP	CRSDSP	INTEGER*4
MIMJK	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	CRSDSP	INTEGER*4
	SBCFLO	INTEGER*4
	WBCFLO	INTEGER*4
MIMJKM	CRSDSP	INTEGER*4
MIMJKP	CRSDSP	INTEGER*4
MIMJMK	CRSDSP	INTEGER*4
MIMJPK	CRSDSP	INTEGER*4
MINUS	PLOT	CHARACTER*1
MIPJK	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	CRSDSP	INTEGER*4
	SBCFLO	INTEGER*4
	WBCFLO	INTEGER*4
MIPJKM	CRSDSP	INTEGER*4
MIPJKP	CRSDSP	INTEGER*4
MIPJMK	CRSDSP	INTEGER*4
MIPJPK	CRSDSP	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
MK	INIT2	INTEGER*4
MKT	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
	WRITE5	INTEGER*4
ML	INIT2	INTEGER*4
MLBC	APLYBC	INTEGER*4 DIMENSION(*)
	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	INIT3	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE3	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
MM	ASEMBL	INTEGER*4
	INIT2	INTEGER*4 DIMENSION(8)
	ITER	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
	WRITE5	INTEGER*4
MNEXT	BSODE	INTEGER*4
MOBW	WELLSS	REAL*8 DIMENSION(*)
MSBC	ASEMBL	INTEGER*4 DIMENSION(*)
	INIT2	INTEGER*4 DIMENSION(*)
	INIT3	INTEGER*4 DIMENSION(*)
	SBCFLO	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE3	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)

Variable name	Referencing programs	Variable type
MT	COEFF	INTEGER*4
	INIT2	INTEGER*4
	SUMCAL	INTEGER*4
MTJP1	COEFF	INTEGER*4
MTWO	BSODE	INTEGER*4
MWEL	INIT2	INTEGER*4
MXITQW	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WELLSS	INTEGER*4
	WRITE2	INTEGER*4
N	MAP2D	INTEGER*4
N1	PRNTAR	INTEGER*4
	ZONPLT	INTEGER*4
N2	PRNTAR	INTEGER*4
	ZONPLT	INTEGER*4
N3	PRNTAR	INTEGER*4
	ZONPLT	INTEGER*4
NA	MAP2D	INTEGER*4
NA1	PRNTAR	CHARACTER*6
NA2	PRNTAR	CHARACTER*2
NAC	ERROR2	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
NAIFC	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	ETOM1	INTEGER*4
	ETOM2	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE1	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
NC	PLOT	INTEGER*4
	PLOTOC	INTEGER*4
NCHARS	MAP2D	INTEGER*4
NCHP	MAP2D	INTEGER*4
	ZONPLT	INTEGER*4
NCHPL	MAP2D	INTEGER*4
NCHPR	MAP2D	INTEGER*4
NCPR	ZONPLT	INTEGER*4

Variable name	Refer-encing programs	Variable type
NDELA	MAP2D	INTEGER*4
NDIM	INTERP	INTEGER*4
	PRNTAR	INTEGER*4
NEHST	HST3D	INTEGER*4
	BLOCKDATA	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	TOFEP	INTEGER*4
NFBC	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	ETOM1	INTEGER*4
	ETOM2	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE1	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
NFC	ERROR2	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
NGRIDX	MAP2D	INTEGER*4 DIMENSION(50)	NHCN	HST3D	INTEGER*4
				APLYBC	INTEGER*4
NGRIDY	MAP2D	INTEGER*4 DIMENSION(50)		DUMP	INTEGER*4
				ERROR1	INTEGER*4
NHC	ERROR2	INTEGER*4		ERROR2	INTEGER*4
				ETOM1	INTEGER*4
NHCBC	HST3D	INTEGER*4		INIT1	INTEGER*4
	APLYBC	INTEGER*4		INIT2	INTEGER*4
	ASEMBL	INTEGER*4		READ1	INTEGER*4
	DUMP	INTEGER*4		READ2	INTEGER*4
	ERROR2	INTEGER*4		WRITE1	INTEGER*4
	ERROR3	INTEGER*4		WRITE2	INTEGER*4
	ETOM1	INTEGER*4			
	ETOM2	INTEGER*4	NL	ZONPLT	INTEGER*4
	INIT1	INTEGER*4			
	INIT2	INTEGER*4			
	INIT3	INTEGER*4			
	ITER	INTEGER*4			
	READ1	INTEGER*4			
	READ2	INTEGER*4			
	READ3	INTEGER*4			
	SUMCAL	INTEGER*4			
	WRITE1	INTEGER*4			
	WRITE2	INTEGER*4			
	WRITE3	INTEGER*4			
	WRITE5	INTEGER*4			

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
NLBC	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	ETOM1	INTEGER*4
	ETOM2	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE1	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
NLC	ERROR2	INTEGER*4
NLP	PLOT	INTEGER*4
	ZONPLT	INTEGER*4
NMAPR	HST3D	INTEGER*4
	CLOSE	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
NMPZON	HST3D	INTEGER*4 DIMENSION(3)
	INIT3	INTEGER*4 DIMENSION(3)
	MAP2D	INTEGER*4
	READ3	INTEGER*4 DIMENSION(3)
	WRITE3	INTEGER*4 DIMENSION(3)
	WRITE5	INTEGER*4 DIMENSION(3)
NN3	ZONPLT	INTEGER*4
NNC	D4DES	INTEGER*4
NNOPPR	PRNTAR	INTEGER*4
NNOUT	MAP2D	INTEGER*4
NNPR	PRNTAR	INTEGER*4
NO	PLOT	INTEGER*4
	PLOTOC	INTEGER*4
NOCV	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	VISCOS	INTEGER*4
	WRITE2	INTEGER*4
NOTV	VSINIT	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
NOTVO	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	VISCOS	INTEGER*4
	WRITE2	INTEGER*4
NOTV1	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	VISCOS	INTEGER*4
	WRITE2	INTEGER*4
NP	ZONPLT	INTEGER*4
NPAGES	MAP2D	INTEGER*4
NPAIF	APLYBC	INTEGER*4
NPCX	ZONPLT	INTEGER*4
NPCY	ZONPLT	INTEGER*4
NPEHDT	HST3D	INTEGER*4
	BLOCKDATA	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	TOFEP	INTEGER*4
NPMZ	HST3D	INTEGER*4
	COEFF	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE1	INTEGER*4
	WRITE2	INTEGER*4
	ZONPLT	INTEGER*4
NPOSNS	MAP2D	INTEGER*4
NPR2	PRNTAR	INTEGER*4
NPR3	PRNTAR	INTEGER*4
NPTAIF	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer- encing programs	Variable type
NPTCBC	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	ERROR2	INTEGER*4
	ERROR3	INTEGER*4
	ETOM1	INTEGER*4
	ETOM2	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	SBCFLO	INTEGER*4
	SUMCAL	INTEGER*4
	WRITE1	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4
NPTSA4	HST3D	INTEGER*4
	D4DES	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ORDER	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
NPTSD4	HST3D	INTEGER*4
	D4DES	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ORDER	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
NPTSUH	HST3D	INTEGER*4
	ASEMBL	INTEGER*4
	D4DES	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	ORDER	INTEGER*4
	READ1	INTEGER*4
	SBCFLO	INTEGER*4
	WBCFLO	INTEGER*4
	WRITE1	INTEGER*4
NR	ERROR2	INTEGER*4
	INIT2	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4
NROWS	MAP2D	INTEGER*4
NRP	ZONPLT	INTEGER*4
NRPR	ZONPLT	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
NRSTTP	HST3D	INTEGER*4
	CLOSE	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
NS	ZONPLT	INTEGER*4
NSC	ERROR2	INTEGER*4
NSHUT	WELLSS	INTEGER*4
NSTD4	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
NSTSOR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT1	INTEGER*4
	READ1	INTEGER*4
	WRITE1	INTEGER*4
NTEHDT	HST3D	INTEGER*4
	BLOCKDATA	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	TOFEP	INTEGER*4
NTHPTC	PLOTOC	INTEGER*4
NTHPTO	PLOTOC	INTEGER*4
NTSCHK	HST3D	INTEGER*4
	CLOSE	INTEGER*4
	DUMP	INTEGER*4
	READ3	INTEGER*4
NTSOPT	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	ITER	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	SOR2L	INTEGER*4
	WRITE2	INTEGER*4
NVST	VSINIT	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
NWEL	HST3D	INTEGER*4	NX	HST3D	INTEGER*4
	ASEMBL	INTEGER*4		APLYBC	INTEGER*4
	DUMP	INTEGER*4		ASEMBL	INTEGER*4
	ERROR2	INTEGER*4		COEFF	INTEGER*4
	ERROR3	INTEGER*4		CRSDSP	INTEGER*4
	ETOM1	INTEGER*4		D4DES	INTEGER*4
	ETOM2	INTEGER*4		DUMP	INTEGER*4
	INIT1	INTEGER*4		ERROR1	INTEGER*4
	INIT2	INTEGER*4		ERROR2	INTEGER*4
	INIT3	INTEGER*4		ETOM1	INTEGER*4
	ITER	INTEGER*4		ETOM2	INTEGER*4
	READ1	INTEGER*4		INIT1	INTEGER*4
	READ2	INTEGER*4		INIT2	INTEGER*4
	READ3	INTEGER*4		INIT3	INTEGER*4
	SUMCAL	INTEGER*4		INTERP	INTEGER*4
	WBBAL	INTEGER*4		IREWI	INTEGER*4
	WELLSS	INTEGER*4		ITER	INTEGER*4
	WRITE1	INTEGER*4		MAP2D	INTEGER*4
	WRITE2	INTEGER*4		ORDER	INTEGER*4
	WRITE3	INTEGER*4		PLOTOC	INTEGER*4
	WRITE5	INTEGER*4		PRNTAR	INTEGER*4
				READ1	INTEGER*4
				READ2	INTEGER*4
				READ3	INTEGER*4
				REWI	INTEGER*4
				REWI3	INTEGER*4
				SBCFLO	INTEGER*4
				SOR2L	INTEGER*4
				SUMCAL	INTEGER*4
				WBBAL	INTEGER*4
				WBCFLO	INTEGER*4
				WELLSS	INTEGER*4
				WRITE1	INTEGER*4
				WRITE2	INTEGER*4
				WRITE3	INTEGER*4
				WRITE4	INTEGER*4
				WRITE5	INTEGER*4
				ZONPLT	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
NX1	ORDER	INTEGER*4	NXY	HST3D	INTEGER*4
	SOR2L	INTEGER*4		APLYBC	INTEGER*4
	ZONPLT	INTEGER*4		ASEMBL	INTEGER*4
				COEFF	INTEGER*4
NX2	ORDER	INTEGER*4		CRSDSP	INTEGER*4
	SOR2L	INTEGER*4		D4DES	INTEGER*4
	ZONPLT	INTEGER*4		DUMP	INTEGER*4
				ERROR1	INTEGER*4
NX3	ORDER	INTEGER*4		ERROR2	INTEGER*4
	SOR2L	INTEGER*4		ETOM1	INTEGER*4
				ETOM2	INTEGER*4
NXPR	PRNTAR	INTEGER*4		INIT1	INTEGER*4
				INIT2	INTEGER*4
NXX	INIT2	INTEGER*4		INIT3	INTEGER*4
	TOFEP	INTEGER*4		IREWI	INTEGER*4
	VISCOS	INTEGER*4		ITER	INTEGER*4
				ORDER	INTEGER*4
				PLOTOC	INTEGER*4
				PRNTAR	INTEGER*4
				READ1	INTEGER*4
				READ2	INTEGER*4
				READ3	INTEGER*4
				REWI	INTEGER*4
				REWI3	INTEGER*4
				SBCFLO	INTEGER*4
				SOR2L	INTEGER*4
				SUMCAL	INTEGER*4
				WBBAL	INTEGER*4
				WBCFLO	INTEGER*4
				WELLSS	INTEGER*4
				WRITE1	INTEGER*4
				WRITE2	INTEGER*4
				WRITE3	INTEGER*4
				WRITE4	INTEGER*4
				WRITE5	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
NXYZ	HST3D	INTEGER*4	NY	HST3D	INTEGER*4
	APLYBC	INTEGER*4		APLYBC	INTEGER*4
	ASEMBL	INTEGER*4		ASEMBL	INTEGER*4
	COEFF	INTEGER*4		COEFF	INTEGER*4
	D4DES	INTEGER*4		DUMP	INTEGER*4
	DUMP	INTEGER*4		ERROR1	INTEGER*4
	ERROR1	INTEGER*4		ERROR2	INTEGER*4
	ERROR2	INTEGER*4		ETOM1	INTEGER*4
	ETOM1	INTEGER*4		INIT1	INTEGER*4
	ETOM2	INTEGER*4		INIT2	INTEGER*4
	INIT1	INTEGER*4		INIT3	INTEGER*4
	INIT2	INTEGER*4		INTERP	INTEGER*4
	INIT3	INTEGER*4		IREWI	INTEGER*4
	IREWI	INTEGER*4		ITER	INTEGER*4
	ITER	INTEGER*4		MAP2D	INTEGER*4
	L2SOR	INTEGER*4		ORDER	INTEGER*4
	ORDER	INTEGER*4		PRNTAR	INTEGER*4
	PLOTOC	INTEGER*4		READ1	INTEGER*4
	PRNTAR	INTEGER*4		READ2	INTEGER*4
	READ1	INTEGER*4		READ3	INTEGER*4
	READ2	INTEGER*4		REWI	INTEGER*4
	READ3	INTEGER*4		REWI3	INTEGER*4
	REWI	INTEGER*4		SOR2L	INTEGER*4
	REWI3	INTEGER*4		SUMCAL	INTEGER*4
	SOR2L	INTEGER*4		WBBAL	INTEGER*4
	SUMCAL	INTEGER*4		WELLSS	INTEGER*4
	WBBAL	INTEGER*4		WRITE1	INTEGER*4
	WELLSS	INTEGER*4		WRITE2	INTEGER*4
	WRITE1	INTEGER*4		WRITE5	INTEGER*4
	WRITE2	INTEGER*4		ZONPLT	INTEGER*4
	WRITE3	INTEGER*4			
	WRITE4	INTEGER*4	NY1	ZONPLT	INTEGER*4
	WRITE5	INTEGER*4			
			NY2	ZONPLT	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
NYPR	PRNTAR	INTEGER*4
NZ	HST3D	INTEGER*4
	APLYBC	INTEGER*4
	ASEMBL	INTEGER*4
	CALCC	INTEGER*4
	COEFF	INTEGER*4
	DUMP	INTEGER*4
	ERROR1	INTEGER*4
	ERROR2	INTEGER*4
	ETOM1	INTEGER*4
	INIT1	INTEGER*4
	INIT2	INTEGER*4
	INIT3	INTEGER*4
	IREWI	INTEGER*4
	ITER	INTEGER*4
	ORDER	INTEGER*4
	PRNTAR	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	READ3	INTEGER*4
	REWI	INTEGER*4
	REWI3	INTEGER*4
	SOR2L	INTEGER*4
	SUMCAL	INTEGER*4
	WBBAL	INTEGER*4
	WELLSS	INTEGER*4
	WRITE1	INTEGER*4
	WRITE2	INTEGER*4
	WRITE5	INTEGER*4
NZPR	PRNTAR	INTEGER*4
NZTPHC	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4
NZTPRO	HST3D	INTEGER*4
	DUMP	INTEGER*4
	ETOM1	INTEGER*4
	INIT2	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4
OCPLOT	HST3D	LOGICAL*4
	CLOSE	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR3	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	READ3	LOGICAL*4
	SOR2L	LOGICAL*4
	WELLSS	LOGICAL*4
	WELRIS	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
	WRITE5	LOGICAL*4
ODD	ORDER	LOGICAL*4
OH	PLOT	CHARACTER*1

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
OMEGA	L2SOR	REAL*8
	SOR2L	REAL*8
OMGMAX	SOR2L	REAL*8
OMGMIN	SOR2L	REAL*8
OMOPT	HST3D	REAL*8 DIMENSION(3)
	DUMP	REAL*8 DIMENSION(3)
	L2SOR	REAL*8 DIMENSION(3)
	READ1	REAL*8 DIMENSION(3)
ORENPR	HST3D	INTEGER*4
	DUMP	INTEGER*4
	INIT2	INTEGER*4
	PRNTAR	INTEGER*4
	READ1	INTEGER*4
	READ2	INTEGER*4
	WRITE2	INTEGER*4
	WRITE3	INTEGER*4
	WRITE4	INTEGER*4
	WRITE5	INTEGER*4
ORFR	WRITE5	INTEGER*4
P	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	CALCC	REAL*8
	COEFF	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	TOFEP	REAL*8
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
	ZONPLT	CHARACTER*10000
P0	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WFDYDZ	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
P00	HST3D	REAL*8	PAATM	HST3D	REAL*8
	WBBAL	REAL*8		APLYBC	REAL*8
	WELLSS	REAL*8		ASEMBL	REAL*8
	WELRIS	REAL*8		DUMP	REAL*8
				ETOM1	REAL*8
P0H	HST3D	REAL*8		INIT2	REAL*8
	APLYBC	REAL*8		READ1	REAL*8
	ASEMBL	REAL*8		READ2	REAL*8
	DUMP	REAL*8		SUMCAL	REAL*8
	ETOM1	REAL*8		TOFEP	REAL*8
	INIT2	REAL*8		WBBAL	REAL*8
	READ1	REAL*8		WELLSS	REAL*8
	READ2	REAL*8		WELRIS	REAL*8
	SUMCAL	REAL*8		WRITE2	REAL*8
	WELLSS	REAL*8		WRITE5	REAL*8
	WELRIS	REAL*8			
	WRITE2	REAL*8	PAEHDT	HST3D	REAL*8 DIMENSION(10)
				BLOCKDATA	REAL*8 DIMENSION(10)
P1	PRNTAR	CHARACTER*2		DUMP	REAL*8 DIMENSION(10)
				INIT2	REAL*8 DIMENSION(10)
P2	PRNTAR	CHARACTER*2		READ1	REAL*8 DIMENSION(10)
				TOFEP	REAL*8 DIMENSION(10)
P4	PRNTAR	CHARACTER*2			
			PAIF	APLYBC	REAL*8 DIMENSION(*)
P5	PRNTAR	CHARACTER*2		INIT2	REAL*8 DIMENSION(*)
			PAR	REWI	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
PARA	REWI3	REAL*8 DIMENSION(*)
PARB	REWI3	REAL*8 DIMENSION(*)
PARC	REWI3	REAL*8 DIMENSION(*)
PCL	WELLSS	REAL*8
PCR	WELLSS	REAL*8
PCS	PLOTOC	REAL*8 DIMENSION(*)
PCW	PLOTOC	REAL*8 DIMENSION(*)
PFAC	MAP2D	REAL*8
PFSLOW	SUMCAL	LOGICAL*4
PHILBC	APLYBC	REAL*8 DIMENSION(*)
	ERROR3	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
PI	APLYBC	REAL*8
	INIT2	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
PINIT	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
PLBL	WRITE5	CHARACTER*20
PLOTWC	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	PLOTOC	LOGICAL*4
	READ3	LOGICAL*4
PLOTWP	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	PLOTOC	LOGICAL*4
	READ3	LOGICAL*4
PLOTWT	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	PLOTOC	LOGICAL*4
	READ3	LOGICAL*4
PLTZON	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
PLUS	PLOT	CHARACTER*1
PMCHDT	CALCC	REAL*8
PMCHV	CALCC	REAL*8
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
PMCV	CALCC	REAL*8
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
PMCVDT	CALCC	REAL*8
PMHV	CALCC	REAL*8
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
PMHVDT	CALCC	REAL*8
PNP	CALCC	REAL*8
	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
POROAR	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
POROS	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
POS	PLOTOC	REAL*8 DIMENSION(*)
POSUP	MAP2D	LOGICAL*4
POW	PLOTOC	REAL*8 DIMENSION(*)
PPCL	WELLSS	REAL*8
PPCR	WELLSS	REAL*8
PRBCF	HST3D	LOGICAL*4
	SUMCAL	LOGICAL*4
	WRITE5	LOGICAL*4
PRDV	HST3D	LOGICAL*4
	SUMCAL	LOGICAL*4
	WRITE5	LOGICAL*4
PRESS	TOFEP	REAL*8
PRGFB	HST3D	LOGICAL*4
	SUMCAL	LOGICAL*4
	WRITE5	LOGICAL*4
PRIBCF	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE5	INTEGER*4
PRIDV	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE5	INTEGER*4
PRIGFB	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE5	INTEGER*4
PRIKD	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE4	INTEGER*4
	WRITE5	INTEGER*4
PRIMAP	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE3	INTEGER*4
	WRITE5	INTEGER*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
PRIPTC	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE5	INTEGER*4
PRISLM	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE5	INTEGER*4
PRIVEL	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE4	INTEGER*4
	WRITE5	INTEGER*4
PRIWEL	HST3D	INTEGER*4
	READ3	INTEGER*4
	WRITE5	INTEGER*4
PRKD	WRITE4	LOGICAL*4
PRNT	WRITE3	LOGICAL*4
PRPTC	HST3D	LOGICAL*4
	SUMCAL	LOGICAL*4
	WRITE5	LOGICAL*4
PRSLM	HST3D	LOGICAL*4
	SUMCAL	LOGICAL*4
	WRITE5	LOGICAL*4
PRTBC	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	READ3	LOGICAL*4
	SOR2L	LOGICAL*4
	WELLSS	LOGICAL*4
	WELRIS	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
PRTCCM	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	SUMCAL	LOGICAL*4
PRTCHR	MAP2D	CHARACTER*125
PRTDV	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
PRTFP	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
PRTIC	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
PRTMPD	HST3D	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
	WRITE5	LOGICAL*4
PRTPMP	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
PRTRE	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	IREWI	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	REWI	LOGICAL*4
	REWI3	LOGICAL*4
	SUMCAL	LOGICAL*4
PRTSLM	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	READ3	LOGICAL*4
	SOR2L	LOGICAL*4
	WELLSS	LOGICAL*4
	WELRIS	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
PRTWEL	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	READ3	LOGICAL*4
	WELLSS	LOGICAL*4
	WELRIS	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
PRVEL	WRITE4	LOGICAL*4
PRWEL	HST3D	LOGICAL*4
	SUMCAL	LOGICAL*4
	WRITE5	LOGICAL*4
PSBC	ASEMBL	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
PSLBL	PLOTOC	CHARACTER*50

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
PSMAX	PLOTOC	REAL*8
PSMIN	PLOTOC	REAL*8
PTOP	WELLSS	REAL*8
PU	APLYBC	REAL*8
PV	CALCC	REAL*8
	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
PVDTN	CALCC	REAL*8
PVK	CALCC	REAL*8
	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
PVKDTN	CALCC	REAL*8
PWCELL	WRITE5	REAL*8
PWKT	ASEMBL	REAL*8 DIMENSION(*)
	ERROR3	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
PWKTS	ITER	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
PWLBL	PLOTOC	CHARACTER*50
PWMAX	PLOTOC	REAL*8
PWMIN	PLOTOC	REAL*8
PWREND	HST3D	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
PWRK	WFDYDZ	REAL*8
PWSUR	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
PWSURS	ERROR3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
QDVSBC	SBCFLO	REAL*8 DIMENSION(*)
QFAC	MAP2D	REAL*8
QFAIF	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QFBC	APLYBC	REAL*8
	ASEMBL	REAL*8
	SUMCAL	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
QFBCV	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	ERROR3	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QFFX	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QFFY	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QFFZ	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QFLBC	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QFSBC	ASEMBL	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QHAIF	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QHBC	APLYBC	REAL*8
	ASEMBL	REAL*8
QHCBC	APLYBC	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QHFAC	HST3D	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
QHFBC	APLYBC	REAL*8 DIMENSION(*)
	ERROR3	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QHFX	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QHFY	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QHFZ	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QHLBC	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QHLYR	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
QHSBC	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QHW	ASEMBL	REAL*8
	SUMCAL	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QHWRK	WFDYDZ	REAL*8
QLIM	APLYBC	REAL*8
	ASEMBL	REAL*8
	SUMCAL	REAL*8
QN	APLYBC	REAL*8
	ASEMBL	REAL*8
	SUMCAL	REAL*8
QNP	ASEMBL	REAL*8
	SUMCAL	REAL*8
QSAIF	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QSBC	APLYBC	REAL*8
	ASEMBL	REAL*8
QSFBC	APLYBC	REAL*8 DIMENSION(*)
	ERROR3	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QSFX	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QSFY	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QSFZ	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
QSLBC	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QSLYR	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QSSBC	SUMCAL	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QSW	ASEMBL	REAL*8
	SUMCAL	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
QUOT	BSODE	REAL*8 DIMENSION(11,2)
QUOTSV	BSODE	REAL*8
QWAV	ASEMBL	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
QWLYR	ASEMBL	REAL*8 DIMENSION(*)	RBW	D4DES	INTEGER*4 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)		ORDER	INTEGER*4 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)			
	WBCFLO	REAL*8 DIMENSION(*)	RCPPM	ETOM1	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)		INIT2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)		READ2	REAL*8 DIMENSION(*)
				WRITE2	REAL*8 DIMENSION(*)
QWM	ASEMBL	REAL*8 DIMENSION(*)			
	ITER	REAL*8 DIMENSION(*)	RDAIF	HST3D	LOGICAL*4
	SUMCAL	REAL*8 DIMENSION(*)		ERROR3	LOGICAL*4
	WBBAL	REAL*8 DIMENSION(*)		ETOM2	LOGICAL*4
	WELLSS	REAL*8 DIMENSION(*)		INIT3	LOGICAL*4
	WELRIS	REAL*8		READ3	LOGICAL*4
	WRITE5	REAL*8 DIMENSION(*)		WRITE3	LOGICAL*4
QWN	ASEMBL	REAL*8	RDCALC	HST3D	LOGICAL*4
				ERROR3	LOGICAL*4
QWNP	ASEMBL	REAL*8		ETOM2	LOGICAL*4
				INIT3	LOGICAL*4
QWR	HST3D	REAL*8		READ3	LOGICAL*4
	WELRIS	REAL*8		WRITE3	LOGICAL*4
	WFDYDZ	REAL*8			
			RDECHO	HST3D	LOGICAL*4
QWV	ASEMBL	REAL*8 DIMENSION(*)		CLOSE	LOGICAL*4
	ERROR3	REAL*8 DIMENSION(*)		DUMP	LOGICAL*4
	READ3	REAL*8 DIMENSION(*)		IREWI	LOGICAL*4
	SUMCAL	REAL*8 DIMENSION(*)		PLOTOC	LOGICAL*4
	WBBAL	REAL*8 DIMENSION(*)		READ1	LOGICAL*4
	WELLSS	REAL*8 DIMENSION(*)		READ2	LOGICAL*4
	WRITE3	REAL*8 DIMENSION(*)		READ3	LOGICAL*4
	WRITE5	REAL*8 DIMENSION(*)		REWI	LOGICAL*4
R	WELLSS	REAL*8 DIMENSION(*)			
RATIO	MAP2D	REAL*8			

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
RDFLXH	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDFLXQ	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDFLXS	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDLBC	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDMPDT	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDPLTP	PLOTOC	LOGICAL*4
RDSCBC	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDSPBC	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDSTBC	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDVAIF	HST3D	LOGICAL*4
	APLYBC	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR2	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
RDWDEF	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	ITER	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WBBAL	LOGICAL*4
	WELLSS	LOGICAL*4
	WELRIS	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE5	LOGICAL*4
RDWFLO	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
RDWHD	HST3D	LOGICAL*4
	ERROR3	LOGICAL*4
	ETOM2	LOGICAL*4
	INIT3	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
REN	WELLSS	REAL*8
	WFDYDZ	REAL*8
RESTRT	HST3D	LOGICAL*4
	READ1	LOGICAL*4
	WRITE1	LOGICAL*4

Variable name	Referencing programs	Variable type
RF	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
RH	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
RH1	ASEMBL	REAL*8 DIMENSION(*)
RHS	ASEMBL	REAL*8 DIMENSION(*)
	D4DES	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	L2SOR	REAL*8 DIMENSION(*)
	SOR2L	REAL*8 DIMENSION(*)
RHSSBC	ASEMBL	REAL*8 DIMENSION(*)
	SBCFLO	REAL*8 DIMENSION(*)
RHSW	ASEMBL	REAL*8 DIMENSION(*)
	WBCFLO	REAL*8 DIMENSION(*)
RIOAR	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
RM	INIT2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
RORW2	INIT2	REAL*8
RPRN	PRNTAR	CHARACTER*1
RS	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
RS1	ASEMBL	REAL*8 DIMENSION(*)
SAVLDO	HST3D	LOGICAL*4
	CLOSE	LOGICAL*4
	DUMP	LOGICAL*4
	READ3	LOGICAL*4
SCALMF	HST3D	LOGICAL*4
	CLOSE	LOGICAL*4
	DUMP	LOGICAL*4
	INIT2	LOGICAL*4
	INIT3	LOGICAL*4
	IREWI	LOGICAL*4
	PLOTOC	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	READ3	LOGICAL*4
	REWI	LOGICAL*4
	SUMCAL	LOGICAL*4
	WRITE1	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE3	LOGICAL*4
	WRITE5	LOGICAL*4

Variable name	Refer-encing programs	Variable type
SDECAY	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SHRES	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SHRESF	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SIR	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SIRO	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
SIRN	SUMCAL	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
SLMETH	HST3D	INTEGER*4	SOLUTE	HST3D	LOGICAL*4
	ASEMBL	INTEGER*4		APLYBC	LOGICAL*4
	DUMP	INTEGER*4		ASEMBL	LOGICAL*4
	ERROR1	INTEGER*4		CALCC	LOGICAL*4
	INIT1	INTEGER*4		COEFF	LOGICAL*4
	INIT2	INTEGER*4		CRSDSP	LOGICAL*4
	ITER	INTEGER*4		DUMP	LOGICAL*4
	READ1	INTEGER*4		ERROR2	LOGICAL*4
	READ2	INTEGER*4		ERROR3	LOGICAL*4
	SBCFLO	INTEGER*4		ETOM1	LOGICAL*4
	WBCFLO	INTEGER*4		INIT1	LOGICAL*4
	WRITE1	INTEGER*4		INIT2	LOGICAL*4
	WRITE2	INTEGER*4		INIT3	LOGICAL*4
	WRITE5	INTEGER*4		ITER	LOGICAL*4
				PLOTOC	LOGICAL*4
SMCALC	CALCC	LOGICAL*4		READ1	LOGICAL*4
				READ2	LOGICAL*4
				READ3	LOGICAL*4
				SUMCAL	LOGICAL*4
				VISCOS	LOGICAL*4
				WBBAL	LOGICAL*4
				WELLSS	LOGICAL*4
				WRITE1	LOGICAL*4
				WRITE2	LOGICAL*4
				WRITE3	LOGICAL*4
				WRITE4	LOGICAL*4
				WRITE5	LOGICAL*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
SOLVE	L2SOR	LOGICAL*4
	SOR2L	LOGICAL*4
SPR	HST3D	REAL*8 DIMENSION(3)
	L2SOR	REAL*8 DIMENSION(3)
SPRAD	L2SOR	REAL*8
	SOR2L	REAL*8
SRES	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SRESF	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SSRES	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SSRESF	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
STOTFI	HST3D	REAL*8
	APLYBC	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
STOTFP	HST3D	REAL*8
	APLYBC	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
STOTHI	HST3D	REAL*8
	APLYBC	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
STOTHP	HST3D	REAL*8
	APLYBC	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
STOTSI	HST3D	REAL*8
	APLYBC	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
STOTSP	HST3D	REAL*8
	APLYBC	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
SUM	INIT2	REAL*8
SUM1	ITER	REAL*8
	WELLSS	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
SUM2	ITER	REAL*8
SUMCWI	WELLSS	REAL*8
SUMMOB	WELLSS	REAL*8
SUMTWI	WELLSS	REAL*8
SUMWI	WELLSS	REAL*8
SUMXX	VSINIT	REAL*8
SUMXY	VSINIT	REAL*8
SVBC	ASEMBL	LOGICAL*4
SXX	ASEMBL	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
SXXM	ASEMBL	REAL*8
SXXP	ASEMBL	REAL*8
SYMID	MAP2D	CHARACTER*1 DIMENSION(5)
SYY	ASEMBL	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
SYYM	ASEMBL	REAL*8
SYYP	ASEMBL	REAL*8
SZZ	ASEMBL	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
SZZM	ASEMBL	REAL*8
SZZP	ASEMBL	REAL*8
SZZW	WELLSS	REAL*8
T	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	CALCC	REAL*8
	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	ITER	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	TOFEP	REAL*8
	VISCOS	REAL*8
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
TO	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WFDYDZ	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
TOO	HST3D	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
TOH	HST3D	REAL*8
	APLYBC	REAL*8
	ASEMBL	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
	WRITE2	REAL*8
	WRITE5	REAL*8
TA	BSODE	REAL*8
TABWR	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)
TAIF	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
TAMBI	HST3D	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
TAMBK	HST3D	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
TATWR	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)
TC	PLOTOC	REAL*8 DIMENSION(*)
TCS	PLOTOC	REAL*8 DIMENSION(*)
TCW	PLOTOC	REAL*8 DIMENSION(*)
TDATA	VSINIT	REAL*8 DIMENSION(*)
TDEHIR	WRITE5	REAL*8
TDFIR	WRITE5	REAL*8
TDSIR	WRITE5	REAL*8
TDX	COEFF	REAL*8
TDXY	COEFF	REAL*8
TDXZ	COEFF	REAL*8
TDY	COEFF	REAL*8
TDYX	COEFF	REAL*8
TDYZ	COEFF	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
TDZ	COEFF	REAL*8
TDZX	COEFF	REAL*8
TDZY	COEFF	REAL*8
TEHDT	HST3D	REAL*8 DIMENSION(14)
	BLOCKDATA	REAL*8 DIMENSION(14)
	DUMP	REAL*8 DIMENSION(14)
	INIT2	REAL*8 DIMENSION(14)
	READ1	REAL*8 DIMENSION(14)
	TOFEP	REAL*8 DIMENSION(14)
TEHST	HST3D	REAL*8 DIMENSION(32)
	BLOCKDATA	REAL*8 DIMENSION(32)
	DUMP	REAL*8 DIMENSION(32)
	INIT2	REAL*8 DIMENSION(32)
	READ1	REAL*8 DIMENSION(32)
	TOFEP	REAL*8 DIMENSION(32)
TEMP	TOFEP	REAL*8
TFLX	APLYBC	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
TFRES	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8

Variable name	Referencing programs	Variable type
TFRESF	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TFW	ASEMBL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
TFX	COEFF	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
TFXM	ASEMBL	REAL*8
TFXP	ASEMBL	REAL*8
TFY	COEFF	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
TFYM	ASEMBL	REAL*8
TFYP	ASEMBL	REAL*8
TFZ	COEFF	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
TFZM	ASEMBL	REAL*8
TFZP	ASEMBL	REAL*8
THCBC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
THETXZ	HST3D	REAL*8
	DUMP	REAL*8
	ERROR2	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
THETYZ	HST3D	REAL*8
	DUMP	REAL*8
	ERROR2	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
THETZZ	HST3D	REAL*8
	DUMP	REAL*8
	ERROR2	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
THI	TOFEP	REAL*8
THRES	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
THRESF	HST3D	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8

Variable name	Referencing programs	Variable type
THRU	HST3D	LOGICAL*4
	COEFF	LOGICAL*4
	READ3	LOGICAL*4
	WRITE4	LOGICAL*4
THX	COEFF	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
THXM	ASEMBL	REAL*8
THXP	ASEMBL	REAL*8
THXY	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
THXZ	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
THY	COEFF	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
THYM	ASEMBL	REAL*8
THYP	ASEMBL	REAL*8
THYX	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
THYZ	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
THZ	COEFF	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
THZM	ASEMBL	REAL*8
THZP	ASEMBL	REAL*8
THZX	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
THZY	COEFF	REAL*8 DIMENSION(*)
	CRSDSP	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)
TILT	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ERROR2	LOGICAL*4
	INIT2	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
TIMCHG	HST3D	REAL*8
	COEFF	REAL*8
	DUMP	REAL*8
	ERROR3	REAL*8
	ETOM2	REAL*8
	INIT3	REAL*8
	READ3	REAL*8
	SUMCAL	REAL*8
	WRITE1	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
TIMDAY	WELLSS	REAL*8
TIME	HST3D	REAL*8
	APLYBC	REAL*8
	CALCC	REAL*8
	CLOSE	REAL*8
	COEFF	REAL*8
	DUMP	REAL*8
	ERROR3	REAL*8
	ETOM2	REAL*8
	INIT2	REAL*8
	INIT3	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	READ3	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
	WRITE2	REAL*8
	WRITE3	REAL*8
	WRITE4	REAL*8
	WRITE5	REAL*8
TIMED	APLYBC	REAL*8
	WELRIS	REAL*8
TIMEDN	APLYBC	REAL*8
TIMRST	HST3D	REAL*8
	READ1	REAL*8
	WRITE1	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
TITLE	HST3D	CHARACTER*160
	DUMP	CHARACTER*160
	MAP2D	CHARACTER*80
	READ1	CHARACTER*160
	WRITE1	CHARACTER*160
	WRITE5	CHARACTER*80
TITLEO	HST3D	CHARACTER*160
	READ1	CHARACTER*160
	WRITE1	CHARACTER*160
TLBC	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
TLBL	WRITE5	CHARACTER*20
TLO	TOFEP	REAL*8
TM1	SOR2L	REAL*8 DIMENSION(*)
TM2	SOR2L	REAL*8 DIMENSION(*)
TMLBL	PLOTOC	CHARACTER*30
TNP	CALCC	REAL*8
	READ3	REAL*8 DIMENSION(*)
TO	PLOTOC	REAL*8 DIMENSION(*)
TOL	PLOT	REAL*8

Variable name	Referencing programs	Variable type
TOLDEN	HST3D	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
TOLDNC	HST3D	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	READ2	REAL*8
TOLDNT	HST3D	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	ITER	REAL*8
	READ1	REAL*8
	READ2	REAL*8
TOLDPW	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
TOLFPW	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8
TOLQW	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WELLSS	REAL*8
	WRITE2	REAL*8
TOS	PLOTOC	REAL*8 DIMENSION(*)
TOTFI	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTFP	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTHI	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTHP	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTSI	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTSP	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
TOTWFI	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTWFP	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTWHI	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTWHP	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTWSI	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOTWSP	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TOW	PLOTOC	REAL*8 DIMENSION(*)
TP1	SOR2L	REAL*8 DIMENSION(*)
TP2	SOR2L	REAL*8 DIMENSION(*)
TPHCBC	APLYBC	REAL*8 DIMENSION(*)
TQFAIF	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer- encing programs	Variable type
TQFFBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQFINJ	HST3D	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WRITE5	REAL*8
TQFLBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQFPRO	HST3D	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WRITE5	REAL*8
TQFSBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQHAIF	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQHFBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQHHBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQHINJ	HST3D	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WRITE5	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type
TQHLBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQHPRO	HST3D	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WRITE5	REAL*8
TQHSBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQSAIF	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQSFBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQSINJ	HST3D	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WRITE5	REAL*8
TQSLBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TQSPRO	HST3D	REAL*8
	SUMCAL	REAL*8
	WELLSS	REAL*8
	WRITE5	REAL*8
TQSSBC	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	SUMCAL	REAL*8
	WRITE5	REAL*8
TRVIS	HST3D	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	VISCOS	REAL*8
	VSINIT	REAL*8
	WRITE2	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
TSBC	ASEMBL	REAL*8 DIMENSION(*)	TSYM	ASEMBL	REAL*8
	INIT3	REAL*8 DIMENSION(*)			
	SUMCAL	REAL*8 DIMENSION(*)	TSYP	ASEMBL	REAL*8
	WRITE3	REAL*8 DIMENSION(*)			
			TSYX	COEFF	REAL*8 DIMENSION(*)
TSLBL	PLOTOC	CHARACTER*50		CRSDSP	REAL*8 DIMENSION(*)
				WRITE4	REAL*8 DIMENSION(*)
TSMAX	PLOTOC	REAL*8			
			TSYZ	COEFF	REAL*8 DIMENSION(*)
TSMIN	PLOTOC	REAL*8		CRSDSP	REAL*8 DIMENSION(*)
				WRITE4	REAL*8 DIMENSION(*)
TSRES	HST3D	REAL*8			
	SUMCAL	REAL*8	TSZ	COEFF	REAL*8 DIMENSION(*)
	WRITE5	REAL*8		WRITE4	REAL*8 DIMENSION(*)
TSRESF	HST3D	REAL*8	TSZM	ASEMBL	REAL*8
	SUMCAL	REAL*8			
	WRITE5	REAL*8	TSZP	ASEMBL	REAL*8
TSX	COEFF	REAL*8 DIMENSION(*)	TSZX	COEFF	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)		CRSDSP	REAL*8 DIMENSION(*)
				WRITE4	REAL*8 DIMENSION(*)
TSXM	ASEMBL	REAL*8			
			TSZY	COEFF	REAL*8 DIMENSION(*)
TSXP	ASEMBL	REAL*8		CRSDSP	REAL*8 DIMENSION(*)
				WRITE4	REAL*8 DIMENSION(*)
TSXY	COEFF	REAL*8 DIMENSION(*)			
	CRSDSP	REAL*8 DIMENSION(*)	TVD	HST3D	REAL*8 DIMENSION(10)
	WRITE4	REAL*8 DIMENSION(*)		DUMP	REAL*8 DIMENSION(10)
				ETOM1	REAL*8 DIMENSION(10)
TSXZ	COEFF	REAL*8 DIMENSION(*)		INIT2	REAL*8 DIMENSION(10)
	CRSDSP	REAL*8 DIMENSION(*)		READ1	REAL*8 DIMENSION(10)
	WRITE4	REAL*8 DIMENSION(*)		READ2	REAL*8 DIMENSION(10)
				WRITE2	REAL*8 DIMENSION(10)
TSY	COEFF	REAL*8 DIMENSION(*)			
	WRITE4	REAL*8 DIMENSION(*)			

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
TVF0	HST3D	REAL*8 DIMENSION(10)
	DUMP	REAL*8 DIMENSION(10)
	ETOM1	REAL*8 DIMENSION(10)
	INIT2	REAL*8 DIMENSION(10)
	READ1	REAL*8 DIMENSION(10)
	READ2	REAL*8 DIMENSION(10)
	VISCOS	REAL*8 DIMENSION(10)
	WRITE2	REAL*8 DIMENSION(10)
TVF1	HST3D	REAL*8 DIMENSION(10)
	DUMP	REAL*8 DIMENSION(10)
	ETOM1	REAL*8 DIMENSION(10)
	INIT2	REAL*8 DIMENSION(10)
	READ1	REAL*8 DIMENSION(10)
	READ2	REAL*8 DIMENSION(10)
	WRITE2	REAL*8 DIMENSION(10)
TVZHC	HST3D	REAL*8 DIMENSION(5)
	DUMP	REAL*8 DIMENSION(5)
	ETOM1	REAL*8 DIMENSION(5)
	INIT2	REAL*8 DIMENSION(5)
	READ1	REAL*8 DIMENSION(5)
	READ2	REAL*8 DIMENSION(5)
	WRITE2	REAL*8 DIMENSION(5)
TWKT	ASEMBL	REAL*8 DIMENSION(*)
	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
TWLBL	PLOTOC	CHARACTER*50
TWMAX	PLOTOC	REAL*8
TWMIN	PLOTOC	REAL*8
TWOPI	INIT2	REAL*8
TWREND	HST3D	REAL*8
	WBBAL	REAL*8
	WELLSS	REAL*8
	WELRIS	REAL*8
TWRK	WFDYDZ	REAL*8
TWSRKT	READ3	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
TWSUR	WBBAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
TX	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
TY	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
TZ	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
U0	BSODE	REAL*8	UBBLB	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8		INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8		READ2	REAL*8 DIMENSION(*)
U1	BSODE	REAL*8	UBBRB	READ2	REAL*8
	COEFF	REAL*8			
	INIT2	REAL*8	UC	INIT2	REAL*8
	PLOT	INTEGER*4		PLOTOC	REAL*8
	SUMCAL	REAL*8		READ3	REAL*8
	WELLSS	REAL*8		SUMCAL	REAL*8
	WRITE2	REAL*8		WBBAL	REAL*8
	WRITE5	REAL*8		WELLSS	REAL*8
				WRITE3	CHARACTER*11
U2	COEFF	REAL*8			
	SUMCAL	REAL*8	UCBC	INIT3	REAL*8 DIMENSION(*)
	WRITE2	REAL*8		READ3	REAL*8 DIMENSION(*)
	WRITE5	REAL*8			
			UCLM	WBBAL	REAL*8
U3	COEFF	REAL*8		WELLSS	REAL*8
	WRITE2	REAL*8			
	WRITE5	REAL*8	UCLP	WBBAL	REAL*8
				WELLSS	REAL*8
U4	WRITE2	REAL*8			
	WRITE5	REAL*8	UCRBC	READ3	REAL*8
U5	WRITE2	REAL*8	UCROSC	ASEMBL	REAL*8
	WRITE5	REAL*8		CRSDSP	REAL*8
U6	WRITE2	REAL*8	UCROST	ASEMBL	REAL*8
	WRITE5	REAL*8		CRSDSP	REAL*8
UARBC	INIT2	REAL*8	UCTC	COEFF	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
UCWT	COEFF	REAL*8
UDC	INIT2	REAL*8
	WELLSS	REAL*8
UDELY	PLOT	REAL*8
UDEN	COEFF	REAL*8
	INIT2	REAL*8
	SUMCAL	REAL*8
	WBBAL	REAL*8
UDENBC	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
UDENLB	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
UDNRBC	READ3	REAL*8
UDPWKT	ITER	REAL*8
UDT	INIT2	REAL*8
	WELLSS	REAL*8
UDTHHC	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
UDTIM	COEFF	REAL*8
UDX	COEFF	REAL*8
UDXDY	INIT2	REAL*8
UDXDYI	INIT2	REAL*8
UDXDYO	INIT2	REAL*8
UDXDZ	INIT2	REAL*8
UDXYZ	INIT2	REAL*8
UDXYZI	INIT2	REAL*8
UDXYZO	INIT2	REAL*8
UDY	COEFF	REAL*8
UDYDZ	INIT2	REAL*8
UDZ	COEFF	REAL*8
UEH	WBBAL	REAL*8
	WELLSS	REAL*8
UEHLM	WBBAL	REAL*8
	WELLSS	REAL*8
UEHLP	WBBAL	REAL*8
UFRAC	COEFF	REAL*8
UFX	COEFF	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
UFY	COEFF	REAL*8
UFZ	COEFF	REAL*8
UGDELX	INIT2	REAL*8
UGDELY	INIT2	REAL*8
UGDELZ	INIT2	REAL*8
UHRBC	READ3	REAL*8
UHWT	COEFF	REAL*8
UJW	READ2	INTEGER*4
UKHCBC	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
UKLB	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
UKRB	READ2	REAL*8
UNIGRX	HST3D	LOGICAL*4
	INIT2	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
UNIGRY	HST3D	LOGICAL*4
	INIT2	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4

Variable name	Referencing programs	Variable type
UNIGRZ	HST3D	LOGICAL*4
	INIT2	LOGICAL*4
	READ2	LOGICAL*4
	WRITE2	LOGICAL*4
UNITEP	REWI3	CHARACTER*10
	WRITE3	CHARACTER*10
UNITH	HST3D	CHARACTER*3
	DUMP	CHARACTER*3
	INIT1	CHARACTER*3
	PLOTOC	CHARACTER*3
	READ1	CHARACTER*3
	REWI3	CHARACTER*3
	WRITE2	CHARACTER*3
	WRITE3	CHARACTER*3
	WRITE4	CHARACTER*3
	WRITE5	CHARACTER*3
UNITHF	HST3D	CHARACTER*7
	DUMP	CHARACTER*7
	INIT1	CHARACTER*7
	PLOTOC	CHARACTER*7
	READ1	CHARACTER*7
	REWI3	CHARACTER*7
	WRITE2	CHARACTER*7
	WRITE3	CHARACTER*7
	WRITE4	CHARACTER*7
	WRITE5	CHARACTER*7

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Refer-encing programs	Variable type	Variable name	Refer-encing programs	Variable type
UNITL	HST3D	CHARACTER*2	UNITT	HST3D	CHARACTER*1
	DUMP	CHARACTER*2		DUMP	CHARACTER*1
	INIT1	CHARACTER*2		INIT1	CHARACTER*1
	PLOTOC	CHARACTER*2		PLOTOC	CHARACTER*1
	READ1	CHARACTER*2		READ1	CHARACTER*1
	REWI3	CHARACTER*2		REWI3	CHARACTER*1
	WRITE2	CHARACTER*2		WRITE2	CHARACTER*1
	WRITE3	CHARACTER*2		WRITE3	CHARACTER*1
	WRITE4	CHARACTER*2		WRITE4	CHARACTER*1
	WRITE5	CHARACTER*2		WRITE5	CHARACTER*1
UNITM	HST3D	CHARACTER*2	UNITTM	HST3D	CHARACTER*3
	DUMP	CHARACTER*2		DUMP	CHARACTER*3
	INIT1	CHARACTER*2		INIT1	CHARACTER*3
	PLOTOC	CHARACTER*2		PLOTOC	CHARACTER*3
	READ1	CHARACTER*2		READ1	CHARACTER*3
	REWI3	CHARACTER*2		REWI3	CHARACTER*3
	WRITE2	CHARACTER*2		WRITE2	CHARACTER*3
	WRITE3	CHARACTER*2		WRITE3	CHARACTER*3
	WRITE4	CHARACTER*2		WRITE4	CHARACTER*3
	WRITE5	CHARACTER*2		WRITE5	CHARACTER*3
UNITP	HST3D	CHARACTER*3	UNITVS	REWI3	CHARACTER*9
	DUMP	CHARACTER*3		WRITE2	CHARACTER*9
	INIT1	CHARACTER*3		WRITE3	CHARACTER*9
	PLOTOC	CHARACTER*3			
	READ1	CHARACTER*3	UNLP	PLOT	INTEGER*4
	REWI3	CHARACTER*3			
	WRITE2	CHARACTER*3	UP	WELLSS	REAL*8
	WRITE3	CHARACTER*3			
	WRITE4	CHARACTER*3	UP1	READ3	REAL*8
	WRITE5	CHARACTER*3		WRITE3	CHARACTER*11

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
UP2	READ3	REAL*8
	WRITE3	CHARACTER*11
UPABD	INIT2	REAL*8
UPHILB	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
UPHIM	APLYBC	REAL*8
UPS	PLOTOC	REAL*8
UPTC	COEFF	REAL*8
UPW	PLOTOC	REAL*8
UPWKT	ITER	REAL*8
	WELLSS	REAL*8
UPWKTS	WELLSS	REAL*8
UQ	INIT3	REAL*8
	READ3	REAL*8
	WRITE3	CHARACTER*11
UQCLM	WBBAL	REAL*8
	WELLSS	REAL*8
UQCLP	WBBAL	REAL*8
	WELLSS	REAL*8
UQH	INIT3	REAL*8
UQHLM	WBBAL	REAL*8
	WELLSS	REAL*8
UQHLP	WBBAL	REAL*8
	WELLSS	REAL*8
UQS	INIT3	REAL*8
UQW	WELLSS	REAL*8
UQWLM	WBBAL	REAL*8
	WELLSS	REAL*8
UQWLP	WBBAL	REAL*8
	WELLSS	REAL*8
UQWLYV	WBBAL	REAL*8
UQWM	WBBAL	REAL*8
	WELLSS	REAL*8
UQWV	WBBAL	REAL*8
	WELLSS	REAL*8
UR1	ASEMBL	REAL*8
UR2	ASEMBL	REAL*8
URH	ASEMBL	REAL*8
URS	ASEMBL	REAL*8
UT	INIT2	REAL*8
	PLOTOC	REAL*8
	READ3	REAL*8
	SUMCAL	REAL*8
	VSINIT	REAL*8
	WRITE3	CHARACTER*11

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
UTBC	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
UTLM	WBBAL	REAL*8
	WELLSS	REAL*8
UTLP	WELLSS	REAL*8
UTRBC	READ3	REAL*8
UTS	PLOTOC	REAL*8
UTTC	COEFF	REAL*8
UTW	PLOTOC	REAL*8
UTXM	ASEMBL	REAL*8
UTXP	ASEMBL	REAL*8
UTYM	ASEMBL	REAL*8
UTYP	ASEMBL	REAL*8
UTZM	ASEMBL	REAL*8
UTZP	ASEMBL	REAL*8
UVAIFC	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
UVAR	READ2	REAL*8
UVEL	COEFF	REAL*8
UVFLM	WELLSS	REAL*8
UVFLP	WELLSS	REAL*8
UVIS	COEFF	REAL*8
UVISLB	ETOM2	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	READ3	REAL*8 DIMENSION(*)
UVSRBC	READ3	REAL*8
UVWLM	WELLSS	REAL*8
UVWLP	WELLSS	REAL*8
UWI	INIT2	REAL*8
UXX1	SOR2L	REAL*8
UXX2	SOR2L	REAL*8
UYMAX	PLOT	REAL*8
UYMIN	PLOT	REAL*8
UZELB	ERROR2	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
UZERB	READ2	REAL*8
VA	ASEMBL	REAL*8 DIMENSION(7,*)
	D4DES	REAL*8 DIMENSION(7,*)
	ITER	REAL*8 DIMENSION(7,*)
	SOR2L	REAL*8 DIMENSION(7,*)
VAIFC	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
VAR	REWI	REAL*8
	REWI3	REAL*8 DIMENSION(3)
VASBC	ASEMBL	REAL*8 DIMENSION(7,*)
	SBCFLO	REAL*8 DIMENSION(7,*)
VAW	ASEMBL	REAL*8 DIMENSION(7,*)
	WBCFLO	REAL*8 DIMENSION(7,*)
VDATA	VSINIT	REAL*8 DIMENSION(*)
VELWRK	WFDYDZ	REAL*8
VIS	APLYBC	REAL*8 DIMENSION(*)
	COEFF	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)

Variable name	Referencing programs	Variable type
VISCTR	HST3D	REAL*8 DIMENSION(10)
	DUMP	REAL*8 DIMENSION(10)
	ETOM1	REAL*8 DIMENSION(10)
	INIT2	REAL*8 DIMENSION(10)
	READ1	REAL*8 DIMENSION(10)
	READ2	REAL*8 DIMENSION(10)
	VISCOS	REAL*8 DIMENSION(10)
	WRITE2	REAL*8 DIMENSION(10)
VISLBC	APLYBC	REAL*8 DIMENSION(*)
	INIT3	REAL*8 DIMENSION(*)
	WRITE3	REAL*8 DIMENSION(*)
VISOAR	HST3D	REAL*8
	APLYBC	REAL*8
	DUMP	REAL*8
	ETOM1	REAL*8
	INIT2	REAL*8
	READ1	REAL*8
	READ2	REAL*8
	WRITE2	REAL*8
VISTFO	HST3D	REAL*8 DIMENSION(10)
	DUMP	REAL*8 DIMENSION(10)
	ETOM1	REAL*8 DIMENSION(10)
	INIT2	REAL*8 DIMENSION(10)
	READ1	REAL*8 DIMENSION(10)
	READ2	REAL*8 DIMENSION(10)
	VISCOS	REAL*8 DIMENSION(10)
	WRITE2	REAL*8 DIMENSION(10)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
VISTF1	HST3D	REAL*8 DIMENSION(10)	W0	HST3D	REAL*8
	DUMP	REAL*8 DIMENSION(10)		DUMP	REAL*8
	ETOM1	REAL*8 DIMENSION(10)		INIT2	REAL*8
	INIT2	REAL*8 DIMENSION(10)		INIT3	REAL*8
	READ1	REAL*8 DIMENSION(10)		READ1	REAL*8
	READ2	REAL*8 DIMENSION(10)		READ2	REAL*8
	WRITE2	REAL*8 DIMENSION(10)		READ3	REAL*8
				SUMCAL	REAL*8
VPA	HST3D	REAL*8 DIMENSION(*)		VISCOS	REAL*8
	ASEMBL	REAL*8 DIMENSION(*)		WBBAL	REAL*8
	DUMP	REAL*8 DIMENSION(*)		WELLSS	REAL*8
	INIT1	REAL*8 DIMENSION(*)		WRITE2	REAL*8
	INIT2	REAL*8 DIMENSION(*)		WRITE3	REAL*8
	ITER	REAL*8 DIMENSION(*)		WRITE5	REAL*8
	L2SOR	REAL*8 DIMENSION(*)			
	READ1	REAL*8 DIMENSION(*)	W1	HST3D	REAL*8
	SUMCAL	REAL*8 DIMENSION(*)		DUMP	REAL*8
	WBBAL	REAL*8 DIMENSION(*)		INIT2	REAL*8
	WELLSS	REAL*8 DIMENSION(*)		INIT3	REAL*8
				READ1	REAL*8
VREF	VSINIT	REAL*8		READ2	REAL*8
				READ3	REAL*8
VSTLOG	VSINIT	REAL*8 DIMENSION(16)		VISCOS	REAL*8
				WRITE2	REAL*8
VXX	COEFF	REAL*8 DIMENSION(*)		WRITE3	REAL*8
	WRITE4	REAL*8 DIMENSION(*)		WRITE5	REAL*8
VYY	COEFF	REAL*8 DIMENSION(*)	WBOD	ETOM1	REAL*8 DIMENSION(*)
	WRITE4	REAL*8 DIMENSION(*)		INIT2	REAL*8 DIMENSION(*)
				READ2	REAL*8 DIMENSION(*)
VZZ	COEFF	REAL*8 DIMENSION(*)		WELRIS	REAL*8
	WRITE4	REAL*8 DIMENSION(*)		WRITE2	REAL*8 DIMENSION(*)

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
WCAIF	APLYBC	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
WCF	ERROR2	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
WCLBL1	WRITE2	CHARACTER*60 DIMENSION(0:5)
WCLBL2	WRITE2	CHARACTER*50 DIMENSION(0:2)
WFICUM	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
WFPCUM	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
WHICUM	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
WHPCUM	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
WI	INIT2	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
WIDLBL	PLOT	CHARACTER*80
	PLOTOC	CHARACTER*80
WQMETH	ASEMBL	INTEGER*4 DIMENSION(*)
	ERROR2	INTEGER*4 DIMENSION(*)
	ERROR3	INTEGER*4 DIMENSION(*)
	ITER	INTEGER*4 DIMENSION(*)
	PLOTOC	INTEGER*4
	READ2	INTEGER*4 DIMENSION(*)
	READ3	INTEGER*4 DIMENSION(*)
	SUMCAL	INTEGER*4 DIMENSION(*)
	WBBAL	INTEGER*4 DIMENSION(*)
	WELLSS	INTEGER*4 DIMENSION(*)
	WRITE2	INTEGER*4 DIMENSION(*)
	WRITE3	INTEGER*4 DIMENSION(*)
	WRITE5	INTEGER*4 DIMENSION(*)
WRANGL	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)
WRCALC	HST3D	LOGICAL*4
	DUMP	LOGICAL*4
	ETOM1	LOGICAL*4
	INIT2	LOGICAL*4
	READ1	LOGICAL*4
	READ2	LOGICAL*4
	WBBAL	LOGICAL*4
	WELLSS	LOGICAL*4
	WELRIS	LOGICAL*4
	WRITE2	LOGICAL*4
	WRITE5	LOGICAL*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
WRID	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)
WRIDT	HST3D	REAL*8
	WELRIS	REAL*8
	WFDYDZ	REAL*8
WRISL	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)
WRRUF	ETOM1	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	WELRIS	REAL*8
	WRITE2	REAL*8 DIMENSION(*)
WSICUM	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
WSPCUM	INIT2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
WT	COEFF	REAL*8
WTMX	ASEMBL	REAL*8
WTMY	ASEMBL	REAL*8
WTMZ	ASEMBL	REAL*8
WTPX	ASEMBL	REAL*8
WTPY	ASEMBL	REAL*8
WTPZ	ASEMBL	REAL*8
X	COEFF	REAL*8 DIMENSION(*)
	ERROR2	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	MAP2D	REAL*8 DIMENSION(*)
	PLOT	REAL*8
	READ2	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
	ZONPLT	REAL*8 DIMENSION(*)
X0	INIT2	REAL*8
X1	MAP2D	REAL*8
	ZONPLT	REAL*8
X2	MAP2D	REAL*8
	ZONPLT	REAL*8
XARG	INTERP	REAL*8
XC	PLOT	REAL*8 DIMENSION(*)
	ZONPLT	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
XD	ZONPLT	REAL*8	Y	COEFF	REAL*8 DIMENSION(*)
				ETOM1	REAL*8 DIMENSION(*)
XI	WELRIS	REAL*8		INIT2	REAL*8 DIMENSION(*)
				MAP2D	REAL*8 DIMENSION(*)
XINC	PLOT	REAL*8		PLOT	REAL*8 DIMENSION(6)
				READ2	REAL*8 DIMENSION(*)
XLBL	PLOT	CHARACTER*30		WRITE2	REAL*8 DIMENSION(*)
				WRITE5	REAL*8 DIMENSION(*)
XMAX	PLOT	REAL*8		ZONPLT	REAL*8 DIMENSION(*)
	ZONPLT	REAL*8			
			Y0	INIT2	REAL*8
XMIN	PLOT	REAL*8			
	ZONPLT	REAL*8	Y1	MAP2D	REAL*8
				WELLSS	REAL*8
XO	PLOT	REAL*8 DIMENSION(*)		WFDYDZ	REAL*8
				ZONPLT	REAL*8
XPRNT	PLOT	REAL*8			
			Y2	MAP2D	REAL*8
XS	INTERP	REAL*8 DIMENSION(*)		ZONPLT	REAL*8
XTOT	MAP2D	REAL*8	YARG	INTERP	REAL*8
XX	INIT2	REAL*8	YC	PLOT	REAL*8 DIMENSION(*)
	SOR2L	REAL*8 DIMENSION(*)		ZONPLT	REAL*8
	TOFEP	REAL*8			
	VISCOS	REAL*8	YD	ZONPLT	REAL*8
	VSINIT	REAL*8			
			YLBL	PLOT	CHARACTER*40
XXN	SOR2L	REAL*8 DIMENSION(*)			
			YMAX	PLOT	REAL*8
XYRAT	ZONPLT	REAL*8		ZONPLT	REAL*8

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type
YMAXHV	BSODE	REAL*8 DIMENSION(2,12)
YMIN	PLOT	REAL*8
	ZONPLT	REAL*8
YNHV	BSODE	REAL*8 DIMENSION(2,12)
YNM1HV	BSODE	REAL*8 DIMENSION(2,12)
YO	PLOT	REAL*8 DIMENSION(*)
	WELLSS	REAL*8
	WFDYDZ	REAL*8
YPOSUP	HST3D	LOGICAL*4
	READ3	LOGICAL*4
	WRITE3	LOGICAL*4
	WRITE5	LOGICAL*4
YREF	VSINIT	REAL*8
YS	INTERP	REAL*8 DIMENSION(*)
YTOT	MAP2D	REAL*8
YY	BSODE	REAL*8 DIMENSION(2)
	VSINIT	REAL*8
	WELRIS	REAL*8 DIMENSION(2)
	WFDYDZ	REAL*8 DIMENSION(2)
YYERR	BSODE	REAL*8 DIMENSION(2)
	WELRIS	REAL*8 DIMENSION(2)
YYMAX	BSODE	REAL*8 DIMENSION(2)
	WELRIS	REAL*8 DIMENSION(2)
YYMXSV	BSODE	REAL*8 DIMENSION(2)
YYN	BSODE	REAL*8 DIMENSION(2)
YYNM1	BSODE	REAL*8 DIMENSION(2)
YYSAVE	BSODE	REAL*8 DIMENSION(2)
Z	APLYBC	REAL*8 DIMENSION(*)
	ASEMBL	REAL*8 DIMENSION(*)
	BSODE	REAL*8
	COEFF	REAL*8 DIMENSION(*)
	ERROR2	REAL*8 DIMENSION(*)
	ETOM1	REAL*8 DIMENSION(*)
	INIT2	REAL*8 DIMENSION(*)
	READ2	REAL*8 DIMENSION(*)
	SUMCAL	REAL*8 DIMENSION(*)
	WELLSS	REAL*8 DIMENSION(*)
	WRITE2	REAL*8 DIMENSION(*)
	WRITE5	REAL*8 DIMENSION(*)
Z0	INIT2	REAL*8
Z1	MAP2D	REAL*8
Z2	MAP2D	REAL*8
ZA	BSODE	REAL*8
ZCHARS	MAP2D	CHARACTER*1 DIMENSION(0:31)
ZEBRA	MAP2D	LOGICAL*4
	WRITE5	LOGICAL*4

Table 11.2--*Cross-reference list of variables*--Continued

Variable name	Referencing programs	Variable type	Variable name	Referencing programs	Variable type
ZELBC	APLYBC	REAL*8 DIMENSION(*)	ZT	HST3D	REAL*8 DIMENSION(10)
	ASEMBL	REAL*8 DIMENSION(*)		DUMP	REAL*8 DIMENSION(10)
	ERROR3	REAL*8 DIMENSION(*)		ETOM1	REAL*8 DIMENSION(10)
	INIT2	REAL*8 DIMENSION(*)		INIT2	REAL*8 DIMENSION(10)
	SUMCAL	REAL*8 DIMENSION(*)		READ1	REAL*8 DIMENSION(10)
	WRITE2	REAL*8 DIMENSION(*)		READ2	REAL*8 DIMENSION(10)
				WRITE2	REAL*8 DIMENSION(10)
ZHCBC	APLYBC	REAL*8 DIMENSION(*)			
	ETOM1	REAL*8 DIMENSION(*)	ZTHC	HST3D	REAL*8 DIMENSION(5)
	INIT2	REAL*8 DIMENSION(*)		DUMP	REAL*8 DIMENSION(5)
	READ2	REAL*8 DIMENSION(*)		ETOM1	REAL*8 DIMENSION(5)
	WRITE2	REAL*8 DIMENSION(*)		INIT2	REAL*8 DIMENSION(5)
				READ1	REAL*8 DIMENSION(5)
ZPINIT	HST3D	REAL*8		READ2	REAL*8 DIMENSION(5)
	DUMP	REAL*8		WRITE2	REAL*8 DIMENSION(5)
	ETOM1	REAL*8			
	INIT2	REAL*8	ZU	BSODE	REAL*8
	READ1	REAL*8			
	READ2	REAL*8	ZWK	WELRIS	REAL*8
	WRITE2	REAL*8		WFDYDZ	REAL*8
ZPOSUP	HST3D	LOGICAL*4	ZWKT	WELLSS	REAL*8
	READ3	LOGICAL*4			
	WRITE3	LOGICAL*4			
	WRITE5	LOGICAL*4			

11.3 CROSS-REFERENCE LIST OF COMMON BLOCKS

A cross-reference list of common blocks (table 11.3) shows in which subprograms each named common block appears. Blank common is used only for the two variably-partitioned arrays. Generally, the common block names relate to the subprogram in which the variables of that common block are defined first. All common blocks are contained in the main program.

Table 11.3--*Cross-reference list of common blocks*

Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs
ASA	MAIN	IN1B	MAIN	IN1G	MAIN	IN1M	MAIN
	ASEMBL		DUMP		DUMP		DUMP
	CRSDSP		ERROR1		INIT1		INIT1
			INIT1		ITER		ITER
CCC	MAIN		READ1		READ1		READ1
	ASEMBL		WRITE1				SUMCAL
	CALC			IN1H	MAIN		
		IN1C	MAIN		DUMP	IN1P	MAIN
ER1	MAIN		DUMP		INIT1		DUMP
	ERROR1		INIT1		L2SOR		INIT1
	ERROR2		L2SOR		READ1		READ1
	ERROR3		READ1				
	ERRPRT			IN1I	MAIN	IN1Q	MAIN
	INIT2	IN1D	MAIN		DUMP		DUMP
	IREWI		DUMP		INIT1		INIT1
	ITER		INIT1		READ1		ITER
	READ1		ITER				READ1
	READ2		READ1	IN1J	MAIN		
	READ3				ASEMBL	IN1S	MAIN
	REWI	IN1E	MAIN		DUMP		DUMP
	REWI3		ASEMBL		INIT1		INIT1
	SOR2L		DUMP		READ1		PLOTOC
	SUMCAL		INIT1				READ1
			READ1	IN1L	MAIN		WRITE2
IN1A	MAIN				DUMP		WRITE3
	DUMP	IN1F	MAIN		INIT1		WRITE4
	INIT1		DUMP		INIT2		WRITE5
	READ1		INIT1		READ1		
			INIT2		WBBAL		
			READ1		WELLSS		

Table 11.3--Cross-reference list of common blocks--Continued

Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs
INIT	MAIN	IN2BV	MAIN	IN2F	MAIN	IN2L	MAIN
	DUMP		BLKDAT		APLYBC		CLOSE
	INIT1		DUMP		DUMP		DUMP
	PLOTOC		INIT2		INIT2		INIT2
	READ1		READ1		READ1		READ1
	REWI3		TOFEP				
	WRITE2			IN2H	MAIN	ITA	MAIN
	WRITE3	IN2C	MAIN		APLYBC		ASEMBL
	WRITE4		APLYBC		CALC		ITER
	WRITE5		ASEMBL		CLOSE		WRITE5
			DUMP		COEFF		
IN2A	MAIN		INIT2		DUMP	L2SAV	MAIN
	APLYBC		READ1		ERROR3		DUMP
	ASEMBL		SUMCAL		ETOM2		L2SOR
	CALC		WBBAL		INIT2		READ1
	COEFF		WELLSS		INIT3		WRITE5
	DUMP		WRITE2		ITER		
	ERROR3		WRITE5		READ1	ORA	MAIN
	INIT2				READ3		ASEMBL
	INIT3	IN2D	MAIN		SUMCAL		D4DES
	READ1		DUMP		WBBAL		DUMP
	READ3		INIT2		WELLSS		INIT1
	SUMCAL		READ1		WRITE2		ORDER
	WELLSS		VISCOS		WRITE3		READ1
	WELRIS				WRITE4		SBCFLO
		IN2E	MAIN		WRITE5		WRITE1
IN2B	MAIN		CALC				
	BLKDAT		DUMP	IN2I	MAIN		
	DUMP		INIT2		APLYBC		
	INIT2		ITER		DUMP		
	READ1		READ1		INIT2		
	TOFEP		SUMCAL		READ1		
			WBBAL		SUMCAL		
			WELLSS		WRITE2		
			WFDYDZ		WRITE5		

Table 11.3--*Cross-reference list of common blocks*--Continued

Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs
RD1A	MAIN	RD1B	MAIN	RD1D	MAIN	RD1E	MAIN
	APLYBC		READ1		ASEMBL		APLYBC
	ASEMBL		WRITE1		CLOSE		ASEMBL
	CALC				COEFF		DUMP
	COEFF	RD1C	MAIN		DUMP		ERROR1
	CRSDSP		APLYBC		ERROR2		ERROR2
	DUMP		ASEMBL		ERROR3		ERROR3
	ERROR2		COEFF		ETOM1		ETOM1
	ERROR3		CRSDSP		INIT1		ETOM2
	ETOM1		D4DES		INIT2		INIT1
	ETOM2		DUMP		INIT3		INIT3
	INIT1		ERROR1		IREWI		INIT3
	INIT2		ERROR2		ITER		ITER
	INIT3		ETOM1		L2SOR		READ1
	ITER		ETOM2		PLOTOC		READ2
	PLOTOC		INIT1		READ1		READ3
	READ1		INIT2		READ2		SBCFLO
	READ2		INIT3		READ3		SUMCAL
	READ3		IREWI		REWI		WRITE1
	SUMCAL		ITER		REWI3		WRITE2
	VISCOS		ORDER		SUMCAL		WRITE3
	WBBAL		PLOTOC		WBBAL		WRITE5
	WELLSS		PRNTAR		WELLSS		
	WRITE1		READ1		WRITE1	RD1F	MAIN
	WRITE2		READ2		WRITE2		COEFF
	WRITE3		READ3		WRITE3		DUMP
	WRITE4		REWI		WRITE4		ERROR1
	WRITE5		REWI3		WRITE5		ERROR2
			SBCFLO		ZONPLT		ETOM1
RD1AC	MAIN		SOR2L				INIT1
	DUMP		SUMCAL				INIT2
	READ1		WBBAL				READ1
	WRITE1		WELLSS				READ2
			WRITE1				WRITE1
			WRITE2				WRITE2
			WRITE3				ZONPLT
			WRITE4				
			WRITE5				

Table 11.3--*Cross-reference list of common blocks*--Continued

Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs
RD1G	MAIN	RD2B	MAIN	RD2E	MAIN	RD2G	MAIN
	APLYBC		APLYBC		ASEMBL		DUMP
	DUMP		ASEMBL		COEFF		ERROR2
	ERROR1		CALC		DUMP		ETOM1
	ERROR2		COEFF		ETOM1		INIT2
	ETOM1		CRSDSP		INIT2		READ1
	INIT1		DUMP		INIT3		READ2
	INIT2		ETOM1		PLOTOC		VISCOS
	READ1		INIT2		READ1		WRITE2
	READ2		READ1		READ2		
	WRITE1		READ2		READ3	RD2GV	MAIN
	WRITE2		SUMCAL		SUMCAL		DUMP
			WELLSS		VISCOS		ETOM1
RD1H	MAIN		WELRIS		WBBAL		INIT2
	ASEMBL		WFDYDZ		WELLSS		READ1
	COEFF		WRITE2		WRITE2		READ2
	DUMP		WRITE5		WRITE3		VISCOS
	ERROR1				WRITE5		WRITE2
	INIT1	RD2C	MAIN				
	INIT2		DUMP	RD2F	MAIN	RD2H	MAIN
	ITER		ETOM1		DUMP		APLYBC
	READ1		INIT2		ETOM1		DUMP
	READ2		READ1		INIT2		ERROR2
	SBCFLO		READ2		READ1		ETOM1
	WRITE1		WRITE2		READ2		INIT2
	WRITE2				WRITE2		READ1
	WRITE4	RD2D	MAIN				READ2
	WRITE5		INIT2				WRITE2
			READ2				
RD2A	MAIN		WRITE2			RD2IV	MAIN
	DUMP						APLYBC
	ERROR2						BLKDAT
	INIT2						DUMP
	READ1						READ1
	READ2						READ2
	WRITE2						WRITE2

Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs
RD2J	MAIN	RD2L	MAIN	RD2P	MAIN	RD2U	MAIN
	APLYBC		DUMP		DUMP		DUMP
	ASEMBL		INIT2		ETOM1		IREWI
	DUMP		ITER		INIT2		READ1
	ETOM1		READ1		READ1		READ2
	INIT2		READ2		READ2		REWI
	READ1		WRITE2		WELLSS		REWI3
	READ2				WRITE2		SUMCAL
	SUMCAL	RD2M	MAIN				
	TOFEP		CLOSE	RD2Q	MAIN	RD2V	MAIN
	WBBAL		DUMP		DUMP		DUMP
	WELLSS		READ1		INIT2		INIT2
	WELRIS		READ2		ITER		READ1
	WFDYDZ		READ3		READ1		READ2
	WRITE2		WELLSS		READ2		READ3
	WRITE5		WELRIS		WRITE2		SUMCAL
			WRITE2				WRITE2
RD2K	MAIN		WRITE3				WRITE3
	DUMP		WRITE5	RD2S	MAIN		WRITE4
	ERROR2				APLYBC		WRITE5
	ETOM1	RD2N	MAIN		ASEMBL		
	INIT2		DUMP		COEFF		
	READ1		ETOM1		DUMP	RD2W	ETOM1
	READ2		INIT2		READ1		
	WRITE2		ITER		READ2	RD3A	MAIN
			READ1		SBCFLO		ERROR3
RD2KV	MAIN		READ2		SUMCAL		ETOM2
	DUMP		WBBAL		WELLSS		INIT3
	ETOM1		WELLSS		WRITE2		READ3
	INIT2		WELRIS		WRITE5		WRITE3
	READ1		WRITE2				
	READ2		WRITE5	RD2T	MAIN	RD3B	MAIN
	WRITE2				DUMP		COEFF
					INIT2		ERROR3
					ITER		ETOM2
					READ1		INIT3
					READ2		READ3
					SOR2L		WRITE3
					WRITE2		

Table 11.3--*Cross-reference list of common blocks*--Continued

Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs	Common block name	Referencing programs
RD3C	MAIN	RD3I	MAIN	SCA	MAIN	SORA	MAIN
	READ3		CLOSE		COEFF		SOR2L
	WRITE4		COEFF		ITER		WRITE5
	WRITE5		DUMP		SUMCAL		
			ERROR3		WRITE5	WRA	MAIN
RD3D	MAIN		ETOM2				WELRIS
	ERROR3		INIT3	SCB	MAIN		WFDYDZ
	READ3		READ3		APLYBC		
	WRITE3		SUMCAL		SUMCAL	WRB	MAIN
	WRITE5		WRITE1		WRITE5		BSODE
			WRITE3				WELRIS
RD3E	MAIN		WRITE4	SCC	MAIN		
	CLOSE		WRITE5		SUMCAL	WRC	MAIN
	DUMP				WELLSS		WELRIS
	READ3				WRITE5		WFDYDZ
RD3EV	MAIN					WSSA	MAIN
	ERROR3						WBBAL
	ETOM2						WELLSS
	INIT3						WELRIS
	READ3						WFDYDZ
	WRITE3						
	WRITE5						
RD3F	MAIN						
	READ3						
RD3G	MAIN						
	COEFF						
	READ3						
	WRITE4						

★U.S. GOVERNMENT PRINTING OFFICE: 1987 774-930/65111

11.4 COMMENT FORM AND MAILING-LIST REQUEST

If you wish to be placed on a mailing list for revisions of the program documentation and announcements of new releases of this program, please return this page to:

HST3D Code Custodian
U.S. Geological Survey
Water Resources Division
Box 25046, Mail Stop 413
Denver Federal Center
Denver, Colorado 80225

Any comments on the program or documentation will be appreciated.

Name____________________________________ Date____________________

Organization____

Street____

City__________________________State______________________________

Zip code or Country____________________

Telephone Number (   ) ____________________