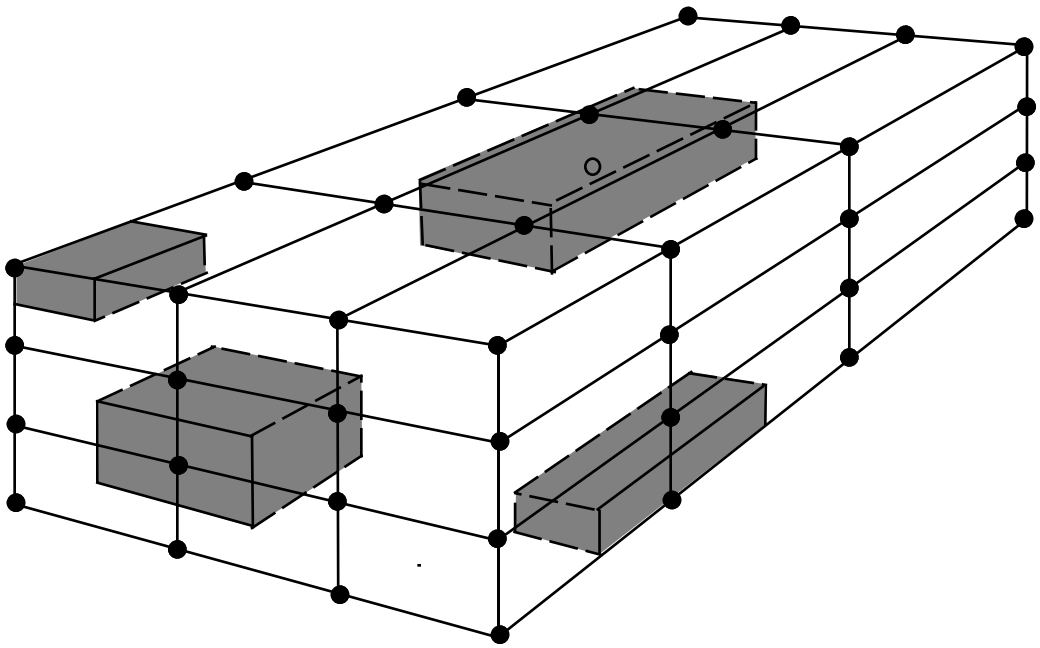
U.S. Department of the Interior
U.S. Geological Survey

Guide to the Revised Heat and Solute Transport Simulator: HST3D — Version 2

Water-Resources Investigations Report 97-4157



USGS
science for a changing world

Guide to the Revised Heat and Solute Transport Simulator: HST3D — Version 2

By Kenneth L. Kipp, Jr.

U.S. Geological Survey

Water-Resources Investigations Report 97-4157

Denver, Colorado
1997

U.S. DEPARTMENT OF THE INTERIOR
BRUCE BABBITT, Secretary

U.S. GEOLOGICAL SURVEY
Gordon P. Eaton, Director

The use of firm, trade, and brand names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

For additional information write to:

Chief, Branch of Regional Research
U.S. Geological Survey
Box 25046, Mail Stop 418
Denver Federal Center
Denver, Colorado 80225-0046

Copies of this report can be purchased from:

U.S. Geological Survey
Branch of Information Services
Box 25286
Denver Federal Center
Denver, Colorado 80225-0286

CONTENTS

ABSTRACT . . . 1
1. OVERVIEW OF THE SIMULATOR . . . 3
1.1 APPLICABILITY AND LIMITATIONS . . . 4
1.2 PURPOSE AND SCOPE . . . 5
1.3 ACKNOWLEDGMENTS . . . 5
2. THEORY . . . 7
2.1 FLOW AND TRANSPORT EQUATIONS . . . 7
2.2 PROPERTY FUNCTIONS AND TRANSPORT COEFFICIENTS . . . 7
2.2.1 Fluid-Viscosity Function . . . 7
2.3 BOUNDARY CONDITIONS . . . 8
2.3.1 Evapotranspiration Boundary Conditions . . . 8
2.3.2 Unconfined Aquifer, Free-Surface Boundary Condition . . . 10
2.4 INITIAL CONDITIONS . . . 10
3. NUMERICAL IMPLEMENTATION . . . 13
3.1 EQUATION DISCRETIZATION . . . 13
3.1.1 Spatial Discretization . . . 13
3.1.2 Automatic Time-Step Algorithm . . . 13
3.2 BOUNDARY CONDITIONS . . . 14
3.2.1 Leakage-Boundary Condition . . . 15
3.2.1.1 River-Leakage Boundary . . . 16
3.2.2 Evapotranspiration Boundary Condition . . . 16
3.2.3 Unconfined-Aquifer, Free-Surface Boundary Condition . . . 16
3.2.4 Simultaneous Specified-Value Boundary Conditions . . . 19
3.3 INITIAL CONDITIONS . . . 19
3.4 EQUATION SOLUTION . . . 19
3.4.1 Modification of the Flow and Transport Equations . . . 19
3.5 MATRIX EQUATION SOLVERS . . . 20
3.5.1 Equation Preconditioning . . . 20
3.5.2 Reduced Matrix Equation . . . 20
3.5.3 Choosing the Equation Solver . . . 21
3.5.4 Choosing the Parameters for an Iterative Equation Solver . . . 21
4. COMPUTER CODE DESCRIPTION . . . 23
4.1 CODE ORGANIZATION . . . 23
4.2 MEMORY ALLOCATION, ARRAY-SIZE REQUIREMENTS, AND SUBPROGRAM COMMUNICATION . . . 30
4.3 FILE USAGE . . . 32
4.4 INITIALIZATION OF VARIABLES . . . 33
4.5 PROGRAM EXECUTION . . . 33
4.6 RESTART OPTION . . . 33
5. THE DATA-INPUT FILE . . . 35
5.1 LIST-DIRECTED INPUT . . . 35

5.2 PREPARING THE DATA-INPUT FILE . . . 35
5.2.1 General Information . . . 40
5.2.2 Input-Record Descriptions . . . 42
6. OUTPUT DESCRIPTION . . . 65
6.1 SUMMARY OF OUTPUT FILES . . . 65
6.2 OUTPUT SELECTION . . . 65
6.3 OUTPUT OF DATA FOR POST-PROCESSING INTO PLOTS . . . 66
6.4 OUTPUT OF DATA FOR FLOW AND AMOUNT CALCULATIONS . . . 68
6.5 DIAGNOSTIC OUTPUT . . . 71
7. COMPUTER SYSTEM CONSIDERATIONS . . . 73
7.1 COMPUTER SYSTEM REQUIREMENTS . . . 73
7.2 COMPUTER PORTING EXPERIENCE . . . 74
8. COMPUTER-CODE VERIFICATION . . . 75
8.1 SUMMARY OF VERIFICATION TEST PROBLEMS . . . 75
8.1.1 Thermal Convection Experiment: Elder's Problem . . . 75
8.1.2 Solute Convection Experiment: Modified Elder's Problem . . . 76
8.1.3 Saltwater Intrusion into a Coastal Aquifer: Henry's Problem . . . 80
8.1.4 Ground-water Flow over a Salt Dome: HYDROCOIN Problem . . . 83
8.1.5 Saltwater Intrusion into a Coastal Aquifer . . . 86
8.2 AN EXAMPLE PROBLEM . . . 94
8.2.1 Use of BCFLOW: the post-processing program for flow rates and cumulative amounts . . . 103
9. NOTATION . . . 105
9.1 ROMAN CHARACTERS . . . 105
9.2 GREEK CHARACTERS . . . 106
9.3 MATHEMATICAL OPERATORS AND SPECIAL FUNCTIONS . . . 106
10. REFERENCES CITED . . . 107
APPENDIX A — DIMEN: A PRE-PROCESSOR FOR DIMENSIONING HST3D . . . 109
A.1 INTRODUCTION . . . 109
A.2 COMPUTER CODE DESCRIPTION . . . 109
A.2.1 Program Structure and Organization . . . 109
A.2.2 Memory Allocation, Array-Size Requirements, and Subprogram Communication . . . 113
A.2.3 File Usage . . . 113
A.2.4 Initialization of Variables . . . 114
A.2.5 Program Execution . . . 114
A.3 OUTPUT DESCRIPTION . . . 114
A.4 COMPUTER SYSTEM CONSIDERATIONS . . . 115
A.4.1 Computer System Requirements . . . 115
A.5 COMPUTER-CODE VERIFICATION . . . 115
APPENDIX B — BCFLOW: A POST-PROCESSOR FOR CALCULATING BOUNDARY FLOW RATES AND CUMULATIVE AMOUNTS FROM HST3D SIMULATIONS . . . 117
B.1 INTRODUCTION . . . 117

B.1.1 Limitations 117
B.2 COMPUTER CODE DESCRIPTION 117
B.2.1 Program Structure and Organization 117
B.2.2 Memory Allocation, Array-Size Requirements, and Subprogram Communication 121
B.2.3 File Usage 121
B.2.4 Initialization of Variables 122
B.2.5 Program Execution 122
B.3 DATA-INPUT FILE DESCRIPTION 122
B.3.1 PREPARING THE DATA-INPUT FILE 122
B.3.1.1 General Information 124
B.3.1.2 Input-Record Descriptions 124
B.4 OUTPUT DESCRIPTION 126
B.5 COMPUTER SYSTEM CONSIDERATIONS 126
B.5.1 Computer System Requirements 126
B.6 COMPUTER-CODE VERIFICATION 126
B.6.1 AN EXAMPLE PROBLEM 127
APPENDIX C — SOFTWARE FOR THE HST3D SIMULATOR AND ASSOCIATED PROGRAMS 145
C.1 STORAGE SPACE ON HARD DISC 145
C.2 OBTAINING THE SOFTWARE 145
C.3 INSTALLATION STEPS 145
C.4 ALTERNATIVE INSTALLATION PROCEDURES 146
APPENDIX D — REPORTING BUGS 147
D.1 PROVIDING COMPUTER AND SOFTWARE INFORMATION 147
D.2 VERIFYING THE BUG 147
D.3 REPRODUCING THE PROBLEM 147
D.4 DESCRIBING THE PROBLEM 147
D.5 SIMPLIFYING THE PROBLEM 147
D.6 CONTACTING FOR HELP 148
D.7 SENDING DATA WITH THE PROBLEM REPORT 148
APPENDIX E — USER REGISTRATION 149

FIGURES

Figure 2.1. Evapotranspiration flux as a function of water-table depth below land surface. 9
Figure 4.1. Connection chart for the HST3D main program and subprograms showing routine hierarchy. 27
Figure 8.1. System geometry and boundary conditions for Elder's problem. 76
Figure 8.2. Scaled temperature field at six dimensionless time values for Elder's problem from HST3D. 77

Figure 8.3. Scaled temperature field at six dimensionless time values for Elder's problem from Elder (1967) (used with permission). 78
Figure 8.4. System geometry and boundary conditions for modified Elder's problem for solute. 79
Figure 8.5. Scaled concentration field at six time values for modified Elder's problem from HST3D. 81
Figure 8.6. Scaled concentration field at six time values for modified Elder's problem from Elder (1967) (used with permission), dashed, and Voss and Souza (1987), solid. 82
Figure 8.7. Region geometry and boundary conditions for Henry's problem. 83
Figure 8.8. Scaled concentration contours for Henry's problem from HST3D. 84
Figure 8.9. Scaled concentration contours for Henry's problem from Ségol (1994). 84
Figure 8.10. Region and boundary conditions for the HYDROCOIN problem. 86
Figure 8.11. Scaled brine concentration field for the HYDROCOIN problem from HST3D. 87
Figure 8.12. Scaled brine concentration field for the HYDROCOIN problem from SUTRA. 87
Figure 8.13. Region and boundary conditions for the coastal aquifer problem of Huyakorn and others (1987). 88
Figure 8.14. Areal view of contours of scaled salt concentration at the bottom and top of the aquifer for the Huyakorn problem from HST3D. 89
Figure 8.15. Areal view of the horizontal components of the velocity vectors (A) at the top and (B) at the bottom of the aquifer for the Huyakorn problem from HST3D. 90
Figure 8.16. Cross-sectional view at y = 2400 ft of (A) contours of scaled salt concentration and (B) velocity vectors for the Huyakorn problem from HST3D. 91
Figure 8.17. Areal view of contours of scaled salt concentration at the bottom and top of the aquifer for the Huyakorn problem from Huyakorn and others (1987). 92
Figure 8.18. Areal view of the horizontal components of the velocity vectors (A) at the top and (B) at the bottom of the aquifer for the Huyakorn problem from Huyakorn and others (1987). 93
Figure 8.19. Cross-sectional view at y = 2400 ft of (A) contours of scaled salt concentration and (B) velocity vectors for the Huyakorn problem from Huyakorn and others (1987). 94
Figure A.1. Connection chart for the DIMEN main program and subprograms showing routine hierarchy. 112
Figure B.1. Connection chart for the BCFLOW main program and subprograms showing routine hierarchy. 120

TABLES

Table 5.1. File of comment lines from the data-input form 36

Table 5.2. Metric and U.S. customary system units used in the heat- and solute-transport simulator 43
Table 5.3. List of options for numerical program-control variables 63
Table 6.1. Structure of the IBC array 71
Table 7.1. Platforms to which the HST3D code has been ported 74
Table 8.1. Data file for example problem with annotations 95
Table B.1. File of comment lines from the data-input form for the BCFLOW program 123
Table B.2. Data file for the BCFLOW program for the coastal-aquifer example with annotations 128
Table B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example. 130

Guide to the Revised Heat and Solute Transport Simulator: HST3D — Version 2

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. Over the years since the release of Versions 1.0 through 1.4, various additions, modifications, and corrections have been made to the original simulator. This report documents the release of Version 2. Major changes included in this version are (1) a revised data-input file with all spatial information described by coordinate location; (2) a new iterative solver for the matrix equations based on a generalized conjugate-gradient method; (3) an evapotranspiration boundary condition; (4) a division of the simulator output into many files; (5) a new set of output files designed for use by post-processing programs for graphical visualization and for flow totalization; (6) a pre-processor for evaluating dimensioning requirements; and (7) a post-processor for totalizing boundary flow rates and cumulative amounts. The post-processing program for graphical visualization must be supplied by the user. Version 2 of the simulator has been verified using five test problems selected from the published literature. One involves heat transport, four involve solute transport, and all have variable-density fluids.

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. 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 HST3D simulator is a rather general computer code with various discretization, boundary-condition, source-sink, and equation-solver options. The computer code described in this documentation is Version 2. It incorporates several major modifications and additions to the current release (Version 1.4) of the code documented in Kipp (1987). These modifications include the following:

1. All geometric data are entered by coordinate location instead of mesh node number.
2. A new iterative solver is included based on a generalized-conjugate-gradient method with red-black or alternating diagonal zig-zag renumbering of the node points to form a reduced matrix.
3. An evapotranspiration boundary condition has been added.
4. The river leakage boundary condition has been extended to allow for multiple branches of a river to meet in a cell.
5. The free-surface boundary-condition algorithm has been revised to allow the free surface to move to any elevation in the simulation region and cells to dry and rewet.
6. The automatic time-step algorithm has been replaced by a more robust method.
7. The large variably partitioned arrays have been replaced by fixed arrays in common blocks incorporated by INCLUDE statements and dimensioned by PARAMETER statements.
8. Common blocks have been reorganized by variable type and usage.
9. Subroutine argument lists have been virtually eliminated.
10. Boundary-condition data can be input as a linear distribution over a boundary segment.
11. Spatial data can be input on a node-by-node or element-by-element basis from gridded data prepared by a geographic information system.
12. The Gauss elimination used for numerical coupling of the difference equations for flow and transport is suppressed in cases of unconfined flow to eliminate instability when cells become dry.
13. A pre-processing program is included for checking the sizes of array dimensions for a given simulation and creating the appropriate PARAMETER statements for redimensioning the arrays.
14. The output file has been split up into separate files based on data type to eliminate the handling of very large files.

15. Output intervals can be specified in simulation time units or number of time steps.
16. Output files can be written for post-processing programs to create contour plots, vector plots, and temporal plots of dependent variables.
17. A post-processing program is included for calculating boundary flow rates and cumulative-flow amounts for boundary zones specified by the user.
18. The full nine-component dispersion-coefficient tensor is used for cross-dispersive flux calculations.
19. The time units to be used in the simulation may be selected by the user.
20. The Fortran unit numbers for input and output files are parameters that can be changed for different computer systems.

1.1 APPLICABILITY AND LIMITATIONS

The HST3D code is suitable for simulating ground-water flow and 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 fresh or saline aquifers, contaminant plume movement, saltwater intrusion in coastal regions, brine disposal, freshwater storage in saline aquifers, heat storage in aquifers, liquid-phase geothermal systems, and similar transport situations. For most applications, only the heat- or the solute-transport equation is solved in conjunction with ground-water flow. Three-dimensional Cartesian and 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. In addition, small cell sizes may be necessary to accurately represent density-field variations that determine the flow field. These three factors can combine to cause an excessive number of nodes to be involved for a given model, thus making the simulations prohibitively demanding of computer-storage and computation-time. In such cases, a simple model of the system, useful for investigating mechanisms and testing hypotheses, may be all that is practical.

Another limitation is that finite-difference grids do not conform to boundaries that are not parallel to the coordinate axes. Stair-step approximations to angular boundaries are inconvenient to specify and can cause local variations in the ground-water flow field that are not realistic.

The simulation region is spatially discretized using a point-distributed grid. Spatial properties are defined by zones, whose boundaries are planes of nodes. This means that spatial averaging over the cells will cause the values of dependent variables in cells that span a large property contrast to not accurately represent conditions in either part of the cell. Computed variables at the cell boundaries will also be unrepresentative averages across property zones. This effect can be reduced by having an additional plane of cells near the boundary of a property change. Then, most of each cell will lie in one zone or the other. The cost of this adjustment is the use of more nodes than a cell centered mesh would require for the same accuracy at the boundary of a large property contrast. A further limitation is that the zones must be convex rectangular parallelopipeds. Zones with staircase boundaries or L-shaped zones are not allowed.

Although cells at edges and corners of the active simulation region may have 2 or 3 boundary faces, only one flux-type and one leakage-type and one aquifer-influence-function-type boundary condition can be specified for a given cell. For example, along edges where a lateral leakage boundary face may meet a vertical river leakage boundary face, one of those faces must be pulled back from that edge so there are not two leakage boundary condition faces for that cell. Furthermore, the associated temperature and concentration at a specified-pressure boundary cell cannot be different for each face of a cell with multiple boundary faces. Similarly the parameters associated with any of the flux boundary conditions must be the same for all boundary faces of a given cell.

The free surface and leakage boundary conditions require that the z-axis be oriented in the vertical direction. Thus, these conditions cannot be used with a tilted coordinate system.

There can only be one well bore open to any given cell. For fully penetrating wells screened over the entire aquifer thickness, this means that only one well bore per vertical column of cells is allowed. Furthermore, the simulator cannot represent portions of well bores that are screened in cells that have different porous-medium property zones in their horizontal quadrants.

Another limitation is that strong diagonal flows through a cell can cause the cross-dispersive heat and solute fluxes to produce negative temperatures and concentrations. One remedy is to suppress the cross-dispersive flux terms in the transport equations.

Because the HST3D simulator is a rather general computer code with a variety of options, it is computationally less efficient than a simulation code designed specifically for the given system being investigated. This limitation is compensated by the ability of the simulator to represent a wide variety of physical situations.

1.2 PURPOSE AND SCOPE

The purpose of this documentation is to provide the user with information about the additions and modifications incorporated into HST3D to form Version 2. Sections on the theory, numerical implementation, code organization, input information, and output information, are provided. This documentation must be used in conjunction with the documentation for HST3D Version 1 (Kipp, 1987).

Each release of the HST3D program code is identified by a version and release number in the form: (*version number.release number*). This number will change as modifications, corrections, and additions are made to the program. Modifications requiring revisions to the documentation change the version number, while minor revisions change the release number.

1.3 ACKNOWLEDGMENTS

The contributions of T. Nishikawa, N. Sepulveda, W. Sanford, J. Mason, and R. Hanson of the U.S. Geological Survey who provided application problems that influenced program development and who helped with program testing are gratefully acknowledged. G. Ségol generously provided a Fortran code of her analytic solution to Henry's problem.

2. THEORY

2.1 FLOW AND TRANSPORT EQUATIONS

The saturated ground-water flow and heat- and solute-transport equations solved by this simulator along with the assumptions used can be found in Kipp (1987) Equations for the new additions will be presented in this chapter. Explanations of the notation will appear after the first usage. A complete table of notation appears in chapter 9. In this report, all variables will be given in metric (SI) units of measure as used internally by the simulator. The coordinate system is always right-handed with the z-axis pointing vertically upward. The terms "solute-mass fraction" and "solute concentration" will be taken to be synonymous.

2.2 PROPERTY FUNCTIONS AND TRANSPORT COEFFICIENTS

Before the three conservation equations can be solved, information about the fluid properties, porous-medium properties, and transport coefficients needs to be obtained. In the HST3D simulator, density, viscosity, and porosity are functions of the dependent variables: pressure, temperature, and solute-mass fraction.

2.2.1 Fluid-Viscosity Function

Fluid viscosity is strongly dependent on temperature, and, to a lesser extent, on pressure and solute concentration. An empirical equation for viscosity of brines as a function of pressure, temperature, and solute concentration was adapted from a set of calculator programs for petroleum fluids published by Hewlett-Packard (1985, p. 102-107) and is written as:

$$\mu_0(p,T) = 243.18\times10^{-7}\cdot10^{[247.8/(T'-140)]}\cdot[1 + (P' - P'_{sat})1.0467\times10^{-6}(T'-305)] \qquad (2.2.1.1a)$$

where

μ_0 is the fluid viscosity of pure water (Pa–s);
P' is the pressure (bar) ;
P'_{sat} is the saturation pressure (bar) ; and
T' is the temperature (K).

The viscosity adjusted for salinity is given by:

$$\mu(p,T,w) = \mu_0(p,T)\left[1 - 1.87\times10^{-3}w_p^{0.5} + 2.18\times10^{-4}w_p^{2.5} + \left(T^{0.5} - 0.0135T\right)\left(2.76\times10^{-3}w_p - 3.44\times10^{-4}w_p^{1.5}\right)\right] \qquad (2.2.1.1b)$$

where

w_p is the weight percent of sodium chloride (–) ; and
T is the temperature (°F).

The saturation pressure is given by:

$$P_{sat} = 22088\exp\left[\frac{374.136 - T''}{T'}\sum_{i=1}^{8}A_i\left[0.65 - 0.01T'\right]^{i-1}\right] \qquad (2.2.1.1c)$$

where

P_{sat} is the saturation pressure (kPa); and
T'' is the temperature (°C).

The coefficients are $A_1 = -7.419242$; $A_2 = -0.29721$; $A_3 = -0.1155286$; $A_4 = -0.008685635$; $A_5 = 0.001094098$; $A_6 = 0.00439993$; $A_7 = 0.002520658$; $A_8 = 0.0005218684$. The range of validity of this viscosity function is:

$$0 < T < 300\ °C; \quad P_{sat} < P < 8\times10^7\ Pa; \quad 0.0887 \le P_{sat} \le 2.2\times10^7\ Pa; \quad 0 \le w_p \le 25\ \%$$

Equations 2.2.1.1a and 2.2.1.1c are used in the case of only thermal simulation while equation 2.2.1.1b is included for solute simulation.

No statement of accuracy was made by Hewlett-Packard (1985) in connection with the ranges of validity of the input variables. Other viscosity functions of pressure, temperature, and solute concentration may be more suitable for certain situations.

2.3 BOUNDARY CONDITIONS

2.3.1 Evapotranspiration Boundary Conditions

An evapotranspiration boundary condition has the property that the fluid flux is a function of the water-table depth below the land surface. The mathematical treatment of evapotranspiration boundaries is based on the following simplifying assumptions: (1) the maximum evapotranspiration flux occurs when the water table is at the land surface; (2) the evapotranspiration flux is zero when the water table is below a specified extinction depth; and (3) for water-table elevations between the extinction depth and the land surface, the evapotranspiration flux varies linearly with water-table elevation. Partially saturated flow effects between the water table and the land surface are neglected. Under these assumptions, the evapotranspiration boundary flux is given by:

$$q_{ET} = \begin{cases} 0, & \text{for } \eta \le 0, \\ \eta q_{max}, & \text{for } 0<\eta<1, \\ q_{max}, & \text{for } \eta \ge 1, \end{cases} \quad \text{for } \mathbf{x} \text{ on } S^7; \tag{2.3.1.1a}$$

with

$$\eta = \frac{z_{FS} - z_{EX}}{z_{LS} - z_{EX}};$$

$$= \frac{z_{FS} - (z_{LS} - b_{ET})}{b_{ET}}; \tag{2.3.1.1b}$$

where

q_{ET} is the fluid flux across the evapotranspiration boundary (m^3/m^2–s);
q_{max} is the maximum evapotranspiration flux (m^3/m^2–s);
η is the attenuation parameter (–);
z_{FS} is the elevation of the water-table free-surface (m);

z_{LS} is the elevation of the land surface (m);
z_{EX} is the elevation of the extinction depth (m);
b_{ET} is the extinction depth below land surface (m); and
S^7 is the region boundary surface over which an evapotranspiration boundary condition exists (m^2).

Figure 2.1 shows the evapotranspiration ßux as a function of water-table depth below land surface. The evapotranspiration ßux is negative because it represents a loss of ßuid from the ground-water system. Although the evapotranspiration boundary condition has a piecewise linear function of evapotranspiration ßux with depth of water table below land surface, the overall function is nonlinear because of the sharp breaks in function slope at the land surface and at the extinction depth.

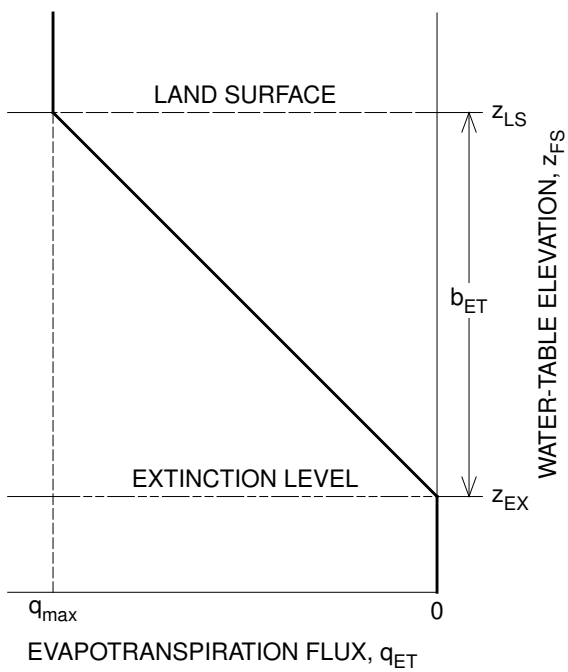


Figure 2.1. Evapotranspiration ßux as a function of water-table depth below land surface.

An evapotranspiration boundary can be adjacent to any part of the simulation region that represents land surface. The terms q_{max}, z_{LS}, and b_{ET} are speciÞed functions of position along the evapotranspiration boundary; q_{max} also can be a function of time. The mass ßux is calculated using the resident density at the evapotranspiration boundary since the ßux is always out from the simulation region.

The heat and solute evapotranspiration ßuxes are assumed to be purely advective. They are obtained from resident enthalpies and resident mass fractions at the evapotranspiration boundary of the simulation region. Thus:

$$q_{HET} = H_B \rho_B q_{ET}, \text{ for } \mathbf{x} \text{ on } S^7; \text{ and} \tag{2.3.1.2}$$

$$q_{SET} = w_B \rho_B q_{ET}, \text{ for } \mathbf{x} \text{ on } S^7; \tag{2.3.1.3}$$

where

q_{HET} is the heat flux across the evapotranspiration boundary (W/m^2);
H_B is the resident specific enthalpy of the fluid at the region boundary (J/kg);
ρ_B is the resident density of the fluid at the evapotranspiration boundary (kg/m^3);
q_{SET} is the solute flux across the evapotranspiration boundary (kg/m^2–s);
w_B is the resident solute mass fraction at the region boundary (-).

Note that H_B and w_B are functions of position along the evapotranspiration boundary and time.

2.3.2 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. 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,

$$p = 0, \text{ on } S^6(\mathbf{x},t); \tag{2.3.2.1}$$

where

S^6 is the free-surface location that varies in space and time (m^2);

is employed, but the kinematic boundary condition is neglected. The absolute pressure on the free surface is atmospheric, so the relative pressure is zero. The location of the free surface is determined by interpolation in the calculated pressure field to determine the location where equation 2.3.2.1 is satisfied. Under this approximation, 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. The fluid- and porous-matrix compressibilities usually are taken to be zero for unconfined flow systems. In these systems, the specific yield becomes equivalent to the effective porosity.

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 that part of the boundary of the region, whereas boundary pressures greater than atmospheric imply that the free surface is above that part of the boundary. The free surface is allowed to rise above the region boundary as well as move to any elevation within the simulation region.

2.4 INITIAL CONDITIONS

This heat- and solute-transport simulator solves only the transient forms of the ground-water flow and the two transport equations; thus, initial conditions are necessary to begin a simulation. Mathematically, the initial conditions are given in the HST3D manual for Version 1 (Kipp, 1987). Several options are available for specification of initial conditions.

For the flow equation, an initial-pressure distribution within the region needs to be specified. The simplest initial condition is an hydrostatic condition, with the pressure specified at a given elevation. A more complex condition is an initial water-table elevation distribution specified over a horizontal plane of the simulation region. Then, an initial pressure distribution will be calculated based on the initial density field. The most general initial condition is a node-by-node pressure specification, perhaps obtained from a previous simulation. No option exists to directly specify a velocity field as an initial condition.

For the heat-transport equation, the initial-temperature field needs to be specified. Again, this can be done as a function of position, or linearly interpolated along a selected coordinate direction. Interpolation in the vertical direction can be used to create an initial geothermal profile.

For the solute-transport equation, the initial mass-fraction field needs to be specified. This can be done by specifying values as a function of position, either by uniform zones, node-by-node, or by linear interpolation.

Pressure, temperature, and mass-fraction fields calculated by one simulation can be used as the initial conditions for another simulation using the file for post-process visualization, as described in section 6.3. 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 is reasonable for the physical situation being simulated.

3. NUMERICAL IMPLEMENTATION

This chapter covers the additions and modifications made to the HST3D simulator pertaining to the numerical calculations. New features concerning equation discretization, boundary conditions, initial conditions, and equation solution are described.

3.1 EQUATION DISCRETIZATION

The classical method of finite differences (Patankar, 1980) is used to discretize the partial-differential equations and boundary conditions in space and time. Several options are available for the differencing.

3.1.1 Spatial Discretization

Although no new features for the spatial discretization have been introduced, a brief review of terminology is given here for reference. Spatial discretization is done by constructing a mesh or grid of node points that covers the simulation region (Fig. 3.1 of Kipp, 1987). This grid of node points is formed by specifying the distribution of nodes in each of the three coordinate directions for the Cartesian-coordinate case and of the two coordinate directions for the cylindrical-coordinate case. A cell is the volume associated with each node. It is a rectangular prism in Cartesian coordinates and an annulus with rectangular cross section in cylindrical coordinates. The simulation region for a single-well in cylindrical coordinates is an annulus with inner radius at the well bore. The cell boundaries are planes that bisect the distance between adjacent node points. Regional boundaries are represented by planes of node points. This type of mesh or grid is called a point-distributed grid.

Other regional volume subdivisions are defined (Fig. 3.3 of Kipp, 1987). An element is the volume bounded by eight corner nodes in three-dimensional Cartesian coordinates. The element is the minimum volume with uniform porous-medium properties. A zone is a contiguous set of elements with the same porous-medium properties. Zones must be rectangular prisms.

A common alternative method for constructing the mesh is to specify the locations of the planes that form the cell boundaries. Their intersections form the cells; then the node points are located in the center of each cell. This type of grid is called a cell-centered or block-centered grid.

The finite-difference equations are formed by integrating the partial differential equations over each cell volume, as explained in Kipp (1987). This means that variations in porous-medium properties over a cell are averaged in proportion to their sub-volume in the cell. Conductance properties are averaged in proportion to their sub-facial area over a given cell face.

3.1.2 Automatic Time-Step Algorithm

Manual time-step selection can be difficult, particularly when many source terms and boundary conditions change frequently 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 a revised automatic time-step option that uses an empirical algorithm from Aziz and Settari (1979, p. 403). This revised algorithm is more robust than that used in Version 1, because it is less susceptible to excessive frequency of time-step adjustment and convergence failure due to a poor time-step selection. 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:

$$\text{if } \left|\delta u_{max}\right| > \delta u^{s}_{max} \text{ or, if } \left|\delta u_{max}\right| < 0.9\delta u^{s}_{max};\ \delta t = \delta t_0 \frac{\delta u^{s}_{max}}{\left|\delta u_{max}\right|}; \tag{3.1.1.1a}$$

with the constraint that,

$$\delta t_{min} \le \delta t \le \min(2\delta t_{0,}\, \delta t_{max}); \tag{3.1.1.1b}$$

where

u	is pressure, temperature, or mass fraction (appropriate units);
δu^{s}_{max}	is the specified maximum change in u (appropriate units);
δt	is the new time step (s);
δt_0	is the previous time step (s); and
$\left|\delta u_{max}\right|$	is the absolute value of the maximum-calculated change in u over the previous time step (appropriate units).

Because the actual changes in the dependent variables are not directly proportional to the changes in time step, the specified maximum changes can be considerably exceeded with an adjusted time step based on a linear predictor. Therefore, an additional test is made at the end of the numerical solution of each system equation:

$$\text{if} \left|\delta u_{max}\right|^{n+1} > 1.5\delta u^{s}_{max}; \tag{3.1.1.2}$$

then the time step is repeated with a new δt calculated from equations 3.1.1.1a and 3.1.1.1b.

The new time step is selected to be the minimum of the three that were calculated on the basis of changes in pressure, temperature, and mass fraction. 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 of the simulation and after boundary-condition changes occur.

If the changes in the dependent variables after the third time step exceed the maximum acceptable values specified, the simulator stops because it cannot reduce the time step below the minimum value. The remedy to this problem is to increase the maximum changes acceptable in the dependent variables or reduce the minimum time step allowed, depending on which approach results in a sufficiently accurate simulation.

When a time step is repeated with an adjusted time-step value, the new value does not get incorporated into boundary conditions that depend on the time-step length. This is a limitation of the simulator that may be removed in a future release.

3.2 BOUNDARY CONDITIONS

All boundary conditions are specified on a cell-by-cell rather than on a zone-by-zone or element-by-element basis. Thus, flux boundary conditions are specified for an entire cell face associated with a

node location. Changes in boundary-condition type or boundary-condition parameters occur at a cell boundary rather than at a node. The default-boundary condition is no dispersive or advective flux through the faces of the cell at the boundary of the region. Cells at edges and corners of the active simulation region may have 2 or 3 boundary faces. 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. However, it is a limitation of the simulator that different parameters for the same type of boundary condition cannot be applied to different boundary faces of a given cell. In other words, the associated temperature and concentration at a specified-pressure boundary cell cannot be different for each face. Similarly, the parameters associated with any of the flux boundary conditions must be the same for all boundary faces of a given cell.

It is presently required that only one flux-type or leakage-type or aquifer-influence-type boundary condition be specified for a given cell. For example, along edges where a lateral leakage-boundary face meets a vertical river leakage-boundary face, one of those faces must be pulled back from that edge so there are not two leakage-boundary condition faces for that cell. This restriction may be removed in a future release of the simulator.

3.2.1 Leakage-Boundary Condition

The leakage-boundary condition has been modified to allow for a reduction in saturated thickness of the boundary cells in the case of lateral leakage with an unconfined-flow condition. Equations 2.5.3.1.1a-c and 2.5.3.2.1a-c from Kipp (1987) become, for boundary cell m:

$$Q_{Lm} = \frac{k_{Lm}}{\mu_{Lm} b_{Lm}} \left[(\rho_e \phi_e)_m - (p_m^n + \rho_m^n g z_m) - (\rho_e - \rho_m^n) g (z_e + z_m)/2 \right] f_m S_{BLm}$$

$$- \frac{k_{Lm}}{\mu_{Lm} b_{Lm}} f_m S_{BLm} \delta p_m; \qquad (3.2.1.1)$$

where

Q_{Lm} is the volumetric flow rate at a leakage boundary (m^3/s);
k_{Lm} is the permeability of the leakage layer (m^2) ;
μ_{Lm} is the viscosity of the fluid in the leakage layer (Pa–s) ;
b_{Lm} is the thickness of the leakage layer (m) ;
ρ_e is the density of the fluid density in the outer aquifer (kg/m^3);
ρ_m is the fluid density at the simulation-region boundary at cell m (kg/m^3);
ϕ_e is the potential energy per unit mass of the fluid in the outer aquifer (N–m/kg) ;
p_m is the pressure at the simulation-region boundary at cell m (Pa);
z_e is the elevation at the top of the leakage layer (m);
z_m is the elevation of the simulation-region boundary at cell m (m);
g is the gravitational constant (m/s^2) ;
f_m is the fraction of cell saturation for boundary cell m (–); and
S_{BLm} is the part of the boundary cell surface that is a leakage boundary (m^2).

The leakage-flow rate, (eq. 3.2.1.1), has an explicit term for the right-hand-side of the discretized

system-flow equation and an implicit factor for the left-hand-side. The fraction of cell saturation attenuates the leakage flow when the boundary cell is not fully saturated, and is updated at the end of each time step. The fraction is applied explicitly to the subsequent time step.

3.2.1.1 River-Leakage Boundary The river-leakage boundary condition has been extended to allow for multiple branches of a river to join in a cell. Up to eight branches may meet in a given cell. Each of the parameters associated with a river branch are averaged for all branches that meet in a given cell to obtain the net influence of river leakage for that cell.

The river-bottom elevation determines to which cell a segment of river is connected in the simulation region. This allows for a wide range of topography to be reflected in the river-bottom elevation incorporated into the simulator. However, rivers with extensive vertical sides that contact multiple layers of cells cannot be represented in this version of HST3D. When the water table drops to a cell below the cell containing the river bottom, the leakage flux is directed to the cell containing the water table.

A river-leakage boundary condition and a specified-flux boundary condition can be applied to the same face of a given cell because they are different types of boundary conditions. This feature is needed to represent river leakage over part of a cell area and precipitation infiltration over the remaining area.

3.2.2 Evapotranspiration Boundary Condition

The evapotranspiration boundary condition is transformed into a sink term in a similar fashion to leakage conditions since they both are a function of pressure. Equations 2.3.1.1a and 2.3.1.1b applied on a discrete grid become, for boundary cell m:

$$Q_{ETm} = \begin{cases} 0, & \text{for } \eta \le 0; \\ \left[\eta^n q_{max} - q_{max}\dfrac{\partial \eta}{\partial p}\delta p_m\right] S_{BETm}, & \text{for } 0<\eta<1; \\ q_{max} S_{BETm}, & \text{for } \eta \ge 1; \end{cases} \qquad (3.2.2.1)$$

where

Q_{ETm} is the volumetric flow rate at an evapotranspiration boundary (m^3/s); and

S_{BETm} is the part of the boundary cell surface that is an evapotranspiration boundary (m^2).

The evapotranspiration flow rate is a nonlinear function of η because of the sharp breaks in function slope at the land surface and at the extinction depth. The evapotranspiration flow rate (equation 3.2.2.1) for $0<\eta<1$ has an explicit term for the right-hand-side of the discretized system-flow equation and an implicit factor for the left-hand-side. The other equations only have an explicit term. The evapotranspiration flow rate is applied to the cells containing the free-surface boundary.

3.2.3 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 cells containing the water table.

A new algorithm for the free-surface boundary condition has been implemented. The location of the free surface within a cell that contains the water table is established by linear interpolation to locate the elevation of zero (atmospheric) pressure. The interpolation is done between two adjacent nodes in the vertical direction, one of which has a positive pressure and the other which has a negative pressure. That is, one node is within the saturated zone and the other is outside it. In the case of the free-surface being located above the uppermost plane of nodes (outside the simulation region), its elevation is determined by extrapolating the pressures of the uppermost node and the adjacent node below. The algorithm allows for a flux through the bottom of each cell containing the free surface. When a cell with negative pressure is dry, and is adjacent to a cell containing the free surface, the pressure of the dry cell is kept up-to-date during the simulation by extrapolating the pressure gradient based on the pressure values of the two cells below. This algorithm is valid only for coordinate systems with the z-axis pointing vertically upward.

The free surface is allowed to rise above the upper boundary of the simulation region. However, large rises of the free surface indicate the need for more cells to be added at the top of the region to accurately represent the aquifer saturated thickness. When designing the grid for a free-surface boundary problem, the location of the uppermost layer of cells should accommodate the maximum rise in the free-surface or water-table location. However, in the special case of a confining layer at the top of the simulation region, no conversion to confined flow conditions is made if the free surface does rise above the uppermost plane of nodes. In the case of confined flow, no conversion to unconfined flow conditions is made if the free surface does drop below the uppermost plane of nodes. In addition, a given simulation region can only be unconfined or confined and not have each of these conditions present over different subareas.

The free surface may fall to any cell below the top of the simulation region. Dry cells are isolated from the ground-water flow region by zero conductance factors for their lateral faces. A dry cell is resaturated by flux through its bottom boundary. Thus, if a column of cells becomes totally dewatered, there is no mechanism to resaturate it. A cell that becomes dry is removed from the computational region at the end of the time step. A cell that becomes rewetted is added back into the computational region also at the end of the time step. The saturation of a cell that went dry during a time step is constrained to be a small value for the remainder of that time step. Because the free-surface position is treated explicitly, it is only adjusted at the end of each time step. This algorithm causes linkage between storage terms of a given cell and its vertically adjacent cells. Storage terms that are coefficients of pressure changes of the adjacent cells are applied to the appropriate off-diagonal terms of the finite difference equations.

To obtain the appropriate storage factors for the discretized flow equation (3.1.4.1a in Kipp, 1987) at the free-surface boundary cells, the terms of equation 3.1.1.8 in Kipp (1987) are evaluated, this time including the saturation fraction of equation 3.4.6.1 in Kipp (1987). Using equation 3.1.3.5 in Kipp (1987), and also assuming that the porosity is constant, the fluid compressibility is zero and isothermal conditions exist, the following storage factors are obtained:

$$C_{33} \equiv \rho^{n+1} \frac{2}{z_{k+1} - z_{k-1}} \left[\frac{(z_k - z_{k+1})p_{k+1}}{(p_k - p_{k+1})^2} \right] \sum_{s=1}^{8} \varepsilon_s V_s / \delta t \tag{3.2.3.1a}$$

$$E_{fp} \equiv \rho^{n+1} \frac{2}{z_{k+1} - z_{k-1}} \left[\frac{(z_{k+1} - z_k)p_k}{(p_{k+1} - p_k)^2} \right] \sum_{s=1}^{8} \varepsilon_s V_s / \delta t \tag{3.2.3.1b}$$

$$C_{fp} \equiv \rho^{n+1} \frac{2}{z_{k+1} - z_{k-1}} \left[\frac{(z_{k-1} - z_k)p_k}{(p_{k-1} - p_k)^2} \right] \sum_{s=1}^{8} \varepsilon_s V_s / \delta t \tag{3.2.3.1c}$$

$$C_{32} \equiv 0 \tag{3.2.3.1d}$$

$$C_{31} \equiv f^n \rho_0 \beta_w \sum_{s=1}^{8} \varepsilon_s V_s / \delta t \tag{3.2.3.1e}$$

where

C_{33}, C_{32}, C_{31} are the fluid storage factors for cell m (appropriate units);
C_{fp} are the fluid storage factors for the adjacent cell below cell m (appropriate units);
E_{fp} are the fluid storage factors for the adjacent cell above cell m (appropriate units);
ε_s is the porosity for cell subdomain s (–);
V_s is the volume of cell subdomain s (m^3);
f^n is the fraction of cell saturation at time t^n (–); and
β_w is the coefficient of density variation with solute mass fraction (–).

Using the same procedure as for equations 3.2.3.1a-e, the corresponding factors the solute-transport equation (3.1.4.5a in Kipp, 1987) are:

$$C_{13} \equiv \rho^{n+1} w^n \frac{2}{z_{k+1} - z_{k-1}} \left[\frac{(z_k - z_{k+1})p_{k+1}}{(p_k - p_{k+1})^2} \right] \sum_{s=1}^{8} K_s V_s / \delta t \tag{3.2.3.2a}$$

$$E_{sp} \equiv \rho^{n+1} w^n \frac{2}{z_{k+1} - z_{k-1}} \left[\frac{(z_{k+1} - z_k)p_k}{(p_{k+1} - p_k)^2} \right] \sum_{s=1}^{8} K_s V_s / \delta t \tag{3.2.3.2b}$$

$$C_{sp} \equiv \rho^{n+1} w^n \frac{2}{z_{k+1} - z_{k-1}} \left[\frac{(z_{k-1} - z_k)p_k}{(p_{k-1} - p_k)^2} \right] \sum_{s=1}^{8} K_s V_s / \delta t \tag{3.2.3.2c}$$

$$C_{12} \equiv 0 \tag{3.2.3.2d}$$

$$C_{11} \equiv \left[f^{n+1} \rho^{n+1} + w^n f^n \rho_0 \beta_w \right] \sum_{s=1}^{8} K_s V_s / \delta t \tag{3.2.3.2e}$$

$$K_s \equiv \varepsilon_s + (\rho_b K_d)_s; \tag{3.2.3.2f}$$

where

C_{13}, C_{12}, C_{11} are the solute storage factors for cell m (appropriate units);
C_{sp} are the solute storage factors for the adjacent cell below cell m (appropriate units);
E_{sp} are the solute storage factors for the adjacent cell above cell m (appropriate units);
K_d is the equilibrium distribution coefficient (m^3/kg) ; and
ρ_b is the bulk density of the porous medium (kg/m^3).

Since the free-surface boundary condition is implemented only for isothermal conditions, the storage factors, C_{32} and C_{12} are identically zero. 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 conductance and source 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.

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 region. Then that flux is applied to the cells that contain the free surface at any given time. The associated mass fraction determines the amount of solute that enters the saturated region through the free surface.

3.2.4 Simultaneous Specified-Value Boundary Conditions

In the case of confined flow, the simulator algorithm does not correctly assemble the modified flow equation and the modified heat transport equation when a specified-pressure condition and a specified-temperature or a specified-mass-fraction condition are applied along the same boundary. This limitation may be removed in a future version of the simulator. Simulations with a free-surface boundary condition are unaffected by this limitation.

3.3 INITIAL CONDITIONS

The initial pressure distribution is based on the initial fluid density distribution, which is a function of the initial temperature and solute-mass fraction distributions. In addition, the hydrostatic-equilibrium pressure distribution takes the fluid compressibility into account unless an unconfined aquifer is being simulated. The calculation of the initial pressure distribution 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 elevation of the water-table surface, when employed, is specified for the upper layer of nodes only, but the elevation at each node in a horizontal plane can be specified to be at any vertical location. Hydrostatic equilibrium is assumed for the computation of the pressure distribution elsewhere in the simulation region based on the water-table elevation, initial temperature, and solute-mass fraction distributions.

3.4 EQUATION SOLUTION

The three sets of difference equations are solved sequentially with iteration for a given time step. The convergence criterion is based on the fractional change in density over an iterative cycle as described in section 3.6.2 of Kipp (1987).

3.4.1 Modification of the Flow and Transport Equations

In the case of unconfined ground-water flow, the modification of the finite-difference equations by Gauss elimination described in section 3.6.1 of Kipp (1987) is not performed. This avoids numerical problems associated with the solute transport-equation and the mass balance calculation that occur with

the modified difference equation when a cell goes dry during a time step.

3.5 MATRIX EQUATION SOLVERS

A new iterative solution algorithm for the linear, sparse-matrix equations has been developed. It is a generalized conjugate-gradient method with preconditioning based on incomplete lower-upper (ILU) triangular factorization as described by Meijerink and van der Vorst (1977), Elman (1982), and Behie and Forsyth (1984). This generalized conjugate gradient algorithm is also known as Orthomin with ILU preconditioning and has been used in petroleum reservoir simulation (Behie and Forsythe, 1984). The application of boundary conditions and simulation-region geometry for the generalized conjugate gradient solver is done by solving a trivial equation for the excluded cells and specified-value cells.

3.5.1 Equation Preconditioning

ILU factorization is a powerful preconditioning technique for use with iterative methods applied to sparse linear systems of equations. The preconditioned system is

$$\mathbf{MA}x = \mathbf{M}b \tag{3.5.1.1}$$

where **M** is an approximate inverse of matrix **A**. For ILU preconditioning,

$$\mathbf{M} \approx [\mathbf{LU}]^{-1} \tag{3.5.1.2}$$

where **L** and **U** are lower and upper triangular factors of matrix **A**. Different orders of ILU factorization can be employed (Behie and Forsythe, 1984). Order one means no fill-in is allowed of any null element locations in the original **A** matrix during the LU factorization. That is the method used in the present version of HST3D. Order one ILU factorization corresponds with method ICCG(1,1), of Meijerink and van der Vorst (1981).

3.5.2 Reduced Matrix Equation

The algorithm employed in HST3D solves a reduced matrix equation, also known as the Schur complement (Axelsson, 1994), obtained by reordering the nodes using either a red-black renumbering scheme (Price and Coats, 1974, Aziz and Settari, 1979) or an alternating-diagonal zig-zag renumbering scheme (Kipp and others, 1994).

For red-black renumbering, the nodes are renumbered in sweeps along a primary coordinate direction skipping every other node. The secondary direction is incremented at the end of each sweep and the tertiary direction is incremented at the end of a set of secondary lines. After the first sweep cycle is complete, a second cycle is done, renumbering the remaining nodes in the same fashion. In two dimensions this renumbering is like numbering all the red squares of a checkerboard followed by numbering all the black squares. The selection of the primary, secondary, and tertiary directions is made by the user. However the convergence behavior of the generalized conjugate-gradient solver can be markedly different under different reorderings (Kipp and others, 1992).

The red-black and alternating-diagonal zig-zag renumbering schemes lead to coefficient matrices of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{D}_R & \mathbf{A}_{RB} \\ \mathbf{A}_{BR} & \mathbf{D}_B \end{bmatrix} \tag{3.5.2.1}$$

where $\mathbf{D}_R$ and $\mathbf{D}_B$ are diagonal matrices. Elimination yields the reduced matrix

$$\mathbf{R} = \mathbf{D}_B - \mathbf{A}_{BR}\mathbf{D}_R^{-1}\mathbf{A}_{RB} \tag{3.5.2.2}$$

The **R** matrix is solved with the ILU preconditioned generalized conjugate-gradient solver. The classical conjugate gradient method is not applicable to the unsymmetric matrices resulting from the difference equations for the modified flow and transport equations.

The alternating-diagonal zig-zag renumbering scheme also leads to a reduced matrix. The scheme is described by Kipp and others (1994). It has the property that the solver convergence rate is insensitive to the sequence of directions used for the renumbering. When the red-black renumbering scheme is used, there can be as much as a factor of two between the minimum and maximum number of iterations needed to converge to a solution at any time step depending on the sequence of directions used for the renumbering.

The generalized conjugate-gradient method is restarted at a frequency specified by the user. This avoids excessive computational burden associated with retaining all of the previous search directions.

3.5.3 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 selected. It is difficult to give a definitive rule for the selection of a matrix solver. Storage requirements probably will determine when it is necessary to use an iterative solver. The DIMEN pre-processing program (Appendix A) writes out the storage requirements for each matrix solver for each renumbering method. The HST3D program writes out the storage requirements for the solver selected.

3.5.4 Choosing the Parameters for an Iterative Equation Solver

The generalized conjugate gradient solver is not completely free of user-specified parameters. If the red-black renumbering scheme is selected, it is necessary to experiment with the six possible renumbering sequences to find the optimum choice for fastest convergence. Short simulations of one or two time steps are sufficient to determine the optimum choice. Actually, the six choices usually group into three pairs of nearly equal iteration counts. It is not critical to find the optimum renumbering sequence for the alternating-direction zig-zag renumbering scheme.

A test of the ILU and the modified ILU (MILU) preconditioning methods may reveal a significant difference in convergence rate. Often the MILU preconditioning method gives faster convergence. The solver does not seem to be very sensitive to the number of search directions to be retained between restarts. The default value of 5 seems to work satisfactorily for most problems. If solver convergence requires more than about 100 iterations, the number of retained search directions should be increased to 10 or 15.

The tolerance for convergence of the iterative solver is based on a reduction of the Euclidean norm of the residual vector from the initial value at the start of iteration. The tolerance is the maximum acceptable value of the ratio of the Euclidean norm of the residual vector to the Euclidean norm of the initial residual vector. Experience with test problems has shown that a tolerance of 10^{-6} to 10^{-12} is necessary to obtain 3 or 4 digit agreement with the direct solver. No algorithm is presently available to set this tolerance based on the problem specifications. For large problems, use of the direct solver is impractical; therefore, the user must experiment with several values of the convergence tolerance to determine the largest value which provides the desired number of digits agreement with a more accurate solution obtained with a much smaller tolerance. An iteration limit prevents runaway conditions when the convergence rate becomes very slow. Under some conditions, the user may need to double or triple the default limit of 100. However, more than a few hundred iterations indicates that adjustments probably need to be made in the spatial or temporal discretization.

4. COMPUTER CODE DESCRIPTION

4.1 CODE ORGANIZATION

The HST3D computer code is written in FORTRAN-77. Code conforming to American National Standards Institute (ANSI) standards (American National Standards Institute, 1978) has been used as closely as possible to provide maximum portability. An exception to these standards is the usage of INCLUDE statements. Version 2.0 of the program code consists of a main routine and 75 subroutines. There are 67 files incorporated into the source code by INCLUDE statements. The program length is approximately 16,000 lines of code including comment and continuation lines. A brief description of the routines and their function appears in the following list.

HST3D Main routine that drives the program execution. The basic steps are (1) open files, read, error check, initialize, and write output for problem specifications and space allocation; (2) read, error check, initialize, and write output for static information; (3) read, error check, initialize and write output for transient information; (4) start the time-step by calculating flow- and transport-equation coefficients, applying the boundary conditions; and calculating the source-sink well terms; (5) assemble and solve each of the three equations in turn, iterating to convergence; (6) calculate the summary rates and cumulative amounts; (7) write the output information for the time-step completed; and (8) if desired, dump restart data to a disk file. Then return to step 4 and continue with the next time step until the time for a change in boundary conditions or source terms occurs. At this time return to step 3 for a new set of transient data. Continue until marching through time until the simulation is finished. Then (9) close files and terminate program execution. If errors occur, the error checking that is in progress is completed, but then, information to that point in the simulation is written out, and the computer run is aborted.

The following subroutines are described in their order of execution.

OPENF Requests the user to enter the data-input file name and an identification suffix for the simulation.

READ1 Reads the data specifying the problem and pertaining to determination of computer storage space required.

INIT1 Initializes the total lengths of the common blocks. If necessary, sets the U.S.-customary-to-metric conversion factors and their inverses, and defines the unit labels.

ERROR1 Checks for errors in the data read by READ1, mostly concerning array limits.

WRITE1 Writes information about the dimensions and array-size requirements.

READ2 Reads the data pertaining to all the 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 by calculating additional static data for the simulation.

ERROR4 Checks for errors in the data calculated by INIT2.

WRITE2 Writes the static data to various output files.

COEFF Calculates the coefficients for the flow-, heat- and solute-transport equations at the beginning of the time step. These coefficients include conductances, dispersion coefficients, and interstitial velocities.

WRITE4 Writes the velocity field to output files.
RHSN Calculates the explicit terms for the right-hand-side of the system equations.
READ3 Reads the transient data, including boundary condition and source-sink information, plus time-step, calculation, and printout information.
INIT3 Initializes by calculating the transient data for the simulation.
ERROR3 Checks for errors in the transient data read by READ3 and calculated by INIT3.
WRITE3 Writes the transient data to various files.
WRITE6 Writes the conductance and dispersive conductance coefficients to an output file.
WELLSC or WELLSR Calculates and applies the well source-sink terms for cylindrical or rectangular coordinate systems, respectively.
APLBCE Applies the explicit terms of the boundary conditions to the set of system equations.
TIMSTP Calculates the change in time step for automatic time step control and/or print-time control and completes the global balance calculation for the beginning of the time step.
ITER Assembles and solves the three equations iteratively for the current time step.
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 various files 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 file, at desired time steps or values.
CLOSEF 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 alphabetical order. Some are called from several routines, and some are optional for a given simulation.

APLBCI Applies the implicit terms of the boundary conditions to the set of system equations.
ASEMBL Assembles the coefficients of the modified flow, modified heat-transport, and solute-transport equations at each time step.
BSODE Integrates the coupled ordinary differential equations for pressure and temperature up or down the well-riser casing, using the Bulirsch-Stoer algorithm (Bulirsch and Stoer, 1966) for extrapolation by rational polynomials.
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, evaluated explicitly in the transport equations at each iteration.
D4DES Solves the reduced-matrix equation, obtained by alternating-diagonal reordering, using a direct triangular factorization and back solution algorithm.
D4ORDR Determines the node numbering for the alternating-diagonal (D4) reordering scheme to form the reduced matrix.
D4ZORD Determines the node numbering for the alternating-diagonal zig-zag (D4Z) reordering scheme to form the reduced matrix.
ERRPRT Writes the error messages for a given simulation to a file and the monitor.

ETOM1 Converts static data from U.S. customary units to metric units.

ETOM2 Converts transient data from U.S. customary units to metric units.

IREWI Reads, error checks, echo writes, and initializes integer-array elements that are input as zones of constant values over a rectangular prism of cells.

RBORD Determines the node numbering for the red-black (RB) reordering scheme to form the reduced matrix.

RCGIES Solves the reduced-matrix equation using a preconditioned-generalized-conjugate-gradient algorithm.

REORDR Performs a reference reordering with the chosen renumbering routine using the selected direction sequence, then performs an intermediate reordering. The product of the two is the final reordering. Also loads indexing arrays used by the RCGIES or D4DES solvers.

REWI Reads, error checks, echo writes, and initializes real-array elements that are input as zones of constant values over a rectangular prism of cells, or as node by node distributions, or by linear interpolation of two values.

REWI3 The same as REWI, but for parameters that occur in sets of three, such as vectors.

SBCFLO Calculates the flow rates at specified-value boundary-condition cells for the global balances.

SOR2L Solves the matrix equation or determines the optimum parameters, using a two-line, successive-overrelaxation algorithm, with the lines oriented in a selected coordinate direction.

SORIES Invokes the two-line, successive-overrelaxation solver for each coordinate direction to estimate the optimum overrelaxation parameter, and to solve the equations.

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.

WELRIS Performs the pressure and temperature calculation up or down the well-riser pipe, using simultaneous solution of the two ordinary differential equations.

The following low-level utility subroutines are listed in alphabetical order. They are called many times from various locations.

ABMULT Multiplies a matrix times a vector for the black nodes for the generalized conjugate-gradient equation solver for the reduced matrix equation.

ARMULT Multiplies a matrix times a vector for the red nodes for the generalized conjugate-gradient equation solver for the reduced matrix equation.

DBMULT Multiplies the diagonal of a matrix times a vector for the black nodes for the generalized conjugate-gradient equation solver for the reduced matrix equation.

EHOFTP Calculates enthalpy as a function of temperature and pressure using tabular interpolation.

FORMR Forms the product of the off-diagonal blocks, scales and subtracts from the black diagonal to get the reduced matrix. Used with the generalized conjugate-gradient equation solver for the reduced matrix equation.

INTERP Performs one-dimensional or two-dimensional linear or bilinear tabular interpolation.

LDCI Loads the indexing array, CI, for the RCGIES solver.

LDCIR Loads the indexing array, CIR, for the RCGIES solver.

LDIND	Loads the indexing array, IND, for the RCGIES solver.
LDMAR1	Loads the local indexing array, MAR1, for the reduced matrix equation.
LSOLV	Solves the lower triangular matrix equation of the reduced system.
MTOIJK	Calculates the x-, y-, and z-coordinate indices from the natural node number of a cell.
NINTRP	Performs one-dimensional interpolation of a coordinate value to the nearest node point.
PRNTAR	Writes one-dimensional or two-dimensional arrays in tabular form to various files.
RBMAP	Establishes the red-black renumbering to form the reduced matrix.
RFACT	Factors the reduced matrix according to an incomplete LU factorization.
RFACTM	Factors the reduced matrix according to a modified incomplete LU factorization.
TOFEP	Determines temperature as a function of enthalpy and pressure by tabular interpolation.
USOLV	Solves the upper triangular matrix equation of the reduced system.
VISCOS	Calculates fluid viscosity as a function of pressure, temperature, and solute-mass fraction.
VPSV	Calculates the result of a vector plus a scalar times a vector.
WFDYDZ	Calculates derivatives of pressure and temperature along the well-riser casing for the Bulirsch-Stoer integration algorithm.
XIP	Calculates the inner or dot product of two vectors.
ZONPLT	Creates a character plot of the zones for a printer.

A chart of the calling tree showing the main sequence of subroutine execution 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 in each column. The sequence of execution on each page is from top to bottom of the left column followed by the right column. The factor in front of some routine names indicates multiple calls. Some utility subroutines are listed more than once for graphical clarity.

```
=====================================
 Calling Tree

    Program HST3D
    |
    +--Subroutine OPENF
    |
    +--Subroutine READ1
    |
    +--Subroutine INIT1
    |
    +--Subroutine ERROR1
    |  |
    |  +--Subroutine ERRPRT
    |
    +--Subroutine WRITE1
    |
    +--Subroutine READ2
    |  |
    |  +--6*Function NINTRP
    |  |
    |  +--3*Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine REWI3
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--4*Function NINTRP
    |  |
    |  +--Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine REWI3
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine REWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |  |
    |  |  +--3*Function INTERP
    |  |
    |  +--Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--6*Subroutine REWI
    |     |
    |     +--6*Function NINTRP
    |     |
    |     +--3*Function INTERP
    |
    +--Subroutine ERROR2
    |  |
    |  +--Subroutine ERRPRT
    |
    +--Subroutine INIT2
    |  |
    |  +--Subroutine ETOM1
    |  |
    |  +--Function EHOFTP
    |  |  |
    |  |  +--2*Function INTERP
    |
    +--5*Function NINTRP
    |
    +--Subroutine D4ORDR
    |
    +--Subroutine REORDR
    |  |
    |  +--Subroutine RBORD
    |  |  |
    |  |  +--Function RBMAP
    |  |
    |  +--Subroutine D4ZORD
    |  |
    |  +--Subroutine LDIND
    |  |
    |  +--Subroutine LDCI
    |  |
    |  +--Subroutine LDCIR
    |  |
    |  +--Subroutine LDMAR1
    |
    +--3*Function INTERP
    |
    +--Function VISCOS
    |
    +--Function EHOFTP
       |
       +--2*Function INTERP
|
+--Subroutine ERROR4
|  |
|  +--Subroutine PRNTAR
|
+--Subroutine WRITE2
|  |
|  +--5*Subroutine PRNTAR
|  |
|  +--2*Subroutine ZONPLT
|  |
|  +--47*Subroutine PRNTAR
|
+--Subroutine COEFF
|
+--Subroutine WRITE4
|  |
|  +--3*Subroutine PRNTAR
|
+--Subroutine RHSN
|  |
|  +--Function EHOFTP
|     |
|     +--2*Function INTERP
|
+--Subroutine READ3
|  |
|  +--5*Subroutine REWI
|  |  |
|  |  +--6*Function NINTRP
|  |  |
|  |  +--3*Function INTERP
|  |
|  +--Subroutine REWI3
|  |  |
|  |  +--6*Function NINTRP
|  |
|  +--3*Subroutine REWI
```

Figure 4.1. Connection chart for the HST3D main program and subprograms showing routine hierarchy.

```
|   |   |
|   |   +--6*Function NINTRP
|   |   |
|   |   +--3*Function INTERP
|   |
|   +--3*Subroutine REWI3
|   |   |
|   |   +--6*Function NINTRP
|   |
|   +--2*Subroutine REWI
|   |   |
|   |   +--6*Function NINTRP
|   |   |
|   |   +--3*Function INTERP
|   |
|   +--4*Function NINTRP
|   |
|   +--4*Subroutine REWI
|       |
|       +--6*Function NINTRP
|       |
|       +--3*Function INTERP
|
+--Subroutine INIT3
|   |
|   +--Subroutine ETOM2
|
+--Subroutine ERROR3
|   |
|   +--Subroutine ERRPRT
|   |
|   +--Subroutine PRNTAR
|
+--Subroutine WRITE3
|   |
|   +--17*Subroutine PRNTAR
|
+--Subroutine WRITE6
|   |
|   +--21*Subroutine PRNTAR
|
+--Subroutine WELLSC
|   |
|   +--Subroutine WELRIS
|   |   |
|   |   +--Subroutine BSODE
|   |   |   |
|   |   |   +--3*Subroutine WFDYDZ
|   |   |       |
|   |   |       +--Function VISCOS
|   |   |
|   |   +--Function EHOFTP
|   |       |
|   |       +--2*Function INTERP
|   |
|   +--2*Function TOFEP
|   |   |
|   |   +--6*Function INTERP
|   |
|   +--Function VISCOS
|   |
|   +--Function EHOFTP
|   |   |
|   |   +--2*Function INTERP
|   |
|   +--2*Function TOFEP
|   |   |
|   |   +--6*Function INTERP
```

```
|   |
|   +--Function VISCOS
|   |
|   +--3*Subroutine WELRIS
|       |
|       +--Subroutine BSODE
|       |   |
|       |   +--3*Subroutine WFDYDZ
|       |       |
|       |       +--Function VISCOS
|       |
|       +--Function EHOFTP
|           |
|           +--2*Function INTERP
|
+--Subroutine WELLSR
|   |
|   +--Subroutine WELRIS
|   |   |
|   |   +--Subroutine BSODE
|   |   |   |
|   |   |   +--3*Subroutine WFDYDZ
|   |   |       |
|   |   |       +--Function VISCOS
|   |   |
|   |   +--Function EHOFTP
|   |       |
|   |       +--2*Function INTERP
|   |
|   +--2*Function TOFEP
|   |   |
|   |   +--6*Function INTERP
|   |
|   +--Function EHOFTP
|   |   |
|   |   +--2*Function INTERP
|   |
|   +--2*Function TOFEP
|   |   |
|   |   +--6*Function INTERP
|   |
|   +--3*Subroutine WELRIS
|       |
|       +--Subroutine BSODE
|       |   |
|       |   +--3*Subroutine WFDYDZ
|       |       |
|       |       +--Function VISCOS
|       |
|       +--Function EHOFTP
|           |
|           +--2*Function INTERP
|
+--Subroutine APLBCE
|   |
|   +--Function EHOFTP
|       |
|       +--2*Function INTERP
|
+--Subroutine TIMSTP
|
+--Subroutine ITER
|   |
|   +--Subroutine ASEMBL
|   |   |
|   |   +--Subroutine MTOIJK
|   |   |
|   |   +--Subroutine CALCC
```

Figure 4.1. Connection chart for the HST3D main program and subprograms showing routine hierarchy— Continued.

```
|   |   |
|   |   +--Subroutine CRSDSP
|   |   |   |
|   |   |   +--Subroutine MTOIJK
|   |   |
|   |   +--Subroutine MTOIJK
|   |   |
|   |   +--2*Subroutine CALCC
|   |
|   +--Subroutine APLBCI
|   |   |
|   |   +--5*Function EHOFTP
|   |       |
|   |       +--2*Function INTERP
|   |
|   +--Subroutine D4DES
|   |
|   +--Subroutine SORIES
|   |   |
|   |   +--2*Subroutine SOR2L
|   |
|   +--Subroutine RCGIES
|   |   |
|   |   +--Function XIP
|   |   |
|   |   +--Subroutine FORMR
|   |   |
|   |   +--Subroutine RFACT
|   |   |
|   |   +--Subroutine RFACTM
|   |   |
|   |   +--Subroutine ARMULT
|   |   |
|   |   +--Subroutine VPSV
|   |   |
|   |   +--Subroutine ABMULT
|   |   |
|   |   +--Subroutine ARMULT
|   |   |
|   |   +--Subroutine DBMULT
|   |   |
|   |   +--2*Subroutine VPSV
|   |   |
|   |   +--Subroutine LSOLV
|   |   |
|   |   +--Subroutine USOLV
|   |   |
|   |   +--Subroutine ABMULT
|   |   |
|   |   +--Subroutine ARMULT
|   |   |
|   |   +--Subroutine DBMULT
|   |   |
|   |   +--Subroutine VPSV
|   |   |
|   |   +--2*Function XIP
|   |   |
|   |   +--2*Subroutine VPSV
|   |   |
|   |   +--Function XIP
|   |   |
|   |   +--Subroutine LSOLV
|   |   |
|   |   +--Subroutine USOLV
|   |   |
|   |   +--Subroutine ABMULT
|   |   |
|   |   +--Subroutine ARMULT
|   |   |
|   |   +--Subroutine DBMULT
|   |   |
|   |   +--Subroutine VPSV
|   |   |
|   |   +--Function XIP
|   |   |
|   |   +--2*Subroutine VPSV
|   |   |
|   |   +--Subroutine ABMULT
|   |   |
|   |   +--Subroutine VPSV
|   |   |
|   |   +--Function XIP
|   |   |
|   |   +--2*Subroutine VPSV
|   |   |
|   |   +--Subroutine ABMULT
|   |   |
|   |   +--2*Subroutine VPSV
|   |
|   +--Subroutine SBCFLO
|   |
|   +--Subroutine WBCFLO
|   |   |
|   |   +--4*Function TOFEP
|   |       |
|   |       +--6*Function INTERP
|
+--Subroutine SUMCAL
|   |
|   +--3*Subroutine MTOIJK
|   |
|   +--Function VISCOS
|   |
|   +--Function EHOFTP
|   |   |
|   |   +--2*Function INTERP
|   |
|   +--Subroutine WBBAL
|   |   |
|   |   +--2*Function TOFEP
|   |   |   |
|   |   |   +--6*Function INTERP
|   |   |
|   |   +--Function EHOFTP
|   |   |   |
|   |   |   +--2*Function INTERP
|   |   |
|   |   +--2*Function TOFEP
|   |   |   |
|   |   |   +--6*Function INTERP
|   |   |
|   |   +--Subroutine WELRIS
|   |       |
|   |       +--Subroutine BSODE
|   |       |   |
|   |       |   +--3*Subroutine WFDYDZ
|   |       |       |
|   |       |       +--Function VISCOS
|   |       |
|   |       +--Function EHOFTP
|   |           |
|   |           +--2*Function INTERP
|   |
|   +--2*Subroutine SBCFLO
|   |
|   +--4*Function EHOFTP
|       |
|       +--2*Function INTERP
|
+--Subroutine WRITE5
|   |
|   +--13*Subroutine PRNTAR
|   |
|   +--Function EHOFTP
```

Figure 4.1. Connection chart for the HST3D main program and subprograms showing routine hierarchy— Continued.

```
   |   |   |
   |   |   +--2*Function INTERP
   |   |
   |   +--15*Subroutine PRNTAR
   |   |
   |   +--Function EHOFTP
   |       |
   |       +--2*Function INTERP
   |
   +--3*Subroutine DUMP
   |
   +--Subroutine CLOSEF
       |
       +--2*Subroutine ERRPRT

=====================================
```

Figure 4.1. Connection chart for the HST3D main program and subprograms showing routine hierarchy— Continued.

4.2 MEMORY ALLOCATION, ARRAY-SIZE REQUIREMENTS, AND SUBPROGRAM COMMUNICATION

The following principles were used in the design of subprogram communication for HST3D Version 2. Low-level utility programs are called with argument lists. Subprogram communication with arguments is also used for arrays whose identity changes for different calls to the same routine. Most variables and arrays are communicated to subprograms through global common blocks. All global arrays are dimensioned in their common block declaration. 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 dimensioned to a maximum length by Fortran PARAMETER statements. Using PARAMETER statements to dimension a class of arrays means that only one number needs to be set. Each PARAMETER statement is in a separate Þle. The INCLUDE statement causes these Þles to be incorporated into each subroutine with the given class of arrays, thus setting the dimensioning parameter for compilation. Some array limits are set by calculation within PARAMETER statements. Some of the arrays share storage space if they are used sequentially and the contents do not need to be retained for the duration of the simulation.

The limits required for array dimensioning and the names of the PARAMETER-statement Þles that contain them appear in the following list.

LX,LY,LZ Maximum number of nodes in the x-, y-, and z-directions (`param6.inc`).

LXYZ Maximum total number of nodes. Needs to be at least the product of $LX \times LY \times LZ$ (`param1.inc`).

LPMZ Maximum number of porous media zones (`param2.inc`).

LSBC Maximum number of cells (nodes) with a speciÞed-value boundary condition (`param3.inc`).

LFBC Maximum number of cell faces with a speciÞed-ßux boundary condition (`param3.inc`).

LLBC Maximum number of cell faces with a leakage boundary condition (`param3.inc`).

LAIFC Maximum number of cell faces with an aquifer-inßuence-function boundary condition (`param3.inc`).

LHCBC Maximum number of cell faces with a heat-conduction boundary condition (`param3.inc`).
LHCN Maximum number of nodes for the finite-difference solution of the heat-conduction boundary condition (`param3.inc`).
LWEL Maximum number of wells (`param4.inc`).
LA4 Maximum dimension for the A4 array for the reduced matrix for the direct or for the generalized conjugate-gradient equation solver, or for the entire matrix for the two-line overrelaxation equation solver (`param5.inc`).
LSDR Maximum number of search directions between restarts of the conjugate-gradient equation solver (`param5.inc`).
LRCGD1 First dimensioning parameter for the generalized conjugate-gradient solver. Set to 19 when no fill-in is allowed by ILU preconditioning (`param5.inc`).
LRCGD2 Second dimensioning parameter for the generalized conjugate-gradient solver. Set to 10 when no fill-in is allowed by ILU preconditioning (`param5.inc`).

A rough estimate of the size required for the A4 array is given by

$$LA4 = 0.5 \times NXYZ \times NY \times NZ \tag{4.2.1}$$

where

NXYZ is the number of nodes in the region;
NY is the number of nodes in the y-direction;
NZ is the number of nodes in the z-direction.

For this estimate NY and NZ are assumed to be the two smaller dimensions of the simulation region. During the storage-computation step, the actual required lengths of the arrays are calculated and printed. Execution is aborted if insufficient space has been allocated during compilation.

Identifying which dimensions are insufficient for a given simulation can be difficult, particularly when program execution is halted before critical output data have been written to files from the computer input/output buffers. Therefore, a pre-processing program, DIMEN, has been provided that reads a given data file, calculates the dimensions required to run the simulation, and checks the current dimensions of HST3D. Required array lengths are printed by the DIMEN program with warnings about those that are too small. New PARAMETER-statement files are written as necessary for replacement into the source code. The details of the DIMEN program and instructions for use are given in Appendix A.

If redimensioning is necessary, the appropriate parameters listed above must be reset in their PARAMETER statements, or the appropriate new parameter files produced by the DIMEN program must be installed into the HST3D source directory. Then the program must be recompiled and relinked. A "make" utility, when available, can be used to have the computer select just which subroutines must be recompiled after a parameter change. The release file contains a makefile for compilation purposes.

Almost all parameters and variables are passed through common blocks. The exceptions are for calls to certain utility subprograms and to the solvers. Named common blocks are used, with the name being constructed as:

C [A or S] [C or I or L or R] [Y or N] [BCGMNPSTVW] (4.2.2)

where the character meanings are as follows:

A	array
S	scalar
C	character
I	integer
L	logical
R	real
Y	written to file for a checkpoint dump
N	not written to file for a checkpoint dump
B	boundary condition
C	control parameter
G	geometry
M	matrix or vector
N	nodes
P	parameter
S	solver
T	temporary
V	dependent variable
W	well
[ ]	select one letter from the group

Thus, the common block name describes the type of variables contained in it. Many common blocks are used in HST3D because the program variables are sorted into usage groups (shown by the last letter). 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.

4.3 FILE USAGE

The heat- and solute-transport program uses several sequential-access files on disc. The Fortran unit numbers can be specified by the user in the `param0.inc` file. The following list gives the Fortran unit names, the default unit number, and a description of the contents of each file.

*	Monitor screen (standard output) and keyboard (standard input) indicated by a "*" unit number in the input and output Fortran statements
FUCLOG {28}	Calculation log of the simulation
FUINC {10}	Input file with comments
FUINS {35}	Input without comments
FURDE {11}	Echo of the input data as it is read
FULP {36}	Output of general information for the line printer or monitor screen
FUORST {8}	Output of checkpoint dump for a later restart run
FUIRST {9}	Input for a restart run from a checkpoint dump

FUPMAP {13}	Pressure, temperature, and mass fraction for plotter or monitor screen maps
FUPMAP2 {33}	Water-table elevation and density for plotter or monitor screen maps
FUVMAP {14}	Velocity components for vector plots
FUPLT {7}	Plot data for dependent variables versus time
FUP {21}	Pressure field
FUT {29}	Temperature field
FUC {22}	Concentration field
FUD {30}	Density field
FUVS {31}	Viscosity field
FUVEL {23}	Velocity field
FUWEL {24}	Well tables
FUBAL {25}	Balance tables
FUBCF {27}	Boundary-cell flow rates
FUBNFR {32}	Boundary-cell flow rates and cumulative amounts for post processing
FUKD {26}	Conductance fields

All of these files are opened from within the program. Their names are supplied or built from a suffix supplied by the user.

4.4 INITIALIZATION OF VARIABLES

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 data-input file for a given simulation must have been created previously. Under interactive execution of the program, the name of the data-input file with comments and a suffix to identify the simulation output files are requested from the user at the keyboard. On many computers, the execution can be interrupted and then continued as a background job of the operating system. However, this mode of execution requires that all output to the monitor be suspended. In order to enable non-interactive or batch execution, some reprogramming in routines READ1 and OPENF is necessary. The name and suffix could be read from a file or provided through an execution script. The output for the monitor in routines CLOSEF, ERRPRT, ITER, TIMSTP, and WRITE5 would need to be redirected to a file. Unused and scratch files are deleted at the end of the simulation by the CLOSEF subroutine.

4.6 RESTART OPTION

The program has the option for restarting from either an intermediate or the ending time of the simulation. The user may specify that restart information is to be written to the restart file every n-th time step or at a time value (record 3.8.3 of the input file). Thus, the ability to continue in a later run is created. This is useful if, for example, boundary conditions or sources need to be changed based on examination of 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 or time-value intervals. This is protection against

computer-system failure, so that long simulation runs will not have to be repeated entirely.

The information written to a file for a future restart consists of those common blocks that contain parameters and variables that are active during the simulation time-step loop and that need to be passed across time step boundaries. 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 that reads the transient information (READ3), 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 in HST3D 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. Neither can the limits on the array dimensions be changed. To do so will cause a fatal error resulting from an array 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 the program to abort.

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. The input records may be up to 130 columns wide.

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 single quotation marks (double quotes on some computers). There is a third delimiter in addition to a comma and a blank: the end-of-record 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:

$$n*d; \tag{5.1.1}$$

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 integer and 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 names are given of the variables used in the program. For definitions see section 5.2.2. 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. For variables where default values are defined, entering a zero selects the default value. The following section contains a line-by-line presentation of the data-file form with an explanation of all the variables and options.

Table 5.1. File of comment lines from the data-input form

```
C.....HST Data-Input Form
C.....Version 2.0
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 choice must be made.
C...          i.e. A or B or C
C...   (O) - Optional data with conditions for requirement
C...   P [n.n.n] - The record where parameter P is set
C.....Input by x,y,z range format is;
C.0.1.. X1,X2,Y1,Y2,Z1,Z2
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.. END OR end
C...   {nnn} - Indicates that the default number, nnn, is used if a zero
C...           is entered for that variable
C...   [T/F] - Indicates a logical variable
C...   [I] - Indicates an integer variable
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Start of the data file
C.....Specification and dimensioning data - READ1
C.1.1 .. TITLE LINE 1
C.1.2 .. TITLE LINE 2
C.1.3 .. RESTRT[T/F],TIMRST
C.....If RESTART, skip to READ3 group
C.1.4 .. HEAT[T/F],SOLUTE[T/F],EEUNIT[T/F],CYLIND[T/F],SCALMF[T/F]
C.1.5 .. TMUNIT[I]
C.1.6 .. NX,NY,NZ,NHCN
C.1.7 .. NSBC,NFBC,NLBC,NETBC,NAIFC,NHCBC,NWEL
C.1.8 .. SLMETH[I]
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Static data - READ2
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
C.2.2B.1A .. R(1),R(NR),ARGRID[T/F];(O) - CYLIND [1.4]
C.2.2B.1B .. R(I);(O) - CYLIND [1.4] and NOT ARGRID [2.2B.1A]
C.2.2B.2 .. UNIGRZ[T/F];(O) - CYLIND [1.4]
C.2.2B.3A .. Z(1),Z(NZ);(O) - UNIGRZ [2.2B.3A],CYLIND [1.4]
C.2.2B.3B .. Z(K);(O) - NOT UNIGRZ [2.2B.3A],CYLIND [1.4]
C.2.3.1 .. TILT[T/F];(O) - NOT CYLIND [1.4]
C.2.3.2 .. THETXZ,THETYZ,THETZZ;(O) - TILT [2.3.1] and NOT CYLIND [1.4]
C-------------------------------------------------------------------------------
C.....Fluid property information
C.2.4.1 .. BP
C.2.4.2 .. P0,T0,W0,DENF0
```

Table 5.1. File of comment lines from the data-input form — Continued

```
C.2.4.3 .. W1,DENF1;(O) - SOLUTE [1.4]
C.2.4.4 .. VISFAC
C-----------------------------------------------------------------------------
C.....Reference condition information
C.2.5.1 .. PAATM
C.2.5.2 .. P0H,T0H
C-----------------------------------------------------------------------------
C.....Fluid thermal property information
C.2.6 .. CPF,KTHF,BT;(O) - HEAT [1.4]
C-----------------------------------------------------------------------------
C.....Solute information
C.2.7 .. DM,DECLAM;(O) - SOLUTE [1.4]
C-----------------------------------------------------------------------------
C.....Porous media zone information
C.2.8.1 .. IPMZ,X1Z(IPMZ),X2Z(IPMZ),Y1Z(IPMZ),Y2Z(IPMZ),Z1Z(IPMZ),Z2Z(IPMZ)
C.....Use as many 2.9.1 lines as necessary
C.2.8.2 .. End with END
C-----------------------------------------------------------------------------
C.....Porous media property information
C.2.9.1 .. KXX(IPMZ),KYY(IPMZ),KZZ(IPMZ),IPMZ=1 to NPMZ [1.7]
C.2.9.2 .. POROS(IPMZ),IPMZ=1 to NPMZ [1.7]
C.2.9.3 .. ABPM(IPMZ),IPMZ=1 to NPMZ [1.7]
C-----------------------------------------------------------------------------
C.....Porous media thermal property information
C.2.10.1 .. RCPPM(IPMZ),IPMZ=1 to NPMZ [1.7];(O) - HEAT [1.4]
C.2.10.2 .. KTHXPM(IPMZ),KTHYPM(IPMZ),KTHZPM(IPMZ),IPMZ=1 to NPMZ [1.7];(O) -
C..          HEAT [1.4]
C-----------------------------------------------------------------------------
C.....Porous media solute and thermal dispersion information
C.2.11 .. ALPHL(IPMZ),ALPHT(IPMZ),IPMZ=1 to NPMZ [1.7];(O) - SOLUTE [1.4]
C..          and/or HEAT [1.4]
C-----------------------------------------------------------------------------
C.....Porous media solute property information
C.2.12 .. DBKD(IPMZ),IPMZ=1 to NPMZ [1.7];(O) - SOLUTE [1.4]
C-----------------------------------------------------------------------------
C.....Source-sink well information
C.2.13.1 .. IWEL,XW,YW,ZWB,ZWT,WBOD,WQMETH[I];(O) - NWEL [1.7] >0
C.2.13.2 .. WCF(L);L = 1 to NZ-1 (EXCLUSIVE) by ELEMENT
C.2.13.3 .. WSF(L);L = 1 to NZ-1 (EXCLUSIVE) by ELEMENT
C.2.13.4 .. WRISL,WRID,WRRUF,WRANGL;(O) - NWEL [1.7] >0 and
C..          WRCALC(WQMETH [2.13.1] >30)
C.2.13.5 .. HTCWR,DTHAWR,KTHAWR,KTHWR,TABWR,TATWR;(O) - NWEL [1.7] >0
C..          WRCALC(WQMETH [2.13.1] >30) and HEAT [1.4]
C.....Use as many sets of 2.13.1-5 lines as necessary for each well
C.2.13.6 .. End with END
C.2.13.7 .. MXITQW{10},TOLDPW{6.E-3},TOLFPW{.001},TOLQW{.001},DAMWRC{2.},
C..          DZMIN{.01},EPSWR{.001};(O) - NWEL [1.7] >0
C..          and WRCALC(WQMETH[2.13.1] >30)
C-----------------------------------------------------------------------------
C.....Boundary condition information
C-----------------------------------------------------------------------------
C.....     Specified value b.c.
C.2.14 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NSBC [1.7] > 0
C-----------------------------------------------------------------------------
```

Table 5.1. File of comment lines from the data-input form — Continued

```
C.....     Specified flux b.c.
C.2.15 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NFBC [1.7] > 0
C-----------------------------------------------------------------------------
C.....     Aquifer and river leakage b.c.
C.2.16.1 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NLBC [1.7] >0
C.2.16.2 .. KLBC,BBLBC,ZELBC by x,y,z range {0.1-0.3};(O) - NLBC [1.7] >0
C.....          River leakage b.c.
C.2.16.3 .. XR1,YR1,XR2,YR2,KRBC,BBRBC,ZERBC;(O) - NLBC [1.7] > 0
C.2.16.4 .. End with END
C-----------------------------------------------------------------------------
C.....     Evapotranspiration b.c.
C.2.17.1 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NETBC [1.7] >0
C.2.17.2 .. ZLSETB,BETBC by x,y,z range {0.1-0.3};(O) - NETBC [1.7] >0
C-----------------------------------------------------------------------------
C.....     Aquifer influence functions
C.2.18.1 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NAIFC [1.7] > 0
C.2.18.2 .. UVAIFC by x,y,z range {0.1-0.3};(O) - NAIFC [1.7] > 0
C.2.18.3 .. IAIF;(O) - NAIFC [1.7] > 0
C.....          Pot  a.i.f.
C.2.18.4A .. ABOAR,POROAR,VOAR;(O) - IAIF [2.18.3] = 1
C.....          Transient, Carter-Tracy a.i.f.
C.2.18.4B .. KOAR,ABOAR,VISOAR,POROAR,BOAR,RIOAR,ANGOAR;(O) -
C..          IAIF [2.18.3] = 2
C-----------------------------------------------------------------------------
C.....     Heat conduction b.c.
C.2.19.1 .. ZHCBC(K);(O) - HEAT [1.4] and NHCBC [1.7] > 0
C.2.19.2 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
c..          HEAT [1.4] and NHCBC [1.7] > 0
C.2.19.3 .. DTHHC by x,y,z range {0.1-0.3} for HCBC nodes;(O) -
C..          HEAT [1.4] and NHCBC [1.7] > 0
C.2.19.4 .. KHCBC by x,y,z range {0.1-0.3} for HCBC nodes;(O) -
C..          HEAT [1.4] and NHCBC [1.7] > 0
C-----------------------------------------------------------------------------
C.....Free surface b.c.
C.2.20 .. FRESUR[T/F]
C-----------------------------------------------------------------------------
C.....Initial condition information
C.2.21.1 .. ICHYDP[T/F]
C.2.21.2 .. ICHWT[T/F];(O) - FRESUR [2.20]
C.2.21.3A .. ZPINIT,PINIT;(O) - ICHYDP [2.21.1] and NOT ICHWT [2.21.2]
C.2.21.3B .. P by x,y,z range {0.1-0.3};(O) - NOT ICHYDP [2.21.1] and
C..          NOT ICHWT [2.21.2]
C.2.21.3C .. HWT by x,y,z range {0.1-0.3};(O) - FRESUR [2.20] and
C..          ICHWT [2.21.2]
C.2.21.4 .. T by x,y,z range {0.1-0.3};(O) - HEAT [1.4]
C.2.21.5 .. NZTPHC, ZTHC(I),TVZHC(I);(O) - HEAT [1.4] and NHCBC [1.7] >0,
C..          limit of 10
C.2.21.6 .. C by x,y,z range {0.1-0.3};(O) - SOLUTE [1.4]
C-----------------------------------------------------------------------------
C.....Calculation information
C.2.22.1 .. FDSMTH,FDTMTH
C.2.22.2 .. TOLDEN{.001},MAXITN{5}
```

Table 5.1. File of comment lines from the data-input form — Continued

```
C.....     Two-line s.o.r. solver
C.2.22.3 .. NTSOPT{5},EPSSLV{1.e-7},EPSOMG{.2},MAXIT1{50},MAXIT2{100};
C..          (O) - SLMETH [1.8] = 2
C.....     Red-black restarted conjugate gradient solver
C.2.22.4 .. IDIR,IORDER,NSDR,EPSSLV{1.e-7},MAXIT2{100};
C..          (O) - SLMETH [1.8] = 3 or 5
C-------------------------------------------------------------------------------
C.....Output information
C.2.23.1 .. PRTPMP,PRTFP,PRTIC,PRTBC,PRTSLM,PRTWEL; all [T/F]
C.2.23.2 .. IPRPTC,PRTDV[T/F];(O) - PRTIC [2.23.1]
C.2.23.3 .. ORENPR[I];(O) - NOT CYLIND [1.4]
C.2.23.4 .. PLTZON[T/F];(O) - PRTPMP [2.23.1]
C.2.23.5 .. PLTTEM[T/F]
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C..... TRANSIENT DATA - READ3
C.3.1 .. THRU[T/F]
C.....If THRU is true, proceed to record 3.99
C-------------------------------------------------------------------------------
C.....The following is for NOT THRU
C-------------------------------------------------------------------------------
C.....Source-sink well information
C.3.2.1 .. RDWTD[T/F];(O) - NWEL [1.7] > 0
C.3.2.2 .. IWEL,QWV,PWSUR,PWKT,TWSRKT,CWKT;(O) - RDWTD [3.2.1]
C.....Use as many 3.2.2 lines as necessary
C.3.2.3 .. End with END
C-------------------------------------------------------------------------------
C.....Boundary condition information
C-------------------------------------------------------------------------------
C.....     Specified value b.c.
C.3.3.1 .. RDSPBC,RDSTBC,RDSCBC; all [T/F];(O) - NSBC [1.7] > 0
C.3.3.2 .. PNP B.C. by x,y,z range {0.1-0.3};(O) - RDSPBC [3.3.1]
C.3.3.3 .. TSBC by x,y,z range {0.1-0.3};(O) - RDSPBC [3.3.1] and
C..          HEAT [1.4]
C.3.3.4 .. CSBC by x,y,z range {0.1-0.3}; (O) - RDSPBC [3.3.1] and
C..          SOLUTE [1.4]
C.3.3.5 .. TNP B.C. by x,y,z range {0.1-0.3};(O) - RDSTBC [3.3.1] and
C..          HEAT [1.4]
C.3.3.6 .. CNP B.C. by x,y,z range {0.1-0.3};(O) - RDSCBC [3.3.1] and
C..          SOLUTE [1.4]
C-------------------------------------------------------------------------------
C.....     Specified flux b.c.
C.3.4.1 .. RDFLXQ,RDFLXH,RDFLXS; all [T/F];(O) - NFBC [1.7] > 0
C.3.4.2 .. QFFX,QFFY,QFFZ B.C. by x,y,z range {0.1-0.3};(O) -
C..          RDFLXQ [3.4.1]
C.3.4.3 .. UDENBC  by x,y,z range {0.1-0.3};(O) - RDFLXQ [3.4.1]
C.3.4.4 .. TFLX B.C. by x,y,z range {0.1-0.3};(O) - RDFLXQ [3.4.1] and
C..          HEAT [1.4]
C.3.4.5 .. CFLX B.C. by x,y,z range {0.1-0.3};(O) - RDFLXQ [3.4.1] and
C..          SOLUTE [1.4]
C.3.4.6 .. QHFX,QHFY,QHFZ B.C. by x,y,z range {0.1-0.3};(O) - RDFLXH [3.4.5]
C.3.4.7 .. QSFX,QSFY,QSFZ B.C. by x,y,z range {0.1-0.3};(O) - RDFLXS [3.4.1]
C-------------------------------------------------------------------------------
```

Table 5.1. File of comment lines from the data-input form — Continued

```
C.....     Leakage b.c.
C.3.5.1 .. RDLBC[T/F];(O) - NLBC [1.7] > 0
C.3.5.2 .. PHILBC,DENLBC,VISLBC by x,y,z range {0.1-0.3};(O) - RDLBC [3.5.1]
C.3.5.3 .. TLBC by x,y,z range {0.1-0.3};(O) - RDLBC [3.5.1] and HEAT [1.4]
C.3.5.4 .. CLBC by x,y,z range {0.1-0.3};(O) - RDLBC [3.5.1] and SOLUTE [1.4]
C.....          River leakage b.c.
C.3.5.5 .. XR1,YR1,XR2,YR2,HRBC,DENRBC,VISRBC,TRBC,CRBC;(O) - RDLBC [3.5.1]
C.....Use as many 3.5.5 lines as necessary
C.3.5.6 .. End with END
C-----------------------------------------------------------------------------
C.....     Evapotranspiration b.c.
C.3.6.1 .. RDETBC[T/F];(O) - NETBC [1.7] > 0
C.3.6.2 .. QETBC by x,y,z range {0.1-0.3};(O) - RDETBC [3.5.1]
C-----------------------------------------------------------------------------
C.....     Aquifer influence function b.c.
C.3.7.1 .. RDAIF[T/F];(O) - NAIFC [1.7] > 0
C.3.7.2 .. DENOAR by x,y,z range {0.1-0.3};(O) - RDAIF [3.7.1]
C.3.7.3 .. TAIF by x,y,z range {0.1-0.3};(O) - RDAIF [3.7.1] and HEAT [1.4]
C.3.7.4 .. CAIF by x,y,z range {0.1-0.3};(O) - RDAIF [3.7.1] and SOLUTE [1.4]
C-----------------------------------------------------------------------------
C.....Calculation information
C.3.8.1 .. RDCALC[T/F]
C.3.8.2 .. AUTOTS[T/F];(O) - RDCALC [3.8.1]
C.3.8.3.A .. DELTIM;(O) - RDCALC [3.8.1] and NOT AUTOTS [3.8.2]
C.3.8.3.B .. DPTAS{5E4},DTTAS{5.},DCTAS{.25},DTIMMN{1.E4},DTIMMX{1.E7};
C..           (O) - RDCALC [3.8.1] and AUTOTS [3.8.2]
C.3.8.4 .. TIMCHG
C-----------------------------------------------------------------------------
C.....Output information
C.3.9.1 .. PRISLM,PRIKD,PRIPTC,PRIDV,PRIVEL,PRIGFB,PRIBCF,PRIWEL; all [I]
C.3.9.2 .. IPRPTC;(O) - IF PRIPTC [3.9.1] NOT = 0
C.3.9.3 .. CHKPTD[T/F],PRICPD,SAVLDO[T/F]
C-----------------------------------------------------------------------------
C.....Contour and vector map information
C.3.10.1 .. CNTMAP[T/F],VECMAP[T/F],PRIMAP[I]
C-----------------------------------------------------------------------------
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-----------------------------------------------------------------------------
C.....End of simulation line follows, THRU=.TRUE.
C.3.99 .. THRU
T
C.....End of the data file
C-----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
```

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, or READ3, reads that record for the values 1 through 3 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 mandatory that it be omitted. A record number in square brackets following a variable indicates the record where that variable is first set.

Input by x,y,z 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 x,y and z coordinates. One may specify a plane, or a line, or a point instead of a prism by making the appropriate coordinates equal. The value to be entered into the array locations pertaining to the nodes contained within 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. A fourth option for data input 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 x, then increasing y, and then increasing z. A fifth option for data input performs linear interpolation along one of the coordinate directions based on values given at two locations.

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. Either node-by-node variation or linear variation along a selected coordinated direction can be specified. It is permissible to use rectangular prisms that include cells that are outside the simulation region, or to include cells that are not of the boundary condition type for which data are being supplied.

A set of record pairs is needed where the first is:

I.1 X1, X2, Y1, Y2, Z1, Z2;

where

Xi, Yj, Zk are inclusive coordinate ranges in the x,y, and z directions,

and the second is:

I.2 VAR1, IMOD1, [VAR2, IMOD2, VAR3, IMOD3];

where

VARi is the value of the i-th variable;

IMODi is the modification code for the i-th 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 *in the order of increasing x, then y, then z;* In this case, the value of VARi in record I.2 is not used and may be given a value of zero. In the case of three data variables being read in, each variable is read in for the entire sub-region in turn.

5 means linearly interpolate the data to each node from the two end-point values. VARi in record I.2 is set to 1., 2., or 3. to select the linear interpolation along the x, y, or z direction respectively. In this case, the following record is needed:

I.3 XS(1), FS(1), XS(2), FS(2);

where

XS(1) and XS(2) specify two points along the coordinate direction for interpolation; and

FS(1) and FS(2) are the values of the variable at the two points.

Under node-by-node input (modification factor of 4), the data must be one element per record or the last record must end with the last element to be read. Any data elements in the last record beyond the last element to be read will be lost. Node-by-node input is useful for data distributions over the region obtained from geographical information system data preparation tools. It can be use only to install data, not modify previously read data. Under modification factor 5, both linear interpolation and extrapolation are used as necessary to fill the defined zone with data. The two points defining the interpolation must be in algebraic ascending order.

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 and I.3 are used as necessary, then the following line is entered to terminate the input:

I.4 `END` or `end`.

The generic dimensions for the various parameters are indicated, and the units must be one of the two sets given in Table 5.2. unless the conversion factors in the HST3D program are modified. User specification of a unique unit for each type of data input is not available. The time unit is selected by the user. All other units involving time are then formed from this selection. 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 below. The conversion factors from U.S. customary units to metric units and vice versa are contained in the common block /CSRYP/ and defined in the subprogram, INIT1. The variable names are in the form, CNV*xxx*. The variable names for the conversion factors from metric units to U.S. customary units are in the form CNV*xx*I. The unit labels are contained in variables named UNIT*xx* 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 and [I] indicates an integer variable. All index ranges are inclusive and all pressure values are relative to atmospheric unless stated otherwise.

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; and (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 data-input form listed in Table 5.1. All equation number references in this section are to Kipp (1987).

Table 5.2. Metric and U.S. customary system units used in the heat- and solute-transport simulator

Quantity (generic units)	Metric (S.I.) units	U.S. customary units
Mass (M)	kilogram (kg)	pound avoirdupois (lb_{avdp})†
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)	cubic meter per second (m^3/s)	cubic foot per day (ft^3/d)
Density (M/L^3)	kilogram per cubic meter (kg/m^3)	pound avoirdupois per cubic foot (lb_{avdp}/ft^3)‡
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)	Pascal-seconds (Pa-s)	centipoise (cP)††
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 (E/M-T or FL/M-T)	Joule per kilogram-degree Celsius (J/kg–°C)	British Thermal Unit per pound avoirdupois-degree Fahrenheit (BTU/lb_{avdp}–°F)
Heat-Transfer Coefficient (E/t–L^2–T)	Watt per square meter-degree Celsius (W/m^2–°C)	British Thermal Unit per hour-square foot-degree Fahrenheit (BTU/h–ft^2–°F)
Volumetric flux (L^3/L^2–t)	cubic meter per square meter-second (m^3/m^2–s)	cubic foot per square foot-day (ft^3/ft^2–d)
Heat flux (E/L^2–t)	Watt per square meter (W/m^2)	British Thermal Unit per square foot-hour (BTU/ft^2–h)
Mass flux (M/L^2–t)	kilogram per square meter-second (kg/m^2–s)	pound avoirdupois per square foot-day (lb/ft^2–d)

† A weight rather than a mass
‡ A weight density rather than a mass density
†† Not U.S. customary unit but common usage

I. Problem specification and dimensioning data - 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 second character string of up to 80 characters.

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 a record terminator, "/", or 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 U.S. customary units; otherwise, metric units are assumed. The conversion factors are set for the U.S. customary and metric units are given in Table 5.2 and appear after the table of contents of this report 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 TMUNIT;[I].

TMUNIT Integer code for selection of time units for simulation: 1 - seconds, 2 - minutes, 3 - hours, 4 - days, 6 - years. All temporal data must be entered in the time unit selected. Conversion is possible as a step in the input by x,y,z range procedure or by redefining conversion factors.

1.6 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 value is used only for a heat-transport simulation if needed for the finite-difference solution of equations 2.5.5.3a-d and 2.5.5.4a-d.

1.7 NSBC, NFBC, NLBC, NETBC, NAIFC, NHCBC, NWEL.

NSBC Number of cells (nodes) with a specified-pressure, specified-temperature, and(or) specified-mass-fraction boundary condition.

NFBC Number of cells with a specified-flux boundary condition; flow, heat, and(or) solute.

NLBC Number of cell faces with a leakage-boundary condition.

NETBC Number of cell faces with an evapotranspiration-boundary condition.

NAIFC Number of cell faces with an aquifer-influence-boundary condition.

NHCBC Number of cell faces with a heat-conduction boundary condition.

NWEL Number of wells in the simulation region.

These values only need to be set to one for each boundary-condition type to be included in the simulation.

1.8 SLMETH;[I].

SLMETH Solution method selection for the matrix equations (integer):

1 selects the triangular-factorization direct solver for the reduced matrix based on the alternating-diagonal (D4) renumbering,

2 selects the two-line, successive-overrelaxation iterative solver,

3 selects the generalized conjugate-gradient iterative solver for the reduced matrix based on red-black (RB) renumbering,

5 selects the generalized conjugate-gradient iterative solver for the reduced matrix based on alternating-diagonal zig-zag (D4Z) renumbering,

II. Static data - READ2.

The data in group two are invariant throughout the simulation.

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.

UNIGR*i* True if the grid in the *i*th 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.

Z(K) Array of node locations along the z-axis (L).

2.2B.1A R(1), R(NX), ARGRID [T/F]; Optional, used for cylindrical-coordinate systems.

R(1) Interior radius of cylindrical-coordinate system (L). Must be greater than zero. Required if a cylindrical-coordinate system.

R(NX) 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].

UNIGRZ True if the grid in the z-direction is uniform.

2.2B.3A Z(1), Z(NZ); Optional, used if uniform grid in the z-direction for cylindrical coordinates.

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).

2.2B.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 (°).

THETYZ Angle that the y-axis makes with the vertical upward direction (°).

THETZZ Angle that the z-axis makes with the vertical upward direction (°).

A. Fluid properties

2.4.1 BP.

BP Compressibility of the fluid $(F/L^2)^{-1}$.

a. Fluid density data

2.4.2 P0, T0, W0, DENF0.

P0 Reference pressure for density (relative to atmospheric) (eq. 2.2.1.1b) (F/L^2).

T0 Reference temperature for density (eq. 2.2.1.1b) (T).

W0 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.

DENF0 Fluid density at P0, T0, and W0 (eq. 2.2.1.1c or eq. 2.2.1.3b) (M/L^3),[(kg/m^3) or (lb_{avdp}/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. This must be in absolute mass-fraction units (not scaled).

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_{avdp}/ft^3)].

If solute transport is being simulated, W0 and W1 are used to calculate the coefficient of density dependence on solute mass fraction by equation 2.2.1.3b. If, in addition, 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.

b. Fluid-viscosity data

2.4.4 VISFAC.

VISFAC Viscosity multiplication factor (-) or, if negative, the constant viscosity value (M/L-t). [(Pa-s) or (cP)].

B. Reference condition information

2.5.1 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.

2.5.2 P0H, T0H.

P0H 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.

T0H 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.

C. Fluid thermal properties

2.6 CPF, KTHF, BT; Optional, required only if heat transport is being simulated.

CPF Fluid heat capacity at constant pressure (E/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 (E/L-t-T), [(W/m–°C) or (BTU/ft–h–°F)]. 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.

D. Solute information

2.7 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 (= ln 2/$T_{½}$) where $T_{½}$ is the half-life of the solute species (t^{-1}).

The following two records describe the zonation of the simulation region.

2.8.1 IPMZ, X1Z(IPMZ), X2Z(IPMZ), Y1Z(IPMZ), Y2Z(IPMZ), Z1Z(IPMZ), Z2Z(IPMZ).

IMPZ Porous-medium zone number.

XiZ, YjZ, ZkZ inclusive coordinate ranges in the x,y, and z directions, respectively.

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 coordinates in the x, y, and z directions define the rectangular prism for a given zone. The zones must be convex and non-overlapping as explained in section 3.1 of Kipp (1987) 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, Y1Z and Y2Z are not used and may be given a zero or blank space. The subscript IPMZ identifies data that are input by zone number in subsequent records. NPMZ will be the total number of zones defined.

2.8.2 End the input with `END` or `end`.

E. Porous-media properties

2.9.1 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 included in the input record.

KZZ Permeability in the z-direction for zone IPMZ (L^2).

2.9.2 POROS(IPMZ), IPMZ = 1 to NPMZ.

POROS Porosity of the medium in zone IPMZ (-).

The effective porosity for unconfined flow systems is often taken to be the specific yield.

2.9.3 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.

a. Porous-media thermal properties

2.10.1 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 (E/L^3–T).

2.10.2 KTHXPM(IPMZ), KTHYPM(IPMZ), KTHZPM(IPMZ), IPMZ = 1 to NPMZ; Optional, required only if heat transport is being simulated.

KTHXPM Thermal conductivity of the porous medium in the x-direction for zone IPMZ (E/L-t-T),[(W/m–°C) or (BTU/ft–h–°F)].

KTHYPM Thermal conductivity of the porous medium in the y-direction for zone IPMZ (E/L-t-T), [(W/m–°C) or (BTU/ft–h–°F)]. Not used if cylindrical coordinates, but a zero or a blank space must be included in the input record.

KTHZPM Thermal conductivity of the porous medium in the z-direction for zone IPMZ (E/L-t-T), [(W/m–°C) or (BTU/ft–h–°F)].

b. Solute and thermal dispersion information

2.11 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).

c. Solute sorption information

2.12 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 (-).

F. Well-model information

Record lines 2.13.1-5 are repeated as a group 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.

2.13.1 IWEL, XW, YW, ZWB, ZWT, WBOD, WQMETH; Optional, required only if there are wells in the simulation region and well-definition data are to be read.

- IWEL Well number.
- XW X-coordinate of well location (L). In cylindrical coordinates, this would be zero for the single well at the center of the coordinate system.
- YW Y-coordinate of well location (L). Not used in cylindrical coordinates, but a zero or a blank space must be included in the input record.
- ZWB Z-coordinate of lowermost completion layer for the well (L).
- ZWT Z-coordinate of uppermost completion layer for the well (L). This is the location at which the dependent-variable data are taken if this is an observation well.
- WBOD Well bore outside diameter (L). This is the drilled diameter or diameter of the screen or perforated casing.
- 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.
  - 12 Specified well-flow rate with allocation by mobility and pressure difference and limiting mass fraction for production.
  - 13 Specified well-flow rate with allocation by mobility and limiting mass-fraction for production.
  - 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.

2.13.2 WCF(L), L = 1 to NZ-1; Optional, required only if there are wells in the simulation region.

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. If WCF is zero for the element between ZWB and ZWB+1 or for the element between ZWT-1 and ZWT, then ZWB or ZWT should be adjusted as necessary to range from the bottom to the top completion layers 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.

2.13.3 WSF(L), L = 1 to NZ-1; Optional, required only if there are wells in the simulation region.

WSF Well-skin-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. A positive skin factor corresponds to a reduced permeability around the well bore. An enhanced permeability around the well bore can be approximately represented by specifying a skin factor less than zero. Note that this is not the same as a well-completion factor.

2.13.4 WRISL, WRID, WRRUF, WRANGL; Optional, required only if 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 in Kipp, 1987).

WRANGL Well-riser pipe angle with vertical direction (°).

2.13.5 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 ($E/t–L^2–T$), [($W/m^2–°C$) or ($BTU/ft^2–h–°F$)]

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 (E/L-t-T),[(W/m–°C) or (BTU/ft–h–°F)].

KTHWR Thermal conductivity of the well-riser pipe (E/L-t-T), [(W/m–°C) or (BTU/ft–h–°F)].

TABWR Ambient temperature at the bottom of the well-riser pipe (T).

TATWR Ambient temperature at the top of the well-riser pipe (T).

2.13.6 End this data set with `END` or `end`.

2.13.7 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 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.

G. Boundary-condition information

a. Specified value b.c.

2.14 IBC by x,y,z range; Optional, required only if specified value boundary-condition cells are present.

IBC Index of boundary-condition type. This is a three-digit number 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. No IMOD parameter is used here.

A specified value for a cell removes that cell from the calculation. Therefore, a specified-value boundary-condition cell may *not* have any other type of boundary condition.

b. Specified flux b.c.

2.15 IBC by x,y,z range; Optional, required only if specified flux boundary- condition cells are present.

IBC Index of boundary-condition type. This is a six-digit number 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. No IMOD parameter is used here.

c. Aquifer leakage b.c.

2.16.1 IBC by x,y,z range for aquifer-leakage boundary cells; Optional, required only if leakage boundary conditions are being used.

IBC Index of boundary-condition type. It is a six-digit number in the form n_100300 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 n_4 place denotes an aquifer-leakage

boundary condition. No IMOD parameter is used here.

2.16.2 KLBC, BBLBC, ZELBC by x,y,z 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. However some null data must be entered for ZELBC or the record terminator," /", used.

d. River leakage b.c.

2.16.3 XR1, YR1, XR2, YR2, KRBC, BBRBC, ZERBC; Optional, required only if the river-leakage boundary condition is being used.

XR1, YR1, XR2, YR2 Coordinate ranges for a river-leakage boundary-condition segment. River segments are lines in the x-, y-, or diagonal coordinate directions from point 1 to point 2. The diagonal directions must be defined by equal numbers of nodal increments in the x- and y-directions.

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 Fig. 2.3 of Kipp (1987)) (L).

Use as many 2.16.3 records as necessary to describe the river.

2.16.4 End this data set (records 2.16.3) with `END` or `end`. This is required if any leakage boundary conditions are being used.

A river leakage boundary condition is not realistic in cylindrical coordinates.

e. Evapotranspiration b.c.

2.17.1 IBC by x,y,z range for evapotranspiration boundary cells; Optional, required only if evapotranspiration boundary conditions are being used.

IBC Index of boundary-condition type. It is a six-digit number in the form $n_1$00700 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 7 in the n_4 place denotes an evapotranspiration boundary condition. No IMOD parameter is used here.

2.17.2 ZLSETB, BETBC by x,y,z range; Optional, required only if evapotranspiration boundary conditions are being used.

ZLSETB Elevation of the land surface for evapotranspiration boundary condition (L),

BETBC Depth below land surface of evapotranspiration extinction (L).

Input for this record is expected as three variables (record I.2), so zeros or an end-of-record terminator are required for the third set of data.

f. Aquifer-influence function b.c.

2.18.1 IBC by x,y,z 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 a six-digit number in the form $n_1$00400 for aquifer-influence functions, where n 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 n_4 place denotes an aquifer-influence-function boundary condition. No IMOD parameter is used here.

2.18.2 UVAIFC by x,y,z range; Optional, required only if aquifer-influence functions are used.

UVAIFC Temporary storage for input of user-specified weighting factors for aquifer-influence-function spatial allocation. These factors multiply the product of boundary-cell facial area and permeability. These factors are defined on a zonal basis, so the factor for all boundary zones that include a common cell for which an aquifer-influence-function boundary condition applies must be the same. If the default weighting of one is desired, no values for UVAIFC need to be entered (except the closing `END` or `end`).

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.

2.18.4A ABOAR, POROAR, VOAR; Optional, required only if pot-aquifer-influence functions (see Kipp, l987) are being used.

ABOAR Porous-medium bulk vertical compressibility for the outer-aquifer region $(F/L^2)^{-1}$.

POROAR Porosity for the outer-aquifer region (-).

VOAR Volume of the outer-aquifer region (L^3).

2.18.4B 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).

VISOAR Viscosity of the fluid in the outer-aquifer region (M/L-t), [(Pa-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 (°). This is the angle subtended by the part of the equivalent cylindrical boundary that is subject to flux determined by the aquifer-influence function.

g. Heat conduction b.c.

2.19.1 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.

2.19.2 IBC by x,y,z 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. It is a six-digit 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 sign of the normal is disregarded. The number 4 in the n_4 place 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. No IMOD parameter is used here.

2.19.3 DTHHC by x,y,z range for heat-conduction boundary-condition cells; Optional, required only if a heat simulation, and if there are heat-conduction boundary-condition cells.

DTHHC Thermal diffusivity of the heat-conducting medium outside the simulation region as a function of boundary-cell location (L^2/t).

2.19.4 KHCBC by x,y,z range for heat-conduction boundary-condition cells; Optional, required only if a heat simulation, and if there are heat-conduction boundary-condition cells.

KHCBC Thermal conductivity of the heat-conducting medium outside the simulation region as a function of boundary-cell location (E/L-t-T), [(W/m–°C) or (BTU/ft–h–°F)].

For the IBC array input, there is a hierarchy for the various types of boundary conditions for the ground-water flow equation. The specified-value boundary definition data can be read in with overlapping zones and each cell will have the most recent definition read. Specified-flux boundary definition data will replace other types of boundary condition definitions unless the previous definition was a leakage boundary condition. In that case it is assumed that a river leakage zone has been overlain by an specified flux zone. Leakage boundary definitions also replace all other definitions except when the previous definition was a specified flux. Aquifer influence function definitions replace any previous boundary type definitions.

h. Free-surface boundary condition

2.20 FRESUR [T/F]

FRESUR True if the region is unconfined, so that a free-surface boundary exists.

H. Initial conditions

2.21.1 ICHYDP [T/F]

ICHYDP True if initial condition of hydrostatic pressure 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 x,y,z 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 x,y,z range; Optional, required only if desired in conjunction with a free-surface boundary condition.

HWT Water-table-elevation distribution for the initial condition (L). Data are to be input for the upper layer of cells only.

A hydrostatic pressure or a specified pressure distribution can be used as the initial condition for confined or unconfined conditions. A water-table-elevation distribution can be used only with unconfined (free-surface) conditions. The pressure distribution resulting from a water-table-elevation distribution is hydrostatic in the vertical direction but not necessarily in the horizontal direction.

2.21.4 T by x,y,z range; Optional, required only if a heat simulation is being done.

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 10.

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 2.19.1, and must be in algebraic ascending order.

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 x,y,z range; Optional, required only if a solute simulation is being done.

C Mass-fraction (or scaled mass fraction) distribution for the initial condition (-).

I. 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-in-space differencing used for advective terms.

FDTMTH Factor for temporal-discretization method.

0.5 centered-in-time or Crank-Nicolson differencing used.

1. backward-in-time or fully-implicit differencing used.

These weighting factors can be set between the limits stated above.

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, EPSSLV, EPSOMG, MAXIT1, MAXIT2; Optional, required only if the two-line, successive-overrelaxation method for solving the matrix equations is selected (SLMETH=2) (see Kipp, 1987).

NTSOPT Number of time steps between recalculations of the optimum-overrelaxation parameter. Default set at 5.

EPSSLV Tolerance for the two-line, successive-overrelaxation iterative solution of the matrix equations at each time plane. Default set at 1×10^{-7}. 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.

2.22.4 IDIR, IORDER, NSDR, EPSSLV, MAXIT2; Optional, required only if the generalized conjugate-gradient method for solving the matrix equations is selected, with either renumbering scheme (SLMETH=3 or 5).

IDIR Direction index for the red-black or alternate-diagonal zig-zag renumbering of the nodes for the flow and transport equations. The order is as follows: 1 - xyz, 2 - xzy, 3 - yxz, 4 - yzx, 5 - zxy, 6 - zyx.

IORDER Set to 1 if standard incomplete LU factorization is to be used to precondition the matrix. Set to 2 if modified incomplete LU factorization is to be used.

NSDR Number of steps between restarts of the generalized conjugate-gradient solver

EPSSLV Tolerance on the relative residual for the restarted conjugate-gradient iterative solution of the matrix equations at each time plane. Default set at 1×10^{-7}. The Euclidean norm of the changes in the vector of dependent-variable values must be less than or equal to this tolerance.

MAXIT2 Maximum number of iterations allowed for the solution of the matrix equations. Default set at 100.

There can be a significant variation in the number of iterations needed to achieve convergence depending on the order of the directions for the red-black renumbering (SLMETH=3). The user should experiment with various choices to determine the one requiring the fewest iterations. The convergence rate with the alternating-diagonal-zig-zag renumbering (SLMETH=5) is much less sensitive to the order of directions.

J. 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 $n_1 n_2 n_3$, 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.

2.23.4 PLTZON [T/F]; Optional, required only if printout of porous-media properties has been requested.

PLTZON True if a plot of the porous-media property zones is to be made.

2.23.5 PLTTEM [T/F].

PLTTEM True if data are to be written for post-processing plots of calculated values of the dependent variables versus time.

III. 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.

A. Well information

3.2.1 RDWTD [T/F]; Optional, required only if there are wells associated with this simulation region.

RDWTD True if transient well data are 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). Positive is injection, negative is production.

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. Alternatively, it is used for specification of a limit on the production concentration (mass-fraction).

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. Fluid temperature and mass fraction data are required for each injection well if heat transport simulation and solute transport simulation are being done.

3.2.3 End this data set with `END` or `end`.

B. Boundary-condition information

For ease of data input, all transient boundary-condition information can be specified over a larger range of nodes than those having that particular boundary-condition type. The M*x*BC arrays determine which nodes receive the boundary-condition data. One example would be the use of a rectangular plane to cover a diagonal staircase boundary segment.

If any transient boundary-condition data of a given type are being changed during the simulation, *all* boundary-condition data for that type must be input, not just those values that are being changed. This is to accommodate conversion of units, if necessary.

a. Specified value b.c.

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 x,y,z 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 x,y,z 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 x,y,z 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 x,y,z 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 x,y,z 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 (-).

b. Specified flux b.c.

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 x,y,z 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 in the x,y, and z-coordinate directions, respectively for a flux boundary (L^3/L^2–t).

3.4.3 UDENBC by x,y,z range; Optional, required only if specified fluid-flux 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 x,y,z range; Optional, required only if specified fluid-flux values 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 x,y,z range; Optional, required only if specified fluid-flux values 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 x,y,z range; Optional, required only if specified heat-flux values are to be read at this time.

QHFX, QHFY, QHFZ Components of the heat-flux vector in the x,y, and z-coordinate directions, respectively for a flux boundary. Heat flux should be specified only through faces where there is no fluid flux (E/L^2–t), [(W/m^2) or ($BTU/h–ft^2$)].

3.4.7 QSFX, QSFY, QSFZ by x,y,z range; Optional, required only if specified solute-flux values are to be read at this time.

QSFX, QSFY, QSFZ Components of the solute-flux vector in the x,y, and z-coordinate directions, respectively for a flux boundary. Solute flux should be specified only through faces where there is no fluid flux (M/L^2–t).

c. Leakage b.c.

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 at this time.

3.5.2 PHILBC, DENLBC, VISLBC by x,y,z 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). The units of PHILBC are equivalent to L^2/t^2.

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), [(Pa–s) or (cP)].

3.5.3 TLBC by x,y,z 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 x,y,z 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 (-).

d. River leakage b.c.

3.5.5 XR1, YR1, XR2, YR2, HRBC, DENRBC, VISRBC, TRBC, CRBC; Optional, required only if river-leakage boundary-condition data are to be read at this time.

XR1, YR1, XR2, YR2 Coordinate 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 2.16.3.

HRBC Potentiometric head in the river (L).

DENRBC Density of the river fluid (M/L^3), [(kg/m^3) or (lb_{avdp}/ft^3)].

VISRBC Viscosity of the river fluid (M/L–t), [(Pa–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 `END` or `end`.

e. Evapotranspiration b.c.

3.6.1 RDETBC[T/F]; Optional, required only if evapotranspiration boundary-condition cells are employed.

RDETBC True if evapotranspiration boundary-condition data are to be read at this time.

3.6.2 QETBC by x,y,z range; Optional, required only if evapotranspiration boundary-condition data are to be read at this time.

QETBC Maximum evapotranspiration flux when water table is at land surface (L^3/L^2–t).

f. Aquifer-influence function b.c.

3.7.1 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.

3.7.2 DENOAR by x,y,z 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_{avdp}/ft^3)].

3.7.3 TAIF by x,y,z range; Optional, required only if aquifer-influence-function 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).

3.7.4 CAIF by x,y,z range; Optional, required only if aquifer-influence-function cells are employed, and if solute transport is being simulated.

CAIF Mass fraction (or scaled mass fraction) of solute in the outer-aquifer region associated with a given aquifer-influence-function cell (-).

In applying the associated temperatures and mass fractions associated with a density at a given boundary condition, care must be taken to maintain a consistency with the density equation relating density to pressure, temperature and mass fraction.

C. Calculation information

The following data pertain to time-step control and the time when new transient data will be read.

3.8.1 RDCALC [T/F].

RDCALC True if calculation information is to be read at this time.

3.8.2 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.

3.8.3A 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).

3.8.3B 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^oC.

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 at the start of the simulation and after a change in boundary conditions, that is, at TIMCHG. Default set at 10^4s.

DTIMMX Maximum time step allowed (t). Default set at 10^7s.

3.8.4 TIMCHG.

TIMCHG Time at which new transient data will be read or at which the simulation will be terminated (t).

D. Output information

3.9.1 PRISLM, PRIKD, PRIPTC, PRIDV, PRIVEL, PRIGFB, PRIBCF, PRIWEL.

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.

PRIDV Printout interval (integer) for fluid-density and fluid-viscosity arrays.

PRIVEL Printout interval (integer) for velocity arrays. These are interstitial velocities at the cell boundaries.

PRIGFB Printout interval (integer) for flow-balance information for the region.

PRIBCF Printout interval (integer) for specified-value boundary-condition flow rates.

PRIWEL Printout interval (integer) for well information.

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.

-n Means that printout will occur at every n simulation time units.

-TIMCHG Means that printout will occur only at the time at which new transient data will be read and at the end of the simulation. TIMCHG must be an integer.

To obtain a printout only at the end of the simulation, set PRI*xxx* to a value greater than the expected maximum number of time steps or to -TIMCHG of the ending time of the simulation.

3.9.2 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 $n_1 n_2 n_3$ 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.

3.9.3 CHKPTD [T/F], PRICPD(I), SAVLDO [T/F].

CHKPTD True if checkpoint dumps are to be made for possible restarts of the simulation.

PRICPD Print control (integer) for frequency of checkpoint dumps. See previous information on printout intervals for possible values.

SAVLDO True if only the last checkpoint dump is to be saved.

The following record controls data output for the generation of contour maps by a post-processing program.

3.10.1 CNTMAP[T/F], VECMAP[T/F], PRIMAP(I).

CNTMAP True if data are to be written for producing a contour map.

VECMAP True if data are to be written for producing a velocity vector map.

PRIMAP Printout interval (integer) for contour and/or vector maps. See previous information on printout intervals for possible values.

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 THRU [T/F].

THRU Set to true at this point to signify the end of the simulation.

This ends the data-input file description. For quick reference, a list of the definitions for the numerical program-control options is provided in Table 5.3.

Table 5.3. List of options for numerical program-control variables

Variable	Option	Definition
SLMETH	1	Selects the alternating diagonal, D4, direct solver
	2	Selects the two-line, successive-overrelaxation iterative solver
	3	Selects the red-black, generalized conjugate-gradient iterative solver
	5	Selects the D4 zig-zag, generalized conjugate-gradient iterative solver
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

Table.5.3. List of options for numerical program-control variables — Continued

Variable	Option	Definition
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 performed
	50	Specified surface-flow rate with limiting surface pressure, allocation by mobility and pressure difference, well-riser calculations 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 implicit differencing
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	Printout of arrays by areal (x-y) layers
	13	Printout of arrays by vertical x-z or (r-z) slices
PRIxxx	0	No printout of this information
	n	Printout will occur every nth time step and at the end of the simulation

Table.5.3. List of options for numerical program-control variables — Continued

Variable	Option	Definition
	-n	Printout will occur at every n simulation time units
	-TIMCHG	Printout will occur only at the time at which new transient data will be read and at the end of the simulation

† 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.

‡ Intermediate weights can be specified for FDSMTH and FDTMTH.

6. OUTPUT DESCRIPTION

Various types of output files result from running the HST3D program. Most of the output is to files on a computer hard disc to be displayed on a monitor or sent to a printer. Some files are intended for post processing by graphical display programs. These files are written in ASCII format, except for the optional checkpoint/restart dumps, which are written in binary format.

6.1 SUMMARY OF OUTPUT FILES

The following list shows the primary names, identification suffix, and contents of each of the possible output files. The user-provided identification suffix is represented by *xxx*.

`Bcnfr.`*xxx*	Boundary nodal flow rates and cumulative flow amounts for post-process totalization
`Clog.`*xxx*	Calculation log of simulation
`Out.`*xxx*	General output file, static data, simulation definition information
`O.bal.`*xxx*	Global flow balance tables
`O.bcf.`*xxx*	Boundary-condition flow rates
`O.c.`*xxx*	Concentration field
`O.d.`*xxx*	Density field
`O.kd.`*xxx*	Conductance and dispersion-coefficient fields
`O.p.`*xxx*	Pressure field
`O.rst.`*xxx*	Checkpoint dump file for restart
`O.t.`*xxx*	Temperature field
`O.vel.`*xxx*	Velocity field
`O.vs.`*xxx*	Viscosity field
`O.wel.`*xxx*	Well output
`Plot.`*xxx*	Temporal data from wells for plotting
`Pmap.`*xxx*	Pressure, temperature, mass fraction fields for post-process visualization
`Pmap2.`*xxx*	Water-table elevation and density fields for post-process visualization
`Pzon.`*xxx*	Zone definition data for plotting
`Rde.`*xxx*	Read echo of input data
`Vmap.`*xxx*	Velocity components for post-process vector plots

A limitation on the computation of boundary-condition flow rates exists for cells that have multiple cell faces with the same type of boundary condition. The total net flow rates for all faces of that type of boundary condition are computed for those cells.

6.2 OUTPUT SELECTION

Much of the output is optional, and the numbers of the records containing the control variables in the data-input form of Table 5.1 are indicated. Output is generated at several stages during the simulation. The heading, title, array-dimensioning data, and geometrical information are always printed. The heading contains the program version and release number. The units employed for the output data are the same as those used for the input data, either metric or U.S. customary system as specified in record 1.4. The time unit for output is selected by the user in record 1.5. Table 5.2 and the input-record descriptions (section 5.2.2) give the U.S. customary or the metric units to be employed for a given simulation. For easier reading, variables are identified in the output by descriptors rather than program-variable names. A

read-echo file (`Rde.`*xxx*) contains a copy of each record of input data as it is read. The suffix *xxx* is provided by the user as requested by the simulator to identify a simulation run. 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. Individual output files are produced for selected variables, while the output of most static parameter information is contained in the general output file (`Out.`*xxx*). The individual output files are easier to examine because of their smaller size. The printout of array variables is arranged by horizontal or vertical slices as chosen by the user in record 2.23.3. Slices of a given array that do not contain any data are not printed.

Print intervals in time can be selected individually for the various types of information that are printed at the end of a time step. The intervals can be defined as number of time steps between printouts or the time interval between printouts. In the latter case, the print interval must be specified as an integer multiple of simulation time units. Furthermore, when the automatic time-step algorithm is being used, the maximum time-step length that the simulator can achieve will be limited by the smallest print interval selected. The types of transient information printed may include the solution-method information, the conductance and dispersion-coefficient distributions, the dependent-variable distributions, the density and viscosity distributions, the velocity distribution, the regional fluid-flow, heat-flow and solute-flow rates, the regional cumulative-flow results, boundary-condition flow rates, and the well-flow data (record 3.9.1). The selection of the dependent variables that will be printed is determined in record 3.9.2. Output of files for checkpoint dumps is controlled in record 3.9.3. Print interval and whether or not only the last time plane dumped is saved can be specified by the user. The amount of disc space needed can be much reduced if only the most recent dump is saved for restart after a computer system failure.

6.3 OUTPUT OF DATA FOR POST-PROCESSING INTO PLOTS

Data for contour maps or iso-surface plots of pressure, temperature, and mass fraction and for vector maps of velocity fields can be written to files (record 3.10.1). Data for temporal plots of selected variables also can be written to files (record 2.23.5). Data for plotting a zone-definition map can be written to a file (record 2.23.4). These plots can be produced by a post-processing program (not included with this distribution). The temporal plot data that may be output include well-datum pressure, well-surface pressure, well-datum temperature, well-surface temperature, and well solute-mass fraction (or scaled-mass fraction). For observation wells, the well-datum values are taken to be the values in the aquifer cell at the well-datum level.

The general formats of the output files for plotting are as follows (see section 5.2.2 for definitions of variables):

I. `Pmap.`*xxx* file
   1. Two title lines
   2. Logical parameters: HEAT, SOLUTE, EEUNIT, CYLIND
   3. Dimensioning parameters: NX, NY, NZ, NXY, NXYZ

4. Boundary-condition index array, IBC
5. Nodal x-location array, X
6. Nodal y-location array, Y
7. Nodal z-location array, Z
8. Time-value header
9. Pressure label
10. Pressure array, P
11. Temperature label
12. Temperature array, T
13. Mass-fraction label
14. Mass-fraction array, C
15. Repeated blocks of items 8-14 based on user-specified print interval

II. `Pmap2.`*`xxx`* file

1. Two title lines
2. Logical parameters: HEAT, SOLUTE, EEUNIT, CYLIND
3. Dimensioning parameters: NX, NY, NZ, NXY, NXYZ
4. Boundary-condition index array, IBC
5. Nodal x-location array, X
6. Nodal y-location array, Y
7. Nodal z-location array, Z
8. Time-value header
9. Free-surface index label
10. Free-surface node number array, MFSBC
11. Potentiometric-head label
12. Potentiometric-head array, H
13. Density label
14. Density array, DEN
15. Repeated blocks of items 8-14 based on user-specified print interval

III. `Vmap.`*`xxx`* file

1. Two title lines
2. Logical parameters: HEAT, SOLUTE, EEUNIT, CYLIND
3. Dimensioning parameters: NX, NY, NZ, NXY, NXYZ
4. Boundary-condition index array, IBC
5. Nodal x-location array, X
6. Nodal y-location array, Y
7. Nodal z-location array, Z
8. Time-value header
9. X-velocity label
10. Velocity in x-direction array, VXX
11. Y-velocity label
12. Velocity in y-direction array, VYY

13. Z-velocity label
14. Velocity in z-direction array, VZZ
15. Repeated blocks of items 8-14 based on user-specified print interval

IV. `Plot.`*xxx* file
1. Two title lines
2. Dimensioning parameter, NWEL
3. Well header
4. Static well data, one line per well: IWEL, XW, YW, ZWB, ZWT, WQMETH
5. Well header, transient output data
6. Transient well data: IWEL, WQMETH, TIME, PWKT, PWSUR, TWKT, TWSUR, CWKT, DENWKT
7. Repeated blocks of items 5 and 6 based on user-specified print interval

V. `Pzon.`*xxx* file
1. Two title lines
2. Logical parameters: HEAT, SOLUTE, EEUNIT, CYLIND
3. Dimensioning parameters: NX, NY, NZ, NXY, NXYZ
4. Dimensioning parameter: NPMZ
5. Zone definition data: IPMZ, X1Z, X2Z, Y1Z, Y2Z, Z1Z, Z2Z
6. Repeated lines of item 5 to define all zones

All arrays of dimension NXY and NXYZ are written in natural node number order. This means the x-direction index is incremented most rapidly, the y-direction index is incremented the next most rapidly, and the z-direction index is incremented the most slowly.

The pressure, temperature, and solute-concentration fields from the `Pmap.`*xxx* file from one simulation may be used to establish initial conditions for a subsequent simulation. Node-by-node input of initial conditions would be accomplished by copying the appropriate blocks of data from the `Pmap.`*xxx* file to the new data-input file.

6.4 OUTPUT OF DATA FOR FLOW AND AMOUNT CALCULATIONS

A post-processing program, BCFLOW, is available to totalize boundary flow rates and cumulative amounts by user-defined zones of boundary face areas. The BCFLOW program is described in Appendix B. The HST3D simulator calculates flow rates and cumulative amounts for each cell for each type of boundary condition. The trapezoidal rule is used for temporal integration of the flow rates to obtain the cumulative amounts. These values are written to a file for post-processing whereby total flow rates and cumulative amounts can be calculated for any desired zone of boundary-condition faces. The general format of the flow-rate and cumulative-amount output file is as follows (see section 5.2.2 for definitions of variables):

`Bcnfr.`*xxx* file
1. Two title lines
2. Logical parameters: HEAT, SOLUTE, EEUNIT, CYLIND
3. Simulation time unit index: TMUNIT

4. Dimensioning parameters: NX, NY, NZ, NXY, NXYZ
5. Boundary-condition index array, IBC
6. Nodal x-location array, X
7. Nodal y-location array, Y
8. Nodal z-location array, Z
9. Number of specified-value boundary-condition cells: NSBC
10. Specified-value boundary-condition index array: MSBC
11. Number of specified-flux boundary-condition cells: NFBC
12. Specified-flux boundary-condition index array: MFBC
13. Number of leakage boundary-condition cells: NLBC
14. Leakage boundary-condition index array: MLBC
15. Number of evapotranspiration boundary-condition cells: NETBC
16. Evapotranspiration boundary-condition index array: METBC
17. Number of aquifer-influence-function boundary-condition cells: NAIFC
18. Aquifer-influence-function boundary-condition index array: MAIFC
19. Number of heat-conduction boundary-condition cells: NHCBC
20. Heat-conduction boundary-condition index array: MHCBC
21. Time-value header
22. Flow-rate label
23. Specified-pressure boundary flow-rate array, QFSBC
24. Heat flow-rate label
25. Specified-pressure or -temperature boundary heat flow-rate array, QHSBC
26. Solute flow-rate label
27. Specified-pressure or -mass-fraction boundary solute flow-rate array, QSSBC
28. Cumulative amount label
29. Specified-pressure boundary cumulative mass flow-amount array, CCFSB
30. Cumulative volumetric-amount label
31. Specified-pressure boundary cumulative volumetric flow-amount array, CCFVSB
32. Cumulative heat flow-amount label
33. Specified-pressure or temperature boundary cumulative heat flow-amount array, CCHSB
34. Cumulative solute flow-amount label
35. Specified-pressure or -mass-fraction boundary cumulative solute flow-amount array, CCSSB
36. Flow-rate label (2 lines)
37. Specified-flux boundary flow-rate array, QFFBC
38. Heat flow-rate label
39. Specified-flux boundary associated heat flow-rate array
40. Solute flow-rate label
41. Specified-flux boundary associated solute flow-rate array
42. Heat flow-rate label
43. Specified-flux boundary heat flow-rate array, QHFBC
44. Solute flow-rate label

45. Specified-flux boundary solute flow-rate array, QSFBC
46. Cumulative amount label
47. Specified-flux boundary cumulative mass flow-amount array, CCFFB
48. Cumulative volumetric amount label
49. Specified-flux boundary cumulative volumetric flow-amount array, CCFVFB
50. Cumulative heat flow-amount label
51. Specified-flux boundary cumulative heat flow-amount array, CCHFB
52. Cumulative solute flow-amount label
53. Specified-flux boundary cumulative solute flow-amount array, CCSFB
54. Flow-rate label
55. Leakage boundary flow-rate array, QFLBC
56. Heat flow-rate label
57. Leakage boundary associated heat flow-rate array, QHLBC
58. Solute flow-rate label
59. Leakage boundary associated solute flow-rate array, QSLBC
60. Cumulative amount label
61. Leakage boundary cumulative mass flow-amount array, CCFLB
62. Cumulative volumetric amount label
63. Leakage boundary cumulative volumetric flow-amount array, CCFVLB
64. Cumulative heat flow-amount label
65. Leakage boundary cumulative heat flow-amount array, CCHLB
66. Cumulative solute flow-amount label
67. Leakage boundary cumulative solute flow-amount array, CCSLB
68. Flow-rate label
69. Evapotranspiration boundary flow-rate array, QFETBC
70. Heat flow-rate label
71. Evapotranspiration boundary heat flow-rate array, QHETBC
72. Solute flow-rate label
73. Evapotranspiration boundary solute flow-rate array, QSETBC
74. Cumulative amount label
75. Evapotranspiration boundary cumulative mass flow-amount array, CCFETB
76. Cumulative volumetric amount label
77. Evapotranspiration boundary cumulative volumetric flow-amount array, CCFVEB
78. Cumulative heat flow-amount label
79. Evapotranspiration boundary cumulative heat flow-amount array, CCHETB
80. Cumulative solute flow-amount label
81. Evapotranspiration boundary cumulative solute flow-amount array, CCSETB
82. Flow-rate label
83. Aquifer-influence-function boundary flow-rate array, QFAIF
84. Heat flow-rate label
85. Aquifer-influence-function boundary heat flow-rate array, QHAIF

86. Solute flow-rate label
87. Aquifer-influence-function boundary solute flow-rate array, QSAIF
88. Cumulative amount label
89. Aquifer-influence-function boundary cumulative mass flow-amount array, CCFAIF
90. Cumulative volumetric amount label
91. Aquifer-influence-function boundary cumulative volumetric flow-amount array, CCFVAI
92. Cumulative heat flow-amount label
93. Aquifer-influence-function boundary cumulative heat flow-amount array, CCHAIF
94. Cumulative solute flow-amount label
95. Aquifer-influence-function boundary cumulative solute flow-amount array, CCSAIF
96. Heat flow-rate label
97. Heat-conduction boundary heat flow-rate array, QHCBC
98. Cumulative heat amount label
99. Heat-conduction boundary cumulative heat flow-amount array, CCHHCB
100. Repeated blocks of items 21-99 based on user-specified print interval.

For the above sets of output, only data for boundary condition types included in a given simulation are written to the file. Furthermore, the output arrays for any component (heat or solute) not being transported are omitted.

6.5 DIAGNOSTIC OUTPUT

The HST3D simulator provides five forms of diagnostic output. The forms are discussed by output file name in alphabetical order. The user-provided identification suffix is represented by *xxx*.

The `Clog.`*xxx* file contains basic dimensioning information, a printout of the IBC array, and numerical solution information for each time step. The IBC array is the index of boundary-condition types and excluded cells for the simulation mesh. The internal format of the 9-digit number is given in Table.6.1 to enable interpretation of the boundary-condition specifications.

Table 6.1. Structure of the IBC array

n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	
	Flow			Heat			Solute		
x	y	z	x	y	z	x	y	z	
1	0	0	1	0	0	1	0	0	Specified value
2	2	2	2	2	2	2	2	2	Specified flux
3	3	3	0	0	0	0	0	0	Leakage
4	4	4	0	0	0	0	0	0	Aquifer influence
0	0	0	4	4	4	0	0	0	Heat conduction
5	5	5	0	0	0	0	0	0	Flux and river leakage
								-1	Excluded cell

The digits indicate the type of boundary condition and the x, y, or z positions indicate which boundary-cell face has that condition. For the representative examples shown in Table 6.1, each digit is shown in all valid x, y, or z positions for that type of boundary condition. A given IBC number may have a combination of the example values. An IBC value of -1 indicates that a cell is outside the simulation

region. The solution information includes iteration counts and maximum changes in the dependent variables written out at a user-specified interval.

The `O.kd.`*xxx* file contains conductance factors and the dispersive conductances for heat and solute transport. These dispersive conductances are functions of the velocity field and thus may be transient. This file is used primarily for diagnostic purposes.

The `Out.`*xxx* file contains general information defining the simulation problem. This includes dimensioning information, parameter information for the fluid and porous medium, initial condition information, and boundary-condition information.

The first set of data in the `O.p.`*xxx*, `O.t.`*xxx*, `O.c.`*xxx*, `O.d.`*xxx*, and `O.vs.`*xxx* files also may contain the initial condition distributions of the variables for pressure, temperature, concentration, density, and viscosity, respectively. The user must specify that the initial conditions are to be written, if desired.

The `Rde.`*xxx* file contains a record-by-record echo of the lines read from the data-input file. The data read by x,y,z range zones is reformatted. This file is most helpful in locating where the data reading routines have lost synchronization with the data-input file.

An additional source of diagnostic information is the file `Out.`*xxx* from the DIMEN pre-processor program. It contains dimensioning information, error messages about array sizes that are too small, and some of the static data for the simulation, including the IBC array.

7. COMPUTER SYSTEM CONSIDERATIONS

The heat- and solute-transport simulation program was developed initially on a Control Data Cyber 170/720 computer, a Prime 9950 computer, a Masscomp 5400 workstation, and finally on a Data General AViiON-530 workstation. The language used for this program is FORTRAN-77, although some FORTRAN-IV coding still exists. Only the ANSI standard FORTRAN-77 (American National Standards Institute, 1978) was used as far as practical to provide maximum portability. Double-precision arithmetic has been used for all real variables.

In the design of Version 2, the availability of tens of megabytes of random-access memory (RAM) and of hundreds of megabytes of virtual memory on most computer workstations was recognized. All arrays were put into common blocks with their dimensions set by Fortran PARAMETER statements. INCLUDE statements (a Fortran extension) were used to incorporate the PARAMETER and common block statements into each subroutine as necessary. The storage requirement on a Data General computer for the executable-code module is about 1 Megabyte, exclusive of the arrays, when compiled with full optimization by the Green Hills Fortran-88000 Compiler, Version 1.8.6.1. 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 affecting program portability between the Cyber, Prime, Masscomp, and Data General computers.

The Green Hills Fortran-88000 Compiler Version 1.8.6.1 forces static storage of variables during program execution. 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 static should be avoided. Variables are explicitly initialized where necessary, so no reliance is made on system-default initialization. There are no alternative entry points into any of the subroutines.

The Fortran unit numbers for the input and output files are set in a PARAMETER statement. This allows for easy modification to suit different operating systems. Some operating systems require the BLOCKDATA subprogram to be contained in the same file as an executable subprogram. Version 2 of the code, as distributed, has the main program, and each subprogram, including BLOCKDATA, contained in a separate file.

7.1 COMPUTER SYSTEM REQUIREMENTS

To successfully run a set of simulations in a reasonable time, an appropriate amount of computational power and storage needs to be available. As an example, a field-scale simulation involving about 44,000 nodes required an executable module of about 10 Megabytes (MB), an input file of 1.6 MB, and generated 12 MB of output files for a 39 time-step simulation. On the Data General-530 this simulation took about 3 hours of central-processing-unit (CPU) time, whereas on the DEC Alpha server it took 15 minutes of CPU time. A desktop workstation with a small amount of memory and hard disc space will be limited to simulating smaller problems for shorter simulation times than a more powerful

computer system.

7.2 COMPUTER PORTING EXPERIENCE

The HST3D computer code has been ported to the computer platforms with the operating systems listed in Table 7.1. However, the nature of program support is at the source code level, and full support at the operating system level for a given platform can not be provided.

Table 7.1. Platforms to which the HST3D code has been ported

Platform	Vendor	Hardware	Operating system
Unix	Data General	AViiON	DGUX
	Masscomp	M680xx	RTU
	IBM	RS/6000	AIX
	Digital	Alpha	Digital Unix
	HP	PaRisc	HP/UX
	Intel	Intel x86	Linux
	SUN	Sparc	Solaris 2
Windows	Intel	Intel x86	Windows 95

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. Verification is a continuing process, as many combinations of program options need to be tested.

8.1 SUMMARY OF VERIFICATION TEST PROBLEMS

All of the test problems cited in the document for HST3D, Version 1, were executed correctly by Version 2. The evapotranspiration boundary condition was verified by hand calculation of a small, 3×3×3 node test problem. Five additional test problems have been used for verification of HST3D, Version 2, and are summarized below. One simulates variable density flow and heat transport, while the other four simulate variable density flow and solute transport. In the cases described below, a centrally weighted finite-difference approximation to the advective term in the transport equations was used unless otherwise noted.

8.1.1 Thermal Convection Experiment: Elder's Problem

This test problem is from Elder (1967) and is presented by Ségol (1994) as one of her collection of benchmark problems. It represents thermal convection in a cross-sectional region with a segment of the bottom boundary held at a fixed temperature greater than the initial uniform temperature in the region. This problem tests the ability of the simulator to represent fluid flow driven purely by density differences created by temperature variations. Elder (1967) presented his results graphically for the temperature and stream-function fields.

The region was a two-dimensional, closed rectangular box with impermeable walls on all four sides, (Fig. 8.1). A source of heat at constant temperature was applied at part of the bottom boundary while the temperature at the top boundary was kept at the initial temperature. The initial condition was that of hydrostatic pressure and a uniform temperature. Heat was conducted into the water at the bottom boundary which decreased its density and induced a circulating flow. Elder's (1967) solution to the thermal convection problem was obtained using finite-difference techniques and a stream-function formulation. Elder (1967) formulated his equations in terms of dimensionless variables only, so a dimensional model of a laboratory-scale experiment with water in a porous medium was set up for this verification simulation. The dimensions of the simulation region were 4 m long and 1 m high with the heat source along the middle 2 m of the bottom boundary. A specified pressure of 0 Pa (atmospheric) was imposed at the upper corners of the region. The region was uniformly discretized with a node spacing of 0.091 m in the horizontal direction and 0.040 m in the vertical direction. The other parameters were set to the following values:

- Porosity, 0.10
- Permeability, 9.084×10^{-9} m^2
- Thermal conductivity of porous medium, 2.0 W/m–°C
- Heat capacity of porous medium, 2.0×10^{6} J/m^3–°C

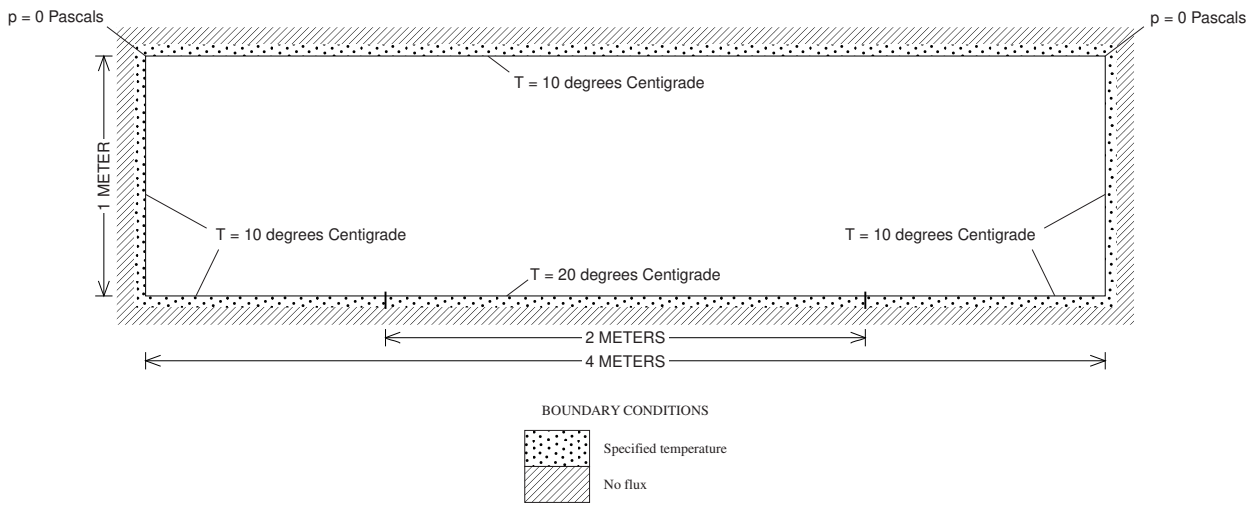


Figure 8.1. System geometry and boundary conditions for Elder's problem

- Density of water, 1,000 kg/m^3
- Fluid viscosity, 0.001 Pa–s
- Dispersivity, 0 m
- Heat capacity of the fluid, 4,182 J/kg–°C
- Thermal conductivity of the fluid, 0.6 W/m–°C
- Thermal expansion coefficient of the fluid, 2.0×10^{-4} °C^{-1}
- Temperature at the bottom boundary, 20 °C
- Temperature at the top boundary, 10 °C
- Initial temperature, 10 °C

A uniform time step of 496 s was used and the simulation was run for 1.1925×10^5 s. This corresponds to a dimensionless time of 0.1 to match the duration of Elder's (1967) simulation. Backwards-in-time and centered-in-space differencing were used.

The results for the dimensionless temperature field are contoured in Figure 8.2 for six values of dimensionless time. Figure 8.3 shows the contoured dimensionless temperature fields obtained by Elder (1967). Note the vertical exaggeration by a factor of two in both of these figures. The results are in visual agreement for the first four time values and the lobes of the plumes of buoyant, warm fluid are apparent. The HST3D simulation shows a persistence of the three fingers at 0.6 dimensionless temperature units in the last two diagrams. Other simulations indicate the possibility that the Elder (1967) results have a greater amount of numerical dispersion than the HST3D results.

8.1.2 Solute Convection Experiment: Modified Elder's Problem

A test problem was designed by transforming the thermal convection example of Elder (1967) into a solute convection problem as done by Voss and Souza (1987). This solute convection problem is also

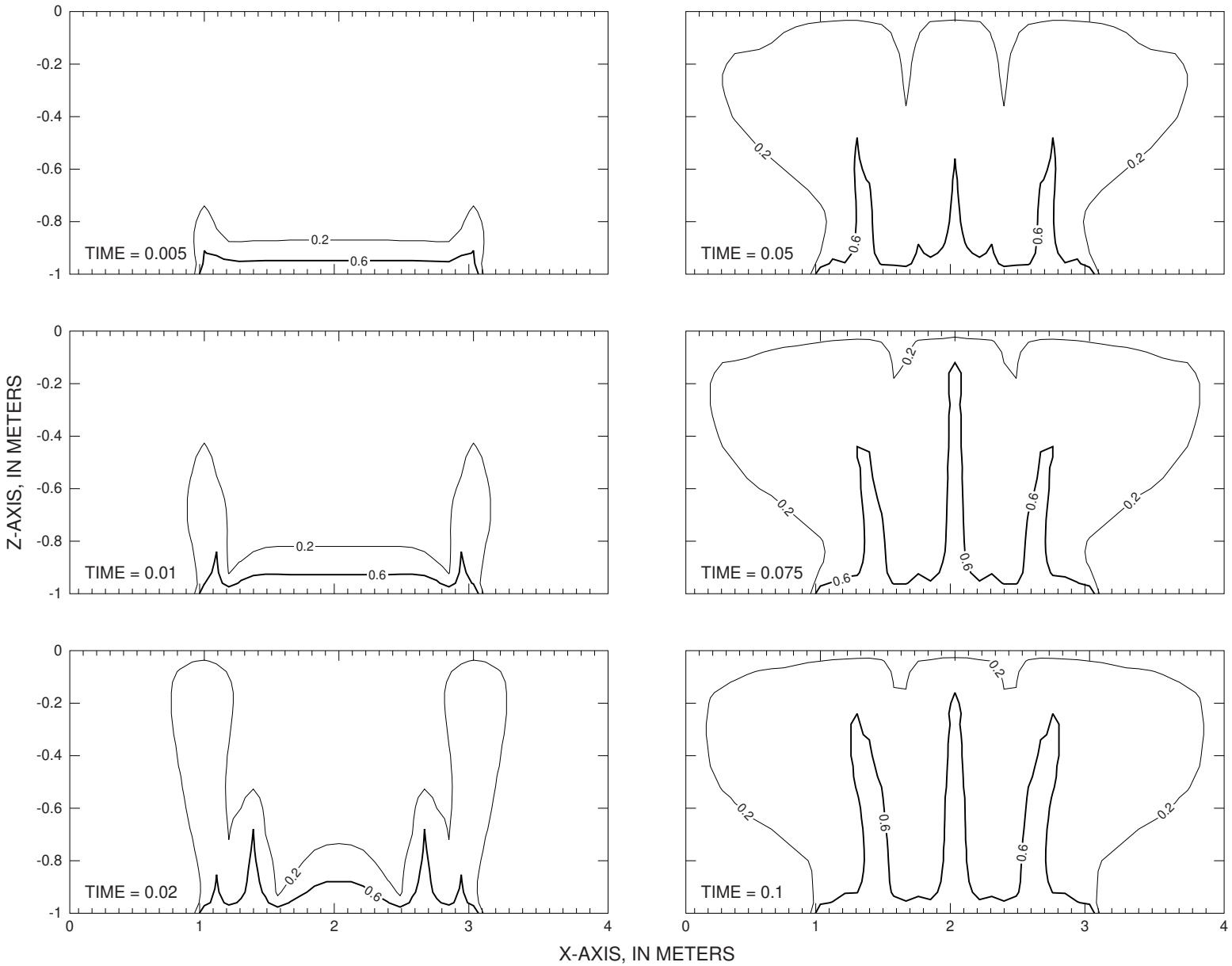


Figure 8.2. Scaled temperature Þeld at six dimensionless time values for Elder's problem from HST3D.

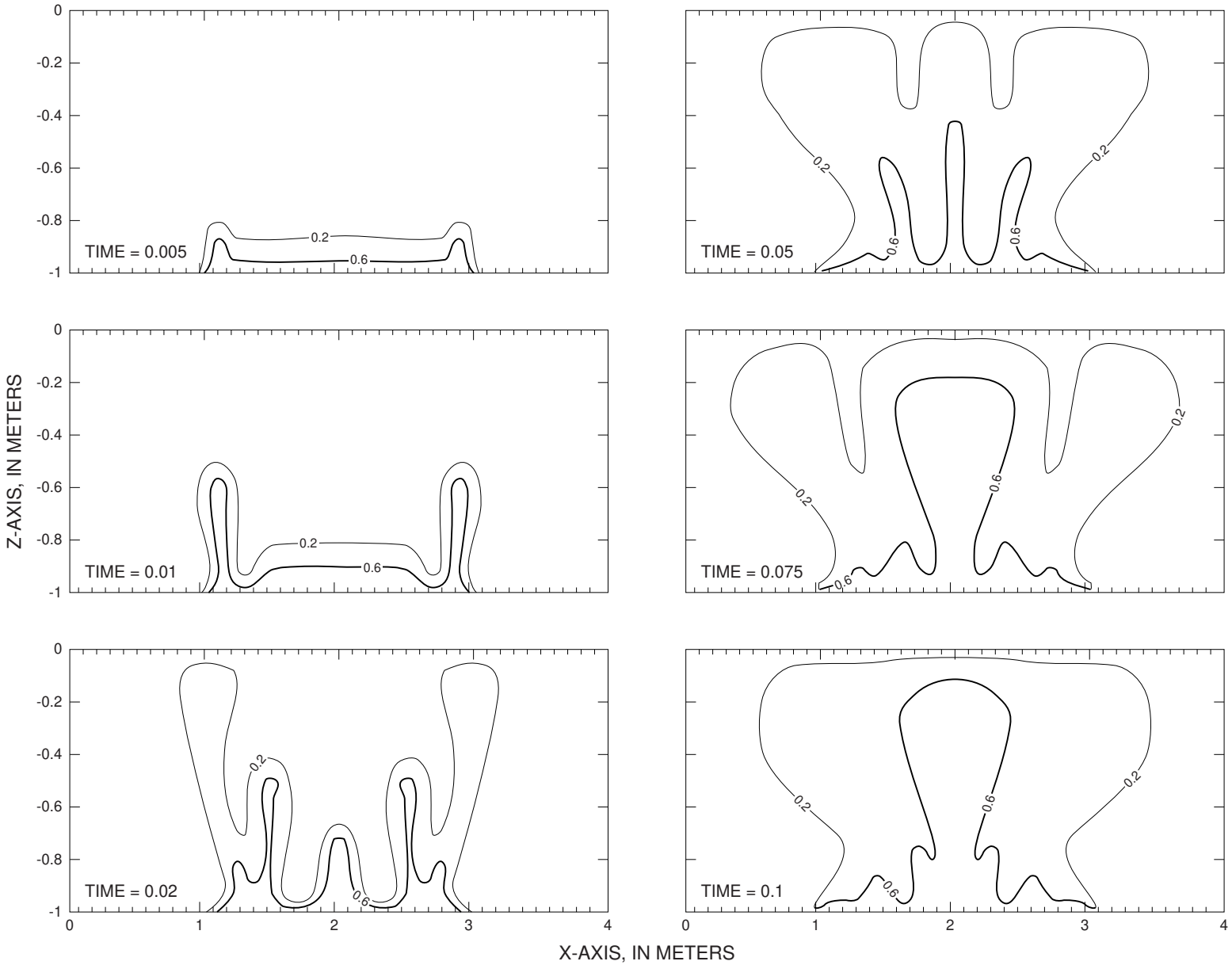


Figure 8.3. Scaled temperature Þeld at six dimensionless time values for Elder's problem from Elder (1967) (used with permission).

presented in Ségol (1994, p. 277-282). It tests the ability of the simulator to represent fluid flow driven purely by density differences created by variations in solute concentration.

The region was a two-dimensional, closed rectangular box with impermeable walls on all four sides, (Fig. 8.4). A source of solute at constant concentration was applied over the middle part of the top boundary while the concentration at the bottom boundary was kept at zero. The initial condition was that of hydrostatic pressure and no solute in the region. The solute diffused into the water which induced a circulation flow as a result of the density increase.

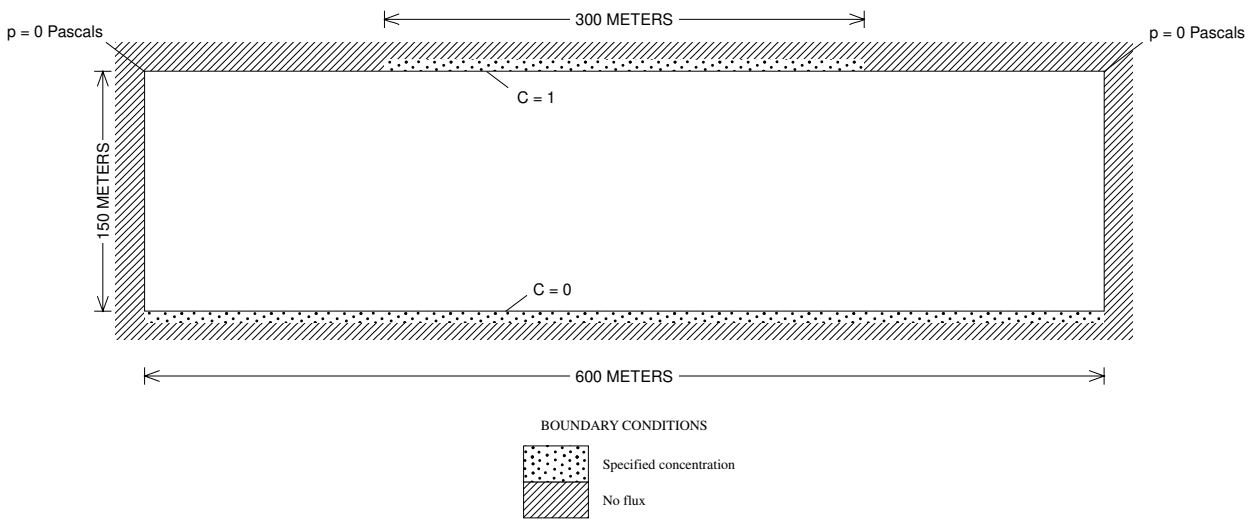


Figure 8.4. System geometry and boundary conditions for modified Elder's problem for solute.

Elder's (1967) solution to the thermal convection problem was obtained in dimensionless form and was converted to solute concentration. The parameters used by Voss and Souza (1987) were used for the HST3D simulation. The dimensions of the simulation region were 600 m long and 150 m high with the solute source along the middle 300 m of the top boundary. A specified pressure of 0 Pa (atmospheric) was imposed at the upper corners of the region. The region was uniformly discretized with a node spacing of 13.636 m in the horizontal direction and 6.0 m in the vertical direction. A scaled mass fraction was used ranging from 0 to 1. The other parameters were set to the following values:

- Porosity, 0.10
- Permeability, 4.845×10^{-13} m^2
- Density of fresh water, 1,000 kg/m^3
- Density of salt water at maximum mass fraction, 1,200 kg/m^3
- Maximum salt mass fraction for scaling, 0.05
- Fluid viscosity, 0.001 Pa–s
- Dispersivity, 0 m
- Diffusion coefficient, 3.565×10^{-6} m^2/s

A uniform time step of 1 month was used and the simulation was run for 20 years. The difference

equations were formed using backwards-in-time and centered-in-space differencing.

The results for the concentration field at six time values are contoured in Figure 8.5. For comparison, Figure 8.6 shows the contoured concentration fields obtained by Voss and Souza (1987) and the equivalent concentration field derived from the temperature field obtained by Elder (1967). Note the vertical exaggeration by a factor of two in both of these figures. The results agree visually and the lobes of the plumes of dense, saline fluid are apparent. The HST3D simulation captures much more of the detail of the multiple lobes than the SUTRA simulation, and the three fingers of more than 0.6 scaled concentration persist throughout the simulation period. The numerical dispersion coefficient was more than 10 times larger than the physical dispersion coefficient over some parts of the region. The lobes of saline water at the 0.6 concentration do not extend as far into the region as those of Elder (1967) at 2 and 4 years, but extend farther at 10, 15, and 20 years. Additional simulations indicated that mesh convergence in time and space had probably not been reached for any of the three simulations of Elder (1967), Voss and Souza (1987), or the present work.

8.1.3 Saltwater Intrusion into a Coastal Aquifer: Henry's Problem

This test problem is from Henry (1964) and is one of the few variable-density problems for which there is an analytical solution. A complete documentation of this problem and several numerical solutions are given by Ségol (1994, p. 221-276). The specific problem formulation used here follows that of Henry (1964) and Ségol (1994).

The domain was rectangular with a length of 2 m and a depth of 1 m, as shown in Figure 8.7. A constant, uniform flux of freshwater was applied to the left boundary. The right side was a salt water boundary with a specified pressure under hydrostatic conditions. The top and bottom boundaries were impermeable. A specified salt concentration of sea water (scaled to 1) was specified at the right boundary, (a) in Figure 8.7. This was the original boundary condition used by Henry (1964). Others, such as Voss and Souza (1987), used an associated seawater concentration for incoming flow at the right boundary, (b) in Figure 8.7. Use of this other boundary condition gave somewhat different results from those of Henry (1964) and Ségol (1994). The spatial discretization was uniform with a mesh spacing of 0.1 m. The time step was automatic with a pressure change constraint of 100 Pa and a solute concentration change constraint of 5 percent and a starting time step of 10 s with fully implicit time differencing. The other parameters used for the simulation were the following:

- Hydraulic conductivity, 1.0×10^{-2} m/s
- Permeability, 1.020408×10^{-9} m^2
- Porosity, 0.35
- Freshwater density, 1,000.00 kg/m^3
- Seawater density, 1,024.99 kg/m^3
- Seawater mass fraction, 0.0357
- Dispersivity, 0 m
- Diffusion coefficient, 18.8571×10^{-6} m^2/s
- Freshwater flux, 6.6×10^{-5} m^3/m^2-s

Simulation duration was 60,000 s at which time the system was assumed to have reached steady-state conditions. Figure 8.8 shows the scaled solute-concentration contours from HST3D and Figure 8.9

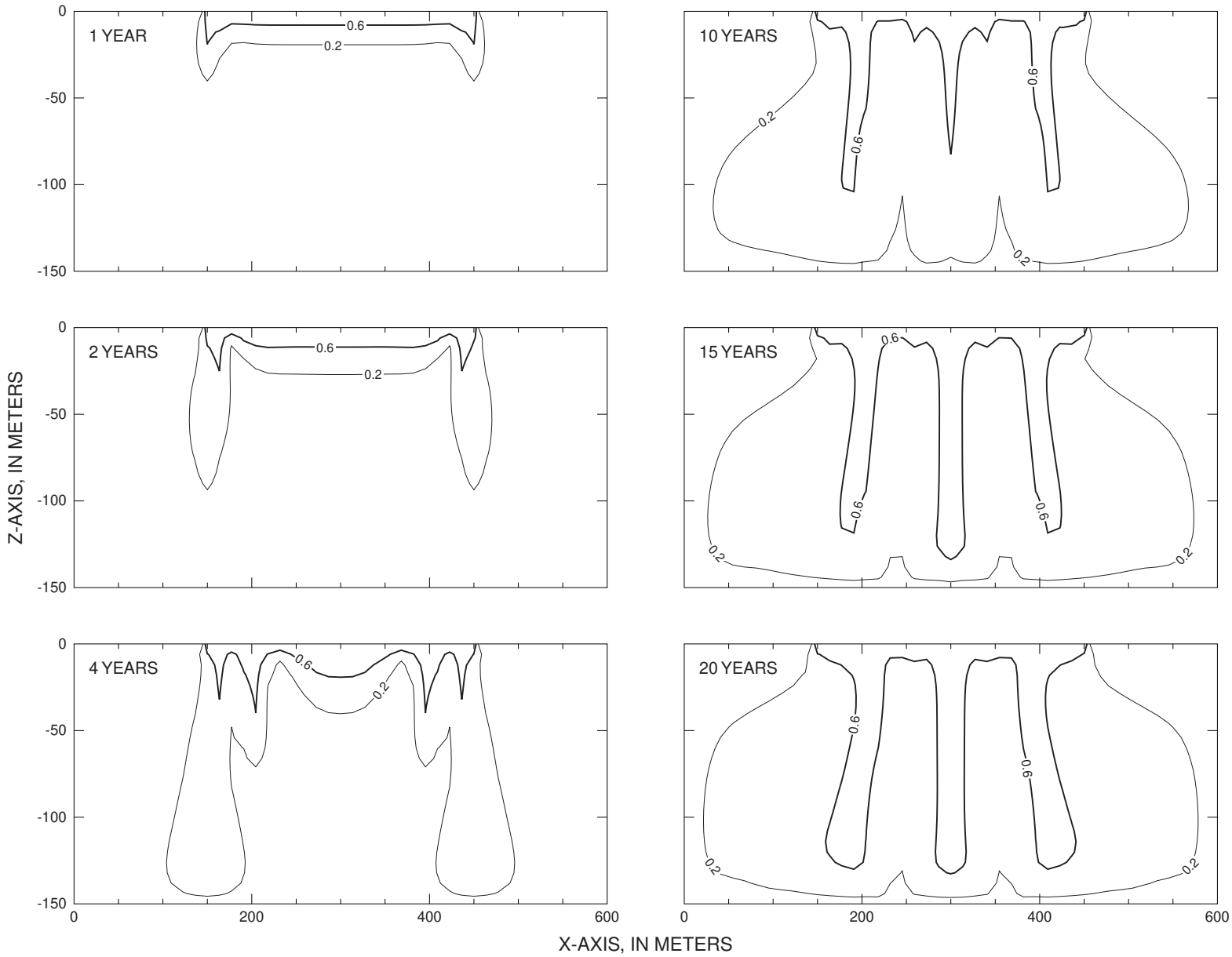


Figure 8.5. Scaled concentration field at six time values for modified Elder's problem from HST3D.

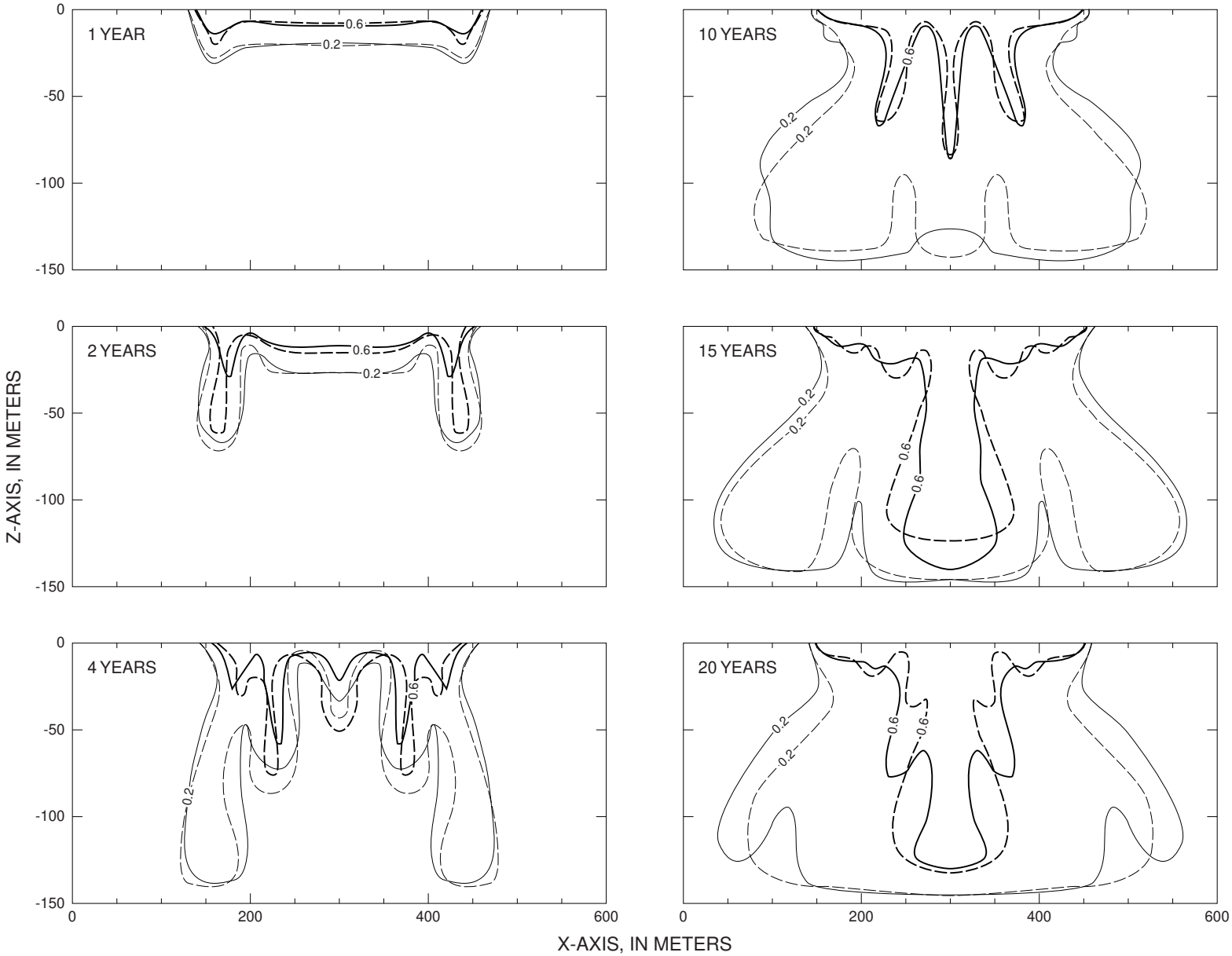


Figure 8.6. Scaled concentration field at six time values for modified Elder's problem from Elder (1967) (used with permission), dashed, and Voss and Souza (1987), solid.

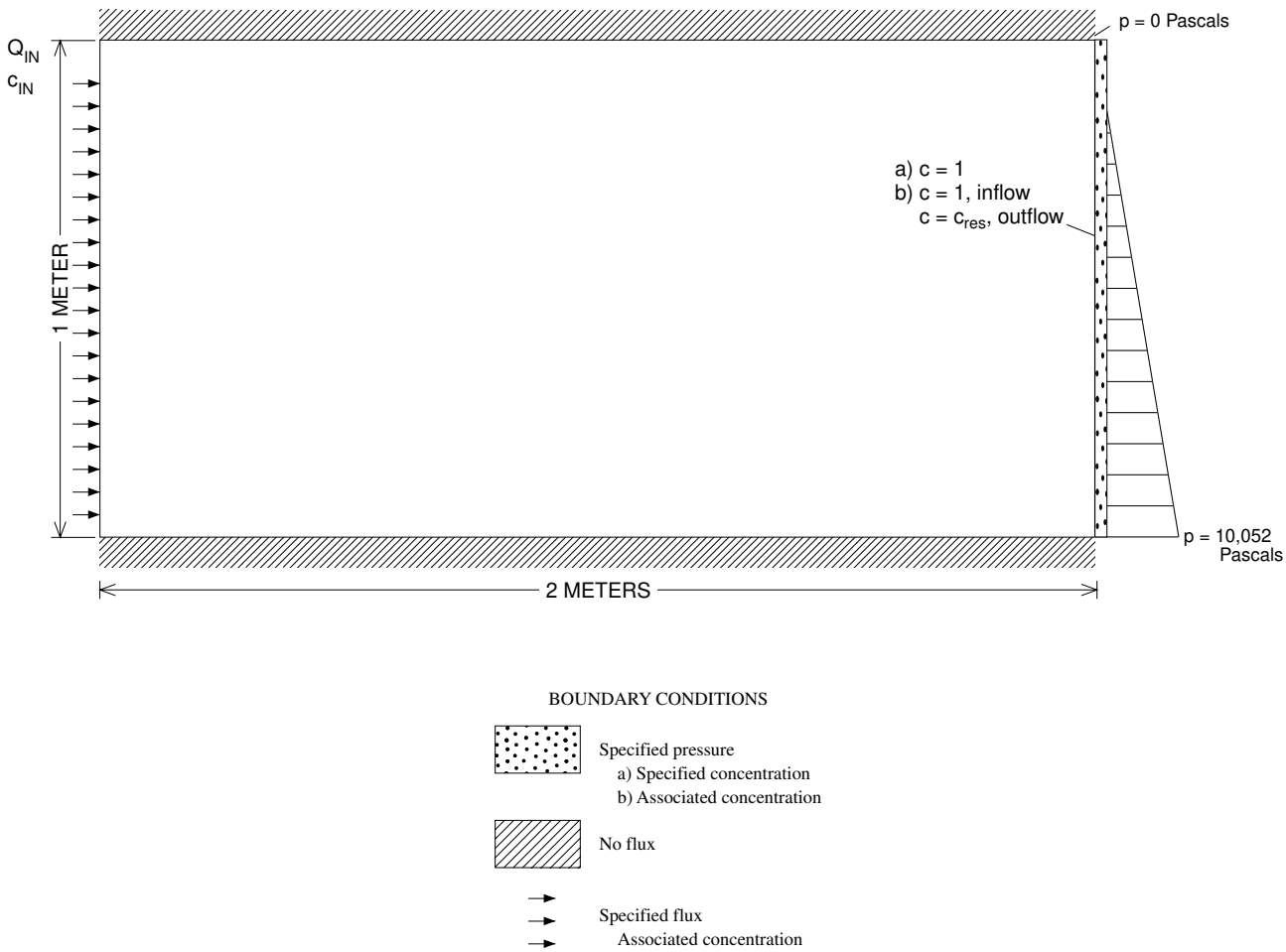


Figure 8.7. Region geometry and boundary conditions for Henry's problem.

shows contours from the analytical solution of Ségol (1994, p. 274). The HST3D results are similar to the analytical solution except in the upper right corner area where a slight numerical oscillation can be seen. There is also a perturbation in the 0.10 scaled-concentration contour of the analytical solution of Ségol (Fig. 8.9). Refining the mesh by a factor of 2 in each direction caused the concentration gradient to increase in the upper right corner and nearly eliminated the numerical oscillation. However, the concentration gradient is greater in both HST3D simulations than in the analytical solution of Ségol (1994).

8.1.4 Ground-water Flow over a Salt Dome: HYDROCOIN Problem

The HYDROCOIN (Hydrologic Code Intercomparison) project was an international cooperative effort to better understand the various analytical and numerical methods used to describe ground-water flow and transport. This test problem is case 5 (Organization for Economic Cooperation and Development (OECD), 1988) of seven test cases forming the Level 1 set.

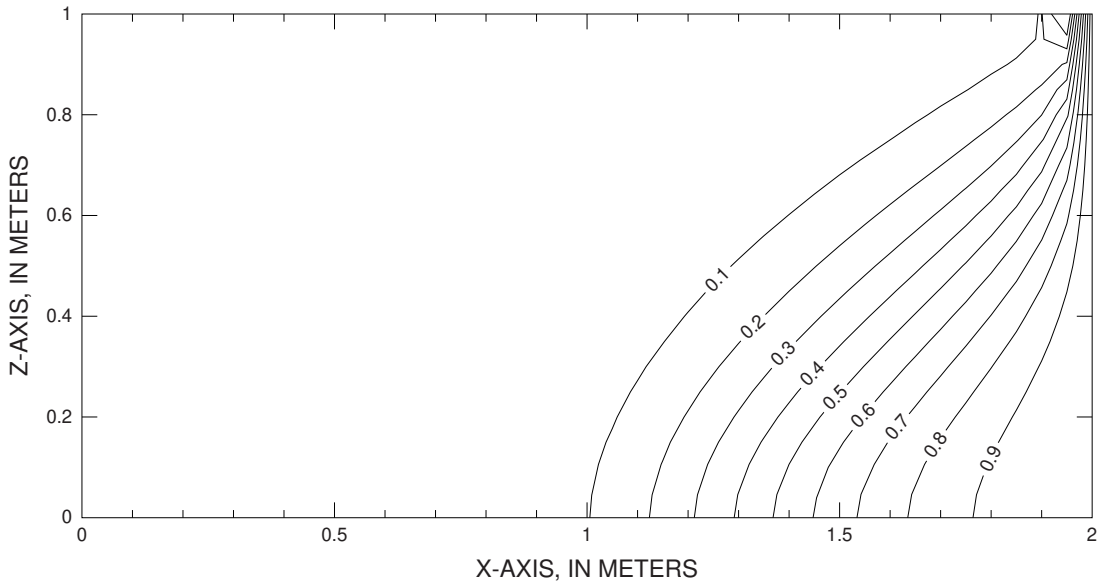


Figure 8.8. Scaled concentration contours for Henry's problem from HST3D.

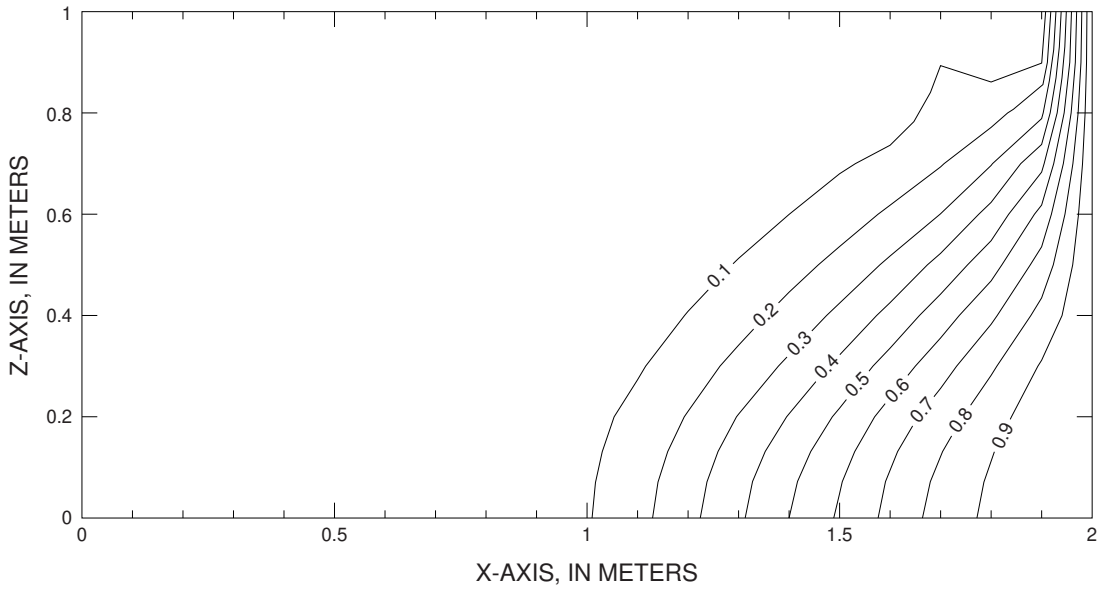


Figure 8.9. Scaled concentration contours for Henry's problem from Ségol (1994).

The problem was a simplified model of ground-water flow over a salt dome in a two-dimensional rectangular region. Dense brine formed from dissolution of the salt which created a saline plume with a steady-state configuration. There was a large fluid density contrast over the region that made this test problem particularly difficult to simulate numerically. No analytical solution was available for this problem, so the HST3D simulation results were compared with those from the SUTRA simulator (Voss, 1984).

The simulation region was 900 m long and 300 m deep, with boundary conditions as shown in Figure 8.10. A linear pressure distribution was applied across the top surface. The initial condition was pure water with a hydrostatic pressure distribution. The spatial discretization was uniform with a mesh spacing of 15 m in the horizontal direction and 10 m in the vertical direction. The time step was automatic with a pressure change constraint of 10 MPa, a solute concentration change constraint of 0.5 percent, and a starting time step of 10^{-6} y with fully implicit time differencing. Upstream-in-space differencing was used for the advective terms. It was critical to limit the maximum change in solute concentration so that the maximum change in density over a time step was less than 0.1 percent to avoid convergence failure with the Picard iteration between the flow and solute-transport equations. The other parameters used for the simulation appear below:

- Permeability, 1×10^{-12} m^2
- Freshwater density, 1,000 kg/m^3
- Brine density, 1,200 kg/m^3
- Brine mass fraction, 0.25
- Viscosity, 0.001 Pa-s
- Porosity, 0.20
- Dispersivity (longitudinal and transverse), 20 m and 2 m
- Diffusion coefficient, 0 m^2/s
- Pressure at the upper left corner of the region, 1×10^5 Pa

The flow and brine-transport system was simulated to a steady state. After 1,000 y of simulation, the system was assumed to have reached steady-state conditions. Actually, the fluid and solute flow balances were in equilibrium to three significant digits.

This test problem failed to converge as originally formulated, because of the large density contrast. Refining the mesh spacing to a uniform 3 m did not remove the convergence failure. An alternative initial condition was generated representing a steady-state salt concentration distribution with seawater salinity (.025) as the maximum boundary concentration and associated seawater density (1,025 kg/m^3). This initial condition led to a successful simulation of the brine problem. The salt concentration field is shown in Figure 8.11 for the HST3D simulation and in Figure 8.12 for the SUTRA simulation. The configurations of the solute plumes are similar, but the SUTRA plume extends farther up the right side boundary, particularly for the 0.10 scaled concentration contour.

The results of this test problem show the limitations of the Picard iteration method for handling the strong density contrasts encountered with the original initial conditions. Perhaps more restrictive controls on the automatic time step algorithm would have enabled the original formulation to give a successful simulation.

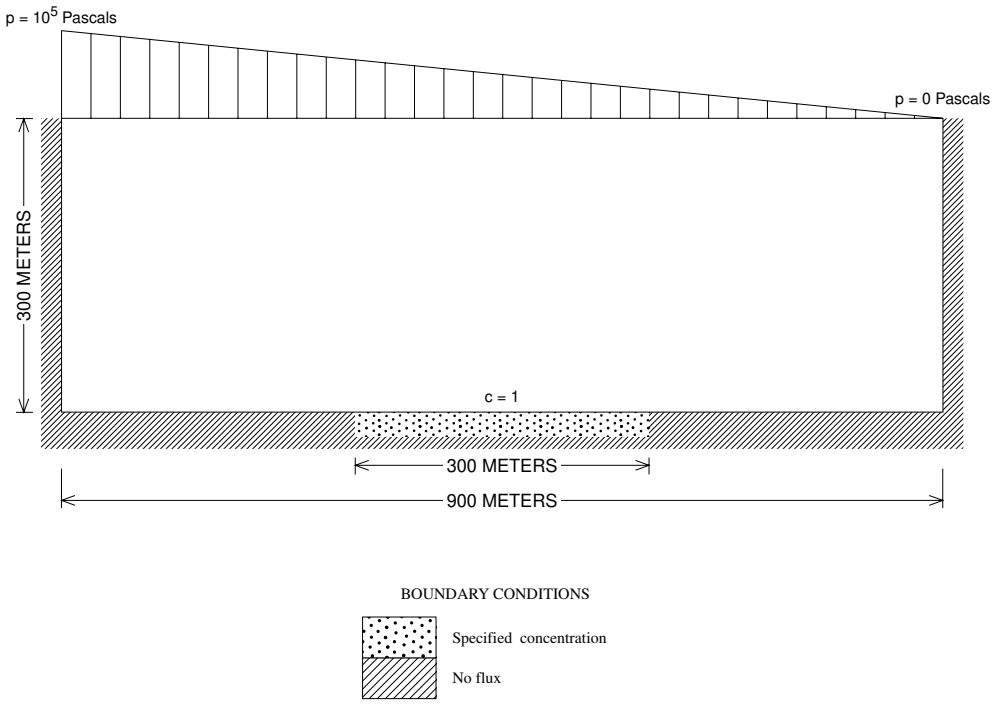


Figure 8.10. Region and boundary conditions for the HYDROCOIN problem.

8.1.5 Saltwater Intrusion into a Coastal Aquifer

This test problem is from Huyakorn and others (1987) and is a simplified, field-scale, three-dimensional simulation of saltwater intrusion at a coastline in response to two pumping wells. The problem is described schematically in Figure 8.13 showing the simulation region, boundary conditions, and well locations. This was a steady-state analysis of flow toward two wells , one screened in the upper 40 ft and the other screened in the lower 40 ft of the aquifer thickness. Withdrawal rates were 0.5×10^6 and 1×10^6 ft^3/d. Uniformly distributed recharge at 1 ft/yr was applied to the land surface. The remaining boundary conditions are depicted in Figure 8.13, showing a freshwater boundary on the inland side of the region and a seawater boundary at the coastal side. Conversion of boundary conditions from equivalent freshwater head to pressure was necessary for the HST3D simulator. In addition, the boundary condition over the seawater face had to be modified to a uniform head in order to obtain results that closely matched those of Huyakorn and others (1987). Also note that Huyakorn and others (1987) used a left-handed coordinate system with the y-axis pointing vertically downward. Aquifer parameters were:

- Hydraulic conductivity (horizontal and vertical), 2000 ft/d and 200 ft/d

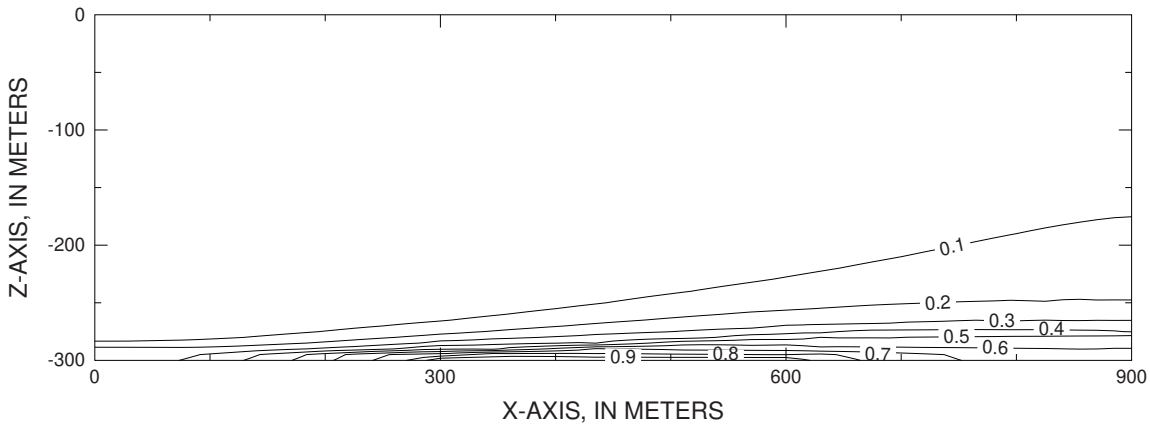


Figure 8.11. Scaled brine concentration field for the HYDROCOIN problem from HST3D.

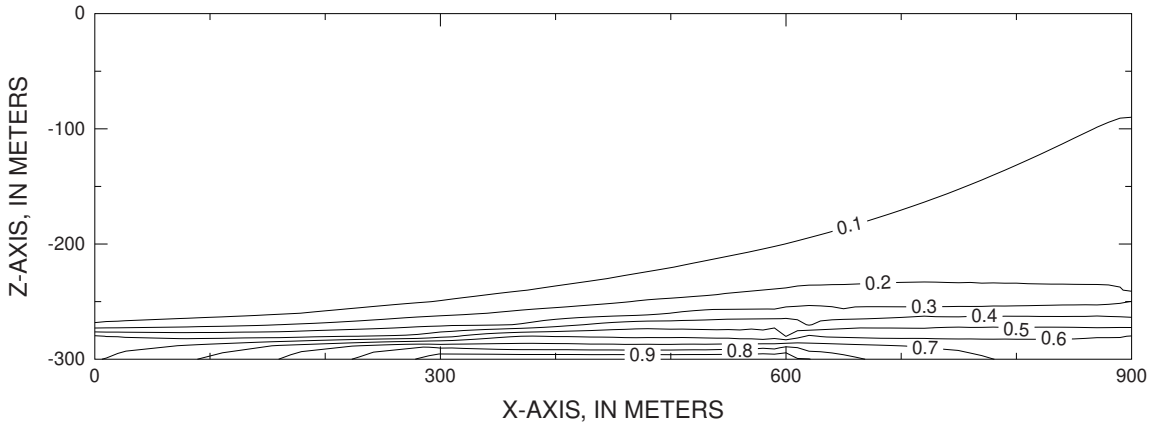


Figure 8.12. Scaled brine concentration field for the HYDROCOIN problem from SUTRA.

- Dispersivity (longitudinal and transverse), 150 ft and 30 ft
- Porosity, 0.20
- Density difference between freshwater and saltwater, 25 kg/m^3
- Saltwater salinity, 35 parts per thousand
- Freshwater density, 62.4 lb/ft^3

The fluid and porous matrix were considered incompressible for this unconfined aquifer. The flow region was discretized using a uniform grid of 62 nodes in the x-direction, 19 nodes in the y-direction, and 11 nodes in the z-direction. Huyakorn and others (1987) used an irregularly spaced finite-element mesh in the horizontal plane, but they did not specify its spacing. Simulations were done for 25,000 days from an initial hydrostatic condition of freshwater. The system was in equilibrium after this time as shown by the

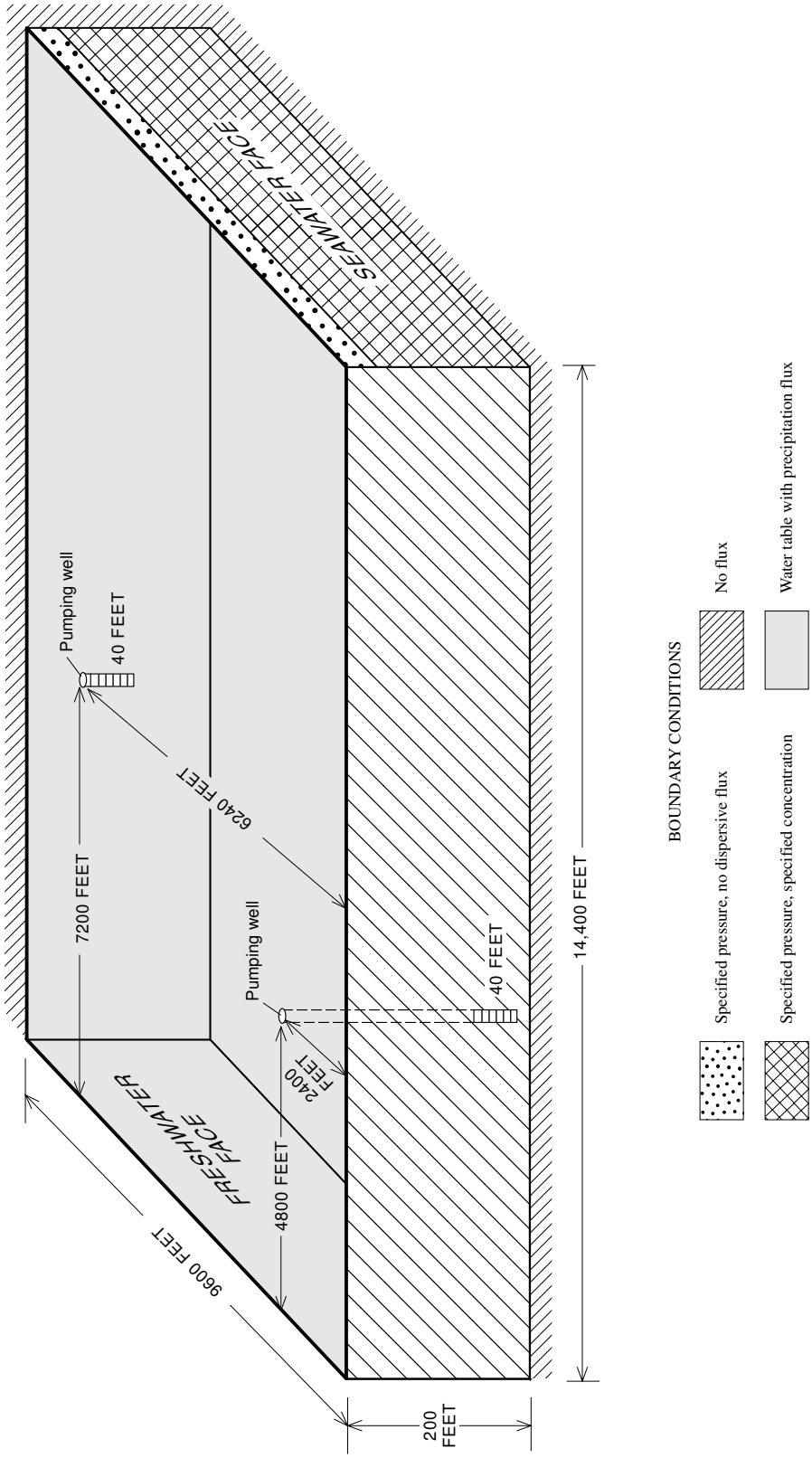


Figure 8.13. System geometry and boundary conditions for the coastal aquifer problem of Huyakorn and others (1987)

solute flow being in balance to five significant digits. An automatic time step adjustment was used with a pressure change constraint of 0.5 psi, a scaled solute concentration change constraint of 5 percent, and an initial time step of 0.01 days.

Figures 8.14-16 show results from HST3D and Figures 8.17-19 show the results of Huyakorn and others (1987). Results from Huyakorn and others (1987) were only available as areal plots of scaled salt concentration, plots of interstitial ground-water velocity vectors at the top and bottom of the aquifer, and as cross-sectional plots of concentration and velocity at the plane of the well located at y = 2400 ft.

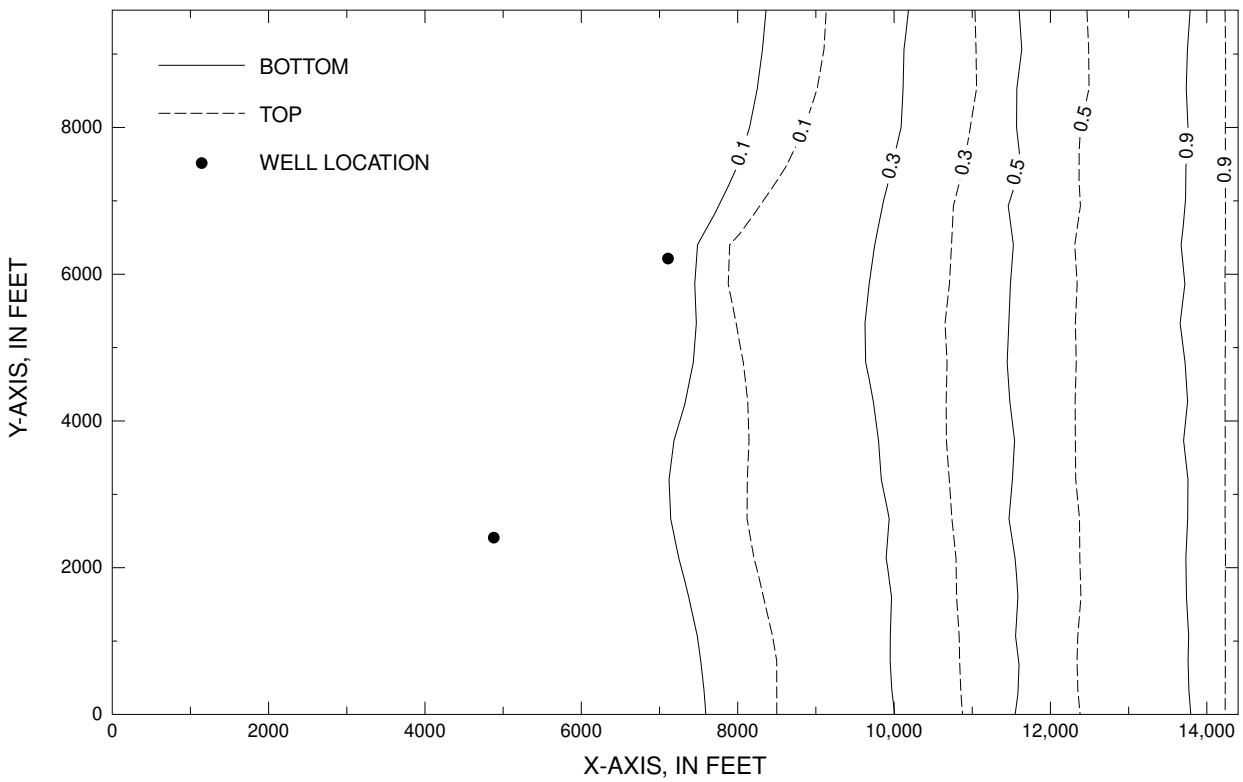


Figure 8.14. Areal view of contours of scaled salt concentration at the bottom and top of the aquifer for the Huyakorn problem from HST3D.

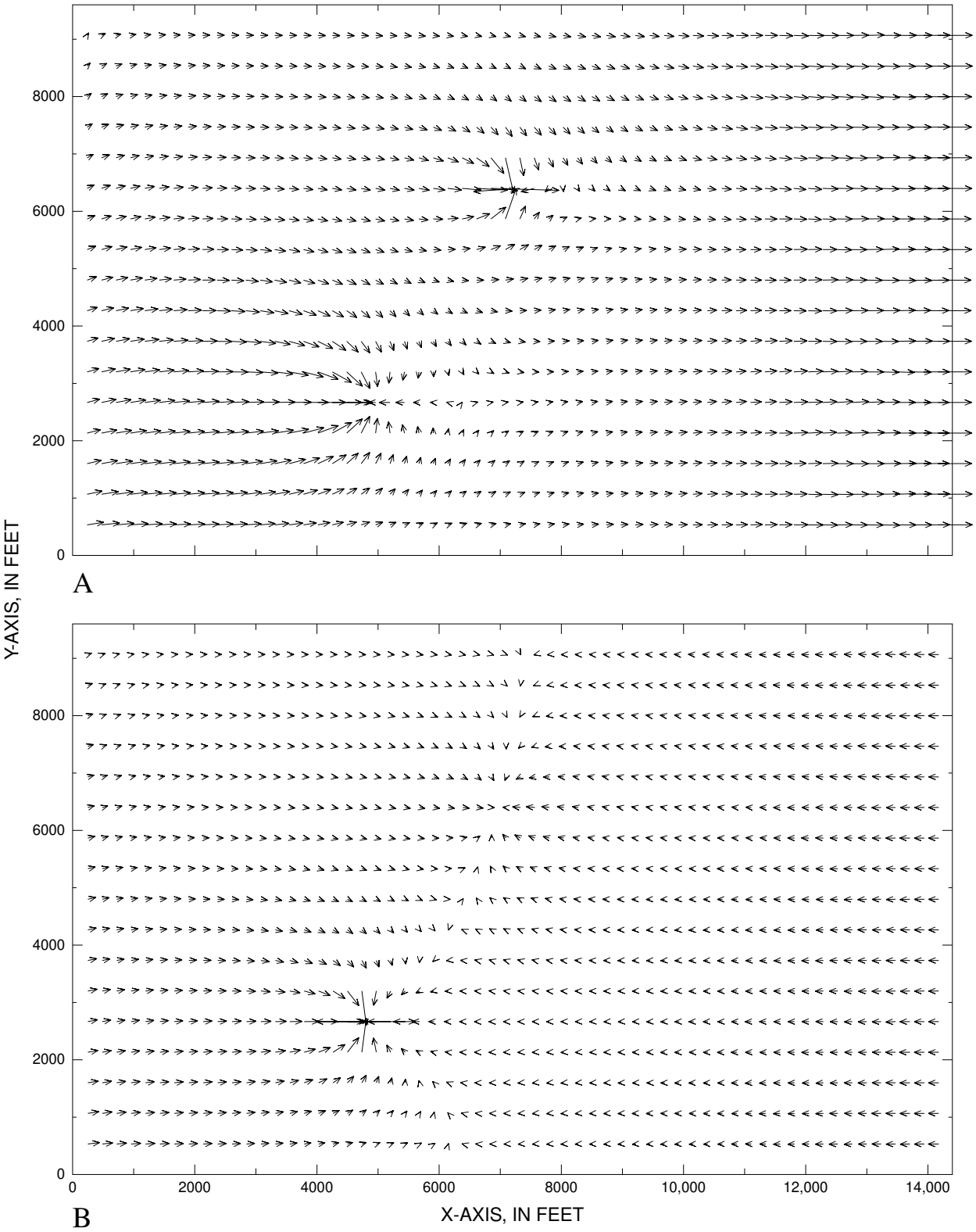


Figure 8.15. Areal view of the horizontal components of the velocity vectors (A) at the top and (B) at the bottom of the aquifer for the Huyakorn problem from HST3D.

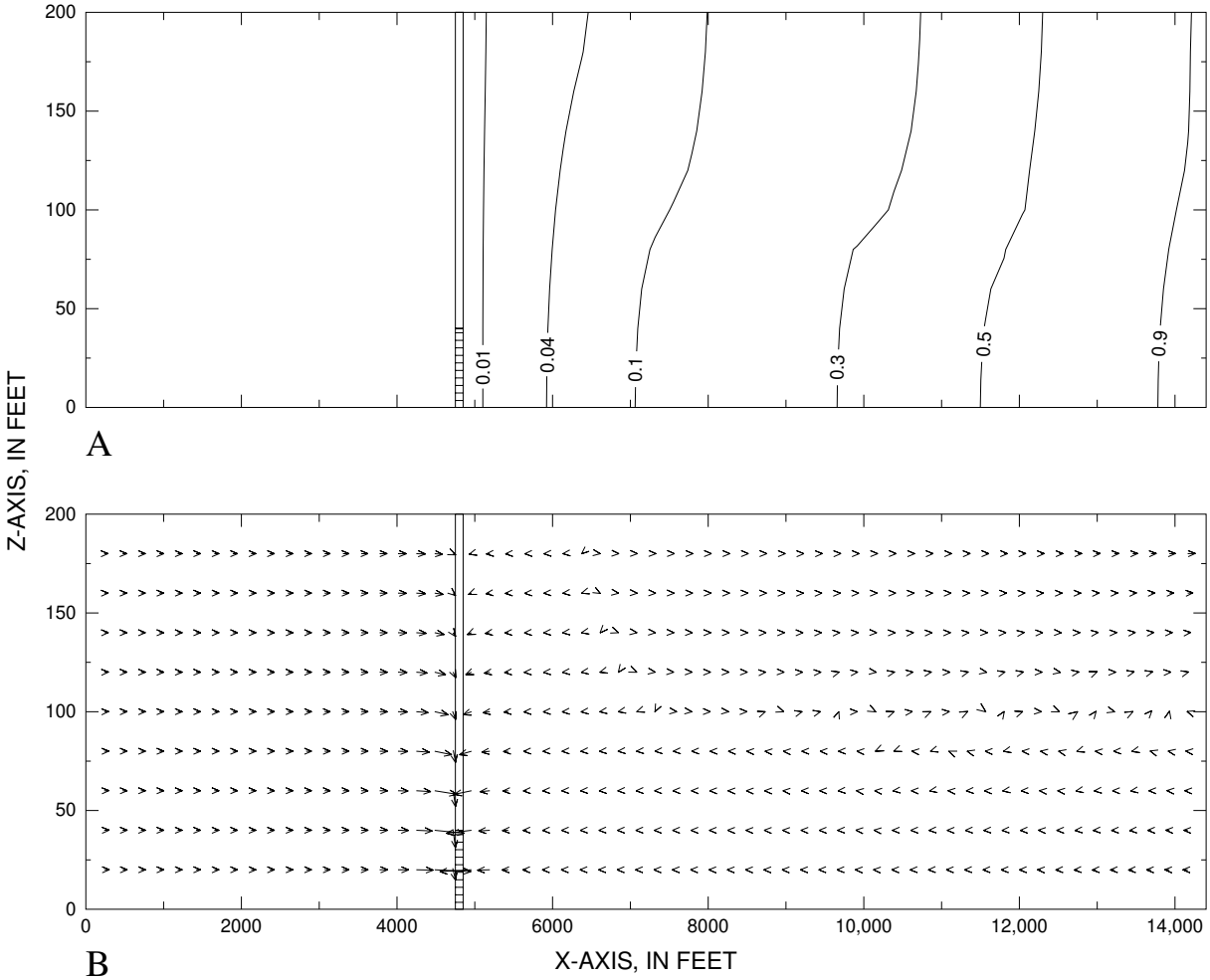


Figure 8.16. Cross-sectional view at y = 2400 ft of (A) contours of scaled salt concentration and (B) velocity vectors for the Huyakorn problem from HST3D.

The invasion of the seawater toward the pumping wells is apparent. There is good visual agreement between the contour plots of the scaled concentration field from the two simulators. Differences can be attributed to the use of the finite-difference method and a uniform mesh in the horizontal plane for HST3D and the use of the finite-element method and a slightly non-uniform mesh in the horizontal plane by Huyakorn and others (1987).

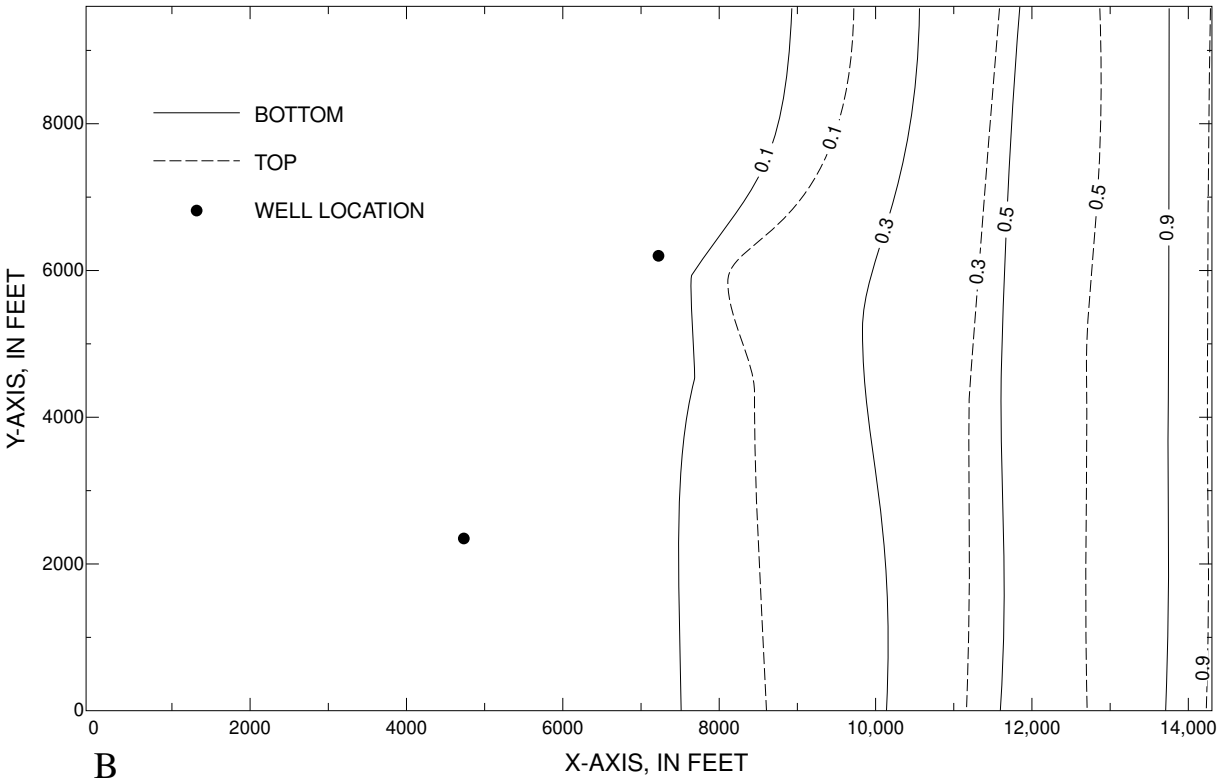


Figure 8.17. Areal view of contours of scaled salt concentration at the bottom and top of the aquifer for the Huyakorn problem from Huyakorn and others (1987).

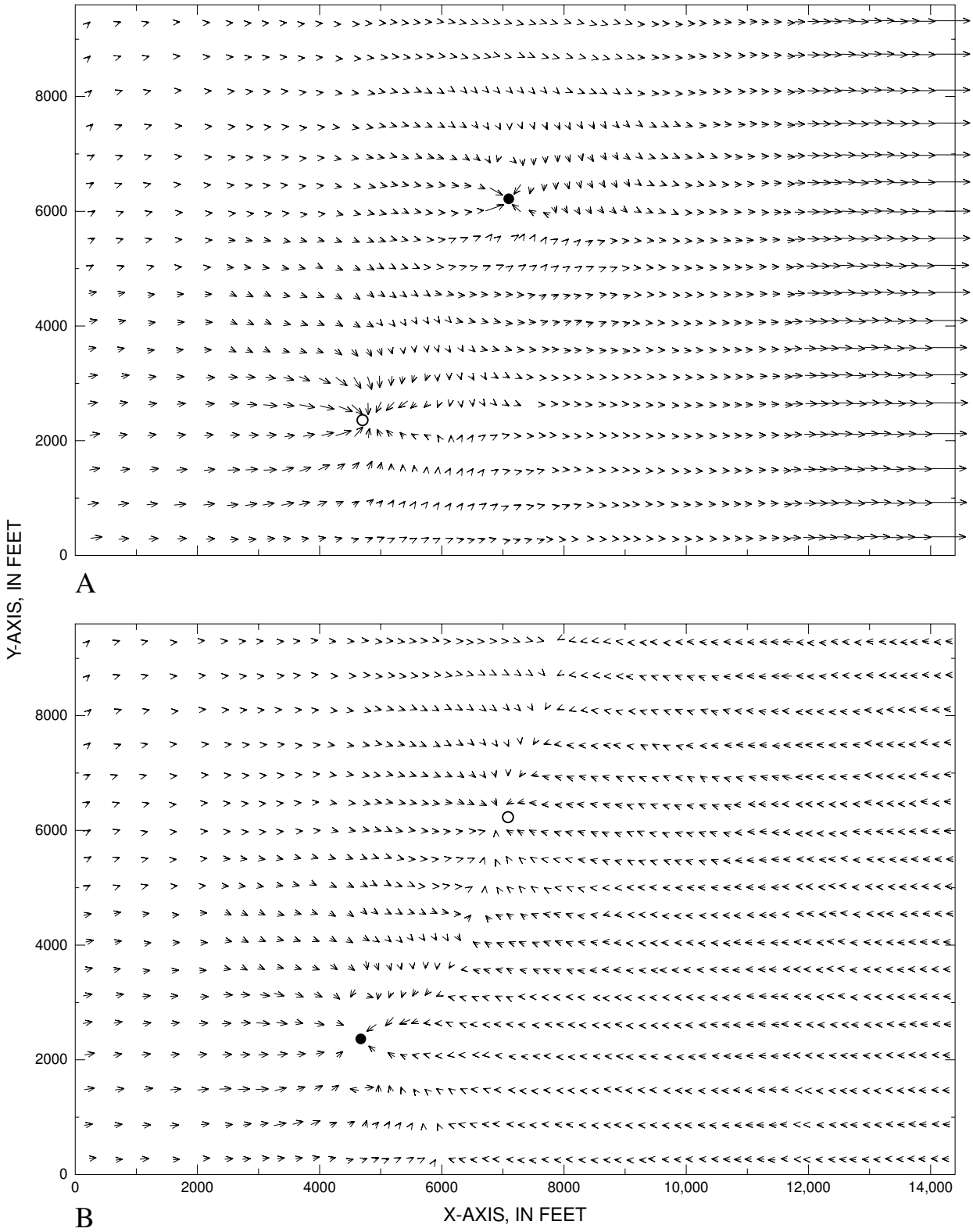


Figure 8.18. Areal view of the horizontal components of the velocity vectors (A) at the top and (B) at the bottom of the aquifer for the Huyakorn problem from Huyakorn and others (1987).

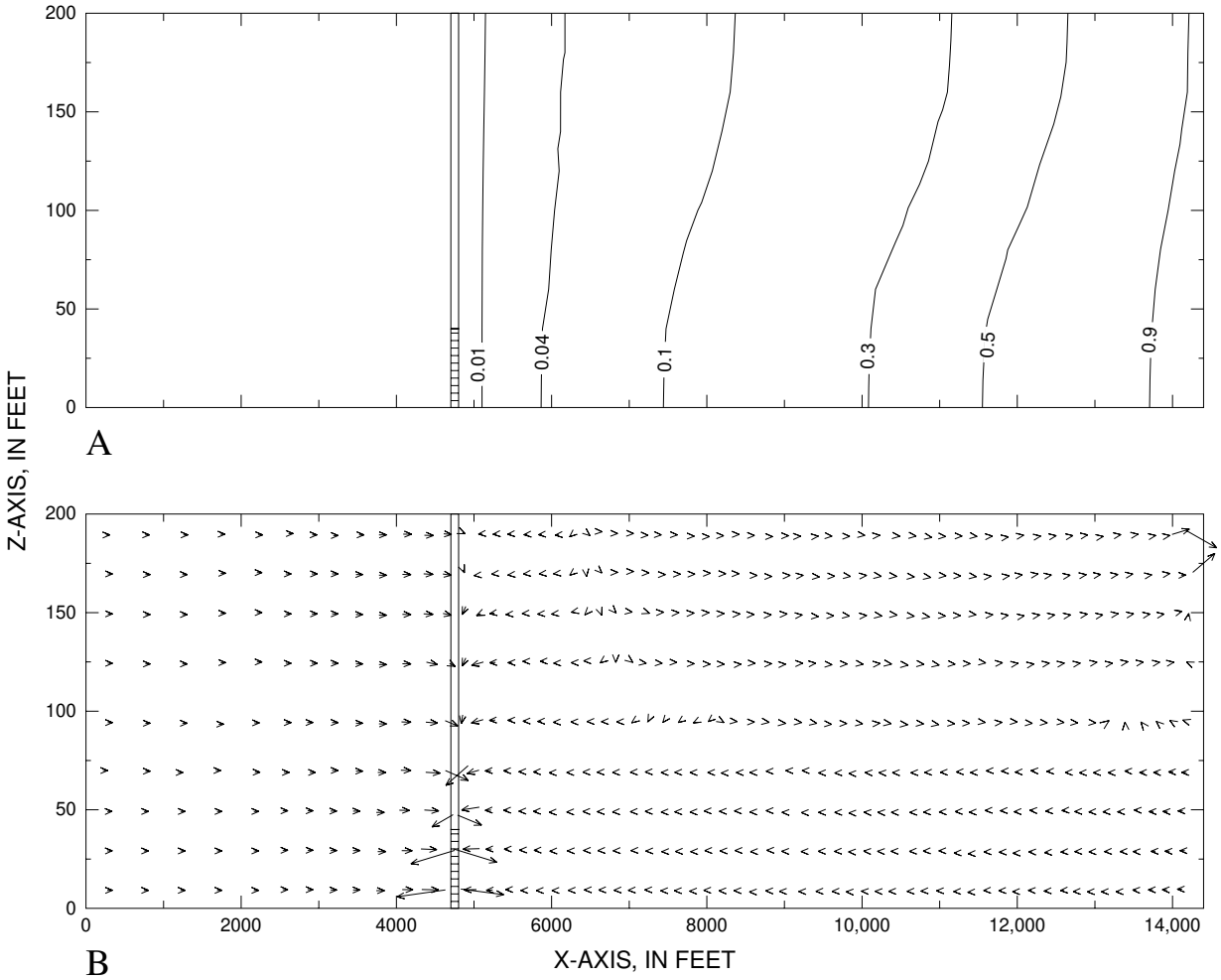


Figure 8.19. Cross-sectional view at y = 2400 ft of (A) contours of scaled salt concentration and (B) velocity vectors for the Huyakorn problem from Huyakorn and others (1987).

8.2 AN EXAMPLE PROBLEM

The verification test problem described above for saltwater intrusion into a coastal aquifer from Huyakorn and others (1987) forms an example for the purpose of giving the new HST3D program user some acquaintance with the data file for a field-scale, three-dimensional simulation. In Table 8.1 an example of the data-input file is presented. Several ways are available to input some of the data so an exact match with the data file is not necessary to execute this example correctly. Reference should be made to chapter 5 to obtain the definitions of the various parameters. Annotations are made in bold within the data file shown Table 8.1. They follow the data descriptor lines but precede the actual data lines to which they refer.

Table 8.1. Data file for example problem with annotations

```
C.....HST Data-Input Form
C.....Version 2.0
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 choice must be made.
C...          i.e. A or B or C
C...   (O) - Optional data with conditions for requirement
C...   P [n.n.n] - The record where parameter P is set
C.....Input by x,y,z range format is;
C.0.1.. X1,X2,Y1,Y2,Z1,Z2
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.. END OR end
C...   {nnn} - Indicates that the default number, nnn, is used if a zero
C...           is entered for that variable
C...   [T/F] - Indicates a logical variable
C...   [I] - Indicates an integer variable
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Start of the data file
C.....Specification and dimensioning data - READ1
C.1.1 .. TITLE Line 1
     Test problem from Huyakorn et al., WRR 23(2), p293, 1987
C.1.2 .. TITLE Line 2
  a three-dimensional salt water intrusion problem, unconfined
C.1.3 .. RESTRT[T/F],TIMRST
```

Any record may be terminated by "/" when there are no data to be entered or changed for the rest of the line.

```
f /
C.....If RESTART, skip to READ3 group
C.1.4 .. HEAT[T/F],SOLUTE[T/F],EEUNIT[T/F],CYLIND[T/F],SCALMF[T/F]
f t t f t
C.1.5 .. TMUNIT[I]
```

The time units are selected here. The selected time unit must be used for all the data expressed with units of time, unless the conversion factors in the program are modified.

```
4
C.1.6 .. NX,NY,NZ,NHCN
62 19 11 0
C.1.7 .. NSBC,NFBC,NLBC,NETBC,NAIFC,NHCBC,NWEL
```

The program will count the number of nodes with each boundary condition type selected here using a non-zero integer.

```
1 1 4*0 2
C.1.8 .. SLMETH[I]
3
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Static data - READ2
C.....Coordinate geometry information
C.....   Rectangular coordinates
C.2.2A.1 .. UNIGRX,UNIGRY,UNIGRZ; all [T/F];(O) - NOT CYLIND [1.4]
3*t
C.2.2A.2A .. X(1),X(NX);(O) - UNIGRX [2.2A.1]
0. 14400.
C.2.2A.2B .. X(I),I=1 to NX;(O) - NOT UNIGRX [2.2A.1]
C.2.2A.3A .. Y(1),Y(NY);(O) - UNIGRY [2.2A.1]
0. 9600.
```

Table.8.1. Data file for example problem with annotations — Continued

```
C.2.2A.3B .. Y(J),J=1 to NY;(O) - NOT UNIGRY [2.2A.1]
C.2.2A.4A .. Z(1),Z(NZ);(O) - UNIGRZ [2.2A.1]
0. 200.
C.2.2A.4B .. Z(K),K=1 to NZ;(O) - NOT UNIGRZ [2.2A.1]
C.....   Cylindrical coordinates
C.2.2B.1A .. R(1),R(NX),ARGRID[T/F];(O) - CYLIND [1.4]
C.2.2B.1B .. R(I),I=1 to NX;(O) - CYLIND [1.4] and NOT ARGRID [2.2B.1A]
C.2.2B.2 .. UNIGRZ[T/F];(O) - CYLIND [1.4]
C.2.2B.3A .. Z(1),Z(NZ);(O) - UNIGRZ [2.2B.3A],CYLIND [1.4]
C.2.2B.3B .. Z(K),K=1 to NZ;(O) - NOT UNIGRZ [2.2B.3A],CYLIND [1.4]
C.2.3.1 .. TILT[T/F];(O) - NOT CYLIND [1.4]
f
C.2.3.2 .. THETXZ,THETYZ,THETZZ;(O) - TILT [2.3.1] and NOT CYLIND [1.4]
C-------------------------------------------------------------------------------
C.....Fluid property information
C.2.4.1 .. BP
0.
C.2.4.2 .. P0,T0,W0,DENF0
0. 15. 0. 62.4
C.2.4.3 .. W1,DENF1;(O) - SOLUTE [1.4]
```

W0 and W1 must be entered as absolute mass fraction values even if scaled mass fraction is being used for the remaining data entry and output. They determine the variation of density with mass fraction and the scaling transformation between absolute and scaled mass fraction.

```
0.0357 63.96
C.2.4.4 .. VISFAC
-1.0
C-------------------------------------------------------------------------------
C.....Reference condition information
C.2.5.1 .. PAATM
0.
C.2.5.2 .. P0H,T0H
0. 15.
C-------------------------------------------------------------------------------
C.....Fluid thermal property information
C.2.6 .. CPF,KTHF,BT;(O) - HEAT [1.4]
C-------------------------------------------------------------------------------
C.....Solute information
C.2.7 .. DM,DECLAM;(O) - SOLUTE [1.4]
0. 0.
C-------------------------------------------------------------------------------
C.....Porous media zone information
C.2.8.1 .. IPMZ,X1Z(IPMZ),X2Z(IPMZ),Y1Z(IPMZ),Y2Z(IPMZ),Z1Z(IPMZ),Z2Z(IPMZ)
1 0. 14400. 0. 9600. 0. 200.
C.....Use as many 2.9.1 lines as necessary
C.2.8.2 .. End with END
```

The end records may be in either upper or lower case for each record.

```
end
C-------------------------------------------------------------------------------
C.....Porous media property information
C.2.9.1 .. KXX(IPMZ),KYY(IPMZ),KZZ(IPMZ),IPMZ=1 to NPMZ [1.7]
7.754e-9 7.754e-9 7.754e-10
C.2.9.2 .. POROS(IPMZ),IPMZ=1 to NPMZ [1.7]
0.20
C.2.9.3 .. ABPM(IPMZ),IPMZ=1 to NPMZ [1.7]
0.
C-------------------------------------------------------------------------------
```

Table.8.1. Data file for example problem with annotations — Continued

```
C.....Porous media thermal property information
C.2.10.1 .. RCPPM(IPMZ),IPMZ=1 to NPMZ [1.7];(O) - HEAT [1.4]
C.2.10.2 .. KTHXPM(IPMZ),KTHYPM(IPMZ),KTHZPM(IPMZ),IPMZ=1 to NPMZ [1.7];(O) -
C..          HEAT [1.4]
C-------------------------------------------------------------------------------
C.....Porous media solute and thermal dispersion information
C.2.11 .. ALPHL(IPMZ),ALPHT(IPMZ),IPMZ=1 to NPMZ [1.7];(O) - SOLUTE [1.4]
C..          and/or HEAT [1.4]
150. 30.
C-------------------------------------------------------------------------------
C.....Porous media solute property information
C.2.12 .. DBKD(IPMZ),IPMZ=1 to NPMZ [1.7];(O) - SOLUTE [1.4]
0.
C-------------------------------------------------------------------------------
C.....Source-sink well information
C.2.13.1 .. IWEL,XW,YW,ZWB,ZWT,WBOD,WQMETH[I];(O) - NWEL [1.6] >0
1 4800. 2400. 0. 40. 2. 11
```

The following two records contain data entered for each element, not each cell. Note that the data are specified from the lowest to the highest element.

```
C.2.13.2 .. WCF(L);L = 1 to NZ-1 (EXCLUSIVE) by ELEMENT
2*1. 8*0.
C.2.13.3 .. WSF(L);L = 1 to NZ-1 (EXCLUSIVE) by ELEMENT
10*0.
C.2.13.4 .. WRISL,WRID,WRRUF,WRANGL;(O) - NWEL [1.6] >0 and
C..          WRCALC(WQMETH [2.13.1] >30)
C.2.13.5 .. HTCWR,DTHAWR,KTHAWR,KTHWR,TABWR,TATWR;(O) - NWEL [1.6] >0
C..          WRCALC(WQMETH [2.13.1] >30) and HEAT [1.4]
C.....Use as many sets of 2.13.1-5 lines as necessary for each well
```

Here the data for the second well are entered as a group.

```
2 7200. 6240. 160. 200. 2. 11
8*0. 2*1.
10*0.
C.2.13.6 .. End with END
end
C.2.13.7 .. MXITQW{10},TOLDPW{6.E-3},TOLFPW{.001},TOLQW{.001},DAMWRC{2.},
C..          DZMIN{.01},EPSWR{.001};(O) - NWEL [1.6] >0
C..          and WRCALC(WQMETH[2.13.1] >30)
C-------------------------------------------------------------------------------
C.....Boundary condition information
C-------------------------------------------------------------------------------
C.....     Specified value b.c.
C.2.14 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NSBC [1.6] > 0
0. 0. 0. 9600. 0. 200.
101
14400. 14400. 0. 9600. 0. 180.
101
14400. 14400. 0. 9600. 180. 200.
100
end
C-------------------------------------------------------------------------------
C.....     Specified flux b.c.
C.2.15 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NFBC [1.6] > 0
```

The range of the specified flux b.c. does not extend to the ends of the region in the x-direction to avoid interference with the specified pressure b.c.

Table.8.1. Data file for example problem with annotations — Continued

```
200. 14200. 0. 9600. 200. 200.
300200
end
C-----------------------------------------------------------------------------
C.....     Aquifer and river leakage b.c.
C.2.16.1 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NLBC [1.6] >0
C.2.16.2 .. KLBC,BBLBC,ZELBC by x,y,z range {0.1-0.3};(O) - NLBC [1.6] >0
C.....          River leakage b.c.
C.2.16.3 .. XR1,YR1,XR2,YR2,KRBC,BBRBC,ZERBC;(O) - NLBC [1.6] > 0
C.2.16.4 .. End with END
C-----------------------------------------------------------------------------
C.....     Evapotranspiration b.c.
C.2.17.1 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NETBC [1.6] >0
C.2.17.2 .. ZLSETB,BETBC by x,y,z range {0.1-0.3};(O) - NETBC [1.6] >0
C-----------------------------------------------------------------------------
C.....     Aquifer influence functions
C.2.18.1 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          NAIFC [1.6] > 0
C.2.18.2 .. UVAIFC by x,y,z range {0.1-0.3};(O) - NAIFC [1.6] > 0
C.2.18.3 .. IAIF;(O) - NAIFC [1.6] > 0
C.....          Pot  a.i.f.
C.2.18.4A .. ABOAR,POROAR,VOAR;(O) - IAIF [2.18.3] = 1
C.....          Transient, Carter-Tracy a.i.f.
C.2.18.4B .. KOAR,ABOAR,VISOAR,POROAR,BOAR,RIOAR,ANGOAR;(O) - IAIF [2.18.3] = 2
C-----------------------------------------------------------------------------
C.....     Heat conduction b.c.
C.2.19.1 .. ZHCBC(K);(O) - HEAT [1.4] and NHCBC [1.6] > 0
C.2.19.2 .. IBC by x,y,z range {0.1-0.3} with no IMOD parameter;(O) -
C..          HEAT [1.4] and NHCBC [1.6] > 0
C.2.19.3 .. DTHHC by x,y,z range {0.1-0.3} for HCBC nodes;(O) -
C..          HEAT [1.4] and NHCBC [1.6] > 0
C.2.19.4 .. KHCBC by x,y,z range {0.1-0.3} for HCBC nodes;(O) -
C..          HEAT [1.4] and NHCBC [1.6] > 0
C-----------------------------------------------------------------------------
C.....Free surface b.c.
C.2.20 .. FRESUR[T/F]
t
C-----------------------------------------------------------------------------
C.....Initial condition information
C.2.21.1 .. ICHYDP[T/F]
t
C.2.21.2 .. ICHWT[T/F];(O) - FRESUR [2.20]
f
C.2.21.3A .. ZPINIT,PINIT;(O) - ICHYDP [2.21.1] and NOT ICHWT [2.21.2]
200. 0.
C.2.21.3B .. P by x,y,z range {0.1-0.3};(O) - NOT ICHYDP [2.21.1] and
C..          NOT ICHWT [2.21.2]
C.2.21.3C .. HWT by x,y,z range {0.1-0.3};(O) - FRESUR [2.20] and
C..          ICHWT [2.21.2]
C.2.21.4 .. T by x,y,z range {0.1-0.3};(O) - HEAT [1.4]
C.2.21.5 .. NZTPHC, ZTHC(I),TVZHC(I);(O) - HEAT [1.4] and NHCBC [1.6] >0,
C..          limit of 5
C.2.21.6 .. C by x,y,z range {0.1-0.3};(O) - SOLUTE [1.4]
0. 14400. 0. 9600. 0. 200.
```

Table.8.1. Data file for example problem with annotations — Continued

```
0. 1
end
C-----------------------------------------------------------------------------
C.....Calculation information
C.2.22.1 .. FDSMTH,FDTMTH
0. 1.
C.2.22.2 .. TOLDEN{.001},MAXITN{5}
0. 0
C.....     Two-line s.o.r. solver
C.2.22.3 .. NTSOPT{5},EPSSLV{1.e-7},EPSOMG{.2},MAXIT1{50},MAXIT2{100};
C..          (O) - SLMETH [1.8] = 2
C.....     Generalized, restarted conjugate-gradient solver
C.2.22.4 .. IDIR,IORDER,NSDR,EPSSLV{1.e-5},MAXIT2{100};
C..          (O) - SLMETH [1.8] = 3 or 5
```

The user should experiment with the choice of IDIR and IORDER, when SLMETH is 3 to maximize the convergence rate. The convergence tolerance, EPSSLV, should be varied to ensure that an accurate solution is being obtained. The solution should not change significantly with a decrease in EPSSLV of one or two orders of magnitude.

```
3 2 5 1.e-6 1000
C-----------------------------------------------------------------------------
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];(O) - PRTIC [2.23.1]
101 f
C.2.23.3 .. ORENPR[I];(O) - NOT CYLIND [1.4]
12
C.2.23.4 .. PLTZON[T/F];(O) - PRTPMP [2.23.1]
f
C.2.23.5 .. PLTTEM[T/F]
f
C-----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
C..... TRANSIENT DATA - READ3
C.3.1 .. THRU[T/F]
```

This is always false for the first set of transient data.

```
f
C.....If THRU is true, proceed to record 3.99
C-----------------------------------------------------------------------------
C.....The following is for NOT THRU
C-----------------------------------------------------------------------------
C.....Source-sink well information
C.3.2.1 .. RDWTD[T/F];(O) - NWEL [1.6] > 0
t
C.3.2.2 .. IWEL,QWV,PWSUR,PWKT,TWSRKT,CWKT;(O) - RDWTD [3.2.1]
```

Note the sign convention on the well flow rate.

```
1 -1.e6 0. 0. 0. 0.
2 -0.5e6 0. 0. 0. 0.
C.....Use as many 3.2.2 lines as necessary
C.3.2.3 .. End with END
end
C-----------------------------------------------------------------------------
C.....Boundary condition information
```

Boundary condition data may be entered as rectangular zones of nodes that contain nodes other than those defined to have a given boundary condition type. Only the nodes within the zone that are identified to have the given boundary condition type will be loaded with the data values.

```
C-----------------------------------------------------------------------------
```

Table.8.1. Data file for example problem with annotations — Continued

```
C.....     Specified value b.c.
C.3.3.1 .. RDSPBC,RDSTBC,RDSCBC; all [T/F];(O) - NSBC [1.6] > 0
t f t
C.3.3.2 .. PNP B.C. by x,y,z range {0.1-0.3};(O) - RDSPBC [3.3.1]
```

Here is a relatively simple, but not unique, method for establishing pressure distributions over a boundary surface. First, a linearly varying head is specified over the entire boundary face. The data are entered using the linear interpolation option in the z-direction. Next, the hydrostatic pressure head, in feet, is added to each node layer in turn below the uppermost layer. Note that the heads are not cumulative because the z-range is set one elevation at a time. Finally, a multiplication step converts the heads to pressure values.

```
c..saltwater boundary
14400. 14400. 0. 9600. 0. 200.
```

Linearly interpolate in the z-direction.

```
3. 5
```

Head at top and bottom elevation.

```
0. 0. 200. 0.
14400. 14400. 0. 9600. 180. 180.
```

Add 20 feet of head to head at top of boundary.

```
20. 3
14400. 14400. 0. 9600. 160. 160.
```

Add 40 feet of head to head at top of boundary.

```
40. 3
14400. 14400. 0. 9600. 140. 140.
```

Continue...

```
60. 3
14400. 14400. 0. 9600. 120. 120.
80. 3
14400. 14400. 0. 9600. 100. 100.
100. 3
14400. 14400. 0. 9600. 80. 80.
120. 3
14400. 14400. 0. 9600. 60. 60.
140. 3
14400. 14400. 0. 9600. 40. 40.
160. 3
14400. 14400. 0. 9600. 20. 20.
180. 3
14400. 14400. 0. 9600. 0. 0.
200. 3
14400. 14400. 0. 9600. 0. 200.
```

Multiply to convert from saltwater head to pressure in psi.

```
0.4443 2
c..freshwater boundary
```

At this boundary, the same procedure is used with a linear pressure variation in the y-direction.

```
0. 0. 0. 9600. 0. 200.
2. 5
0. 5. 9600. 4.
0. 0. 0. 9600. 180. 180.
20. 3
0. 0. 0. 9600. 160. 160.
40. 3
0. 0. 0. 9600. 140. 140.
60. 3
0. 0. 0. 9600. 120. 120.
80. 3
0. 0. 0. 9600. 100. 100.
100. 3
```

Table.8.1. Data file for example problem with annotations — Continued

```
0. 0. 0. 9600. 80. 80.
120. 3
0. 0. 0. 9600. 60. 60.
140. 3
0. 0. 0. 9600. 40. 40.
160. 3
0. 0. 0. 9600. 20. 20.
180. 3
0. 0. 0. 9600. 0. 0.
200. 3
0. 0. 0. 9600. 0. 200.
```

Multiply to convert from freshwater head to pressure in psi.

```
0.4335 2
end
C.3.3.3 .. TSBC by x,y,z range {0.1-0.3};(O) - RDSPBC [3.3.1] and
C..          HEAT [1.4]
C.3.3.4 .. CSBC by x,y,z range {0.1-0.3}; (O) - RDSPBC [3.3.1] and
C..          SOLUTE [1.4]
```

This data set is used to define the associated solute mass fraction for advected fluid that enters the simulation region. Fluid that leaves the region has the ambient concentration of the respective boundary cells. Only cells that do not have a specified value mass fraction b.c. need to be included here.

```
14400. 14400. 0. 9600. 180. 200.
1. 1
end
C.3.3.5 .. TNP B.C. by x,y,z range {0.1-0.3};(O) - RDSTBC [3.3.1] and
C..          HEAT [1.4]
C.3.3.6 .. CNP B.C. by x,y,z range {0.1-0.3};(O) - RDSCBC [3.3.1] and
C..          SOLUTE [1.4]
```

This data set is used to define the solute mass fraction for specified value mass fraction b.c. cells. Cells that have only an associated concentration for inflow should not be included here.

```
0. 0. 0. 9600. 0. 200.
0. 1
```

Note the overlap of the ending node at 180 ft with the starting node of the C.3.3.4 record. This latter specification will be the value applied since it will be the data most recently read. Here it makes no difference because the values are the same for both data sets.

```
14400. 14400. 0. 9600. 0. 180.
1. 1
end
C-----------------------------------------------------------------------------
C.....     Specified flux b.c.
C.3.4.1 .. RDFLXQ,RDFLXH,RDFLXS; all [T/F];(O) - NFBC [1.6] > 0
t f f
C.3.4.2 .. QFFX,QFFY,QFFZ B.C. by x,y,z range {0.1-0.3};(O) -
C..          RDFLXQ [3.4.1]
200. 14200. 0. 9600. 200. 200.
```

The data are entered in a set of three pairs of value and modification code, unless node-by-node input is selected. Flux values are input as vector components in the regional coordinate system. The z-component of the flux points downward. Here, the flux is defined over the x-direction range as one cell in from the left and right boundaries so there is no conflict with the specified value boundary conditions.

```
0. 1 0. 1 -2.7397e-3 1
end
C.3.4.3 .. UDENBC  by x,y,z range {0.1-0.3};(O) - RDFLXQ [3.4.1]
```

The following records for associated density and mass fraction apply to flux entering the simulation region.

```
200. 14200. 0. 9600. 200. 200.
62.4 1
end
```

Table.8.1. Data file for example problem with annotations — Continued

```
C.3.4.4 .. TFLX B.C. by x,y,z range {0.1-0.3};(O) - RDFLXQ [3.4.1] and
C..          HEAT [1.4]
C.3.4.5 .. CFLX B.C. by x,y,z range {0.1-0.3};(O) - RDFLXQ [3.4.1] and
C..          SOLUTE [1.4]
```

The solute concentration value needs to be compatible with the relation given previously for density as a function of solute concentration.

```
200. 14200. 0. 9600. 200. 200.
0. 1
end
C.3.4.6 .. QHFX,QHFY,QHFZ B.C. by x,y,z range {0.1-0.3};(O) - RDFLXH [3.4.5]
C.3.4.7 .. QSFX,QSFY,QSFZ B.C. by x,y,z range {0.1-0.3};(O) - RDFLXS [3.4.1]
C-------------------------------------------------------------------------
C.....     Leakage b.c.
C.3.5.1 .. RDLBC[T/F];(O) - NLBC [1.6] > 0
C.3.5.2 .. PHILBC,DENLBC,VISLBC by x,y,z range {0.1-0.3};(O) - RDLBC [3.5.1]
C.3.5.3 .. TLBC by x,y,z range {0.1-0.3};(O) - RDLBC [3.5.1] and HEAT [1.4]
C.3.5.4 .. CLBC by x,y,z range {0.1-0.3};(O) - RDLBC [3.5.1] and SOLUTE [1.4]
C.....          River leakage b.c.
C.3.5.5 .. XR1,YR1,XR2,YR2,HRBC,DENRBC,VISRBC,TRBC,CRBC;(O) - RDLBC [3.5.1]
C.....Use as many 3.5.5 lines as necessary
C.3.5.6 .. End with END
C-------------------------------------------------------------------------
C.....     Evapotranspiration b.c.
C.3.6.1 .. RDETBC[T/F];(O) - NETBC [1.6] > 0
C.3.6.2 .. QETBC by x,y,z range {0.1-0.3};(O) - RDETBC [3.5.1]
C-------------------------------------------------------------------------
C.....     Aquifer influence function b.c.
C.3.7.1 .. RDAIF[T/F];(O) - NAIFC [1.6] > 0
C.3.7.2 .. DENOAR by x,y,z range {0.1-0.3};(O) - RDAIF [3.7.1]
C.3.7.3 .. TAIF by x,y,z range {0.1-0.3};(O) - RDAIF [3.7.1] and HEAT [1.4]
C.3.7.4 .. CAIF by x,y,z range {0.1-0.3};(O) - RDAIF [3.7.1] and SOLUTE [1.4]
C-------------------------------------------------------------------------
C.....Calculation information
C.3.8.1 .. RDCALC[T/F]
t
C.3.8.2 .. AUTOTS[T/F];(O) - RDCALC [3.8.1]
t
C.3.8.3A .. DELTIM;(O) - RDCALC [3.8.1] and NOT AUTOTS [3.8.2]
C.3.8.3B .. DPTAS{5E4},DTTAS{5.},DCTAS{.25},DTIMMN{1.E4},DTIMMX{1.E7};
C..           (O) - RDCALC [3.8.1] and AUTOTS [3.8.2]
0.5 0. 0.05 1.e-2 10000.
C.3.8.4 .. TIMCHG
25000.
C-------------------------------------------------------------------------
C.....Output information
C.3.9.1 .. PRISLM,PRIKD,PRIPTC,PRIDV,PRIVEL,PRIGFB,PRIBCF,PRIWEL; all [I]
```

Output intervals are specified as a number of time steps (positive number) or an interval of simulation time (negative number).

```
1 0 -5000 -25000 -25000 2 -25000 -5000
C.3.9.2 .. IPRPTC;(O) - IF PRIPTC [3.9.1] NOT = 0
201
C.3.9.3 .. CHKPTD[T/F],PRICPD,SAVLDO[T/F]
```

Since no checkpoint dump is being requested, the final two data items do not need to be specified.

```
f /
C-------------------------------------------------------------------------
C.....Contour and vector map information
```

Table.8.1. Data file for example problem with annotations — Continued

```
C.3.10.1 .. CNTMAP[T/F],VECMAP[T/F],PRIMAP[I]
```

The interval of contour-map and vector-map data output is specified as 10000 days. Specifying a value greater than the time of simulation will produce output only at the end of the simulation.

```
t t -10000
C-----------------------------------------------------------------------------
C.....End of first set of transient information
```

There is only one set of transient boundary and calculation parameter data for this example because all boundary conditions are constant in time.

```
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.....Read sets of READ3 data at each TIMCHG until THRU  (Lines 3.N1.N2)
C-----------------------------------------------------------------------------
C.....End of simulation line follows, THRU=.TRUE.
C.3.99.1 .. THRU
T
C.....End of the data file
C-----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
```

The input of the transient boundary-condition data at the inland and coastal boundaries of the region is an example of how to establish a specified-pressure boundary condition for a case where the equivalent reference head varies linearly with depth or lateral location. It can be done in three steps using the input by x,y,z range zones. At the inland boundary, the equivalent reference head varies with y-location along the boundary. The first step is to install the linearly varying reference head from 0 to 4,600 ft in the y-direction. The second step is to subtract the elevation head from the reference head at each layer of nodes in the z-direction. This gives the pressure head distribution. The third step is to convert the pressure heads to pressures by multiplying by the density-gravitational acceleration term and a unit conversion factor. The unit conversion factor is necessary because this problem is in the U.S. customary unit system, so pressure is in pounds per square inch. The same three steps are used to establish the boundary-pressure distribution at the coastal boundary.

There are six output files produced for this example. However, because of the large sizes of these files, they will not be included in printed form in this report. All output files from this example are included in the program distribution package available as described in Appendix C. Reviewing these input and output files and preparing your own version is a good way to gain familiarity with preparing input and running the HST3D simulator. Using these files, one can verify that the data file for the example problem has been prepared correctly and that the simulator is performing correctly as installed.

8.2.1 Use of BCFLOW: the post-processing program for flow rates and cumulative amounts

An example of the use of the post-processing program, BCFLOW, that calculates flow rates and cumulative amounts is given in Appendix B, section B.6.1. The data-input file is used to define three zones of boundary-facial areas over which flow rates and cumulative flow amounts are desired at the end of the simulation. Nodal flow-rate and cumulative-amount data are obtained from the file, `Bcnfr.example`. Parameter PRIBCF (record 3.9.1) of HST3D was used to write the `Bcnfr.example` file with data only from the end of the simulation period.

The output file in the distribution is named `O.bcfz.example`. The defined boundary zones are written followed by the various flow rates and cumulative amounts. U.S. customary units are used because the original simulation was defined with these units. The coastal boundary zones separate the flow rates and cumulative amounts into saltwater inflow (or intrusion) rates and amounts and brackish-water outflow rates and amounts.

9. NOTATION

9.1 ROMAN CHARACTERS

$\mathbf{A}$	matrix of finite difference equations (appropriate units)
b_{ET}	extinction depth below land surface (m)
b_{Lm}	thickness of the leakage layer (m)
C_{ij}	fluid storage factors for a cell (appropriate units)
C_{fp}	fluid storage factors for the adjacent cell below a cell (appropriate units)
C_{sp}	solute storage factors for the adjacent cell below a cell (appropriate units)
$\mathbf{D}_R$	diagonal matrix of **A**, red nodes
$\mathbf{D}_B$	diagonal matrix of **A**, black nodes
d	data item
E_{fp}	fluid storage factors for the adjacent cell above a cell (appropriate units)
E_{sp}	solute storage factors for the adjacent cell above a cell (appropriate units)
f_m	fraction of cell saturation for boundary cell m (–)
f^n	fraction of cell saturation at time t^n (–)
g	gravitational constant (m/s^2)
H_B	specific enthalpy of the fluid at the region boundary (J/kg)
K_d	equilibrium distribution coefficient (m^3/kg)
k_{Lm}	permeability of the leakage layer (m^2)
$\mathbf{L}$	lower triangular matrix of **A**
$\mathbf{M}$	approximate inverse of **A**
n	repeat count integer
P'	pressure (bar)
P_{sat}	saturation pressure (kPa)
P'_{sat}	saturation pressure (bar)
p_m	pressure at the simulation-region boundary at cell m (Pa)
Q_{ETm}	volumetric flow rate at an evapotranspiration boundary (m^3/s)
Q_{Lm}	volumetric flow rate at a leakage boundary (m^3/s)
Q_{Rm}	volumetric flow rate at a river-leakage boundary (m^3/s)
q_{ET}	fluid flux across the evapotranspiration boundary (m^3/m^2–s)
q_{HET}	heat flux across the evapotranspiration boundary (W/m^2)
q_{SET}	solute flux across the evapotranspiration boundary (kg/m^2–s)
q_{max}	maximum evapotranspiration flux (m^3/m^2–s)
$\mathbf{R}$	reduced matrix of **A**, black nodes
S^6	free-surface location that varies in space and time (m^2)
S^7	region boundary surface over which an evapotranspiration (m^2) boundary condition exists
S_{BETm}	evapotranspiration-boundary boundary cell surface (m^2)
S_{BLm}	leakage-boundary boundary cell surface (m^2)
T	temperature (°F)
T'	temperature (K)
T''	temperature (°C)

$T_{½}$	half life of solute species (s)
U	upper triangular matrix of **A**
u	pressure, temperature, or mass fraction
V_s	volume of cell subdomain s (m^3)
w_B	solute mass fraction at the region boundary (-)
w_p	weight percent of sodium chloride (–)
z_{EX}	elevation of the extinction depth (m)
z_{FS}	elevation of the water-table free surface (m)
z_{LS}	elevation of the land surface (m)
z_e	elevation at the top of the leakage layer (m)
z_m	elevation of the simulation-region boundary at cell m (m)

9.2 GREEK CHARACTERS

β_w	coefficient of density variation with solute mass fraction (–)
ε_s	porosity for cell subdomain s (–)
η	attenuation parameter (–)
μ_0	fluid viscosity of pure water (Pa–s)
μ_{Lm}	viscosity of the fluid in the leakage layer (Pa–s)
ρ_B	density of the fluid at the evapotranspiration boundary (kg/m^3)
ρ_b	bulk density of the porous medium (kg/m^3)
ρ_e	density of the fluid density in the outer aquifer (kg/m^3)
ρ_m	fluid density at the simulation-region boundary at cell m (kg/m^3)
ϕ_e	potential energy per unit mass of the fluid in the outer aquifer (N–m/kg)

9.3 MATHEMATICAL OPERATORS AND SPECIAL FUNCTIONS

δt	new time step (s)
δt_0	previous time step (s)
δu^s_{max}	specified maximum change in u (appropriate units)
$\lvert\delta u_{max}\rvert$	absolute value of the maximum-calculated change in u over the previous time step (appropriate units)

10. REFERENCES CITED

American National Standards Institute, 1978, ANSI X3.9-1978 programming language FORTRAN: New York, 85 p.

Axelsson, Owe, 1994, Iterative solution methods: New York, Cambridge University Press, 654 p.

Aziz, Khalid and Settari, Antonin, 1979, Petroleum reservoir simulation: Barking, Essex, England, Applied Science, 476 p.

Behie, G.A. and Forsyth, P.A., 1984, Incomplete factorization methods for fully implicit simulation of enhanced oil recovery: SIAM Journal of Scientific and Statistical Computation, v. 5, p. 543-560.

Bulirsch, R. and Stoer, J., 1966, Numerical treatment of ordinary differential equations by extrapolation methods: Numerische Mathematik, v. 8, no. 1, p. 1-13.

Elder, J.W., 1967, Transient convection in a porous medium: Journal of Fluid Mechanics, v. 27, p. 609-623.

Elman, H.C., l982, Iterative methods for large sparse nonsymmetric systems of linear equations: Ph.D. dissertation, Department of Computer Science, Yale University, New Haven, CT.

Henry, H.R., 1964, Effects of dispersion on salt encroachment in coastal aquifers, *in* Sea water in coastal aquifers: U.S. Geological Survey Water-Supply Paper 1613-C, p. C70-C84.

Hewlett-Packard, 1985, HP-41C manual, Petroleum fluids pac: Corvallis, OR, 136 p.

Huyakorn, P.S., Andersen, P.F., Mercer, J.W., and White Jr., H.O., 1987, Saltwater intrusion in aquifers: Development and testing of a three-dimensional finite element model: Water Resources Research, v. 23, no. 2, p. 293-312.

Kipp, K.L., Russell, T.F., and Otto, J.S., 1992, D4Z -- A new renumbering for iterative solution of ground-water flow and solute-transport equations, *in* Russell, T.F., Ewing, R.E., Brebbia, C.A., Gray, W.G., and Pinder, G.F., eds., Computational methods in water resources IX, v. 1, Numerical methods in water resources: Boston, Computational Mechanics Publications and London, Elsevier Applied Science, p. 495-502.

Kipp, K.L., Russell, T.F., and Otto, J.S., 1994, Three-dimensional D4Z renumbering for iterative solution of ground-water flow and transport equations, *in* Peters, A., Wittum, G., Herrling, B., Meissner, U.,Brebbia, C.A., Gray, W.G., and Pinder, G.F., eds., Computational methods in water resources X, v. 2: Dordrecht, The Netherlands, Kluwer Academic Publishers, p. 1417-1424.

Kipp Jr., K.L., 1987, HST3D: A computer code for simulation of heat and solute transport in three-dimensional ground-water flow systems: U.S. Geological Survey Water-Resources Investigations Report 86-4095, 597 p.

Meijerink, J.A. and van der Vorst, H.A., 1977, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix: Mathematics of Computation, v. 31, no. 137, p. 148-162.

Meijerink, J.A. and van der Vorst, H.A., 1981, Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems: Journal of Computational Physics, v. 44, p. 134-155.

Organization for Economic Cooperation and Development (OECD), 1988, The international HYDROCOIN project, level 1: code verification, Report 71617:, Paris, 198 p.

Patankar, S.V., 1980, Numerical heat transfer and fluid flow: New York, Hemisphere Publishing Corporation, 197 p.

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: Illinois State Water Survey, ISWS-71-BUL-55, Urbana,IL, Department of Registration and Education, Illinois State Water Survey, 62 p.

Ségol, Geneviève, 1994, Classic groundwater simulations: Englewood Cliffs, New Jersey, PTR Prentice Hall, 531 p.

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

Voss, C.I., 1984, SUTRA: 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.

Voss, C.I. and Souza, W.R., 1987, Variable density flow and solute transport simulation of regional aquifers containing a narrow freshwater-saltwater transition zone: Water Resources Research, v. 23, no. 10, p. 1851-1866.

APPENDIX A — DIMEN: A PRE-PROCESSOR FOR DIMENSIONING HST3D

A.1 INTRODUCTION

Version 2 of the HST3D simulator uses named common blocks defined by Fortran COMMON statements with array dimensions set by Fortran PARAMETER statements. The various arrays must be dimensioned large enough to accommodate the particular problem being simulated. The necessary dimensions for some of the arrays are computed by the simulator after parts of the data file are read. In some cases, processing cannot continue to determine required sizes of some of the arrays if the storage for other arrays is not sufficiently large. Thus, it may take as many as three recompilations of the HST3D source code to establish appropriate array storage for a given problem. The DIMEN program was developed to deal with the problem of establishing the necessary dimensions for recompilation of HST3D, and reduces the number of recompilations to one. This appendix documents Version 2 of DIMEN designed to work with Version 2 of HST3D.

The DIMEN program reads the data file for a given simulation problem and the PARAMETER statements that determined the array dimensions for the current executable file of the HST3D simulator. The program then calculates the necessary sizes for all arrays and checks whether or not sufficient space for each array has been dimensioned in the current executable file. Finally, the program writes out new PARAMETER statements with array-size values modified as necessary. These new PARAMETER statements can be used for a recompilation of the HST3D source code making the array sizes sufficiently large for the given problem to be simulated. The DIMEN program does limited error checking of the problem data file.

A.2 COMPUTER CODE DESCRIPTION

A.2.1 Program Structure and Organization

The DIMEN computer code is written in FORTRAN-77. Code conforming to ANSI standards (American National Standards Institute, 1978) has been used as closely as possible to provide maximum portability. An exception to these standards is the usage of INCLUDE statements. Version 2.0 of the program code consists of a main routine and 30 subroutines. There are 55 common blocks and PARAMETER statements incorporated into the source code by INCLUDE statements. The program length is approximately 4,000 lines of code including comment and continuation lines. The following is a list of the routines and a brief description of their function:

DIMEN — Main routine that drives the program execution. The basic steps are (1) open files, read, error check, initialize, and write output for simulation-problem specifications and space allocation; (2) read, error check, initialize, and write output for time-invariant information; (3) read transient information; (4) check and modify parameter values as necessary; (5) write the new PARAMETER statements; and (6) close files and terminate program execution.

The following subroutines are described in their order of execution:

OPENF — Requests the user to enter the data-input file name and an identification suffix for output from the pre-processing run.

READPF — Reads the HST3D source files containing the PARAMETER statements and interprets the parameter values.

READ1 Reads the first section of the data file.

INIT1 Initializes the total lengths of the common blocks. If necessary, sets the U.S. customary-to-metric conversion factors, and defines unit labels.

ERROR1 Checks for errors in the data read by READ1.

WRITE1 Writes information about problem definition and the first set of array-size requirements.

READ2 Reads the data pertaining to the time-invariant information, including mesh specifications, boundary-condition arrangement, and parameters for the equation-solution method.

ERROR2 Checks for errors in the data read by READ2.

INIT2 Initializes the geometric data by calculating distribution and numbers of boundary-condition nodes of each type and computes the array requirements for each type of equation solver available.

ERROR4 Checks for errors in the parameters calculated by INIT2.

WRITE2 Writes the array dimension requirements, the boundary-condition locations, and the boundary-condition index array to the output file.

CHGPAR Changes the values of the dimensioning parameters as necessary for the given data set.

WRITPF Writes the new PARAMETER statements for the HST3D source files containing the parameter values modified as necessary.

READ3 Reads the transient data, including boundary condition, source-sink, time-step, calculation, and printout information.

CLOSEF Closes files, deletes scratch files, and prints any error messages.

The following subroutines are listed in alphabetical order. Some are called from several routines.

D4ORDR Determines the node numbering and the dimension requirements for the alternating-diagonal (D4) reordering scheme to form the reduced matrix.

ERRPRT Writes the error messages to the output file.

ETOM1 Converts time-invariant data from U.S. customary units to metric units.

IREWI Reads, error checks, echo writes, and initializes integer-array data elements that are input as zones of constant values over a rectangular prism of cells, or are input as node-by-node distributions.

REWI Reads, error checks, echo writes, and initializes real-array data 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.

The following subroutines are low-level utility routines listed in alphabetical order. They are called many times from various locations.

CTONB Copies spaces and record separators from one character string to another up to the next non-blank character.

EXVAL Extracts the parameter value from the PARAMETER statement and associates it with the parameter.

INSVAL Installs the new parameter value into the PARAMETER statement associated with the proper parameter.

INTERP	Performs one-dimensional or two-dimensional linear or bilinear tabular interpolation.
LINCUT	Cuts a Fortran statement into lines for output with the number of continuation lines necessary.
MTOIJK	Calculates the x-, y-, and z-coordinate indices from the natural node number of a cell.
NINTRP	Performs one-dimensional interpolation of a coordinate value to the nearest node point.
PRNTAR	Writes one-dimensional or two-dimensional arrays in tabular form to a specified file.
STONB	Spaces forward or backward along a character string to the next non-blank character.
WRITLN	Writes a line to the output file.

A chart showing the main sequence of subroutine execution, and the linkage between the subroutines appears in Figure A.1. The primary subroutines are on the left and the secondary and utility subroutines are to the right in each column. The sequence of execution is from top to bottom of the left column followed by the right column. The factor in front of some routine names indicates multiple calls. Some utility subroutines are listed more than once for graphical clarity.

```
=====================================
 Calling Tree

    Program DIMEN
    |
    +--Subroutine OPENF
    |
    +--Subroutine READPF
    |  |
    |  +--2*Subroutine STONB
    |  |
    |  +--Subroutine CTONB
    |  |
    |  +--3*Subroutine STONB
    |  |
    |  +--Subroutine CTONB
    |  |
    |  +--Subroutine EXVAL
    |
    +--Subroutine READ1
    |
    +--Subroutine INIT1
    |
    +--Subroutine ERROR1
    |
    +--Subroutine WRITE1
    |
    +--Subroutine READ2
    |  |
    |  +--6*Function NINTRP
    |  |
    |  +--3*Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine REWI3
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--4*Function NINTRP
    |  |
    |  +--Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine REWI3
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine REWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine IREWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--6*Subroutine REWI
    |     |
    |     +--6*Function NINTRP
    |
    +--Subroutine ERROR2
    |
    +--Subroutine INIT2
    |  |
    |  +--Subroutine ETOM1
    |  |
    |  +--4*Function NINTRP
    |  |
    |  +--Subroutine D4ORDR
    |
    +--Subroutine ERROR4
    |
    +--Subroutine WRITE2
    |  |
    |  +--12*Subroutine PRNTAR
    |
    +--Subroutine CHGPAR
    |
    +--Subroutine WRITPF
    |  |
    |  +--2*Subroutine STONB
    |  |
    |  +--2*Subroutine CTONB
    |  |
    |  +--Subroutine INSVAL
    |  |
    |  +--Subroutine LINCUT
    |  |
    |  +--Subroutine WRITLN
    |  |
    |  +--Subroutine LINCUT
    |  |
    |  +--Subroutine WRITLN
    |
    +--Subroutine READ3
    |  |
    |  +--5*Subroutine REWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--Subroutine REWI3
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--3*Subroutine REWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--3*Subroutine REWI3
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--2*Subroutine REWI
    |  |  |
    |  |  +--6*Function NINTRP
    |  |
    |  +--4*Function NINTRP
    |  |
    |  +--4*Subroutine REWI
    |     |
    |     +--6*Function NINTRP
    |
    +--Subroutine CLOSEF
       |
       +--Subroutine ERRPRT
=====================================
```

Figure A.1. Connection chart for the DIMEN main program and subprograms showing routine hierarchy.

A.2.2 Memory Allocation, Array-Size Requirements, and Subprogram Communication

Each of the arrays used by DIMEN is dimensioned by a Fortran PARAMETER statement. The dimension values are set to be larger than needed for any simulation problem that would be feasible on the current generation (1997) of workstation computers. Using PARAMETER statements to dimension a class of arrays means that only one number is set. PARAMETER statements are in separate files. The INCLUDE statement causes these files to be incorporated into each subroutine containing the given class of arrays, thus setting the dimensioning parameter for compilation. Some array limits are set by calculation within PARAMETER statements. The structure of the files containing PARAMETER statements and common blocks is the same as for the HST3D program (see chapter 4).

In case the dimension limits for the DIMEN program need to be increased, the parameter names and the names of the files where they are located are listed below.

KX,KY,KZ Maximum number of nodes in the x-, y-, and z-directions (`param6.inc`).
KXYZ Maximum number of nodes in simulation region (`param1.inc`).
KPMZ Maximum number of porous media zones (`param2.inc`).
KSBC Maximum number of cells (nodes) with a specified-value boundary condition (`param3.inc`).
KFBC Maximum number of cell faces with a specified-flux boundary condition (`param3.inc`).
KLBC Maximum number of cell faces with a leakage boundary condition (`param3.inc`).
KETBC Maximum number of cell faces with an evapotranspiration boundary condition (`param3.inc`).
KAIFC Maximum number of cell faces with an aquifer-influence-function boundary condition (`param3.inc`).
KHCBC Maximum number of cell faces with a heat-conduction boundary condition (`param3.inc`).
KHCN Maximum number of nodes for the finite-difference solution of the heat-conduction boundary condition (`param3.inc`).
KWEL Maximum number of wells (`param4.inc`).

The user may need to increase these values to handle very large field-scale simulation problems on large computers. A "make" utility, when available, can be used to have the computer select just which subroutines must be recompiled for DIMEN after a parameter change. The release file contains a makefile for compilation purposes.

Almost all parameters and variables are passed through common blocks. Many are the same as in the HST3D program. Other named common blocks are used, with the naming convention the same as for the HST3D program, but without the Y or N component. This component indicates which common blocks are included in check-point dumps for HST3D, and therefore, is unnecessary for the DIMEN program. All common blocks appear in the main program for easy reference and for static-storage allocation on some computer systems.

A.2.3 File Usage

The DIMEN program uses several sequential-access files on disc. The Fortran unit numbers can be specified by the user in the `param0.inc` file for all but the last two items. The following list gives the

Fortran unit names, the default unit number, and a description of the contents of each file.

FUINC {10}	Input file with comments
FUINS {45}	Input without comments
FUP {12}	Temporary file for input of each `paramn.inc` file
FURDE {11}	Echo of the input data as it is read
FULP {46}	Output data for the line printer or computer screen
FUC {13}	Scratch file for first pass
FU {2*n*}	Temporary file for `paramn.inc`
FU {3*n*}	New file for `paramn.inc`

The last two sets of file unit numbers are created within the program. All files are opened from within the program, and their names are supplied or built from a suffix supplied by the user or created by the program. Temporary, scratch, and read echo files are deleted at the end of the run.

A.2.4 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. There are no default values for this program.

A.2.5 Program Execution

The data-input file for a given problem to be simulated must have been created previously. The DIMEN program is executed interactively. The name of the data-input file for the simulation problem and a suffix to identify the output files are requested from the user at the terminal. The user is also requested to choose a redimensioning option from the following:

1. Increase or reduce each array dimension as needed to the minimum required for the problem data set being analyzed.
2. Increase the array dimensions as needed but leave the others unchanged.

The current PARAMETER statement files (`paramn.inc`) must be present in the same sub-directory as the data-input file. The new PARAMETER statement files with modified parameter values will be created with the suffix `.n`. If the current PARAMETER statement files are identical to those used for the compilation that produced the current executable file of HST3D, then those dimensions in that executable file that are too small for the simulation problem being analyzed will be identified. If some other set of PARAMETER statement files is in the current sub-directory, the new parameter values will still provide sufficient dimensions for the simulation problem being analyzed. However, some of the new parameter files may be identical to their old versions. At the end of execution, all files are closed and scratch files are deleted by the CLOSEF subroutine.

A.3 OUTPUT DESCRIPTION

Two types of output result from running the DIMEN program. Two output files are written that may be displayed on a computer screen or sent to a printer. The suffix *xxx* is a user-supplied identifier for the current execution of the program.

A read-echo file (`Rde.`*xxx*) is produced that is the same as for the HST3D program. The general output file (`Out.`*xxx*) contains tables listing values of dimensioning parameters. The first table contains those used by the DIMEN program. The next table includes the values from the PARAMETER statements provided as DIMEN input data. Usually they will be identical to those used in the currently compiled executable file for HST3D. The last table contains the dimension-parameter values needed for the simulation defined by the data-input file to be performed. The remainder of the listing contains the coordinate locations of the node points, locations of each type of boundary condition, and the boundary-condition index array. Any error messages concerning simulation region geometry and dimensioning are included at the end of this general output file. The new PARAMETER statement files (`param`*n*`.inc.n`) contain parameter values adjusted as necessary to accommodate the dimensioning requirements of the simulation problem being pre-processed.

A.4 COMPUTER SYSTEM CONSIDERATIONS

All arrays appear in Fortran COMMON statements defining common blocks with their dimensions set by Fortran PARAMETER statements. The INCLUDE statement is used to incorporate the parameter and common block files into each appropriate subroutine. The storage requirement on the Data General computer for the executable-code module is about 460 kilobytes, exclusive of the arrays, when compiled with full optimization by the Green Hills Fortran-88000 Compiler Version 1.8.6.1.

Although the standards for FORTRAN-77 have been followed as closely as possible, there are always problems of portability to different computer systems. There are no alternative entry points into any of the subroutines. 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.

The Fortran unit numbers for the input and output files are set in a PARAMETER statement. This allows for easy modification to suit different operating systems. Some operating systems require the BLOCKDATA subprogram to be contained in the same file as an executable subprogram. Version 2 of the DIMEN code, as distributed, has the main program, and each subprogram, including BLOCKDATA, contained in a separate file.

A.4.1 Computer System Requirements

To successfully run a dimension-checking calculation, an appropriate amount of computational power and storage needs to be available. The size of the DIMEN program is large due to a generous array space being allocated to handle large-scale simulation data sets. The executable module is about 14 MB. As an example, an input file for HST3D of 400 kilobytes generated 2.2 MB of output files. On the Data General-530, this calculation took about 2 minutes of CPU time. The computer resources needed for the DIMEN program are minor relative to those required by HST3D.

A.5 COMPUTER-CODE VERIFICATION

The DIMEN program was verified by comparing the dimension requirements that it calculated with those calculated by HST3D. The new PARAMETER statements were verified by comparing to the originals and by successful compilation when included in the HST3D code. No example problem is presented, because no new data files are required to be prepared by the user of this program.

APPENDIX B — BCFLOW: A POST-PROCESSOR FOR CALCULATING BOUNDARY FLOW RATES AND CUMULATIVE AMOUNTS FROM HST3D SIMULATIONS

B.1 INTRODUCTION

Version 2 of the HST3D simulator calculates flow rates and cumulative amounts of fluid, heat, and solute for each boundary cell for each type of boundary condition. These values are written to a file, `Bcnfr.`*xxx*, for post-processing whereby total flow rates and cumulative amounts can be calculated for a set of boundary-face subareas selected by the user. In the HST3D simulator, the trapezoidal rule is used for temporal integration of the flow rates to obtain the cumulative amounts.

The flow rates and cumulative amounts are segregated by boundary-condition type: specified value, specified flux, leakage, evapotranspiration, aquifer-influence function, and heat conduction. They are further segregated by fluid mass, fluid volume, heat, and solute as the transported quantities. The final segregation is by the user-specified facial area zones.

The program BCFLOW does the summations of total flow rates and cumulative flow amounts for zones of cells forming boundary facial areas selected by the user. This appendix documents Version 2.3 of BCFLOW which is designed to work with Version 2 of the HST3D code.

The BCFLOW program reads the boundary-node flow-rate output file, `Bcnfr.`*xxx*, generated by HST3D for a given simulation, and a data file defining the zones of facial areas of boundary cells for which total flow rates and cumulative flow amounts of fluid, heat, and solute are desired. Then, the program totals the flow rates and cumulative amounts over the boundary nodes for the specified zones of boundary cells.

B.1.1 Limitations

Quantities for only one type of boundary condition per zone of boundary facial area can be summed. At present, BCFLOW cannot maintain separate sums for the same type of boundary condition on two or three faces of the same cell. All quantities of a given boundary-condition type for every boundary face of a given cell are combined to form a net summation. Quantities associated with wells are not recognized in Version 2.3 of the BCFLOW program.

B.2 COMPUTER CODE DESCRIPTION

B.2.1 Program Structure and Organization

The BCFLOW computer code is written in FORTRAN-77. Code conforming to ANSI standards (American National Standards Institute, 1978) has been used as closely as possible to provide maximum portability. An exception to these standards is the usage of INCLUDE statements. Version 2.3 of the program code consists of a main routine and 19 subroutines. There are 23 files incorporated into the source code by INCLUDE statements. The program length is approximately 1700 lines of code including comment and continuation lines. The following is a list of the routines and a brief description of their function:

BCFLOW Main routine that drives the program execution. The basic steps are (1) open files, read, initialize unit labels, error check for dimensioned space, and write output for defining the

boundary-condition types to be used for the simulation being post-processed, and reading the flow-rate and cumulative-amount data for the boundary nodes generated by HST3D; (2) read, initialize accumulation variables, error check for dimensioned space, and write output for boundary-face zone definition information; (3) calculate the total flow rates and cumulative amounts for each zone of boundary nodes specified; (4) write the tables of results; and (5) close files and terminate program execution.

The following subroutines are described in their order of execution.

OPENF Requests the user to enter the name of the data-input file, the identification suffix for the output file of boundary nodal-flow rates from HST3D, and an identification suffix for output from this post-processing run.

READ1 Reads the data file identifying the simulation and specifying the boundary-condition types for which zones will be subsequently defined, reads the file of flow rates and cumulative amounts at boundary nodes.

INIT1 Initializes the unit labels.

ERROR1 Checks array dimensions for sufficient space.

WRITE1 Writes information about problem definition and basic dimensioning.

READ2 Reads the data that defines zones of boundary faces and the boundary-condition type for each zone specified for flow-rate and cumulative amount totalization.

INIT2 Initializes the accumulation arrays to zero.

ERROR4 Checks array dimensions for sufficient space for each boundary-condition type and the number of zones defined.

WRITE2 Writes the node location arrays, boundary-zone area definitions, and a geometric printout of the boundary-zone configuration.

SUMCAL Performs the summation calculations of total flow rates and cumulative amounts for each boundary-zone area.

WRITE4 Writes a table of the total flow rates and the total cumulative amounts of fluid, heat, and solute for each boundary zone defined.

CLOSEF Closes files, deletes scratch files, and prints any error messages.

The following subroutines are listed in alphabetical order. Some are called from several routines.

ERRPRT Writes the error messages to the output file.

SUMCF Sums the cumulative flow amounts from zones of boundary cells for flow, volumetric flow, heat, and solute.

SUMFLO Sums the flow rates from zones of boundary cells for flow, volumetric flow, heat, and solute.

The following subroutines are low-level utility routines listed in alphabetical order. Some are called several times from various locations.

HUNT Hunts through an array to locate the index, i, of the element such that the given number lies between element i and element i+1.

INCIDX Finds the two indices, i_1 and i_2 , of an array such that element i_1 and element i_2 are the range of the array elements contained between two given values.

IREWI Reads, error checks, echo writes, and initializes integer-array elements that are input as zones of uniform values over a rectangular area of boundary-cell faces, or are input as node-

by-node distributions.

PRNTAR Writes one-dimensional or two-dimensional arrays in tabular form to a specified file.

PRCHAR Writes two-dimensional character arrays in tabular form to various files.

A chart showing the main sequence of subroutine execution, and the linkage between the subroutines appears in Figure B.1. The primary subroutines are on the left and the secondary and utility subroutines are to the right in each column. The sequence of execution on each page is from top to bottom of the left column followed by the right column. The factor in front of some routine names indicates multiple calls. Some utility subroutines are listed more than once for graphical clarity.

```
========================================
 Calling Tree

    Program BCFLOW
    |
    +--Subroutine OPENF
    |
    +--Subroutine READ1
    |
    +--Subroutine INIT1
    |
    +--Subroutine ERROR1
    |
    +--Subroutine WRITE1
    |
    +--Subroutine READ2
    |  |
    |  +--Subroutine IREWI
    |     |
    |     +--3*Subroutine INCIDX
    |        |
    |        +--2*Subroutine HUNT
    |
    +--Subroutine INIT2
    |
    +--Subroutine ERROR4
    |
    +--Subroutine WRITE2
    |  |
    |  +--5*Subroutine PRNTAR
    |  |
    |  +--Subroutine PRCHAR
    |
    +--Subroutine SUMCAL
    |  |
    |  +--Subroutine SUMFLO
    |  |
    |  +--Subroutine SUMCF
    |  |
    |  +--Subroutine SUMFLO
    |  |
    |  +--Subroutine SUMCF
    |  |
    |  +--Subroutine SUMFLO
    |  |
    |  +--Subroutine SUMCF
    |  |
    |  +--Subroutine SUMFLO
    |  |
    |  +--Subroutine SUMCF
    |  |
    |  +--Subroutine SUMFLO
    |  |
    |  +--Subroutine SUMCF
    |
    +--Subroutine WRITE4
    |
    +--Subroutine CLOSEF
       |
       +--Subroutine ERRPRT

========================================
```

Figure B.1. Connection chart for the BCFLOW main program and subprograms showing routine hierarchy.

B.2.2 Memory Allocation, Array-Size Requirements, and Subprogram Communication

Each of the arrays used by BCFLOW is dimensioned by a Fortran PARAMETER statement. The dimension values need to be at least as large as those used to allocate the corresponding type of array space for the executable file of HST3D that was used to simulate the problem for which the post-processing BCFLOW program is being run. It may be convenient to use the same parameter files as were used to create the executable file of HST3D. Using PARAMETER statements to dimension a class of arrays means that only one number is set. PARAMETER statements are in separate files. The INCLUDE statement causes these files to be incorporated into each subroutine with the given class of arrays, thus setting the dimensioning parameter for compilation. Some array limits are set by calculation within PARAMETER statements. The structure of the parameter and common-block files is the same as for the HST3D program (see chapter 4).

Almost all parameters and variables are passed through common blocks. Many are the same as in the HST3D program. All common blocks appear in the main program for easy reference and for static-storage allocation on some computer systems.

In case the dimension limits for the BCFLOW program need to be increased, the parameter names and the names of the files where they are located are listed below.

LX,LY,LZ Maximum number of nodes in the x-, y-, and z-directions in the simulation region (`param6.inc`).
LXYZ Maximum number of nodes in the simulation region (`param1.inc`).
LBCZ Maximum number of boundary-condition zones (`param2.inc`).
LSBC Maximum number of cells (nodes) with a specified-value boundary condition (`param3.inc`).
LFBC Maximum number of cell faces with a specified-flux boundary condition (`param3.inc`).
LLBC Maximum number of cell faces with a leakage boundary condition (`param3.inc`).
LETBC Maximum number of cell faces with an evapotranspiration boundary condition (`param3.inc`).
LAIFC Maximum number of cell faces with an aquifer-influence-function boundary condition (`param3.inc`).
LHCBC Maximum number of cell faces with a heat-conduction boundary condition (`param3.inc`).

A "make" utility, when available, can be used to have the computer select just which subroutines must be recompiled for BCFLOW after a parameter change. The release file contains a makefile for compilation purposes.

B.2.3 File Usage

The BCFLOW program uses five sequential-access files on disc. The Fortran unit numbers can be specified by the user in the file `param0.inc`. The following list gives the Fortran unit names, the default unit number, and a description of each file.

FUINC {10} Input file with comments
FUINS {15} Input without comments

FUBNFR {13}	Input of boundary nodal-flow rates and cumulative amounts from the HST3D simulation
FURDE {11}	Echo of the input data as it is read
FULP {16}	Output data for the line printer or monitor screen

These files are opened from within the program, and their names are supplied or built from a suffix supplied by the user. Scratch and read echo files are deleted at the end of the run.

B.2.4 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. There are no default values for this program.

B.2.5 Program Execution

The data-input file (`Bcnfr.`*xxx*) containing the boundary-nodal flow rates and cumulative amounts from a given simulation problem must have been created by a previous simulation run. If output at more than one time value is present in file, `Bcnfr.`*xxx*, an edited file must be created containing the results at only the time value desired. The BCFLOW program is executed interactively. The user is requested to provide the name of a data-input file defining the zones of boundary cells to be totalized, a suffix to identify the HST3D file containing boundary-nodal flow rates, and a suffix to identify the output files. At the end of execution, all files are closed and scratch files are deleted by the CLOSEF subroutine.

B.3 DATA-INPUT FILE DESCRIPTION

The data-input file is free format and may be freely commented like the HST3D file. Please refer to chapter 5 of the present documentation for general information on list-directed input.

B.3.1 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 B.1. These comments identify all the input-variable names, indicate those that are integer and logical, and show the conditions for the optional items. The user actually can enter the data between the lines of a copy of this file, using section B.3.1.2 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 B.3.1.2. Optional input records are indicated by (O) followed by the 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.

Table B.1. File of comment lines from the data-input form for the BCFLOW program

```
C.....BCFLOW Data-Input Form
C.....Version 2.3
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...   (O) - Optional data with conditions for requirement
C...   P [n.n.n] - The record where parameter P is set
C.....Input by x,y,z range format is;
C.0.1.. X1,X2,Y1,Y2,Z1,Z2
C.0.2.. VAR,IMOD
C...     Use as many of line 0.1 & 0.2 sets as necessary
C...     End with line 0.3
C.0.3.. END OR end
C...   [I] - Indicates an integer variable
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Start of the data file
C.....Identification and boundary condition type data - READ1
C.1.1 .. TITLE line 1
C.1.2 .. TITLE line 2
C.1.3 .. NSBC,NFBC,NLBC,NETBC,NAIFC,NHCBC; all[I]
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Boundary zone definition information - READ2
C-------------------------------------------------------------------------------
C.....     Specified value b.c., BCTYPE=1
C.2.1.1 .. ZONNAM
C.2.1.2 .. BCTYPE by x,y,z range {0.1-0.3} for NSBC nodes;(O) -
C..          NSBC [1.3] > 0
C.....Use as many 2.1.n lines as necessary
C-------------------------------------------------------------------------------
C.....     Specified flux b.c., BCTYPE=2
C.2.2.1 .. ZONNAM
C.2.2.2 .. BCTYPE by x,y,z range {0.1-0.3} for NFBC nodes;(O) -
C..          NFBC [1.3] > 0
C.....Use as many 2.2.n lines as necessary
C-------------------------------------------------------------------------------
C.....     Aquifer and river leakage b.c., BCTYPE=3
C.2.3.1 .. ZONNAM
C.2.3.2 .. BCTYPE by x,y,z range {0.1-0.3} for NLBC nodes;(O) -
C..          NLBC [1.3] > 0
C.....Use as many 2.3.n lines as necessary
C-------------------------------------------------------------------------------
C.....     Evapotranspiration b.c., BCTYPE=7
C.2.4.1 .. ZONNAM
C.2.4.2 .. BCTYPE by x,y,z range {0.1-0.3} for NETBC nodes;(O) -
C..          NETBC [1.3] > 0
C.....Use as many 2.4.n lines as necessary
C-------------------------------------------------------------------------------
C.....     Aquifer influence function b.c., BCTYPE=4
C.2.5.1 .. ZONNAM
C.2.5.2 .. BCTYPE by x,y,z range {0.1-0.3} for NAIFC nodes;(O) -
C..          NAIFC [1.3] > 0
C.....Use as many 2.5.n lines as necessary
C-------------------------------------------------------------------------------
C.....     Heat conduction b.c., BCTYPE=5
```

Table.B.1. File of comment lines from the data-input form for the BCFLOW program — Continued

```
C.2.6.1 .. ZONNAM
C.2.6.2 .. BCTYPE by x,y,z range {0.1-0.3} for NHCBC nodes;(O) -
C..          NHCBC [1.3] > 0
C.....Use as many 2.6.n lines as necessary
C-----------------------------------------------------------------------------
C.2.9 .. End with END
end
C.....End of the data file
C-----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
```

B.3.1.1 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, or READ2 reads that record for the values 1 or 2. The record number identifies a line of input data, with the two component number (N2.N3) enabling a logical group structure to be assigned. Optional data records are indicated to the right of the list of variables, and the conditions under which these data are required are noted. If a particular input record is not needed, it is mandatory that it be omitted. A record number in square brackets following a variable indicates the record where that variable is first set. The notation [I] indicates an integer variable.

Input by x,y,z range is used to define a rectangular plane of nodes. Further explanation of the set of records required to input by x,y,z range is given in chapter 5. A plane implies that one pair of the coordinate range values must be equal. The extent of this plane is all boundary-cell faces associated with nodes contained within the ranges of the x, y, and z coordinates. An identifier of boundary-condition type follows the coordinate ranges. It is permissible to use rectangular planes that include cells that are outside the simulation region, or to include cells that are not of the boundary-condition type that is being selected for summation over the facial area. It is also permissible to define overlapping zones.

The units for the various parameters must be the same as those used by HST3D for the simulation problem being post-processed. The time unit used by BCFLOW will be the one that was employed for the simulation.

B.3.1.2 Input-Record Descriptions The following order generally has been observed for data input (1) fundamental and selection of boundary-condition type information; and (2) definition of the planar areas for which flow rates and cumulative amounts are to be totalized. Each input record number identifies a particular record in the data-input form listed in Table B.1.

I. Problem specification and boundary-type data - 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 second character string of up to 80 characters.

1.3 NSBC, NFBC, NLBC, NETBC, NAIFC, NHCBC; all integers.

NSBC Number of cells (nodes) with a specified-pressure, specified-temperature, and(or) specified-mass-fraction boundary condition.
NFBC Number of cells with a specified-flux boundary condition; flow, heat, and(or) solute.
NLBC Number of cell faces with a leakage boundary condition.
NETBC Number of cell faces with an evapotranspiration boundary condition.
NAIFC Number of cell faces with an aquifer-influence-function boundary condition.
NHCBC Number of cell faces with a heat-conduction boundary condition.

These values only need to be set to one for each boundary-condition type to be selected for totalization.

II. Boundary-zone definition information - READ2.

The following record sets are used to select boundary facial areas for totalization. At least one facial area must be defined for each boundary-condition type flagged in record 1.3. The digit, n, ranges from 1 to 6 for the 6 boundary-condition types that currently can be processed, as explained below.

2.n.1 ZONNAM; Optional, required only if boundary-condition cells of the current b.c. type have been selected for totalization in record 1.3.

ZONNAM Character string of up to 60 characters to identify a zone of boundary-cell faces.

2.n.2 BCTYPE by x,y,z range; Optional, required only if boundary-condition cells of the current b.c. type have been selected for totalization in record 1.3.

BCTYPE Flag indicating the type of boundary condition selected for totalization within this defined facial area zone (integer).

1 Specified-value boundary condition.
2 Flux boundary condition.
3 Leakage boundary condition (aquifer or river).
4 Aquifer-influence-function boundary condition.
5 Heat-conduction boundary condition.
7 Evapotranspiration boundary condition.

Record sets of this form define the facial-area zones within which the boundary-cell flow rates and cumulative amounts are summed. The ranges of the coordinates in the x, y, and z directions define the rectangular plane of nodes included for a given zone. To define a plane, as opposed to a volume, requires that the coordinate range in one of the directions must be zero. Zones may be defined that include elements outside the simulation region or elements of a different boundary-condition type than that selected by BCTYPE. Overlap of defined zones is allowed. For cylindrical coordinates, the y range is not used and may be given zero or blank entries. The total number of zones defined will be counted and assigned a sequential index number by the program. Boundary facial-area zones may be defined in any order with respect to boundary-condition type. However, it is convenient to keep all defined zones of the same boundary-condition type grouped together in the data file. For this reason, the `Data.form` file lists a pair of zone definition records for each boundary-condition type supported.

2.9 End the input of zone definitions with `END` or `end`.

This ends the input data-file description.

B.4 OUTPUT DESCRIPTION

Two output files are created by the BCFLOW program. These files may be displayed on a monitor screen or sent to a printer. A user-selected suffix, *xxx*, is used to identify the output files.

A read-echo file (`Rde.`*xxx*) is produced that echos the zone definition information. The output file (`O.bczf.`*xxx*) contains a table listing values of total boundary-flow rates and total cumulative amounts for each zone arranged first by boundary-condition type and second by boundary-zone definition. Both mass and volumetric flow amounts are listed. The sign convention used is that inflows are positive and outflows are negative. The number of nodes of each boundary-condition type and the nodal-coordinate locations for each axis of the grid are also written to the output file. The output file also contains an array showing the geometry of the defined boundary-facial area zones identified by zone number.

B.5 COMPUTER SYSTEM CONSIDERATIONS

All arrays appear in common blocks with their dimensions set by Fortran PARAMETER statements. The INCLUDE statement is used to incorporate the parameter and common-block files into each appropriate subroutine. The storage requirement on the Data General computer for the executable-code module is about 300 kilobytes, exclusive of the arrays, when compiled with full optimization by the Green Hills Fortran-88000 Compiler Version 1.8.6.1.

Although the standards for FORTRAN-77 have been followed as closely as possible, there are always problems of portability to different computer systems. There are no alternative entry points into any of the subroutines. 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.

The Fortran unit numbers for the input and output files are set in a PARAMETER statement. This allows for easy modification to suit different operating systems. Some operating systems require the BLOCKDATA subprogram to be contained in the same file as an executable subprogram. Version 2.3 of the code, as distributed, has the main program, and each subprogram, including BLOCKDATA, contained in a separate file.

B.5.1 Computer System Requirements

To successfully run a set of post-processing calculations in a reasonable time, an appropriate amount of computational power and storage needs to be available. As an example, a field-scale simulation involving about 44,000 nodes required an executable module of about 3.8 MB, an input file from HST3D of 1.3 MB, and generated 0.5 MB of output files. On the Data General-530, this calculation took about 5 minutes of CPU time. The computer resources needed for the BCFLOW program are minor relative to those needed by HST3D.

B.6 COMPUTER-CODE VERIFICATION

The BCFLOW program was verified by hand calculation for a small test problem involving 3 nodes in each direction. This problem will not be presented here.

B.6.1 AN EXAMPLE PROBLEM

An example of the use of the BCFLOW post-processing program for flow rates and cumulative amounts is given here. It is derived from the verification example of saltwater intrusion into a coastal aquifer from Huyakorn and others (1987). The input-data file is used to define the boundary-facial areas over which flow rates and cumulative flow amounts are desired at selected times during the simulation. For this example, only results from the end of the simulation are calculated.

Table B.2 contains the input-data file for this example. Four boundary zones are defined. One is at the inland boundary, the second is at the coastal boundary from 20 ft below the land surface to the aquifer bottom, the third is at the coastal boundary within 20 ft of the land surface, and the fourth is the land surface. The coastal boundary zones will separate the flow rates and cumulative amounts into saltwater inflow or intrusion rates and amounts and outflow brackish-water rates and amounts. Annotations are made in bold within the data file shown Table B.2. They follow the data descriptor lines but precede the actual data lines to which they refer.

Table B.2. Data file for the BCFLOW program for the coastal-aquifer example with annotations

```
C.....BCFLOW Data-Input Form
C.....Version 2.3
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...   (O) - Optional data with conditions for requirement
C...   P [n.n.n] - The record where parameter P is set
C.....Input by x,y,z range format is;
C.0.1.. X1,X2,Y1,Y2,Z1,Z2
C.0.2.. VAR,IMOD
C...     Use as many of line 0.1 & 0.2 sets as necessary
C...     End with line 0.3
C.0.3.. END OR end
C...   [I] - Indicates an integer variable
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Start of the data file
C.....Identification and boundary condition type data - READ1
C.1.1 .. TITLE line 1
   Exercise problem 3: adapted from Example 4 from Huyakorn WRR 23,2,p.293(1987)
C.1.2 .. TITLE line 2
    a three-dimensional salt water intrusion problem
C.1.3 .. NSBC,NFBC,NLBC,NETBC,NAIFC,NHCBC; all[I]
```

The program will count the number of nodes with each boundary condition type selected here using a non-zero integer.

```
1 1 4*0
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C.....Boundary zone definition information - READ2
C-------------------------------------------------------------------------------
C.....     Specified value b.c., BCTYPE=1
C.2.1.1 .. ZONNAM
C.2.1.2 .. BCTYPE by x,y,z range {0.1-0.3} for NSBC nodes;(O) -
C..          NSBC [1.3] > 0
```

The ZONNAM and BCTYPE records appear in sets for each boundary-area defined.

```
 Inland boundary
0. 0. 0. 9600. 0. 200.
1 1
end
 Coastal boundary saline lower
14400. 14400. 0. 9600. 0. 180.
1 1
end
 Coastal boundary fresh upper
14400. 14400. 0. 9600. 181. 200.
1 1
end
C.....Use as many 2.1.n lines as necessary
C-------------------------------------------------------------------------------
C.....     Specified flux b.c., BCTYPE=2
C.2.2.1 .. ZONNAM
```

Here is an example of separating the ZONNAM and BCTYPE records.

```
 Land surface
C.2.2.2 .. BCTYPE by x,y,z range {0.1-0.3} for NFBC nodes;(O) -
C..          NFBC [1.3] > 0
200. 14200. 0. 9600. 200. 200.
2 1
```

Table.B.2. Data file for the coastal-aquifer example with annotations — Continued

```
end
C.....Use as many 2.2.n lines as necessary
C-------------------------------------------------------------------------
C.....     Aquifer and river leakage b.c., BCTYPE=3
C.2.3.1 .. ZONNAM
C.2.3.2 .. BCTYPE by x,y,z range {0.1-0.3} for NLBC nodes;(O) -
C..          NLBC [1.3] > 0
C.....Use as many 2.3.n lines as necessary
C-------------------------------------------------------------------------
C.....     Evapotranspiration b.c., BCTYPE=7
C.2.4.1 .. ZONNAM
C.2.4.2 .. BCTYPE by x,y,z range {0.1-0.3} for NETBC nodes;(O) -
C..          NETBC [1.3] > 0
C.....Use as many 2.4.n lines as necessary
C-------------------------------------------------------------------------
C.....     Aquifer influence functions, BCTYPE=4
C.2.5.1 .. ZONNAM
C.2.5.2 .. BCTYPE by x,y,z range {0.1-0.3} for NAIFC nodes;(O) -
C..          NAIFC [1.3] > 0
C.....Use as many 2.5.n lines as necessary
C-------------------------------------------------------------------------
C.....     Heat conduction b.c., BCTYPE=4
C.2.6.1 .. ZONNAM;(O) - NHCBC [1.3] > 0
C.2.6.2 .. BCTYPE by x,y,z range {0.1-0.3} for NHCBC nodes;(O) -
C..          NHCBC [1.3] > 0
C.....Use as many 2.5.n lines as necessary
C-------------------------------------------------------------------------
C.2.9 .. End with END
end
C.....End of the data file
C-------------------------------------------------------------------------
C-------------------------------------------------------------------------
```

The program requires nodal flow rate and cumulative amount data obtained from file, `Bcnfr.example`. Parameter PRIBCF (record 3.9.1 of Table 5.1) was used to cause the `Bcnfr.example` file to be written with data at the end of the simulation. If output had been written for several time values, the `Bcnfr.example` file would need to be edited to contain only the data for the time value desired.

Table B.3 contains the output file, `O.bczf.example`. The boundary-condition zone mask array shows the zone numbers of the various boundary zones defined for this calculation. A table at the beginning identifies the zones. The various flow rates and cumulative amounts are shown with labels identifying the boundary zones. They are arranged by boundary-condition type.

Table B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example

```
*************************************************
*                                               *
*    Boundary Flow-rate Accumulator for the     *
*    Three-Dimensional Flow, Heat and Solute    *
*   Transport Simulator:BCFLOW -- Release 2.3   *
*                                               *
*                                               *
*************************************************



          Exercise problem 3: adapted from Example 4 from Huyakorn WRR 23,2,p.293(1987)
           a three-dimensional salt water intrusion problem

            Test problem from Huyakorn et al., WRR 23(2), p293, 1987
         a three-dimensional salt water intrusion problem, unconfined

       *** Fundamental Information ***
   Isothermal simulation
   Solute transport simulation
   Cartesian coordinates
   Data is in  U.S. customary units
   Time unit is days

       *** Problem Dimension Information ***
Number of nodes in x-direction ................................ NX ...     62
Number of nodes in y-direction ................................ NY ...     19
Number of nodes in z-direction ................................ NZ ...     11
Total number of nodes ......................................... NXYZ .  12958

Number of specified pressure or temperature or mass fraction b.c. NSBC .    418
Number of specified flux b.c. cells ........................... NFBC .   1140
Number of leakage cells ....................................... NLBC .      0
Number of evapotranspiration cells............................. NETBC       0
Number of aquifer influence function cells..................... NAIFC       0
Number of heat conduction b.c. cells .......................... NHCBC       0
-------------------------------------------------------------------------------------------------------------------------


                X-direction node coordinates    (ft)

   1          2          3          4          5          6          7          8          9          10
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
   0.00      236.07      472.13      708.20      944.26     1180.33     1416.39     1652.46     1888.53     2124.59


  11          12          13          14          15          16          17          18          19          20
2360.66     2596.72     2832.79     3068.85     3304.92     3540.98     3777.05     4013.11     4249.18     4485.25


  21          22          23          24          25          26          27          28          29          30
4721.31     4957.38     5193.44     5429.51     5665.57     5901.64     6137.70     6373.77     6609.84     6845.90


  31          32          33          34          35          36          37          38          39          40
7081.97     7318.03     7554.10     7790.16     8026.23     8262.30     8498.36     8734.43     8970.49     9206.56


  41          42          43          44          45          46          47          48          49          50
9442.62     9678.69     9914.75    10150.82    10386.89    10622.95    10859.02    11095.08    11331.15    11567.21


  51          52          53          54          55          56          57          58          59          60
11803.28    12039.34    12275.41    12511.48    12747.54    12983.61    13219.67    13455.74    13691.80    13927.87


  61          62
14163.93    14400.00


                     Y-direction node coordinates   (ft)

   1           2           3           4           5           6           7           8           9          10
   0.00      533.33     1066.67     1600.00     2133.33     2666.67     3200.00     3733.33     4266.67     4800.00


  11          12          13          14          15          16          17          18          19
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
      5333.33     5866.67     6400.00     6933.33     7466.67     8000.00     8533.33     9066.67     9600.00



                          Z-direction node coordinates   (ft)

         1           2           3           4           5           6           7           8           9          10
         0.00       20.00       40.00       60.00       80.00      100.00      120.00      140.00      160.00      180.00



        11
       200.00



                  ** Boundary facial area zone definitions  **
Zone no.       Identification                                          B.C. type
    1      Inland boundary                                               1
    2      Coastal boundary saline lower                                 1
    3      Coastal boundary fresh upper                                  1
    4      Land surface                                                  2

                                 Zone numbers for boundary facial areas

                  Areal Layers

                                                          K =  1

         1          2          3          4          5          6          7          8          9         10
 19   1
 18   1
 17   1
 16   1
 15   1
 14   1
 13   1
 12   1
 11   1
 10   1
  9   1
  8   1
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
 7   1
 6   1
 5   1
 4   1
 3   1
 2   1
 1   1

       11        12        13        14        15        16        17        18        19        20

       21        22        23        24        25        26        27        28        29        30

       31        32        33        34        35        36        37        38        39        40

       41        42        43        44        45        46        47        48        49        50

       51        52        53        54        55        56        57        58        59        60

       61        62
19            2
18            2
17            2
16            2
15            2
14            2
13            2
12            2
11            2
10            2
 9            2
 8            2
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
 7              2
 6              2
 5              2
 4              2
 3              2
 2              2
 1              2

                                                  K =  2

        1          2          3          4          5          6          7          8          9         10
19      1
18      1
17      1
16      1
15      1
14      1
13      1
12      1
11      1
10      1
 9      1
 8      1
 7      1
 6      1
 5      1
 4      1
 3      1
 2      1
 1      1


       11         12         13         14         15         16         17         18         19         20


       21         22         23         24         25         26         27         28         29         30


       31         32         33         34         35         36         37         38         39         40
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
        41         42         43         44         45         46         47         48         49         50


        51         52         53         54         55         56         57         58         59         60


        61         62
 19              2
 18              2
 17              2
 16              2
 15              2
 14              2
 13              2
 12              2
 11              2
 10              2
  9              2
  8              2
  7              2
  6              2
  5              2
  4              2
  3              2
  2              2
  1              2

                                                   K =  3

         1          2          3          4          5          6          7          8          9         10
 19   1
 18   1
 17   1
 16   1
 15   1
 14   1
 13   1
 12   1
 11   1
 10   1
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
 9   1
 8   1
 7   1
 6   1
 5   1
 4   1
 3   1
 2   1
 1   1

       11        12        13        14        15        16        17        18        19        20

       21        22        23        24        25        26        27        28        29        30

       31        32        33        34        35        36        37        38        39        40

       41        42        43        44        45        46        47        48        49        50

       51        52        53        54        55        56        57        58        59        60

       61        62
19            2
18            2
17            2
16            2
15            2
14            2
13            2
12            2
11            2
10            2
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
 9          2
 8          2
 7          2
 6          2
 5          2
 4          2
 3          2
 2          2
 1          2
                                                  K =  4 to 9 are the same as 3

                                                  K = 10

        1         2         3         4         5         6         7         8         9        10
19   1
18   1
17   1
16   1
15   1
14   1
13   1
12   1
11   1
10   1
 9   1
 8   1
 7   1
 6   1
 5   1
 4   1
 3   1
 2   1
 1   1

       11        12        13        14        15        16        17        18        19        20

       21        22        23        24        25        26        27        28        29        30
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
        31         32         33         34         35         36         37         38         39         40

        41         42         43         44         45         46         47         48         49         50

        51         52         53         54         55         56         57         58         59         60

        61         62
 19              2
 18              2
 17              2
 16              2
 15              2
 14              2
 13              2
 12              2
 11              2
 10              2
  9              2
  8              2
  7              2
  6              2
  5              2
  4              2
  3              2
  2              2
  1              2

                                                  K = 11

         1          2          3          4          5          6          7          8          9         10
 19   1          4          4          4          4          4          4          4          4          4
 18   1          4          4          4          4          4          4          4          4          4
 17   1          4          4          4          4          4          4          4          4          4
 16   1          4          4          4          4          4          4          4          4          4
 15   1          4          4          4          4          4          4          4          4          4
 14   1          4          4          4          4          4          4          4          4          4
 13   1          4          4          4          4          4          4          4          4          4
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

12	1	4	4	4	4	4	4	4	4	4
11	1	4	4	4	4	4	4	4	4	4
10	1	4	4	4	4	4	4	4	4	4
9	1	4	4	4	4	4	4	4	4	4
8	1	4	4	4	4	4	4	4	4	4
7	1	4	4	4	4	4	4	4	4	4
6	1	4	4	4	4	4	4	4	4	4
5	1	4	4	4	4	4	4	4	4	4
4	1	4	4	4	4	4	4	4	4	4
3	1	4	4	4	4	4	4	4	4	4
2	1	4	4	4	4	4	4	4	4	4
1	1	4	4	4	4	4	4	4	4	4

	11	12	13	14	15	16	17	18	19	20
19	4	4	4	4	4	4	4	4	4	4
18	4	4	4	4	4	4	4	4	4	4
17	4	4	4	4	4	4	4	4	4	4
16	4	4	4	4	4	4	4	4	4	4
15	4	4	4	4	4	4	4	4	4	4
14	4	4	4	4	4	4	4	4	4	4
13	4	4	4	4	4	4	4	4	4	4
12	4	4	4	4	4	4	4	4	4	4
11	4	4	4	4	4	4	4	4	4	4
10	4	4	4	4	4	4	4	4	4	4
9	4	4	4	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4
3	4	4	4	4	4	4	4	4	4	4
2	4	4	4	4	4	4	4	4	4	4
1	4	4	4	4	4	4	4	4	4	4

	21	22	23	24	25	26	27	28	29	30
19	4	4	4	4	4	4	4	4	4	4
18	4	4	4	4	4	4	4	4	4	4
17	4	4	4	4	4	4	4	4	4	4
16	4	4	4	4	4	4	4	4	4	4

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

15	4	4	4	4	4	4	4	4	4	4
14	4	4	4	4	4	4	4	4	4	4
13	4	4	4	4	4	4	4	4	4	4
12	4	4	4	4	4	4	4	4	4	4
11	4	4	4	4	4	4	4	4	4	4
10	4	4	4	4	4	4	4	4	4	4
9	4	4	4	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4
3	4	4	4	4	4	4	4	4	4	4
2	4	4	4	4	4	4	4	4	4	4
1	4	4	4	4	4	4	4	4	4	4

	31	32	33	34	35	36	37	38	39	40
19	4	4	4	4	4	4	4	4	4	4
18	4	4	4	4	4	4	4	4	4	4
17	4	4	4	4	4	4	4	4	4	4
16	4	4	4	4	4	4	4	4	4	4
15	4	4	4	4	4	4	4	4	4	4
14	4	4	4	4	4	4	4	4	4	4
13	4	4	4	4	4	4	4	4	4	4
12	4	4	4	4	4	4	4	4	4	4
11	4	4	4	4	4	4	4	4	4	4
10	4	4	4	4	4	4	4	4	4	4
9	4	4	4	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4
3	4	4	4	4	4	4	4	4	4	4
2	4	4	4	4	4	4	4	4	4	4
1	4	4	4	4	4	4	4	4	4	4

	41	42	43	44	45	46	47	48	49	50
19	4	4	4	4	4	4	4	4	4	4

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

18	4	4	4	4	4	4	4	4	4	4
17	4	4	4	4	4	4	4	4	4	4
16	4	4	4	4	4	4	4	4	4	4
15	4	4	4	4	4	4	4	4	4	4
14	4	4	4	4	4	4	4	4	4	4
13	4	4	4	4	4	4	4	4	4	4
12	4	4	4	4	4	4	4	4	4	4
11	4	4	4	4	4	4	4	4	4	4
10	4	4	4	4	4	4	4	4	4	4
9	4	4	4	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4
3	4	4	4	4	4	4	4	4	4	4
2	4	4	4	4	4	4	4	4	4	4
1	4	4	4	4	4	4	4	4	4	4

	51	52	53	54	55	56	57	58	59	60
19	4	4	4	4	4	4	4	4	4	4
18	4	4	4	4	4	4	4	4	4	4
17	4	4	4	4	4	4	4	4	4	4
16	4	4	4	4	4	4	4	4	4	4
15	4	4	4	4	4	4	4	4	4	4
14	4	4	4	4	4	4	4	4	4	4
13	4	4	4	4	4	4	4	4	4	4
12	4	4	4	4	4	4	4	4	4	4
11	4	4	4	4	4	4	4	4	4	4
10	4	4	4	4	4	4	4	4	4	4
9	4	4	4	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4
3	4	4	4	4	4	4	4	4	4	4
2	4	4	4	4	4	4	4	4	4	4
1	4	4	4	4	4	4	4	4	4	4

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
        61          62
 19   4           3
 18   4           3
 17   4           3
 16   4           3
 15   4           3
 14   4           3
 13   4           3
 12   4           3
 11   4           3
 10   4           3
  9   4           3
  8   4           3
  7   4           3
  6   4           3
  5   4           3
  4   4           3
  3   4           3
  2   4           3
  1   4           3


-----------------------------------------------------------------------------------------------------

         Time ..............................................................   2.500E+04  (d  )

    Total flow rates and cumulative amounts for each zone


    Specified pressure, temperature, or mass fraction b.c.
              (average over time step, positive is into the region)

Zone no.  1   Inland boundary

              Current Time Value                                    Rates

Total fluid flow ........................................  8.050612E+07  (lb/d  )

Total specified concentration cell solute flow ............ -7.576581E-06  (lb/d  )

              Cumulative Summary                                  Amounts
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
Total fluid mass flow ....................................  2.015448E+12  (lb)

Total fluid volumetric flow ..............................  3.229886E+10  (ft^3)

Total specified concentration cell solute flow ............ -9.879566E-02  (lb)


    Specified pressure, temperature, or mass fraction b.c.
              (average over time step, positive is into the region)

Zone no.  2   Coastal boundary saline lower

              Current Time Value                                  Rates

Total fluid flow .........................................  9.635356E+06  (lb/d  )

Total specified concentration cell solute flow ............  7.356201E+05  (lb/d  )

              Cumulative Summary                                Amounts

Total fluid mass flow ....................................  2.747061E+11  (lb)

Total fluid volumetric flow ..............................  4.294972E+09  (ft^3)

Total specified concentration cell solute flow ............  2.179139E+10  (lb)


    Specified pressure, temperature, or mass fraction b.c.
              (average over time step, positive is into the region)

Zone no.  3   Coastal boundary fresh upper

              Current Time Value                                  Rates

Total fluid flow ......................................... -1.976753E+07  (lb/d  )

Total associated advective solute flow ................... -6.974297E+05  (lb/d  )

              Cumulative Summary                                Amounts

Total fluid mass flow .................................... -5.246034E+11  (lb)

Total fluid volumetric flow .............................. -8.204713E+09  (ft^3)
```

Table.B.3. Output file of flow rates and cumulative amounts for the coastal-aquifer example— Continued

```
Total associated advective solute flow ..................... -1.847960E+10  (lb)


    Specified flux b.c.
              (at end of time step, positive is into the region)

Zone no.  4   Land surface

              Current Time Value                                  Rates

Total fluid flow ........................................  2.324570E+07  (lb/d  )

Total associated advective solute flow ..................  0.000000E+00  (lb/d  )

              Cumulative Summary                                Amounts

Total fluid mass flow ...................................  5.811427E+11  (lb)

Total fluid volumetric flow .............................  9.313186E+09  (ft^3)

Total associated advective solute flow ..................  0.000000E+00  (lb)

                                      ***** Job Completed *****
```

APPENDIX C — SOFTWARE FOR THE HST3D SIMULATOR AND ASSOCIATED PROGRAMS

This appendix covers obtaining and installing the computer code for HST3D and associated programs. The primary source for the code package is through the Internet.

C.1 STORAGE SPACE ON HARD DISC

A directory where the HST3D program package is to reside will need to be created. Installation will be done from within that directory. Installing the new version into a new directory is a good practice. Any previous versions of HST3D should be backed up, particularly if they contain any custom modifications to be ported over to the new version.

When unpacked, this distribution requires roughly 30 MB of disc space for sources, include files, example data sets, and documentation. About 8 MB of executable files and object files will be created and installed to their respective destinations. Once installation has been completed, disc space may be recovered by removing the files no longer needed, such as object files, or perhaps even source. The space requirement can be reduced further by stripping the executable files of their symbols. Documentation might be removed once it has been printed.

C.2 OBTAINING THE SOFTWARE

The source code for the HST3D simulator and the DIMEN and BCFLOW programs with the include files, makefiles, and example data files with selected output files can be obtained from either of two hosts on the Internet using the File Transfer Protocol (ftp) utility. The USGS software repository is at `water.usgs.gov` and the developer's public repository is at `brrcrftp.cr.usgs.gov`. Instructions for obtaining the distribution file from the USGS software repository are available at the USGS Water Resources Information web page at URL;
`http://water.usgs.gov/software/ground_water.html`.

The procedure for obtaining the distribution file from the developer's repository is as follows:

1. Connect to the ftp server, `brrcrftp.cr.usgs.gov`.
2. Login as **`anonymous`**.
3. Change directory to `pub/klkipp/hst3d.release`.
4. Specify **`binary`** mode for transfer of a compressed file.
5. Download the compressed tar file, `hst3d-2.0.tar.gz` (about 5 MB).

C.3 INSTALLATION STEPS

1. Load the software
   Uncompress the tar file and extract the software. Several subdirectories will be created, three of which will be under the top-level installation directory. There will be makefiles in each of the source (`src`) subdirectories, in the three second-level subdirectories, and in the top-level installation directory.
2. Configure the makefiles
   Edit each of the four makefiles to set the particular options for the Fortran compiler and link editor to be used. Decide whether or not to enable run-time debugging and set the compiler flag appropriately. The desired locations for the executable files also needs to be specified. The

defaults are the three directories listed under BINDIR below.

Makefile macros that often require customization:

SYSTEM_NAME the name of the computer system where this installation is being done.

BINDIR the installation directory for the executable modules (defaults are `hst`, `dimen`, and `bcflow` for the three programs). Write permission is needed for this directory.

MANDIR the installation directory for the documentation (default is `doc`). Write permission is needed for this directory.

FDEBUG a flag for the FORTRAN-77 compile command used to specify debug compilation (usually `-g`).

FOPT a flag for the FORTRAN-77 compile command used to specify the level of optimization.

FFLAGS a set of flags specifying options for the FORTRAN-77 compiler.

LD_FFLAGS a set of flags specifying options for the link editor. For example, `-s` means strip the executable files of their symbol tables and relocation bits to reduce file size.

Makefile macros that do not usually require change:

F77 the FORTRAN-77 compile command.

LD the load or link-edit command.

3. Compile and install HST3D
   From the top-level installation directory, initiate the compilation and installation by typing, **`make Everything`**.
4. Test the installation
   After the installation process has finished, the simulator can be tested using the examples located in the `hst/examples` subdirectory. It is recommended that each example be run from its own subdirectory so the many output files are kept together and not commingled with output from other examples. The user needs to create the appropriate subdirectory for each example to be run.

C.4 ALTERNATIVE INSTALLATION PROCEDURES

The standard installation process, initiated by **`make Everything`**, causes all old object code to be removed, installation directories to be created, then compilation to be performed, and, finally, executables to be linked and installed. If an error occurs during the process, the alternative installation procedures allow for the re-use of work that has been accomplished. Using **`make all`** causes compilation to continue while retaining the old object files. Using **`make install`** causes objects to be link edited and then executable binary files to be moved to their destinations. Using **`make All`** is the same as **`make all`** followed by **`make install`**.

APPENDIX D — REPORTING BUGS

The goal is to make HST3D as reliable and bug free as possible. However, no computer code of the size and complexity of HST3D is perfect. When a problem is encountered, the manner of reporting it has a significant role in how well and how quickly it can be fixed. This appendix should help the user to report any bug in a way that will help it be identified and corrected rapidly.

D.1 PROVIDING COMPUTER AND SOFTWARE INFORMATION

Sometimes, a bug only occurs when running on a certain computer or operating system or a certain release of HST3D. Thus, the following information is needed when reporting a bug:

- The version of HST3D.
- The computer hardware model.
- The operating system and version.

D.2 VERIFYING THE BUG

Before reporting a bug, always verify that it is indeed a bug. Double check the manual or with a local colleague familiar with the simulator. If the problem is related to the documentation, the manual may need improvement. Please report problems of vagueness and incompleteness. One question to ask is whether the problem lies within HST3D, or with the computer system running it. Check for sufficient available memory and disc space. Verify that adequate array dimensioning has been provided by running the DIMEN pre-processing program with the problematic data file.

D.3 REPRODUCING THE PROBLEM

Intermittent bugs are by far the most difficult to identify and fix. In general, if the problem can not be recreated here at the U.S. Geological Survey on a local machine with a local version of the simulator, it may take a long time to identify.

D.4 DESCRIBING THE PROBLEM

When describing the problem, it is important to use precise language. Vague terms like "crashes" or "fails to converge" are open to several interpretations. The use of clear descriptions about what happened, what messages were produced, and what was going on when the problem occurred help to locate possible causes. If applicable, please describe the nature of any modifications made to the source code of the simulator.

D.5 SIMPLIFYING THE PROBLEM

It is extremely helpful to be able to recreate the problem with a short, simple data set that does not contain a lot of extraneous data that do not pertain to the problem. A small data set greatly accelerates finding and subsequent fixing the problem. Eliminate as much data as possible that is not involved with demonstrating the bug. Do this in stages, checking to see that the bug remains after each stage. Save the intermediate data files, because sometimes it will be necessary to go back a step and take another route toward simplification. When possible, use a smaller simulation region than the original. Identifying the problem is more complicated and takes more time if the data for the original, large, heterogeneous region are used for debugging. Turn off the logical switches that produce output that clearly is not relevant to the bug. It is possible to eliminate too much output and loose critical information, but, if the problem has

been simplified, the cost of a rerun is not great. The end result of such simplification should be a small data file that demonstrates the problem.

D.6 CONTACTING FOR HELP

To report a suspected bug or a problem, contact the U.S.G.S. in one of the following ways:

Mail
HST3D Code Custodian
U.S. Geological Survey
Water Resources Division
P.O. Box 25046, Mail Stop 413
Denver Federal Center
Denver, Colorado, USA 80225

Telephone
(303) 236-4991
(303) 236-5034 Fax

Electronic Mail
klkipp@usgs.gov

D.7 SENDING DATA WITH THE PROBLEM REPORT

In many cases, it is necessary to send in a data set on magnetic media or via electronic mail. However, many e-mailers truncate or wrap lines longer than 70 characters. In this case, it is better to provide an ftp address and subdirectory location from where the problematic data file can be retrieved. If local modifications have been made to the source code, it may be necessary to provide the revised subroutines. Please telephone or send fax or e-mail for further instructions.

APPENDIX E — USER REGISTRATION

If you wish to register as a user of HST3D, please return or fax this page to:

HST3D Code Custodian
U.S. Geological Survey
Water Resources Division
P.O. Box 25046, Mail Stop 413
Denver Federal Center
Denver, Colorado, USA 80225

fax: (303) 236-5034

or send e-mail to:

klkipp@usgs.gov

with your:

Name

Organization

Address

E-mail address

Date

You will be notified of new releases of the programs, and revisions to the documentation.