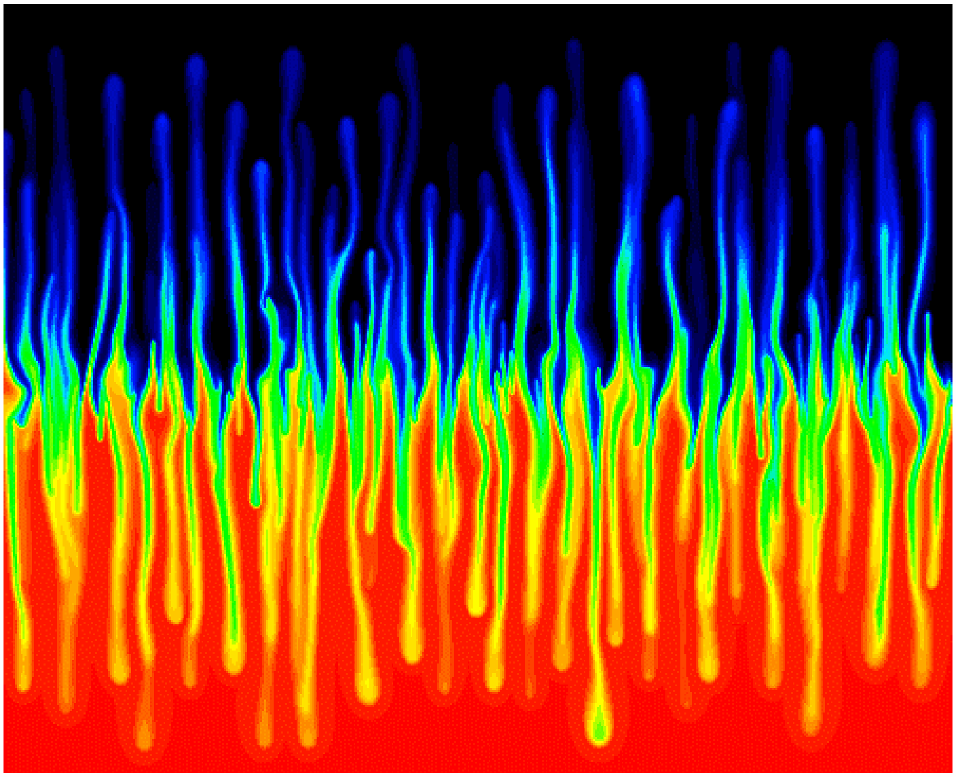
SUTRA—MS
A Version of SUTRA Modified to Simulate Heat and Multiple-Solute Transport

By Joseph D. Hughes and Ward E. Sanford



Open-File Report 2004-1207

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

Cover: Simulated double-diffusive convection using SUTRA-MS

SUTRA—MS
A Version of SUTRA Modified to Simulate Heat and Multiple-Solute Transport

By Joseph D. Hughes and Ward E. Sanford

Open-File Report 2004-1207

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

U.S. Department of the Interior
Gale A. Norton, Secretary

U.S. Geological Survey
Charles G. Groat, Director

U.S. Geological Survey, Reston, Virginia 2004

For product and ordering information:
World Wide Web: http://www.usgs.gov/pubprod
Telephone: 1-888-ASK-USGS

For more information on the USGS—the Federal source for science about the Earth,
its natural and living resources, natural hazards, and the environment:
World Wide Web: http://www.usgs.gov
Telephone: 1-888-ASK-USGS

Preface

This report describes a complex computer model for analysis of fluid flow and multi-species and/or energy transport in subsurface systems. The user is cautioned that although the model will accurately reproduce the physics of flow and transport when used with proper discretization, it will give meaningful results only for well-posed problems based on sufficient supporting data.

This modified version of SUTRA has not been extensively beta tested beyond the example problems given in this report. Please report any problems to:

SUTRA-MS Support
U.S. Geological Survey
431 National Center
Reston, Virginia 20192
USA

Copies of the computer program are available free of charge from the U.S. Geological Survey's web site:

http://water.usgs.gov/nrp/gwsoftware/sutra.html

Abstract

A modified version of SUTRA is introduced that is capable of simulating variable-density flow and transport of heat and multiple dissolved species through variably to fully saturated porous media. The original version of SUTRA is capable or simulating variable-density flow and transport of either heat or one dissolved species through variably to fully saturated porous media. This modified version was developed because of the importance of temperature and solute concentrations in many variable-density flow environments and the desire to implicitly simulate the transport of multiple dissolved species that may or may not affect fluid density. Users familiar with SUTRA should have little difficulty applying this version of SUTRA to multi-species transport problems.

All modifications to SUTRA are general, the number of dissolved species that can be simulated is unrestricted by the program, any of the simulated species can affect fluid density and or viscosity, and simulation of heat transport is unrestricted by the number of simulated dissolved species. The model assumes density and viscosity are linear functions of solute concentration. For simulation of energy transport, the model assumes density is a linear function of temperature but the relation between temperature and viscosity is non-linear. A limitation of the current temperature-viscosity relationship is that it can be scaled only with user-specified parameters unless the code is modified and recompiled. However, alternative temperature-viscosity relationships can be easily incorporated in the source code. In addition to modifications that allow for multi-species transport, SUTRA has been modified to allow use of a spatially distributed solid matrix thermal conductivity and the option to use a geometric-mean approximation for bulk thermal conductivity that accounts for partial saturation. The geometric-mean approximation for bulk thermal conductivity was added because of supporting empirical evidence (Sass and others, 1971).

In addition to modifications to the numerical algorithms in SUTRA, a number of optional functions have been added to minimize solution non-convergence, minimize user coding to simulate time-varying boundary conditions, reduce output file size, allow specification of hydraulic parameters using zones, and specify observation locations with spatial coordinates. A simplified automatic time-step algorithm has been added that reduces the time-step length, if the number of iterations exceeds a user-specified value, and can rerun time steps if the specified maximum number of iterations is exceeded. A simple algorithm has been added that allows any boundary condition to be time varying without the need for user-programmed functions, with the limitation that boundary conditions between the times when conditions are changed (at the beginning of stress periods) are not interpolated. Time steps are reduced to the minimum time step when new boundary-condition values are read. A simple technique has been implemented that permits the user to specify the exact time to write data to the nodal and elemental output files. Functionality has been added that allows hydraulic parameters to be specified using zones in order to reduce the memory requirements for large two- and three-dimensional simulations and to better facilitate the use of model-independent parameter-estimation codes (*e.g.,* UCODE, Poeter and Hill, 1998; PEST, Doherty, 1994). A simple routine that

allows observation locations to be specified using coordinates rather than node numbers also has been included.

The modified version of SUTRA has retained all the functionality of the original SUTRA and can solve the flow and transport equations in either two or three dimensions. This version of SUTRA is backward compatible with standard two- and three-dimensional SUTRA single species input data sets. Multi-species data sets are different from standard data sets only when additional data are required for each simulated species.

Three examples are presented that demonstrate the ability of the modified version of SUTRA to simulate multi-species transport. Two of these examples demonstrate the ability to simulate fluid density affected by more than one species, and one example shows several additional applications of the modified version of SUTRA. Where possible, the modified code is compared to observed data, the standard version of SUTRA (Voss and Provost, 2002), or other codes capable of simulating variable-density flow and heat and solute transport. Comparisons indicate the modified version of SUTRA is comparable to HST3D (Kipp, 1997) for a coupled heat and solute variable-density-flow problem, and is capable of simulating a coupled variable-density multi-species Hele-Shaw experiment.

Contents

Figures

Tables

1.0 Introduction

Ground water in coastal environments, areas of high evaporation rates, and urban and industrial areas may have dissolved constituents that can affect fluid density. For example, in coastal aquifers, chloride concentrations can vary from 0 to 18,500 parts per thousands and have fluid densities ranging from 1,000 to 1,025 kg/m^3 (kilograms per cubic meter) respectively. The range in fluid density of chloride concentrations from freshwater to seawater represents only a 2.5-percent increase in fluid density, but this small difference has been shown to have significant effects on ground-water-flow rates and patterns. Typically, multiple dissolved constituents are present in variable-density flow environments, but density variations usually can be related to a single constituent (*i.e.,* chloride concentrations, total dissolved solids.) because of the positive correlation of each constituent with all the other constituents.

Fluid temperature variations also affect fluid density. A 10°C increase in fluid temperature of freshwater results in a reduction in fluid density of approximately 4 kg/m^3 (~0.5-percent reduction). The small change in fluid density over the range of fluid temperatures found in many shallow coastal systems allows temperature effects on fluid density to be ignored (Konikow and Reilly, 1999). In some natural systems, however, temperature variations have a significant effect on the flow system (*e.g.,* hydrothermal systems, thick and/or deep aquifers). Similarly, fluid density may be a function of more than one dissolved species in some natural systems (Pringle and others, 2002). Furthermore, lateral and vertical aquifer heterogeneities in many natural systems cause the variable-density-flow system to be three dimensional in nature.

A variety of numerical approaches have been used to simulate variable-density-flow problems (for a summary of available methods, see Sorak and Pinder, 1999). The methods that have been used include finite differences (*e.g.,* INTERA, 1979), finite elements (*e.g.,* Voss, 1984; Huyakorn and others, 1987; and Diersch, 1988), finite volumes (*e.g.,* Kipp, 1987; Oldenburg and Pruess, 2000), analytical elements (Strack, 1995), and hybrid Eulerian-Lagrangian finite differences/finite elements (*e.g.,* Sanford and Konikow, 1985; Yeh, 1987 and 1990). These approaches have included two-dimensional and three-dimensional implementations of the available methods.

This report describes a modified version of SUTRA capable of simulating variable-density flow that is dependent on multiple dissolved constituents and temperature. SUTRA was selected for modification because of its wide use and modular nature that easily accommodates additional functionality.

1.1 Purpose

The principal objectives of this report are to (1) introduce a general methodology that incorporates multiple dissolved species into the set of partial differential equations that describe variable-density flow and solute transport, (2) present a numerical model, SUTRA-MS (Saturated-Unsaturated Transport of Multiple Species), that utilizes this methodology; (3) benchmark the computer code against existing numerical codes, previous numerical solutions, and experimental data; and (4) use SUTRA-MS results to demonstrate some of the processes that are important in the simultaneous transport of multiple species.

1.2 SUTRA Fundamentals and Previous Applications

SUTRA uses a hybridized finite element and implicit finite-difference technique to solve the fluid mass-balance equation and unified energy- and solute- balance equation for variable-density, single-phase, saturated-unsaturated flow and single-species transport. Solving the equations in the time domain is accomplished by the implicit finite-difference method for non-flux terms (*e.g.,* time derivatives and sources) in the region surrounding each specified node. All flux terms and associated parameters are discretized on an element basis and are solved by a modified Galerkin method. SUTRA uses bilinear quadrilateral elements in two dimensions and trilinear hexahedrons in three dimensions.

SUTRA uses a modified form of the standard Galerkin finite-element method to allow for calculation of a consistent vertical velocity for each element. This is a significant but important modification for variable-density-flow simulations. In the standard Galerkin finite-element method, spurious vertical velocities are generated everywhere there is a vertical concentration gradient within the finite-element mesh, even if a hydrostatic pressure distribution is used (Voss, 1984; Voss and Souza, 1987). These spurious vertical velocities make it impossible to simulate a narrow transition zone between fluids with significant density differences (*e.g.,* freshwater and seawater) with the standard finite-element method, regardless of the specified dispersivity for the system. The modification used in SUTRA provides for a vertical velocity calculation within each finite element based on consistent spatial variability of the pressure gradient, ∇p, and the buoyancy term, $\rho \mathbf{g}$, in the variable-density form of Darcy's equation.

SUTRA is a very general code that was developed using a modular coding convention (Voss, 1999). The modular nature of SUTRA has allowed it to be modified for specific projects and applications. Some of the modifications have included addition of equilibrium chemical reactions (Lewis and others, 1986), addition of kinetic reactions (e.g., Smith and others, 1997; Sahoo and others, 1998), and calculation of path lines (Cordes and Kinzelbach, 1992).

SUTRA was initially released as a two-dimensional code that could solve constant-density areal problems or constant- to variable-density cross-sectional problems. A 3D version of SUTRA was developed and tested in 1982. After running several test problems with the 3D version of SUTRA, the three-dimensional simulation capabilities were removed because of the limited computer resources that most users had access to in the early 1980s. Three-dimensional problems that were sufficiently discretized required access to super-computer facilities and had limited use to a small number of users (Voss, 1999). The decision to release a 2D version of SUTRA was appropriate and has allowed SUTRA to be widely used and applied to a variety of problems.

The 2D version of SUTRA has been applied to a number of problems including: (1) unsaturated flow studies; (2) contaminant/tracer studies; (3) energy-transport studies; (4) inverse modeling and optimization studies; (5) solute-dependent variable-density flow; and (6) temperature- dependent variable-density flow. A summary of published SUTRA applications is given in Voss (1999).

Recent improvements in computer processors and associated hardware have allowed an updated and improved 3D version of SUTRA to be released (Voss and Provost, 2002). Advances in techniques to solve numerical problems have allowed incorporation of robust, iterative solvers in the 3D version of SUTRA and made developing and solving 3D variable-density-flow problems practical. The 3D version of SUTRA retains all the original functionality of the 2D version and has been applied successfully to contaminant-transport problems and variable-density-flow problems.

1.3 The Current Study

This report outlines the development of and some applications for a modified version of SUTRA: SUTRA-MS. The principal modification in SUTRA-MS is an extension of existing numerical methods to solve for the transport of multiple species and allow for dependence of density and viscosity on any of the simulated species. All the features of the original version of SUTRA have been retained in SUTRA-MS, including variably saturated to fully saturated flow, advection, and production and decay of simulated species.

Other modifications include:

- capability to simulate a spatially varying solid matrix thermal conductivity,
- capability to use either a volumetric average or geometric-mean approximation for bulk thermal conductivity,
- ability to simulate simple time-varying boundary conditions that do not need to be user programmed,
- capability to output nodal- and element-simulation data at user-specified times,
- ability to use a simplified automatic time-stepping algorithm to adjust time steps based on solution convergence, including the capability to rerun a time step if user-specified parameters are not achieved,
- ability to enter observation locations using spatial coordinates instead of node numbers,
- capability to specify hydraulic parameters using zones in order to reduce memory requirements for large two- and three-dimensional problems and facilitate the use of generic parameter-estimation codes, and
- use of Fortran 95 intrinsic functions or programming standards where beneficial to program execution times, maintainability, or source-code clarity.

The current modifications do not place an arbitrary limit on the number of species that can be simulated. The number of species that can be simulated is a function of the problem size and the amount of available random access memory (RAM). Furthermore, the modifications are such that more complicated relationships between solute concentration and system state (*e.g.,* density, *etc.*) can be easily incorporated.

2.0 Physical-Mathematical Basis of SUTRA-MS Modifications

This section summarizes the governing equations used in SUTRA to simulate fluid flow and solute transport in constant-density and variable-density environments. The modifications to the original equations are summarized here to provide background for the numerical implementation discussed in Section 3. To allow for simulation of multi-species transport with possible fluid-density and viscosity dependence on solute concentrations and temperature, all equations that include fluid density, fluid viscosity, velocity-dependent dispersion, molecular dispersion, thermal conductivity, solute adsorption, solute production and decay, solute or temperature boundary conditions, solute concentrations, or temperature require modification. Notations used in the equations are explained after their first use and also are summarized in appendix 1.

2.1 General Mass-Balance Formulation

SUTRA-MS is a modified version of SUTRA capable of simulating unsaturated to saturated, variable-density fluid flow dependent on the transport of heat and multiple dissolved species. Because the modifications required to allow SUTRA to simulate transport of more than one species, including temperature, may affect both the flow and transport equations solved by SUTRA, a brief overview of the general mass-balance equations used in SUTRA is given below. A more complete discussion of SUTRA fundamentals is given in Voss and Provost (2002).

2.1.1 Fluid Mass-Balance Equation

The fluid mass-balance equation, which is usually referred to as the ground-water-flow equation, is

$$(S_w \rho S_{op} + \varepsilon \frac{\partial S_w}{\partial p})\frac{\partial p}{\partial t} + (\varepsilon S_w \frac{\partial p}{\partial U})\frac{\partial U}{\partial t} - \nabla \cdot \left[\left(\frac{\mathbf{k} k_r}{\mu} \right) \cdot \left(\nabla p - \rho \mathbf{g} \right) \right] = Q_p \qquad \textbf{Eq. 2.1}$$

where

S_w is the water saturation [-],
ε is the porosity [-],
p is the fluid pressure [M/LT^2],
t is time [T],
U is either temperature [°C] or solute mass fraction [M$_{\text{solute}}$/M$_{\text{fluid}}$],
$\mathbf{k}$ is the permeability tensor [L^2],
k_r is the relative permeability for unsaturated flow [-],
μ is the fluid viscosity [M/LT],
ρ is the fluid density [M/L^3],
$\mathbf{g}$ is the gravitational vector [L/T^2],
Q_p is a fluid mass source [M/L^{3}T], and
S_{op} is the specific pressure storativity [M/LT^2]$^{-1}$.

Specific pressure storativity, S_{op}, is defined as

$$S_{op} = (1-\varepsilon)\alpha + \varepsilon\beta \qquad \textbf{Eq. 2.2}$$

where

α is the compressibility of the porous matrix $[M/LT^2]^{-1}$, and
β is the compressibility of the fluid $[M/LT^2]^{-1}$.

2.1.2 Modified Form of the Fluid Mass- Balance Equation

In SUTRA-MS, the generalized fluid mass-balance equation (eq. 2.1) has been modified to allow for more than one species to affect fluid density. The modified fluid mass-balance equation is

$$(S_w \rho S_{op} + \varepsilon \frac{\partial S_w}{\partial p})\frac{\partial p}{\partial t} + \varepsilon S_w \left(\sum_{k=1}^{NS} \frac{\partial p}{\partial C_k}\frac{\partial C_k}{\partial t} + \tau \frac{\partial p}{\partial T}\frac{\partial T}{\partial t} \right)$$
$$-\nabla \cdot \left[\left(\frac{\mathbf{k}k_r}{\mu} \right) \cdot (\nabla p - \rho \mathbf{g}) \right] = Q_p \qquad \textbf{Eq. 2.3}$$

where

NS is the number of dissolved species simulated,
$\tau = 1$ if heat transport is being simulated and $\tau = 0$ if otherwise,
C_k is the solute concentration of species k $[M_{solute}/M_{fluid}]$, and
T is the temperature of the solution [°C].

The modified form of the fluid density, ρ, equation used in SUTRA-MS is

$$\rho = \rho_o + \sum_{k=1}^{NS} \frac{\partial \rho}{\partial C_k}(C_k - C_{ko}) + \tau \frac{\partial \rho}{\partial T}(T - T_o) \qquad \textbf{Eq. 2.4}$$

where

ρ_o is the fluid density at the base concentration of all simulated species at the base solute concentrations and temperature $[M_{solute}/L^3_{solute}]$,
C_{ko} is the base solute mass fraction of species k $[M_{solute}/M_{fluid}]$, and
T_o is the base temperature [°C].

Equation 2.4 assumes $\partial\rho/\partial C_k$ and $\partial\rho/\partial T$ are constant over the range of simulated mass fractions, and temperatures and density relationships are linear and additive. This is an ad-hoc relationship but it is consistent with relationships used by others for similar applications (*e.g.,* Kipp, 1987). This *ad-hoc* relationship should be evaluated prior to use and modified if it is determined to be inappropriate for a particular application.

The modified form of the fluid viscosity, μ, used in SUTRA-MS is

$$\mu = \mu_o(T) + \sum_{k=1}^{NS} \frac{\partial \mu}{\partial C_k}\left(C_k - C_{ko}\right) \qquad \textbf{Eq. 2.5}$$

where

$\mu_o(T)=$ a user-defined base fluid viscosity for isothermal flow at the base mass fraction for each species [M/LT], or $\left(239.4x10^{-7}\right)10^{\left(\frac{248.37}{T+133.15}\right)}$ for temperature-dependent flow [M/LT].

If temperature-dependent flow is simulated, the units of $\mu_o(T)$ can be converted to those desired using a scaling factor in the program input data. Equation 2.5 assumes $\partial\mu/\partial C_k$ is constant over the range of simulated mass fractions, the specified non-linear viscosity-temperature relationship is appropriate for the range of simulated temperatures, and viscosity mass-fraction relationships are linear and are not affected by species or temperature interaction. This is an *ad-hoc* relationship based on the concentration-density relationship described above. This *ad-hoc* relationship should be evaluated prior to use and modified if it is determined to be inappropriate for a particular application.

2.1.3 Unified Energy- and Solute- Balance Equation

SUTRA uses a unified energy- and solute-balance equation to solve both energy and solute transport. A unified energy-solute balance is possible because of the similarity of the mechanisms affecting the fluxes of energy and solute mass in solution when energy and solute transport equations are formulated in terms of energy per unit volume and total species mass per unit volume, respectively. In fact, of the transport processes represented in SUTRA, only non-linear sorption processes and first-order production of solute and adsorbate mass have no analogy in energy transport.

A mass-conservative form of the unified balance equation is solved by SUTRA and is derived from a general form of the equation in Voss and Provost (2002). The derivation of the mass-conservative form of the unified balance equation involves removing terms accounted for in the fluid mass-balance equation (fluid saturation and pressure-change contribution to energy and solute balances) and terms related to change in the solid matrix density with time. The reader is referred to Voss and Provost (2002) for a more complete description of the derivation of the mass-conservative form of the unified balance equation. The mass-conservative form of the unified balance equation is

$$\begin{aligned}
&\left[\varepsilon S_w \rho c_w + (1-\varepsilon)\rho_s c_s\right]\frac{\partial U}{\partial t} + \varepsilon S_w \rho c_w \mathbf{v}\cdot\nabla U \\
&-\nabla\cdot\left\{\rho c_w\left[\varepsilon S_w\left(\sigma_w \mathbf{I} + \mathbf{D}\right) + (1-\varepsilon)\sigma_s \mathbf{I}\right]\cdot\nabla U\right\} \qquad \textbf{Eq. 2.6} \\
&= Q_p c_w\left(U^* - U\right) + \varepsilon S_w \rho \gamma_1^w U + (1-\varepsilon)\rho_s \gamma_1^s U_s + \varepsilon S_w \rho \gamma_o^w + (1-\varepsilon)\rho_s \gamma_o^s
\end{aligned}$$

where

c_w is the fluid specific heat [E/M°C] for heat transport or one (1) for solute transport,

ρ_s is the density of the solid matrix [M/L^3],

c_s is solid matrix specific heat [E/M°C] for heat transport or the sorption coefficient [-] for solute transport,

v is the fluid velocity [L/T],

σ_w is the fluid thermal diffusivity [L^2/T] for heat transport or molecular diffusivity (D_m) for solute transport [L^2/T],

I is the identity tensor,

D is the dispersion tensor [L^2/T] for heat or solute transport,

σ_s is the solid matrix thermal diffusivity [L^2/T] for heat transport or zero (0) for solute transport,

U^* is the temperature of the source fluid [°C] for heat transport or the solute concentration of the source fluid [M solute/M fluid] for solute transport,

γ_1^w is zero (0) for heat transport or the first-order mass-production rate of solute [T^{-1}] for solute transport,

γ_1^s is zero (0) for heat transport or the first-order mass-production rate of adsorbate [T^{-1}] for solute transport,

U_s is the specific energy content of the solid matrix [E/M] for heat transport (but does not contribute to the unified balance equation because γ_1^s is zero (0)) or the concentration of the adsorbate on the solid matrix (C_s) for solute transport [M adsorbate/M solid matrix],

γ_0^w is the zero-order energy-production rate in the fluid [E/MT] for heat transport or the zero-order mass-production rate of solute [(M solute/M fluid)T^{-1}] for solute transport, and

γ_0^s is the zero-order energy-production rate in the solid matrix [E/MT] for heat transport or the zero-order mass-production rate of adsorbate [(M adsorbate/M solid matrix and adsorbate)T^{-1}] for solute transport.

Fluid thermal diffusivity, σ_w, in equation. 2.6 is defined as

$$\sigma_w = \frac{\lambda_w}{\rho c_w} \qquad \textbf{Eq. 2.7}$$

where

λ_w is the fluid thermal conductivity [E/TL°C].

Solid matrix thermal diffusivity, σ_s, in equationq. 2.6 is defined as

$$\sigma_s = \frac{\lambda_s}{\rho c_w} \qquad \textbf{Eq. 2.8}$$

where

λ_s is the solid thermal conductivity [E/TL°C].

Three equilibrium sorption models, which specify c_s in equation. 2.6, are possible in SUTRA. These three sorption models assume fluid density is constant. The three sorption models are

Linear equilibrium sorption model

$$U_s = (\chi_1 \rho_o) U \qquad \text{Eq. 2.9a}$$

$$\frac{\partial U_s}{\partial t} = (\chi_1 \rho_o) \frac{\partial U}{\partial t} \qquad \text{Eq. 2.9b}$$

$$c_s = \kappa_1 = \chi_1 \rho_o \qquad \text{Eq. 2.9c}$$

where

χ_1 is the linear distribution coefficient [L^3 fluid/M solid and adsorbate], and

κ_1 is the first general sorption coefficient.

Freundlich equilibrium sorption model

$$U_s = \chi_1 (\rho_o U)^{\left(\frac{1}{\chi 2}\right)} \qquad \text{Eq. 2.10a}$$

$$\frac{\partial U_s}{dt} c_s = \left(\frac{\chi_1}{\chi_2}\right) (\rho_o U)^{\left(\frac{1-\chi 2}{\chi 2}\right)} \rho_o \frac{\partial U}{\partial t} \qquad \text{Eq. 2.10 b}$$

$$c_s = \kappa_1 = \left(\frac{\chi_1}{\chi_2}\right) \rho_o^{\left(\frac{1}{\chi 2}\right)} U^{\left(\frac{1-\chi 2}{\chi 2}\right)} \qquad \text{Eq. 2.10c}$$

where

χ_1 is a Freundlich distribution coefficient [L^3 fluid/M solid and adsorbate], and

χ_2 is the Freundlich coefficient [-].

Langmuir equilibrium sorption model

$$U_s = \frac{\chi_1 (\rho_o U)}{1 + \chi_2 (\rho_o U)} \qquad \text{Eq. 2.11a}$$

$$\frac{\partial U_s}{\partial t} = \frac{\chi_1 \rho_o}{(1 + \chi_2 \rho_o U)^2} \frac{\partial U}{\partial t} \qquad \text{Eq. 2.11b}$$

$$c_s = \kappa_1 = \frac{\chi_1 \rho_o}{\left(1 + \chi_2 \rho_o U\right)^2} \qquad \textbf{Eq. 2.11 c}$$

where

χ_1 is a Langmuir distribution coefficient [L^3 fluid/M solid and adsorbate], and

χ_2 is the Langmuir coefficient [-].

2.1.4 Modified Form of the Unified Energy- and Solute- Balance Equation

In SUTRA-MS, the mass-conservative form of the unified balance equation (eq. 2.6) has been modified to allow for simultaneous, sequential simulation of more than one species. The three forms of the sorption equations (eqs. 2.9, 2.10, and 2.11) also have been modified to allow simulation of multiple species, each of which can have different adsorption isotherms.

The modified mass-conservative form of the unified balance equation is

$$\begin{aligned}
&\left[\varepsilon S_w \rho c_{wk} + (1-\varepsilon)\rho_s c_{sk}\right]\frac{\partial U_k}{\partial t} + \varepsilon S_w \rho c_{wk} \mathbf{v} \cdot \nabla U_k \\
&- \nabla \cdot \left\{\rho c_{wk}\left[\varepsilon S_w\left(\sigma_{wk}\mathbf{I} + \mathbf{D}_k\right) + (1-\varepsilon)\sigma_{sk}\mathbf{I}\right] \cdot \nabla U_k\right\} \\
&= Q_p c_{wk}\left(U_k^* - U_k\right) + \varepsilon S_w \rho \left(\gamma_1^w\right)_k U_k + (1-\varepsilon)\rho_s\left(\gamma_1^s\right)_k U_{sk} \\
&+ \varepsilon S_w \rho \left(\gamma_o^w\right)_k + (1-\varepsilon)\rho_s \left(\gamma_o^s\right)_k
\end{aligned} \qquad \textbf{Eq. 2.12}$$

where

c_{wk} is the fluid specific heat [E/M°C] for heat transport if species k is "ENERGY" or one (1) for solute transport of species k,

c_{sk} is the solid matrix specific heat [E/M°C] for heat transport if species k is "ENERGY" or the sorption coefficient [-] for solute transport of species k ,

U_k is either fluid temperature [°C] if species k is "ENERGY" or concentration [M solute/M fluid] for solute transport of species k,

$\mathbf{D}_k$ is the thermal-dispersion tensor [L^2/T] if species k is "ENERGY" or the solute transport dispersion tensor or [L^2/T] of species k,

σ_{sk} is the solid matrix thermal diffusivity [L^2/T] if species k is "ENERGY" or zero (0) for solute transport of species k,

σ_{wk} is the fluid thermal diffusivity [L^2/T] if species k is "ENERGY" or molecular diffusivity, D_M, [L^2/T] of species k,

U_k^* is the temperature of the source fluid [°C] if species k is "ENERGY" or the solute concentration of the source fluid [M solute/M fluid] for species k,

$\left(\gamma_1^w\right)_k$ is zero (0) if species k is "ENERGY" or the first-order mass-production rate of species k solute [T^{-1}],

$\left(\gamma_1^s\right)_k$ is zero (0) if species k is “ENERGY” or the first-order mass-production rate of species k adsorbate [T^{-1}],

U_{sk} is the specific energy content of the solid matrix [E/M] if species k is “ENERGY” (does not contribute to the unified balance equation because $\left(\gamma_1^s\right)_n$ is zero (0)) or the concentration of species k adsorbate on the solid matrix [M solute/M solid matrix],

$\left(\gamma_0^w\right)_k$ is the zero-order energy-production rate in the fluid [E/MT] if species k is “ENERGY” or zero-order mass-production rate of species k solute [(M solute/M fluid)T^{-1}], and

$\left(\gamma_0^s\right)_k$ is the zero-order energy-production rate in the solid matrix [E/MT] if species k is “ENERGY” or zero-order mass-production rate of species k adsorbate [(M adsorbate/M solid matrix and adsorbate)T^{-1}].

The modified forms of the three equilibrium models, which specify c_{sk} in equation . 2.12, are

Linear equilibrium sorption model

$$c_{sk} = \kappa_{1k} = \chi_{1k}\rho_o$$

Eq. 2.13

where

χ_{1k} is the linear distribution coefficient [L^3 fluid/M solid and adsorbate] of species k, and

κ_{1k} is the first general sorption coefficient of species k.

Freundlich equilibrium sorption model

$$c_{sk} = \kappa_{1k} = \left(\frac{\chi_{1k}}{\chi_{2k}}\right)\rho_o^{\left(\frac{1}{\chi_{2k}}\right)} U_k^{\left(\frac{1-\chi_{2k}}{\chi_{2k}}\right)}$$

Eq. 2.14

where

χ_{1k} is a Freundlich distribution coefficient [L^3 fluid/M solid and adsorbate] for species k, and

χ_{2k} is the Freundlich coefficient [-] for species k.

Langmuir equilibrium sorption model

$$c_{sk} = \kappa_{1k} = \frac{\chi_{1k}\rho_o}{\left(1 + \chi_{2k}\rho_o U_k\right)^2}$$

Eq. 2.15

where

χ_{1k} is a Langmuir distribution coefficient [L^3 fluid/M solid and adsorbate] for species k, and

χ_{2k} is the Langmuir coefficient [-] for species k.

Equations 2.13 through 2.15 also assume fluid density, ρ_o, is constant and aqueous reactions among the simulated species are limited and can be ignored. The appropriateness of this assumption should be evaluated before using SUTRA-MS for a particular application. Although sorption of multiple species has been implemented in a simple fashion, SUTRA-MS is modular and a more sophisticated method of sorption could be implemented in the current program if required. More sophisticated sorption methods that account for aqueous interactions among dissolved species that also adsorb to the solid matrix have been implemented in a constant-density version of SUTRA, SATRA-CHEM (Lewis and others, 1986), and could be used as a template for modifying SUTRA-MS.

2.1.5 Modified Form of the Energy- Balance Equation used with Geometric-Mean Approximation for Bulk Thermal Conductivity

The thermal conductivity of upper crustal materials generally varies by less than a factor of 5 whereas permeability can vary over 16 orders of magnitude (Ingebritsen and Sanford, 1998). SUTRA uses a volumetric average approximation for bulk thermal conductivity (eq. 2.16).

$$\lambda \equiv \varepsilon S_w \lambda_w + (1-\varepsilon)\lambda_s \qquad \textbf{Eq. 2.16}$$

where

λ is the bulk thermal conductivity.

Empirical evidence shows that bulk thermal conductivity is well modeled using the geometric mean of the rock conductivity and fluid conductivity (Sass and others, 1971). The geometric-mean approximation of bulk thermal conductivity modified for variably saturated media is

$$\lambda \equiv \left(\lambda_A{}^{(1-Sw)} \lambda_w{}^{Sw}\right)^{\varepsilon} \lambda_s^{(1-\varepsilon)} \qquad \textbf{Eq. 2.17}$$

where

λ_A is the thermal conductivity of air.

The thermal conductivity of air is a function of temperature but varies by less that 15percent (0.025 to 0.027 W/m°C) over typical appropriate temperature ranges for SUTRA-MS simulations (between 5 and 50°C). Because the thermal conductivity of air is approximately 1 to 2 orders of magnitude less that the thermal conductivity of water and geologic materials, respectively, a constant value of 0.026 W/m°C is used in bulk thermal-conductivity calculations when the geometric-mean approximation is used.

Use of a geometric-mean approximation for bulk thermal conductivity typically results in a reduction of approximately 15 percent in bulk thermal conductivity from volumetric average bulk thermal conductivities at typical porosities and solid matrix thermal conductivities. The differences in bulk thermal conductivities calculated using the volumetric average and geometric-mean approximations generally increase with increasing porosity and/or decreasing fluid saturation (fig. 2.1.1).

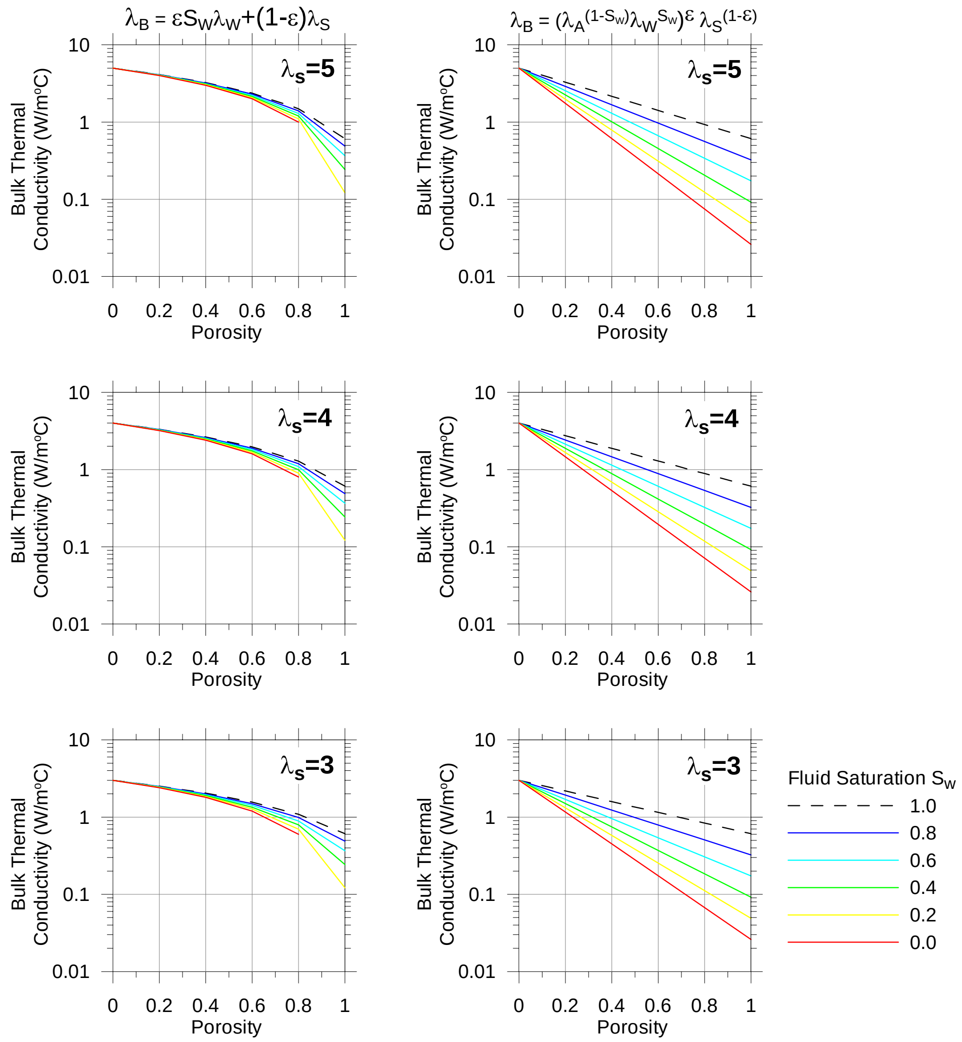


Figure 2.1.1 **Comparison of bulk thermal conductivity as a function of porosity, fluid saturation, and solid matrix thermal conductivity using a volumetric average and geometric-mean approximations.**

When the geometric mean thermal-conductivity approximation is used, bulk thermal diffusivity in equation. 2.6 is defined as

$$\sigma_{bg} = \left(\frac{\left(\lambda_A^{1-S_w}\,\lambda_w^{S_w}\right)^{\varepsilon}\,\lambda_s^{(1-\varepsilon)}}{\rho c_w} \right) \qquad \textbf{Eq. 2.18}$$

where

σ_{bg} is the bulk thermal diffusivity [L^2/T].

Equation 2.12 is modified for energy transport in the following manner when using a geometric-mean approximation for bulk thermal conductivity

$$\begin{aligned}
&\left[\varepsilon S_w \rho c_{wk} + (1-\varepsilon)\rho_s c_{sk}\right]\frac{\partial U_k}{\partial t} + \varepsilon S_w \rho c_{wk}\,\mathbf{v}\cdot\nabla U_k \\
&-\nabla\cdot\left\{\rho c_{wk}\left[\sigma_{bgk}\mathbf{I} + \varepsilon S_w \mathbf{D}_k\right]\cdot\nabla U_k\right\} \\
&= Q_p c_{wk}\left(U_k^* - U_k\right) + \varepsilon S_w \rho\left(\gamma_1^w\right)_k U_k + (1-\varepsilon)\rho_s\left(\gamma_1^s\right)_k U_{sk} \\
&+ \varepsilon S_w \rho\left(\gamma_o^w\right)_k + (1-\varepsilon)\rho_s\left(\gamma_o^s\right)_k
\end{aligned} \qquad \textbf{Eq. 2.19}$$

where

σ_{bgk} is the bulk thermal diffusivity [L^2/T] if species k is “ENERGY” or molecular diffusivity, D_M, [L^2/T] of species

3.0 Numerical Methods

As a result of the modified fluid mass-balance equation (eq. 2.3) and the modified unified energy- and solute--balance equation (eq. 2.12), the discretized, weighted-residual relations approximating these equations have been modified in SUTRA-MS. Complete development of the discretized, weighted residual approximations of equations 2.1 and 2.6 are given in Voss and Provost (2002). Explanation of the notation used in the equations developed in Section 3 is given after its first use in an equation and also is summarized in appendix 1.

3.1 Numerical Approximation of SUTRA-MS Fluid Mass Balance

The modified unified energy- and solute-balance equation (eq. 2.3) is approximated numerically through nodewise, elementwise, and cellwise discretization. The weighted residual relation that allow for solution of pressures at nodes at the end of the present time step is

$$\sum_{j=1}^{NN}\left[\left(\frac{AF_i^{n+1}\delta_{ij}}{\Delta t_{n+1}}\right)+BF_{ij}^{n+1}+\nu_{pi}\delta_{ij}\right]p_j^{n+1}$$

$$=Q_i^{n+1}+\nu_{pi}\,p_{BC_i}^{n+1}+DF_i^{(n+1)*}+\left(\frac{AF_i^{n+1}}{\Delta t_{n+1}}\right)p_i^n+\sum_{k=1}^{NS+\tau}\left[\left(CF_{ik}^{n+1}\right)\left(\frac{dU_{ik}}{dt}\right)^n\right] \quad \mathrm{i}=\overline{1,\mathrm{NN}} \qquad \textbf{Eq. 3.1}$$

where

$$AF_i^{n+1}=\left(S_w\rho S_{op}+\varepsilon\rho\frac{\partial S_w}{\partial p}\right)_i V_i \qquad \textbf{Eq. 3.2}$$

$$CF_{ik}^{n+1}=\left(\varepsilon S_w\frac{\partial\rho}{\partial U_k}\right)_i V_i \qquad \textbf{Eq. 3.3}$$

$$BF_{ij}^{n+1}=\int_x\int_y\int_z\left\{\left[\left\langle\left\langle\mathbf{k}^L\right\rangle\right\rangle\left(\frac{k_r\rho}{\mu}\right)\cdot\nabla\phi_j\right]\right\}\cdot\nabla\omega_i\,dz\,dy\,dx \qquad \textbf{Eq. 3.4}$$

$$DF_i^{(n+1)*}=\int_x\int_y\int_z\left\{\left[\left\langle\left\langle\mathbf{k}^L\right\rangle\right\rangle\left(\frac{k_r\rho}{\mu}\right)\cdot\left\langle\left\langle\rho\mathbf{g}\right\rangle\right\rangle^*\right]\right\}\cdot\nabla\omega_i\,dz\,dy\,dx \qquad \textbf{Eq. 3.5}$$

δ_{ij} is the Kronecker delta:

$$\delta_{ij}=\begin{cases}0 \text{ if } i\neq j\\ 1 \text{ if } i=j\end{cases}$$

Δt_{n+1} is the length of the current time step,

p_i^{n+1} is the pressure in cell i at the end of the current time step,

p_i^n is the pressure in cell i at the end of the previous time step,

Q_i^{n+1} is the total mass source [M/T] to cell i for the current time step,

ν_{pi} is the pressure-based conductance [LT] for the specified pressure source in cell i (zero (0) for all nodes that are not specified pressure nodes),

$p_{BC_i}^{n+1}$ is the specified pressure in cell i,(zero (0) for all nodes that are not specified pressure nodes),

$\left(\frac{dU_{ik}}{dt}\right)^n$ is the rate of change in solute concentration of species k in cell i during the previous time step $\left(\frac{U_{ik}^n - U_{ik}^{n-1}}{\Delta t_n}\right)$,

V_i is the volume of cell i,

ϕ_j is the symmetric bilinear (2D) or trilinear (3D) basis function in global coordinates for node j ,

ω_i is the asymmetric weighting function in global coordinates for node i,

$\langle\langle \mathbf{k}^L \rangle\rangle$ is the permeability tensor [L^2] that is discretized elementwise, and

$\langle\langle \rho\mathbf{g} \rangle\rangle^*$ is an elementwise discretization of ($\rho\mathbf{g}$) that is consistent with the discretization of ∇p.

For two-dimensional simulations, equations. 3.4 and 3.5 are written as

$$BF_{ij} = \int_x \int_y \left\{ \left[\langle\langle \mathbf{k}^L \rangle\rangle \left(\frac{k_r \rho}{\mu} \right) \cdot \nabla \phi_j \right] \right\} \cdot \nabla \omega_i B \, dy \, dx \qquad \textbf{Eq. 3.6}$$

$$DF_i = \int_x \int_y \left\{ \left[\langle\langle \mathbf{k}^L \rangle\rangle \left(\frac{k_r \rho}{\mu} \right) \cdot \langle\langle \rho\mathbf{g} \rangle\rangle^* \right] \right\} \cdot \nabla \omega_i B \, dy \, dx \qquad \textbf{Eq. 3.7}$$

where the thickness of the mesh, *B(x,y)*, is evaluated at each Gauss point according to the following nodewise discretization

$$B(x,y) = \sum_{i=1}^{NN} B_i \phi_i (x,y) \qquad \textbf{Eq. 3.8}$$

The superscript involving (n) or ($n+1$) indicates the value is evaluated at the end of the previous time step and the end of the current time step, respectively.

The only integrals requiring Gaussian integration are BF_{ij} and DF_{ij}. These integrals are evaluated in an element-by-element manner in the SUTRA-MS subroutine ELEMN2 (for

2D) or ELEMEN3 (for 3D). The other terms, except for those involving ν_{pi}, are evaluated cellwise (one for each node) by the SUTRA-MS subroutine NODAL. The specified pressure terms are evaluated by subroutine BC. Equation 3.1 is assembled for each node in the model mesh in a given time step and is solved using either Gaussian elimination or an iterative solver. For the iterative solvers, an approximate solution is achieved when the residual error (summation of eq. 3.1 for each node) is less than a user-specified convergence criteria. Details of the flow and transport solution scheme used in SUTRA-MS is given in Section 3.4.

3.2 Numerical Approximation of SUTRA-MS Unified Energy- and Solute-Balance Equation

The modified unified energy- and solute-balance equation (eq. 2.12) is approximated numerically through nodewise, elementwise, and cellwise discretization. The weighted residual relations that allow for solution of concentration or temperature at nodes at the end of the present time step are

$$\left\{\begin{array}{l} \displaystyle\sum_{j=1}^{NN}\left[\left(\frac{AT_{ik}^{n+1}\delta_{ij}}{\Delta t_{n+1}}\right)+DT_{ijk}^{(n+1)*}+BT_{ijk}^{n+1}+\left[\nu_{U_{ik}}+GT_{ik}^{n+1}+G_s TL_{ik}^{n+1}+\left(Q_i^{n+1}+Q_{BC_i}^{n}\right)c_{wk}\right]\delta_{ij}\right]U_{ik}^{n+1} \\ =c_{wk}\left(Q_i^{n+1}U_{ik}^{(n+1)*}+Q_{BC_i}^{n}U_{BC_{ik}}^{n+1}\right)+\nu_{U_{ik}}U_{UBC_{ik}}^{n+1}+\Psi_{IN_{ik}}^{n+1}+ET_{ik}^{n+1} \\ +G_s TR_{ik}^{n+1}+\left(\dfrac{AT_{ik}^{n+1}}{\Delta t_{n+1}}\right)U_{ik}^{n} \quad \mathrm{i}=\overline{1,\mathrm{NN}} \end{array}\right\}\mathrm{k}=\overline{1,\mathrm{NS}+\tau}$$

Eq. 3.9

where

$$AT_{ik}^{n+1}=\left(\varepsilon S_w\rho c_{wk}+(1-\varepsilon)\rho_s c_{sk}\right)_i V_i$$ **Eq. 3.10**

$$DT_{ijk}^{(n+1)*}=\int_x\int_y\int_z\left[\langle\langle\varepsilon\rangle\rangle(S_w\rho)c_{wk}\langle\langle\mathbf{v}\rangle\rangle^{*}\cdot\nabla\phi_j\right]\omega_i\,dz\,dy\,dx$$ **Eq. 3.11**

$$BT_{ijk}^{n+1}=$$
$$\int_x\int_y\int_z\left\{\rho c_{wk}\left[\langle\langle\varepsilon\rangle\rangle S_w\left(\sigma_{wk}\mathbf{I}+\langle\langle\mathbf{D}_k\rangle\rangle\right)+\left(1-\langle\langle\varepsilon\rangle\rangle\right)\langle\langle\sigma_{sk}^{L}\rangle\rangle\mathbf{I}\right]\cdot\nabla\phi_j\right\}\cdot\nabla\phi_i\,dz\,dy\,dx$$ **Eq. 3.12**

$$GT_{ik}^{n+1}=\left(\varepsilon S_w\rho\gamma_{1k}^{w}\right)_i V_i$$ **Eq. 3.13**

$$G_s TL_{ik}^{n+1}=\left((1-\varepsilon)\rho_s\gamma_{1k}^{s}s_{Lk}\right)_i V_i$$ **Eq. 3.14**

$$G_s TR_{ik}^{n+1}=\left((1-\varepsilon)\rho_s\gamma_{1k}^{s}s_{Rk}\right)_i V_i$$ **Eq. 3.15**

$$\Psi_{IN_{ik}}^{n+1}=\int_\Gamma\left\langle\left\langle\rho c_{wk}\left[\varepsilon S_w\left(\sigma_{wk}\mathbf{I}+\mathbf{D}_k\right)+(1-\varepsilon)\langle\langle\sigma_s^{L}\rangle\rangle\mathbf{I}\right]\right\rangle\right\rangle^{n+1}\cdot\mathbf{n}\phi_i\,d\Gamma$$ **Eq. 3.16**

$$ET_{ik}^{n+1} = \left(\varepsilon S_w \rho \gamma_{0k}^{w} + (1-\varepsilon)\rho_s \gamma_{0k}^{s}\right)_i V_i \qquad \textbf{Eq. 3.17}$$

U_{ik}^{n+1} is the concentration or temperature of species k in cell i at the end of the current time step,

U_{ik}^{n} is the concentration or temperature of species k in cell i at the end of the previous time step,

$Q_{BC_i}^{n}$ is the total solute [M/MT] or energy source [E/T] of species k to cell i due to a specified pressure for the previous time step,

ν_{Uik} is the concentration [LT] or temperature-based conductance [LT] for the specified concentration or temperature of species k in cell i (zero (0) for all nodes that are not specified pressure nodes) for the current time step,

s_{Lk} is the sorption isotherm contribution to the left-hand side of equation 3.9 for species k (discussed further in section 3.3),

s_{Rk} is the sorption isotherm contribution to the right-hand side of equation 3.9 for species k (discussed further in section 3.3),

$U_{BC_{ik}}^{n+1}$ is the specified concentration or temperature of species k in cell i for inflow due to a specified pressure (zero (0) for all nodes that are not specified pressure nodes) for the current time step,

n is the unit outward normal vector,

ϕ_j is the symmetric bilinear (2D) or trilinear (3D) basis function in global coordinates for node j,

Γ is the external boundary area of the simulated region [L^2],

$\langle\langle \sigma_s^L \rangle\rangle$ is the solid matrix thermal diffusivity [L^2/T] that is discretized elementwise, and

$\langle\langle \varepsilon \rangle\rangle$ is an elementwise discretization of porosity.

For two-dimensional simulations, equations 3.11 and 3.12 reduce to

$$DT_{ijk}^{(n+1)*} = \int_x \int_y \left[\langle\langle \varepsilon \rangle\rangle (S_w \rho) c_{wk} \langle\langle \mathbf{v} \rangle\rangle^{*} \cdot \nabla \phi_j \right] \omega_i B \, dy \, dx \qquad \textbf{Eq. 3.18}$$

$$BT_{ijk}^{n+1} =$$
$$\int_x \int_y \left\{ \rho c_{wk} \left[\langle\langle \varepsilon \rangle\rangle S_w \left(\sigma_{wk} \mathbf{I} + \langle\langle \mathbf{D}_k \rangle\rangle \right) + \left(1 - \langle\langle \varepsilon \rangle\rangle\right) \langle\langle \sigma_{sk}^{L} \rangle\rangle \mathbf{I} \right] \cdot \nabla \phi_j \right\} \cdot \nabla \phi_i B \, dy \, dx \qquad \textbf{Eq. 3.19}$$

where

the thickness of the mesh, B(x,y), is evaluated at each Gauss point according to equation 3.8.

For three-dimensional simulations, when the geometric-mean approximation for thermal conductivity is used, equation 3.12 for energy transport is modified to

$$BT_{ijk}^{n+1} = \int\limits_{x}\int\limits_{y}\int\limits_{z} \left\{\rho c_{wk} \left[\left\langle\left\langle \sigma_{bgk}^{L} \right\rangle\right\rangle \mathbf{I} + \left\langle\left\langle \varepsilon \right\rangle\right\rangle S_{w} \left\langle\left\langle \mathbf{D}_{k} \right\rangle\right\rangle\right] \cdot \nabla\phi_{j} \right\} \cdot \nabla\phi_{i}\, dz\, dy\, dx \tag{Eq. 3.20}$$

where

$\left\langle\left\langle \sigma_{bgk}^{L} \right\rangle\right\rangle$ is the bulk thermal diffusivity [L^2/T], which includes the constants λ_A and λ_w, and is approximated using a geometric mean and is discretized elementwise if species k is "ENERGY" or molecular diffusivity (D_M) of species k for solute transport, which is a constant value for all elements.

For two-dimensional simulations, the equivalent modification to equation 3.19 for simulation of heat transport using a geometric-mean approximation for thermal conductivity is

$$BT_{ijk}^{n+1} = \int\limits_{x}\int\limits_{y} \left\{\rho c_{wk} \left[\left\langle\left\langle \sigma_{bgk}^{L} \right\rangle\right\rangle \mathbf{I} + \left\langle\left\langle \varepsilon \right\rangle\right\rangle S_{w} \left\langle\left\langle \mathbf{D}_{k} \right\rangle\right\rangle\right] \cdot \nabla\phi_{j} \right\} \cdot \nabla\phi_{i}\, B\, dy\, dx \tag{Eq. 3.21}$$

Equations 3.12 and 3.19 are unmodified for solute transport.

Equation 3.9 is assembled for each node in the model mesh for a given species and time step. The only integrals requiring Gaussian integration are BT_{ij} and DT_{ij}. These integrals are evaluated in an element-by-element manner in the SUTRA-MS subroutine ELEMN2 (for 2D) or ELEMEN3 (for 3D). The other terms, except for those involving ν_{Uik}, are evaluated cellwise (one for each node) by the SUTRA-MS subroutine NODAL. The specified concentration terms are evaluated by subroutine BC.

The matrix assembled for a given species is solved using either Gaussian elimination or an iterative solver. For the iterative solvers, an approximate solution is achieved when the residual error (summation of eq. 3.9 for each node) is less than a specified convergence criterion for a given species. After an approximate solution is achieved for a given species, equation 3.9 is assembled for the next species. This process is continued until an approximate solution is achieved for each species simulated. Details of the flow and transport solution scheme used in SUTRA-MS are given in Section 3.4.

3.3 Temporal Evaluation of Adsorbate Mass Balance

The terms in the modified unified energy- and solute-balance equation (eq. 3.9) that stem from the adsorbate mass balance ($AT_{ik}^{n+1}, GT_{ik}^{n+1}, \text{and } ET_{ik}^{n+1}$) require particular temporal evaluation because some are non-linear. For solute transport, the coefficient, $(c_{sk})_i$, in

AT_{ik}^{n+1} becomes $\left(\kappa_{1k}^{n+1}\right)_i$. The relation that defines $\left(\kappa_{1k}^{n+1}\right)_i$ is given by either equation 2.13, 2.14, or 2.15, depending on the sorption isotherm simulated. The variable, $\left(U_s^{n+1}\right)_i$, is expressed in terms of the concentration of adsorbate $\left(C_{sk}^{n+1}\right)_i$, in a form given by

$$U_s = C_{sk} = s_{Lk}C + s_{Rk} \qquad \textbf{Eq. 3.22}$$

where

C_{sk} is the concentration of the adsorbate for species k, and

all other terms are as previously defined.

The parameters in equation 3.22, s_{Lk} and s_{Rk}, are defined in this section based on the simulated sorption isotherm (eq. 2.13, 2.14, or 2.15).

For linear sorption, all terms and coefficient related to adsorbate mass are linear and are evaluated at the new time level and strictly solved for at this level:

$$\left(U_{sk}^{n+1}\right)_i = \left(C_{sk}^{n+1}\right)_i = \chi_{1k}\rho_o\left(C_k^{n+1}\right)_i \qquad \textbf{Eq. 3.23}$$

$$\left(C_{sk}^{n+1}\right)_i = \left(\kappa_{1k}^{n+1}\right)_i = \chi_{1k}\rho_o \qquad \textbf{Eq. 3.24}$$

$$s_{Lk} = \chi_{1k}\rho_o \qquad \textbf{Eq. 3.25}$$

$$s_{Rk} = 0 \qquad \textbf{Eq. 3.26}$$

For Freundlich sorption, the adsorbate concentration is split into a product of two parts for temporal evaluation. The first part is treated as a first-order term, such as linear sorption, evaluated at the new time level, and solved for on each iteration or time step. The last part is evaluated as a known quantity, either based on a projected value of the initial solute concentration at the end of the time step on the first iteration, or based on the most recent concentration, C_{ik}, on any subsequent iteration. Freundlich adsorption is determined using:

$$\left(U_{sk}^{n+1}\right)_i = \left(C_{sk}^{n+1}\right)_i = \left[\chi_{1k}\left(\rho_o\right)^{\left(\frac{1}{\chi_{2k}}\right)}\left(C_k^{proj}\right)_i^{\left(\frac{1-\chi_{2k}}{\chi_2}\right)}\right]C_{ik}^{n+1} \qquad \textbf{Eq. 3.27}$$

$$\left(C_{sk}^{n+1}\right)_i = \left(\kappa_{1k}^{n+1}\right)_i = \chi_{1k}\rho_o \qquad \textbf{Eq. 3.28}$$

$$s_{Lk} = \chi_{1k}\rho_o^{\left(\frac{1}{\chi_{2k}}\right)}\left(C_k^{proj}\right)_i^{\left(\frac{1-\chi_{2k}}{\chi_2}\right)} \qquad \textbf{Eq. 3.29}$$

$$s_{Rk} = 0 \qquad \textbf{Eq. 3.30}$$

where the coefficient, κ_{1k}^{n+1}, is evaluated using the projected or most recent value of C_{ki}, depending on the iteration.

For Langmuir sorption, the form used preserves the dependence on a linear relationship to C_{ki}. The linear relationship is appropriate only at low solute concentrations. At high concentrations, the adsorbate concentration approaches (χ_{1k}/χ_{2k}). Therefore, one temporal approximation for low concentrations and one temporal approximation for high concentrations are used. When $\chi_2\rho_o C_{ki} << 1$, the following approximation for low values of C, referred to as C_{sk}^0, is used

$$C_{sk}^0 = \left(\chi_{1k}\rho_o C_k^{n+1}\right)\left[1-\frac{\chi_{2k}\rho_o C_k^{proj}}{1+\chi_{2k}\rho_o C_k^{proj}}\right] \qquad \textbf{Eq. 3.31}$$

When $\chi_2\rho_o C_{ki} >> 1$, the following approximation for high values of C, referred to as C_{sk}^∞, is used

$$C_{sk}^\infty = \left(\frac{\chi_{1k}}{\chi_{2k}}\rho_o C_k^{n+1}\right)\left[1-\frac{1}{1+\chi_{2k}\rho_o C_k^{proj}}\right] \qquad \textbf{Eq. 3.32}$$

Thus $\left(C_{sk}^{n+1}\right)_i$ can be defined as

$$\left(U_{sk}^{n+1}\right)_i = \left(C_{sk}^{n+1}\right)_i = W_{0k}C_{sk}^0 + W_{\infty k}C_{sk}^\infty \qquad \textbf{Eq. 3.33}$$

where the weights W_o and W_∞ are

$$W_{\infty k} = \frac{\chi_{2k}\rho_o C_k^{proj}}{1+\chi_{2k}\rho_o C_k^{proj}} \qquad \textbf{Eq. 3.34}$$

$$W_{0k} = 1 - W_{\infty k} \qquad \textbf{Eq. 3.35}$$

By substituting equations. 3.31, 3.32, 3.33, and 3.34 into equation 3.33, the following temporal evaluation of $\left(C_{sk}^{n+1}\right)_i$ is obtained after algebraic manipulation

$$\left(C_{sk}^{n+1}\right)_i = \frac{\chi_{1k}\rho_o C_{ki}^{n+1}}{\left(1+\chi_{2k}\rho_o C_{ki}^{proj}\right)^2} + \frac{\left(\chi_{1k}\rho_o C_{ki}^{proj}\right)\left(\chi_{2k}\rho_o C_{ki}^{proj}\right)}{\left(1+\chi_{2k}\rho_o C_{ki}^{proj}\right)^2} \qquad \textbf{Eq. 3.36}$$

The coefficient, κ_{1k}^{n+1}, is defined as

$$\left(C_{sk}^{n+1}\right)_i = \left(\kappa_{1k}^{n+1}\right)_i = \frac{\chi_{1k}\rho_o}{\left(1+\chi_{2k}\rho_o C_{ki}^{proj}\right)^2} \qquad \textbf{Eq. 3.37}$$

$$s_{Lk} = \frac{\chi_{1k}\rho_o}{\left(1+\chi_{2k}\rho_o C_{ki}^{proj}\right)^2} \qquad \textbf{Eq. 3.38}$$

$$s_{Rk} = \frac{\left(\chi_{1k}\chi_{2k}\right)\left(\rho_o C_{ki}^{proj}\right)^2}{\left(1+\chi_{2k}\rho_o C_{ki}^{proj}\right)^2} \qquad \textbf{Eq. 3.39}$$

The first term in equation 3.36 is solved for each iteration, and the second term is treated as a known value. In equations 3.36, 3.37, 3.38, and 3.39, C_{ki}^{proj} is based on a projection for the first time step and is the most recent value of C_{ki} on subsequent iterations for the time step. (Spell out equation xx in a sentence; use eq. xx when it's in parentheses.)

3.4 Solution Sequencing

On any given time step, the matrix equations are created and solved in the following order: (1) set up the matrix equation for the fluid mass balance, (2) set up the transport-balance matrix equation for the first species, (3) solve for pressure, (4) solve for concentration or temperature of the first species, and (5) set up the transport-balance matrix equation for the second species and solve for each of the remaining species (k=2,NS+τ). The balances for fluid mass transport for the first species are set up at the same time to limit elementwise calculations to a single pass. Elementwise calculations for each of the remaining species are done individually, after a solution is achieved for the preceding species. Fluid flow and transport of all the species (NS+τ) are not solved in a single pass in order to keep storage requirements reasonable for multi-species simulations.

Functionality exists in SUTRA-MS to set up and solve either the solute mass balance or fluid mass balance only every few time steps in a cyclic manner based on parameters NPCYC and NUCYC. This functionality is derived from SUTRA, and the values NPCYC and NUCYC represent the solution cycle in time steps. Currently, a unique NUCYC value cannot be specified for each species but could be easily implemented. Examples include setting up and solving for both flow and transport for three species each time step (NPCYC = NUCYC = 1):

Time step	1	2	3	4	5	6	7	…
Assemble equations for	p U_1	p U_1	p U_1	p U_1	p U_1	P U_1	p U_1	…
Solve for	p U_1	p U_1	p U_1	p U_1	p U_1	p U_1	p U_1	…
Assemble equation for	U_2	U_2	U_2	U_2	U_2	U_2	U_2	…
Solve for	U_2	U_2	U_2	U_2	U_2	U_2	U_2	…
Assemble equation for	U_3	U_3	U_3	U_3	U_3	U_3	U_3	…
Solve for	U_3	U_3	U_3	U_3	U_3	U_3	U_3	…

or solving for flow every three time steps and transport for three species each time step (NPCYC = 3 and NUCYC = 1):

Time step	1	2	3	4	5	6	7	8	9	10	11	12	13	…
Assemble equations for	p	•	P	•	•	p	•	•	p	•	•	p	•	
	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	…
Solve for	p	•	P	•	•	p	•	•	p	•	•	p	•	
	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	U_1	…
Assemble equation for	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	…
Solve for	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	…
Assemble equation for	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	…
Solve for	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	U_3	…

However, either flow or transport must be solved on each time step and requires setting either NPCYC or NUCYC to one (1).

For a simulation with steady flow and transient transport of three species, the sequencing is:

Time step	0	1	2	3	4	5	…
Assemble equations for	p	•	•	•	•	•	
	U_1	U_1	U_1	U_1	U_1	U_1	…
Solve for	p	•	•	•	•	•	
	U_1	U_1	U_1	U_1	U_1	U_1	…
Assemble equation for	U_2	U_2	U_2	U_2	U_2	U_2	…
Solve for	U_2	U_2	U_2	U_2	U_2	U_2	…
Assemble equation for	U_3	U_3	U_3	U_3	U_3	U_3	…
Solve for	U_3	U_3	U_3	U_3	U_3	U_3	…

For a simulation with steady flow and steady transport of three species, the sequencing is:

Time step	0	1
Assemble equations for	p	•
	•	U_1
Solve for	p	•
	•	U_1
Assemble equation for	•	U_2
Solve for	•	U_2
Assemble equation for	•	U_3
Solve for	•	U_3

The only exception to the cycling is that for non-steady cases, both unknowns are solved on the first time step, as shown in the case for NPCYC = 3 and NUCYC = 1, above, and on the last time step, regardless of the values of NPCYC and NUCYC.

It is computationally advantageous to avoid unnecessarily reconstructing the transport equation and, when the direct solver is used, to avoid the transport matrix decomposition steps by allowing for back substitution only. This is begun on the second time step by solving for transport only after the time step on which both fluid mass balance and transport are solved. To do this, the matrix coefficients (including the time step) must remain constant. Thus, non-linear variables and fluid velocity are held constant with values used on the first time step for transport after the step for flow and transport. An example is when NPCYC = 6 and NUCYC = 1 for a simulation with two species:

Time step	1	2	3	4	5	6	7	8	9	10	11	12	…
Assemble equations for	p U_1	• U_1	• U_1	• U_1	• U_1	p U_1	• U_1	• U_1	• U_1	• U_1	• U_1	p U_1	…
Solve for	p U_1	• U_1	• U_1	• U_1	• U_1	p U_1	• U_1	• U_1	• U_1	• U_1	• U_1	p U_1	…
Assemble equation for	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	…
Solve for	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	U_2	…
			constant values						constant values				
				back-substitute						back-substitute			

Note that flow and transport solutions must be set to occur on time steps when relevant boundary conditions, such as sources or sinks, are set to change in value.

4.0 Additional SUTRA-MS Options

This section outlines the additional options available in SUTRA-MS that have not been previously discussed. A general discussion of each option is given and how it is implemented in SUTRA-MS. Details on input data required to execute these options are given in Section 6.

4.1 *Simple Time-Varying Boundary Conditions*

Simple functionality has been included in SUTRA-MS that allows time-varying boundary conditions to be used without the requirement that they be user programmed in the subroutine BCTIME. It is assumed that boundary conditions are constant between user-specified times (time-varying boundary conditions are step functions).

The simple functionality comprises a single subroutine (RDTBCS) that reads the time when boundary conditions are modified and the updated boundary-condition data. An internal logical parameter controls whether the subroutine should process time data or boundary-condition data.

When reading boundary-condition data, the user has complete flexibility in determining which boundary conditions are modified. There is no requirement that the type or number of boundary conditions modified be the same for every boundary-condition modification time period (stress period). For example, the following could be implemented with the simple time-varying-boundary condition option:

Boundary Condition	P_{BC1}	P_{BC2}	P_{BC3}	P_{BC4}	$U(1)_{BC1}$	$U(1)_{BC2}$	$U(1)_{BC3}$	$U(2)_{BC1}$	Q_{fBC1}	$Q_U(2)_{BC1}$
Stress Period 2	•		•			•		•	•	•
Stress Period 3		•	•			•				
Stress Period 4	•	•	•			•		•		
Stress Period 5			•			•				
Stress Period 6			•	•		•		•	•	
…	…	…	…	…	…	…	…	…	…	…

Where P_{BCi} is specified pressure boundary condition i with specified temperatures and/or solute concentrations, $U(n)_{BCi}$ is specified concentration i for species n, Q_{fBCi} is specified fluid flux i, and $Q_U(n)_{BCi}$ is specified solute or heat flux i for species n. The bullets indicate that the specified boundary condition is modified at the beginning of the specified time step.

Boundary conditions that are not modified retain values from the previous time step or the initial values, depending on the current stress period. Unlike previous versions of SUTRA, a negative node number should not be specified for transient boundary conditions that use the simple time-varying-boundary condition routine. The simple time-varying-boundary condition routine can be used in conjunction with the standard user-programmable routine BCTIME.

Time data are read as simulation time, and boundary conditions for the next stress period are read once the specified simulation time is reached. The time-step length is automatically adjusted, if necessary, to ensure that the beginning of each stress period coincides with the beginning of a time step.

In order to accurately simulate early-time transient effects when boundary conditions are modified, the time-step interval is reduced to the minimum time-step length (initial time step length). In subsequent time steps, the time step is allowed to increase according to the user-specified time-step multiplier and number of time steps between increases in time steps.

Error checking has been included in the subroutine RDTBCS to ensure that there are no formatting errors in the transient boundary-condition data set.

4.2 Specified User Output Times

Simple functionality has been included in SUTRA-MS that allows simulation output to be saved at user-specified times. This functionality has been added because the standard version of SUTRA prints output at a user-specified time-step interval. This sometimes makes it difficult to get output at a specific time, especially if a time-step multiplier is used, and can result in large output files with unwanted or unnecessary data.

A single subroutine processes output times (RDPRNT), and output of simulation results at specified times is done within the SUTRA subroutine. Whether output is printed for a particular item (*i.e.*, velocity in the x-direction) depends on whether this item has been included in the output control for the listing file, the nodal data file, or the elemental data file, and whether this file has been included in SUTRA.FIL.

Standard SUTRA output printing at fixed time steps is disabled when the user-specified output time option is implemented. Time data are read as simulation time, and output is written at the end of the time step when the specified output time is reached. The time-step length is automatically adjusted, if necessary, to ensure that time steps coincide with specified output times.

4.3 Simple Automatic Time-Stepping Algorithm

A simple method to reduce the time-step length automatically, as needed, has been included in SUTRA-MS. A single subroutine has been added to process input data for the option (RDATS), and all logic to control the option is contained in the SUTRA subroutine. This option has been added to minimize termination of SUTRA-MS simulations due to non-convergence. This was found to be beneficial during development of SUTRA-MS because of the long run times experienced with several of the sample problems developed during code testing and the need for a reduced time-step length for select periods in the simulations.

The algorithm monitors the maximum number of iterations required for convergence of the pressure and transport solutions. If the number of iterations exceeds a user-specified criteria, the time step is reduced using the following equation

$$\Delta t_{NEW} = \Delta t^{n+1} \ / \ \mathrm{DTMULT}^2 \qquad \textbf{Eq. 4.1}$$

where

Δt_{NEW} is the revised time-step length [sec],
$\Delta t_{(N+1)}$ is the original time-step length for the current time step [sec], and
DTMULT is the user-specified time-step multiplier [ ].

The square of DTMULT is used to calculate Δt_{NEW} in order to reduce the time-step length to the value that was used prior to the last update (two previous time-step cycle changes (ITCYC)). This increases the likelihood that the reduced time-step length is sufficient to overcome convergence problems resulting from inappropriate time-step cycle change parameters and/or maximum time-step lengths.

A user-specified minimum time-step length is used as the lower limit on the time-step length. If Δt_{NEW} is less than the specified minimum time-step length, then it is set to the minimum time-step length. No operations are done on the time-step length if it is already equal to the minimum value. The form of the equation used to reduce the simulation time-step length (eq. 4.1) was chosen for its simplicity and its general appropriateness for most applications. The implementation of the simple automatic time-stepping algorithm is general enough that another method for reducing the time step (*e.g.,* based on Courant criteria, *etc.*) could easily be implemented.

Two options are available that control how the simulation proceeds after the user-specified maximum iteration criterion is exceeded. The first and simplest option is to reduce the time-step length and proceed to the next time step. The second option is more involved and allows the time-step length to be reduced and the solution to be rerun using the new time-step length and results from the previous time step. This option requires resetting the current solution results (n+1) to the solution results from the previous time step (n) and results from the end of the previous time step (n) to results from the time step prior to the previous time step (n-1). This requires additional storage to save the pressure and transport solutions from the previous time step (n), the pressure and transport solutions from the time step prior to the previous time step (n-1), the density, adsorbate mass for all species, and saturation from the previous time step (n). To prevent excessive recalculation of the current time step when the second option is selected , the time-step length is reduced and the solution is rerun until a user-specified number of reruns are completed or the user-specified maximum number of iterations criterion is satisfied.

Increases in time-step length are handled by standard SUTRA time-step-adjustment algorithms. The simple automatic time-stepping algorithm in combination with standard SUTRA time-step algorithms affords the user great flexibility in tailoring the solution scheme to specific problems. However, because the simple automatic time-stepping algorithm requires the maximum allowable iteration criterion be set by the user, this parameter should be adjusted to provide the optimal value. A maximum iteration criterion that is excessively small will result in excessive time-step-length reductions and may unnecessarily increase run times. Conversely, a maximum iteration criterion that is excessively large will reduce the effectiveness of the algorithm and may allow non-convergent solutions to occur. For optimal performance, DTMULT and the number of time steps in a time-step cycle change, ITCYC, used by the standard SUTRA time-step algorithms, should be adjusted in combination with tuning of simple automatic time-stepping parameters for best results. It should be noted that this algorithm does not ensure that convergence will be achieved for all model set-ups. Chronic non-convergence may indicate problems with the data set, discretization that does not meet stability requirements (Peclet criteria), time-step lengths that exceed stability criteria (Courant criteria), or incorrect boundary conditions.

4.4 Specified Observation Locations

Simple functionality has been included in SUTRA-MS that allows specification of observation locations using actual coordinates. This functionality has been added because the standard version of SUTRA requires observation locations be specified using node locations. This additional functionality allows definition of observations to be mesh independent.

Specified observation location input data are processed by a single subroutine (Allo_Rd_SobData), which also allocates the required storage, and a single subroutine to calculate the closest node to the specified location (CalcObsNode). The correspondence of a specified observation location to a specific node is printed in the standard output file (".lst"), and information also is given if the specified observation location is outside the mesh. If a specified observation location is outside the mesh, the closest node is used; a warning message is written to the screen, but the simulation proceeds.

4.5 Specification of Hydraulic Parameters Using Zones

Functionality has been included in SUTRA-MS that allows hydraulic parameters to be specified using zones. This additional functionality can significantly reduce memory requirements through use of data structures dimensioned based on the number of hydraulic zones rather than vectors of hydraulic-parameter data dimensioned based on the number of nodes and elements. The ability to specify hydraulic parameters using zones reduces the effort required to set up and use SUTRA-MS with parameter-estimation codes (*i.e.,* UCODE or PEST).

In order to reduce the memory requirements for problems using zones, the original data structure for hydraulic parameters was modified from vectors dimensioned by nodes or elements to a number of data structures dimensioned by zones. A simple representation of the differences between the modified data structure and the original SUTRA structure is shown in figure 4.5.1. A separate data structure has been established for nodewise and elementwise discretized hydraulic data. Currently, the nodewise data structure includes porosity, matrix compressibility, and matrix density. The elementwise data structure includes permeability, permeability angle, dispersivity, and matrix thermal conductivity. Except in the case where the number of node and element zones is equal to the number of nodes and elements, memory requirements are less for problems that specify hydraulic parameters using zones.

Elements	26
Zones	2

Element	PMAX	PMIN	ANGLEX	ALMAX	ALMIN	ATMAX	ATMIN	SIGMAS*
1	*A1*	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*	*NA*
2	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*	*NA*
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
26	*Z1*	*Z2*	*Z3*	*Z4*	*Z5*	*Z6*	*Z7*	*NA*
Bytes	208	208	208	208	208	208	208	1
Total Bytes	1457							

	Zone	1	2
	PMAX	*A1*	*B1*
	PMIN	*A2*	*B2*
	ANGLEX	*A3*	*B3*
	ALMAX	*A4*	*B4*
	ALMIN	*A5*	*B5*
	ATMAX	*A6*	*B6*
	ATMIN	*A7*	*B7*
	SIGMAS	*A8*	*B8*
Bytes		64	64
Total Bytes		128	

* SIGMAS is constant for all nodes in SUTRA

NA - Not Applicable

Figure 4.5.1 Simple representation of the differences in memory requirements for hydraulic parameters that are discretized by elements for SUTRA and SUTRA-MS using zones.

When using zones, all hydraulic-parameter data are specified in the ".zon" file, and the spatial distribution of nodal and elemental zones are specified using the existing NREG and LREG vectors that were used previously to specify the distribution of unsaturated zone properties. Addition of matrix compressibility and matrix density to nodewise parameters that can be varied on a zone basis allows more flexibility with storage and matrix density properties than was possible with SUTRA. This means that transient responses due to storage differences and spatial differences in adsorption resulting from differences in aquifer materials can be accommodated directly without additional modifications to the source code.

The data structure used to specify hydraulic parameter zones has been made general so that it can be easily extended to add additional parameters. For example, it could easily be extended to allow specification of unsaturated zone properties and would eliminate the need to use the user programmed UNSAT subroutine for unsaturated flow problems.

All input processing is handled by a single subroutine (RdZoneData), which also includes some functionality to write zone information to the standard output file (".lst"). The modifications resulting from implementation of the zone module are widespread throughout the code. Every location where an element of a hydraulic-parameter vector was called for has been replaced with the equivalent zone data-structure element. An example of the modification is:

PERMXX (L) ⇒ ElemData(ElemMap(L))%permxx

where,

PERMXX (L) is the permeability of element L in the XX direction,

ElemData() is the hydraulic permeability data structure for elementwise discretized hydraulic-parameter data,

ElemMap(L) is the zone number for element L (LREG(L) in SUTRA), and

ElemData(ElemMap(L))%permxx is the permeability of zone ElemMap(L) in the XX direction.

Error checking has been included in the subroutine RdZoneData to ensure that there are no formatting errors in the node and element zone data sets. In addition, a subroutine (MkZoneSOP) to calculate the specific storage for each zone replaces the standard calculation in subroutine INDAT1.

5.0 SUTRA-MS Simulation Examples

This section outlines three example problems that demonstrate some of the specific capabilities of SUTRA-MS. The examples show results that are compared with numerical solutions from SUTRA or other numerical codes. In several cases, the examples demonstrate some interesting applications possible with SUTRA-MS. Additional examples of the capabilities of SUTRA are contained in Voss and Provost (2002).

5.1 Density-dependent flow, heat transport, and solute transport, Solution for multi-component fluid flow in a saline aquifer system

5.1.1 Physical Setup

This example considers seawater intrusion into both an isothermal and a non-isothermal confined aquifer studied in cross section under steady-state conditions (Henry and Hilleke, 1972). Freshwater recharge flows from an inland boundary over more saline water derived from a seaward boundary and discharges in the upper portions of a vertical sea boundary.

The problem is non-linear and is solved by gradually approaching steady state with a series of time steps. Initially the aquifer has hydrostatic-pressure conditions with either isothermal and 0.0 percent seawater concentrations or linearly varying percent seawater concentrations and temperatures. At time zero, heat begins to be transported inward from the top, bottom, and left boundaries, and seawater begins to intrude the freshwater system by moving in laterally under the freshwater from the right (seaward) boundary. Seawater intrusion in the aquifer is primarily the result of the greater density of the seawater. Lateral temperature variations at the top and bottom of the aquifer and vertical temperature variations at the freshwater boundary increase vertical freshwater fluid movement and intrusion of seawater at the base of the aquifer when compared to a similar isothermal case.

Dimensions of the problem were selected to allow comparison with the steady-state dimensionless solution of Henry and Hilleke (1972). Two different aspect ratios are evaluated to assess the impact of the aquifer-aspect ratio (ratio of aquifer length to height – $\{\upsilon\}$) on simulation results. Two different total simulation times (3.3 and 33.5 hours) were used for the two aspect ratios evaluated; both simulation times are sufficient for the simulations to reach steady-state conditions.

5.1.2 Simulation Setup

Two different meshes were used in order to evaluate the effect of aquifer-aspect ratios on the simulation results. The first mesh consists of 40 by 40 elements, each of size 0.025 m (meters) by 0.025 m, and was used for $\upsilon=1$. The second mesh consists of 50 by 501 elements, each of size 0.02 m by 0.02 m, and was used for $\upsilon=0.10$. An example of the mesh geometry for $\upsilon=1$ is shown on figure 5.1.1. The thickness was 1.0 m in the y-dimension for both meshes. A constant time-step length of 60 and 30 seconds was used for υ=1 and υ=0.10, respectively. A total of 200 and 4,022 time steps were taken for the υ=1 and 0.10 simulations. The number of time steps was increased for the υ=0.10 simulations because the problem is an order of magnitude larger in the x-dimension than the υ=1 problems. A reduced time-step length also was used for the υ=0.10 simulations to reduce numerical oscillations. Pressure, concentration, and temperature were solved on each time step (NPCYC=NUCYC=1).

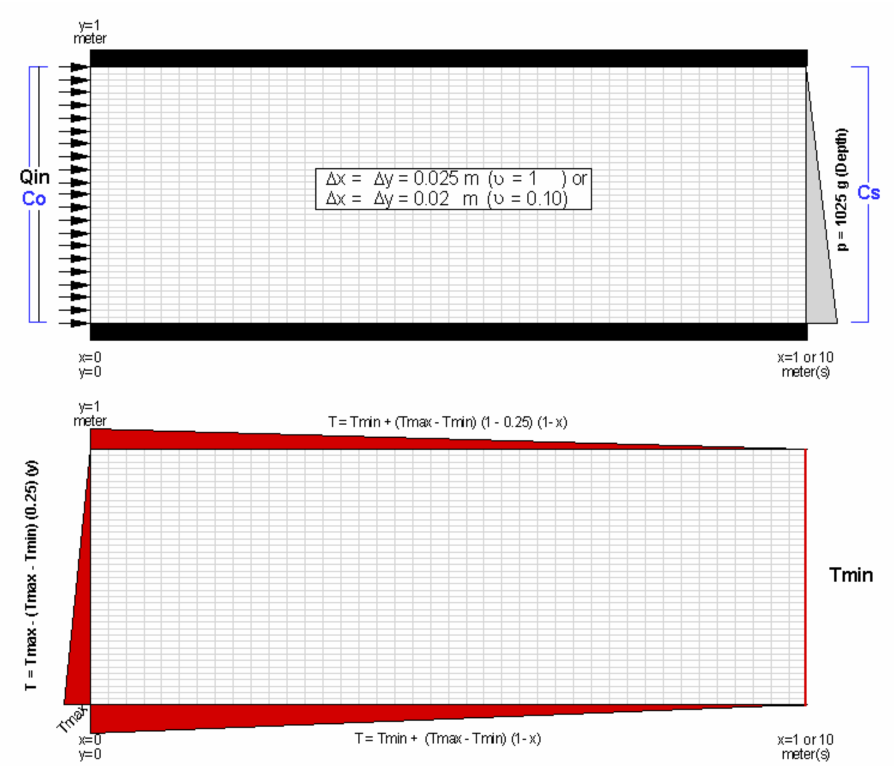


Figure 5.1.1 **Finite element mesh (υ=1) and pressure (gray), concentration (blue), and temperature (red) boundary conditions for Henry and Hilleke (1972) solution.**

5.1.3 Parameters

$k = 1.02041 \times 10^{-9}$ [m^2] based on $K = 0.01$ [m / s]

$\varepsilon = 0.35$ [-]

$C_S = 0.0357$ [kg$_{\text{dissolved solids}}$ / kg$_{\text{seawater}}$]

$\rho_S = 1024.99$ [kg / m^3]

$\rho_{\text{SOLID}} = 2600.$ [kg / m^3]

$$\frac{\partial \rho}{\partial C} = 700.\ [\text{kg}_{\text{seawater}}^{2} / (\text{kg}_{\text{dissolved solids}}\ \text{m}^3)\ ]$$

$b = 1.0$ [m]

$|\mathbf{g}| = 9.8$ [m / s^2]

$C_O = 0.000$ [kg$_{\text{dissolved solids}}$ / kg$_{\text{seawater}}$]

$\rho_O = 1000.$ [kg / m^3]

$Q_{IN} = 8.333 \times 10^{-5}$ [kg /s] distributed over 41 nodes
or
8.333×10^{-6} [kg /s] distributed over 51 nodes

$\mu = 1 \times 10^{-3}$ [kg / (m s)]

$\frac{\partial \rho}{\partial T}$ = -0.375 [$kg_{seawater}$ / (°C m^3)]

D_{SOLUTE} = 2.381 x 10^{-5} [m^2 / s]
or
= 2.381 x 10^{-7} [m^2 / s]

λ_W = 995.7342 [J / (m °C s)]
or
= 9.957 [J / (m °C s)]

c_W = 4182. [J / (kg °C)]

T_{MIN} = 5 °C

$\alpha_L = \alpha_T$ = 0.0 [m]

$D_{TEMPERATURE}$ = 2.381 x 10^{-4} [m^2 / s]
or
= 2.381 x 10^{-6} [m^2 / s]

λ_{SOLID} = 0.0 [J / (m °C s)]

c_{SOLID} = 0.0 [J / (kg °C)]

T_{MAX} = 50 °C

5.1.4 Boundary Conditions

No-flow conditions were maintained across the top and bottom boundaries. A freshwater source, with a concentration of 0.000 $kg_{dissolved\ solids}/kg_{freshwater}$ for all cases and a temperature that varies linearly from 50°C at y=0 m to 38.75°C at y=1 m for the non-isothermal case, was implemented by using fluid source nodes at the left vertical boundary. The right vertical boundary was held at hydrostatic pressure, assuming a constant density equal to that of seawater, through use of specified pressure nodes. Any water entering through these nodes has the concentration, C_S, of seawater for all cases and a temperature, T_{MIN}, of 5°C for the non-isothermal case. Temperatures were not assigned to the specified flux nodes on the left boundary or the specified pressure nodes on the right boundary in the isothermal case.

In the original Henry and Hilleke (1972) simulation, fluid concentrations were fixed on the left and right boundaries and temperatures were fixed at all the external boundaries. In order to replicate the original results, specified concentration nodes were assigned to the right and left boundaries for the isothermal and non-isothermal cases. In the non-isothermal case, specified temperature nodes were assigned to the right, left, top, and bottom boundaries. Specified temperature nodes were not assigned in the isothermal case. Use of specified concentration and temperature nodes at the boundaries also allow the SUTRA-MS simulations to be compared directly with results using HST3D (Kipp, 1987).

Concentrations of 0.0357 and 0.0000 $kg_{dissolved\ solids}/kg_{seawater}$ were assigned to the specified concentration nodes along the right and left boundaries, respectively. In the non-isothermal case, a constant temperature of 5°C was assigned to the specified temperature nodes along the right boundary, and temperatures ranging from 50 to 38.5°C were assigned to specified temperature nodes along the left boundary. Additionally, in the non-isothermal case, a specified temperature varying from 50 to 5°C and 38.5 to 5°C in the direction of increasing x was assigned to specified temperature nodes on the bottom and top boundaries, respectively..

The assigned pressure and concentration boundary conditions are shown graphically on figure 5.1.1A. Assigned temperature conditions for the non-isothermal case are shown graphically on figure 5.1.1B.

5.1.5 Initial Conditions

Hydrostatic pressures based on an initial solute concentration of 0.000 $kg_{dissolved\ solids}/kg_{seawater}$ and temperature of 5°C were used as initial conditions for υ=1 simulations. Hydrostatic pressures based on initial solute concentrations and temperatures linearly varying from 0.000 to 0.0357 $kg_{dissolved\ solids}/kg_{seawater}$ and 44.375 to 5.000°C from x = 0.0 to 10.0 m were used for υ=0.10 simulations. Linear variations in initial solute concentrations and temperatures were used for υ=0.10 simulations in an effort to provide initial conditions that were closer to steady-state conditions and to minimize numerical instabilities.

5.1.6 Results

Henry and Hilleke's (1972) solution assumes that total solute dispersion can be accounted for through use of a large solute molecular-diffusion coefficient rather than a combination of velocity-dependent dispersivity and molecular diffusion. Similarly, thermal dispersion is simulated using a large fluid thermal conductivity and insulated solid matrix. Use of an insulated solid matrix is an appropriate simplification because use of a large fluid thermal conductivity is equivalent to the net effect of a smaller, more reasonable fluid thermal conductivity and a physically realistic solid matrix thermal conductivity. The total solute and thermal dispersion coefficient of Henry and Hilleke (1972), D_M and D_T, are equivalent to the product of the porosity and solute and thermal molecular diffusivity, respectively (eqs. 5.1 and 5.2).

$$D_M = \varepsilon D_{SOLUTE} \qquad \textbf{Eq. 5.1}$$

$$D_T = \varepsilon D_{TEMP} = \frac{\varepsilon \lambda_W}{c_W \rho} \qquad \textbf{Eq. 5.2}$$

Henry and Hilleke's results are given for two of their non-dimensional simulations: N_Ψ = 3 and N_C = 10 for an isothermal case and N_Ψ = 3, N_C = 10, and N_T = 1 for a non-isothermal case. The dimensionless parameters N_Ψ, N_C, and N_T are defined in eqs. 5.3, 5.4, and 5.5, respectively.

$$N_\Psi = \frac{KbR'\upsilon}{Q_T} \qquad \textbf{Eq. 5.3}$$

$$N_C = \frac{Q_T \upsilon}{D_M} \qquad \textbf{Eq. 5.4}$$

$$N_T = \frac{Q_T \upsilon}{D_T} \qquad \textbf{Eq. 5.5}$$

where

K is the hydraulic conductivity [L/T],
b is the aquifer thickness [L],
R' is the fractional difference in freshwater and seawater density = 0.025,

$$\upsilon_{\cdot} = \left[\frac{b}{L}\right] = 1 \text{ or } 0.1\text{, and} \qquad \textbf{Eq. 5.6}$$

Q_T is the total freshwater inflow per unit width [L^3/LT].

In order to match the non-dimensional Henry and Hilleke parameters listed above, values of $D_{SOLUTE} = 2.381\text{x}10^{-5}$ m^2 / s, $D_M = 8.333\text{x}10^{-6}$ m^2 / s, $D_{TEMP} = 2.381\text{x}10^{-4}$ m^2 / s, and $D_T = 8.333\text{x}10^{-5}$ m^2 / s are required for υ=1. For υ=0.10, $D_{SOLUTE} = 2.381\text{x}10^{-7}$ m^2 / s, $D_M = 8.333\text{x}10^{-8}$ m^2 / s, $D_{TEMP} = 2.381\text{x}10^{-6}$ m^2 / s, and $D_T = 8.333\text{x}10^{-7}$ m^2 / s are required.

Results for the isothermal problem and the υ=1 mesh using SUTRA-MS and HST3D are compared with the Henry and Hilleke solution for the 0.5-percent seawater contour (figure 5.1.2). The simulated results from SUTRA-MS and HST3D compare favorably at the base of the model but differ slightly at the top of the model for the 0.5-, 0.2-, and 0.05-percent seawater contours. Neither SUTRA-MS nor HST3D compare favorably to the original Henry and Hilleke (1972) solution using the υ=1 mesh.

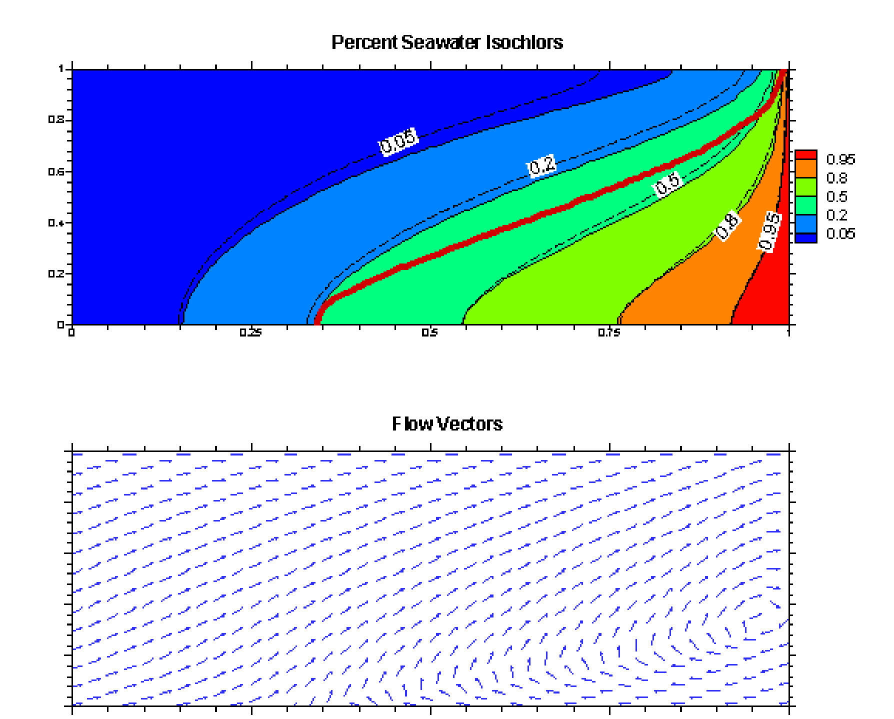


Figure 5.1.2 Match of percent-seawater contours and the SUTRA-MS flow field for N_Ψ=3, N_C=10, and υ=1 Henry and Hilleke numerical solution (0.5-percent seawater concentration only) (solid red line), HST3D code solution (dashed black line), and SUTRA-MS solution (colored).

Results for the isothermal problem and the $\upsilon=0.10$ mesh using SUTRA-MS and HST3D are shown in figure 5.1.3 and are compared with the Henry and Hilleke solution for the 0.5-percent seawater contour. The simulated results from SUTRA-MS and HST3D are in very close agreement with each other except at the base of the model where the 0.95- and 0.8-percent seawater contours differ slightly. SUTRA-MS and HST3D solutions are closer to the original Henry and Hilleke (1972) solution using the $\upsilon=0.10$ mesh than they are with the $\upsilon=1$ mesh.

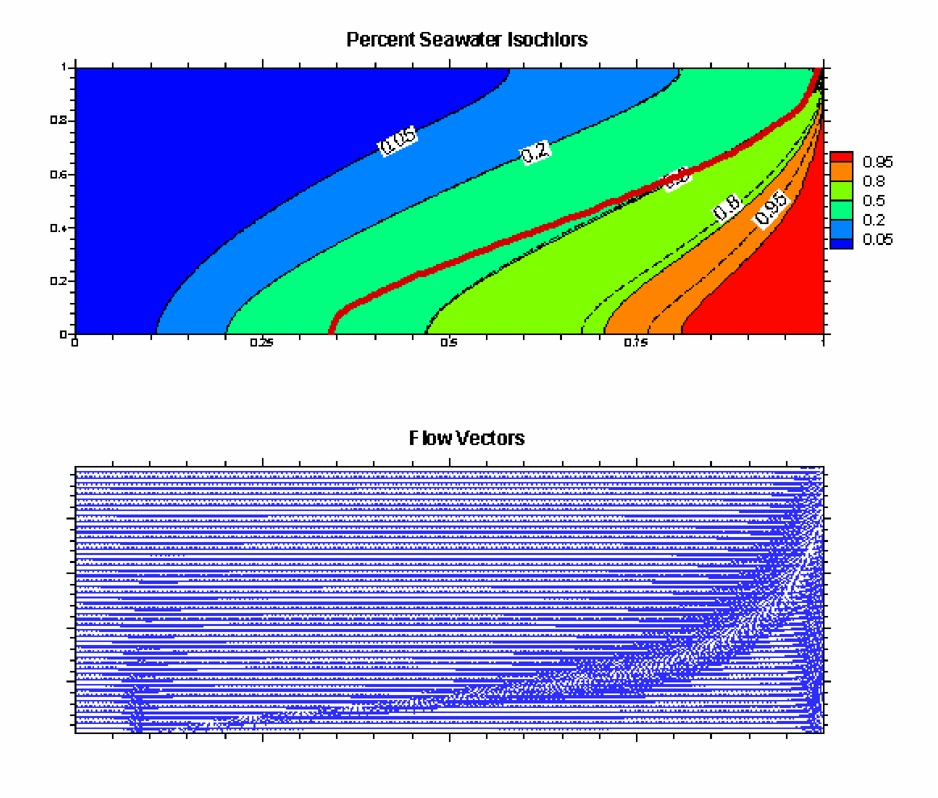


Figure 5.1.3 **Match of percent-seawater contours and the SUTRA-MS flow field for $N_{\Psi}=3$, $N_C=10$, and $\upsilon=0.10$ Henry and Hilleke numerical solution (0.5-percent seawater concentration only) (solid red line), HST3D code solution (dashed black line), and SUTRA-MS solution (colored).**

Results for the non-isothermal problem using SUTRA-MS, HST3D, and the $\upsilon=1$ mesh are shown in figure 5.1.4 and are compared with the Henry and Hilleke solution for the 0.5 percent seawater contour. The simulated results from SUTRA-MS and HST3D compare favorably at the 0.95-, 0.8-, 0.5-, and 0.2-percent seawater contour except at the top of the model where the 0.2- and 0.05-percent seawater contours differ slightly. SUTRA-MS and HST3D isotherms compare favorably at all temperatures. Neither SUTRA-MS nor HST3D solute concentrations compare favorably to the original Henry and Hilleke (1972) solution with the $\upsilon=1$ mesh. Simulated temperatures for SUTRA-MS and HST3D compare favorably to the original Henry and Hilleke (1972) solution at the top and bottom of the model but not in the center of the model.

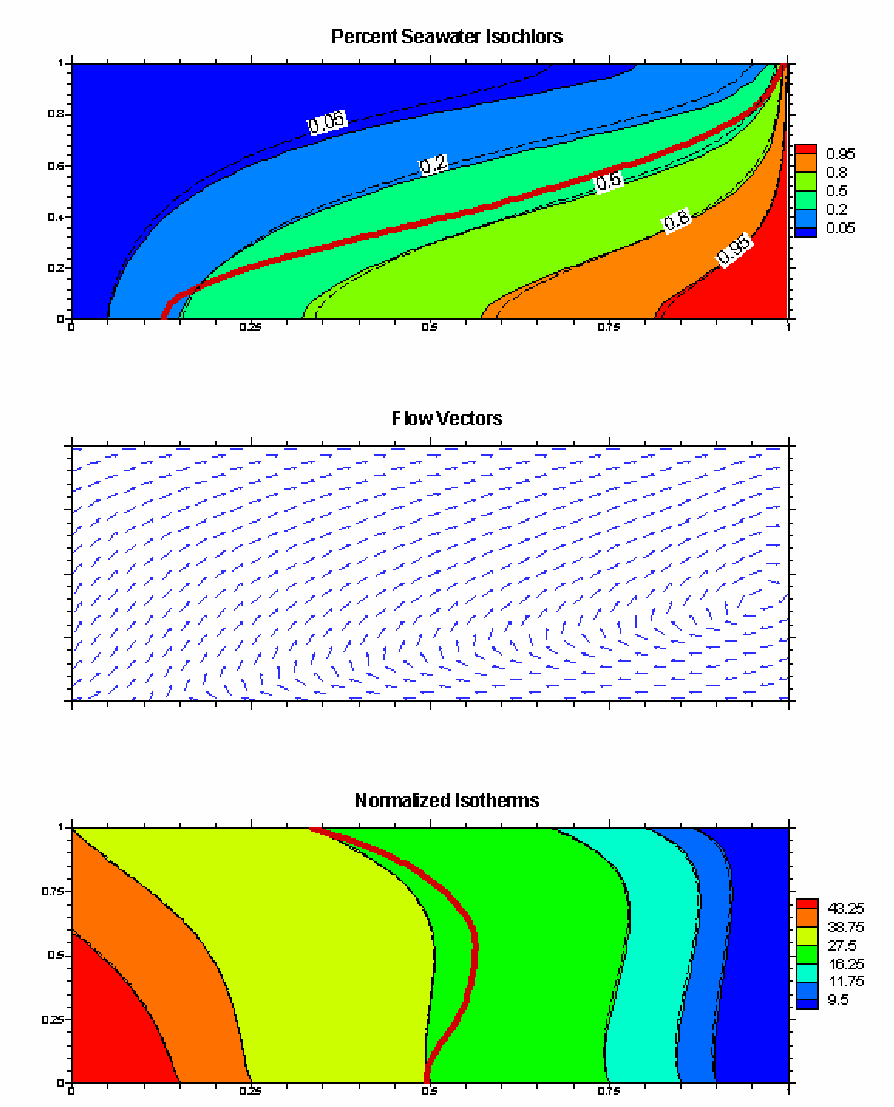


Figure 5.1.4 Match of percent seawater contours, SUTRA-MS flow field, and match of isotherms for N_{Ψ}=3, N_C=10, N_T=1, and υ=1 Henry and Hilleke numerical solution (0.5 percent seawater concentration and isotherm only) (solid red line), HST3D code solution (dashed black line), and SUTRA solution (colored) with D_M=8.333x10^{-6} and D_T=8.333x10^{-5}.

Results for the non-isothermal problem using SUTRA-MS, HST3D, and the υ=0.10 mesh are shown in figure 5.1.5 and are compared with the Henry and Hilleke solution for the 0.5-percent seawater contour. The simulated results from SUTRA-MS and HST3D compare favorably at the top of the model but are slightly off for the 0.95-, 0.8-, and 0.5-percent seawater contours at the base of the model. SUTRA-MS and HST3D isotherms compare favorably at all temperatures. SUTRA-MS and HST3D percent-seawater concentrations do not compare favorably to the original Henry and Hilleke (1972) solution with the υ=0.10 mesh but are closer than simulations that used the υ=1 mesh. Simulated temperatures for SUTRA-MS and HST3D compare favorable to the original Henry and Hilleke (1972) solution.

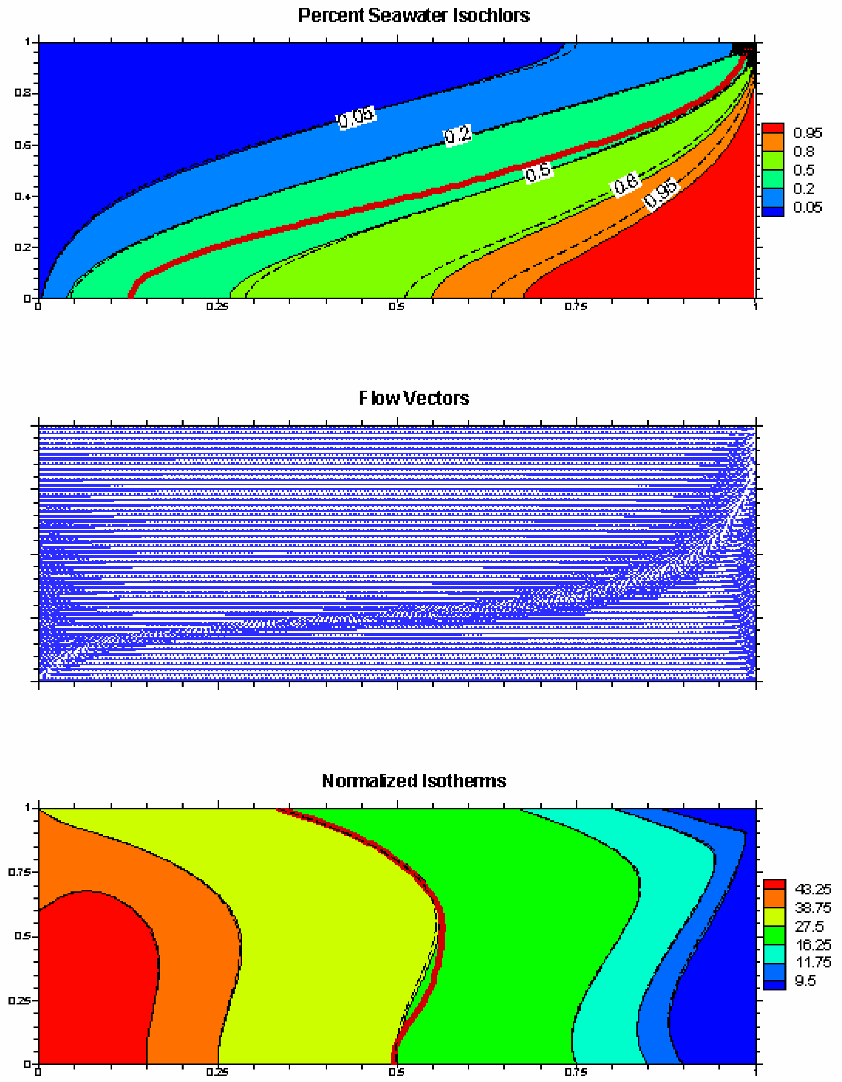


Figure 5.1.5 Match of percent-seawater contours, SUTRA-MS flow field, and match of isotherms for $N_{\Psi}=3$, $N_C=10$, $N_T=1$, and $\upsilon=0.10$ Henry and Hilleke numerical solution (0.5-percent seawater concentration and isotherm only) (solid red line), HST3D code solution (dashed black line), and SUTRA solution (colored) with $D_M=8.333\text{x}10^{-8}$ and $D_T=8.333\text{x}10^{-7}$.

The original Henry and Hilleke (1972) solutions were based on a finite-difference model that used a simplified form of the variable-density flow and transport equation. The simplified form approximates the full equation when $\upsilon<<1$. The finite-difference model was developed based on an aquifer length of 268 km (kilometers) and an aquifer depth of 0.762 km ($\upsilon<<1$). The simulations presented for the $\upsilon=1$ and $\upsilon=0.10$ meshes indicate that a closer correspondence to the original Henry and Hilleke (1972) solution could be obtained with meshes having aspect ratios (υ) less than 0.10. Aspect ratios less than 0.10 were not simulated because of the spatial discretization required to satisfy numerical stability criteria (table 5-1).

Table 5-1. Discretization requirements for several aquifer aspect ratios (υ). See Appendix 1 for symbol definitions.

Mesh Geometry	X (m)	Y (m)	width (m)	K (m/s)	ε (-)	ρ_o (kg/m^3)	υ	Q (m^3/ms)
1	*1*	1	1	0.01	0.35	1000	1	8.333E-05
2	*10*	1	1	0.01	0.35	1000	0.1	8.333E-06
3	*100*	1	1	0.01	0.35	1000	0.01	8.333E-07
4	*1000*	1	1	0.01	0.35	1000	0.001	8.333E-08
5	*10000*	1	1	0.01	0.35	1000	0.0001	8.333E-09

Mesh Geometry	Q_{IN} (kg/s)	D_S (m^2/s)	D_T (m^2/s)	λ_w	v_x (m/s)	Δx from Pe_x=2 and D_T (m)	Δx from Pe_x=2 and D_S (m)	nx_{MAX}
1	8.333E-02	2.381E-05	2.381E-04	9.957E+02	2.381E-04	2.00	0.2	5
2	8.333E-03	2.381E-07	2.381E-06	9.957E+00	2.381E-05	0.20	0.02	500
3	8.333E-04	2.381E-09	2.381E-08	9.957E-02	2.381E-06	0.02	0.002	50000
4	8.333E-05	2.381E-11	2.381E-10	9.957E-04	2.381E-07	0.002	0.0002	5000000
5	8.333E-06	2.381E-13	2.381E-12	9.957E-06	2.381E-08	0.0002	0.00002	500000000

Henry and Hilleke (1972) indicated that their solution had erratic values in the upper right-hand corner of the model, and convergence was dependent on using initial conditions that were reasonably close to the final solution. Henry and Hilleke (1972) suggested the erratic values in the upper right-hand corner were a result of coarse discretization at the outflow face, and they reset calculated concentrations and temperatures to zero or one if numerical results exceeded the minimum or maximum values of zero and one, respectively. These same numerical issues are present in the SUTRA-MS and HST3D results for the υ=0.10 simulations but were not resolved, because instabilities were restricted to the outflow face, did not propagate a significant distance into the problem domain, and would require finer discretization at the outflow face that would increase numerical overhead and simulation times substantially.

5.2 Solution for double-diffusive finger convection induced by different fluid dispersivities and viscosities

5.2.1 Physical Setup

The intent of this example is to verify the accuracy of the SUTRA-MS in representing bulk fluid flow initiated by differences in the dispersivity and viscosity of two solute species (NaCl and sucrose). Unlike the other examples presented in this report, this problem is driven exclusively by internal changes in fluid density and viscosity and not by external boundary conditions. In order to compare simulated and experimental results, a conservative dye solution also is simulated that approximately maps the evolving NaCl field. The mapping of the NaCl field is approximate because the dispersivity of the conservative dye is less than the dispersivity of NaCl by a factor of about 2.5.

Initially, the two solutes were unmixed and vertically stratified in a stable density configuration. To initiate flow, the interface between the unmixed fluids was randomly perturbed. A conservative dye tracer, initially coincident with the NaCl solution, also was randomly perturbed at the interface. The initial perturbations of the species that affect fluid density form the nucleus for the developing fingers.

The dimensions of the problem were selected to be identical to those of the original Hele-Shaw experiment by Pringle and others (2002) in order to allow for a direct comparison of simulated and observed results. The total simulation time was 16 hours, which is identical to the duration of the original experiment.

5.2.2 Simulation Setup

A non-intrusive light-transmission technique and a charge-coupled-device (CCD) camera were used to measure the light intensity field transmitted through the Hele-Shaw cell. The evolution of the flow field was monitored with a dye tracer (Warner Jenkins FD&C Blue #1) added to the NaCl solution. The Hele-Shaw cell was 0.2541 m wide and 0.1625 m high and the progression of the experiment was recorded using images captured with the CCD camera and converted to concentrations using calibration curves obtained from a series of dye concentrations in a base salt solution. Captured images had a resolution of 1650 x 1055 pixels with a pixel size of 1.54 x 10^{-4} m.

The simulation mesh consisted of 1,024 by 656 square elements that are 2.48 x 10^{-4} m by 2.48 x 10^{-4} m. The pixel resolution in the experimental setup of Pringle and others (2002) was a factor of 1.7 times smaller than the cell size used in the numerical simulation. The selected numerical discretization was a compromise between computer run times and desired numerical accuracy. A simplified version of the simulation mesh is shown in figure 5.2.1.

A variable time step, ranging from 0.1 to 10 seconds, and a time-step multiplier of 1.05 were used. Small time steps were required to achieve convergence during early stages of the simulation when mass transfer rates and vertical velocities were higher. Pressure, NaCl concentration, sucrose concentration, and dye concentration were solved on each time step (NPCYC=NUCYC(k)=1), and an iterative solution was used to resolve non-linearities during each time step (ITRMAX=10).

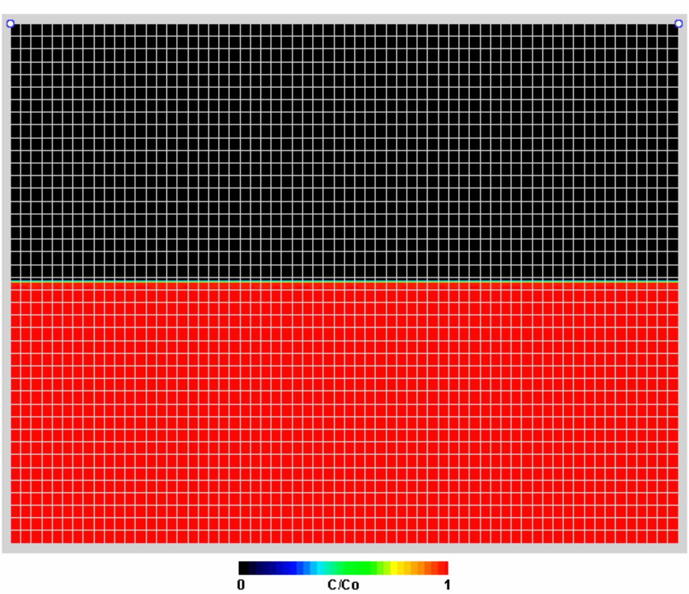


Figure 5.2.1 **Boundary and initial concentration conditions and finite-element mesh (every 16th element) used to simulate the Pringle and others (2002) Hele-Shaw experiment.**

5.2.3 Parameters

$k = 2.61 \times 10^{-9}$ [m^2]

$\varepsilon = 1.0$ [-]

$C_O(NaCl) = 0.000$ [kg_{NaCl} / kg_{fluid}]

$C_{MAX}(NaCl) = 0.03463$ [kg_{NaCl} / kg_{fluid}]

$C_O(Dye) = 0.000$ [kg_{Dye} / kg_{fluid}]

$\rho_O = 998.$ [kg / m^3]

$$\frac{\partial \rho}{\partial C_{NaCl}} = 689. \ [kg_{fluid}^2 / (kg_{NaCl} \ m^3)]$$

$$\frac{\partial \mu}{\partial C_{NaCl}} = 1.59 \times 10^{-3} \ [kg_{fluid} \ kg / (kg_{NaCl} \ m \ s)]$$

$$\frac{\partial \rho}{\partial C_{Dye}} = 0.00 \ [kg_{fluid}^2 / (kg_{Dye} \ m^3)]$$

$D_{NaCl} = 1.477 \times 10^{-9}$ [m^2 / s]

$D_{Dye} = 5.670 \times 10^{-10}$ [m^2 / s]

$b = 1.0$ [m]

$|\mathbf{g}| = 4.14$ [m / s^2] based on 9.8[m / s^2] sin (25°)

$C_O(Sucrose) = 0.000$ [$kg_{Sucrose} / kg_{fluid}$]

$C_{MAX}(Sucrose) = 0.05234$ [$kg_{Sucrose} / kg_{fluid}$]

$C_{MAX}(Dye) = 0.00025$ [kg_{Dye} / kg_{fluid}]

$\mu_o = 1 \times 10^{-3}$ [$kg / (m \ s)$]

$$\frac{\partial \rho}{\partial C_{Sucrose}} = 371. \ [kg_{fluid}^2 / (kg_{Sucrose} \ m^3)]$$

$$\frac{\partial \mu}{\partial C_{Sucrose}} = 2.75 \times 10^{-3} \ [kg_{fluid} \ kg / (kg_{Sucrose} \ m \ s)]$$

$$\frac{\partial \mu}{\partial C_{Dye}} = 0.00 \ [kg_{fluid} \ kg / (kg_{Dye} \ m \ s)]$$

$D_{Sucrose} = 4.878 \times 10^{-10}$ [m^2 / s]

$\alpha_L = \alpha_T = 0.0$ [m]

5.2.4 Boundary Conditions

No-flow conditions occured across the top, bottom, left, and right boundaries except at the upper right and left nodes. (blue circles on figure 5.2.1). The top right and top left boundaries were specified to have a 0.0 [Pa] pressure, 0.00000 kg_{NaCl} / kg_{fluid}, 0.05234 $kg_{sucrose}$ / kg_{fluid}, and 0.00000 kg_{Dye} / kg_{fluid}] in order to prevent calculation of negative pressures. No external sources of fluid were used.

5.2.5 Initial Conditions

Hydrostatic pressures based on the initial concentrations of NaCl and sucrose shown on figure 5.2.1 were used as initial conditions. The conservative dye tracer was restricted initially to the lower half of the Hele-Shaw cell, coincident with the initial distribution of NaCl. In order to simulate the small-scale perturbations that existed in the original experiment, the concentrations of NaCl, sucrose, and the conservative dye tracer were randomly perturbed at the interface.

The interface was randomly perturbed using the following procedure. Random numbers, ranging from 0 to 1, were generated for concentrations of NaCl, sucrose, and the conservative dye tracer for all nodes located at the interface between the NaCl and sucrose solutions. If the random number generated for a specific node and solute species was less than 0.5, the initial concentration of the solute species was reduced. Conversely, if the random number generated for a specific node and solute species was greater than or equal to 0.5, the concentration of the solute species was not perturbed. A maximum perturbation of 1 percent of the maximum solute concentrations was used for all the SUTRA-MS simulations. Hydrostatic pressures were reset based on the perturbed concentrations developed at the interface.

5.2.6 Results

Concentration-dependent density and viscosity relationships for NaCl and sucrose were derived from Weast (1986). As shown in figure 5.2.2, NaCl and sucrose show approximately linear concentration-density and concentration-viscosity relationships over the range of concentrations used in the Hele-Shaw experiment. Fluid density and viscosity are assumed to be independent of the dye concentration. Furthermore, it is assumed that interaction of NaCl, sucrose, and the dye tracer was negligible, and the net effect of the three individual solute species is equal to the sum of the effect of the three components separately.

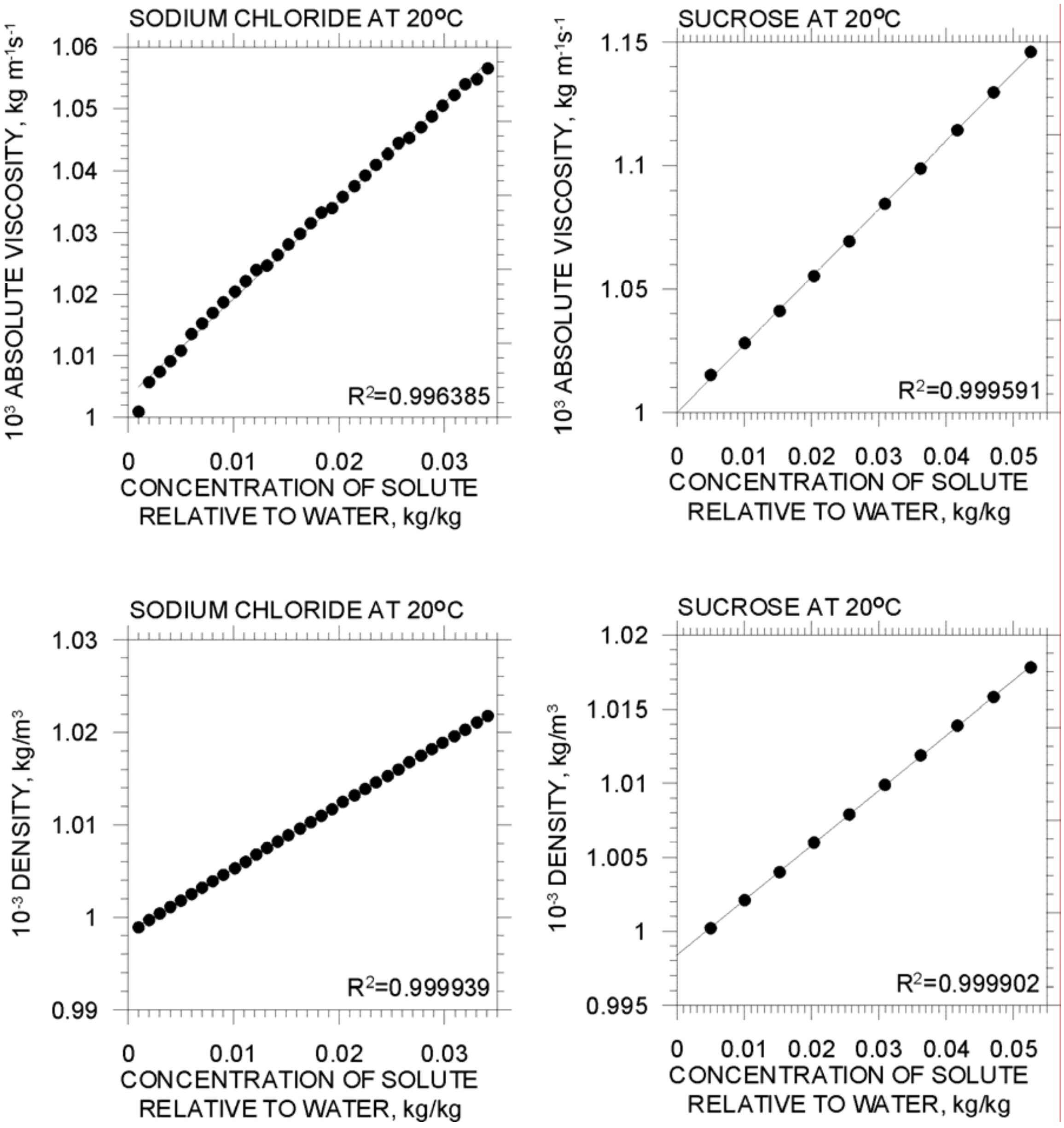


Figure 5.2.2 Absolute viscosity and fluid density relationships with NaCl and sucrose concentrations used in all SUTRA-MS simulations. Data arefrom Weast (1986).

The observed results from the Hele-Shaw experiment (Pringle and others, 2002) are shown in figure 5.2.3. Observed and simulated time is presented as dimensionless and calculated as

$$t^* = t\, D_{NaCl} / H^2 \qquad \textbf{Eq. 5.7}$$

where

t is the elapsed simulation time [sec],
D_{NaCl} is the molecular diffusion of the NaCl solution, and
H is the height of the Hele-Shaw cell.

*t** is based on the dimensionless governing equations of Nield and Bejan (1998). Fluid concentrations are presented as a ratio of the concentration at time *t**, C, and the maximum concentration, C_O, varying from 0.0 to 1.0, for both observed and simulated results.

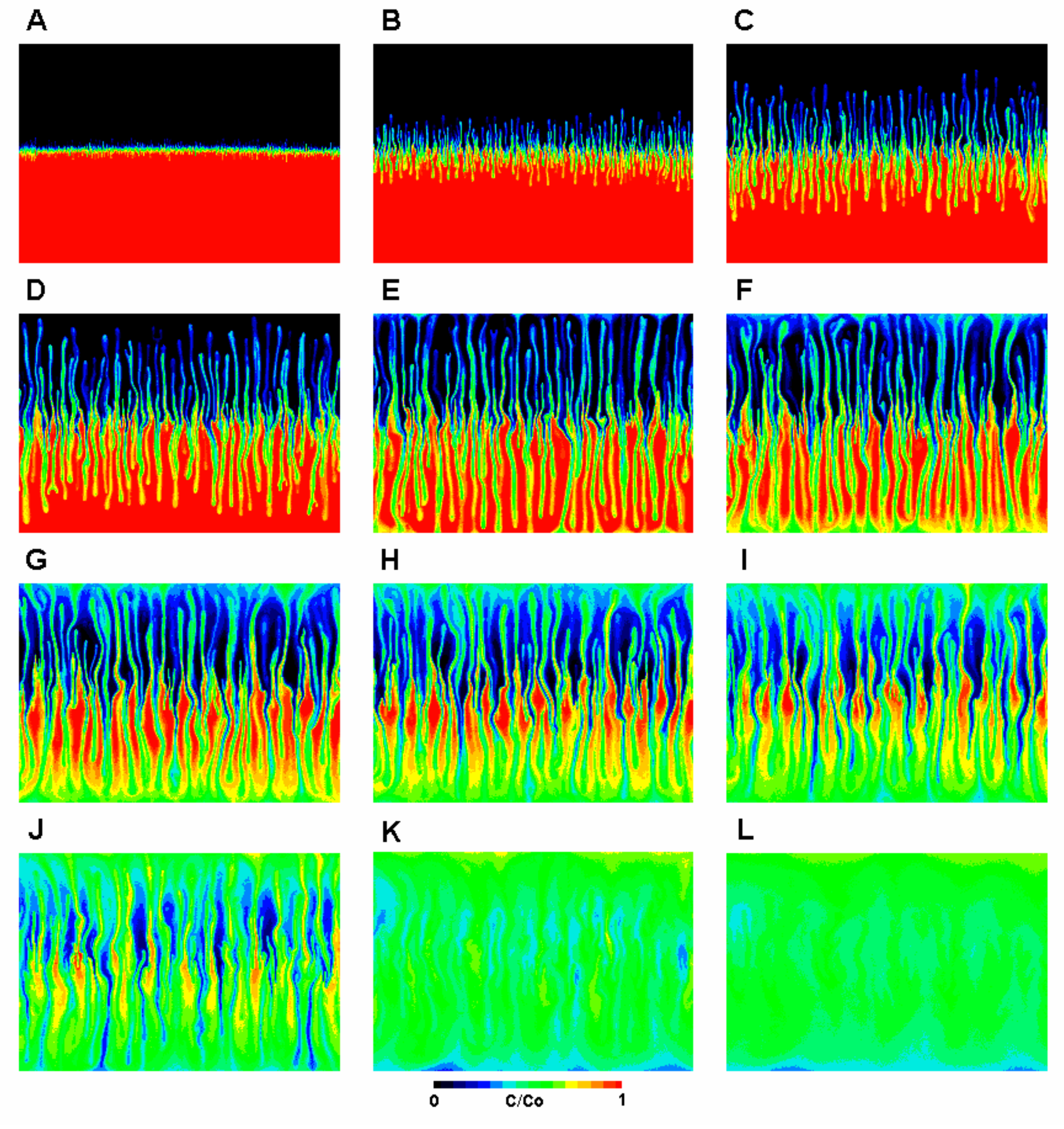


Figure 5.2.3 Observed results from Pringle and others (2002) at (A) $t^* = 4.06\times10^{-5}$, (B) $t^* = 1.29\times10^{-4}$, (C) $t^* = 3.96\times10^{-4}$, (D) $t^* = 3.35\times10^{-4}$, (E) $t^* = 4.35\times10^{-4}$, (F) $t^* = 5.36\times10^{-4}$, (G) $t^* = 6.03\times10^{-4}$, and (H) $t^* = 7.37\times10^{-4}$, (I) $t^* = 8.04\times10^{-4}$, (J) $t^* = 1.04\times10^{-3}$, (K) $t^* = 1.78\times10^{-3}$, and (L) $t^* = 3.19\times10^{-3}$. Color sequence black-blue-green-yellow-orange-red depicts normalized NaCl concentration from 0 to 1.

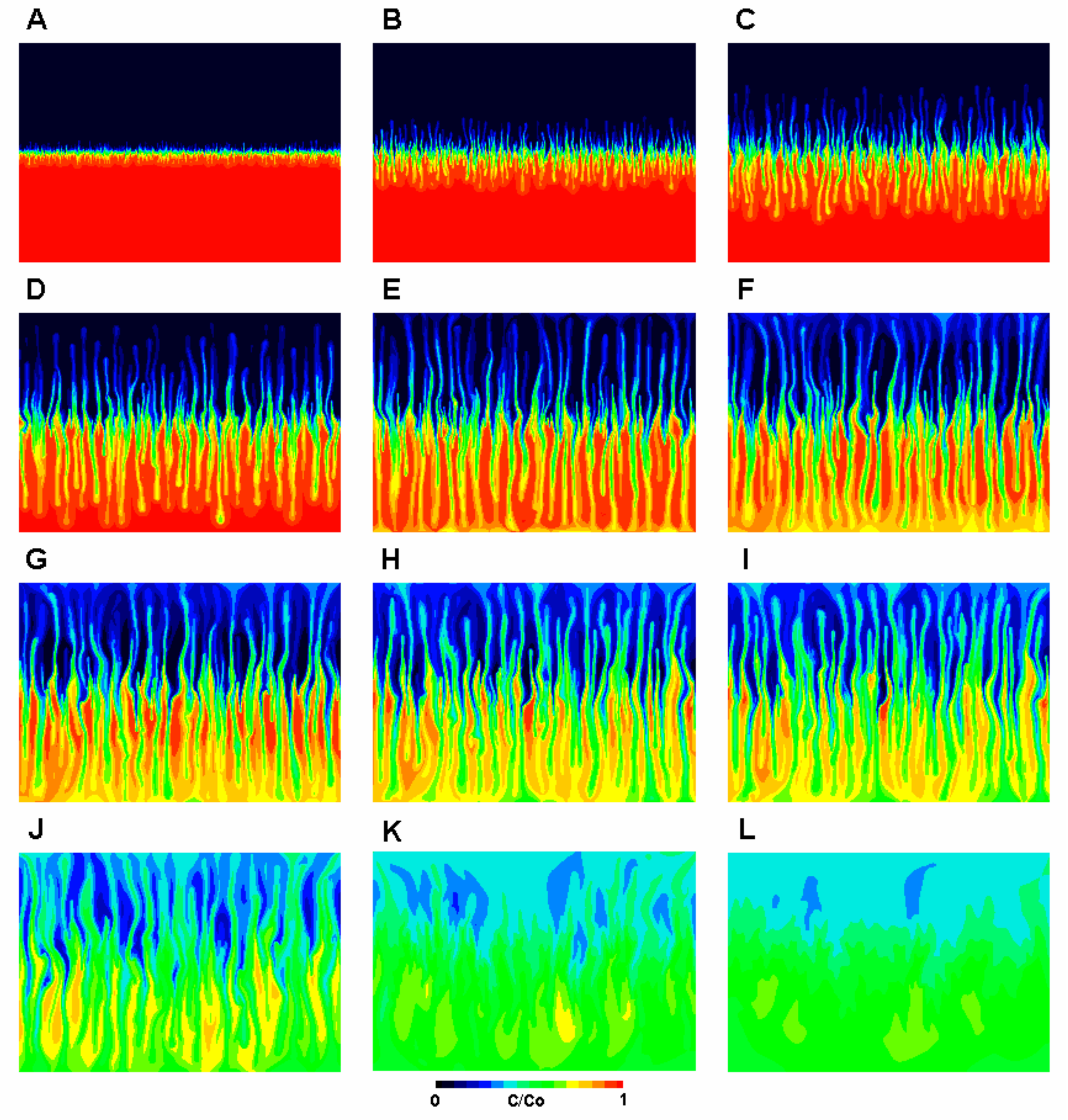


Figure 5.2.4 **Simulated results SUTRA-MS results at (A) t* = 4.06x10^{-5}, (B) t* = 1.29x10^{-4}, (C) t* = 3.96x10^{-4}, (D) t* = 3.35x10^{-4}, (E) t* = 4.35x10^{-4}, (F) t* = 5.36x10^{-4}, (G) t* = 6.03x10^{-4}, and (H) t* = 7.37x10^{-4}, (I) t* = 8.04x10^{-4}, (J) t* = 1.04x10^{-3}, (K) t* = 1.78x10^{-3}, and (L) t* = 3.19x10^{-3}. Color sequence black-blue-green-yellow-orange-red depicts normalized dye concentrations from 0 to 1.**

Simulated results compare well until t^*=3.19x10^{-3} although the rate of finger development appears to be slightly slower than that observed (fig. 5.2.3). Two quantitative measures were used to compare simulated SUTRA-MS concentration fields to observed concentrations.

The first quantitative measure is the normalized vertical length scale that is calculated as

$$h^* = (h/H) \qquad \textbf{Eq. 5.8}$$

where

h is the vertical distance between the horizontally averaged 0.05 and 0.95 normalized dye concentration (C/C_O), and

H is the height of the Hele-Shaw cell.

The normalized length scale, h^*, is a measure of the evolving finger structure. h^* becomes undefined when fingers reach the top and bottom boundaries and concentration profiles invert.

The second quantitative measure is the normalized mass transfer of dye upward across the centerline of the cell in time that is calculated as

$$M^* = (M/M_O) \qquad \textbf{Eq. 5.9}$$

where

M is the total mass of dye above the centerline at time t^*, and

M_O is the mass of dye below the centerline at $t^*=0$.

Quantitative comparisons of the evolution of simulated and observed dye concentrations are shown in figure 5.2.5.

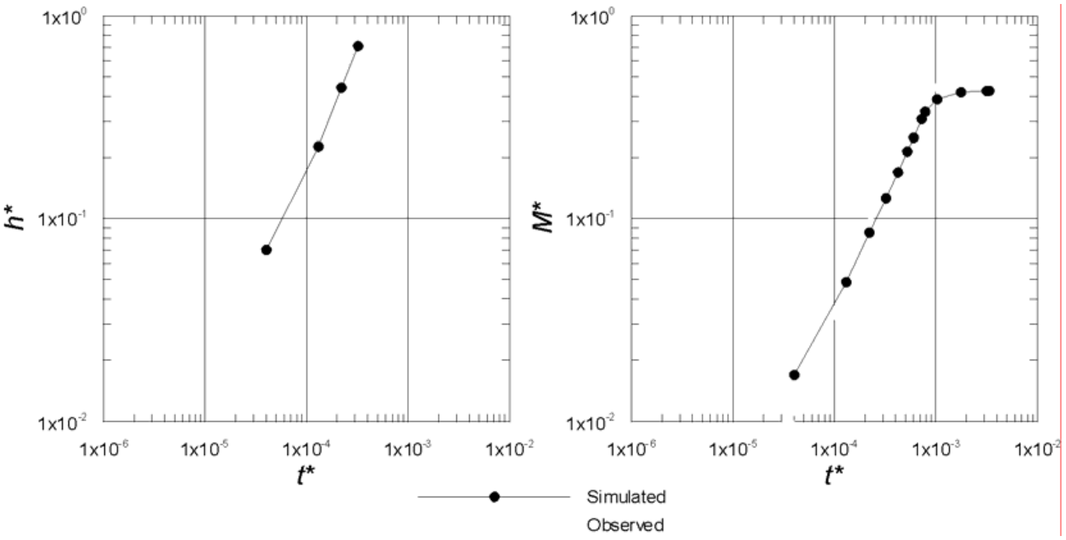


Figure 5.2.5 **Normalized vertical length, $h^*=h/H$, and mass transfer across the center line, $M^*=M/M_o$, as a function of time showing regions of steady growth for the original Hele-Shaw experiment (open gray circles) and the SUTRA-MS simulation (solid black circles and solid black line).**

Figure 5.2.5 quantitatively demonstrates that SUTRA-MS is capable of duplicating the general behavior observed in the Hele-Shaw experiment. Some key aspects that are captured in the SUTRA-MS simulation are the linear increases in h^* at early and mature stages and the timing of the initiation of the rundown stage observed in the M^* data. A noticeable difference is that the simulated normalized mass transfer, M^*, is less than that observed in the Hele-Shaw experiment.

5.3 Density-dependent flow with transport of a non-reactive tracer and zero-order production and transport of a solute to simulate ground-water age

5.3.1 Physical Setup

This problem involves seawater intrusion into a confined aquifer studied in cross section under steady-state conditions (Henry, 1964) with the release of a non-reactive tracer after steady flow conditions are established. Ground-water age is simulated using a third species that is generated internally using a zero-order production rate (Goode, 1996). Freshwater recharge from inland sources flows over seawater and discharges at a vertical sea boundary.

This problem demonstrates the utility of SUTRA-MS to simulate multiple species, some of which have no effect on fluid density. With SUTRA, simulation of more than one transport species (solute or heat) could only be approximated only for variable-density-flow problems , even under steady flow conditions. Approximation of conservative tracer transport in a variable-density flow field required simulation of steady variable-density-flow conditions and use of variable-density-flow results as the initial conditions for a separate, subsequent SUTRA simulation.

SUTRA and SUTRA-MS use the velocity field from the previous time step or the velocity field based on the initial conditions for the first time step to simulate solute transport for the current time step. Because the velocity field used for solute transport is lagged one time step, a simulation that uses the final velocity field from a variable-density problem and a single large time step can be used to simulate conservative transport of a non-density-dependent species. Application of this technique is limited: only a single time step can be simulated because tracer concentrations would alter the variable-density velocity field on the second time step.

Results from the cross-section SUTRA-MS simulation (two-dimensional simulation) also are compared to a three-dimensional version of the problem where the model domain has been extended a factor of 20 in the y-dimension. The comparison of two- and three-dimensional SUTRA-MS results is meant to show differences in non-reactive tracer concentrations that occur when dispersion in the y-dimension is permitted (three-dimensional simulation).

The intrusion problem is non-linear and is solved by gradually approaching steady state with a series of time steps. Initially, there is no saltwater in the aquifer. At time zero, saltwater begins to intrude the freshwater system by flowing under the freshwater from the sea boundary. The intrusion of saltwater is caused by differences in saltwater and freshwater density and hydrostatic seawater boundary conditions specified at the sea boundary.

Dimensions of the two-dimensional problem are based on the SUTRA setup described in Voss (1984), but length scales have been increased by a factor of 100. A stabilization simulation time of 13.89 days was selected because this is sufficient time for the variable-density flow field and chloride concentrations to reach steady state at the scale simulated. A non-reactive solute is released at 13.89 days from three constant concentration nodes. A total tracer-transport-simulation time of 13.89 days is sufficient time for the non-reactive solute to move more than half the total length of the problem in the x-direction.

5.3.2 Simulation Setup

The mesh for the two-dimensional problem consists of 100 by 50 elements, each of size 2.0 m by 2.0 m (fig. 5.3.1). The mesh for the three-dimensional problem consists of 100 by 50 by 10 elements, each of size 2.0 m by 2.0 m by 2.0 m. The three-dimensional problem uses the same discretization in the x-z plane as shown in figure 5.3.1 and has a total length of 40-m in the y-direction.

A constant time-step length of 100 minutes was used for all SUTRA-MS simulations, and 400 time steps were taken. Fluid flow and solute transport were solved during each time step in the two- and three-dimensional SUTRA-MS simulations (NUCYC=NPCYC=1).

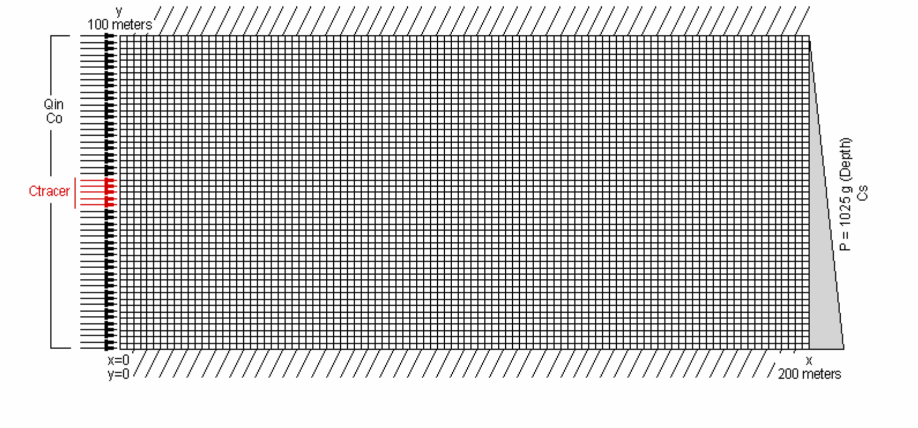


Figure 5.3.1 SUTRA finite-element mesh and boundary conditions.

5.3.3 Parameters

$k = 1.02041 \times 10^{-9}$ [m^2] based on K = 0.01 [m / s]

$\varepsilon = 0.35$ [-]

$C_S = 0.0357$ [kg$_{\text{dissolved solids}}$ / kg$_{\text{fluid}}$]

$\rho_S = 1024.99$ [kg / m^3]

$\rho_{SOLID} = 2600.$ [kg / m^3]

$\frac{\partial \rho}{\partial C_{SOLUTE}} = 700.$ [kg$_{\text{fluid}}^2$ / (kg$_{\text{NaCl}}$ m^3)]

$\alpha_{LSOLUTE} = 0.0$ [m]

$\frac{\partial \rho}{\partial C_{Tracer}} = 0.00$ [kg$_{\text{fluid}}^2$ / (kg$_{\text{Tracer}}$ m^3)]

$\alpha_{LTracer} = 0.5$ [m]

$\frac{\partial \rho}{\partial C_{GWAge}} = 0.00$ [kg$_{\text{fluid}}^2$ / (kg$_{\text{GWAge}}$ m^3)]

$\alpha_{LGWAge} = 0.000$ [m]

$D_{SOLUTE} = 18.8571 \times 10^{-4}$ [m^2 / s]

$D_{GWAge} = 18.8571 \times 10^{-4}$ [m^2 / s]

$B = 2.0$ [m]

$|\mathbf{g}| = 9.8$ [m / s^2]

$C_O = 0.000$ [kg$_{\text{dissolved solids}}$ / kg$_{\text{fluid}}$]

$\rho_O = 1000.$ [kg / m^3]

$Q_{IN} = 6.6$ [kg / s m]

$\mu = 1 \times 10^{-3}$ [kg / (m s)]

$\alpha_{TSOLUTE} = 0.0$ [m]

$C_{Tracer} = 100.0$ [kg$_{\text{Tracer}}$ / kg$_{\text{seawater}}$]

$\alpha_{TTracer} = 0.5$ [m]

$C_{GWAge} = 0.0$ [kg$_{\text{GWAge}}$ / kg$_{\text{seawater}}$]

$\alpha_{TGWAge} = 0.0000$ [m]

$D_{Tracer} = 0.0$ [m^2 / s]

$PRODF0_{GWAge} = 1.667 \times 10^{-2}$ [1 / s]

5.3.4 Boundary Conditions

No-flow conditions occurred across the top and bottom boundaries. A source of freshwater was implemented by using source nodes that injected freshwater at rate Q_{IN} with concentration C_O at the left vertical boundary. A total freshwater source of 13.2 kg/s (kilograms per second), distributed over 51 nodes, was applied at the left vertical boundary of the two-dimensional problem. A total freshwater source of 264 kg/s, distributed over 1,071 nodes, was applied at the left vertical face of the three-dimensional problem. The right vertical boundary was held at hydrostatic pressure, assuming a constant density equal to seawater, through use of specified pressure nodes. Any water entering through these nodes has the concentration, C_S, of seawater.

Five specified concentration nodes at x = 0.0 m and z = 47.0 to 52.0 m were used to simulate the release of the non-reactive tracer in the two-dimensional SUTRA-MS problem (red arrows on figure 5.3.1). In the three-dimensional problem, five specified concentration nodes at x = 0.0 m, y = 40.0 m, and z = 47.0 to 52.0 m were used to simulate the release of the non-reactive tracer. In both problems, specified concentration nodes had a concentration of 0.0 kg$_{\text{dissolved solids}}$/kg$_{\text{fluid}}$ from 0 to 13.89 days and 100.0 kg$_{\text{dissolved solids}}$/kg$_{\text{fluid}}$ from 13.89 to 27.78 days. A high tracer concentration was simulated after 13.89 days so that the total tracer mass was sufficient to develop a plume that extended a significant distant from the inflow boundary in both problems.

No boundary conditions were specified for species simulating ground-water age using zero-order production, except that the age of the incoming fluid was specified to be zero.

5.3.5 Initial Conditions

Hydrostatic pressures based on an initially freshwater aquifer with a constant concentration of C_O were used for all simulations. Initial concentrations for the non-reactive tracer and the species simulating ground-water age were set to zero in all simulations.

5.3.6 Results

In Henry's solution, dispersion is represented by a constant, large diffusion coefficient, rather than by a velocity-dependent dispersivity. Two different values of diffusivity have been used to test simulators against Henry's solution (Voss, 1984). The total dispersion coefficient of Henry (1964), D, is equivalent to the product of porosity and molecular diffusivity in SUTRA (see eq. 5.1).

Henry's results are given for his non-dimensional parameters: ξ=2.0, b=0.1, a≈0.264 (Henry, 1964, p. C80, fig. 34). In order to match Henry's parameters using simulation parameters as listed above, values of $D = 6.6\text{x}10^{-4}$ m^2/s and $D_{SOLUTE} = 18.8571\text{x}10^{-4}$ m^2/s are required. Because this example is not a comparison of results with simulators other than SUTRA-MS, D and D_{SOLUTE} values evaluated are limited to the values listed above.

Figure 5.3.2 shows that percent-seawater contours are essentially identical at 13.89 and 27.78 days in the two-dimensional SUTRA-MS simulation, and the variable-density flow field has essentially achieved steady-state conditions by 13.89 days. Because the variable-density flow field has achieved steady conditions by 13.89 days, use of the simulated pressure field is appropriate for simulation of conservative tracer transport in a steady variable-density flow field. Percent-seawater contours at 27.78 days for the three-dimensional SUTRA-MS simulation also are also shown in figure 5.3.2 and are comparable to contours in the two-dimensional SUTRA-MS simulation.

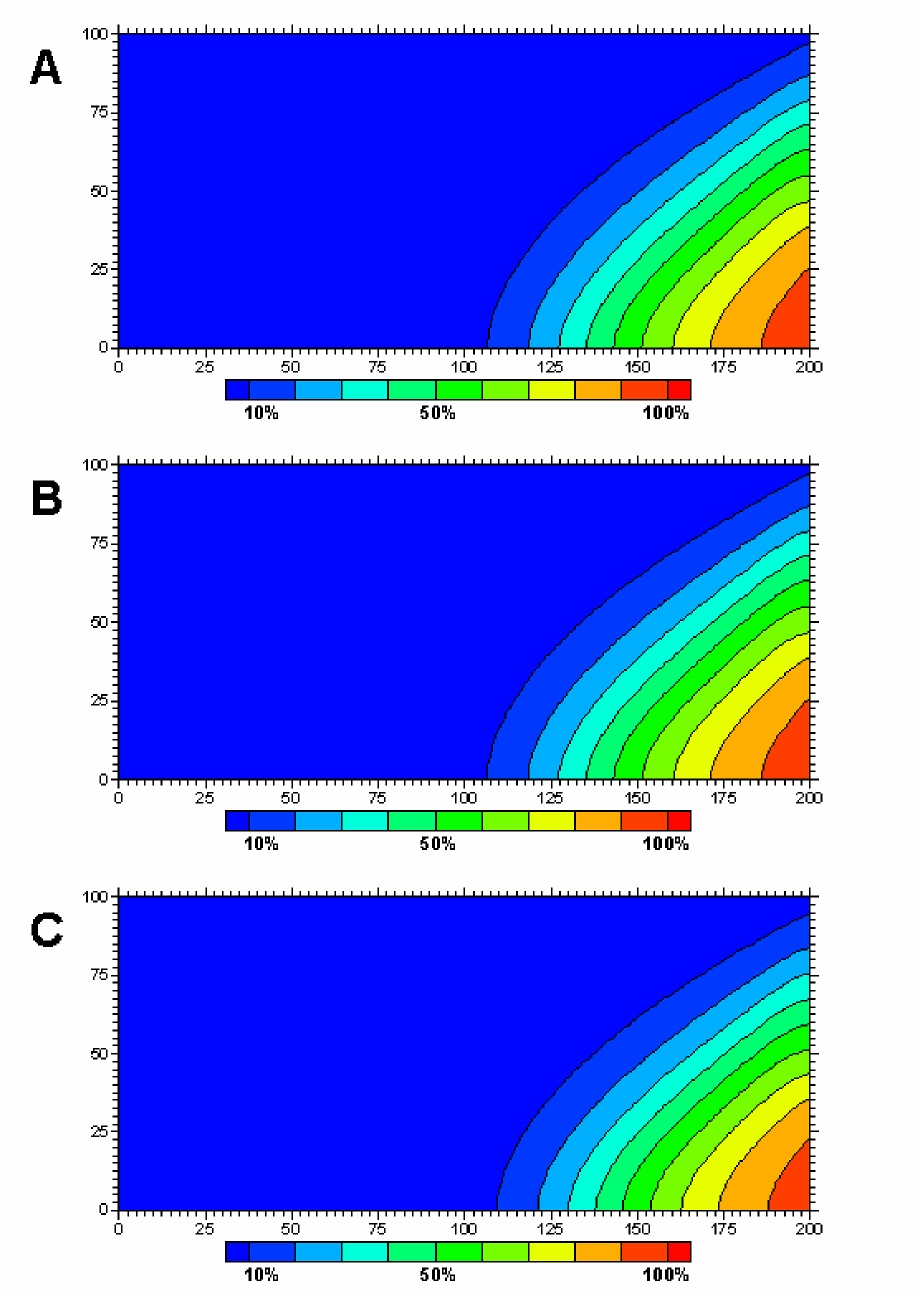


Figure 5.3.2 **Simulated percent-seawater contours from the 2D SUTRA-MS simulation after (A) 13.89 and (B) 27.78 days and the (C) 3D SUTRA-MS simulation after 27.78 days.**

A velocity-dependent dispersivity was used for the non-reactive tracer. Molecular dispersion was assumed to be zero (0) for the non-reactive tracer. A longitudinal and transverse dispersivity of 0.5 and 0.5 m, respectively, were used for the non-reactive tracer.

Simulated SUTRA-MS non-reactive tracer-concentration contours for the two- and three-dimensional simulations are shown in figure 5.3.3. Non-reactive tracer-concentration contours for the three-dimensional SUTRA-MS simulation are significantly more dispersed in the y-dimension and slightly less dispersed in the z-dimension than in the two-dimensional SUTRA-MS simulation. As a result, the non-reactive tracer does not extend as far into the problem domain in the three-dimensional SUTRA-MS problem.

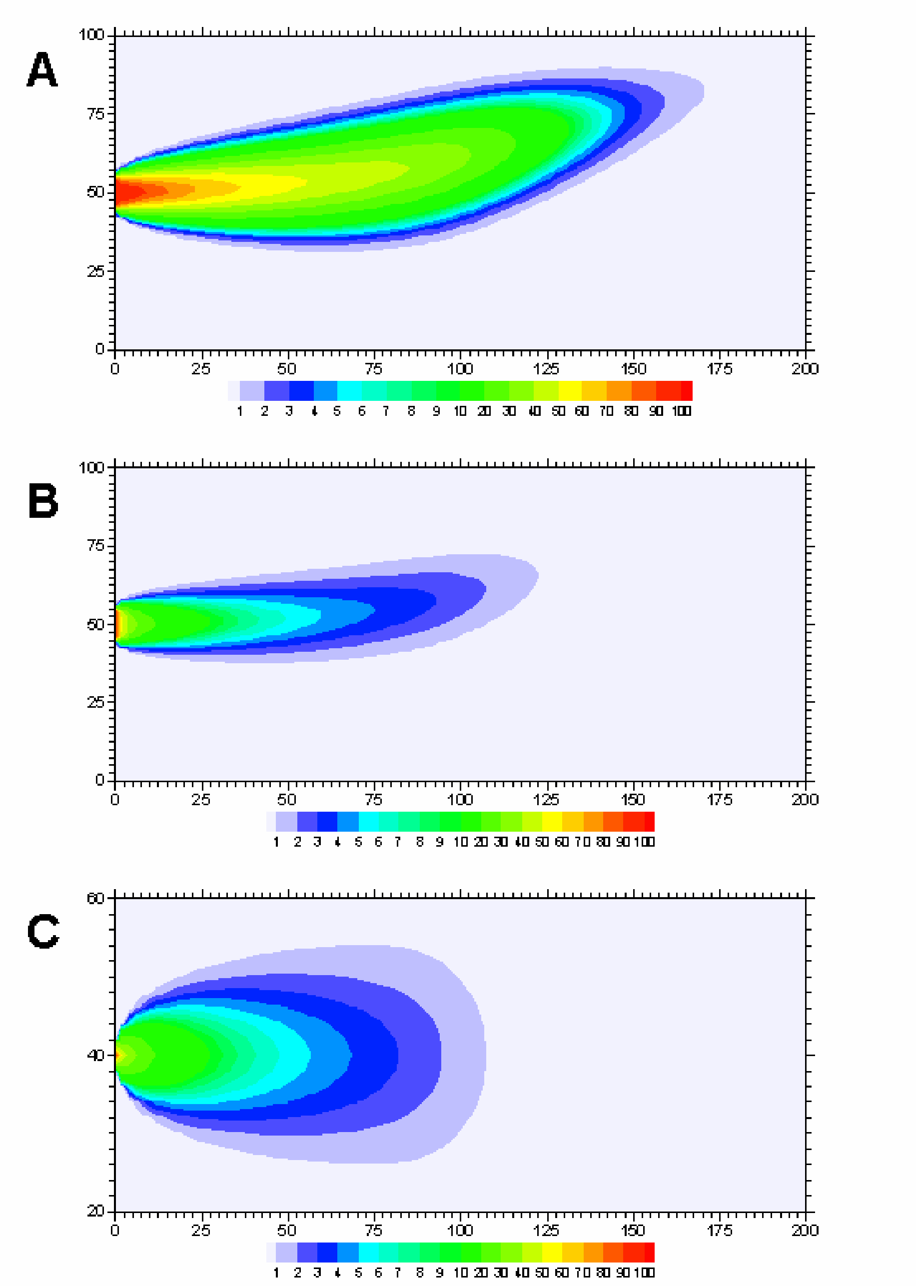


Figure 5.3.3 **Simulated non-reactive tracer-concentration contours from (A) 2D SUTRA-MS, (B) 3D SUTRA-MS in XZ Plane at y=40 m, and (C) 3D SUTRA-MS in XY Plane at z=50 m.**

Figure 5.3.3 shows the simulated ground-water age after 27.78 days for the two-dimensional SUTRA-MS simulation. This is done to further illustrate the utility of SUTRA-MS. Ground-water age was simulated with a third species (that does not affect fluid density) having a zero-order production rate of 1.667×10^{-2} sec^{-1} (1 min^{-1}).

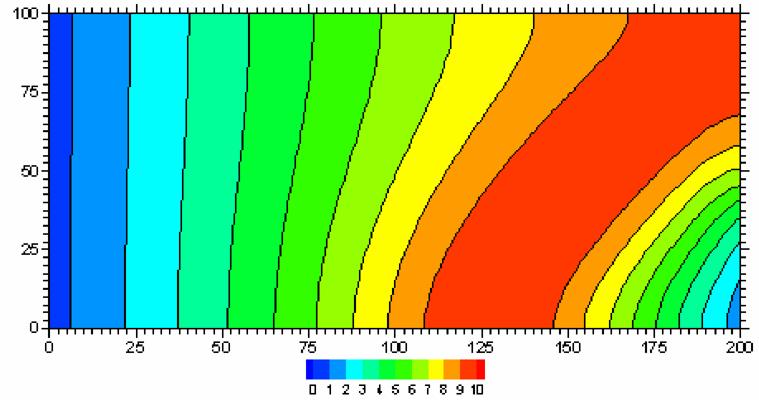


Figure 5.3.4 **Simulated ground-water age, in days, from SUTRA-MS.**

6.0 SUTRA-MS Data Input

This section contains the data input format for all required SUTRA-MS files. It also includes the input data format of all options available in SUTRA-MS.

6.1 SUTRA Input Data List

<u>List of Input Data for the File Assignment Input File, "SUTRA.FIL"</u>

Model Series:	SUTRA-MS
Model Version:	2D3DMS.1

The file "SUTRA.FIL" contains file assignments (one line for each assignment) in the following format:

Variable	Type	Description
FTYPE	Character	File type. Valid values are as follows: 'INP' = ".inp" input file 'ICS' = ".ics" input file 'LST' = ".lst" output file 'RST' = ".rst" output file 'NOD' = ".nod" output file 'ELE' = ".ele" output file 'OBS' = ".obs" output file 'TBC' = ".tbc" input file 'OTM' = ".otm" input file 'ATS' = ".ats" input file 'SOB' = ".sob" input file 'ZON' = ".zon" input file
IUNIT	Integer	FORTRAN unit number to be assigned to the file.
FNAME	Character	Full name of the file.

<u>Note:</u>

Assignments for the ".nod", ".ele", ".obs", ".tbc", ".otm", ".ats", ".sob", and ".zon" files are optional. If any of these assignments are omitted, the corresponding output files will not be created by SUTRA or the optional SUTRA-MS functionality will not be used.

Assignments for the remaining input and output files are required. Assignments may be listed in any order.

Example:

```
‘INP’ 50    ‘project.inp’
‘ICS’ 55    ‘project.ics’
‘LST’ 60    ‘project.lst’
‘RST’ 66    ‘project.rst’
‘NOD’ 70    ‘project.nod’
‘ELE’ 80    ‘project.ele’
‘OBS’ 90    ‘project.obs’
‘TBC’ 91    ‘project.tbc’
‘OTM’ 92    ‘project.otm’
‘ATS’ 93    ‘project.ats’
‘SOB’ 94    ‘project.sob’
‘ZON’ 95    ‘project.zon’
```

6.2 General Format of the ".inp", ".ics", ".tbc", ".otm", ".ats", ".sob", and ".zon" Input Files

SUTRA reads the ".inp", ".ics", ".tbc", ".otm", ".ats", ".sob", and ".zon" input files in a list-directed fashion (except for DATASET 1 of the ".inp" file):

a. Input data appearing on the same line should be space- or tab-separated.

b. Any data that are not optional must be given values in the input file (blanks are not sufficient).

c. Enclose input variables of "character" type in single quotation marks (unless specified otherwise) to provide maximum compatibility across computing platforms.

d. Comment lines may be placed within the ".inp", ".ics", ".tbc", ".otm", ".ats", ".sob", and ".zon" files, subject to the following restrictions:

- Comment lines must have a pound sign, #, in the first column.
- Comments lines can be placed before or after any data set.
- Comments lines may not be placed within a data set (such as in the middle of a list of specified pressures). For this purpose, data sets with letter designations (such as 2A and 2B) count as distinct data sets – comment lines may be placed between them.
- Comments lines may not be placed within any of the "restart" information that follows DATASET 3 when a ".rst" (restart) file is used as a ".ics" (initial conditions) file.

Comments (or any text) can be appended to the end of any line of input data, provided all the required parameters have first been entered on that line. Be sure to leave at least one space or tab between the last required parameter and the beginning of the comment. All input lines unique to SUTRA-MS are listed in blue.

6.3 List of Input Data for the ".inp" (Main Input) File

Model Series: SUTRA-MS
Model Version: 2D3DMS.1

DATASET 1: Output Heading (two lines)

Variable	Type	Description
TITLE1	Character	First line of heading for the input data set.
TITLE2	Character	Second line of heading for the input data set. The first 80 characters of each line are printed as a heading on SUTRA output. In this data set, the character inputs need not be enclosed in quotation marks.

DATASET 2A: Simulation Type (one line)

Variable	Type	Description
SIMULA	Character	Two to three words. The first word must be "SUTRA". The second word indicates the type of transport, and must be either "ENERGY", "SOLUTE", "ENERGY SOLUTE", or "MULTI SOLUTE". Any additional words are ignored by SUTRA.

Examples:

For energy transport simulation, one may write the following:

```
'SUTRA ENERGY TRANSPORT'
```

For solute transport simulation, one may write the following:

```
'SUTRA SOLUTE TRANSPORT'
```

For multi-species energy and solute transport, one may write the following:

```
'SUTRA ENERGY SOLUTE TRANSPORT' or
'SUTRA SOLUTE ENERGY TRANSPORT'
```

For multi-species solute transport, one may write the following:

```
'SUTRA SOLUTE MULTI TRANSPORT' or
'SUTRA MULTI SOLUTE TRANSPORT'
```

In these examples, the word "TRANSPORT" is ignored by SUTRA-MS but is included to make the input more readable.

DATASET 2B: Mesh Structure (four lines)

Variable	Type	Description
Line 1:		
MSHSTR	Character	Two words. The first word indicates the dimensionality of the mesh, and must be either “2D” or “3D”. The second word indicates the regularity of the mesh, and must be either “REGULAR”, “BLOCKWISE”, or “IRREGULAR”. Any additional words are ignored by SUTRA. (“3D IRREGULAR” meshes are not currently supported.)
NN1	Integer	For a REGULAR mesh, the number of nodes in the first numbering direction. May be omitted if the mesh is IRREGULAR.
NN2	Integer	For a REGULAR mesh, the number of nodes in the second numbering direction. May be omitted if the mesh is IRREGULAR.
NN3	Integer	For a REGULAR, 3-D mesh, the number of nodes in the third numbering direction. May be omitted if the mesh is IRREGULAR and/or 2-D.
Omit lines 2 – 4 if mesh is NOT ‘BLOCKWISE’.		
Line 2:		
NBLK1	Integer	Number of blocks in the first numbering direction.
LDIV1	Integer	A list of the number of elements into which to divide each of the NBLK1 blocks along the first numbering direction.
Line 3:		
NBLK2	Integer	Number of blocks in the second numbering direction.
LDIV2	Integer	A list of the number of elements into which to divide each of the NBLK2 blocks along the second numbering direction.
Line 4:		
NBLK3	Integer	Number of blocks in the third numbering direction.

LDIV3	Integer	A list of the number of elements into which to divide each of the NBLK3 blocks along the third numbering direction.

Note:

A BLOCKWISE mesh is a special type of REGULAR mesh that is created by the preprocessor SUTRAPREP. The additional data associated with a BLOCKWISE mesh are not currently used by SUTRA and are included only for postprocessing purposes.

Examples:

For a 3-D, regular (logically-rectangular), 10x20x30-node mesh, one may write the following:

```
   ‘3D REGULAR MESH’  10  20  30
```

For a 2-D, irregular mesh, one may write the following:

```
   ‘2D IRREGULAR MESH’
```

In these examples, the word “MESH” is ignored by SUTRA-MS but is included to make the input more readable.

DATASET 3A: Simulation Control Numbers (one line)

Variable	Type	Description
NN	Integer	Exact number of nodes in finite-element mesh.
NE	Integer	Exact number of elements in finite-element mesh.
NPBC	Integer	Exact number of nodes at which pressure is a specified constant value or function of time.
MNUBC	Integer	Maximum number of nodes at which temperature or concentration is a specified constant value or function of time for each simulated species.
NSOP	Integer	Exact number of nodes at which a fluid source/sink is a specified constant value or function of time.
MNSOU	Integer	Maximum number of nodes at which an energy or solute mass source/sink is a specified constant value or function of time for each simulated species.
NOBS	Integer	Exact number of nodes at which observations will be made. Set to zero for no observations or if using the optional specified observation locations (".sob") data set.
NSPE	Integer	Number of species simulated (includes energy and all solute species). <u>O M I T</u> for single-species transport simulations.

DATASET 3B: Species Control Numbers (one line for each of species specified in DATASET 3A, plus one line)

O M I T when only one transport species is being simulated (NSPE = 1).

Variable	Type	Description
Lines 1 to NSPE:		
NSN	Integer	Species Number.
SPNAME	Character	Ten-character descriptive name for species. The energy transport species must be named 'ENERGY', if energy transport is being simulated.
NUBC	Integer	Exact number of nodes at which temperature or concentration is a specified constant value or function of time for simulated species NSN. Must be less than or equal to MNUBC specified in DATASET 3A.
NSOU	Integer	Exact number of nodes at which an energy or solute mass source/sink is a specified constant value or function of time for simulated species NSN. Must be less than or equal to MNSOU specified in DATASET 3A.
Last line:		
	Integer	Placed immediately following all DATASET 3B lines. Line must begin with the integer 0.

Examples:

For a data set to simulate two transport species (*e.g.*, Energy and Brine), one may write the following:

```
##
##  DATASET 3B:  Species Names and number of boundary conditions for each
   species
##    [NSN]    [SPNAME]    [NUBC]      [NSOU]
         1     'ENERGY'        63           0
         2     'BRINE'          0          10
         0
```

DATASET 4: Simulation Mode Options (one line)

Variable	Type	Description
CUNSAT	Character	One word. Set to 'UNSATURATED' for simulation of unsaturated/saturated ground-water flow. Set to 'SATURATED' for simulation of only saturated flow. When UNSATURATED flow is allowed, the unsaturated flow functions must be programmed by the user in subroutine UNSAT.
CSSFLO	Character	One word. Set to 'TRANSIENT' for simulation of transient ground-water flow. Set to 'STEADY' for simulation of steady-state flow. If fluid density is to change with time, then TRANSIENT flow must be selected.
CSSTRA	Character	One word. Set to 'TRANSIENT' for simulation of transient solute or energy transport. Set to 'STEADY' for simulation of steady-state transport. Note that steady-state transport requires a steady-state flow field. So, if CSSTRA = 'STEADY', then also set CSSFLO = 'STEADY'.
CREAD	Character	One word. Set to 'COLD' to read initial condition data (".ics" file) for a "cold start" (the very first time step of a simulation or series of restarted simulations). Set to 'WARM' to read initial condition data (".ics" file) for a "warm restart" of a simulation using data that were previously stored by SUTRA in a ".rst" file.
ISTORE	Integer	To store results for each ISTORE time step in the ".rst" file for later use as initial conditions on a restart, set to +1 or greater value. To cancel storage, set to 0. This option is recommended as a backup for storage of results of intermediate time steps during long simulations. Should the execution halt unexpectedly, it may be restarted with initial conditions consisting of results of the last successfully completed time step stored in the ".rst" file. When ISTORE > +1, results are always stored in the ".rst" file after the last time step of a simulation regardless of whether the step is an even multiple of ISTORE.

Any extra words included in the character variables in this data set are ignored by SUTRA.

Example:

To simulate saturated, steady-state ground-water flow with transient solute or energy transport from a cold start, storing intermediate results every 10 time steps, one may write the following:

```
'SATURATED FLOW' 'STEADY FLOW'  'TRANSIENT TRANSPORT' 'COLD START' 10
```

In this example, the words “FLOW”, “TRANSPORT”, and “START” are ignored by SUTRA but are included to make the input more readable.

DATASET 5: Numerical Control Parameters (one line)

Variable	Type	Description
UP	Real	Fractional upstream weight for stabilization of numerical oscillations due to highly advective transport or unsaturated flow. UP may be given any value from 0.0 to +1.0. UP=0.0 implies no upstream weighting (Galerkin method). UP=0.5 implies 50% upstream weighting, UP=1.0 implies full (100%) upstream weighting. Recommended value is zero. WARNING: Upstream weighting increases the local effective longitudinal dispersivity of the simulation by approximately UP*(ΔL)/2 where ΔL is the local distance between element sides along the direction of flow. Note that the amount of this increase varies from place to place depending on flow direction and element size. Thus a non-zero value for UP actually changes the value of longitudinal dispersivity used by the simulation, and also broadens otherwise sharp concentration, temperature or saturation fronts.
GNUP	Real	Pressure boundary condition, “conductance”. A high value causes SUTRA simulated and specified pressure values at specified pressure nodes to be equal in all significant figures. A low value causes simulated pressure to deviate significantly from specified values. The ideal value of GNUP causes simulated and specified pressures to match in the largest six or seven significant figures only, and deviate in the rest. Trial-and-error is required to determine an ideal GNUP value for a given simulation by comparing values specified with those calculated at the appropriate nodes for different values of GNUP. An initial guess of 0.01 is suggested.
GNUU	Real	Concentration/temperature boundary condition factor. A high value causes SUTRA simulated values and specified values at specified concentration/temperature nodes to be equal in all significant figures. A low value causes simulated values to deviate significantly from specified values. The ideal value of GNUU causes simulated

and specified concentrations or temperatures to match in the largest six or seven significant figures only, and deviate in the rest. Trial-and-error is required to determine an ideal GNUU value for a given simulation by comparing specified values with those calculated at the appropriate nodes for different values of GNUU. An initial guess of 0.01 is suggested.

For multi-species transport simulations, enter a GNUU value for each species. GNUU values must be entered in order for species 1 to NSPE, as specified in DATASET 3B.

Examples:

For a data set to simulate two transport species (*e.g.*, Energy and Brine), one may write the following:

```
##
##  DATASET 5:  Numerical Control Parameters
##         [UP]         [GNUP]       [GNUU1]       [GNUU2]
           0.0            0.01          0.01          0.01
```

or for a data set to simulate a single transport species, one may write the following:

```
##
##  DATASET 5:  Numerical Control Parameters
##         [UP]         [GNUP]       [GNUU]
            0.0           0.01         0.01
```

DATASET 6: Temporal Control and Solution Cycling Data (one line)

Variable	Type	Description
ITMAX	Integer	Maximum allowed number of time steps in simulation.
DELT	Real	Duration of initial time step. [s]
TMAX	Real	Maximum allowed simulation time. [s] SUTRA time units are always in seconds. Other time measures are related as follows: [min] = 60. [s] [h] = 60. [min] [d] = 24. [h] [week] = 7. [d] [mo] = 30.4375 [d] [yr] = 365.250 [d]
ITCYC	Integer	Number of time steps in time-step change cycle. A new time-step size is begun at time steps numbered: 1+ n*(ITCYC).
DTMULT	Real	Multiplier for time-step change cycle. New time step size is: (DELT)*(DTMULT).
DTMAX	Real	Maximum allowed size of time step when using time-step multiplier. Time-step size is not allowed to increase above this value.
NPCYC	Integer	Number of time steps in pressure solution cycle. Pressure is solved on time steps numbered: n*(NPCYC), as well as on the initial time step.
NUCYC	Integer	Number of time steps in temperature/concentration solution cycle. Transport equation is solved on time steps numbered: n*(NUCYC) as well as on the initial time step. Either NPCYC or NUCYC must be set to 1.

DATASET 7A: Iteration Controls for Resolving Non-linearities (one line)

Variable	Type	Description
ITRMAX	Integer	Maximum number of iterations allowed per time step to resolve non-linearities. Set to a value of +1 for non-iterative solution. Non-iterative solution may be used for saturated aquifers when density variability of the fluid is small, or for unsaturated aquifers when time steps are chosen to be small.
RPMAX	Real	Absolute iteration convergence criterion for pressure solution. Pressure solution has converged when largest pressure change from the previous iteration's solution of any node in mesh is less then RPMAX. May be omitted for non-iterative solution.
RUMAX	Real	Absolute iteration convergence criterion for transport solution. Transport solution has converged when largest concentration or temperature change from the previous iteration's solution of any node in mesh is less than RUMAX. May be omitted for non-iterative solution. For multi-species transport simulations, enter a RUMAX value for each species. RUMAX values must be entered in order for species 1 to NSPE, as specified in DATASET 3B.

Examples:

For a data set to simulate two transport species (*e.g.*, Energy and Brine), one may write the following:

```
##
##  DATASET 7A:  Iteration and Matrix Solver Controls
##    [ITRMAX]    [RPMAX]     [RUMAX1]    [RUMAX2]
           100        0.1          0.1         0.1
```

or for a data set to simulate a single transport species, one may write the following:

```
##
##  DATASET 7A:  Iteration and Matrix Solver Controls
##    [ITRMAX]    [RPMAX]     [RUMAX]
           100        0.1         0.1
```

DATASET 7B: Matrix Solver Controls for Pressure Solution (one line)

Variable	Type	Description
CSOLVP	Character	Name of solver to be used to obtain the pressure solution. Valid values are as follows: 'DIRECT' = Banded Gaussian elimination (direct) 'CG' = IC-Preconditioned Conjugate Gradient (CG) 'GMRES' = ILU-Preconditioned Generalized Minimum Residual (GMRES) 'ORTHOMIN' = ILU-Preconditioned ORTHOMIN If the DIRECT solver is used, it must be used for both the pressure and the transport solution; if either CSOLVP or CSOLVU (DATASET 7C) is set to 'DIRECT', then the other must also be set to 'DIRECT'. If an ITERATIVE solver is used, the recommended value is 'CG' if no upstream weighting is used (UP=0.). If upstream weighting is used (UP>0.), use 'GMRES' or 'ORTHOMIN'.
ITRMXP	Integer	Maximum number of solver iterations during pressure solution. May be omitted if the DIRECT solver is used.
ITOLP	Integer	Type of convergence criterion to be applied to solver iterations during pressure solution. Recommended value is 0. (For other options, see SLAP solver documentation.) May be omitted if the DIRECT solver is used.
TOLP	Real	Convergence tolerance for solver iterations during pressure solution. May be omitted if the DIRECT solver is used.
NSAVEP	Integer	For GMRES and ORTHOMIN solvers, the number of direction vectors to save and orthogonalize against during pressure solution. Recommended value is 10. (For other options, see SLAP solver documentation.) For CG solver, set to any integer value. May be omitted if the DIRECT solver is used.

DATASET 7C: Matrix Solver Controls for Transport Solution (one line or NSPE lines plus one for multi-species transport simulations)

Variable	Type	Description
Lines 1 to NSPE:		
NSN	Integer	Species Number. OMIT if a single species transport simulation.
CSOLVU	Character	Name of solver to be used to obtain the transport solution for species NSN. Valid values are as follows: 'DIRECT' = Banded Gaussian elimination (direct) 'CG' = IC-Preconditioned Conjugate Gradient (CG) 'GMRES' = ILU-Preconditioned Generalized Minimum Residual (GMRES) 'ORTHOMIN' = ILU-Preconditioned ORTHOMIN If the DIRECT solver is used, it must be used for both the pressure and the transport solution; if either CSOLVU or CSOLVP (DATASET 7B) is set to ‘DIRECT’, then the other must also be set to ‘DIRECT’. If an ITERATIVE solver is used, the recommended values are ‘GMRES’ and ‘ORTHOMIN’.
ITRMXU	Integer	Maximum number of solver iterations during transport solution for species NSN. May be omitted if the DIRECT solver is used.
ITOLU	Integer	Type of convergence criterion to be applied to solver iterations during transport solution for species NSN. Recommended value is 0. (For other options, see SLAP solver documentation.) May be omitted if the DIRECT solver is used.
TOLU	Real	Convergence tolerance for solver iterations during transport solution for species NSN. May be omitted if the DIRECT solver is used.

NSAVEU	Integer	For GMRES and ORTHOMIN solvers, the number of direction vectors to save and orthogonalize against during transport solution for species NSN. Recommended value is 10. (For other options, see SLAP solver documentation.) For CG solver, set to any integer value. May be omitted if the DIRECT solver is used.
STARTU	Real	Starting time of initiation of transport of species NSN. OMIT if a single species transport simulation and OPTIONAL if simulating multiple species. [s]

Last line:

	Integer	Placed immediately following all DATASET 7B lines. Line must begin with the integer 0. OMIT if a single species transport simulation.

Examples:

For a data set to simulate two transport species (*e.g.*, Energy and Brine), one may write the following:

```
##  DATASET 7C:  Matrix Solver Controls for Transport Solution
##    [NSN]      [CSOLVU]  [ITRMXU] [ITOLU] [TOLU] [NSAVEU] [STARTU]
        1     'ORTHOMIN'      100       0    0.1       10       0.0
        2     'ORTHOMIN'      100       0    0.1       10    1000.0
        0
```

For a data set to simulate a single transport species, one may write the following:

```
##  DATASET 7C:  Matrix Solver Controls for Transport Solution
##               [CSOLVU] [ITRMXU] [ITOLU] [TOLU] [NSAVEU]
              'ORTHOMIN'      100      0     0.1      10
```

DATASET 8A: Output Controls and Options for “.lst” (Main Output) File (one line)

Variable	Type	Description
NPRINT	Integer	Printed output is produced on time steps numbered: n (NPRINT), as well as on first and last time step.
CNODAL	Character	A value of ‘N’ cancels printout of node coordinates, nodewise element thicknesses, and nodewise porosities. Set to ‘Y’ for full printout.
CELMNT	Character	A value of ‘N’ cancels printout of elementwise permeabilities and elementwise dispersivities. Set to ‘Y’ for full printout.
CINCID	Character	A value of ‘N’ cancels printout of node incidences and pinch node incidences in elements. Set to ‘Y’ for full printout.
CVEL	Character	Set to a value of ‘Y’ to calculate and print fluid velocities at element centroids each time printed output is produced. Note that for non-steady state flow, velocities are based on results and pressures of the previous time step or iteration and not on the newest values. Set to ‘N’ to cancel option.
CBUDG	Character	Set to a value of ‘Y’ to calculate and print a fluid mass budget and energy or solute mass budget each time printed output is produced. A value of ‘N’ cancels the option.

DATASET 8B: Output Controls and Options for “.nod” File (Nodewise Results Listed in Columns) (one line)

Variable	Type	Description
NCOLPR	Integer	Printed output of nodewise solution data are produced on time-step multiples of NCOLPR, as well as on first and last time step.
K5COL	Character	List of names of variables to be printed in columns in the “.nod” file. The ordering of columns corresponds to the ordering of variable names in the list. Valid names are as follows: ‘N’ = node number (if used, it must appear first in list) ‘X’ = X-coordinate of node ‘Y’ = Y-coordinate of node ‘Z’ = Z-coordinate of node (3-D only) ‘P’ = pressure (or head) ‘U’ = concentration or temperature (all species) ‘S’ = saturation ‘-‘ = end of list (any names following ‘-‘ are ignored) The symbol ‘-‘ (a single dash) must be used at the end of the list.

Example:

To output the 3-D node coordinates, pressure, and solute concentration in columns in the “.nod” file every 5 time steps, write the following:

```
5  ‘X’  ‘Y’  ‘Z’  ‘P’  ‘U’  ‘-‘
```

DATASET 8C: Output Controls and Options for “.ele” File (Velocities at Element Centroids Listed in Columns) (one line)

Variable	Type	Description
LCOLPR	Integer	Printed output of elementwise solution data are produced on time-step multiples of LCOLPR, as well as on first and last time step.
K6COL	Character	List of names of variables to be printed in columns in the “.ele” file. The ordering of columns corresponds to the ordering of variable names in the list. Valid names are as follows: ‘E’ = element number (if used, it must appear first in list) ‘X’ = X-coordinate of element centroid ‘Y’ = Y-coordinate of element centroid ‘Z’ = Z-coordinate of element centroid (3-D only) ‘VX’ = X-component of fluid velocity ‘VY’ = Y-component of fluid velocity ‘VZ’ = Z-component of fluid velocity ‘-‘ = end of list (any names following ‘-‘ are ignored) The symbol ‘-‘ (a single dash) must be used at the end of the list.

Example:

To output the 3-D element centroid coordinates and velocity components in columns in the “.ele” file every 10 time steps, write the following:

```
10 ‘X’  ‘Y’  ‘Z’  ‘VX’  ‘VY’  ‘VZ’  ‘-‘
```

DATASET 8D: Output Controls and Options for “.obs” File (Observation Node Results Listed in Columns – list directed input with no comments between entries)

O M I T when there are no observation nodes or if using the optional specified observation locations file (“.sob”).

Variable	Type	Description
NOBCYC	Integer	Printed output of observation node data are produced on time-step multiples of NOBCYC, as well as on first and last time step.
INOB	Integer	List of node numbers of observation nodes. Enter a value of zero as an extra observation node number following the last real observation node to indicate the end of the list.

Example:

To output pressures/heads, concentrations/temperatures, and saturations at nodes 1, 22, 333, and 4444 in columns in the “.obs” file every 100 time steps, write the following:

```
100  1  22  333  4444  0
```

DATASET 9A: Base Fluid Properties (one line)

Variable	Type	Description
COMPFL	Real	Fluid compressibility, $\beta=(1/\rho)(\partial \rho/\partial p)$. $[M/(L\cdot s^2)]^{-1}$. Note, specific pressure storativity is: $S_{op} = (1-\varepsilon)\alpha + \varepsilon\beta$
CW	Real	Fluid specific heat, c_w. [E/(M·ºC)] (May be left blank for solute transport simulation.)
BSIGMAW	Real	Base fluid diffusivity, σ_w. For energy transport represents fluid thermal conductivity, λ_w. [E/(L·ºC·s)]. For solute transport represents molecular diffusivity of solute in pure fluid, D_m $[L^2/s]$. May be decreased from pure value to account for tortuosity of fluid paths.
RHOW0	Real	Density of fluid at base concentration or temperature. $[M/L_3]$.
BURHOW0	Real	Base value of solute concentration (as mass fraction) or temperature of fluid at which base fluid density, RHOW0 is specified. $[M_s/M]$ or [ºC].
BDRWDU	Real	Base fluid coefficient of density change with concentration (fraction) or temperature: ρ = RHOW0 + DRWDU (U-URHOW0). $[M^2/(L^3\cdot M_s)]$ or $[M/(L^3\cdot {}^{o}C)]$
BVISC0	Real	For solute transport: base fluid viscosity, μ, [M/L·s]. For energy transport and multi-species energy and solute transport, this value is a scale factor. It multiplies the viscosity that is calculated internally in units of [kg/m·s]. BVISC0 may be used for energy transport to convert units of [kg/m·s] to desired units of viscosity.

Examples:

See the examples given below for DATASET 9B to see how DATASET 9A and 9B are related for multi-species transport simulations.

DATASET 9B: Fluid Properties (Up to NSPE lines plus one line). DATASET 9B data only needs to be entered for those species for which fluid properties are different from the base values specified in DATASET 9A.

O M I T when only one transport species is being simulated (NSPE = 1).

Maximum Lines NSPE:

Variable	Type	Description
NSN	Integer	Species Number.
SIGMAW	Real	Fluid diffusivity, $\sigma_{w,}$, for species NSN. For energy transport represents fluid thermal conductivity, λ_w. $[E/(L\cdot {}^{\circ}C\cdot T)]$. For solute transport represents molecular diffusivity of solute in pure fluid, D_m $[L^2/T]$. May be decreased from pure value to account for tortuosity of fluid paths.
URHOW0	Real	Value of solute concentration (as mass fraction) or temperature of fluid of species NSN at which base fluid density, RHOW0 is specified. $[M_{solute}/M_{fluid}]$ or $[{}^{\circ}C]$.
DRWDU	Real	Fluid coefficient of density change with concentration (fraction) or temperature for species NSN: $\rho = RHOW0 + \Sigma\, DRWDU_n\,(U_n - URHOW0_n)$ (See Eq. 2.4). $[M_{fluid}^2/(L^3\cdot M_{solute})]$ or $[M_{fluid}/(L^3\cdot {}^{\circ}C)]$
DVSDU	Real	Fluid coefficient of viscosity change with concentration (fraction) or temperature for species NSN: $\mu = BVISC0 + \Sigma\, DVSDU_n\,(U_n - URHOW0_n)$. For energy transport species DVSDU can be any real value because it is internally replaced with BVISC0, BVISC0 is subsequently set to 0, and the non-linear viscosity temperature relationship is used (see eq. 2.5). $[M_{fluid}^2/(L\cdot T\cdot M_{solute})]$ or $[M_{fluid}/(L\cdot T\cdot {}^{\circ}C)]$

Last line:

Variable	Type	Description
	Integer	Placed immediately following all DATASET 9B lines. Line must begin with the integer 0.

Examples:

For a data set to simulate four transport species of which only two have fluid properties that are different from the base values in DATASET 9A (*e.g.*, Energy and Brine), one may write the following:

```
##
##  DATASET 9A:  Base Fluid Properties
## [COMPFL]     [CW] [SIGMAW] [RHOW0] [URHOW0] [DRWDU] [BISC0]
     0.00D0 4182.D0  1.000D-9     1000.0     0.00    0.00     1.0
##
##  DATASET 9B:  Multiple Species Fluid Properties
## [NSN] [SIGMAW] [URHOW0] [DRWDU] [DVSDU]
      2    6.6D-6     0.00  7.00D2     0.0
      1       0.6     25.5 -0.15D0     0.0
      0
```

DATASET 10: Solid Matrix Properties (one line)

Variable	Type	Description
COMPMA	Real	Solid matrix compressibility, $\alpha=(1-\varepsilon)^{-1}\ \partial\varepsilon/\partial p$. $[M/(L\cdot s^2)]^{-1}$
CS	Real	Solid grain specific heat, c_s. $[E/(M\cdot{}^{\circ}C)]$ (May set to any real value for solute transport simulation.)
SIGMAS	Real	Solid grain diffusivity, σ_s. For energy transport represents thermal conductivity of a solid grain. $[E/(L\cdot{}^{\circ}C\cdot s)]$ (May be set to any real value for solute transport simulation.) Set SIGMAS to a negative value if spatially varying thermal conductivities are specified in DATASET 15D and/or to use geometric mean bulk thermal conductivities in a simulation. | SIGMAS | is used as the base thermal conductivity for all elements not modified in DATASET 15D. If the hydraulic parameters are defined using the optional zone definition file spatially varying thermal conductivities can be specified without specifying a negative SIGMAS value and inclusion of DATASET 15D in the SUTRA-MS *.inp file. However, a negative SIGMAS must be specified to use geometric mean thermal conductivities even if the optional zone definition file is used. In this case, a zero value can be entered on a line immediately after the THERMEQ value for DATASET 15D.
RHOS	Real	Density of a solid grain, ρ_s. $[M/L^3]$

DATASET 11: Adsorption Parameters (up to NSPE plus one line). Adsorption parameters must be defined for single species transport simulations even if ADSMOD is ‘NONE’ but do not have to be defined for multi-species transport simulations if ADSMOD is ‘NONE’ for all species. All adsorption parameters are initialized to zero prior to reading DATASET 11. DATASET 11 must be terminated with a zero NSN value for multi-species transport simulations.

Variable	Type	Description
Up to a Maximum of NSPE Lines:		
NSN	Integer	Species Number. OMIT for single species transport simulations (NSPE=1).
ADSMOD	Character	For no sorption or for energy transport simulation, set to ‘NONE’ and leave rest of line blank. For linear sorption model, set to ‘LINEAR’. For Freundlich sorption model, set to ‘FREUNDLICH’. For Langmuir sorption model, set to ‘LANGMUIR’.
CHI1	Real	Value of linear, Freundlich, or Langmuir distribution coefficient for species NSN, depending on sorption model chosen in ADSMOD, χ_1. $[L_f^3/M_G]$.
CHI2	Real	Value of Freundlich or Langmuir coefficient for species NSN, depending on sorption model chosen in ADSMOD. Set to any real value for linear sorption. χ_2. [1] for Freundlich. $[L_f^3/M_s]$ for Langmuir.
Last line:		
	Integer	Placed immediately following all DATASET 11 lines. Line must begin with the integer 0. OMIT if a single species transport simulation.

<u>Examples:</u>

For a data set to simulate two transport species of which only one is not equal to 'NONE', one may write the following:

```
##
##  DATASET 11:  Adsorption Parameters
## [NSN] [ADSMOD] [CHI1] [CHI2]
      1    ‘NONE’    0.0    0.0
      2  'LINEAR' 1.0D-1    0.0
      0
```

or

```
##
##  DATASET 11:  Adsorption Parameters
## [NSN] [ADSMOD] [CHI1] [CHI2]
       2 'LINEAR' 1.0D-1  0.0D0
       0
```

or for a data set to simulate a single transport species, one may write the following:

```
##
##  DATASET 11:  Adsorption Parameters
## [ADSMOD] [CHI1] [CHI2]
     ‘NONE’    0.0    0.0
```

DATASET 12: Production of Energy or Solute Mass (up to NSPE plus one lines). Production rates must be defined for single species transport simulations but do not have to be defined for all species in multi-species transport simulations. All production rates are initialized to zero prior to reading DATASET 12. DATASET 12 must be terminated with a zero NSN value for multi-species transport simulations.

Variable	Type	Description
Up to a Maximum of NSPE Lines:		
NSN	Integer	Species Number. O M I T for single species transport simulations (NSPE=1).
PRODF0	Real	Zero-order rate of production for species NSN in the fluid γ_o^w. [(E/M)/s] for energy production, [(M$_s$/M)/s] for solute mass production.
PRODS0	Real	Zero-order rate of production for species NSN in the immobile phase, γ_o^s. [(E/M$_G$)/s] for energy production, [(M$_s$/M$_G$)/s] for adsorbate mass production.
PRODF1	Real	First-order rate of solute mass production for species NSN in the fluid, γ_i^w. [s^{-1}]. Set to any real value for energy transport.
PRODS1	Real	First-order rate of adsorbate mass production for species NSN in the immobile phase, γ_i^s. [s^{-1}]. Set to any real value for energy transport.
Last line:		
	Integer	Placed immediately following all DATASET 12 lines. Line must begin with the integer 0. O M I T if a single species transport simulation.

Examples:

For a data set to simulate two transport species, one may write the following:

```
##
##  DATASET 12:  Production of Energy or Solute Mass
## [NSN]    [PRODF0] [PRODS0] [PRODF1] [PRODS1]
       1       0.0D0    0.0D0    0.0D0    0.0D0
       2 3.16881D-08    0.0D0    0.0D0    0.0D0
       0
```

or

```
##
##  DATASET 12:  Production of Energy or Solute Mass
## [NSN]    [PRODF0] [PRODS0] [PRODF1] [PRODS1]
       2 3.16881D-08    0.0D0    0.0D0    0.0D0
       0
```

or for a data set to simulate a single transport species, one may write the following:

```
##
##  DATASET 12:  Production of Energy or Solute Mass
##   [PRODF0] [PRODS0] [PRODF1] [PRODS1]
  3.16881D-08    0.0D0    0.0D0    0.0D0
```

DATASET 13: Orientation of Coordinates to Gravity (one line)

Variable	Type	Description
GRAVX	Real	Component of gravity vector in +x-direction. $[L/s^2]$ GRAVX = $-\lvert \underline{g} \rvert (\partial \text{ELEVATION}/\partial x)$, where $\lvert \underline{g} \rvert$ is the total acceleration due to gravity in $[L/s^2]$.
GRAVY	Real	Component of gravity vector in +y-direction. $[L/s^2]$ GRAVY= $-\lvert \underline{g} \rvert (\partial \text{ELEVATION}/\partial y)$, where $\lvert \underline{g} \rvert$ is the total acceleration due to gravity in $[L/s^2]$.
GRAVZ	Real	Component of gravity vector in +z-direction. $[L/s^2]$ GRAVY= $-\lvert \underline{g} \rvert (\partial \text{ELEVATION}/\partial y)$, where $\lvert \underline{g} \rvert$ is the total acceleration due to gravity in $[L/s^2]$. Set to any real value for 2-D problems.

DATASET 14A: Scale Factor for Nodewise Data (one line)

Variable	Type	Description
	Character	Line must begin with the word ‘NODE’.
SCALX	Real	The scaled x-coordinates of nodes in DATASET 14B are multiplied by SCALX in SUTRA. May be used to change from map to field scales, or from English to SI units. A value of 1.0 gives no scaling.
SCALY	Real	The scaled y-coordinates of nodes in DATASET 14B are multiplied by SCALY in SUTRA. May be used to change from map to field scales, or from English to SI units. A value of 1.0 gives no scaling.
SCALZ	Real	For 3-D problems, the scaled z-coordinates of nodes in DATASET 14B are multiplied by SCALZ in SUTRA. May be used to change from map to field scales, or from English to SI units. A value of 1.0 gives no scaling. For 2-D problems, the scaled element (mesh) thicknesses at nodes in DATASET 14B are multiplied by SCALZ in SUTRA. May be used to easily change entire mesh thickness or to convert English to SI units. A value of 1.0 gives no scaling.
PORFAC	Real	The scaled nodewise porosities of DATASET 14B are multiplied by PORFAC in SUTRA. May be used to easily assign a constant porosity value to all nodes by setting PORFAC=porosity, and all POR(II)=1.0 in DATASET 14B.

DATASET 14B: Nodewise Data (one line for each of NN nodes)

Variable	Type	Description
II	Integer	Number of node to which data on this line refers, i.
NREG(II)	Integer	Unsaturated flow property region number to which node II belongs. Set to any integer value when flow simulation is saturated only and hydraulic parameters are not defined by zones. The node region number is usually the same as the region number of the elements it appears in. When the node is to be at the boundary of two regions, it may be assigned to either region. If nodewise data are defined using zones then NREG defines the spatial distribution of the zones specified in the SUTRA-MS zone file.
X(II)	Real	Scaled x-coordinate of node II, x_i. [L]
Y(II)	Real	Scaled y-coordinate of node II, y_i. [L]
Z(II)	Real	For 3-D problems, scaled z-coordinate of node II, z_i. [L] For 2-D problems, scaled thickness of mesh at node II. [L] In order to simulate radial cross-sections, set THICK(II) = $(2\pi)(\text{radius}_i)$, where radius_i is the radial distance from the vertical center axis to node i.
POR(II)	Real	Scaled porosity value at node II, ε_i. [1]

Note:

The order in which the nodes are numbered affects the bandwidth of the global banded matrix which in turn affects computational and storage efficiency when the DIRECT solver is used. In this case, **the user should take care to number the nodes to minimize the problem bandwidth**. SUTRA sets the problem bandwidth equal to one plus twice the maximum difference in node numbers in the element containing the largest node number difference in the mesh. See figure 6.3.1 for an example. When an iterative solver is used, it is still advantageous to minimize the problem bandwidth, although not as critical as in the case of the DIRECT solver.

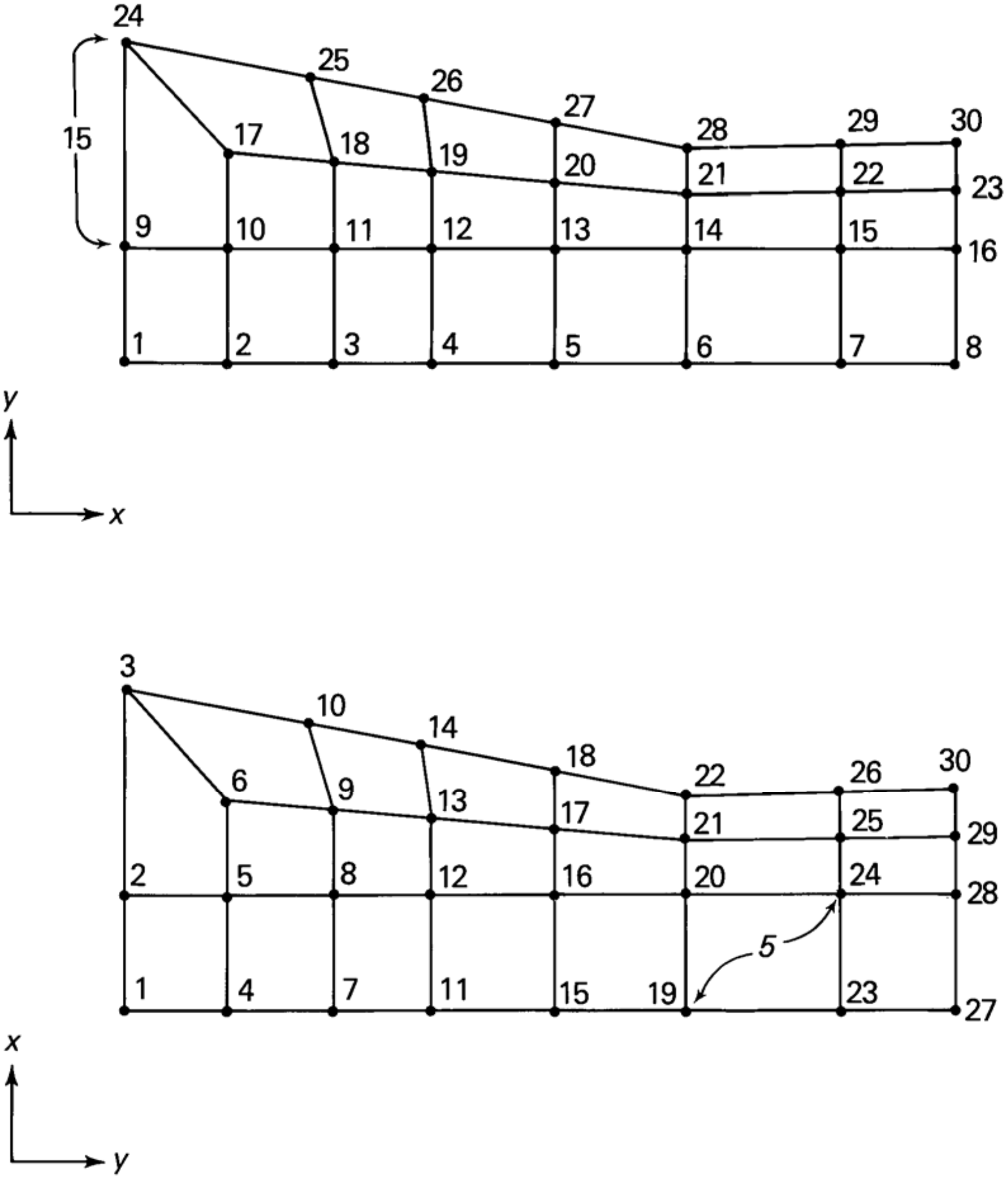


Figure 6.3.1 Minimization of bandwidth by careful numbering of nodes (Fig. 7.1 , Voss and Provost, 2002). In this 2D example, the same mesh has been numbered two different ways. In the first numbering scheme, the largest difference between node numbers in a single element is 15, giving a bandwidth of 1+2(15)=31. In the second numbering scheme, the largest difference between node numbers in a single element is 5, giving a bandwidth of 1+2(5)=11. The same principle applies to 3D meshes: the bandwidth equals one plus the maximum difference between node numbers in the element that contains the largest node number difference in the mesh.

DATASET 15A: Scale Factors for Elementwise Data (one line)

Variable	Type	Description
	Character	Line must begin with the word 'ELEMENT'.
PMAXFA	Real	The scaled maximum permeability values, PMAX, in DATASET 15B are multiplied by PMAXFA in SUTRA. May be used to convert units or to aid in assignment of maximum permeability values in elements.
PMIDFA	Real	The scaled "middle" permeability values, PMID, in DATASET 15B are multiplied by PMIDFA in SUTRA. May be used to convert units or to aid in assignment of maximum permeability values in elements. **Omit for 2-D problems.**
PMINFA	Real	The scaled minimum permeability values, PMIN, in DATASET 15B are multiplied by PMINFA in SUTRA. May be used to convert units or to aid assignment of minimum permeability values in elements.
ANG1FA	Real	The scaled angles ANGLE1 in DATASET 15B are multiplied by ANG1FA in SUTRA. For 2-D problems, may be used to easily assign a uniform direction of anisotropy by setting ANG1FA=angle, and all ANGLE1(L)=1.0 in DATASET 15B.
ANG2FA	Real	The scaled angles ANGLE2 in DATASET 15B are multiplied by ANG2FA in SUTRA. **Omit for 2-D problems.**
ANG3FA	Real	The scaled angles ANGLE3 in DATASET 15B are multiplied by ANG3FA in SUTRA. **Omit for 2-D problems.**
ALMAXF	Real	The scaled longitudinal dispersivities in the maximum permeability direction, ALMAX, in DATASET 15B are multiplied by ALMAXF in SUTRA. May be used to convert units or to aid in assignment of dispersivities.

ALMIDF	Real	The scaled longitudinal dispersivities in the “middle” permeability direction, ALMAX, in DATASET 15B are multiplied by ALMAXF in SUTRA. May be used to convert units or to aid in assignment of dispersivities. **Omit for 2-D problems.**
ALMINF	Real	The scaled longitudinal dispersivities in the minimum permeability direction, ALMIN, in DATASET 15B are multiplied by ALMINF in SUTRA. May be used to convert units or to aid in assignment of dispersivities.
ATMXF	Real	The scaled first transverse dispersivities in the maximum permeability direction, AT1MAX, in DATASET 15B are multiplied by AT1MXF in SUTRA. May be used to convert units or to aid in assignment of dispersivity.
ATMDF	Real	The scaled first transverse dispersivities in the “middle” permeability direction, AT1MID, in DATASET 15B are multiplied by AT1MDF in SUTRA. May be used to convert units or to aid in assignment of dispersivity. **Omit for 2-D problems.**
ATMNF	Real	The scaled first transverse dispersivities in the minimum permeability direction, AT1MIN, in DATASET 15B are multiplied by AT1MNF in SUTRA. May be used to convert units or to aid in assignment of dispersivity.

DATASET 15B: Elementwise Data (one line for each of NE elements)

Variable	Type	Description
L	Integer	Number of element to which data on this line refers.
LREG(L)	Integer	Unsaturated flow property region number to which this element belongs. Set to any integer value when flow simulation is saturated only and hydraulic parameters are not defined by zones. If elementwise data are defined using zones then LREG defines the spatial distribution of the zones specified in the SUTRA-MS zone file.
PMAX(L)	Real	Scaled maximum permeability value of element L, $k_{max}(L)$. $[L^2]$
PMID(L)	Real	Scaled "middle" permeability value of element L, $k_{mid}(L)$. $[L^2]$ Isotropic permeability requires: PMID(L)=PMAX(L). **Omit for 2-D problems.**
PMIN(L)	Real	Scaled minimum permeability value of element L, $k_{min}(L)$. $[L^2]$ Isotropic permeability requires: PMIN(L)=PMAX(L).
ANGLE1(L)	Real	Scaled angle within the x,y-plane, measured counterclockwise from +x-direction to maximum permeability direction in element L, $\theta_1(L)$. [°] In 3-D, this is the "yaw" of the principal permeability axes relative to the x, y, and z coordinate axes; arbitrary if, after application of scale factors, PMID(L)=PMAX(L), ALMID(L)=ALMAX(L), AT1MID(L)=AT1MAX(L), and AT2MID(L)=AT2MAX(L). In 2-D, arbitrary if, after application of scale factors, PMIN(L)=PMAX(L) and ALMIN(L)=ALMAX(L), and AT1MIN(L)=AT1MAX(L).
ANGLE2(L)	Real	Scaled angle measured (after ANGLE1 has been applied) upward from x,y-plane to maximum permeability direction in element L, $\theta_2(L)$. [°]

		In 3-D, this is the “pitch” of the principal permeability axes relative to the x, y, and z coordinate axes; arbitrary if, after application of scale factors, PMIN(L)=PMAX(L), ALMIN(L)=ALMAX(L), AT1MIN(L)=AT1MAX(L), and AT2MIN(L)=AT2MAX(L). **Omit for 2-D problems.**
ANGLE3(L)	Real	Scaled angle about the axis of maximum permeability, measured (after ANGLE1 and ANGLE2 have been applied) looking down the positive half of the axis toward the origin, and counterclockwise from the horizontal to the “middle” permeability direction in element L, θ_3(L). [°] In 3-D, this is the “roll” of the principal permeability axes relative to the x, y, and z coordinate axes; arbitrary if, after application of scale factors. PMIN(L)=PMID(L), ALMIN(L)=ALMID(L), AT1MIN(L)=AT1MID(L), and AT2MIN(L)=AT2MID(L). **Omit for 2-D problems.** If the permeability ellipsoid is initially oriented with the maximum, “middle”, and minimum permeability axes aligned in the x-, y-, and z-directions, respectively, then the "yaw, pitch, and roll" convention is equivalent to rotating the ellipsoid by ANGLE3 about the x-axis, by ANGLE2 about the y-axis, and by ANGLE1 about the z-axis, in that order.
ALMAX(L)	Real	Scaled longitudinal flow dispersivity value of element L for flow in the direction of maximum permeability PMAX(L), α_{Lmax}(L). [L]
ALMID(L)	Real	Scaled longitudinal flow dispersivity value of element L for flow in the direction of “middle” permeability PMID(L), α_{Lmid}(L). [L] **Omit for 2-D problems.**

ALMIN(L)	Real	Scaled longitudinal dispersivity value of element L for flow in the direction of minimum permeability PMIN(L), $\alpha_{Lmin}(L)$. [L]
ATMAX(L)	Real	Scaled first transverse dispersivity value of element L for flow in the direction of maximum permeability PMAX(L), $\alpha_{T1max}(L)$. [L]
ATMID(L)	Real	Scaled first transverse dispersivity value of element L for flow in the direction of “middle” permeability PMID(L), $\alpha_{T1mid}(L)$. [L] **Omit for 2-D problems.**
ATMIN(L)	Real	Scaled first transverse dispersivity value of element L for flow in the direction of minimum permeability PMIN(L), $\alpha_{T1min}(L)$. [L]

Notes:

Note that the convention for determining the 2D transverse dispersivity, α_T, differs from the one used in versions of SUTRA prior to version 2D3D.1 (Voss, 1984). See section 2.5 of Voss and Provost (2002).

The following notes are provided to assist the user in understanding the meaning of the ellipsoids and angles used in 3D SUTRA simulations.

The dispersivities ALMAX(L), ALMID(L), and ALMIN(L) represent the radii of the longitudinal dispersivity ellipsoid in the maximum, middle, and minimum permeability directions, respectively, for element L. The effective longitudinal dispersivity is the radius of this ellipsoid measured in the direction of flow. Thus, ALMAX(L), ALMID(L), and ALMIN(L) are the effective longitudinal dispersivities for flow in the maximum, middle, and minimum permeability directions, respectively. Note that “MAX”, “MID”, and “MIN” do not refer to the size of the dispersivities, but rather to direction.

The dispersivities ATMAX(L), ATMID(L), and ATMIN(L) represent the radii of the transverse dispersivity ellipsoid in the maximum, middle, and minimum permeability directions, respectively, for element L. The two effective transverse dispersivities are the principal radii of the ellipse formed by the intersection of this ellipsoid with the plane that passes through the center of the ellipsoid and is perpendicular to the flow direction. Thus, each of the values ATMAX(L), ATMID(L), and ATMIN(L) is an effective transverse dispersivity for flow within a plane containing two of the three principal permeability directions:

- For all flow directions within the (MAX,MID)-plane, ATMIN(L) is the effective dispersivity that controls transverse dispersion in the MIN direction.
- For all flow directions within the (MAX,MIN)-plane, ATMID(L) is the effective dispersivity that controls transverse dispersion in the MID direction.
- For all flow directions within the (MID,MIN)-plane, ATMAX(L) is the effective dispersivity that controls transverse dispersion in the MAX direction.

  It follows that when the flow direction coincides with one of the principal permeability directions, the effective transverse dispersivities are those corresponding to the remaining two principal permeability directions:

- For flow in the MAX permeability direction, the effective transverse dispersivities are ATMID(L) and ATMIN(L).
- For flow in the MID permeability direction, the effective transverse dispersivities are ATMAX(L) and ATMIN(L).
- For flow in the MIN permeability direction, the effective transverse dispersivities are ATMAX(L) and ATMID(L).

The angles ANGLE1(L), ANGLE2(L), and ANGLE3(L) may be thought of, in aeronautical terms, as the “yaw,” “pitch,” and “roll” of the permeability ellipsoid for element L with respect to the x-, y-, and z-coordinate axes. That is, if the maximum, middle, and minimum permeability axes of this ellipsoid are initially aligned with the x-, y-, and z-axes, the ellipsoid is oriented by rotating it by ANGLE1(L) within the x,y-plane, by ANGLE2(L) upward out of the x,y-plane, and by ANGLE3(L) about the maximum permeability axis, in that order. Note that this is equivalent to rotating the ellipsoid by ANGLE3(L) about the x-axis, by ANGLE2(L) about the y-axis, and by ANGLE1(L) about the z-axis, in that order.

In 3D simulations, ANGLE3(L) is arbitrary if the permeability and dispersion tensors are isotropic within the (MID,MIN)-plane, that is, if, after the application of scale factors, PMIN(L)=PMID(L), ALMIN(L)=ALMID(L), and ATMIN(L)=ATMID(L). All three angles, ANGLE1(L), ANGLE2(L), and ANGLE3(L), are arbitrary if the permeability and dispersion tensors are completely isotropic, that is, if, after the application of scale factors, PMIN(L)=PMID(L)=PMAX(L), ALMIN(L)=ALMID(L)=ALMAX(L), and ATMIN(L)=ATMID(L)=ATMAX(L). In 2D simulations, ANGLE1(L) is arbitrary if the permeability and dispersion tensors are isotropic, that is, if, after application of scale factors, PMIN(L)=PMAX(L), ALMIN(L)=ALMAX(L), and ATMIN(L)=ATMAX(L).

DATASET 15C: Dispersivity Multipliers (up to NSPE plus one line). Multipliers are applied to element dispersivity values specified in DATASET 15B for each species in a multi-species transport simulation. Only species that differ from the values in DATASET 15B need to be specified.

O M I T when only one transport species is being simulated (NSPE = 1).

Variable	Type	Description
Up to a Maximum of NSPE Lines:		
NSN	Integer	Species Number.
ATSPMULT	Real	Dispersivity multiplier for species NSN. $\alpha_{LMAX(NSN)} = \text{ALMAX} * \text{ATSPMULT}_{(NSN)}$ $\alpha_{LMID(NSN)} = \text{ALMID} * \text{ATSPMULT}_{(NSN)}$ $\alpha_{LMIN(NSN)} = \text{ALMIN} * \text{ATSPMULT}_{(NSN)}$ $\alpha_{TMAX(NSN)} = \text{ATMAX} * \text{ATSPMULT}_{(NSN)}$ $\alpha_{TMID(NSN)} = \text{ATMID} * \text{ATSPMULT}_{(NSN)}$ $\alpha_{TMIN(NSN)} = \text{ATMIN} * \text{ATSPMULT}_{(NSN)}$
Last line:		
	Integer	Placed immediately following all DATASET 15C lines. Line must begin with the integer 0.

Examples:

For a data set to simulate two transport species, one may write the following:

```
##
##  DATASET 15C:  DISPERSIVITY MULTIPLIERS
## [NSN] [ATSPMULT]
      1         0.1
      2         1.0
      0
```

or

```
##
##  DATASET 15C:  DISPERSIVITY MULTIPLIERS
## [NSN] [ATSPMULT]
       1        0.1
       0
```

DATASET 15D: Solid grain diffusivities (thermal conductivity) (up to NE plus one line). Multipliers applied to element dispersivity values specified in DATASET 15B for each species in a multi-species transport simulation. Only elements that differ from the | SIGMAS | value in DATASET 10 need to be specified.

OMIT when energy transport is not being simulated and/or the SIGMAS value specified in DATASET 10 is greater than or equal to zero.

Variable	Type	Description
THERMEQ	Character	Bulk thermal conductivity equation to use for energy transport or multi-species transport and energy transport simulations (eq. 2.16 or 2.17). OPTIONAL parameter. If specified must be ‘AVERAGE’ or ‘GEOMETRIC’. If THERMEQ is not specified then the volumetric mean thermal conductivity equation is used (eq. 2.16).

Up to a Maximum of NE Lines:

Variable	Type	Description
L	Integer	Element Number
LAMBDAS	Real	Solid grain diffusivity, σ_s. For energy transport represents thermal conductivity of a solid grain. [E/(L·°C·s)]

Last line:

Variable	Type	Description
	Integer	Placed immediately following all DATASET 15D lines. Line must begin with the integer 0.

Examples:

For a data set to simulate heat transport using a geometric mean bulk thermal conductivity, one may write the following:

```
##
##  DATASET 15D:  SOLID GRAIN DIFFUSIVITIES
##  THERMEQ
‘GEOMETRIC’
##  [L] [LAMBDAS]
      1      4.77
      2      4.77
         .
         .
         .
     10      5.22
     11      5.22
      0
```

DATASET 16: **NO LONGER USED**

DATASET 17: Data for Fluid Source and Sinks
(one line for each of NSOP fluid source nodes as specified
in DATASET 3, plus one line)

OMIT when there are no fluid source nodes

Variable	Type	Description
IQCP	Integer	Number of node to which source/sink data on this line refers. Specifying the node number with a negative sign indicates to SUTRA that the source flow rate or concentration or temperature of the source fluid vary in a specified manner with time. Information regarding a time-dependent source node must be programmed by the user in Subroutine BCTIME.
QINC	Real	Fluid source (or sink) which is a specified constant value at node IQCP, Q_{IN}. $\lvert M/s \rvert$ A positive value is a source of fluid to the aquifer. May be omitted if this value is specified as time-dependent in Subroutine BCTIME (IQCP<0). Sources are allocated by cell as shown in figure 6.3.2 for equal-sized elements. For unequal-sized elements, sources are allocated in proportion to the cell length, area, or volume over which the source fluid enters the system.
UINC	Real	Temperature or solute concentration (mass fraction) of fluid entering the aquifer which is a specified constant value for a fluid source at node IQCP, U_{IN}. $\lvert °C \rvert$ or $\lvert M_s/M \rvert$ Enter a UINC value for each species for a multi-species transport simulation. UINC values must be entered in order from 1 to NSPE as specified in DATASET 3B. May be omitted if this value is specified as time-dependent in Subroutine BCTIME (IQCP<0) or if QINC≤0.

Last line:

Integer	Placed immediately following all NSOP fluid source node lines. Line must begin with the integer 0.

Example:

For a data set to simulate two transport species, one may write the following:

```
##
##  DATASET 17:  Data for Fluid Source and Sinks
## [IQCP]   [QINC]   [UINC1]  [UINC2]
        1 8.25D+02 3.875E+01 0.00D+00
       22 1.50D+03 3.875E+01 0.00D+00
               .
               .
               .
      673 8.25D+02 5.000E+01 0.000D+00
        0
```

or for a data set to simulate a single transport species, one may write the following:

```
##
##  DATASET 17:  Data for Fluid Source and Sinks
## [IQCP]   [QINC]   [UINC1]
        1 8.25D+02 3.875E+01
       22 1.50D+03 3.875E+01
               .
               .
               .
      673 8.25D+02 5.000E+01
        0
```

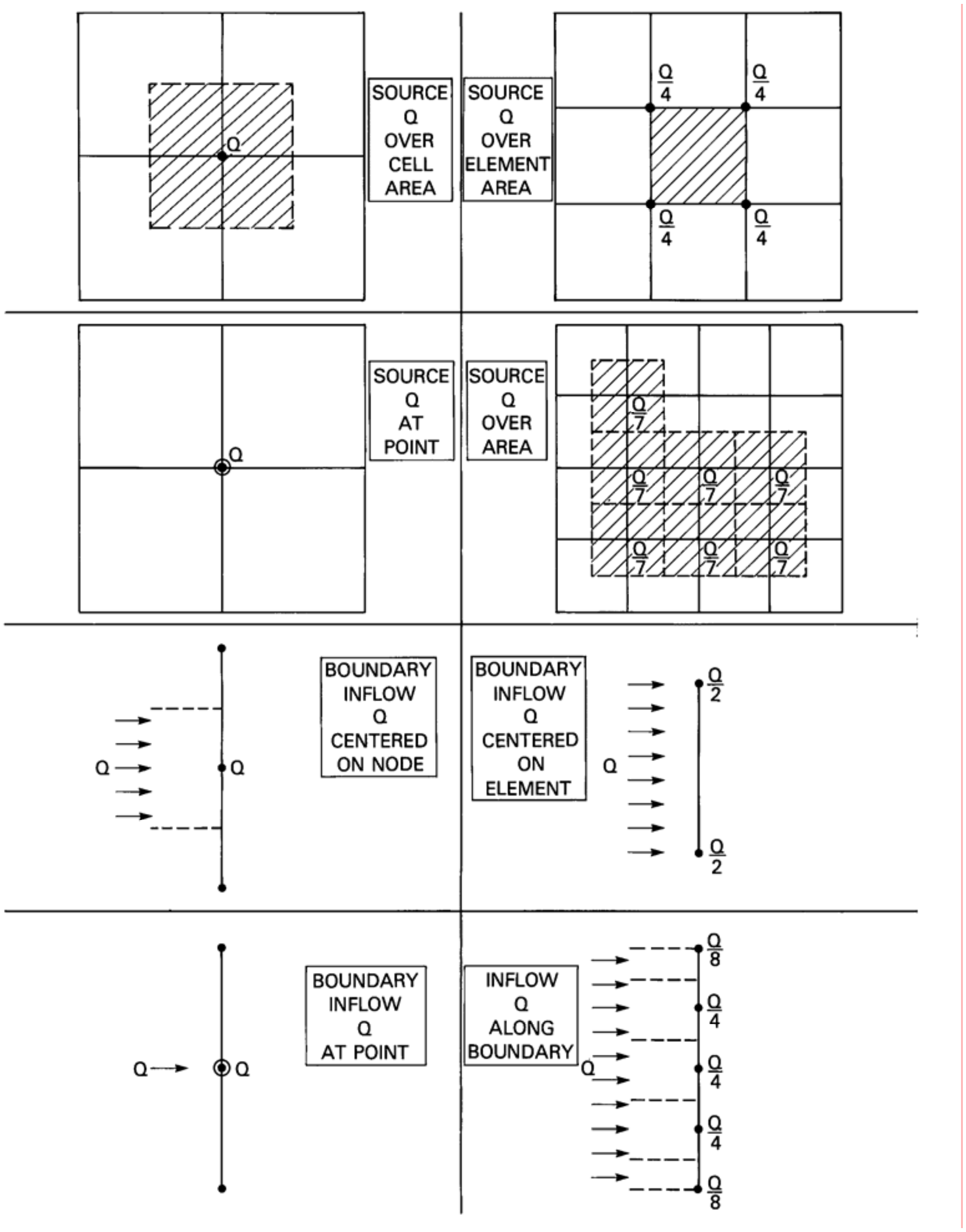


Figure 6.3.2 Allocation of sources and boundary fluxes in equal-sized elements (Fig. B.1 , Voss and Provost, 2002). The top four panels pertain to 2D areal and 3D meshes. The bottom four panels pertain to 2D cross-sectional meshes.

DATASET 18: Data for Energy or Solute Mass Sources and Sinks. For multi-species simulations, source/sink data for each species can be specified in any order as long as the total number of nodes for each species is equal to the NSOU values specified for each species in DATASET 3B.
(one line for each NSOU energy or solute source nodes as specified in DATASET 3B, plus one line)

O M I T when there are no energy or solute source nodes

Variable	Type	Description
Lines 1 to Σ(NSOU):		
NSN	Integer	Species Number. O M I T for single species transport simulations.
IQCU	Integer	Number of node to which source/sink data on this line refers. Specifying the node number with a negative sign indicates to SUTRA that the source rate varies in a specified manner with time. All information regarding a time-dependent source node must be programmed by the user in Subroutine BCTIME. Sources are allocated by cell as shown in figure 6.3.2 for equal-sized elements. For unequal-sized elements, sources are allocated in proportion to the cell length, area, or volume over which the source energy or solute mass enters the system.
QUINC	Real	Source (or sink) which is a specified constant value at node IQCU, ψ_{IN}. $|E/s|$ for energy transport, $|M_s/s|$ for solute transport. A positive value is a source to the aquifer. May be omitted if IQCU is negative, and this value is specified as time-dependent in Subroutine BCTIME.
Last line:		
	Integer	Placed immediately following all NSOU energy or solute mass source node lines. Line must begin with the integer 0.

Examples:

For a data set to simulate two transport species, one may write the following:

```
##
##  DATASET 18:  Data for Energy or Solute Mass Sources and Sinks
## [SP] [IQCU] [QUINC]
      2    650 1.0D-01
      1    671 2.0D-01
      0
```

or for a data set to simulate one transport species, one may write the following:

```
##
##  DATASET 18:  Data for Energy or Solute Mass Sources and Sinks
## [IQCU] [QUINC]
      650 1.0D-01
      671 1.0D-01
        0
```

DATASET 19: Data for Specified Pressure Nodes
(one line for each of NPBC specified pressure nodes
as indicated in DATASET 3B, plus one line)

O M I T when there are no specified pressure nodes

Variable	Type	Description
Lines 1 to NPBC:		
IPBC	Integer	Number of node to which specified pressure data on this line refers. Specifying the node number with a negative sign indicates to SUTRA that the specified pressure value or inflow concentration or temperature at this node vary in a specified manner with time. Information regarding a time-dependent specified pressure node must be programmed by the user in Subroutine BCTIME.
PBC	Real	Pressure value which is a specified constant at node IPBC. $\vert M/(L\cdot s^2)\vert$ May be omitted if this value is specified as time-dependent in Subroutine BCTIME.
UBC	Real	Temperature or solute concentration of any external fluid which enters the aquifer at node IPBC. UBC is a specified constant value. [°C] or $\vert M_s/M\vert$ Enter a UBC value for each species for a multi-species transport simulation. UBC values must be entered in order from 1 to NSPE as specified in DATASET 3B. May be omitted if this value is specified as time-dependent in Subroutine BCTIME.
Last line:		
	Integer	Placed immediately following all NPBC specified pressure lines. Line must begin with the integer 0.

Example:

For a data set to simulate two transport species, one may write the following:

```
##
##  DATASET 19:  Data for Specified Pressure Nodes
## [IPBC]    [PBC]   [UBC1]   [UBC2]
        1 0.00D+00 5.00E+01 3.57D-02
       22 0.00D+00 5.00E+01 3.57D-02
                .
                .
                .
      673 1.00D+06 5.00E+01 3.57D-02
        0
```

or for a data set to simulate a single transport species, one may write the following:

```
##
##  DATASET 19:  Data for Specified Pressure Nodes
## [IPBC]    [PBC]    [UBC]
        1 0.00D+00 5.00E+01
       22 0.00D+00 5.00E+01
                .
                .
                .
      673 1.00D+06 5.00E+01
        0
```

DATASET 20: Data for Specified Concentration or Temperature Nodes. For multi-species simulations, source/sink data for each species can be specified in any order as long as the total number of nodes for each species is equal to the NUBC values specified for each species in DATASET 3B. (one line for each NUBC specified concentration or temperature nodes indicated in DATASET 3B, plus one line)

O M I T when there are no specified concentration or temperature nodes

Variable	Type	Description
Lines 1 to Σ(NUBC):		
NSN	Integer	Species Number. O M I T for single species transport simulations.
IUBC	Integer	Number of node to which specified concentration or temperature data on this line refers. Specifying the node number with a negative sign indicates to SUTRA that the specified value at this node varies in a specified manner with time. This time-dependent concentration or temperature must be programmed by the user in Subroutine BCTIME.
UBC	Real	Temperature or solute concentration value which is a specified constant at node IUBC. │°C│ or [M_s/M] May be omitted if IUBC is negative and this value is specified as time-dependent in Subroutine BCTIME.
Last line:		
	Integer	Placed immediately following all NUBC specified temperature or concentration lines. Line must begin with the integer 0.

Examples:

For a data set to simulate two transport species, one may write the following:

```
##
##  DATASET 20:  Data for Specified Concentration or Temperature
Nodes
## [NSN] [IUBC]     [UBC]
       1    631 5.000E+01
       2    631 0.357E-02
         .
         .
         .
       1    692 7.250E+00
       2    692 0.000E+00
       0
```

or for a data set to simulate one transport species, one may write the following:

```
##
##  DATASET 20:  Data for Specified Concentration or Temperature
Nodes
## [IUBC]     [UBC]
      631 5.000E+01
         .
         .
         .
      692 7.250E+00
        0
```

DATASET 21: **NO LONGER USED**

DATASET 22: Element Incidence Data (one line, plus one line for each of NE elements)

Variable	Type	Description
Line 1:		
	Character	Line must begin with the word 'INCIDENCE'.
Lines 2 to NE+1:		
LL	Integer	Number of element to which data on this line refers.
IIN	Integer	Node incidence list; list of corner node numbers in element LL, beginning at any node. For 2-D problems, the four nodes are listed in an order counterclockwise about the element. For 3-D problems, the eight nodes are listed as follows: Approach the element from any of its six sides. On the face farthest from you (the "back" face), list the four nodes in an order counterclockwise about the face. Then, on the face closest to you (the "front" face), again list the four nodes counterclockwise, starting with the node directly in front of the node that was listed first. (This convention assumes a right-handed coordinate system.)

End of Input Data List for ".inp" File

6.4 List of Input Data for ".ics" (Initial Conditions) File

Model Series:	SUTRA-MS
Model Version:	2D3DMS.1

Data in the ".ics" input file need be created by the user only for "cold starts" of SUTRA simulation (i.e., for the very first time step of a given simulation or series of restarted simulations).

SUTRA will optionally store final results of a simulation in a ".rst" file, which is directly useable as a ".ics" file for later restarts. The restart options are controlled by CREAD and ISTORE in DATASET 4 of the ".inp" file.

DATASET 1: Simulation Starting Time (one line)

Variable	Type	Description
TSTART	Real	Elapsed time at which the initial conditions for simulation specified in the ".ics" file are given. [s] This sets the simulation clock starting time. Usually set to a value of zero for a "cold start".

DATASET 2: Initial Pressure Values at Nodes (two lines; second line can be broken up over multiple lines)

Variable	Type	Description
Line 1:		
CPUNI	Character	One word. Set to 'UNIFORM' to specify a uniform pressure for all nodes. Set to 'NONUNIFORM' to specify a separate pressure for each node.
Line 2:		
PVEC	Real	For UNIFORM pressure specification, a single value of initial (starting) pressure to be applied at all NN nodes at time TSTART. $[M/(L\cdot s^2)]$ For NONUNIFORM pressure specification, a list of values of initial (starting) pressures at time TSTART, one value for each of NN nodes, in exact order of node numbers. $[M/(L\cdot s^2)]$ If the STEADY (steady-state) flow option in DATASET 4 of the ".inp" file has been chosen, PVEC serves as an initial guess for the pressure solution when an ITERATIVE solver is used, and is ignored when the DIRECT solver is used. Initial hydrostatic or natural pressures in a cross-section may be obtained by running a single steady-flow time step with the store option. Then the natural pressures are calculated and stored in the ".rst" file, and may be copied to the "cold start" ".ics" file without change in format, to be used as initial conditions for a transient run.

DATASET 3: Initial Temperature or Concentration Values at Nodes. For multi-species simulations, DATASET 3 must be repeated NSPE times.
(two lines; second line can be broken up over multiple lines)

Variable	Type	Description
Line 1:		
CUUNI	Character	One word. Set to 'UNIFORM' to specify a uniform temperature to solute concentration for all nodes. Set to 'NONUNIFORM' to specify a separate value for each node.
CTSPE	Character	One word. Set to 'SPECIES NSN' to specify the temperature or solute concentration on Line 2 is for species NSN, where NSN varies from 1 to NSPE. OMIT if a single species transport simulation data set (NSPE=1).
Line 2:		
UVEC	Real	For UNIFORM temperature or solute concentration specification, a single initial (starting) value to be applied at all NN nodes at time TSTART. [°C] or [M_S/M] For NONUNIFORM temperature or solute concentration specification, a list of initial (starting) values at time TSTART, one value for each of NN nodes, in exact order of node numbers. [°C] or [M_S/M]

Examples:

For a data set to simulate two transport species, one may write the following:

```
##
##  DATASET 3:  Initial Temp or Conc at Nodes [SPECIES 1]
  'UNIFORM'  'SPECIES 1'
   3.875D+01
##
##  DATASET 3:  Initial Temp or Conc at Nodes [SPECIES 2]
  'UNIFORM'  'SPECIES 2'
  0.0357D+00
```

or for a data set to simulate one transport species one may write the following:

```
##
##  DATASET 3:  Initial Temp or Conc at Nodes
  'UNIFORM'
   3.875D+01
```

End of Input Data List for ".ics" File

6.5 List of Input Data for ".tbc" (Simple Transient Boundary Conditions) File

Model Series:	SUTRA-MS
Model Version:	2D3DMS.1

Data in the ".tbc" input file need be created by the user only if the simple transient boundary conditions option is required and "TBC" is specified in the File Assignment Input File (SUTRA.FIL). This package is compatible with user-programmed boundary conditions in BCTIME. If a simple boundary condition is specified for a node with a user-programmed boundary condition (specified with a negative node number in the main SUTRA-MS input file) or does not correspond to a node specified in DATASETS 17, 18, 19, or 20 contained in the main SUTRA-MS data set (".inp"), the item is ignored.

Repeat Data Set TBC1, TBC 2, TBC 3A, TBC 3B, TBC 3C, and TBC 3D as many times as required.

DATASET TBC1: Starting Time of Next Simple Transient Boundary Conditions (one line)

Variable	Type	Description
DNTIME	Real	Starting time of New Transient Boundary Conditions. Starting time is the elapsed time from the start of the simulation. [s]

DATASET TBC2: Transient Boundary Condition Control Parameters (one line)

Variable	Type	Description
NTPBC	Integer	Number of transient specified pressure nodes.
NTUBC	Integer	Number of transient specified concentration or temperature nodes.
NTSOP	Integer	Number of transient specified fluid source or sink nodes.
NTSOU	Integer	Number of transient specified fluid solute or heat source or sink nodes.

Note:

An error condition will occur if the NTPBC, NTUBC, NTSOP, or NTSOU exceed NPBC, MNUBC, NSOP, or MNSOU specified in DATASET 3A, respectively.

DATASET TBC3A: Data for Transient Specified Pressure Nodes
(one line for each of NTPBC specified pressure nodes as indicated in DATASET 2, plus one line)

O M I T when there are no transient specified pressure nodes

Variable	Type	Description
Lines 1 to NTPBC:		
ITPBC	Integer	Number of node to which specified pressure data on this line refers.
TPBC	Real	Pressure value which is a specified constant at node ITPBC. $\left\|M/(L\cdot s^2)\right\|$
TUBC	Real	Temperature or solute concentration of any external fluid which enters the aquifer at node ITPBC. TUBC is a specified constant value. [°C] or $\left\|M_s/M\right\|$ Enter a TUBC value for each species for a multi-species transport simulation. TUBC values must be entered in order from 1 to NSPE as specified in DATASET 3B.
Last line:		
	Integer	Placed immediately following all NTPBC specified pressure lines. Line must begin with the integer 0.

DATASET TBC3B: Data for Transient Specified Concentration or Temperature Nodes. For multi-species simulations, source/sink data for each species can be specified in any order as long as the total number of nodes for each species is equal to the NTUBC values specified for each species in DATASET 2.
(one line for each of NTUBC specified concentration or temperature nodes indicated in DATASET 2, plus one line)

O M I T when there are no transient specified concentration or temperature nodes

Variable	Type	Description
Lines 1 to Σ(NTUBC):		
NSN	Integer	Species Number. O M I T for single species transport simulations.
ITUBC	Integer	Number of node to which specified concentration or temperature data on this line refers.
TUBC	Real	Temperature or solute concentration value which is a specified constant at node ITUBC. │°C│ or $[M_s/M]$
Last line:		
	Integer	Placed immediately following all NTUBC specified temperature or concentration lines. Line must begin with the integer 0.

DATASET TBC3C: Data for Transient Fluid Source and Sinks

(one line for each of NTSOP fluid source nodes as specified in DATASET 2, plus one line)

O M I T when there are no transient fluid source nodes

Variable	Type	Description
ITQCP	Integer	Number of node to which source/sink data on this line refers.
TQINC	Real	Fluid source (or sink) which is a specified constant value at node ITQCP, Q_{IN}. $\lvert M/s \rvert$ Sources are allocated by cell as shown in figure 6.3.2 for equal-sized elements. For unequal-sized elements, sources are allocated in proportion to the cell length, area, or volume over which the source fluid enters the system.
TUINC	Real	Temperature or solute concentration (mass fraction) of fluid entering the aquifer which is a specified constant value for a fluid source at node ITQCP, U_{IN}. $\lvert °C \rvert$ or $\lvert M_s/M \rvert$ Enter a UINC value for each species for a multi-species transport simulation. UINC values must be entered in order from 1 to NSPE as specified in DATASET 3B.
Last line:		
	Integer	Placed immediately following all NTSOP fluid source node lines. Line must begin with the integer 0.

DATASET TBC3D: Data for Transient Energy or Solute Mass Sources and Sinks. For multi-species simulations, source/sink data for each species can be specified in any order as long as the total number of nodes for each species is equal to the NTSOU values specified for each species in DATASET 2.
(one line for each NTSOU energy or solute source nodes as specified in DATASET 2, plus one line)

OMIT when there are no transient energy or solute source nodes

Variable	Type	Description
Lines 1 to Σ(NTSOU):		
NSN	Integer	Species Number. OMIT for single species transport simulations.
ITQCU	Integer	Number of node to which source/sink data on this line refers. Sources are allocated by cell as shown in figure 6.3.2 for equal-sized elements. For unequal-sized elements, sources are allocated in proportion to the cell length, area, or volume over which the source energy or solute mass enters the system.
TQUINC	Real	Source (or sink) which is a specified constant value at node ITQCU, ψ_{IN}. $\lvert E/s \rvert$ for energy transport, $\lvert M_s/s \rvert$ for solute transport. A positive value is a source to the aquifer. May be omitted if ITQCU is negative, and this value is specified as time-dependent in Subroutine BCTIME.
Last line:		
	Integer	Placed immediately following all NTSOU energy or solute mass source node lines. Line must begin with the integer 0.

DATASET 4: Transient Boundary Condition Termination Control Parameter (one line)

Variable	Type	Description
CTERM	Character	One word. Must be set to ‘END’ to terminate reading transient boundary condition data.

Examples:

For a data set to simulate transient boundary conditions that change three times (including initial plus two during the simulation) for a simulation with two transport species, one may write the following:

```
##
## DATASET TBC1
## DNTIME - 1
   18000.0
##
## DATASET TBC2
## NTPBC,NTUBC,NTSOP,NTSOU
       0     1     0     0
##
## DATASET TBC3A
## SPECIFIED PRESSURES
## ***NONE***
##
## DATASET TBC3B
## SPECIFIED CONCENTRATION
##[NSN]  [ITUBC] [UTBC]
      2        6   10.0
      0
##
## DATASET TBC3C
## SPECIFIED FLUID FLUXES
## ***NONE***
##
## DATASET TBC3D
## SPECIFIED SOLUTE OR TEMPERATURE FLUXES
## ***NONE***
##
## DATASET TBC1
## DNTIME - 2
   19000.0
## DATASET TBC2
## NTPBC,NTUBC,NTSOP,NTSOU
       0     2     0     0
##
## DATASET TBC3A
## SPECIFIED PRESSURES
## ***NONE***
##
```

```
## DATASET TBC3B
## SPECIFIED CONCENTRATION
##[NSN]  [ITUBC] [UTBC]
      2        6    9.0
      1      100    1.0
      0
##
## DATASET TBC3C
## SPECIFIED FLUID FLUXES
## ***NONE***
##
## DATASET TBC3D
## SPECIFIED SOLUTE OR TEMPERATURE FLUXES
## ***NONE***
##
## DATASET TBC4
## DATA SET TERMINATION
'END'
##
```

End of Input Data List for “.tbc” File

6.6 List of Input Data for ".otm" (Specified User Output Times) File

Model Series: SUTRA-MS
Model Version: 2D3DMS.1

Data in the ".otm" input file need be created by the user only if the specified user output times option is required and "OTM" is specified in the File Assignment Input File (SUTRA.FIL). If this optional package is used then output control options (NPRINT, CNADAL, and CELEMNT) specified in DATASET 8A in the main SUTRA-MS input file (".inp") are not used for output control for the ".lst", ".nod", and ".ele" output files. All other output control items in the main SUTRA-MS input file (".inp") are still used.

Repeat DATASET OTM1 as many times as required.

DATASET OTM1: Print Time for output to the ".lst", ".nod", and ".ele" output files (one line)

Variable	Type	Description
DPTIME	Real	Starting time of New Transient Boundary Conditions. Starting time is the elapsed time from the start of the simulation. [s]

DATASET OTM2: Specified User Output Times Termination Control Parameter (one line)

Variable	Type	Description
CTERM	Character	One word. Must be set to 'END' to terminate reading transient boundary condition data.

Examples:

For a data set to output simulation data to the ".lst", ".nod", and ".ele" at three specified times, one may write the following:

```
##
## Dataset OTM1- DPTIME 1
150.0
```

```
##
## Dataset OTM1- DPTIME 2
18000.0
##
## Dataset OTM1- DPTIME 3
24000.0
##
## DATASET OTM2
## DATA SET TERMINATION
'END'
##
```

End of Input Data List for “.otm” File

6.7 List of Input Data for “.ats” (Simple Automatic Time-Stepping Algorithm) File

Model Series:	SUTRA-MS
Model Version:	2D3DMS.1

Data in the “.ats” input file need be created by the user only if the specified user output times option is required and “ATS” is specified in the File Assignment Input File (SUTRA.FIL). If this optional package is used then the simulation time-step is reduced if user-specified iteration criteria are not satisfied. If the user-specified iteration criteria are not satisfied, the user has the option to rerun the current solution with a reduced time-step length. If the user elects to rerun the current solution with a reduced time step, the user specifies the maximum number of times the current solution is to be rerun if the user-specified iteration criteria are not met before continuing on to the next time step. All other time-step control items specified in DATASET 6 of the main SUTRA-MS input file (“.inp”) are still used to increase the time step.

DATASET ATS1: Control Parameters for the ATS option (one line)

Variable	Type	Description
imaxtarget	Integer	Maximum iteration allowed for the current time step before implementing a time-step length reduction.
dtmin	Real	Minimum simulation time step [s].
LContinue	Logical	Logical parameter used to control rerun of current time step. If true, the current time step **is not** rerun if imaxtarget is exceeded. If false, the current time step **is** rerun if imaxtarget is exceeded for either the flow or any transport solution.

DATASET ATS2: Control parameters for rerunning time steps (one line)

O M I T when LContinue is true

Variable	Type	Description

NConvMax	Integer	Maximum number of times to rerun the current time step.

Examples:

For a data set to reduce the time step if a user-specified iteration criteria is exceeded and rerun the current time-step, one may write the following:

```
##
## INPUT DATA FOR AUTOMATIC TIME STEP (ATS) ALGORITHM
## UNIT K12
##
## Dataset ATS1
## [imaxtarget] [dtmin] [LContinue]
             40     0.1           F
## Dataset ATS2
##   [NConvMax]
             10
##
## END OF ATS DATA
```

End of Input Data List for “.ats” File

6.8 List of Input Data for ".sob" (Specified Observation Locations) File

Model Series:	SUTRA-MS
Model Version:	2D3DMS.1

Data in the ".sob" input file need be created by the user only if the specified observation locations are specified by spatial coordinates rather than node numbers and "SOB" is specified in the File Assignment Input File (SUTRA.FIL). If this optional package is used then the specified coordinates (x-, y-, and z-location) are specified for each observation location instead of node number (DATASET 8D).

DATASET SOB1: Control Parameters for the SOB input file (one line)

Variable	Type	Description
NOBS	Integer	Exact number of nodes at which observations will be made.
NOBCYC	Integer	Printed output of observation node data are produced on time-step multiples of NOBCYC, as well as on first and last time step.

DATASET SOB2: Coordinates of observation locations (one line for each NOBS specified concentration or temperature nodes indicated in DATASET SOB2)

Variable	Type	Description
X(N)	Real	Unscaled x-coordinate of observation location N, x_n. [L]
Y(II)	Real	Unscaled y-coordinate of observation location N, y_n. [L]
Z(II)	Real	For 3-D problems, unscaled z-coordinate of observation location N, z_n. [L] **O M I T** for 2-D problems.

Examples:

For a data set to observe simulated results at five (5) specified locations in a 2D problem one may write the following:

```
##
## Specified Observation Locations (“.SOB”) file
## Dataset SOB1
## NOBS NOBCYC
      5      1
##
## Dataset SOB2
##    x       y
   0.00    0.00
   0.50    0.75
   0.75    0.50
   1.00    0.25
   2.00    1.00
##
## END OF SOB DATA
##
```

End of Input Data List for “.sob” File

6.9 List of Input Data for ".zon" (Node and Element zone parameters) File

Model Series:	SUTRA-MS
Model Version:	2D3DMS.1

Data in the ".zon" input file need be created by the user only if the user wants to specify nodal and elemental hydraulic parameters by zones and "ZON" is specified in the File Assignment Input File (SUTRA.FIL). If this optional package is used then the nodal and hydraulic parameters usually specified in DATASETS 14B and 15B in the main SUTRA input file (".inp") are specified in the ".zon" file. Zones are defined using the NREG and LREG variables for nodes and elements in DATASETS 14B and 15B, respectively.

DATASET ZON1: Control Parameters for the ZON option (one line)

Variable	Type	Description
NodeZones	Integer	Number of zones used for hydraulic parameters with nodewise discretization (*i.e.*, porosity, matrix compressibility, and solid matrix density.
ElemZones	Integer	Number of zones used for hydraulic parameters with elementwise discretization (*i.e.*, permeability, permeability angles, dispersivity, and solid matrix thermal conductivity.

DATASET ZON2: Hydraulic parameter for the exact number of nodewise zones specified in DATASET ZON1 (NodeZones+1 line)

Variable	Type	Description
NodeZones Lines:		
POR(II)	Real	Scaled porosity value at node II, ε_i. [1]
COMPMA	Real	Solid matrix compressibility, $\alpha=(1-\varepsilon)^{-1}\ \partial\varepsilon/\partial p$. $[M/(L\cdot s^2)]^{-1}$
RHOS	Real	Density of a solid grain, ρ_s. $[M/L^3]$

Last line:

	Integer	Placed immediately following all DATASET Z2 lines. Line must begin with the integer 0.

DATASET ZON3: Hydraulic parameter for the exact number of elementwise zones specified in DATASET ZON1 (ElemZones+1 line)

Variable	Type	Description
ElemZones Lines:		
PMAX(L)	Real	Scaled maximum permeability value of element L, $k_{max}(L)$. [L^2]
PMID(L)	Real	Scaled “middle” permeability value of element L, $k_{mid}(L)$. [L^2] Isotropic permeability requires: PMID(L)=PMAX(L). **Omit for 2-D problems.**
PMIN(L)	Real	Scaled minimum permeability value of element L, $k_{min}(L)$. [L^2] Isotropic permeability requires: PMIN(L)=PMAX(L).
ANGLE1(L)	Real	Scaled angle within the x,y-plane, measured counterclockwise from +x-direction to maximum permeability direction in element L, $\theta_1(L)$. [°]
ANGLE2(L)	Real	Scaled angle measured (after ANGLE1 has been applied) upward from x,y-plane to maximum permeability direction in element L, $\theta_2(L)$. [°] **Omit for 2-D problems.**
ANGLE3(L)	Real	Scaled angle about the axis of maximum permeability, measured (after ANGLE1 and ANGLE2 have been applied) looking down the positive half of the axis toward the origin, and counterclockwise from the horizontal to the “middle” permeability direction in element L, $\theta_3(L)$. [°] **Omit for 2-D problems.**
ALMAX(L)	Real	Scaled longitudinal flow dispersivity value of element L for flow in the direction of maximum permeability PMAX(L), $\alpha_{Lmax}(L)$. [L]

ALMID(L)	Real	Scaled longitudinal flow dispersivity value of element L for flow in the direction of "middle" permeability PMID(L), $\alpha_{Lmid}(L)$. [L] **Omit for 2-D problems.**
ALMIN(L)	Real	Scaled longitudinal dispersivity value of element L for flow in the direction of minimum permeability PMIN(L), $\alpha_{Lmin}(L)$. [L]
ATMAX(L)	Real	Scaled first transverse dispersivity value of element L for flow in the direction of maximum permeability PMAX(L), $\alpha_{T1max}(L)$. [L]
ATMID(L)	Real	Scaled first transverse dispersivity value of element L for flow in the direction of "middle" permeability PMID(L), $\alpha_{T1mid}(L)$. [L] **Omit for 2-D problems.**
ATMIN(L)	Real	Scaled first transverse dispersivity value of element L for flow in the direction of minimum permeability PMIN(L), $\alpha_{T1min}(L)$. [L]
LAMBDAS	Real	Solid grain diffusivity, σ_s. For energy transport represents thermal conductivity of a solid grain. [E/(L·°C·s)]

Last line:

	Integer	Placed immediately following all DATASET Z3 lines. Line must begin with the integer 0.

Notes:

See input instructions for DATASET 15B for a complete description of the hydraulic parameters for DATASET Z3.

Examples:

For a 2D data set with one (1) nodewise and two (2) hydraulic parameter zones, one may write the following:

```
##
## Zone file
## Dataset ZON1
```

```
## NodeZones, ElemZones
           1          2
##
## Node data
## Dataset ZON2
##
## zone, por, compma, rhos
      1 0.20     0.0    1.
      0
##
## Element data
## Dataset ZON3
##
## zone,  pmax,  pmin, anglex, almax, almin, atmax, atmin, lambdas
      1  2.e-4  2.e-4     0.0  500.0  500.0  100.0  100.0      0.0
      2 2.e-10 2.e-10     0.0  500.0  500.0  100.0  100.0      0.0
      0
```

End of Input Data List for “.zon” File

7.0 References

Cordes, C., and Kinzelbach, W., 1992, Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements: Water Resources Research, v. 28, no. 11, p. 2,903-2,911.

Diersch, H.J., 1988, Finite element modeling of recirculating density driven saltwater intrusion processes in ground water: Advances in Water Resources, v. 11, p. 25-43.

Doherty, J., 1994, PEST: Corinda, Australia, Watermark Computing, 122 p.

Goode, D.J., 1996, Direct simulation of groundwater age: Water Resources Research, v. 32, no. 2, p. 289-296.

Henry, H.R., 1964, Effects of dispersion on salt encroachment in coastal aquifers, *in* Cooper, H.H., Sea Water in Coastal Aquifers:U.S. Geological Survey Water-Supply Paper 1613-C, p. C71-C84.

Henry, H.R., and Hilleke, J.B., 1972, Exploration of multiphase fluid flow in a saline aquifer system affected by geothermal heating: Bureau of Engineering Research, Report No. 150-118, University of Alabama, U.S. Geological Survey Contract No. 14-08-0001-12681, National Technical Information Service Publication No. PB234233, 105 p.

Huyakorn, P.S., Andersen, P.F., Mercer, J.W., and White, H.O., 1987, Saltwater intrusion in aquifers--Development and testing of a three dimensional finite element model: Water Resources Research, v. 23, p. 293-319.

Kipp, K.L.,Jr., 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, 517 p.

Kipp, K.L.,Jr., 1997, Guide to the revised heat and transport solute transport simulator: HST3D – Version 2: U.S. Geological Survey Water-Resources Investigations Report 97-4157, 149 p.

Konikow, L.F., and Reilly, T.E., 1999, Seawater intrusion in the United States, *in* Bear, A.H., and others (eds.), Seawater intrusion in coastal aquifers – Concepts, methods, and practices: Dordrecht, Kluwer Academic Publishers, p. 463-506.

Ingebritsen, S.E., and . Sanford, W.E., 1998, Ground water in geologic processes: Cambridge, Cambridge University Press, 341 p.

INTERA Environmental Consultants, 1979, Revision of the documentation for a model calculating effects of liquid waste disposal in deep saline aquifers: U.S. Geological Survey Water-Resources Investigations Report 79-61, 263 p.

Lewis, F.M., Voss, C.I., and Rubin, J., 1986, Numerical simulation of advective-dispersive multisolute transport with sorption, ion exchange and equilibrium chemistry: U.S. Geological Survey Water-Resources Investigations Report 86-4022, 165 p.

Nield, D.A., and Bejan, A., 1998, Convection in porous media: Springer-Verlag, 496 p..

Oldenburg, C.M., and Pruess, K., 2000, Simulation of propagating fronts in geothermal reservoirs with the implicit Leonard total variation diminishing scheme, Geothermics, v. 29, p. 1-25.

Poeter, E.P., and Hill, M.C., 1998, Documentation of UCODE, a computer code for universal inverse modeling: U. S. Geological Survey Water-Resources Investigations Report 98-4080, 116 p.

Pringle, S.E., Glass, R.J., and . Cooper, C.A., 2002, Double-diffusive finger convection in a Hele-Shaw cell--An experiment exploring the evolution of concentration fields, length scales and mass transfer: Transport in Porous Media, v. 47, no. 2, p. 195-214.

Sahoo, D., Smith, J.A., Imbrigiotta, T.E., and McLellan, H.M., 1998, Surfactant enhanced remediation of a trichloroethene-contaminated aquifer, 2. Transport of TCE: Environmental Science Technology, v. 32, p. 1,686-1,693.

Sanford, W.E., and Konikow, L.F., 1985, A two-constituent solute-transport model for ground water having variable density: U.S. Geological Survey Water-Resources Investigations Report 85-4279, 88 p.

Sass, J.H., Lachenbruch, A.H., and Munroe, R.J., 1971, Thermal conductivity of rocks from measurements on fragments and its application to heat-flow determination: Journal of Geophysical Research, v. 76, no. 14, p. 3,391-3,401.

Smith J.A., Sahoo, D., McLellan, H.M., and Imbrigiotta, T.E., 1997, Surfactant enhanced remediation of a trichloroethene-contaminated aquifer, 1. Transport of Triton X-100: Environmental Science Technology, v. 31, p. 3,565-3,572.

Sorak S., and Pinder, G.F., 1999, Survey of computer codes and case histories, *in* Bear and others (eds.), Seawater intrusion in coastal aquifers–Concepts, methods, and practices: DordrechtKluwer Academic Publishers, p. 299-462.

Strack, O.D.L., 1995, A Dupuit-Forsheimer model for three-dimensional flow with variable density: Water Resources Investigation, v. 12, p. 3,007-3,017.

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 transport: U.S. Geological Survey Water-Resources Investigations Report 84-4369, 409 p.

Voss, C.I., 1999, USGS SUTRA Code–History, practical use, and application in Hawaii, *in* Bear and others (eds.), Seawater intrusion in coastal aquifers–Concepts, methods, and practices: Dordrecht Kluwer Academic Publishers, p. 249-313.

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. 1,851-1,866.

Voss, C.I., and Provost, A.M., 2002, SUTRA--A model for saturated-unsaturated variable-density ground-water flow with solute or energy transport: U.S. Geological Survey Water-Resources Investigations Report 02-4231, 260 p.

Weast, R.C. (ed.), 1986, CRC handbook of chemistry and physics, 66th ed.: Boca Raton, Florida, CRC Press, 2,363 p.

Yeh, G.T., 1987, 3DFEMWATER--A three-dimensional finite element model of WATER flow through saturated-unsaturated porous media: Oak Ridge, Tenn., Oak Ridge National Laboratory report ORNL-6386, 317 p.

Yeh, G.T., 1990, 3DLEWASTE--A hybrid Lagrangian-Eularian finite element model of water flow through saturated-unsaturated media: University Park, Penn., The Pennsylvania State University, Department of Civil Engineering, PSU Technical Report.

Appendix 1: Notation

Generic Units

[-]	dimensionless
[1]	unity - implies dimensionless
[0]	zero - implies dimensionless
[E]	energy units or $[ML^2/T^2]$
[L]	length units
$[L_f^3]$	fluid volume
$[L_G^3]$	solid grain volume
[M]	fluid mass units
$[M_G]$	solid grain mass units (matrix and adsorbate)
$[M_S]$	solute mass units
[T]	time units

Units

[°C]	degrees Celsius

Special Notation

$\frac{\partial \Psi}{\partial t}$ or $\frac{d\Psi}{dt}$	time derivative of Ψ
$\mathbf{v} = \mathbf{i}v_x + \mathbf{j}v_y + \mathbf{k}v_z$	vector **v** with components in the **i**, **j**, and **k** directions
$\nabla\Psi = \mathbf{i}\frac{\partial\Psi}{\partial x} + \mathbf{j}\frac{\partial\Psi}{\partial y} + \mathbf{k}\frac{\partial\Psi}{\partial z}$	gradient of scalar Ψ
$\nabla \cdot \mathbf{v} = \frac{\partial v}{\partial x}x + \frac{\partial v}{\partial y}y + \frac{\partial v}{\partial z}z$	divergence of vector **v**
$i = \overline{1, NN} = 1,2,3,4,\ldots, NN$	index i takes on all integer values between one and NN
$\Delta\Psi$	discrete change in value of Ψ (*e.g.*, $\Delta\Psi = \Psi_1 - \Psi_2$)
Ψ_o	initial value or zeroth value of Ψ
Ψ_{BC}	value of Ψ at a boundary condition node
Ψ_i or Ψ_j	value of Ψ at node or cell i or j
Ψ_{IN}	value of Ψ in inflow
Ψ_L	value of Ψ in element L
v_x	value of vector v in x direction
v_y	value of vector v in y-direction
v_z	value of vector v in z-direction
Ψ^L	value of Ψ in element L
Ψ^n	value of Ψ at time step n
Ψ^{n+1}	value of Ψ at time step n+1

$\langle\langle\varepsilon\rangle\rangle$	(3.11)	[-]	elementwise discretization of porosity
$\langle\langle\mathbf{k}^L\rangle\rangle$	(3.4)	[L^2]	permeability tensor that is discretized elementwise
$\left\langle\left\langle\sigma_s^L\right\rangle\right\rangle$	(3.16)	[L^2/T]	elementwise discretization of solid matrix thermal diffusivity for heat transport
		[0]	not used for for solute transport of species k
$\left\langle\left\langle\sigma_{bgk}^L\right\rangle\right\rangle$	(3.20)	[L^2/T]	elementwise discretization of bulk thermal diffusivity [L^2/T] that is approximated using a geometric mean and includes the constants λ_A and λ_w
		[L^2/T]	constant molecular diffusivity (D$_M$) of species k for solute transport
$\langle\langle\rho\mathbf{g}\rangle\rangle^*$	(3.5)	[M/L^{2}T^2]	elementwise discretization of ($\rho\mathbf{g}$) that is consistent with the discretization of ∇p
$\langle\langle\mathbf{v}\rangle\rangle^*$	(3.11)	[M/T]	elementwise discretization of **v** that is consistent with the discretization of ∇p
$\sum_{i=1}^{NN}\Psi_i = \Psi_1 + \Psi_2 + \Psi_3 + ... + \Psi_{NN}$			summation

Roman Lowercase

b	(5.3)	[L]	aquifer thickness
c_s	(2.6)	[E/M°C]	solid matrix specific heat for heat transport
[-]			sorption coefficient for solute transport
c_{sk}	(2.12)	[E/M°C]	solid matrix specific heat for heat transport
		[-]	sorption coefficient for species k for solute transport
c_w	(2.6)	[E/M°C]	fluid specific heat for heat transport
		[1]	for solute transport
c_{wk}	(2.12)	[E/M°C]	fluid specific heat for heat transport
		[1]	for solute transport of species k
$\mathbf{k}$	(2.1)	[L^2]	permeability tensor
k_r	(2.1)	[-]	relative permeability for unsaturated flow
$\mathbf{g}$	(2.1)	[L/T^2]	gravitational vector
$\mathbf{n}$	(3.16)	[-]	unit outward normal vector
p	(2.1)	[M/LT^2]	fluid pressure
p_i^{n+1}	(3.1)	[M/LT^2]	fluid pressure at node i at the end of the current time step

$p_{BC_i}^{n+1}$	(3.1)	$[M/LT^2]$	specified fluid pressure at node i at the end of the current time step (zero for all nodes that are not specified pressure nodes)
s_{Lk}	(3.14)	[1]	is the sorption isotherm contribution to the left-hand side of unified solute mass balance equation
s_{Rk}	(3.15)	$[M_{solute}/ M_{fluid}]$	is the sorption isotherm contribution to the right-hand side of unified solute mass balance equation
t	(2.1)	[T]	time
$\mathbf{v}$	(2.6)	[L/T]	fluid velocity

Roman Uppercase

AF_i^{n+1}	(3.1)		matrix coefficient of pressure time derivative
AT_{ik}^{n+1}	(3.9)		matrix coefficient of U time derivative for species k
$B(x,y)$	(3.6)	[L]	aquifer thickness
BF_{ij}^{n+1}	(3.1)		matrix coefficient in pressure equation
BT_{ijk}^{n+1}	(3.9)		matrix coefficient in U equation for species k
C	(2.10)	$[M_{solute}/ M_{fluid}]$	fluid solute concentration.
C_k	(2.3)	$[M_{solute}/ M_{fluid}]$	fluid solute concentration of species k
C_{ko}	(2.4)	$[M_{solute}/ M_{fluid}]$	base solute concentration of species k
C_{sk}	(3.22)	$[M_{adsorbate}/ M_{solid}]$	is the concentration of adsorbate for species k
C_{sk}^{n+1}	(3.23)	$[M_{adsorbate}/ M_{solid}]$	is the concentration of adsorbate for species k at the end of the current time step
$\left(C_{sk}^{n+1}\right)_i$	(3.36)	$[M_{adsorbate}/ M_{solid}]$	is the concentration of adsorbate for species k at node i at the end of the current time step
C_{sk}^{PROJ}	(3.31)	$[M_{adsorbate}/ M_{solid}]$	is the concentration of adsorbate for species k based on a projection for the first time step or at the end of the last time step
$\left(C_{sk}^{PROJ}\right)_i$	(3.36)	$[M_{adsorbate}/ M_{solid}]$	is the concentration of adsorbate for species k at node i based on a projection for the first time step or at the end of the last time step

C_{sk}^{0}	(3.31)	$[M_{adsorbate}/M_{solid}]$	is the approximation used for concentration of adsorbate for species k at low fluid concentrations of species k
C_{sk}^{∞}	(3.32)	$[M_{adsorbate}/M_{solid}]$	is the approximation used for concentration of adsorbate for species k at high fluid concentrations of species k
CF_{ij}^{n+1}	(3.1)		matrix coefficient of *U* time derivative in pressure equation
C_{sk}	(3.20)	$[M_{solute}/M_{solid\ matrix}]$	concentration of the adsorbate for species k
$\mathbf{D}$	(2.6)	$[L^2/T]$	solute dispersion tensor
		$[L^2/T]$	thermal dispersion tensor
$\mathbf{D}_k$	(2.12)	$[L^2/T]$	solute dispersion tensor for species k
		$[L^2/T]$	thermal dispersion tensor for species k
D_M	(5.1)	$[L^2/T]$	total solute dispersion which is the product of the molecular dispersion of NaCl and porosity
D_{NaCl}	(5.7)	$[L^2/T]$	molecular dispersion of NaCl
D_{SOLUTE}	(5.1)	$[L^2/T]$	molecular diffusivity of NaCl
D_T	(5.2)	$[L^2/T]$	total thermal dispersion which is the product of the thermal diffusivity and porosity
D_{TEMP}	(5.2)	$[L^2/T]$	thermal diffusivity
DTMULT	(4.1)	[-]	time step multiplier
DF_{ij}^{n+1}	(3.1)		element of vector on right side of pressure equation
DT_{ijk}^{n+1}	(3.9)		element of vector on right side of U equation for species k
ET_{ik}^{n+1}	(3.9)		element of vector on right side of U equation for species k
$G_sTL_{ik}^{n+1}$	(3.9)		element of vector on left side of U equation for species k
$G_sTR_{ik}^{n+1}$	(3.9)		element of vector on right side of U equation for species k
GT_{ik}^{n+1}	(3.9)		element of vector on left side of U equation for species k
$\mathbf{I}$	(2.6)	[1]	identity tensor
K	(5.3)	[L/T]	hydraulic conductivity
N_C	(5.4)	[-]	non-dimensional parameter of Henry and Hilleke (1972) for relationship of advective flux to dispersion of salt

N_T	(5.5)	[-]	non-dimensional parameter of Henry and Hilleke (1972) for relationship of advective flux to dispersion of heat
N_Ψ	(5.3)	[-]	non-dimensional parameter of Henry and Hilleke (1972) for relationship of porous media tranmissibility to advective flux
NE			number of elements in mesh
NN	(3.1)		number of nodes in mesh
NS	(2.3)	[-]	number of simulated of solute species
$Q^{n}_{BC_i}$	(3.9)	[E/T]	total energy source of species k to cell I for previous time step
		[M_{solute}/M_{fluid}T]	total concentration of species k to cell I for previous time step
Q_i	(3.1)	[M/T]	total mass source to cell i
Q_i^{n+1}	(3.1)	[M/T]	total mass source to cell i for the current time step
Q_p	(2.1)	[M/L^3T]	fluid mass source (including pure water mass plus solute mass dissolved in source water)
Q_T	(5.3)	[L^3/LT]	total freshwater inflow per unit width
R'	(5.3)	[-]	fractional difference between the density of seawater and freshwater (0.025)
S_{op}	(2.1)	[M/LT^2]$^{-1}$	specific pressure storativity
S_w	(2.1)	[-]	fractional water saturation
T_o	(2.4)	[°C]	base fluid temperature
T_c	(2.5)	[-]	
U	(2.1)	[°C] [M_{solute}/ M_{fluid}]	fluid temperature for heat transport or fluid solute concentration
U_k	(2.12)	[°C]	fluid temperature of species k for heat transport
		[M_{solute}/ M_{fluid}]	fluid solute concentration of species k for solute transport
U^{n}_{ik}	(3.9)	[°C]	temperature of cell i at the end of the previous time step
		[M_{solute}/ M_{fluid}]	concentration of species k in cell i at the end of the previous time step
U^{n+1}_{ik}	(3.9)	[°C]	temperature of cell i at the end of the current time step
		[M_{solute}/ M_{fluid}]	concentration of species k in cell i at the end of the current time step
$U^{n+1}_{UBC_{ik}}$	(3.9)	[°C]	specified temperature of cell i at the end of the current time step
		[M_{solute}/ M_{fluid}]	specified concentration of species k in cell i at the end of the current time step

U_{ik}	(3.1)	[°C]	fluid temperature of species k at node i for heat transport
		$[M_{solute}/ M_{fluid}]$	fluid solute concentration of species k at node i for solute transport
U_s	(2.6)	[E/M]	specific energy content of the solid matrix for heat transport
		$[M_{solute}/M_{solid\ matrix}]$	concentration of the adsorbate on the solid matrix for solute transport
U_{sk}	(2.12)	[E/M]	specific energy content of the solid matrix [E/M] for heat transport
		$[M_{solute}/M_{solid\ matrix}]$	concentration of the adsorbate on the solid matrix for species k for solute transport
U^*	(2.6)	[°C]	temperature of the source fluid for heat transport
		$[M_{solute}/ M_{fluid}]$	solute concentration of the source fluid for solute transport
U_k^*	(2.12)	[°C]	temperature of the source fluid for heat transport
		$[M_{solute}/ M_{fluid}]$	solute concentration of the source fluid for species k for solute transport
V_i	(3.2)	$[L^3]$	cell volume at node I
W_{0k}	(3.34)	[-]	low solid concentration weight used for Langmuir isotherm for species k
$W_{\infty k}$	(3.35)	[-]	high solid concentration weight used for Langmuir isotherm for species k

Greek Lowercase

α	(2.2)	$[M/LT^2]$	compressibility of the porous matrix
β	(2.2)	$[M/LT^2]^{-1}$	compressibility of the fluid
γ_1^w	(2.6)	[0]	not used for heat transport
		$[T^{-1}]$	first-order mass production rate of solute for solute transport
$(\gamma_1^w)_k$	(2.12)	[0]	not used for heat transport
		$[T^{-1}]$	first-order fluid mass production rate of species k for solute transport
γ_1^s	(2.6)	[0]	not used for heat transport
		$[T^{-1}]$	first-order mass production rate of adsorbate for solute transport,
$(\gamma_1^s)_k$	(2.12)	[0]	not used for heat transport
		$[T^{-1}]$	first-order mass production rate of species k adsorbate for solute transport

γ_0^w	(2.6)	[E/MT]	zero-order energy production rate in the fluid for heat transport
		$[(M_{solute}/M_{fluid})T^{-1}]$	zero-order mass production rate of solute for solute transport
$\left(\gamma_0^w\right)_k$	(2.12)	[E/MT]	zero-order energy production rate in the fluid for heat transport
		$[(M_{solute}/M_{fluid})T^{-1}]$	zero-order fluid mass production rate of species k for solute transport
γ_0^s	(2.6)	[E/MT]	zero-order energy production rate in the solid matrix for heat transport
		$[(M_{adsorbate}/M_{solid})T^{-1}]$	zero-order mass production rate of solute for solute transport
$\left(\gamma_0^s\right)_k$	(2.12)	[E/MT]	zero-order energy production rate in the solid matrix for heat transport
		$[(M_{adsorbate}/M_{solid})T^{-1}]$	zero-order mass production rate of solute for species k for solute transport
δij	(3.1)	[0 or 1]	Kronecker delta
ε	(2.1)	[-]	porosity
κ_1	(2.9)	$[M/M_G]$	first general sorption coefficient
κ_{1k}	(2.13)	$[M/M_G]$	first general sorption coefficient of species k
λ	(2.16)	[E/TL°C]	bulk thermal conductivity
λ_A	(2.23)	[E/TL°C]	thermal conductivity of air
λ_s	(2.8)	[E/TL°C]	solid thermal conductivity
λ_w	(2.7)	[E/TL°C]	fluid thermal conductivity
μ	(2.1)	[M/LT]	fluid viscosity
$\mu_o(T)$	(2.5)	[M/LT]	fluid viscosity at the base solute concentration for each species or a temperature-viscosity relationship used for non-isothermal conditions
ν	(5.6)	[-]	aquifer aspect ratio - ratio of aquifer thickness to aquifer length
ν_{pi}	(3.1)	[LT]	pressure-based conductance for the specified pressure source in cell i
ν_{Uik}	(3.9)	[LT]	concentration- or temperature-based conductance for the specified concentration or temperature of species k source in cell i
ρ	(2.1)	$[M/L^3]$	fluid density
ρ_s	(2.6)	$[M/L^3]$	density of the solid matrix
ρ_o	(2.4)	$[M/L^3]$	is the fluid density at the base mass fraction and temperature,
σ_s	(2.6)	$[L^2/T]$	solid matrix thermal diffusivity for heat transport

		[0]	not used for for solute transport
σ_{sk}	(2.12)	$[L^2/T]$	solid matrix thermal diffusivity for heat transport if species k is “ENERGY”
		[0]	not used for for solute transport of species k
σ_w	(2.6)	$[L^2/T]$	fluid thermal diffusivity for heat transport
		$[L^2/T]$	molecular diffusivity (D_M) for solute transport
σ_{wk}	(2.12)	$[L^2/T]$	fluid thermal diffusivity for heat transport if species k is “ENERGY”
		$[L^2/T]$	molecular diffusivity (D_M) of species k for solute transport
σ_{bg}	(2.24)	$[L^2/T]$	geometric-mean approximation of bulk thermal diffusivity for heat transport
σ_{bgk}	(2.25)	$[L^2/T]$	geometric-mean approximation of bulk thermal diffusivity for heat transport if species k is “ENERGY”
		$[L^2/T]$	molecular diffusivity (D_M) of species k for solute transport
ϕ_j	(3.4)		symmetric bilinear (2D) or trilinear (3D) basis function in global coordinates for node j
ϕ_{jk}	(3.11)		symmetric bilinear (2D) or trilinear (3D) basis function in global coordinates for species k at node j
ϕ_{ik}	(3.12)		symmetric bilinear (2D) or trilinear (3D) basis function in global coordinates for species k at node i
τ	(2.3)	[0 or 1]	value equals 1 if heat transport is being simulated or 0 if heat transport is not being simulated
χ_1	(2.9)	$[L_f^3 / M_G]$	linear distribution coefficient
χ_1	(2.10)	$[L_f^3 / M_G]$	Freundlich distribution coefficient
χ_2	(2.10)	[-]	Freundlich coefficient
χ_1	(2.11)	$[L_f^3 / M_G]$	Langmuir distribution coefficient
χ_2	(2.11)	[-]	Langmuir coefficient
χ_{1k}	(2.13)	$[L_f^3 / M_G]$	linear distribution coefficient for species k
χ_{1k}	(2.14)	$[L_f^3 / M_G]$	Freundlich distribution coefficient for species k
χ_{2k}	(2.14)	[-]	Freundlich coefficient for species k
χ_{1k}	(2.15)	$[L_f^3 / M_G]$	Langmuir distribution coefficient for species k

χ_{2k}	(2.15)	[-]	Langmuir coefficient for species k
ω_i	(3.4)		asymmetric weighting function in global coordinates for node i

Greek Uppercase

Γ	(3.16)		external boundary of simulated region
Δt_{NEW}	(4.1)	[T]	modified revised time step length for simple automatic time stepping algorithm
Δt^{n+1}	(3.1)	[T]	time step length for the current time step
$\Psi_{IN_{ik}}^{n+1}$	(3.9)	[E/T]	total energy source to boundary of cell i due to diffusion or dispersion for current time step
		[$M_{solute}/M_{fluid}T$]	total concentration source of species k to cell i due to diffusion or dispersion for current time step