♻EPA

3DHYDROGEOCHEM: A 3-Dimensional Model of Density-Dependent Subsurface Flow and Thermal Multispecies-Multicomponent HYDROGEOCHEMical Transport

3DHYDROGEOCHEM:
A 3-Dimensional Model of Density-Dependent Subsurface Flow and Thermal Multispecies-Multicomponent HYDROGEOCHEMical Transport

by
Gour-Tsyh (George) Yeh and Hwai-Ping (Pearce) Cheng
Department of Civil and Environmental Engineering
Pennsylvania State University

Disclaimer

The development of 3DHYROGEOCHEM was supported by the Subsurface Science Program, Office of Health and Environmental Research, U.S. Department of Energy under contract with the Pennsylvania State University. The final phase of this document was prepared under the sponsorship by National Risk Management Research Laboratory, U.S. Environmental Protection Agency. This report has been subjected to the Agency's peer and adminstrative review and has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

All research projects making conclusions or recommendations based on environmentally related measurements and funded by the Environmental Protection Agency are required to participate in the Agency Quality Assurance Program. This project did not involve environmentally related measurements and did not involve a Quality Assurance Project Plan.

When available, the software described in this document is supplied on "as is" basis without guarantee or warranty of any kind, express or implied. Neither the United States Government (United States Environmental Protection Agency), the Pennsylvania State University, nor any of the authors accept any liability resulting from use of this software.

FOREWORD

The U.S. Environmental Protection Agency is charged by Congress with protecting the Nation's land, air, and water resources. Under a mandate of national environmental laws, the Agency strives to formulate and implement actions leading to a compatible balance between human activities and the ability of natural systems to support and nurture life. To meet these mandates, EPA's research program is providing data and technical support for solving environmental problems today and building a science knowledge base necessary to manage our ecological resources wisely, understand how pollutants affect our health, and prevent or reduce environmental risks in the future.

The National Risk Management Research Laboratory (NRMRL) is the Agency's center for investigation of technological and management approaches for reducing risks from threats to human health and the environment. The focus of the Laboratory's research program is on methods for the prevention and control of pollution to air, land, water, and subsurface resources; protection of water quality in public water systems; remediation of contaminated sites and ground water; and prevention and control of indoor air pollution. The goal of this research effort is to catalyze the development and implementation of innovative, cost-effective environmental technologies; develop scientific and engineering information needed by EPA to support regulatory and policy decisions; and provide technical support and information transfer to ensure effective implementation of environmental regulations and strategies.

Subsurface fate and transport models provide (1) a tool of application, with which one is able to deal with a variety of real-world problems, (2) a tool of education, with which one can study how a factor would affect the whole system, and (3) a substructure, which one could modify to handle specific problems. The hydrological environment to which this model is applied includes heterogeneous, anisotropic, saturated-unsaturated subsurface medium, and either transient-state or steady-state flow conditions. This report presents a three-dimensional coupled model of subsurface flow, chemical transport, and heat transfer (3DHYDROGEOCHEM). It is developed mainly for dealing with reactive chemical transport problems, under the local equilibrium assumptions in subsurface systems.

Clinton W. Hall, Director
Subsurface Protection and Remediation Division
National Risk Management Research Laboratory

Abstract

This report presents a three-dimensional finite-element numerical model designed to simulate chemical transport in subsurface systems with temperature effect taken into account. The three-dimensional model is developed to provide (1) a tool of application, with which one is able to deal with a variety of real-world problems, (2) a tool of education, with which one can study how a factor would affect the whole system, and (3) a substructure, which one could modify to handle specific problems. The hydrological environment to which the model can be applied is a heterogeneous, anisotropic, saturated-unsaturated subsurface medium under either transient-state or steady-state flow conditions. In addition, the temperature within the system of interest can be both time- and location-dependent. For steady-state simulations, strong coupling among subsurface flow, chemical transport, and heat transfer is used in the model. For transient-state simulations, weak coupling is used, but a density effect is still considered in computations. Both the strong and the weak couplings are pictured and explained. The model employs chemical equilibrium to describe the relationship among chemicals. The chemical reactions included in the model are aqueous complexation, multi-site adsorption/desorption, multi-site ion-exchange, precipitation/dissolution, redox, and acid-base reactions. To discretize the domain of interest appropriately, the element used in the model can be a hexahedral, a triangular prism, or a tetrahedral element. To extend its applicability to more real-world problems, two approaches are presented for the chemical transport module in this report. The first approach uses the pore velocity and dispersion coefficient to handle advection and dispersion, respectively, for aqueous components, whereas the second approach employs the retarded pore velocity and the retarded dispersion coefficient. Both approaches are designed to handle the problems with dominating precipitated species involved. The governing equations of subsurface flow, chemical transport, chemical equilibrium, and heat transfer are stated and/or derived. The numerical approaches with the finite element method to solve the governing equations are described. In this study, 25 designed examples have been used to verify the model. Four application examples, including a three-dimensional subsurface flow example, a three-dimensional reactive chemical transport example, a three-dimensional heat transfer example, and a three-dimensional coupled flow-transport-transfer example, are presented to demonstrate the capability of the model. The modeling experience from this study is summarized in the last chapter of this report. Data input guide and program description are stated in Appendix A and Appendix B, respectively. In Appendix C, parameter specification for array dimensions in 3DHYDROGEOCHEM is given.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
INTRODUCTION

3DHYDROGEOCHEM (a 3-Dimensional coupled model of HYDROlogic transport, GEOCHEMical equilibria, subsurface flow, and heat transfer in reactive multi-component/multi-species systems) has been developed mainly to study reactive chemical transport in subsurface systems. It is composed of two basic modules: flow and transport. In the flow module, the Galerkin finite element method is used to discretize the modified Richards' equation for simulating density-dependent subsurface flow. In the transport module, both reactive chemical transport and heat transfer are simulated. Their governing equations are discretized by using either the conventional Eulerian finite element method or the hybrid Lagrangian-Eulerian finite element method. In reactive chemical transport, solute transport is coupled with chemical equilibrium to consider the interaction among chemical species as they are transported by subsurface flow. For simplicity, the term "chemical transport" is used to represent reactive chemical transport.

In subsurface systems, chemical transport is subject to both hydrophysical and hydrogeochemical processes [Yeh and Tripathi, 1990; 1991]. The Darcy flux is determined based on the spatial distributions of hydraulic head as well as the dissolved chemical concentrations in the systems. The law of mass balance must be satisfied in deriving both the flow and the transport governing equations. Numerically, subsurface flow and chemical transport can be dealt with either weak coupling when the influence of the chemical concentration on flow is minor, or strong coupling when that effect is significant. For a weakly coupled system, one solves the flow equation for both velocity and moisture content distributions, and then uses those two hydrological variables as the input data to solve the transport equation for the distribution of chemical concentrations. For a strongly coupled system, flow and transport processes must be solved either simultaneously or iteratively [Cheng and Yeh, 1993b]. Heat transfer will also be coupled with flow and transport if the temperature effect is significant. Figure 1.1 pictures the basic structure of strong coupling. Figure 1.2 shows the basic structure of weak coupling. In both figures, H, C, and T represent the computed variables associated with subsurface flow, chemical transport, and heat transfer, respectively. HW, CW, and TW are the working values of H, C, and T, respectively. In Figure 1.1, an iteration loop is used for the strongly coupled subsurface flow, heat transfer, and chemical transport, whereas the computation with weak coupling for any time step is achieved based on the previous-time values of system variables (e.g., hydraulic head, temperature, and concentration) as shown in Figure 1.2. For clarity, subsurface flow, chemical transport, and heat transfer are considered to be three separate modules in 3DHYDROGEOCHEM.

3DHYDROGEOCHEM has the following features which make it flexible and versatile in modeling a wide range of real-world problems.

(1) Irregularly-shaped domains can be dealt with.
One of the main reasons that the finite element method has been rapidly developed and utilized in the past decade is its flexibility in dealing with irregularly-shaped domains, which are mostly associated with real-world problems. To increase the ability in handling irregular boundaries, 3DHYDROGEOCHEM contains three shapes of elements: hexahedral, triangular prism, and tetrahedral. Figure 1.3 demonstrates a combination of these three types of elements. The only requirement for using more than one type of element to discretize the domain of interest is the consistency of both element connection and base (shape) functions.

(2) Both heterogeneous and anisotropic media, as many as desired, can be taken into account.
3DHYDROGEOCHEM is designed to be capable of dealing with subsurface systems which contain more than one material. For each material, one may provide values for the six nonzero components of the saturated conductivity/permeability tensor in the Richards' equation to consider the case of anisotropy. As shown in Eq. (2.1), the saturated conductivity/permeability tensor, $\mathbf{K}$, has three diagonal components, K_{xx}, K_{yy}, and K_{zz}, and three off-diagonal ones, K_{xy}, K_{xz}, and K_{yz}.

$$\mathbf{K} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{xy} & K_{yy} & K_{yz} \\ K_{xz} & K_{yz} & K_{zz} \end{bmatrix} \tag{2.1}$$

```
┌──────────────────────────────────────────────────────────┐
│  Set up the initial guess for the coupling iteration loop:  │
│              HW=H, CW=C, TW=T.                              │
└──────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────┐
│      Increment of the count of coupling iteration.         │
└──────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────┐
│   Compute the new iterative hydraulic head, H, based       │
│     on the working iterative values HW, CW, and TW.        │
└──────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────┐
│   Compute the new iterative temperature, T, based on       │
│     the working iterative values HW, CW, and TW.           │
└──────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────┐
│   Compute the new iterative concentration, C, based        │
│     on the working iterative values HW, CW, and TW.        │
└──────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────┐
│       Check convergence by comparing H and HW,             │
│          C and CW, and T and TW.                           │
└──────────────────────────────────────────────────────────┘

      If not convergent                    If convergent

┌────────────────────────────────────┐   ┌────────────────────────────────┐
│  Prepare new working iterative      │   │   End of the strong coupling;   │
│  values, HW, CW, and TW, for the    │   │      H=HW, C=CW, T=TW.          │
│  next iteration.                    │   └────────────────────────────────┘
└────────────────────────────────────┘
```

Figure 1.1 The basic structure of strong coupling.

```
┌──────────────────────────────────────────────────────────┐
│  Set up the initial working values from the previous time step:  │
│              HW=H, CW=C, TW=T.                              │
└──────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────┐
│       Compute the new hydraulic head, H, based             │
│         on the working values HW, CW, and TW.              │
└──────────────────────────────────────────────────────────┘
                          │       Update the working value HW
                          │       based on H and the previous HW.
                          ▼
┌──────────────────────────────────────────────────────────┐
│       Compute the new temperature, T, based on            │
│          the working values HW, CW, and TW.               │
└──────────────────────────────────────────────────────────┘
                          │       Update the working value TW
                          │       based on T and the previous TW.
                          ▼
┌──────────────────────────────────────────────────────────┐
│       Compute the new concentration, C, based             │
│          on the working values HW, CW, and TW.            │
└──────────────────────────────────────────────────────────┘
                          │       Update the working value CW
                          │       based on C and the previous CW.
                          ▼
┌──────────────────────────────────────────────────────────┐
│        End of the weak coupling;                           │
│            H=HW, C=CW, T=TW.                               │
└──────────────────────────────────────────────────────────┘
```

Figure 1.2 The basic structure of weak coupling.
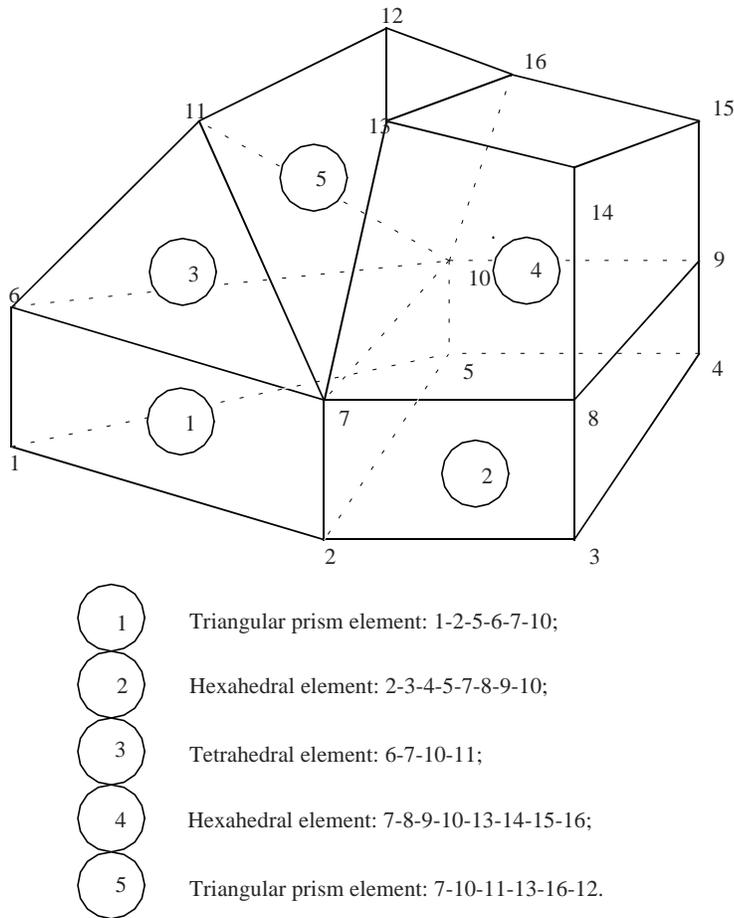
Figure 1.3 A combination of three types of elements.

In addition, a number of parameters, including the referenced water density and dynamic viscosity, the bulk density, the longitudinal and transverse dispersivities, the tortuosity, the specific heat, the apparent thermal conductivity, and the saturated moisture content of each material are to be provided by users, for the related computations.

(3) Both steady-state and transient-state simulations can be processed.
The model can solve for steady-state and/or transient-state solutions under appropriate given boundary and/or initial conditions. In steady-state simulations, strong coupling is used to reflect real situations being considered. In transient-state simulations, weak coupling is employed to save computational time. As shown in Figure 1.2, for each time step, the subsurface flow equation is first solved using the information computed from the previous time step, and then the flow information is updated. Second, the heat transfer equation is solved using the partially-updated information, and then the temperature information is updated. Finally, the chemical transport equations are solved using the partially-updated information, and then the chemical concentration information is updated for use in the next time step. There is no iteration loop existing for weak coupling. Rather, the updated information is used directly.

(4) The initial conditions can be either prescribed or obtained by simulating a steady-state system under consideration.
The initial conditions of a transient simulation cannot be arbitrary. An inappropriate initial condition may introduce either nonconvergent or nonrealistic solutions. It should be specified as close to the real situation as possible. In other words, it should be as natural as possible. However, sometimes it is not straightforward to prescribe natural initial conditions. For instance, without any computation, it is difficult to accurately describe the hydraulic head distribution within an irregularly-shaped domain under given boundary conditions (Figure 1.4) when the hydraulic head no longer changes with time. In this model, an option to compute the initially steady-state conditions based on given boundary conditions is provided. Therefore, the initial conditions can be set up by either prescription or the computation of the initial steady-state conditions.

3

$h_1 = 10$, $h_2 = 3$, $h_3 = 8$, $h_4 = 5$.
Plane A-B-C-D is a curvalinear plane.
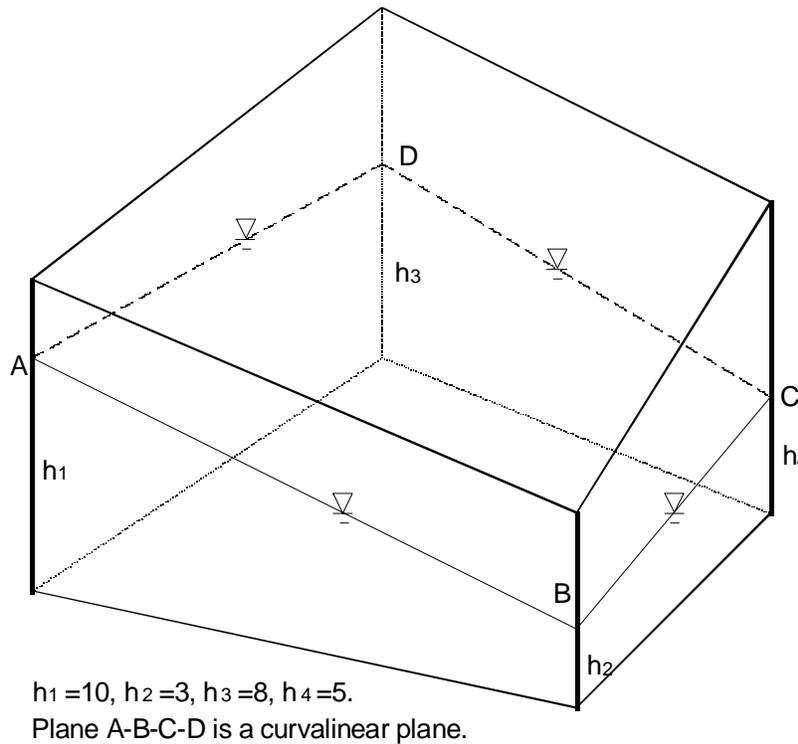
Figure 1.4 The initial hydraulic head distribution under given boundary conditions for irregular domains.

(5)   Both distributed and point sources/sinks can be considered spatially- and/or temporally-dependent.
The distributed (elemental) sources/sinks simulate the volumetric sources/sinks in the real world. The point (nodal) sources/sinks simulate well injection/withdrawal occurring in real-world problems. A specific composition of several point sources/sinks can be considered to represent a line or a plane source/sink. The source/sink intensity can be functions of time.

(6)   All types of boundary conditions can be considered spatially- and/or temporally-dependent.
Four types of boundary conditions (Dirichlet, Variable, Cauchy, and Neumann) are employed for the simulations of subsurface flow, heat transfer, and chemical transport. These four boundary conditions are Dirichlet, Cauchy, Neumann, and Variable boundary conditions.   Table 1.1 briefly describes these boundary conditions and the data needed for describing them. All boundary conditions can be prescribed as functions of time.

(7)   The appropriate Variable boundary conditions are determined automatically.
A Variable boundary is a boundary on which the subsurface flow direction is not predetermined.   It needs to be set up so that consistent computational results can be obtained. In other words, Variable boundary conditions and computational results depend on each other. If Variable boundaries exist in the simulation of subsurface flow, then a nonlinear iteration loop for adjusting Variable boundary conditions is employed to assure that the convergent computational results are consistent to the final Variable boundary conditions. Figure 1.5 shows the logic of adjusting Variable boundary conditions in the nonlinear iteration loop of the subsurface flow simulation. When a boundary is flux-type for the rainfall period, a complete adsorption of throughfall water is assumed, while a potential evapotranspiration is simulated if it is for the evaporation period. The ponding-type boundary is to simulate the accumulation of water above ground surface. The minimum pressure-type boundary is to describe the allowed minimum pressure associated with the soil being considered.
On the other hand, there is no nonlinear iteration loop used for adjusting Variable boundary conditions in the simulations of both heat transfer and chemical transport. The subsurface flow has to be determined before either the heat transfer or the chemical transport simulation can be computed. Then, the flow direction at a Variable boundary point is no longer undetermined. If the flow is directed into the domain of interest, the chemical concentration of the flow-in groundwater (i.e., the flow-in total dissolved concentration) and the temperature of the flow-in groundwater (i.e., the flow-in temperature) need to be prescribed for the chemical transport and the heat transfer simulations, respectively, so that the flow-in flux can be calculated. However, if the flow is outward, then a Neumann boundary condition of zero flux is implemented

Table 1.1 Boundary conditions.

Simulation type	B.C. Type	When to use	Input data needed
G. W. Flow	Dirichlet	Hydraulic head is prescribed	Total head distribution w.r.t. time
	Cauchy	Total water flux is prescribed	Total water flux distribution w.r.t. time
	Neumann	Water flux due to pressure head gradient is prescribed	Water flux distribution w.r.t. time
	Variable	Subsurface flow on the boundary is not predetermined	Rainfall/evaporation rate, ponding depth, and minimum pressure head distributions w.r.t. time
C. Transport	Dirichlet	Concentration is prescribed	Total analytical concentration w.r.t. time
	Cauchy	Total chemical flux is prescribed	Total chemical flux w.r.t. time
	Neumann	Diffusive chemical flux is prescribed	Diffusive chemical flux w.r.t. time
	Variable	Subsurface flow on the boundary is not predetermined	Flow-in total dissolved concentration distribution w.r.t. time
H. Transfer	Dirichlet	Temperature is prescribed	Temperature distribution w.r.t. time
	Cauchy	Total heat flux is prescribed	Total heat flux distribution w.r.t. time
	Neumann	Conductive heat flux is prescribed	Conductive heat flux distribution w.r.t. time
	Variable	Subsurface flow on the boundary is not predetermined	Flow-in temperature distribution w.r.t. time

(8)   The "in-element" particle tracking technique is used to accurately and efficiently perform particle tracking. The process of particle tracking is the principal portion of the Lagrangian step in the chemical transport/ heat transfer transient simulation. An inaccurate particle tracking scheme will lead to not only unreliable but also unreasonable solutions. However, an accurate particle tracking scheme might demand a significant amount of computational efforts. The "in-element" particle tracking technique is designed to consider both accurate and efficient particle tracking for real-world problems. This technique basically contains two parts: tracking a fictitious particle one element by one element within the domain of interest and tracking a particle one subelement by one subelement within the element being considered. Figure 1.6 explains the relationship between these two parts schematically. The more subelements that are used in the second part, the more accurate particle tracking results will be obtained and the more computational efforts are needed. The number of subelements needed to get satisfactory results can be roughly determined by examining the complexity of the velocity fields in the discretized elements. The more complex the velocity fields, the more subelements are needed [Cheng et al., 1996a].

(9)   Either the conventional Eulerian or the Lagrangian-Eulerian finite element method can be used to solve transport equations.
The Lagrangian-Eulerian finite element method has been broadly used to solve transient transport equations, especially for the advection-dominant transport equations, in the past several years. It attracts attention mainly because it considers the superposition nature of a typical linear advection-dispersion transport equation and it is able to greatly reduce most numerical errors caused by the advection term [Yeh, 1990]. In addition, if the linear group base functions (i.e., linear, bi-linear, tri-linear base functions) are utilized, then the numerical results of the Lagrangian step are always non-negative. This is very important for the chemical transport simulation because negative concentrations are neither allowed nor realistic to be used to compute chemical equilibrium and/or kinetics. An option to use either the Lagrangian-Eulerian finite element method or the conventional Eulerian finite element method to deal with the transient-state simulation of chemical transport and/or heat transfer is provided by the model. For steady-state simulations, the conventional finite element method is used.

(10)   A numerical scheme of capturing peaks/valleys is employed to increase the computational accuracy of chemical transport.
Instead of using any adaptive local grid refinement numerical scheme to increase the computational accuracy, a peak/valley numerical scheme is employed as an alternative to improve the numerical results

Start adjusting Variable boundary conditions.

Check which period is under consideration, the rainfall period? or the evaporation period?

For the rainfall period

For the evaporation period

If the boundary of the previous nonlinear iteration is flux-type, then check if the computed pressure head is larger than the prescribed ponding depth?

No

Yes

The boundary is changed to the ponding-type boundary.

If the boundary of the previous nonlinear iteration is flux-type, then check if the computed pressure head is larger than the prescribed minimum pressure head?

Yes

No

The boundary is changed to the minimum pressure-type boundary.

If the boundary of the previous nonlinear iteration is ponding-type, then check if the computed Darcy flux is larger than the prescribed rainfall rate?

Yes

No

The boundary is changed to the flux-type boundary.

If the boundary of the previous nonlinear iteration is ponding-type, then check if the computed Darcy flux larger than 0 (outward)?

Yes

No

The boundary is changed to the flux-type boundary.

Automatically convert the minimum pressure-type boundary of the previous nonlinear iteration loop to the flux-type boundary because it is not possible to have the minimum pressure-type boundary existing during the rainfall period.

If the boundary of the previous nonlinear iteration is minimum pressure-type, then check if the computed Darcy flux is larger than the prescribed evaporation rate ?

No

Yes

The boundary is changed to the minimum pressure-type boundary.

End adjusting Variable boundary conditions.

Note: The inward flux is negative and the outward one is positive

Figure 1.5 The flow chart of automatically adjusting Variable boundary conditions for subsurface flow.

as well as to save computational time. This scheme is only applied to solving chemical transport when the Lagrangian-Eulerian finite element method is utilized for transient-state simulations. More details regarding this scheme can be found elsewhere [Cheng, 1995].

(11) Multi-adsorbing site and/or multi-ion exchange site options are available to account for chemical reactions. To be capable of dealing with more complicated real-world problems, the model provides the options of multi-adsorbing sites and multi-ion exchange sites for chemical reactions. For adsorbing sites, the surface

```
┌─────────────────────────┐
│ Start particle tracking. │
└─────────────────────────┘

┌──────────────────────────────────────────┐
│ Determine the first working element, which is │
│ the element the particle is originally located in. │
└──────────────────────────────────────────┘

┌──────────────────────────────────────────┐
│ Start the elemental loop of particle tracking. │
└──────────────────────────────────────────┘

┌──────────────────────────────────────────┐
│ Construct the subelements for the working element. │
└──────────────────────────────────────────┘

┌──────────────────────────────────────────┐
│ Determine the first working subelement, which │
│ is the subelement in the working element the │
│ particle is originally located in. │
└──────────────────────────────────────────┘
```

The element loop

```
┌──────────────────────────────────────────┐
│ Start the subelemental loop of particle tracking. │
└──────────────────────────────────────────┘

┌──────────────────────────────────────────┐
│ Determine the target point in the working subelement. │
└──────────────────────────────────────────┘

      < Is there time available for the next tracking? >   No
```

The subelement loop

```
      Yes

      < Is the next tracking still in this working element? >   No

      Yes

┌──────────────────────────────────────────┐
│ Determine the successive working subelement │
│ for particle tracking. │
└──────────────────────────────────────────┘

┌──────────────────────────────────────────┐
│ End the subelemental loop of particle tracking. │
└──────────────────────────────────────────┘

┌──────────────────────────────────────────┐
│ Determining the successive working element for particle tracking. │
└──────────────────────────────────────────┘

┌──────────────────────────────────────────┐
│ End the elemental loop of particle tracking. │
└──────────────────────────────────────────┘

┌─────────────────────────┐
│ End particle tracking. │
└─────────────────────────┘
```

Figure 1.6 The basic structure of "in-element" tracking technique.

complexation, the constant capacitance, or the triple-layer model can be used. For ion-exchange sites, cation- and/or anion-exchanged site can be assigned as desired.

(12) Many options are available to both compose and solve matrix equations.

As can be imagined, a reactive chemical transport system has the strong characteristics of nonlinearity from a mathematical point of view. Since there is no unique way to guarantee achieving convergent solutions for nonlinear systems from a numerical point of view, the best way to achieve the convergent

soluti!ons is to have as many approaches available as possible. In 3DHYDROGEOCHEM, the following options are provided to attain convergent solutions.

(a) Three options (under- , exact-, and over-relaxation) are available to estimate the matrix in the strong coupling loop of subsurface flow, chemical transport, and heat transfer.
(b) Three options (under-, exact-, and over-relaxation) are available to estimate the matrix in the nonlinear loops (i.e., the rainfall-evaporation loop in the subsurface and the solute transport-chemical reaction coupling loop in the chemical transport).
(c) Four options (successive subregion block iteration, successive point iteration, polynomial preconditioned conjugate gradient, and incomplete Cholesky preconditioned conjugate gradient methods) are available to solve the linear/linearized matrix equations.
(d) Two options (consistent and lumping) are available to treat the mass matrix.
(e) The time derivative weighting factors can be assigned, ranging from 0.0 to 1.0, to change the type of numerical scheme.
(f) Either the N+1 upstream or the Galerkin weighting functions can be used to compose the matrix equations when transport is simulated by the conventional Eulerian finite element method. When the N+1 upstream weighting functions of an element are calculated, both the velocities and dispersion coefficients associated with every edge of the element are involved.
In the model, the basic strategy for composing and solving matrix equations is to make the linearized matrix equations as SPD (symmetric positive definite) as possible and adjust the relaxation factors to treat the nonlinear parts. Different solvers are set up not only to obtain convergent solutions but also to speed up the convergency process under certain conditions.

(13) The time step size can be reset automatically when boundary conditions and/or sources/sinks change abruptly.
It is usually necessary to have a small time-step size when a drastic change of either boundary conditions or sources/sinks occurs. Such a change, in general, produces a sharp front in the distribution of the variable of interest. If the geometric discretization is not fine enough or the time step size is not small enough, then a convergent solution might not be obtained under such a change. As the simulation continues, the sharp front is smoothed out and the time step size is allowed to be greater. In the model, the time step size can be originally set small and be increased gradually to a desired extent. It can also be reset to the original small value as many times as needed. What one needs to do is to prescribe when the time step size needs to be reset. The computer code will do the reset according to the prescribed values of time. (Data Set 4 in Appendix A)

(14) The examination of mass balance over the entire domain of interest for every time step is provided.
3DHYDROGEOCHEM can generate the mass balance table for the time step of interest. The table includes the flux through each type of boundary nodes and the mass increased in the domain of interest. By checking with these values, one is able to determine whether or not the simulation results can realistically represent what happens in the system being considered.

(15) Many types of chemical reactions are included in 3DHYDROGEOCHEM.
The model is able to compute the chemical equilibrium of a system with (1) aqueous complexation, (2) adsorption/desorption, (3) ion-exchange, (4) precipitation/dissolution, (5) redox, and (6) acid-base reactions involved. There is no limit on the number of reactions or chemical species that can be considered in the model once chemical equilibrium is valid. From the mathematical point of view, any reaction that can be described by equations specified in Section 2.

(16) The extension of chemical reactions from chemical equilibrium to kinetics can be implemented without much difficulty in formulation.
Since the model uses the law of mass action, instead of the Gibbs free energy function of thermodynamics, to describe chemical equilibrium, there is no difficulty to extend the description to chemical kinetics from a viewpoint of mathematical formulation [Iskra, 1994]. Table 1.2 briefly describes 3DHYDROGEOCHEM from another view.

Table 1.2 The property of 3DHYDROGEOCHEM

Model Number	18
Model	3DHYDROGEOCHEM
Authors	Cheng and Yeh
Year	1995
Dimensions	3
Coupling Relation[a]	F-H-C-T
Coupling Method[b]	Direct and 2-step
Activity Coefficient[c]	Davies
Aqueous Complexation	yes
Sorption/Desorption[d]	IE, SC, SL, and TL
Precipitation/Dissolution	yes
Redox	yes
Temperature[e]	V

[a] *F-H-C-T denotes coupling among the flow equation, the heat transfer equation, the chemical relations, and the transport equations.*
[b] *Direct denotes strong coupling among the flow equation, the heat transfer equation, the chemical relations, and the transport equations; 2-step, weak coupling among flow, transfer, and transport but still strong coupling between the chemical relations and the transport equations.*
[c] *D-H denotes the extended Debye-H ckel equation.*
[d] *IE denotes ion-exchange; SC, surface complexation; SL, single-layer model; and TL, triple-layer model.*
[e] *The temperature distribution can be either read in or computed.*

Here we also give a comparison among the computer models that were developed by G.-T. Yeh and his co-workers in relation to 3DHYDROGEOCHEM as in the following Table. (Table 1.3)

Table 1.3 Models developed by G.-T. Yeh and his co-workers in relation to 3DHYDROGEOCHEM

Model	Dimension	Main Capabilities
FEMWATER or HYDROFLOW [Yeh and Ward, 1980]	2	Computing saturated/unsaturated subsurface flow. Galerkin finite element discretization. Incorporating Variable boundary condition. Using triangular and/or quadrilateral elements.
3DFEMWATER [Yeh and Cheng, 1993a]	3	Computing saturated/unsaturated subsurface flow. Galerkin finite element discretization. Incorporating Variable boundary condition. Using hexahedral, triangular prism and/or tetrahedral elements.
FEMWASTE [Yeh and Ward, 1981]	2	Computing solute transport in saturated/unsaturated porous media. Galerkin or N+1 upstream weighting finite element discretization (Conventional Eulerian finite element method). Incorporating Variable boundary condition. Incorporating linear, Langmuir, and Freundlich isotherms. Using triangular and/or quadrilateral elements.
LEWASTE [Yeh and Cheng, 1993c]	2	Computing solute transport in saturated/unsaturated porous media. Hybrid Lagrangian-Eulerian or conventional Eulerian finite element method. Incorporating Variable boundary condition. Incorporating linear, Langmuir, and Freundlich isotherms. Using triangular and/or quadrilateral elements.
3DLEWASTE [Yeh and Cheng, 1993b]	3	Computing solute transport in saturated/unsaturated porous media. Hybrid Lagrangian-Eulerian or conventional Eulerian finite element method. Incorporating Variable boundary condition. Incorporating linear, Langmuir, and Freundlich isotherms. Using hexahedral, triangular prism and/or tetrahedral elements..
2DFEMFAT	2	Simulating coupled system of subsurface flow and solute transport by combining FEMWATER and LEWASTE. (The 2-D version of 3DFEMFAT)
3DFEMFAT [Yeh et al., 1994]	3	Simulating coupled system of subsurface flow and solute transport by combining 3DFEMWATER and 3DLEWASTE.
EQMOD [Yeh, 1992]	0	Computing chemical equilibrium among (1) complexation, (2) adsorption/desorption, (3) ion-change reactions, (4) precipitation/dissolution, (5) redox, and (6) redox reactions. Unlimited number of chemical reactions and chemical species.
KEMOD [Yeh et al., 1995a]	0	Computing chemical kinetics and/or equilibrium among (1) complexation, (2) adsorption/desorption, (3) ion-change reactions, (4) precipitation/dissolution, (5) redox, and (6) redox reactions. Unlimited number of chemical reactions and chemical species.
BIOKEMOD [Salvage and Yeh, 1998]	0	Computing chemical kinetics and/or equilibrium among (1) complexation, (2) adsorption/desorption, (3) ion-change reactions, (4) precipitation/dissolution, (5) redox, and (6) redox reactions. Incorporating microbiological effects. Unlimited number of chemical reactions and chemical species.
HYDROGEOCHEM [Yeh and Tripathi, 1990]	2	Simulating multi-component/multi-species reactive chemical transport by combining both EQMOD and FEMWASTE.
LEHGC [Yeh et al., 1995b]	2	Simulating multi-component/multi-species reactive chemical transport by combining both EQMOD and LEWASTE.
2DHYDROGEOCHEM	2	Simulating coupled systems of subsurface flow, heat transfer, and reactive chemical transport by combining EQMOD and 2DFEMFAT (The 2-D version of 3DHYDROGEOCHEM).
3DHYDROGEOCHEM [Cheng, 1995]	3	Simulating coupled systems of subsurface flow, heat transfer, and reactive chemical transport by combining EQMOD and 3DFEMFAT.

Chapter 2
MATHEMATICAL STATEMENTS

2.1 The Governing Equation of Subsurface Flow

The flow of groundwater is governed by the principles of conservation of mass and momentum. Like most flow models, 3DHYDROGEOCHEM applies Darcy's law as the general equation of the motion for groundwater in subsurface systems. That is, the linear laminar flow is assumed during investigation. The governing equation for flow is basically the modified Richards' equation, which is derived in the documentation of the 3DFEMFAT model [Yeh et al., 1994]. Under the assumption rigid media, the governing equation can be written as follows.

$$\frac{\rho}{\rho_o} \frac{d\theta}{dh} \frac{\partial h}{\partial t} = \nabla \cdot \left[\mathbf{K} \cdot \left(\nabla h + \frac{\rho}{\rho_o} \nabla z \right) \right] + \frac{\rho^*}{\rho_o} q \tag{2.1}$$

where h is the pressure head; t is time; $\nabla \cdot$ is the divergence operator; $\mathbf{K}$ is the hydraulic conductivity tensor; z is the potential head; q is the source and/or sink; ρ is the groundwater density with dissolved chemical concentrations; ρ_o is the referenced groundwater density; ρ^* is the density of either the injected or the withdrawn groundwater; and θ is the moisture content.

The hydraulic conductivity $\mathbf{K}$ is given by

$$\mathbf{K} = \frac{\rho g}{\mu} \mathbf{k} = \frac{(\rho/\rho_o)}{(\mu/\mu_o)} \frac{\rho_o g}{\mu_o} \mathbf{k_s} k_r = \frac{\rho/\rho_o}{\mu/\mu_o} \mathbf{K_{so}} k_r \tag{2.2}$$

where μ is the dynamic viscosity of the groundwater with dissolved chemical concentrations; μ_o is the referenced groundwater dynamic viscosity; g is the gravity; $\mathbf{k}$ is the permeability tensor; $\mathbf{k_s}$ is the saturated permeability tensor; k_r is the relative permeability or relative hydraulic conductivity; and $\mathbf{K_{so}}$ is the referenced saturated hydraulic conductivity tensor. The referenced values are usually taken at zero dissolved chemical concentration.

The Darcy flux is calculated as follows

$$\mathbf{V} = -\mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) \tag{2.3}$$

2.1.1 Initial Conditions for the Governing Equation of Subsurface Flow

The initial conditions used for transient-state simulations can be written in the form of

$$h = h^{initial}(x, y, z) \text{ in R} \tag{2.4}$$

where R is the domain of interest and $h^{initial}$ is the initial hydraulic head distribution over R. This initial distribution can be obtained by either field measurements or by solving the steady-state version of Eq. (2.1).

2.1.2 Boundary Conditions for the Governing Equation of Subsurface Flow

Four types of boundary conditions, from a mathematical point of view, are considered in 3DHYDROGEOCHEM to deal with a variety of physical phenomena that can occur on the boundary. They are:

(1) Prescribed total head (Dirichlet) boundary conditions:

$$h + z = H_D(x_b, y_b, z_b, t) \qquad \text{on } B_D \tag{2.5}$$

where (x_b, y_b, z_b) is the spatial coordinate on the boundary; H_D is the prescribed functional value of the total head on the Dirichlet boundary B_D. A Dirichlet boundary is a boundary where the hydraulic head is described. For instance, the bottom surface of a river or a lake can be defined as a Dirichlet boundary of the hydrologically connected subsurface system and the water depth can be used as the pressure head on the boundary under the hydrostatic assumption.

    (2)   Neumann boundary conditions:

$$-\mathbf{n} \cdot \mathbf{K} \cdot \frac{\rho_o}{\rho} \nabla h = q_N(x_b, y_b, z_b, t) \qquad \text{on } B_N \qquad (2.6)$$

where $\mathbf{n}$ is the outward unit vector normal to the boundary; q_N is the prescribed functional value of the Neumann flux on the Neumann boundary B_N. A Neumann boundary is a boundary where the groundwater flux due to the pressure head gradient is described. On the drainage boundary, for example, the flux is due to gravity and it can be considered a Neumann boundary with a zero Neumann flux.

    (3)   Cauchy boundary conditions:

$$-\mathbf{n} \cdot \mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) = q_C(x_b, y_b, z_b, t) \qquad \text{on } B_C \qquad (2.7)$$

where q_C is the prescribed functional value of the Cauchy flux on the Cauchy boundary B_C. A Cauchy boundary is a boundary where the total groundwater flux, due to both the pressure and the potential head gradients, is described. Once the groundwater flux is given on a boundary, the boundary can be treated as a Cauchy boundary. For example, the infiltration rate and ground surface can be considered a Cauchy flux and a Cauchy boundary, respectively, if the infiltration rate is measured.

    (4a)   Variable boundary conditions - during precipitation period:

$$h = h_p(x_b, y_b, z_b, t) \qquad \text{on } B_V \qquad (2.8a)$$

or

$$-\mathbf{n} \cdot \mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) = q_p(x_b, y_b, z_b, t) \qquad \text{on } B_V \qquad (2.8b)$$

    (4b)   Variable boundary conditions - during non-precipitation period:

$$h = h_p(x_b, y_b, z_b, t) \qquad \text{on } B_V \qquad (2.8c)$$

or

$$h = h_m(x_b, y_b, z_b, t) \qquad \text{on } B_V \qquad (2.8d)$$

or

$$-\mathbf{n} \cdot \mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) = q_e(x_b, y_b, z_b, t) \qquad \text{on } B_V \qquad (2.8e)$$

where h_p is the allowed ponding depth and q_p is the throughfall of precipitation, respectively, on the Variable boundary B_V; h_m is the allowed minimum pressure head and q_e is the allowed maximum evaporation rate on the Variable boundary B_V. A Variable boundary is a boundary where the flow cannot be predetermined. For example, the air-soil interface is usually the case that the Variable boundary condition would apply. For any Variable boundary node, only one of Eqs. (2.8a) through (2.8e) is used at any time. During a precipitation period, the throughfall water can either completely or partially penetrate into the subsurface system, depending on the soil condition. For the complete penetration, a Cauchy-type boundary condition (Eq. (2.8b)) is used, whereas a Dirichlet-type boundary condition (Eq. (2.8a)) is employed for the partial penetration because either a pond or a surface runoff is built. While during a non-precipitation period, three situations are possible: a ponding condition may still exist in the early stage, followed by soil evaporation for a period of time, and finally a dry condition can be reached if the period is long enough. Eqs. (2.8c) through (2.8e) are employed to simulate these three situations in order.

2.2 The Governing Equations of Chemical Transport

Two most frequently mentioned nomenclatures in chemical transport modeling are components and species. Components are defined as a set of linearly independent "basis" chemical entities such that every species can be uniquely represented as a combination of those components; no component can be represented by other components than itself. A species is the product of a chemical reaction involving components as reactants [Yeh and Tripathi, 1990, 1991]. In 3DHYDROGEOCHEM, two basic types of components are included: aqueous (mobile) and adsorbent (immobile) components. For the transport of aqueous components, the major processes are advection, dispersion/ diffusion, decay, sources/sinks, and chemical reactions. For the transport of adsorbent components, only decay and chemical reactions need to be considered. The governing equations for chemical transport are derived based on the principle of conservation of mass (for both types of components) and flux laws (for aqueous components only). They can be found elsewhere [Cheng, 1995].

Before the governing equations are described, the following nomenclature used in the model needs to be introduced:

$$\overline{T}_j = c_j + \sum_{i=1}^{M_x} a_{ij}^x x_i + \sum_{i=1}^{M_y} a_{ij}^y y_i + \sum_{i=1}^{M_z} a_{ij}^z z_i + \sum_{i=ndps+1}^{M_p} a_{ij}^p p_i$$

$$T_j = c_j + \sum_{i=1}^{M_x} a_{ij}^x x_i + \sum_{i=1}^{M_y} a_{ij}^y y_i + \sum_{i=1}^{M_z} a_{ij}^z z_i + \sum_{i=1}^{M_p} a_{ij}^p p_i$$

$$C_j = c_j + \sum_{i=1}^{M_x} a_{ij}^x x_i$$

$$s_j = \sum_{i=1}^{M_y} a_{ij}^y y_i + \sum_{i=1}^{M_z} a_{ij}^z z_i$$

$$P_j = \sum_{i=1}^{M_p} a_{ij}^p p_i$$

$$\overline{P}_j = \sum_{i=ndps+1}^{M_p} a_{ij}^p p_i$$

where T_j and $\overline{T}_j$ are the total analytical and modified total analytical concentrations of the *j*-th component, respectively; C_j, S_j, P_j, and $\overline{P}_j$ are the total dissolved, total sorbed, total precipitated, and modified total precipitated concentrations of the *j*-th component, respectively; c_j is the concentration of the *j*-th component; x_i and a_{ij}^x are the i-th complexed species and its associated stoichiometric coefficient with respect to the *j*-th component, respectively; y_i and a_{ij}^y are the i-th adsorbed species and its associated stoichiometric coefficient with respect to the *j*-th component, respectively; z_i and a_{ij}^z are the *i*-th ion-exchanged species and its associated stoichiometric coefficient with respect to the *j*-th component, respectively; p_i and a_{ij}^p are the *i*-th precipitated species and its associated stoichiometric coefficient with respect to the *j*-th component, respectively; M_x, M_y, M_z, and M_p are the numbers of the complexed, adsorbed, ion-exchanged, and precipitated species, respectively; ndps is the number of the dominating precipitated species [Cheng and Yeh, 1994a, 1997a] which are precipitated species with a concentration much higher than the total dissolved concentrations of their associated aqueous components.

The aqueous components are mobile in subsurface systems, whereas the adsorbent components are immobile. The aqueous components may be involved in the complexation, adsorption/desorption, ion-exchange, and precipitation/ dissolution reactions, whereas the adsorbent components can only be involved in the adsorption/desorption reaction. From another viewpoint, the aqueous components exist at least in the aqueous phase, whereas the adsorbent components exist only in the solid phase. The complexed, adsorbed, ion-exchanged, and precipitated species are the products of the complexation, adsorption, ion-exchange, and precipitation reactions, respectively. The dominating precipitated species are arranged as the first ndps precipitated species so that they can be easily indexed in both derivation and formulation. To compute chemical transport, three types of governing equations are basically solved: the

governing equations of aqueous components, of adsorbent components, and of ion-exchange sites. Two approaches, defined as Approach 1 and Approach 2 in this report, are considered to deal with the chemical transport of aqueous components.

For aqueous components:

<Approach 1>

$$\theta\frac{\partial C_j}{\partial t} + \mathbf{V}\cdot\nabla C_j + \theta\frac{\partial(S_j+\overline{P})}{\partial t} + \left(\theta\lambda_j + \frac{\partial\theta}{\partial t} + \frac{\rho^*}{\rho}q\right)\overline{T}_j - \nabla\cdot\left(\theta\mathbf{D}\cdot\nabla\overline{T}_j\right) - \frac{\rho_o}{\rho}\overline{T}_j\nabla\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V}$$

$$= -\nabla\cdot\left[\theta\mathbf{D}\cdot\nabla(S_j+\overline{P}_j)\right] + qC_j^* + \frac{\partial\theta}{\partial t}C_j + \frac{\rho^*}{\rho}q(S_j+\overline{P}_j) - \frac{\rho_o}{\rho}(S_j+\overline{P}_j)\nabla\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V}$$

$$-\sum_{i=1}^{ndps}a_{ij}^p\left[\frac{\partial(\theta p_i)}{\partial t}+\theta\iota_i^p\right] \qquad\qquad j\in\left[1,N_a\right]$$

(2.9)

<Approach 2>

$$\theta\frac{\partial\overline{T}_j}{\partial t} + V\cdot\Lambda_{jj}\nabla\overline{T}_j + \left(\theta\lambda_j^a + \frac{\partial\theta}{\partial t} + \frac{\rho^*}{\rho}q\right)\overline{T}_j - \nabla\cdot\left(\theta\mathbf{D}\cdot\Lambda_{jj}\nabla\overline{T}_j\right) - \frac{\rho_o}{\rho}\overline{T}_j\nabla\left(\frac{\rho}{\rho_o}\right)\cdot V$$

$$= qC_j^* + \frac{\rho^*}{\rho}q(S_j+\overline{P}_j) - \sum_{i=1,i\neq j}^{NON}\mathbf{V}\cdot\Lambda_{j,i}\nabla\overline{T}_i + \sum_{i=1,i\neq j}^{NON}\nabla\cdot\left(\theta\mathbf{D}\cdot\Lambda_{j,i}\nabla\overline{T}_i\right)$$

$$- \sum_{i=1}^{NSITE}\mathbf{V}\cdot\Lambda_{j,NON+i}\nabla EC_i + \sum_{i=1}^{NSITE}\nabla\cdot\left(\theta\mathbf{D}\cdot\Lambda_{j,NON+i}\nabla EC_i\right) - \frac{\rho_o}{\rho}(S_j+\overline{P}_j)\nabla\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V}$$

$$+ \frac{\partial\theta}{\partial t}C_j - \sum_{i=1}^{ndps}a_{ij}^p\left[\frac{\partial(\theta p_i)}{\partial t}+\theta\iota_i^p\right] \qquad\qquad j\in\left[1,N_a\right]$$

(2.10)

if

$$\left|\frac{\partial C_j}{\partial\overline{T}_j}\right| > \sum_{i=1,i\neq j}^{NON}\left|\frac{\partial C_j}{\partial\overline{T}_i}\right| + \sum_{i=1}^{NSITE}\left|\frac{\partial C_j}{\partial EC_i}\right|$$

(2.11)

For adsorbent components:

$$\theta\frac{\partial T_j}{\partial t} + \frac{\partial\theta}{\partial t}T_j = -\theta\lambda_j T_j \qquad\qquad j\in\left[N_a+1,NON\right]$$

(2.12)

For ion-exchange sites:

$$\theta\frac{\partial EC_i}{\partial t} + \frac{\partial\theta}{\partial t}EC_i = -\theta\lambda_i^{EC}EC_i \qquad i\in\left[1,NSITE\right]$$

(2.13)

where θ is the water content; ι_i^p is the first-order decay rate of the i-th precipitated species; EC_i is the ion-exchange capacity of the i-th ion-exchange site; t is time; $\mathbf{V}$ is the Darcy flux; ∇ is the del operator; $\mathbf{D}$ is the dispersion coefficient tensor; λ_j and λ_i^{EC} are the decay constants of the j-th component and the i-th ion-exchange site, respectively; $\partial(\ )/\partial\tau$ is the partial derivative with respect to time; C_j^* is the total dissolved concentration of the j-th component in the source; N_a is the number of aqueous components; NON and NSITE are the numbers of components and ion-exchange sites, respectively; and $\Lambda_{j,j}$, $\Lambda_{j,i}$, and $\Lambda_{j,NON+i}$ are the concentration derivatives of $\partial C_j/\partial\overline{T}_j, \partial C_j/\partial\overline{T}_i (1\leq i\leq NON)$,and

14

$\partial C_j / \partial EC_i \, (1 \le i \le NSITE)$, respectively.

The computation of concentration derivatives can be found elsewhere [Cheng, 1995]. The dispersion coefficient tensor **D** in Eq. (2.8) is given by

$$\theta \mathbf{D} = \alpha_T |\mathbf{V}| \delta + (\alpha_L - \alpha_T) \mathbf{V}\mathbf{V}/|\mathbf{V}| + \theta \alpha_m \tau \delta \qquad (2.14)$$

where $|\mathbf{V}|$ is the magnitude of $\mathbf{V}$; δ is the Kronecker delta tensor; α_T is transverse dispersivity; α_L is the longitudinal dispersivity; α_m is the molecular diffusion coefficient; and τ is the tortuosity.

The first approach for aqueous components uses the pore velocities to both track particles in the Lagrangian step and compose matrix equations in the Eulerian step, whereas the second approach employs the retarded pore velocities [Yeh and Cheng, 1996; Cheng and Yeh, 1993a, 1993b] to perform particle tracking as well as matrix composition. The derivation and interpretation of both approaches can be found elsewhere [Cheng, 1995]. In both Eqs. (2.9) and (2.10), the dominating precipitated species is taken into account. If there is no dominating precipitated species existing in the system, the modified total precipitated concentration will be equal to the total precipitated concentration. In addition, the term showing the rate change of dominating precipitated species on the right-hand side of both equations no longer appears. In this case, the total analytical concentration, instead of the modified total analytical concentration, is the variable of interest to be solved for. The reason why the dominating precipitated species needs to be taken into account is to be given later in Chapter 3. The whole set of governing equations of chemical transport needs to be solved simultaneously because they are not a set of independent transport equations. In the model, all the chemical species in a system are related to one another under the consideration of chemical equilibrium. To solve for the concentration distributions of chemical species, the governing equations of chemical transport are coupled with the governing equations of chemical equilibrium. They are solved iteratively until a convergent solution is reached. The detailed approach is stated in Chapter 3. The processes of constructing and solving the governing equations of chemical equilibrium can be found elsewhere [Cheng, 1995].

2.2.1 Initial Conditions for the Governing Equations of Chemical Transport

The initial conditions used in transient-state simulations for chemical transport can be written as

$$\begin{aligned}
\overline{T}_j &= \overline{T}_j^{\text{initial}}(x,y,z) &&\text{in R,} &&j \in [1, N_a] \\
p_i &= p_i^{\text{initial}}(x,y,z) &&\text{in R,} &&i \in [1, \text{ndps}] \\
T_j &= T_j^{\text{initial}}(x,y,z) &&\text{in R,} &&j \in [N_a + 1, \text{NON}] \\
EC_i &= EC_i^{\text{initial}}(x,y,z) &&\text{in R} &&i \in [1, \text{NSITE}]
\end{aligned} \qquad (2.15)$$

where R is the domain of interest; $\overline{T}_j^{\text{initial}}$ is the initial modified total analytical concentration of the j-th aqueous component; p_i^{initial} is the initial concentration of the i-th dominating precipitated species; T_j^{initial} and EC_i^{initial} are the initial total analytical concentration of the $(j-N_a)$-th adsorbent component and the initial capacity of the i-th ion-exchange site, respectively. They can be gained by either field measurements or by solving the steady-state version of Eqs. (2.9) to (2.13).

2.2.2 Boundary Conditions for the Governing Equations of Chemical Transport

Four types of boundary conditions are employed for the simulation of chemical transport. The boundary conditions are only required for aqueous (mobile) components because both adsorbent components and ion-exchange sites are considered immobile. Rather, the total analytical concentrations of adsorbent components and the ion-exchange capacities at a boundary node are computed in the same way as that of a node inside the domain.

The four types of boundary conditions for aqueous components are described as follows.

(1) Prescribed concentration (Dirichlet) boundary conditions:

$$\overline{T}_j = T_j^{\text{Dirichlet}}(x_b, y_b, z_b, t) \quad \text{on } B_D \qquad j \in [1, N_a] \qquad (2.16)$$

where (x_b, y_b, z_b) is the spatial coordinate on the boundary; $T_j^{\text{Dirichlet}}$ is the prescribed functional value of the modified total

analytical concentration of the j-th aqueous component on the Dirichlet boundary B_D. A Dirichlet boundary is a boundary where the modified total analytical concentration is specified. Usually, the modified total analytical concentrations of nonconservative mobile components cannot be known when the model is used for prediction. However, with the measured data available, the inclusion of the Dirichlet boundary condition will enhance the progress of model calibration and validation.

(2)   Variable boundary conditions:

$$\mathbf{n} \cdot \left(\mathbf{V}C_j - \theta \mathbf{D} \cdot \nabla C_j \right) = n \cdot \mathbf{V}C_j^{Variable} \left(x_b, y_b, z_b, t \right) \qquad \text{if } \mathbf{n} \cdot \mathbf{V} \leq 0 \text{ on } B_V \quad j \in \left[1, N_a \right] \qquad (2.17a)$$

$$\mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C_j \right) = 0 \qquad \text{if } \mathbf{n} \cdot \mathbf{V} > 0 \text{ on } B_V \quad j \in \left[1, N_a \right] \qquad (2.17b)$$

where n is the outward unit vector normal to the boundary; $C_j^{Variable}$ is the specified total dissolved concentration of the j-th aqueous component in the water passing through the Variable boundary B_V. A Variable boundary is a boundary where the flow is not predetermined. The flow needs to be computed in advance in the subsurface flow module. If the flow is inward on the boundary, the incoming flux can be specified and a flux-type boundary condition (Eq. (2.17a)) can be used. If the flow is outward, on the other hand, the concentration is considered brought out from the domain through advection and Eq. (2.17b) is employed.

(3)   Cauchy boundary conditions:

$$\mathbf{n} \cdot \left(\mathbf{V}C_j - \theta \mathbf{D} \cdot \nabla C_j \right) = q_j^{Cauchy} \left(x_b, y_b, z_b, t \right) \qquad \text{on } B_C \qquad j \in \left[1, N_a \right] \qquad (2.18)$$

where q_j^{Cauchy} is the prescribed total flux of the j-th aqueous component through the Cauchy boundary B_C. A Cauchy boundary is a boundary where the total flux, due to both advection and dispersion, is known. This boundary condition can be used to simulate many practical situations. For example, the leakage rate of a chemical from a disposal site can be considered a Cauchy flux.

(4)   Neumann boundary conditions:

$$\mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C_j \right) = q_j^{Neumann} \left(x_b, y_b, z_b, t \right) \qquad \text{on } B_N \qquad j \in \left[1, N_a \right] \qquad (2.19)$$

where $q_j^{Neumann}$ is the prescribed gradient flux of the j-th aqueous component through the Neumann boundary B_N. A Neumann boundary is a boundary where the dispersive flux is described. For instance, a zero dispersive flux is employed to simulate a boundary located far away from the disturbence.

2.3   The Governing Equation of Heat Transfer

The governing equation for heat transfer in a subsurface system can be derived based on the principles of conservation of energy and the law of thermal flux. Yeh and Luxmoore derived the governing equations describing the moisture and temperature fields in an unsaturated aquifer system in 1983 [Yeh and Luxmoore, 1983]. Based on their derivation, the following governing equation for heat transfer can be obtained by further assuming (1) the thermal flux due to moisture gradient is small compared to that due to temperature and (2) the medium is rigid.

$$\left(\rho C_1 \theta + \rho_b C_m \right) \frac{\partial Temp}{\partial t} + C_1 \rho \mathbf{V} \cdot \nabla Temp - \nabla \cdot \left(\lambda \mathbf{I} \cdot \nabla Temp \right) = S - \frac{\rho^*}{\rho} q C_1 \rho Temp \qquad (2.20)$$

where Temp represents temperature; ρ_b is the bulk density of the dry medium; C_l and C_m are the specific heats of groundwater and the dry medium in subsurface systems, respectively; λ is the apparent thermal conductivity; and S is the heat source. In using the results of solving Eq. (2.20), the temperatures in both the solid and the liquid phases are thought as the same if they are both associated with the same representative elementary volume (REV) of a point of interest. It is noted that the heat source (i.e., S) may include the latent heat due to evaporation or condensation and heat exchange due to chemical reaction, though they are not taken into account in this version. Since we consider single-phase flow in this version, the model cannot solve problems that involve two- or multi-phase flow. Thus, the applicable temperature range is restricted accordingly.

2.3.1 Initial Conditions for the Governing Equation of Heat Transfer

The initial conditions for transient-state simulations of heat transfer can be written as

$$Temp = Temp^{initial}(x,y,z) \qquad in \ R \tag{2.21}$$

where $Temp^{initial}$ represents the initial temperature distribution over the entire domain of interest, R. $Temp^{initial}$ can be obtained by either experimental measurements or solving the steady-state version of Eq. (2.20).

2.3.2 Boundary Conditions for the Governing Equation of Heat Transfer

Four types of boundary conditions are included to simulate a variety of situations in practice:

(1) Prescribed temperature (Dirichlet) boundary conditions:

$$Temp = Temp^{Dirichletl}(x_b,y_b,z_b,t) \qquad\qquad on \ B_D \tag{2.22}$$

where (x_b,y_b,z_b) is the spatial coordinate on the boundary; $Temp^{Dirichlet}$ is the prescribed temperature on the Dirichlet boundary B_D. This boundary condition is used when the boundary temperature can be described.

(2) Neumann boundary conditions:

$$-\mathbf{n} \cdot \lambda \mathbf{I} \cdot \nabla Temp = q^{Neumannl}(x_b,y_b,z_b,t) \ on \ B_N \tag{2.23}$$

where $\mathbf{I}$ is an identity tensor of rank two; $\mathbf{n}$ is the outward unit vector normal to the boundary; $q^{Neumann}$ are the prescribed gradient thermal flux through the Neumann boundary B_N. This boundary is employed when the thermal flux due to conduction is described. For example, a heat-resistant boundary can be simulated by applying a zero Neumann flux to the boundary.

(3) Cauchy boundary conditions:

$$-\mathbf{n} \cdot (\lambda \mathbf{I} \cdot \nabla Temp - \rho C_1 \mathbf{V} Temp) = q^{Cauchy}(x_b,y_b,z_b,t) \quad on \ B_C \tag{2.24}$$

where q^{Cauchy} is the prescribed total thermal flux through the Cauchy boundary B_C. This boundary condition can be used when the thermal flux due to both conduction and convection is given.

(4) Variable boundary conditions:

$$-\mathbf{n} \cdot (\lambda \mathbf{I} \cdot \nabla Temp - \rho C_1 \mathbf{V} Temp) = \rho C_1 \mathbf{V} Temp^{Variable}(x_b,y_b,z_b,t) \qquad if \ \mathbf{n} \cdot \mathbf{V} \le 0 \ on \ B_V \tag{2.25a}$$

$$\mathbf{n} \cdot \nabla Temp = 0 \qquad\qquad if \ \mathbf{n} \cdot \mathbf{V} > 0 \quad on \ B_V \tag{2.25b}$$

where $Temp^{Variable}$ is the specified temperature associated with the water passing through the Variable boundary B_V. This boundary condition is commonly used when the flow is not predetermined, rather, it is computed for the subsurface flow module. Eq. (2.25a) is considered when the flow is inward, and Eq. (2.25b) is for the outgoing flow.

2.4 The Governing Equations of Chemical Equilibrium

In solving the governing equations of aqueous components, C_j, S_j, and P_j are estimated through the computation of chemical equilibrium in 3DHYDROGEOCHEM. The governing equations to determine chemical equilibrium are stated as follows.

Mole balance equations for aqueous components:

$$T_j = c_j + \sum_{i=1}^{M_x} a_{ij}^x x_i + \sum_{i=1}^{M_y} a_{ij}^y y_i + \sum_{i=1}^{M_z} a_{ij}^z z_i + \sum_{i=1}^{M_p} a_{ij}^p p_i \qquad j \in [1,N_a] \tag{2.26}$$

Mole balance equations for adsorbent components:

$$T_j = c_j + \sum_{i=1}^{M_y} a_{ij}^y y_i \qquad j \in [N_a + 1, N_a + N_s] \tag{2.27}$$

Mole balance equations for ion-exchange sites:

$$EC_i = \sum_{k=NOMZJ(i)+1}^{NOMZJ(i)+NOMZI(i)} \nu_k z_k \qquad i \in [1,NSITE] \tag{2.28}$$

Mass action equation for aqueous complexation:

$$A_i^x = K_i^x \prod_{j=1}^{N_a} X_j^{a_{ij}^x} \qquad i \in [1,M_x] \tag{2.29}$$

Mass action equation for adsorption/desorption:

$$B_i^y = K_i^y \left(\prod_{j=1}^{N_a} X_j^{a_{ij}^y} \right) \left(\prod_{j=N_a+1}^{N_a+N_s} X_j^{a_{ij}^y} \right) \quad i \in \left[1, M_y \right] \tag{2.30}$$

Mass action equation for ion-exchange reactions:

$$K_{k,LNI(i)} = \left(\frac{B_k}{A_k} \right)^{\nu_{LNI(i)}} \left(\frac{A_{LNI(i)}}{B_{LNI(i)}} \right)^{\nu_k} \tag{2.31}$$

$$k \in \left[NOMZJ(i) + 1, \ NOMZJ(i) + NOMZI(i) \right] \quad k \neq LNI(i) \qquad i \in \left[1, \ NSITE \right]$$

Mass action equation for precipitation/dissolution

$$1 = K_i^p \prod_{j=1}^{N_a} X_j^{a_{ij}^p} \qquad i \in \left[1, \ M_p \right] \tag{2.32}$$

where N_s is the number of adsorbent components; ν_k is the valence of the k-th ion-exchanged species; NOMZI(i) is the number of ion-exchanged species involved in the i-th ion-exchange site; NOMZJ(i) is the number of ion-exchanged species involved in the 1-st through the (i-1)-th ion-exchange sites; A_i^x is the activity of the i-th complexed species; K_i^x is the equilibrium constant of the i-th complexed species; X_j is the activity of the j-th component species; B_i^y is the activity of the i-th adsorbed species of surface complexation; K_i^y is the equilibrium constant of the i-th adsorbed species; LNI(i) is the number of the ion-exchanged species, which is taken as the reference species for the i-th ion-exchange site, on the ion-exchanged species list; $K_{k,LNI(i)}$ is equilibrium constant of the k-th ion-exchanged species; K_i^p is the equilibrium constant of the i-th precipitated species.

Given T_j ($j \in [1, N_a]$), T_j ($j \in [N_a + 1, N_a + N_s]$), and EC_i, Eqs. (2.26) through (2.28) constitute ($N_a + N_s + NSITE$) equations in ($N_a + N_s + M_x + M_y + M_z + M_p$) unknowns: c_j's ($j \in [1, N_a]$), c_j's ($j \in [N_a + 1, N_a + N_s]$), x_i's, y_i's, z_i's, and p_i's, while Eqs. (2.29) through (2.32) provide the other ($M_x + M_y + M_z + M_p$ - NSITE) equations to complete the formulation. when redox reactions are involved, electron is considered to be an aqueous component while its component concentration is zero since there is no free electron. In a system involving acid-base reactions, the operational proton is taken as an aqueous component for describing the acidity of the system. In 3DHYDROGEOCHEM, Davies equation is employed to compute activity coefficients of aqueous species, while the coefficients of adsorbed species are set to 1. The activity of precipitated species is assumed to be 1, while the activity of ion-exchanged components are computed by dividing the molar concentration of the ion-exchanged species by the total concentration of all ion-exchanged species at the ion-exchange site that the ion-exchanged species dwells. Adsorption can be described with a simple surface complexation model when the effect of electrostatic force is not included. A simple surface complexation model can numerically be treated in the same manner as the aqueous complexation model and no special consideration in the numerical approach is needed. When the effect is considered in modeling adsorption, either the constant capacitance model or the triple layer mode can be used. If a constant capacitance model is used, one additional unknown (c_o) is needed for each adsorbent component (or adsorbing site). If a triple layer model is employed, two additional unknowns (c_o and c_b) are introduced for each adsorbent component (or adsorbing site). These two additional unknowns are defined as

$$c_o = \exp\left(-\frac{e\psi_o}{kT} \right) \tag{2.33}$$

$$c_b = \exp\left(-\frac{e\psi_b}{kT} \right) \tag{2.34}$$

where k is Boltzman constant; T is absolute temperature; e is electronic charge; ψ_o is the electric potential at the surface; and ψ_b is the electric potential at the beta layer. All details to compute chemical equilibriums can be found elsewhere [Cheng, 1995].

When the dominating precipitated species exists in the system, we introduce the modified total analytical concentration, rather than the total analytical concentration, of components to be the primary dependent variable for the system of chemical transport. Detailed descriptions to compute chemical equilibrium with the dominating precipitated species involved can be found elsewhere [Cheng, 1995].

2.5 Solving the Governing Equations of Subsurface Flow, Chemical Transport and Heat Transfer

The subsurface flow equation, Eq. (2.1), subject to initial and boundary conditions, Eqs. (2.5) through (2.8), is solved with the Galerkin finite element method. The chemical transport equations of aqueous components, Eqs. (2.9) and (2.10), subject to initial and boundary conditions, Eqs. (2.15) through (2.19), are solved with either the conventional Eulerian or the hybrid Lagrangian-Eulerian finite element method. The chemical transport equations of both adsorbent components and ion-exchanged sites, Eqs. (2.11) and (2.12), subject to the initial condition, Eq. (2.15), are solved with analytical solutions. The heat transfer equation, Eq. (2.20), subject to initial and boundary conditions, Eqs. (2.21) through (2.25), is solved with either the conventional Eulerian or the hybrid Lagrangian-Eulerian finite element method. The detailed implementation of the numerical approximation of subsurface flow, chemical transport, and heat transfer can be found elsewhere [Cheng, 1995].

2.6 The Coupling of Subsurface Flow, Chemical Transport, and Heat Transfer

As mentioned in Chapter 1, either strong or weak coupling can be applied to deal with a system of subsurface flow, chemical transport, and heat transfer. In the 3DHYDROGEOCHEM model, the following points are always taken into account regardless of whether strong or weak coupling is used.

(1) Both the groundwater density and dynamic viscosity are concentration- and temperature-dependent. The density of groundwater is a function of chemical concentration and can be derived to the equation as follows [Cheng, 1995].

$$\rho = \rho_w + \sum_{i=1}^{N_{aq}} c_i W_i \left(1 - \frac{\rho_w}{\rho_i} \right) \tag{2.35a}$$

where N_{aq} is the number of species existing in the aqueous phase; and c_i, W_i, and ρ_i are the concentration, molecular weight, and intrinsic density of the i-th dissolved species, respectively. In order to make the computation easier, it is assumed that the total dissolved concentration of a component represents the contribution to the groundwater density from all the dissolved species with that component involved. With the assumption verified, it is allowed to use the following equation to compute the groundwater density.

$$\rho = \rho_w + \sum_{j=1}^{N_a} C_j W_j \left(1 - \frac{\rho_w}{\rho_j} \right) \tag{2.35b}$$

where NON is the number of components; and C_j, W_j, and ρ_j are the total dissolved concentration, molecular weight, and intrinsic density of the j-th aqueous component, respectively. If all the intrinsic densities of aqueous components are much larger than that of pure water, Eq. (2.35b) can be further simplified to the following equation.

$$\rho = \rho_w + \sum_{j=1}^{N_a} C_j W_j \tag{2.35c}$$

The dynamic viscosity of groundwater is assumed to take the following form which is similar to Eq. (2.35c).

$$\mu = \mu_W + \sum_{i=1}^{N_a} a_i C_i W_i \tag{2.35d}$$

where a_i is the i-th weighting parameter representing the dependence of dynamic viscosity on the total dissolved concentration of the i-th component.

It should be noted that Eq. (2.35b) is a simplification of Eq. (2.35a) which can be derived theoretically, whereas Eq. (2.35d) is an assumed equation to describe the reality. Thus, Eq. (2.35d) can be modified with other desired empirical formulae to match the situation being considered. This modification in the computer code is straightforward. In 3DHYDROGEOCHEM, Eq. (2.35c) is used to described the groundwater density due to the dissolved concentration. For some cases, however, the associated assumptions may not be true. Moreover, the intrinsic densities of dissolved species or aqueous components might not be available. To overcome this, one can use an empirical formula to characterize the groundwater density.

When the temperature effect is taken into account, then both the density and dynamic viscosity of groundwater should be further described as functions of temperature. Under the assumption that the

intrinsic densities of aqueous components are not affected by temperature, the groundwater density can be calculated by

$$\rho = \rho_w(\text{Temp}) + \sum_{j=1}^{N_a} C_j W_j \tag{2.35e}$$

where $\rho_w(\text{Temp})$ is the temperature-dependent density of pure water. Likewise, the dynamic viscosity of groundwater can be computed with

$$\mu = \mu_W(\text{Temp}) + \sum_{i=1}^{N_a} a_i C_i W_i \tag{2.35g}$$

where $m_w(\text{Temp})$ is the temperature-dependent dynamic viscosity of pure water. Both $r_w(\text{Temp})$ and $m_w(\text{Temp})$ can be determined empirically.

(2)   The equilibrium constants of chemical reactions are temperature-dependent:
Based on the well-known van't Hoff relationship, the temperature-dependent equilibrium constant, K, can be computed with the following equation:

$$\ln \frac{K}{K_o} = \frac{\Delta H^o}{R} \left[\frac{1}{T_o} - \frac{1}{T} \right]$$

$$\Rightarrow K = K_o \exp \left[\frac{\Delta H^o}{R} \left(\frac{T - T_o}{T T_o} \right) \right] \tag{2.36}$$

where K and K_o are the temperature-dependent and referenced equilibrium constants, respectively; H^o, assumed unchanged with temperature, is the enthalpy of the associated chemical reaction; T is the temperature of interest; T_o is the referenced temperature associated with K_o; R is the ideal gas constant; ln is the natural logarithm operator; and exp represents the exponential operator. In Eq. (2.36), $K_o, \Delta H^o$, R, and T_o are all input by users. Any consistent unit system can be used to perform the computation. Although van't Hoff relationship is theoretically sound, it is not applicable for many species mainly due to the limitation that their standard state enthalpies are not available. In the computer code, therefore, appropriate experimental data-based empirical formulae can be employed to describe K. This needs effort to modify Subroutine TEMPEQ in the computer code.

(3)   The governing equation of subsurface flow is implicitly applied to the governing equations of both chemical transport and heat transfer.
Recalling the governing equation of subsurface flow, Eq. (2.1)

$$\frac{\rho}{\rho_o} \frac{d\theta}{dh} \frac{\partial h}{\partial t} = \nabla \cdot \left[\mathbf{K} \cdot \left(\nabla h + \frac{\rho}{\rho_o} \nabla z \right) \right] + \frac{\rho^*}{\rho_o} q \tag{2.1}$$

and the equation of Darcy flux, Eq. (2.3)

$$\mathbf{V} = -\mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) \tag{2.3}$$

Substituting Eq. (2.3) into Eq. (2.1) yields

$$\frac{\rho}{\rho_o} \frac{\partial \theta}{\partial t} = -\frac{\rho}{\rho_o} \nabla \cdot \mathbf{V} - \nabla \left(\frac{\rho}{\rho_o} \right) \cdot \mathbf{V} + \frac{\rho^*}{\rho_o} q \tag{2.37}$$

Eq. (2.37) can be multiplied by the total dissolved concentration of the j-th component (i.e., C_j) to yield Eq. (2.38) after rearrangement.

$$\frac{\rho}{\rho_o} C_j \nabla \cdot \mathbf{V} = -\frac{\rho}{\rho_o} C_j \frac{\partial \theta}{\partial t} - C_j \nabla \left(\frac{\rho}{\rho_o} \right) \cdot \mathbf{V} + \frac{\rho^*}{\rho_o} q C_j \tag{2.38}$$

It can also be multiplied by the product of the temperature, the referenced groundwater density, and the specific heat of groundwater (i.e., Temp $_oC_l$) to yield Eq. (2.39) after the rearrangement as that for Eq. (2.38).

$$\rho C_l \text{Temp} \nabla \cdot \mathbf{V} = -\rho C_l \text{Temp} \frac{\partial \theta}{\partial t} - \frac{\rho}{\rho_o} \rho C_l \text{Temp} \nabla \left(\frac{\rho}{\rho_o} \right) \cdot \mathbf{V} + \frac{\rho^*}{\rho_o} \rho C_l q \text{Temp} \tag{2.39}$$

Eq. (2.29) is used to cancel the term, $C_i \cdot \mathbf{V}$, which results from the process of advection of chemical transport, in the original governing equations of chemical transport to yield Eqs. (2.9) and (2.10). Eq. (2.30) is used to substitute the term of $C_i \text{Temp} \cdot \mathbf{V}$, which results from the process of convection of heat transfer, in the original governing equations of heat transfer to yield Eq. (2.20). The detailed processes of cancellation/substitution for both chemical transport and heat transfer can be found elsewhere [Cheng, 1995].

2.6.1 Strong Coupling

Conceptually, strong coupling should always be used to solve for a system of subsurface flow, chemical transport, and heat transfer. It is so mainly because (1) the groundwater density and dynamic viscosity are functions of dissolved concentrations of chemical species and temperature, (2) the equilibrium constants of chemical reactions are temperature-dependent, and (3) the subsurface flow velocity affects the temporal-spatial distributions of chemical concentrations and temperature. Therefore, all the governing equations should in general be solved simultaneously. In 3DHYDROGEOCHEM, strong coupling is applied for steady-state simulations. A coupling loop is employed to interrelate subsurface flow, chemical transport, and heat transfer. During an iteration, the governing equations are solved in the order of subsurface flow, heat transfer, and chemical transport. As already demonstrated in Figure 1.1, during every iteration of the coupling loop, the new iteration values of the variables of interest are computed based on the working iteration values of the variables of interest which are determined by relaxing the previous two iteration values. The new iteration values are compared with the previous iteration values to determine if a convergent solution is obtained. If a convergent solution is obtained, then the computation is finished and the new iteration values become the solutions. If convergency is not met, then the working iteration values are updated for use in the next iteration. As shown in Figure 1.1, the working iteration values are updated after all the governing equations are solved and the nonconvergency is detected. However, the model treats a coupling iteration with the following more detailed steps:

Step 1. Compute the new value for subsurface flow. All the working iteration values are computed in the previous iteration.

Step 2. Examine the convergency of subsurface flow by comparing the newly-computed value with the previous iteration value of subsurface flow.

Step 3. Compute the new value for heat transfer. The newly-computed value of subsurface flow is used in this computation.

Step 4. Examine the convergency of heat transfer by comparing the newly-computed value with the previous iteration value of heat transfer .

Step 5. Compute the new value for chemical transport. The newly-computed values of both subsurface flow and heat transfer in the current iteration are employed in this computation.

Step 6. Examine the convergency of chemical transport by comparing the newly-computed and the previous iteration value of chemical transport.

Step 7. Determine if the next iteration is needed. If convergency has been reached for all the three modules, then the computation is finished. Otherwise, the working iteration values are updated by relaxing the newly-computed and the previous iteration values. The coupling iteration continues after the updating.

Figure 2.1 shows the complete structure of the strong coupling loop, with the above seven steps executed, in the model. The structure is so designed for enhancing the convergency associated with the coupling loop.

2.6.2 Weak Coupling

As mentioned above, strong coupling is used to solve all the governing equations simultaneously. From the viewpoint of obtaining accurate solutions, it is required for steady-state simulations because the initial guess may be far away from the solution. However, it is time-consuming from a computational point of view. There is only one coupling loop needed for steady-state simulations, but the coupling loop is needed for every time step if it is applied to transient-state simulations. As one can imagine, strong coupling on transient-state simulations will require large computer resources. Therefore, it was determined that a more efficient approach to transient-state simulations should be attempted. The answer to this issue in 3DHYDROGEOCHEM is weak coupling. For weak coupling, it is still taken into account that groundwater density and dynamic viscosity are functions of dissolved concentrations of chemical species as well as temperature, the equilibrium constants of chemical reactions are temperature-dependent, and the subsurface flow velocity affects the significance of the advection term of chemical transport and the convection term of heat transfer. Nevertheless, no coupling loop is utilized, reducing greatly the computational time. Although only weak coupling is used

in the current version of 3DHYDROGEOCHEM for transient-state simulations, strong coupling can also be employed with a coupling iteration loop implemented for each time step.

Figure 2.2 shows the structure of weak coupling used in the model. In both Figure 2.1 and Figure 2.2, H, C, and T represent the computed variables associated with subsurface flow, chemical transport, and heat transfer, respectively. Instead of the working iteration values of the variables of interest used in the strong coupling (i.e., HW, CW, and TW in Figure 2.1), the working values of the variables of interest are used to compute the new values of the variables of interest in weak coupling (i.e., HW, CW, and TW in Figure 2.2). The weak coupling in Figure 2.2 mainly includes the following four steps for every time step:

Step 1. Solve the governing equation of subsurface flow. All the working values are computed in the previous time step.

Step 2. Solve the governing equation of heat transfer based on the newly-computed value of subsurface flow.

Step 3. Solve the governing equations of chemical transport. The newly-computed values of subsurface flow and heat transfer in the current time step are used in this computation.

Step 4. Prepare for the computation of the next time step.

The order of solving the governing equations, subsurface flow, heat transfer, and then chemical transport, is so designed because of the following reasons:
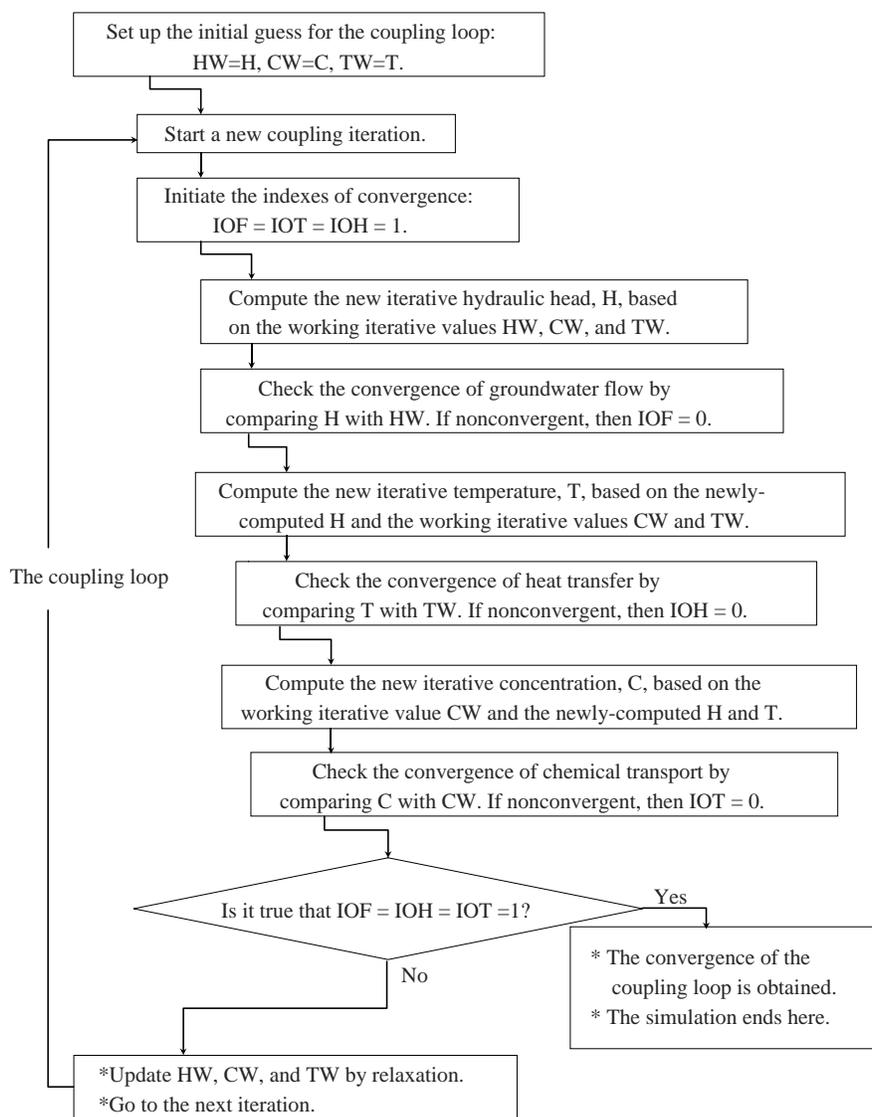
Figure 2.1 Strong coupling in 3DHYDROGEOCHEM.

(1)   The temperature effect on chemical reactions is assumed important. In other words, the simulation of heat transfer should be achieved before the simulation of chemical transport, so that the updated temperature information is used for chemical reactions.

(2)   The significance of the advection term on the results of chemical transport is assumed greater than that of both the groundwater density and dynamic viscosity on the results of subsurface flow when the concentration effect is considered. As mentioned earlier, the dissolved concentrations of chemical species affect both the groundwater density and dynamic viscosity, and consequently influence the results of subsurface flow. This is how chemical transport would impact subsurface flow. On the other hand, the velocity field directly influences the magnitude of the advection term. As a result, it affects chemical transport. Since the advective transport is usually more important than density effect, subsurface flow should be simulated prior to chemical transport.

(3)   The importance of the convection term on the results of heat transfer is assumed greater than that of both groundwater density and dynamic viscosity on the results of subsurface flow when the temperature effect is taken into account. Both groundwater density and dynamic viscosity are functions of temperature. Therefore, temperature would have an effect on subsurface flow. Meanwhile, the velocity field describes the consequence of the convection term, and is of importance for the results of heat transfer. Because the heat convection is usually more important than density effect, subsurface flow should be computed prior to heat transfer.

From the viewpoint of obtaining accurate solutions, the above three arguments provide the logic why the computation order of first subsurface flow, then heat transfer, and then chemical transport is used.
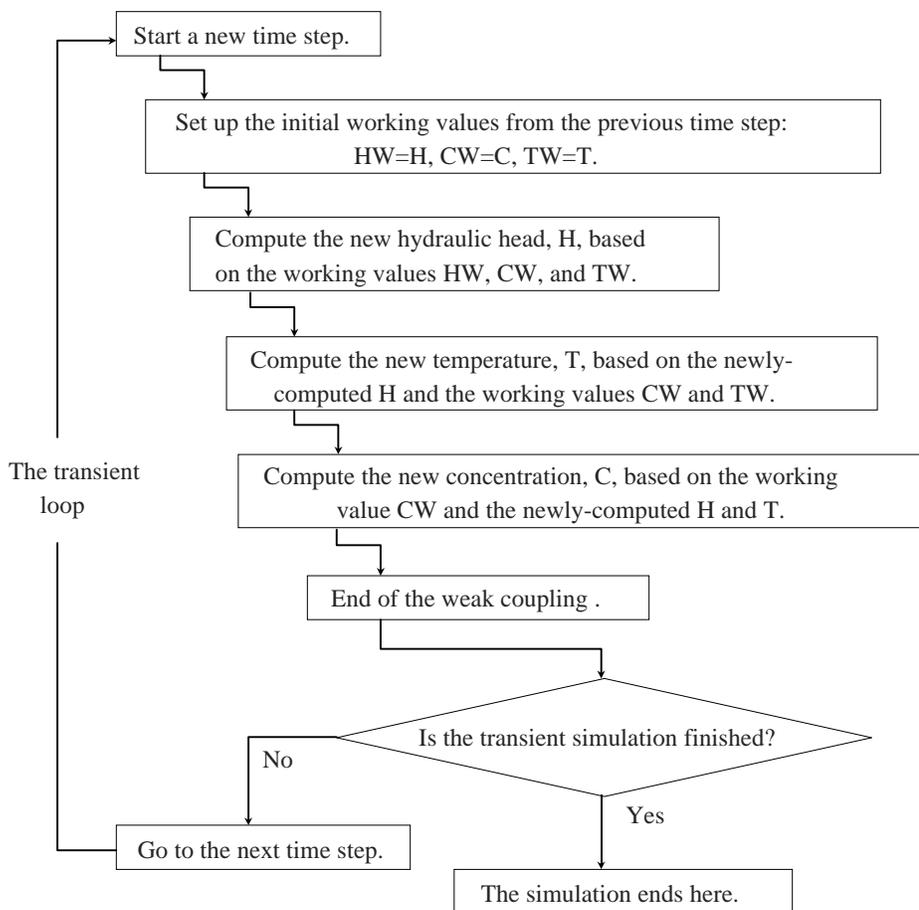
Figure 2.2 Weak coupling in 3DHYDROGEOCHEM.

Chapter 3
NUMERICAL STRATEGIES

This chapter describes the numerical strategies involved in 3DHYDROGEOCHEM, which include (1) the strategy for steady-state simulations, (2) the strategy for transient-state simulations, (3) the strategy for solving the governing equation of subsurface flow, (4) the strategy for solving the governing equations of chemical transport, and (5) the strategy for solving the governing equation of heat transfer. In solving the governing equations of chemical transport, the two approaches (Approaches 1 and 2 as mentioned earlier in Chapter 2) employed in the model to solve for the transport of aqueous components are emphasized. The associated numerical strategies, under the considerations of peak/valley capturing as well as the existence of dominating precipitated species are also described in the computation dealing with the transport of aqueous components. In order to reduce the difficulties in programming, compiling, and executing, the two approaches used for chemical transport are implemented in separate computer codes.

3.1   The Strategy for Steady-State Simulations

As mentioned earlier, the strong coupling of subsurface flow, chemical transport, and heat transfer is considered for steady-state simulations in 3DHYDROGEOCHEM. Figure 3.1 shows how strong coupling is implemented in the associated computer codes. During a coupling iteration, the steady-state version of subsurface flow governing equation is first solved based on the working values of system variables (e.g., pressure head, temperature and chemical concentrations) from the previous iteration. The pre-initial condition has to be provided as the values of the zeroth iteration (or the initial guess) to begin the steady-state simulation. The distribution of pressure head in the domain of interest is then obtained. The distributions of moisture content, water capacity, and groundwater velocity can be further determined based on the currently-computed pressure head. The computed pressure head is compared with that from the previous iteration for examining convergency. All the variables computed in the subsurface flow module will be applied to computing heat transfer and chemical transport later on in the current iteration (i.e., the subsurface flow variables are updated). The steady-state version of heat transfer governing equation is solved next to obtain the distribution of temperature in the domain of interest. Likewise, the convergency of the heat transfer module is examined by comparing the currently-computed and previous iteration value of temperature, and the currently-computed temperature is to be used in modifying the equilibrium constants for the computation of chemical transport coming up in the current iteration (i.e., temperature is updated). Lastly, the steady-state governing equations of chemical transport are solved for the distribution of total analytical component concentrations in the domain of interest. The currently-computed and previous iteration values of the total analytical component concentrations are compared for checking convergency.

At the end of every iteration, the overall convergency is checked. If all the three modules are convergent for the current iteration, then the convergent solution represents a steady-state solution. Otherwise, the next coupling iteration is required. In this case, all the working values of system variables are updated by relaxing the currently-computed values and the previous working values (i.e., from the previous iteration). The updated working values will be applied to the computation of the next iteration. After the updating is finished, the updated working total analytical component concentrations are employed to compute chemical equilibrium, so that the total dissolved component concentrations can be obtained to calculate the estimated groundwater density for the next iteration. A steady-state simulation is finished when a convergent solution is reached.

3.2   The Strategy for Transient-State Simulations

The weak coupling of subsurface flow, chemical transport, and heat transfer is used for transient-state simulations in 3DHYDROGEOCHEM. Figure 3.2 shows how weak coupling is implemented in the associated computer codes. In the simulation of a time step, the transient-state version of subsurface flow governing equation is first solved for the distribution of pressure head. The distributions of moisture content, water capacity, and groundwater velocity are then determined based on the computed pressure heads. The updated moisture content, rate change of moisture content, and groundwater velocity are to be directly involved in computing chemical transport and heat transfer later on in the current time step. Secondly, the transient-state version of heat transfer governing equation is solved to obtain the
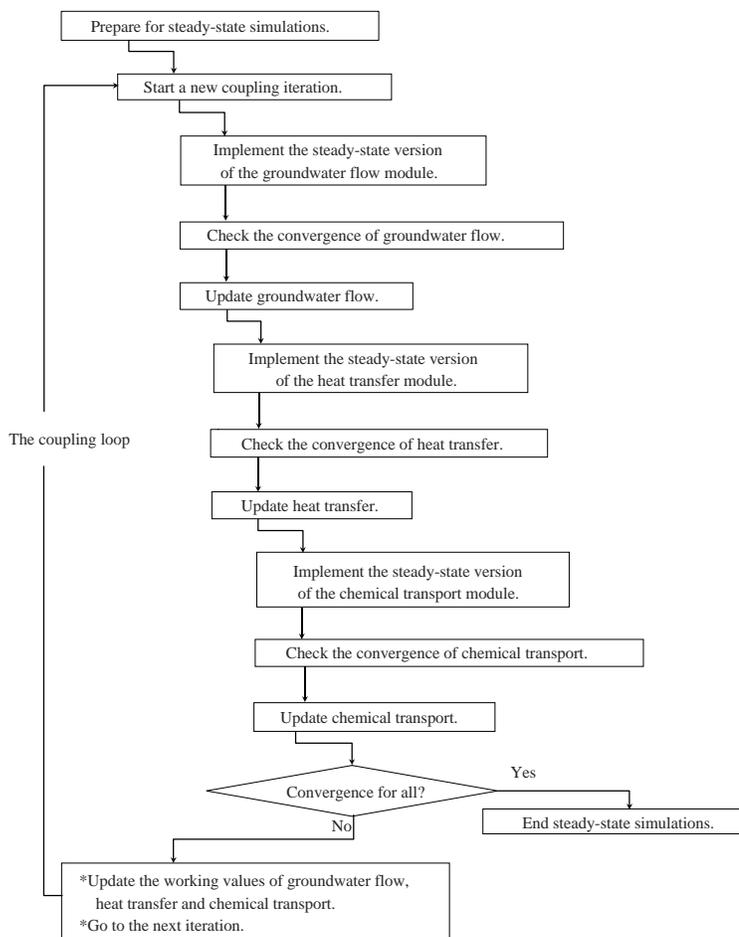
Figure 3.1 The numerical strategy for steady-state simulations.

distribution of temperature for the current time step. The updated temperature is utilized to modify the equilibrium constants of chemical reactions involved in computing chemical transport for the current time step and groundwater density and viscosity involved in computing subsurface flow for the next time step.

Finally, the transient-state governing equations of chemical transport are solved for the distribution of total analytical component concentrations. These computed values are then used to calculate the total dissolved component concentrations under the consideration of chemical equilibrium. The total dissolved component concentrations are to be used to reflect the density effect involved in the computation of the next time step. Instead of checking the overall convergency, checking the completion of simulations is located at the end of a time step in a transient simulation. A transient-state simulation is stopped when either the final simulation time or the given simulation time-step number is reached. In all the three modules, the updated values themselves represent the transient-state solution for the current time step.

3.3 The Strategy for Solving the Governing Equation of Subsurface Flow

In 3DHYDROGEOCHEM, the subsurface flow module is basically the modified version of the 3DFEMWATER model where neither the density nor the temperature effect is taken into account. In addition to the Dirichlet, Neumann, and Cauchy boundary conditions, 3DFEMWAT includes the Variable boundary condition which treats the flow-in/flow-out boundary appropriately by its nature. The determination of employing a suitable Variable boundary condition is described in Chapter 1. Since the Variable boundary condition is not determined a priori, it needs to be correctly addressed by using the technique of iteration. Only when the implemented Variable boundary conditions for two consecutive iterations are exactly the same, is it that Variable boundary condition and its associated convergent solution represent the situation being considered. Figure 3.3 shows the flow chart of how the subsurface flow equation is solved numerically in 3DHYDROGEOCHEM.
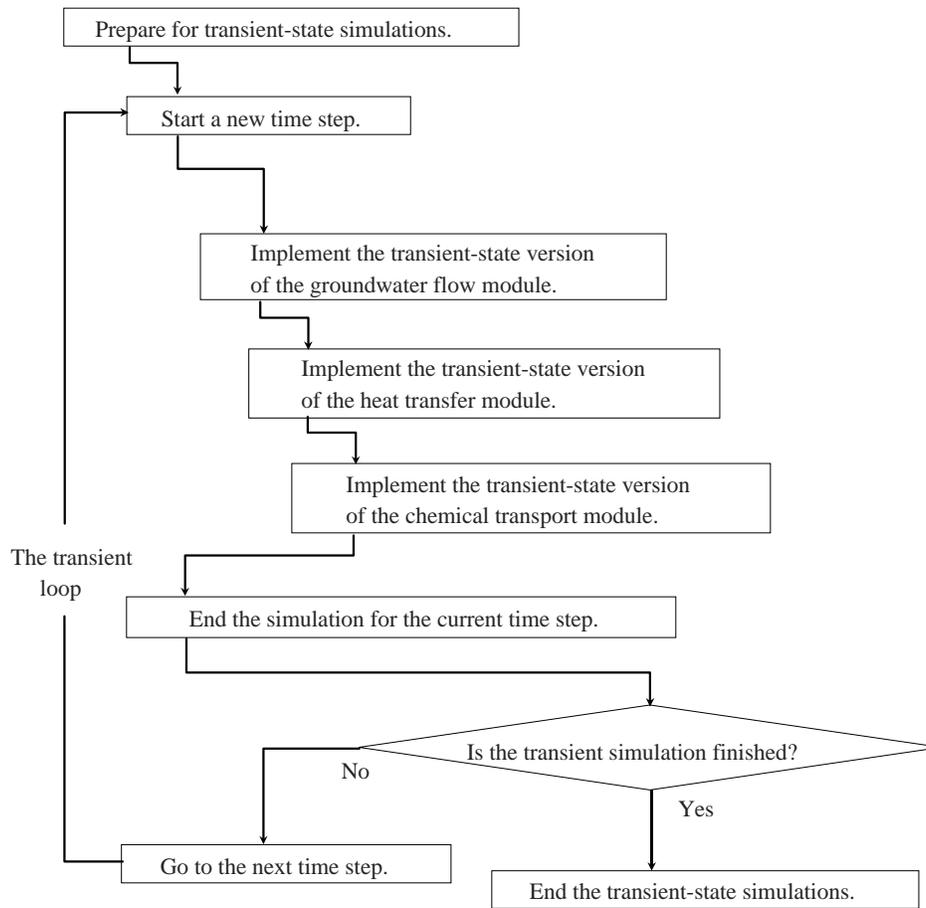
Figure 3.2 The numerical strategy for transient-state simulations.

In Figure 3.3, two iteration loops are included: one is to address the Variable boundary condition, the other one is to deal with the nonlinearity of unsaturated media. During a Variable- boundary-condition iteration, the Variable boundary condition for the flow-in/flow-out boundary is predetermined based on either the given initial guess or the updated subsurface flow condition from the previous iteration. Thus, the boundary condition is completely specified and a nonlinear partial differential equation (PDE) is ready to be solved. A nonlinear iteration loop is utilized to handle this nonlinear PDE. In a nonlinear iteration, the working soil properties are first calculated based on the working pressure head which is predetermined either from initial conditions or from the previous nonlinear iteration. With the working soil properties, the linearized matrix equation can be composed and solved. The nonlinear convergency is examined by comparing the currently-computed pressure head with the working pressure head at all global nodes. If the maximum head difference is greater than the given error tolerance, then the working pressure head is updated by the relaxation technique and another nonlinear iteration is needed. Otherwise, the nonlinear convergency is reached and the consistency of the Variable boundary condition is checked. If the currently-used Variable boundary condition is the same as that of the previous Variable-boundary-condition iteration, then the Variable-boundary-condition consistency is reached and the convergent solution from the nonlinear iteration loop is the numerical solution for the time being. Otherwise, the Variable boundary condition needs to be updated for the use in the next Variable-boundary-condition iteration. Both iterations continue until consistent/convergent results are obtained.

3.4 The Strategy for Solving the Governing Equations of Chemical Transport

In general, the governing equations of chemical transport are a set of nonlinear partial differential equations. To efficiently deal with this nonlinear system, four types of components are considered in 3DHYDROGEOCHEM: (1) adsorbent (immobile) components, (2) conservative aqueous (mobile) components, (3) nonconservative aqueous (mobile) components, and (4) fixed-concentration components. Adsorbent components, considered immobile, are the
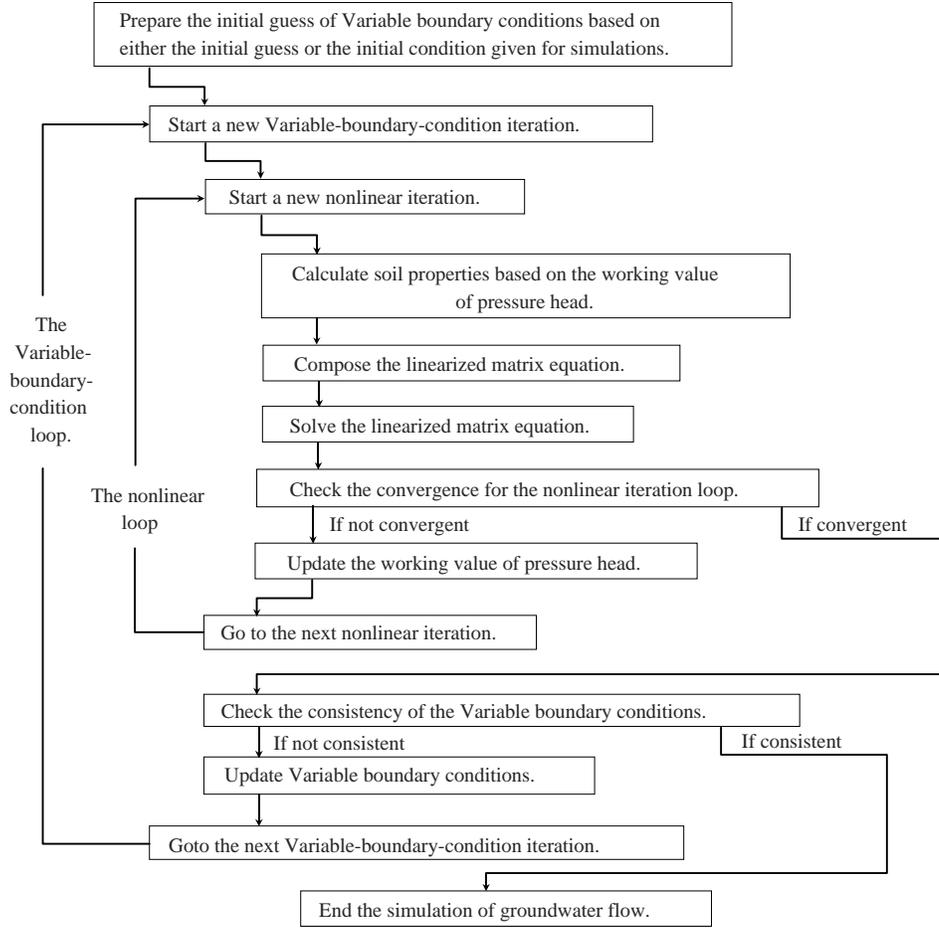
Figure 3.3 The numerical strategy for solving the governing equation of subsurface flow.

components involved in the species of the solid phase only. Conservative aqueous components, considered mobile, are the components involved in the species of the aqueous phase only. Nonconservative aqueous components, also mobile, are the components involved in both the aqueous and solid phases. Fixed-concentration components, which may be either mobile or immobile, are the components with fixed concentrations of their component species. To reduce the difficulty in numerically solving the governing equations, the total analytical component concentrations are chosen as the primary dependent variables (PDV) of chemical transport systems. The reason for this has been discussed by Yeh and Tripathi [Yeh and Tripathi, 1990; 1991]. The total analytical concentrations of components are to be computed by directly solving their associated governing equations, which are called primary governing equations (PGEs). Furthermore, all the ion-exchange sites have to be identified so that the concentration distributions of all components at all nodes can be computed under the consideration of chemical equilibrium.

As shown in Eq. (B.1.31), the PGEs of adsorbent components are written as

$$\theta \frac{\partial T_j}{\partial t} + \left(\theta \lambda_j + \frac{\partial \theta}{\partial t} \right) T_j = 0 \qquad\qquad j \in \left[N_a + 1, \, N_a + N_s \right] \qquad\qquad (3.1)$$

where λ_j is the 1-st order decay constant of the $(j-N_a)$-th adsorbent component; θ is moisture content; N_a and N_s are the numbers of aqueous components and adsorbent components, respectively; and T_j is the total analytical concentration of the $(j-N_a)$-th adsorbent component.

The PGEs of ion-exchange sites, as shown in Eq. (B.1.38), are written as

$$\theta \frac{\partial EC_j}{\partial t} + \left(\theta \lambda_j^{EC} + \frac{\partial \theta}{\partial t} \right) EC_j = 0 \qquad\qquad j \in \left[1, \, NSITE \right] \qquad\qquad (3.2)$$

27

where λ^{EC} is the 1-st order decay constant of the j-th ion-exchange site; NSITE is the number of ion-exchange sites; and EC_j is the ion-exchange capacity of the j-th site. Based on Eqs. (B.1.15) through (B.1.25), the PGEs for conservative and nonconservative aqueous components can be written as follows:

For conservative aqueous components:

$$\theta\frac{\partial T}{\partial t} + \mathbf{V}\Delta T_j + \left(\theta\lambda_j + \frac{\rho^*}{\rho}q\right)T_j - \nabla\cdot\left(\theta\mathbf{D}\cdot\nabla T_j\right) - \frac{\rho_o}{\rho}T_j\nabla\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V} = qT_j \qquad j\in\left[1, N_a\right] \qquad (3.3)$$

For nonconservative aqueous components:

$$\theta\frac{\partial C_j}{\partial t} + \mathbf{V}\cdot\nabla C_j + \theta\frac{\partial\left(S_j + P_j\right)}{\partial t} + \theta\lambda_j T_j + \frac{\partial\theta}{\partial t}\left(S_j + P_j\right) + \frac{\rho^*}{\rho}qC_j$$

$$-\nabla\cdot\left(\theta\mathbf{D}\cdot\nabla C_j\right) - \frac{\rho_o}{\rho}C_j\nabla\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V} = qC_j^* \qquad j\in\left[1, N\right]$$

(3.4)

where T_j, C_j, S_j, and P_j are the total analytical, the total dissolved, the total sorbed, and total precipitated concentrations of the j-th aqueous component, respectively; $\mathbf{V}$ is Darcy flux; λ_φ is the 1-st order decay constant of the j-th aqueous component; $\mathbf{D}$ is dispersion coefficient tensor; ρ, ρ_o, and ρ^* are the real, the referenced, and the source densities of groundwater, respectively; and N_a is the number of aqueous components.

For fixed-concentration components, the following equation exists.

$$c_j = c_j^{fixed}(x, y, z) \qquad (3.5)$$

where c_j is the concentration of the j-th component species and $c_j^{fixed}(x,y,z)$ is the given fixed concentration for the j-th component species.

From Eqs. (3.1) through (3.5), the following things can be observed: (1) under given initial conditions, both the total analytical concentrations of adsorbent components and ion-exchange capacities can be computed analytically, (2) under given appropriate boundary and/or initial conditions, the total analytical concentrations of conservative aqueous components can be solved for independently, (3) under given appropriate boundary and/or initial conditions, all the total analytical concentrations of nonconservative aqueous components have to be solved for simultaneously, and (4) the component species concentrations need only be specified for fixed-concentration components. Based on these observations, Figure 3.4 shows the strategy used in 3DHYDROGEOCHEM to approach a nonlinear chemical transport system. This strategy actually includes a very important concept, which is, the better the initial guess of a nonlinear system, the easier a convergent solution of the system can be obtained.

As shown in Figure 3.4, the total analytical concentrations of adsorbent components and ion-exchange capacities are computed first; the component species concentration of fixed-concentration components are specified secondly; then the total analytical concentrations of conservative aqueous components are individually solved for thirdly. With what has been computed, the best initial guess for solving the nonlinear governing equations associated with nonconservative aqueous components can be made by computing chemical equilibrium. Finally, the total analytical concentrations of nonconservative aqueous components are solved for simultaneously by using the iteration technique. It should be noted that there is no need to compute either the total analytical concentrations of adsorbent components or ion-exchange capacities in steady-state simulations because they have to be the same as specified in the pre-initial condition if they do exist.

3.4.1   The Strategy for Solving the Transport Equations of Aqueous Components

In 3DHYDROGEOCHEM, two approaches are included to solve the transport equations of aqueous components: one uses the pore velocity and dispersion coefficient to handle advection and dispersion, respectively, for aqueous components (i.e., Approach 1) and the other one employs both the retarded pore velocity and the retarded dispersion coefficient to deal with advection and dispersion, respectively, for aqueous components (i.e., Approach 2). The advection velocity in Approach 1 is just the pore velocity for all aqueous components, whereas it is component-dependent in Approach 2.
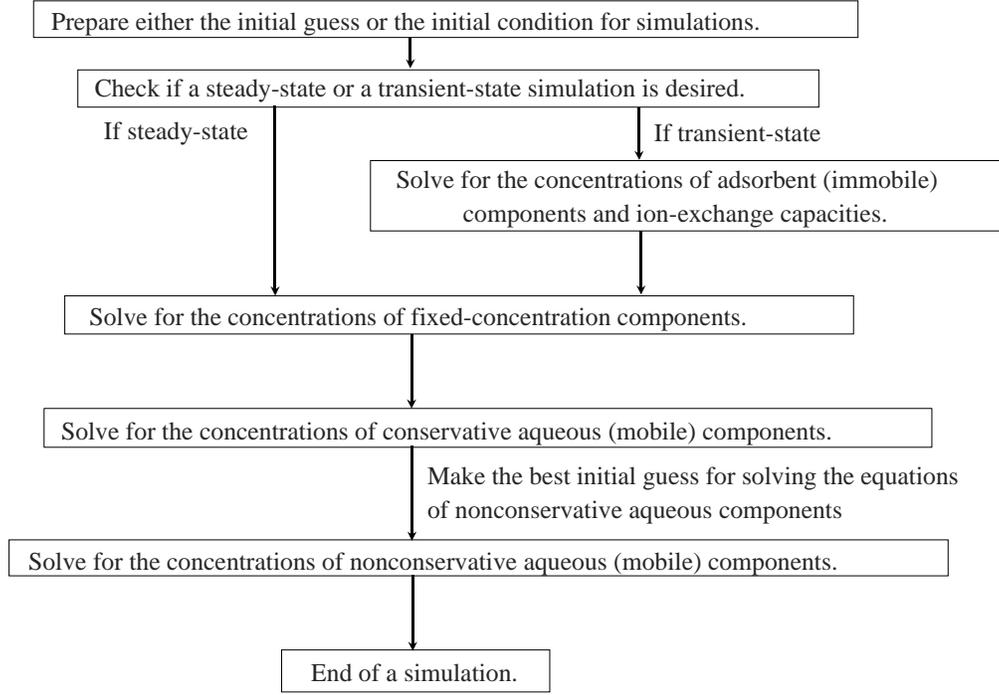
Figure 3.4 The numerical strategy for solving the governing equations of chemical transport.

In Approach 1, the transport equations for aqueous components are written as follows.

For conservative aqueous components:

$$\theta \frac{\partial C_j}{\partial t} + \mathbf{V} \cdot \nabla C_j + \left(\theta \lambda_j + \frac{\rho^*}{\rho} q \right) C_j - \nabla \cdot \left(\theta \mathbf{D} \cdot \nabla C_j \right) - \frac{\rho_o}{\rho} C_j \nabla \left(\frac{\rho}{\rho_o} \right) \cdot \mathbf{V} = qC_j^* \qquad j \in [1, N_a] \tag{3.6}$$

For nonconservative aqueous components:

$$\theta \frac{\partial C_j}{\partial t} + \mathbf{V} \cdot \nabla C_j + \theta \frac{\partial (S_j + P_j)}{\partial t} + \left(\theta \lambda_j + \frac{\partial \theta}{\partial t} + \frac{\rho^*}{\rho} q \right) T_j - \nabla \cdot \left(\theta \mathbf{D} \cdot \nabla T_j \right) - \frac{\rho_o}{\rho} T_j \nabla \left(\frac{\rho}{\rho_o} \right) \cdot \mathbf{V}$$

$$= -\nabla \cdot \left[\theta \mathbf{D} \cdot \nabla (S_j + P_j) \right] + qC_j^* + \frac{\partial \theta}{\partial t} C_j + \frac{\rho^*}{\rho} q (S_j + P_j) - \frac{\rho_o}{\rho} (S_j + P_j) \nabla \left(\frac{\rho}{\rho_o} \right) \cdot \mathbf{V} \qquad j \in [1, N_a] \tag{3.7}$$

In Approach 2, the transport equations for aqueous components are written as follows.

For conservative aqueous components:

$$\theta \frac{\partial T_j}{\partial t} + \mathbf{V} \cdot \Lambda_{j,j} \nabla T_j - \nabla \cdot \left(\theta \mathbf{D} \cdot \Lambda_{j,j} \nabla T_j \right) + \left(\theta \lambda_j^a + \frac{\rho^*}{\rho} q \right) C_j - \frac{\rho_o}{\rho} C_j \nabla \left(\frac{\rho}{\rho_o} \right) \cdot \mathbf{V} = qC_j^* \qquad j \in [1, N_a] \tag{3.8}$$

29

For nonconservative aqueous components:

$$\theta\frac{\partial T_j}{\partial t} + \sum_{i=1}^{NON}\mathbf{V}\cdot\Lambda_{j,i}\nabla T_i - \sum_{i=1}^{NON}\nabla\cdot\left(\theta\mathbf{D}\cdot\Lambda_{j,i}\nabla T_i\right) + \sum_{k=1}^{NSITE}\mathbf{V}\cdot\Lambda_{j,NON+k}\nabla EC_i$$

$$-\sum_{k=1}^{NSITE}\nabla\cdot\left(\theta\mathbf{D}\cdot\Lambda_{j,NON+k}\nabla EC_i\right) + \left(\theta\lambda_j + \frac{\partial\theta}{\partial t} + \frac{\rho^*}{\rho}q\right)T_j - \frac{\rho_o}{\rho}T_j\nabla\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V}$$

$$= qC_j^* + \frac{\rho^*}{\rho}q\left(S_j + P_j\right) - \frac{\rho_o}{\rho}\left(S_j + P_j\right)\nabla\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V} + \frac{\partial\theta}{\partial t}C_j \tag{3.9}$$

$$j\in\left[1, N_a\right] \qquad i\in\left[1, NON\right] \qquad k\in\left[1, NSITE\right]$$

where NON is the number of components; $\Lambda_{j,i}$ and $\Lambda_{j,NON+k}$ represent the concentration derivatives of the j-th aqueous component associated with the i-th component and the k-th ion-exchange site, respectively. They are defined as follows.

$$\Lambda_{j,i} = \frac{\partial C_j}{\partial T_i} \qquad \text{and} \qquad \Lambda_{i,NON+k} = \frac{\partial C_j}{\partial EC_k} \tag{3.10}$$

The detailed computational process can be found elsewhere [Cheng, 1995]. Eqs. (3.6) and (3.8) are exactly the

same because the total dissolved concentrations are equal to the total analytical concentrations and $\Lambda_{j,j} = \frac{\partial C_j}{\partial T_i} = 1$ for

conservative aqueous components. Therefore, there is actually no difference between the two approaches for computing the concentrations of conservative aqueous components. However, since Eq. (3.7) is so much different from Eq. (3.9), a significant difference in solving for the concentrations of nonconservative aqueous components between the two approaches can be expected. Figure 3.5 describes how Approaches 1 and 2 handle solving the governing equations of aqueous components.

In both approaches, the conventional Eulerian finite element method is used for steady-state simulations while either the hybrid Lagrangian-Eulerian or the conventional Eulerian finite element method can be employed for transient-state simulations. When the conventional Eulerian finite element method is used, either the Galerkin or the upstream finite element method can be applied for dealing with the advection term. When the hybrid Lagrangian-Eulerian approach is employed for transient-state simulations, Approach 1 applies the pore velocity as the particle tracking velocity to advance the total dissolved concentrations of all aqueous components in the Lagrangian step, whereas Approach 2 uses the retarded pore velocity, which combines the pore velocity with the concentration derivatives, to advance the total analytical concentration of each aqueous component in the Lagrangian step. Suppose NRIT represents the iteration number needed to reach convergency for the coupling loop of solute transport and chemical equilibrium (i.e., the nonlinear loop of chemical transport). Thus, only 1 Lagrangian step is needed for all aqueous components in Approach 1. But (1 + NRIT × N_1) Lagrangian steps, including 1 for all conservative aqueous components and (NRIT × N_1) for N_1 nonconservative aqueous components with NRIT nonlinear iterations required for convergency, are needed in Approach 2. In addition, only the Galerkin finite element method is considered for those advection terms appearing on the right-hand side of the transport equations of nonconservative aqueous components.

As mentioned earlier, the transport equations for conservative aqueous components are solved prior to those of nonconservative aqueous components because the former ones are linear and the latter ones are nonlinear. From Figures 3.5, it can be counted that 1 Lagrangian and (1 + NRIT) Eulerian steps are needed for Approach 1, whereas (1 + NRIT × N_1) Lagrangian and (1 + NRIT) Eulerian steps are needed for Approach 2. Within a nonlinear iteration (i.e., a coupling iteration of solute transport and chemical equilibrium), only the Eulerian step is included for Approach 1 but both the Lagrangian and the Eulerian steps are included for Approach 2. This is because in a nonlinear iteration, chemical equilibrium is updated in Approach 1, whereas concentration derivatives as well as chemical equilibrium are updated in Approach 2. The updating in Approach 1 can only affect the Eulerian step but the updating in Approach 2 influences both the Lagrangian and Eulerian steps. Figures 3.6 and 3.7 show how the nonlinear iteration loops work in Approaches 1 and 2, respectively.

In Approach 1, chemical equilibrium is computed to make the best initial guess of working dependent variables for solving the nonlinear transport equations of nonconservative aqueous components. Based on the working dependent variables, the linearized matrix equations associated with nonconservative aqueous components can be composed and

(a)

Start solving for aqueous (mobile) components in Approach 1.

If the conventional Eulerian finite element method is used

If the hybrid Lagrangian-Eulerian finite element method is used

Implement the Lagrangian step for both conservative and nonconservative aqueous (mobile) components.

Implement the Eulerian step for conservative aqueous (mobile) components.

The coupling loop of solute transport and chemical equilibrium

Implement the Eulerian step for nonconservative aqueous (mobile) components.

End solving for aqueous (mobile) components in Approach 1.

(b)

Start solving for aqueous (mobile) components in Approach 2.

If the conventional Eulerian finite element method is used

If the hybrid Lagrangian-Eulerian finite element method is used

Implement the Lagrangian step for conservative aqueous (mobile) components.

Implement the Eulerian step for conservative aqueous (mobile) components.

If the conventional Eulerian finite element method is used

If the hybrid Lagrangian-Eulerian finite element method is used

Implement the Lagrangian step for nonconservative aqueous (mobile) components.

The coupling loop of solute transport and chemical equilibrium

Implement the Eulerian step for nonconservative aqueous (mobile) components.

End solving for aqueous (mobile) components in Approach 2.

Figure 3.5 The numerical strategies for solving the transport equation of aqueous components in (a) Approach 1 and (b) Approach 2.

then be solved with a desired solver. The convergency is examined by comparing the currently-computed total analytical component concentrations with the working total analytical component concentrations. If the relative error is smaller than the given error tolerance, then a convergent solution of the nonlinear iteration is reached for the time being. Otherwise, the working dependent variables need to be updated for use in the next nonlinear iteration.

In Approach 2, not only chemical equilibrium but also concentration derivatives need to be computed to perform the initial guess for solving the nonlinear system. As mentioned above, the Lagrangian steps associated with nonconservative aqueous components must be included in the nonlinear iteration loop in this approach. Similar to Approach 1, the convergency is checked on a relative-error basis. However, if a convergent solution is not reached in the current iteration, both the working dependent variables and the working concentration derivatives need to be updated for the next iteration.

3.4.2 The Strategy of Using the Peak/Valley Capturing Technique in Solving the Transport Equations of Aqueous Components

It is well known that numerical dispersion may significantly reduce the computational accuracy, especially in solving an advection-dominant transport equation. In order to solve transport equations accurately, one can use the Lagrangian-Eulerian method as well as the adaptive local grid refinement technique to eliminate most numerical dispersion. When an appropriate numerical scheme (e.g., 2DLEZOOMPC [Cheng et al., 1996b]) is used, a linear

< Define >

T's = the total analytical concentrations of aqueous components;
C's = the total dissolved concentrations of aqueous components;
S's = the total sorbed concentrations of aqueous components;
P's = the total precipitated concentrations of aqueous components;
W's = the total analytical concentrations of adsorbent components;
EC's = the capacities of ion-exchange sites.

Prepare for computing the nonconservaive T's in Approach 1:
  Make the initial guess of the working nonconservative C's, S's, and P's by computing
  chemical equilibrium based on (1) the already computed conservative T's, (2)the already
  computed W's and EC's, (3) the concentrations of those fixed-concentration components,
  and (4) the updated working nonconservative T's.

Start an new iteration of the coupling between chemical equilibrium
and solute transport to solve for the nonconservative T's.

Compute the nonconservative T's by solving the linearized matrix equations
associated with nonconservative aqueous components.

Examine if the computed nonconservative T's are convergent.
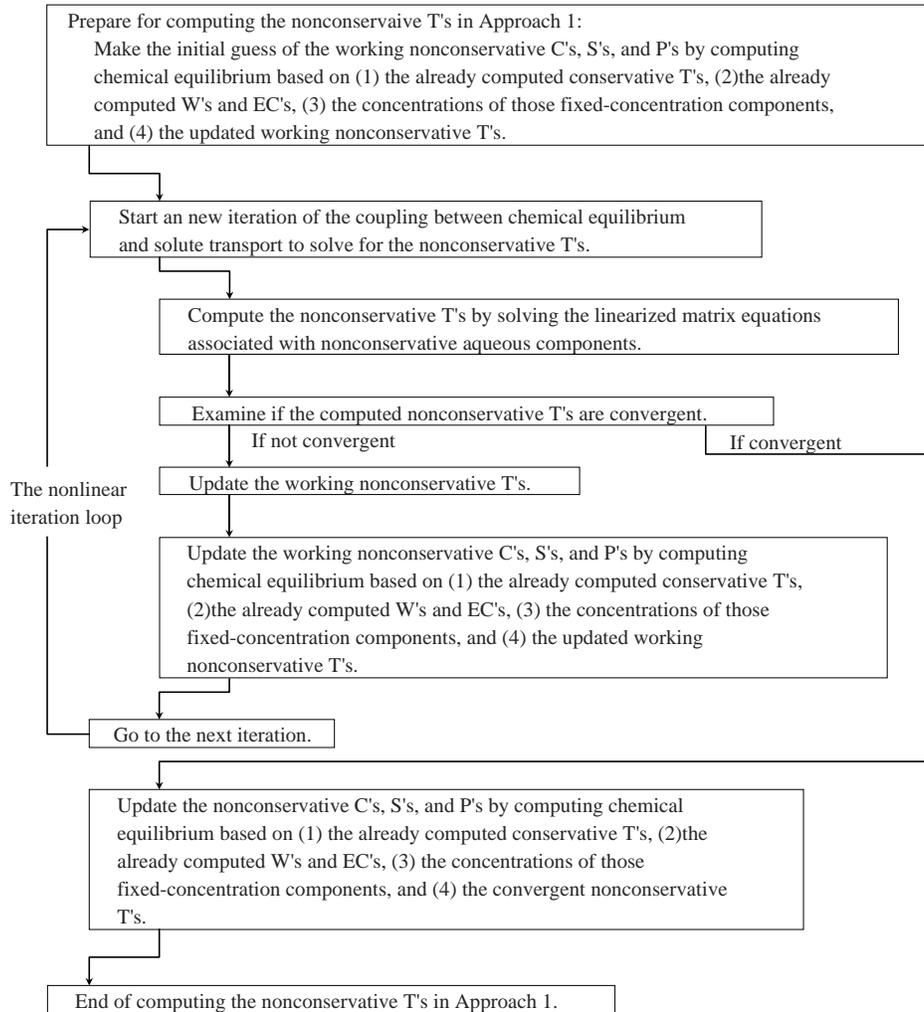
If not convergent   If convergent

Update the working nonconservative T's.

The nonlinear
iteration loop

Update the working nonconservative C's, S's, and P's by computing
chemical equilibrium based on (1) the already computed conservative T's,
(2)the already computed W's and EC's, (3) the concentrations of those
fixed-concentration components, and (4) the updated working
nonconservative T's.

Go to the next iteration.

Update the nonconservative C's, S's, and P's by computing chemical
equilibrium based on (1) the already computed conservative T's, (2)the
already computed W's and EC's, (3) the concentrations of those
fixed-concentration components, and (4) the convergent nonconservative
T's.

End of computing the nonconservative T's in Approach 1.

Figure 3.6 The numerical strategy of the nonlinear iteration loop in Approach 1.

transport equation can be solved accurately within the given error tolerance. However, it is time-consuming if such a numerical scheme is applied to solving the governing equations of chemical transport in a three-dimensional domain. To compensate for this, the peak/valley capturing technique accompanying the Lagrangian-Eulerian method is employed in the model. In this technique, the peak/valley of every aqueous component detected in every rough element is recorded for use in the following time step. With this technique, the numerical dispersion due to peak clipping/valley elevating can be reduced while at the same time the simulation will not require too much more extra computational time and memory. Figures 3.8 and 3.9 show how the technique is implemented in both approaches. Since the pore velocity is used to provide the velocity field for particle tracking in Approach 1, the backward particle tracking of global nodes and the forward particle tracking of both peak/valley points and global nodes need to be implemented only one time. However, these particle tracking processes need to be executed $(1 + NRIT \times N_1)$ times in Approach 2 because there is 1 pore velocity for conservative aqueous components and $(NRIT \times N_1)$ component-dependent modified pore velocities for nonconservative aqueous components.

32

Prepare for computing the nonconservative T's in Approach 2:
1. Make the initial guess of the working nonconservative C's, S's, and P's by computing chemical equilibrium based on (1) the already computed conservative T's, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, and (4) the updated working nonconservative T's.
2. Computing the initial working concentration derivatives associated with the nonconservative aqueous components based on the updated working values of species concentrations.

Start a new iteration of the coupling between chemical equilibrium and solute transport to solve for the nonconservative T's.

Compute the nonconservative T's by solving the linearized matrix equations associated with nonconservative aqueous components.

Examine if the computed nonconservative T's are convergent.

If not convergent   If convergent

Update the working nonconservative T's.

The nonlinear iteration loop

Update the working nonconservative C's, S's, and P's by computing chemical equilibrium based on (1) the already computed conservative T's, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, and (4) the updated working nonconservative T's.

Update the working concentration derivatives associated with the nonconservative aqueous components based on the updated working species concentrations.

Go to the next iteration.

Update the nonconservative C's, S's, and P's by computing chemical equilibrium based on (1) the already computed conservative T's, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, and (4) the convergent nonconservative T's.

Update the concentration derivatives associated with the nonconservative aqueous components based on the convergent species concentrations.

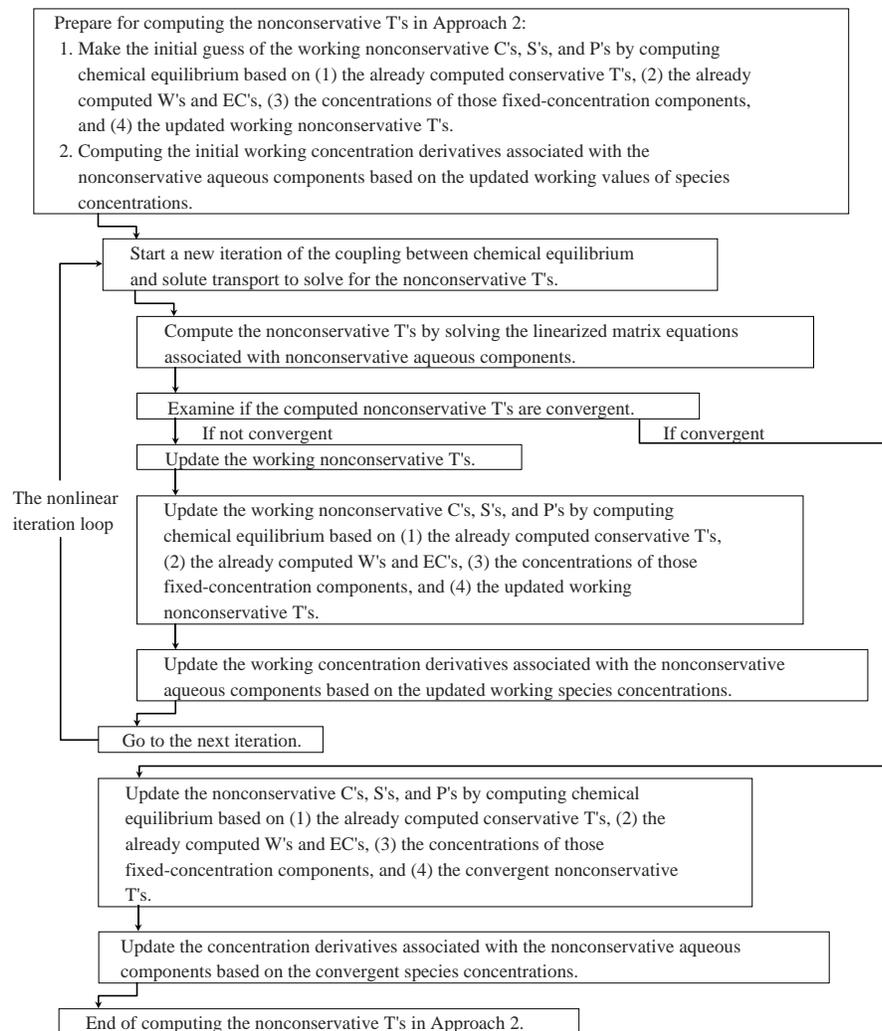End of computing the nonconservative T's in Approach 2.

Figure 3.7 The numerical strategy of the nonlinear iteration loop in Approach 2.

During the particle tracking process, the following steps are performed.

(1) Impose the specified boundary conditions, including Dirichlet, Cauchy, and flow-in Variable boundary conditions, onto all upstream boundaries for the concentration interpolation in the backward particle tracking.
(2) Implement backward particle tracking of global nodes to determine the Lagrangian concentrations at all global nodes for the current time step.
(3) Implement forward particle tracking of both peak/valley points and global nodes to maintain the shape of concentration distributions as much as possible.

A rough element, in which concentrations are not smoothly distributed, can be determined by comparing the concentrations of every forward-tracked point occurring in the element with their corresponding concentrations due to the interpolation of the concentrations of elemental nodes.   If the difference of any forward-tracked point is significant (i.e.,
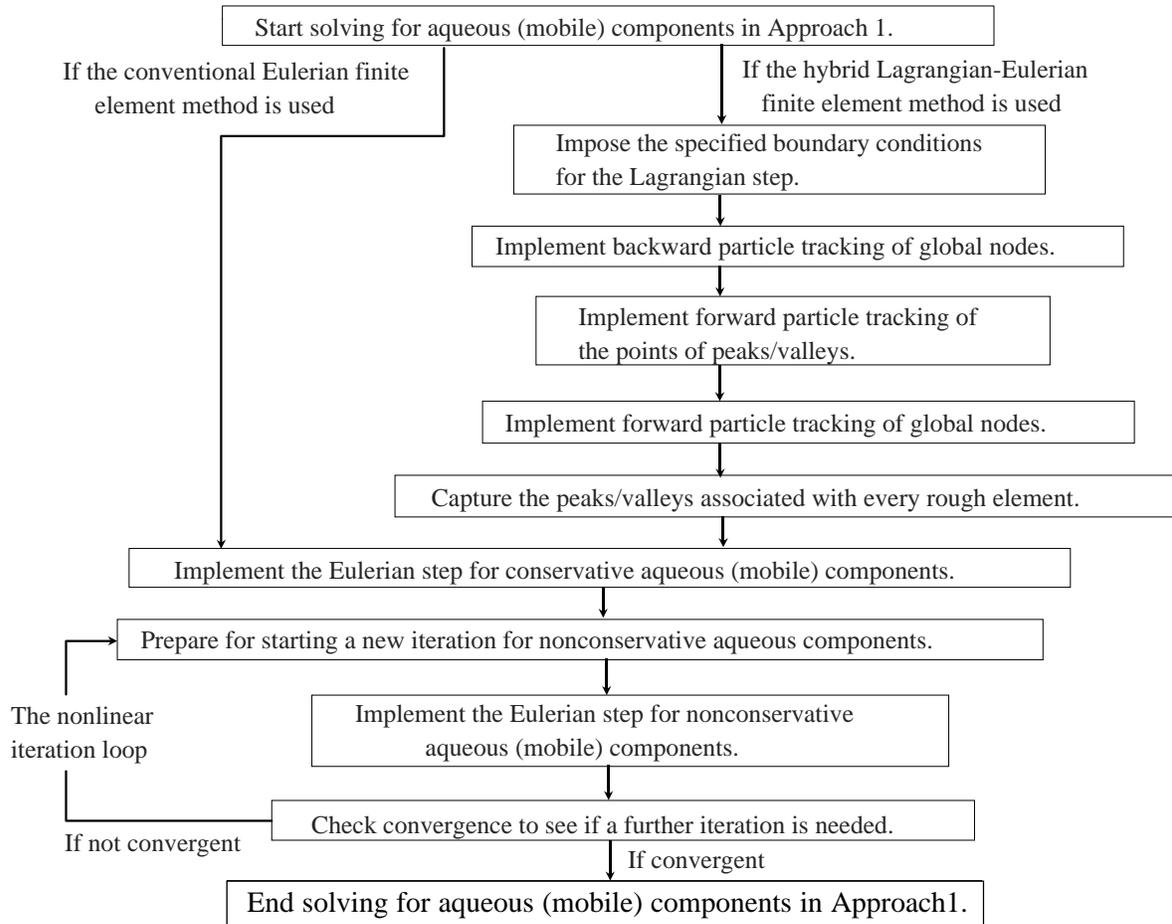
33

Start solving for aqueous (mobile) components in Approach 1.

If the conventional Eulerian finite element method is used

If the hybrid Lagrangian-Eulerian finite element method is used

Impose the specified boundary conditions for the Lagrangian step.

Implement backward particle tracking of global nodes.

Implement forward particle tracking of the points of peaks/valleys.

Implement forward particle tracking of global nodes.

Capture the peaks/valleys associated with every rough element.

Implement the Eulerian step for conservative aqueous (mobile) components.

Prepare for starting a new iteration for nonconservative aqueous components.

The nonlinear iteration loop

Implement the Eulerian step for nonconservative aqueous (mobile) components.

Check convergence to see if a further iteration is needed.

If not convergent

If convergent

End solving for aqueous (mobile) components in Approach1.

Figure 3.8 The numerical strategy of the peak/valley capturing technique in Approach 1.

not satisfying the prescribed criterion), then the element is a rough element. The peak/valley points in a rough element are captured by searching for the forward tracked points with the highest/lowest concentrations among all forward tracked nodes occurring in the element. If there are 5 aqueous components in a system, for instance, it is possible to have at most 5 peak and 5 valley points in a rough element of the system. This is so because the peak/valley capturing is implemented on a component concentration basis.

In Approach 1, the implementation of the peak/valley capturing is straightforward. As soon as the rough elements are determined, the peak/valley points of every rough element can be determined without any complexity. However, both the determination of rough elements and the capturing of peaks/valleys are more complicated in Approach 2. The rough element cannot be determined until a convergent solution of nonconservative aqueous components is obtained because the Lagrangian step is also involved in the nonlinear iteration loop. If a component of a system makes a captured peak/valley point, then the concentrations of the other components of this system need to be made up with backward particle tracking associated with the captured peak/valley point. This situation occurs in Approach 2 when nonconservative aqueous components exist in the system because the modified pore velocity used for particle tracking is component-dependent [Cheng, 1995].

3.4.3 The Strategy for Simulating Chemical Transport with Precipitated Species Dominating Chemical Equilibrium

The nonlinear iteration loop in solving the transport equations of nonlinear aqueous components plays a role in coupling chemical equilibrium with solute transport. Therefore, the basic requirement for computing chemical transport is the capability of computing chemical equilibrium at any point of interest. In other words, the model used for computing
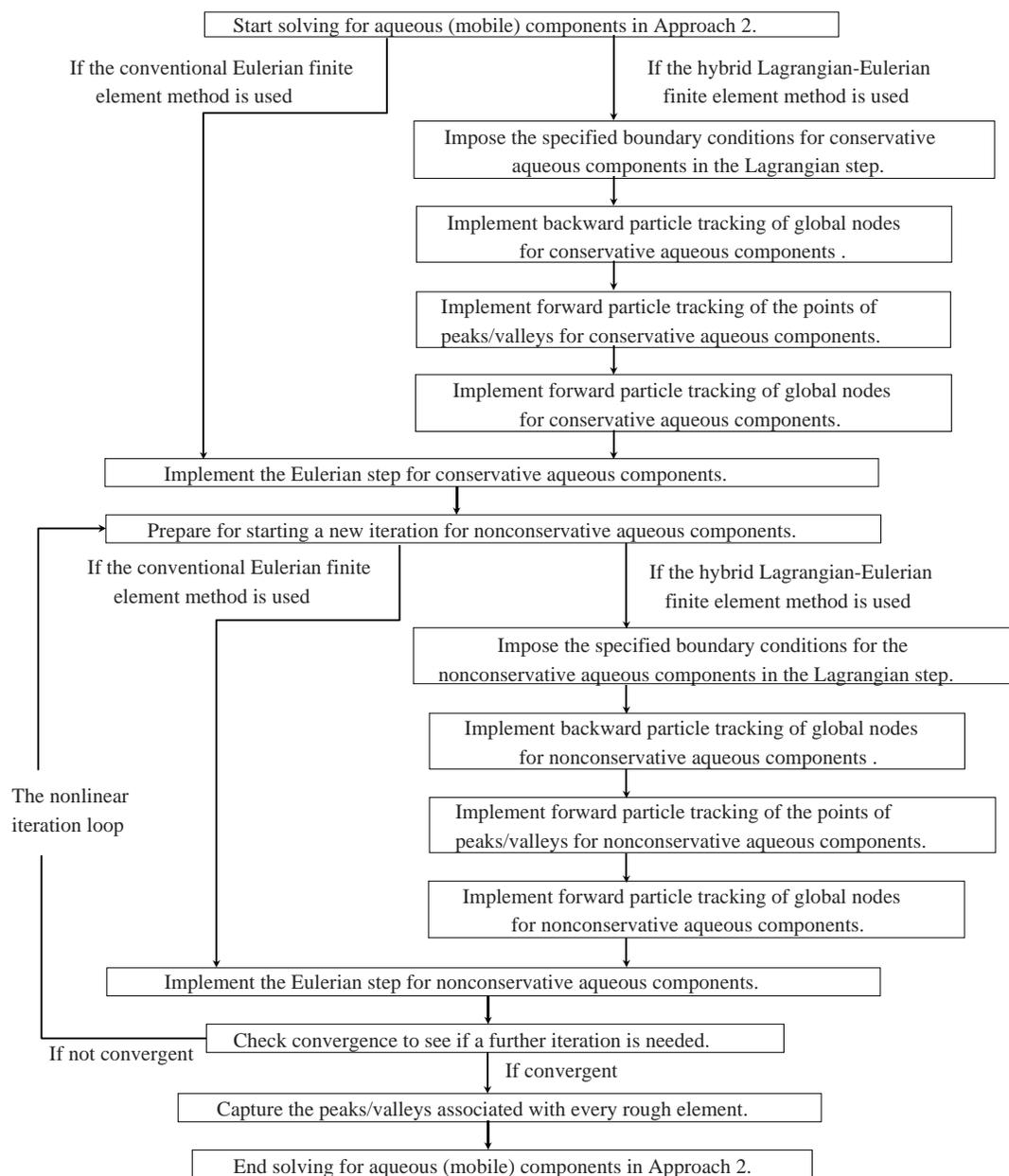
Start solving for aqueous (mobile) components in Approach 2.

If the conventional Eulerian finite element method is used

If the hybrid Lagrangian-Eulerian finite element method is used

Impose the specified boundary conditions for conservative aqueous components in the Lagrangian step.

Implement backward particle tracking of global nodes for conservative aqueous components .

Implement forward particle tracking of the points of peaks/valleys for conservative aqueous components.

Implement forward particle tracking of global nodes for conservative aqueous components.

Implement the Eulerian step for conservative aqueous components.

Prepare for starting a new iteration for nonconservative aqueous components.

If the conventional Eulerian finite element method is used

If the hybrid Lagrangian-Eulerian finite element method is used

Impose the specified boundary conditions for the nonconservative aqueous components in the Lagrangian step.

Implement backward particle tracking of global nodes for nonconservative aqueous components .

Implement forward particle tracking of the points of peaks/valleys for nonconservative aqueous components.

Implement forward particle tracking of global nodes for nonconservative aqueous components.

The nonlinear iteration loop

Implement the Eulerian step for nonconservative aqueous components.

Check convergence to see if a further iteration is needed.

If not convergent

If convergent

Capture the peaks/valleys associated with every rough element.

End solving for aqueous (mobile) components in Approach 2.

Figure 3.9 The numerical strategy of the peak/valley capturing technique in Approach 2.

chemical equilibrium should be able to handle as many chemical systems as possible. Most models take the aqueous component concentrations as their independent variables to compute chemical equilibrium when they are coupled with hydrological transport. However, if the amount of an aqueous component dissolved in the aqueous phase is negligible when compared with that precipitating in the solid phase (i.e., the precipitated concentration dominates), then the model might fail to conserve the mass of that aqueous component. In order to deal with such a situation, an essential strategy is developed [Cheng and Yeh, 1994a; 1997a] and included in 3DHYDROGEOCHEM. Based on the strategy originated in 3DHYDROGEOCHEM (without dominating precipitated species included), the modified total analytical concentrations (with the concentrations of dominating precipitated species excluded) rather than the total analytical concentrations are treated as the dependent variables of transport. In addition, the concentrations of all the dominating precipitated species at both the previous and current time steps are recorded for calculating the rate of change due to precipitation/ dissolution. The rate, playing the role of a source/sink term, appears on the right-hand side of the transport governing equation. The details in deriving transport equations and computing chemical equilibrium associated with the existence of dominating precipitated species can be found elsewhere [Cheng, 1995].

35

The governing equations can be written as follows.

for Approach 1:

$$
\theta\frac{\partial C_j}{\partial t} + \mathbf{V}\cdot\nabla C_j + \theta\frac{\partial\left(S_j+\overline{P}_j\right)}{\partial t} + \left(\theta\lambda_j + \frac{\partial\theta}{\partial t} + \frac{\rho^*}{\rho}q\right)\overline{\overline{T}}_j - \nabla\cdot\left(\theta\mathbf{D}\cdot\nabla\overline{T}_j\right) - \frac{\rho_o}{\rho}\,\overline{T}_j\nabla\!\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V}
$$

$$
= -\nabla\cdot\left[\theta\mathbf{D}\cdot\nabla\left(S_j+\overline{P}_j\right)\right] + qC_j^* + \frac{\partial\theta}{\partial t}C_j + \frac{\rho^*}{\rho}q\left(S_j+\overline{P}_j\right)
$$

$$
- \frac{\rho_o}{\rho}\left(S_j+\overline{P}_j\right)\nabla\!\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V} - \sum_{i=1}^{ndps} a_{ij}^p\left(\frac{\partial\theta p_i}{\partial t} + \theta\iota_i^p\right) \qquad j\in\left[1,\,N_a\right]
$$

(3.11)

for Approach 2:

$$
\theta\frac{\partial\overline{T}_j}{\partial t} + \sum_{i=1}^{NON}\mathbf{V}\cdot\Lambda_{j,i}\nabla\overline{T}_i - \sum_{i=1}^{NON}\nabla\cdot\left(\theta\mathbf{D}\cdot\Lambda_{j,i}\nabla\overline{T}_i\right) + \sum_{i=1}^{NSITE}\mathbf{V}\cdot\Lambda_{j,NON+i}\nabla EC_i
$$

$$
- \sum_{i=1}^{NSITE}\nabla\cdot\left(\theta\mathbf{D}\cdot\Lambda_{j,\,NON+i}\nabla EC_i\right) + \left(\theta\lambda_j + \frac{\partial\theta}{\partial t} + \frac{\rho^*}{\rho}q\right)\overline{\overline{T}}_j - \frac{\rho_o}{\rho}\,\overline{T}_j\nabla\!\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V}
$$

$$
= qC_j^* + \frac{\rho^*}{\rho}q\left(S_j+\overline{P}_j\right) - \frac{\rho_o}{\rho}\left(S_j+\overline{P}_j\right)\nabla\!\left(\frac{\rho}{\rho_o}\right)\cdot\mathbf{V} + \frac{\partial\theta}{\partial t}C_j - \sum_{i=1}^{ndps} a_{ij}^p\left(\frac{\partial\theta p_i}{\partial t} + \theta\iota_i^p\right)
$$

(3.12)

$$
j\in\left[1,\,N_a\right]
$$

where $\overline{T}_j$ is the modified total analytical concentration of the j-th aqueous component; $\overline{P}_j$ is the modified total precipitated concentration of the j-th aqueous component; p_i is the i-th dominating precipitated species; ndps is the number of dominating precipitated species; a_{ij}^p is the stoichiometric coefficient of the i-th precipitated species with respect to the j-th aqueous component; and ι_i^p is the first-order decay rate of the i-th precipitated species. The calculation of concentration derivatives associated with the existence of dominating precipitated species can be found elsewhere [Cheng, 1995]. Figures 3.10 and 3.11 show how the nonlinear iteration loops associated with the existence of dominating precipitated species work in Approaches 1 and 2, respectively.

Basically, the nonlinear loops for the cases with and without dominating precipitated species included are almost the same. This point can be easily observed by comparing (1) Figure 3.6a with Figure 3.8a and (2) Figure 3.6b with Figure 3.8b for Approaches 1 and 2, respectively. One should keep in mind that the existence of dominating precipitated species can only influence the computation corresponding to nonconservative aqueous components because only the nonconservative aqueous components can possibly be involved in dominating precipitated species. Therefore, the conservative $\overline{T}$'s in Figures 3.10 and 3.11 are exactly the same as T's in Figures 3.6 and 3.7. As mentioned earlier, the concentrations of dominating precipitated species must also be recorded for use in the computations of both solute transport and chemical equilibrium. In computing chemical equilibrium, the modified total analytical concentrations of nonconservative aqueous components and the concentrations of dominating precipitated species in the output might be different from those in the input. This difference is due to the precipitation/dissolution of dominating precipitated species. Precipitation makes a positive change and dissolution makes a negative change to the dominating precipitated species concentration. However, this change should approach zero when chemical equilibrium is computed according to a convergent solution because chemical equilibrium and solute transport are tightly coupled with the nonlinear iteration loop. The change might be negligible when compared with the dominating precipitated species concentration.

In order to take into account the change from a computational point of view, two arrays are employed to save the concentrations of dominating precipitated species in the computer codes [Cheng and Yeh, 1997a]. One array is used to save the fixed dominating precipitated species concentration. The other one is used to save the accumulated variable dominating precipitated species concentration. As soon as the accumulated variable reaches a certain absolute magnitude after a period of time, the fixed variable is reset with the sum of its original value and the accumulated variable

one. Meanwhile, the accumulated variable is reset to zero. The sum of these two arrays, representing the actual dominating precipitated species concentration, is routinely checked in the computer codes to assure the existence of dominating precipitated species during simulations. This needs to be performed because it is considered possible to completely dissolve a dominating precipitated species. When a dominating precipitated species is completely dissolved, the mass reaction equation of this dominating precipitated species may not be needed in the subsequent computations of chemical equilibrium.

3.5 The Strategy for Solving the Governing Equation of Heat Transfer

The governing equation of heat transfer is very similar to the transport equation of a conservative aqueous component. Thus, the strategy used in computing the transport of conservative aqueous components is applied to computing heat transfer. Figure 3.12 shows the numerical strategy. The technique of peak/valley capturing is not applied to solving the governing equation of heat transfer because the temperature distribution is assumed smooth across any discretized element in the subsurface system of interest.

< Define >
$\overline{T}$'s = the modified total analytical concentrations of aqueous components;
C's = the total dissolved concentrations of aqueous components;
S's = the total sorbed concentrations of aqueous components;
$\overline{P}$'s = the modified 1-st type total precipitated concentrations of aqueous components;
W's = the total analytical concentrations of adsorbent components;
EC's = the capacities of ion-exchange sites.
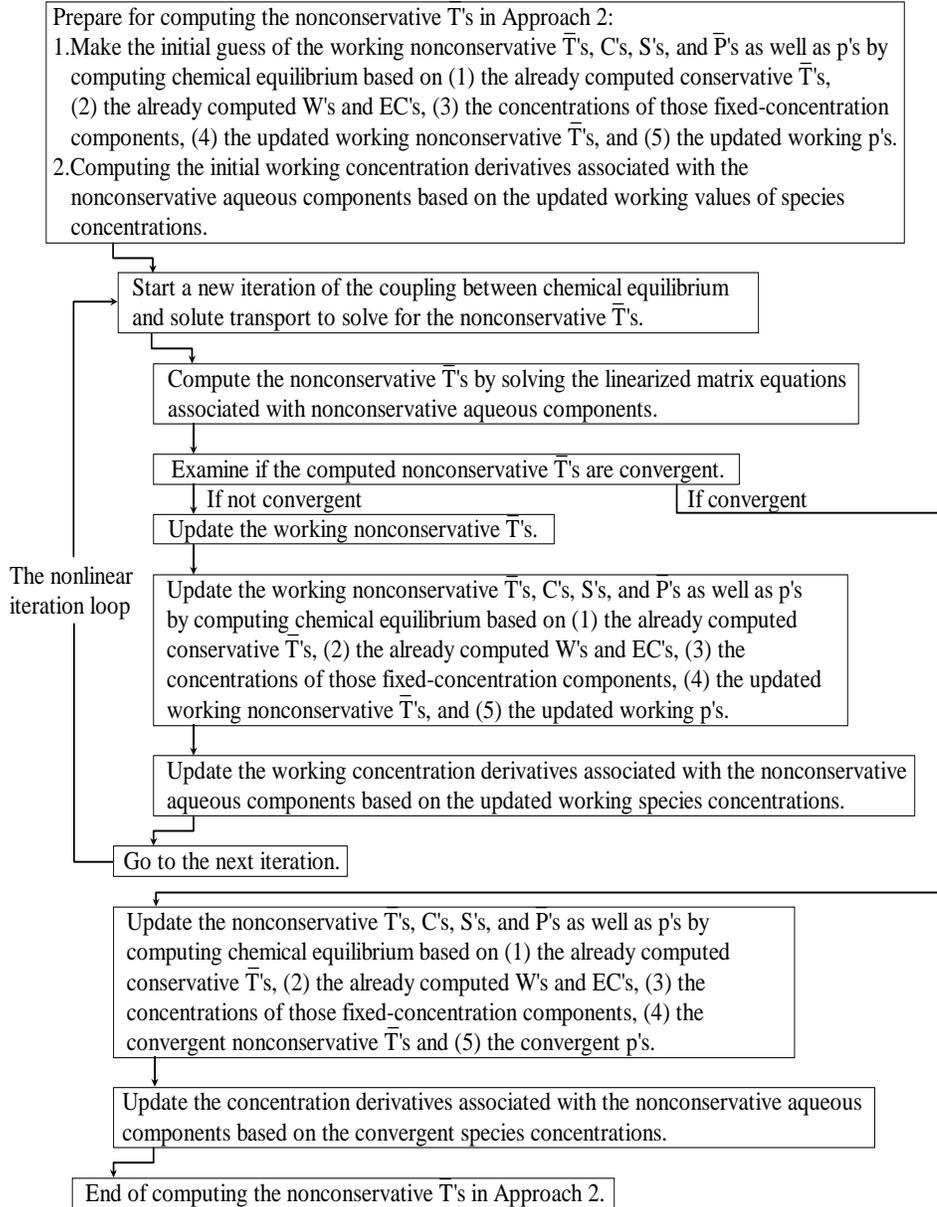p's = the concentrations of dominating precipitated species.

Prepare for computing the nonconservative $\overline{T}$'s in Approach 1:
  Make the initial guess of the working nonconservative $\overline{T}$'s, C's, S's, and $\overline{P}$'s as well as p's by computing chemical equilibrium based on (1) the already computed conservative $\overline{T}$'s, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, (4) the updated working nonconservative $\overline{T}$'s, and (5) the updated working p's.

Start an new iteration of the coupling between chemical equilibrium and solute transport to solve for the nonconservative $\overline{T}$'s.

Compute the nonconservative $\overline{T}$'s by solving the linearized matrix equations associated with nonconservative aqueous components.

Examine if the computed nonconservative $\overline{T}$'s are convergent.

If not convergent / If convergent

Update the working nonconservative $\overline{T}$'s.

The nonlinear iteration loop

Update the working nonconservative $\overline{T}$'s, C's, S's, and $\overline{P}$'s by computing chemical equilibrium based on (1) the already computed conservative $\overline{T}$'s, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, (4) the updated working nonconservative $\overline{T}$'s, and (5) the updated working p's.

Go to the next iteration.

Update the nonconservative $\overline{T}$'s, C's, S's, and $\overline{P}$'s as well as p's by computing chemical equilibrium based on (1) the already computed conservative $\overline{T}$'s, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, (4) the convergent nonconservative $\overline{T}$'s, and (5) the convergent p's.

End of computing the nonconservative $\overline{T}$'s in Approach 1.

Figure 3.10 The numerical strategy of the nonlinear iteration loop associated with the existence of dominating precipitated species in Approach 1.
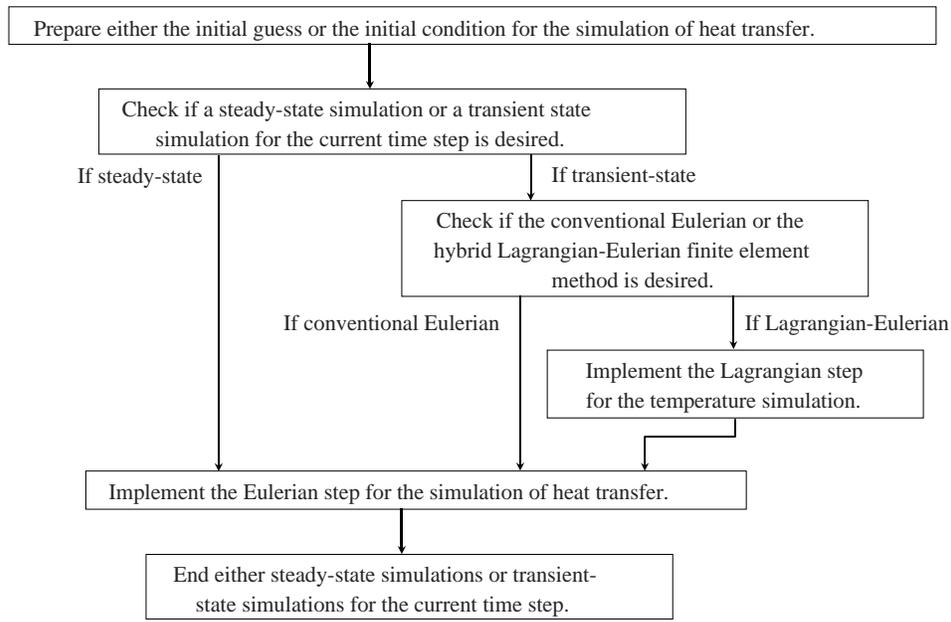
< *Define* >

$\bar{\bar{T}}$'s = the modified total analytical concentrations of aqueous components;

C's = the total dissolved concentrations of aqueous components;

S's = the total sorbed concentrations of aqueous components;

$\bar{P}$'s = the modified 1-st type total precipitated concentrations of aqueous components;

W's = the total analytical concentrations of adsorbent components;

EC's = the capacities of ion-exchange sites.

 p's = the concentrations of dominating precipitated species.

Prepare for computing the nonconservative $\bar{\bar{T}}$'s in Approach 2:

1. Make the initial guess of the working nonconservative $\bar{\bar{T}}$'s, C's, S's, and $\bar{P}$'s as well as p's by computing chemical equilibrium based on (1) the already computed conservative $\bar{\bar{T}}$'s, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, (4) the updated working nonconservative $\bar{\bar{T}}$'s, and (5) the updated working p's.

2. Computing the initial working concentration derivatives associated with the nonconservative aqueous components based on the updated working values of species concentrations.

Start a new iteration of the coupling between chemical equilibrium and solute transport to solve for the nonconservative $\bar{\bar{T}}$'s.

Compute the nonconservative $\bar{\bar{T}}$'s by solving the linearized matrix equations associated with nonconservative aqueous components.

Examine if the computed nonconservative $\bar{\bar{T}}$'s are convergent.

If not convergent   If convergent

Update the working nonconservative $\bar{\bar{T}}$'s.

The nonlinear iteration loop

Update the working nonconservative $\bar{\bar{T}}$'s, C's, S's, and $\bar{P}$'s as well as p's by computing chemical equilibrium based on (1) the already computed conservative $\bar{\bar{T}}$'s, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, (4) the updated working nonconservative $\bar{\bar{T}}$'s, and (5) the updated working p's.

Update the working concentration derivatives associated with the nonconservative aqueous components based on the updated working species concentrations.

Go to the next iteration.

Update the nonconservative $\bar{\bar{T}}$'s, C's, S's, and $\bar{P}$'s as well as p's by computing chemical equilibrium based on (1) the already computed conservative $\bar{\bar{T}}$'s, (2) the already computed W's and EC's, (3) the concentrations of those fixed-concentration components, (4) the convergent nonconservative $\bar{\bar{T}}$'s and (5) the convergent p's.

Update the concentration derivatives associated with the nonconservative aqueous components based on the convergent species concentrations.

End of computing the nonconservative $\bar{\bar{T}}$'s in Approach 2.

Figure 3.11   The numerical strategy of the nonlinear iteration loop associated with the existence of dominating precipitated species in Approach 2.

Figure 3.12   The numerical strategy for solving the governing equation of heat transfer.

Chapter 4
VERIFICATION AND APPLICATION

4.1 Verification

3DHYDROGEOCHEM has been verified with a variety of designed example problems as indicated in Table 4.1. As mentioned in Chapter 2, two approaches (i.e., Approaches 1 and 2 described by Eqs. (2.9) and (2.10), respectively) are employed to deal with chemical transport of nonconservative mobile components. For clarity in programming, they are included in two separate computer codes. Each code contains the modules for subsurface flow, heat transfer, and chemical transport. The implementation of the subsurface flow and the heat transfer modules are identical in both codes. For convenience, "Approach 1" and "Approach 2" are still used to represent these two codes as shown in Table 4.1. In the following, brief descriptions are given to the 25 example problems in Table 4.1 even though their results are not presented here.

Example problems 1 through 3 are used to verify the subsurface flow module. Since they were originally designed to demonstrate the 3DFEMWATER model [Yeh and Cheng, 1993a], the computational results from the model are compared with the associated numerical solutions of 3DFEMWATER.

Example problems 4 through 6, originally designed for illustrating the 3DLEWASTE model [Yeh and Cheng, 1993b], are used to verify the computation of solute transport involving the physical processes of advection, dispersion, and diffusion in the chemical transport module. The associated computational results, then, are compared with the numerical solutions of 3DLEWASTE.

Example problem 7 is used to verify the heat transfer module. As discussed, the governing equation of heat transfer is similar to that for a conservative mobile component of chemical transport. They differ only in the parameter coefficients shown in the governing equations. In other words, one can adjust those parameter coefficients to make the two governing equations identical mathematically. Therefore, the computational results of heat transfer can be compared with that of a conservative component transport, which has been verified with example problems 4 through 6 above, to perform the verification of the heat transfer module.

Example problem 8 is a two-dimensional salt water intrusion problem (known as "Henry's example") [Cheng and Yeh, 1993b]. This example problem is employed to verify the coupling between subsurface flow and chemical transport. The coupling logic can be examined through this even though only a single component transport is included here. The associated computational result is compared with that from Huyakorn et al.'s models [Cheng and Yeh, 1993b].

To verify the use of hexahedral, triangular prism, and tetrahedral elements in the 3DHYDROGEOCHEM model, these three types of elements have been applied to compose the simulated domains of the above eight example problems. Very good agreement among the computational results associated with the three types of elements have been obtained.

Example problems 9 through 14 are employed to verify the basic coupling structure between chemical equilibrium and solute transport, which is built in the chemical transport module, with respect to a variety of chemical systems. There are four components and six species taken into account in example problem 9, seven components and fifteen species for example problem 10, four components and seven species for example problem 11, six components and twenty-seven species for example problem 12, seven components and fifteen species for example problem 13, and four components and twenty-three species for example problem 14.   In these examples, a uniform distribution of the total analytical component concentration is given initially (different values for different components). The total analytical component concentration is specified to be the same as the initial condition on the upstream boundary (i.e., a Dirichlet boundary condition on one side of a one-dimensional column) for each mobile component, while the natural outgoing dissolved concentration is considered on the downstream boundary (i.e., a flow-out Variable boundary condition on the other side of the one-dimensional column). With such a setup, the analytical solution is available for a transient-state simulation. The corresponding verification can thus be achieved.

Example problem 15 is designed to verify the implementation of van't Hoff equation for the temperature-dependent chemical equilibrium. With a specific setting of initial and boundary conditions, analytical solutions are available [Cheng and Yeh, 1997b].

Table 4.1 Example problems for verification.

	Example problems		Verification results	
Number	Description		Approach 1	Approach 2
1	1-D subsurface flow (T)		V	V
2	2-D subsurface flow (S)		V	V
3	3-D subsurface flow (S)		V	V
4	1-D single component transport (T)		V	V
5	2-D single component transport (T)		V	V
6	3-D single component transport (T)		V	V
7	1-D heat transfer (S,T)		V	V
8	Henry's salt water intrusion example (S)		V	V
9	1-D transport w/ complexation (S, T)		V	V
10	1-D transport w/ adsp. (S, T)		V	V
11	1-D transport w/ ion-exchange (S, T)		V	V
12	1-D transport w/ complexation and prep. (S, T)		V	V
13	1-D transport w/ complexation and adsp. (S, T)		V	V
14	1-D transport w/ acid-base and redox (S, T)		V	V
15	1-D heat transfer and transport w/ prep. (T)		V	V
16	1-D transport w/ adsp. under LI (T)		V	V
17	1-D transport w/ complexation and prep. (T)		V	V
18	1-D transport w/ complexation and adsp. (T)		V	V
19	2-D Uranium Tailing (T)		V	NA
20	1-D transport w/ DP (T)		V	V
21	1-D transport w/ DP and adsp. under LI (T)		V	V
22	1-D single component transport w/ PC (T)		V	V
23	1-D two-component transport w/ PC (T)		V	V
24	1-D transport w/ decay (T)		V	V
25	1-D transport w/ decay and DP (T)		V	V

S=steady, T=transient, adsp.=adsorption, prep.=precipitation, LI=linear isotherm, PC= peak/valley capturing, DP= dominating precipitated species, V=verified, NA=not applicable.

In Example problem 16, the linear isotherm is utilized to distribute chemical concentrations into the aqueous and the solid phases (i.e., the well-known Kd approach). Five components, with different levels of partitioning between the aqueous and the solid phases, are included. When the mesh Courant number (with respect to the groundwater pore velocity) is less than or equal to 1, Approaches 1 and 2 provide identical computational results. However, a more accurate result can be obtained by using Approach 2 [Cheng and Yeh, 1993a] when the mesh Courant number is greater than 1. This is because Approach 2 contains the spirit of the method of characteristics that considers chemical transport along the chemical reaction-based characteristics [Yeh and Cheng, 1996]. For this particular case, the spirit can be fully implemented with Approach 2. In other words, the retarding effect can be correctly presented with Approach 2.

Example problems 17 through 19 were originally employed for demonstrating the capability of the HYDROGEOCHEM model [Yeh and Tripathi, 1991] in dealing with chemical transport. These problems are also used to further test 3DHYDROGEOCHEM. The computational results are thus compared with those of HYDROGEOCHEM. As indicated in Table 4.1, Approach 2 is not applicable to solve example problem 19. This is because Eq. (2.11) is not satisfied everywhere in the simulated domain through the simulation. As a consequence, an oscillating iteration, rather than a convergent solution, is obtained in the first time step during the transient simulation.

Example problem 20 is to verify the implementation associated with dominating precipitated species (DP). From a mathematical point of view, any precipitated species can be assigned as a DP in 3DHYDROGEOCHEM even though it may not really be dominating chemical equilibrium. Therefore, the numerical approach associated with DP can be verified by comparing the computational results from simulations with and without DP assigned for the same system (i.e., a precipitated species is assigned as a DP in the "with" case but as a regular precipitated species in the "without" case) [Cheng and Yeh, 1994a].

Example problem 21 combines example problems 16 and 20 to help examine if the computational results, especially the one associated with Approach 2, are as expected according to the computational logic for both DP and the retarding effect [Cheng and Yeh, 1994b]. With a specific design on the total analytical component concentration, the distribution coefficient, and the mesh Courant number for a one-dimensional problem, the quantitative behavior of the computational results can be predicted.

Example problems 22 and 23 are designed to verify the implementation of the peak/valley capturing technique in both Approaches 1 and 2. A FORTRAN debugger has been used to help check the associated computational procedures in 3DHYDROGEOCHEM.

For the convenience of dealing with DP, the radioactive decay term (i.e., the first-order decay term) is not taken into account when composing matrix equations of chemical transport (i.e., it is not included in the Eulerian step). Rather, it is computed within the Lagrangian step [Cheng, 1995]. Example problems 24 (without DP) and 25 (with DP) are designed to investigate the correctness of taking care of this term in 3DHYDROGEOCHEM.

4.2 Application

Verification of the designed example problems are employed to determine the correctness of the numerical formulation. Therefore, the verification example problems are usually simplified. In order to demonstrate the capability of 3DHYDROGEOCHEM in handling real three-dimensional problems, four example problems are designed. These four example problems, including (1) a three-dimensional subsurface flow problem, (2) a three-dimensional heat transfer problem, (3) a three-dimensional reactive chemical transport problem, and (4) a three-dimensional problem with all the three modules coupled, are investigated within a domain of ($500 \times 200 \times 200$ dm^3) as shown in Figure 4.1. The domain is discretized with 8000 (= $20 \times 20 \times 20$) elements and 9261 (= $21 \times 21 \times 21$) nodes. Instead of an equally-spaced discretization, the nodes are distributed at 0, 10, 25, 50, 80, 125, 170, 200, 225, 240, 250, 260, 275, 300, 330, 375, 420, 450, 475, 490, and 500 dm in the x-direction, while at 0, 5, 10, 20, 35, 50, 65, 80, 90, 95, 100, 105, 110, 120, 135, 150, 165, 180, 190, 195, and 200 dm in both the y- and the z-directions. The successive point iteration method [Cheng, 1995] is employed to solve linear/linearized matrix equations in the four example problems of application. To also investigate the performances of the steady-state and the transient-state computations, both simulations are included in each of the application example problems. Basically, a steady-state solution is first computed based on the given pre-initial and boundary conditions. Then this computed steady-state solution is used as the initial condition to precede the transient-state simulation. Therefore, the pre-initial condition, time-dependent boundary condition, and time-dependent sources/sinks are given for the simulation in this application.

4.2.1 Application 1: A Three-Dimensional Example of Subsurface Flow

4.2.1.1   Problem Description

In this example problem, the simulated domain is composed of two materials as shown in Figure 4.2. Material parameters needed for the simulation of subsurface flow are listed in Table 4.2. The follow equations are assumed to describe the soil properties of the two materials.

$$\mathbf{q} = \begin{cases} \mathbf{q}_{\text{eff}} & \text{if } h \geq 0 \\ \mathbf{q}_{\text{eff}} - \left[1 - \dfrac{h_a - h}{h_a} \right] (\mathbf{q}_{\text{eff}} - \mathbf{q}_a) & \text{if } h < 0 \end{cases} \qquad (4.1)$$
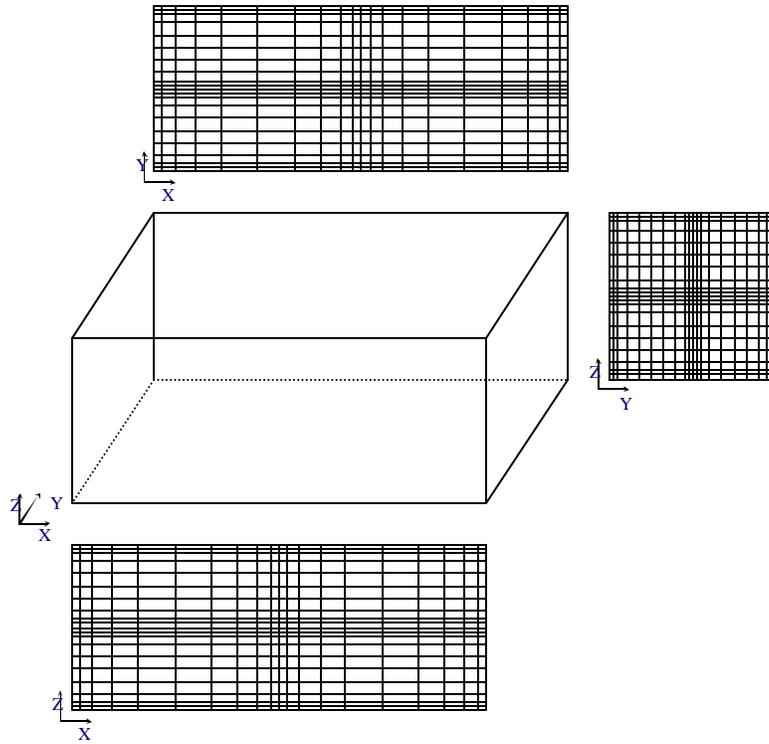
Figure 4.1 The geometry and discretization of the simulated domain in application.
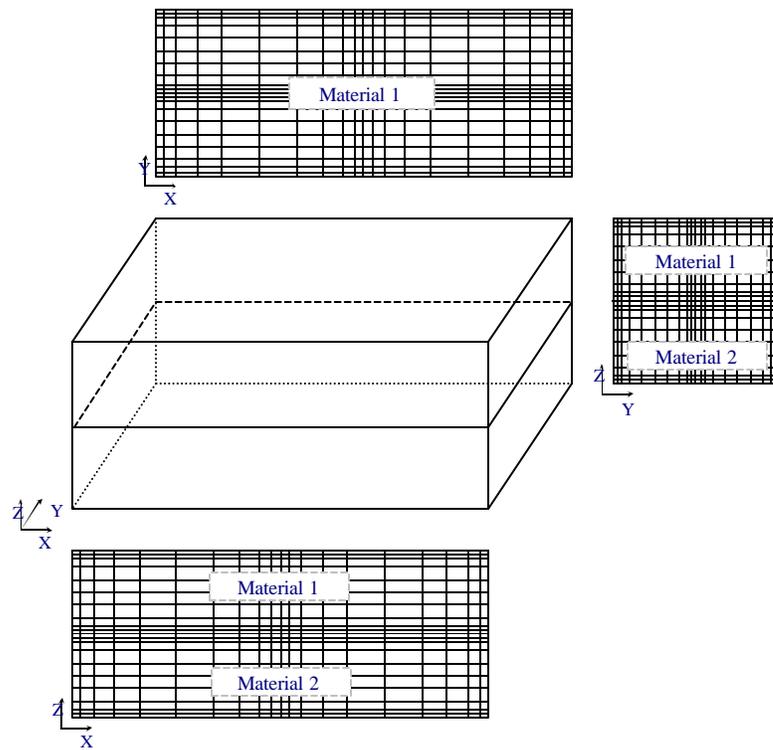
Figure 4.2 A two-material simulated domain in application.

Table 4.2 Material parameters needed for the simulation in application 1.

Material parameter	Material 1	Material 2
Saturated K_{xx} [dm day^{-1}]	1.0	5.0
Saturated K_{yy} [dm day^{-1}]	1.0	5.0
Saturated K_{zz} [dm day^{-1}]	1.0	5.0
Saturated K_{xy} (or K_{xy}) [dm day^{-1}]	0.0	0.0
Saturated K_{xz} (or K_{xz}) [dm day^{-1}]	0.0	0.0
Saturated K_{yz} (or K_{yz}) [dm day^{-1}]	0.0	0.0
Effective porosity, θ_{eff} [dimensionless]	0.3	0.2
Soil property parameter 1, θ_a [dimensionless]	0.15	0.1
Soil property parameter 2, h_a [dm]	-1000.0	-1000.0

$$\mathbf{K} = \mathbf{K_{sat}} \qquad\qquad\qquad \text{if } h \geq 0$$

$$\mathbf{K} = \mathbf{K_{sat}} \left[\frac{h_a - h}{h_a} \right] \qquad\qquad \text{if } h < 0 \qquad\qquad (4.2)$$

$$\frac{d\theta}{dt} = \frac{\theta_a - \theta_{sat}}{h_a} \qquad\qquad (4.3)$$

where q is moisture content; q_{eff} is effective porosity; h is pressure head; q_a is the first soil property parameter; h_a is the second soil property parameter; **K** is the hydraulic conductivity tensor; $\mathbf{K_{sat}}$ is the hydraulic conductivity tensor under the saturated condition, which is composed of six components: the saturated K_{xx}, K_{yy}, K_{zz}, K_{xy}, K_{xz}, and K_{yz}.

The pressure head is given 200 dm everywhere as the pre-initial condition. The top, bottom, front, and back boundaries are assumed impermeable, while the total head is specified at both the left and the right boundaries. Three point sinks are activated at (250, 100, 80), (250, 100, 90), and (250, 100, 95) in the transient simulation. Such a hypothetical setup is considered to simulate pumping in a confined aquifer. Mathematically, the four impermeable boundaries are described with the Cauchy boundary condition of a zero flux, and the Dirichlet boundary condition is used to describe the total heads on the left as well as the right boundaries. In this example problem, the total head is specified at 200 dm for the right boundary. On the other hand, the specified total head on the left boundary is time-dependent and is described by Figure 4.3. The withdrawal rate of the three point sinks is kept constant during the transient-state simulation, as described by Figure 4.4. In both Figures 4.3 and 4.4, the corresponding values at time = 0 day are used in the steady-state simulation. There is actually no point sink (the withdrawal rate is 0 dm^3/day at time = 0 day) and a total head of 200 dm is specified on the left boundary for the steady-state simulation. In Figure 4.3, the specified total head is 200 dm from time = 0 day to 20 days, and then it linearly increases with time from 200 dm (time = 20 day) to 220 dm (time = 40 days). In Figure 4.4, the pumping rate is 0 dm^3/day initially, but is suddenly changed to 5000 dm^3/day at time = 10^{-6} day (the negative sign associated with rate values in Figure 4.4 represents withdrawal; if a source is considered, the sign is positive to represent injection). This rate is kept unchanged until the end of the transient simulation (i.e., 40 days). There is no pumping until the start of the transient simulation. In this case, three time points at time = 0, 10^{-6}, and 40 days are used to describe the time-dependent source/sink rate in Figure 4.4. Twenty time steps, with 5 days as the time-step size, are performed in the transient-state simulation. Absolute error tolerances of 5×10^{-5} dm and 1×10^{-5} dm are used to control the convergency of iteration solutions for the steady-state and the transient-state simulations, respectively.
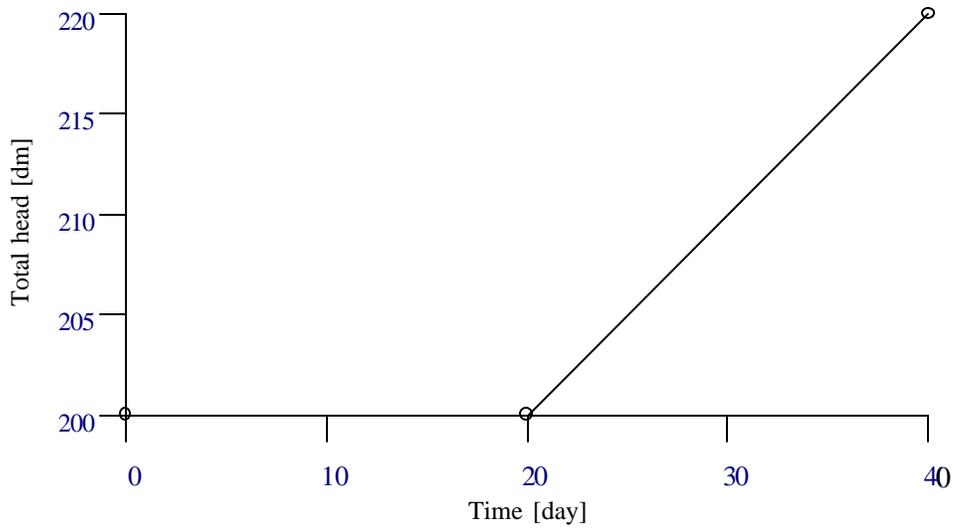
Figure 4.3 The time-dependent specified total head on the left boundary in application 1.
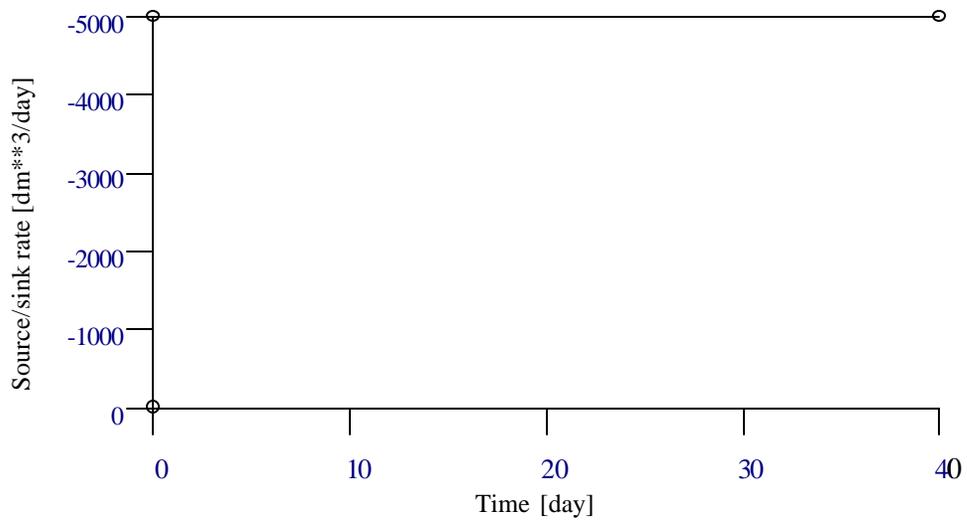
Figure 4.4 The time-dependent source/sink rate for each sink in application 1.

4.2.1.2 Simulation Results

According to the boundary condition specified at time = 0 day, it is straightforward to determine the associated steady-state solution for this application. The true steady-state solution should have a total head of 200 dm throughout the simulated domain.   Although the pre-initial condition (playing a role as an initial guess in computing for a steady-state solution) is given a pressure head of 200 dm everywhere, which was far from the true solution for most nodes, a satisfactory convergent solution has been still obtained. In this particular case, the steady-state solution mirrors water saturation within the entire domain. For some cases, however, a convergent solution could be difficult to reach when the pre-initial condition is far from the true solution. In addition to this, one may not be able to obtain a convergent solution if moisture content and hydraulic conductivity change drastically with the pressure head.

Using the steady-state solution as the initial conditions, the transient-state simulation is performed. Figures 4.5 and 4.6 show the computational results at time = 20 and 40 days, respectively. Both Figures 4.5(a) and 4.6(a) show the equipotential planes of the pressure head at 0 dm, 90 dm, and 180 dm. In Figure 4.5(a), a unsaturated zone can be observed, which is above the pressure head plane of 0 dm. This zone depletes as the specified total head on the left boundary increases from 200 dm (at time = 20 days) to 220 dm (at time = 40 days), as shown in Figure 4.6(a). It is also obvious to see how this increase affects the flow field within the simulated domain by comparing Figures 4.5(b) and 4.6(b). In Figure 4.5(b), the equipotential lines of the total head at 197, 190, 185, 180, 175, 150, and 120 dm are plotted (to make the figure clear, the last three are not marked) on the plane of y = 100 dm. The magnitude and direction of the flow rate are also described with different sizes of arrows at x = 0 and 500 dm. Since the three pumping wells are at the center of the domain with respect to both the x- and the y-directions, a symmetric result in these two directions can be observed. On the other hand, the off-center locations of the three wells and two-material composition in the z-direction yield an evident contrast between the two parts above and below z = 100 dm.

The symmetry in the x-direction no longer exists as soon as the specified total head on the left boundary increases with time (Figure 4.6(b)). For clarity, the equipotential lines of the total head at 185, 180, 170, 150, and 120 dm are not marked. Along with the asymmetry of the total head distribution in the x-direction, one can also see the big difference between the incoming flow rate on the right boundary and that on the left boundary.

4.2.2 Application 2: A Three-Dimensional Example of Heat Transfer

4.2.2.1 Problem Description

Three materials are included to compose the simulated domain as pictured in Figure 4.7. The required material parameters for computing heat transfer are listed in Table 4.3. The specific heat of groundwater is 3.12035×10^{15} dm^2 day^{-1} ^{o}K^{-1} [Yeh and Luxmoore, 1983]. A uniformly distributed temperature of 290 ^{o}K is given as the pre-initial condition. The entire domain is assumed water-saturated. A one-dimensional constant subsurface flow of 0.3 dm/day, along the x-direction, is set to flow through the simulated domain. The time-step size is 10 days, which combines with 20 time steps to perform a 200 day transient simulation.

A flow-in Variable boundary condition is applied to the left boundary, while a flow-out Variable boundary condition can be simulated for the right boundary.   Figure 4.8 describes the time-dependent flow-in temperature for region A on the left boundary, and Figure 4.9 is for region B. Regions A and B are indicated in Figure 4.10. The four corner nodes of region A are (0, 65, 65), (0, 135, 65), (0, 135, 135), and (0, 65, 135). In Figure 4.8, the flow-in temperature is specified at 298.3 ^{o}K at time = 0 day and is changed to 308.3 ^{o}K from time = 10^{-6} day on, whereas the flow-in temperature stays at 298.3 ^{o}K in Figure 4.9. This is to imitate a heat boundary source from region A through the transient-state simulation. A Dirichlet boundary condition is also implemented for the nodes on the top, bottom, front, and back boundaries if the x-coordinates are greater than or equal to 5 dm and less than or equal to 95 dm (i.e., x [5, 195]). Therefore, there are 1520 (= 19×21×4 - 19×4) Dirichlet nodes in total.   The time-dependent specified temperature associated with these nodes is described in Figure 4.11: 298.3 ^{o}K at time = 0 day and 288.3 ^{o}K from time = 10^{-6} day until the end of the transient-state simulation. The above setup can hardly happen in the real world, but it can be achieved in the laboratory. A relative error tolerance of 10^{-5} is employed to determine the convergency of numerical solutions in both the steady-state and the transient-state simulations.

4.2.2.2 Simulation Results

Because the governing equation of heat transfer is a linear partial differential equation, the pre-initial condition is irrelevant to the computation for a convergent steady-state solution. Thus, the whole simulated domain is at a temperature of 298.3 ^{o}K after the steady-state simulation in this application.

By using this steady-state solution as the initial condition and the given boundary condition for the transient simulation, Figure 4.12 displays the computational result at time = 100 days, while Figure 4.13 is for time = 200 days. In
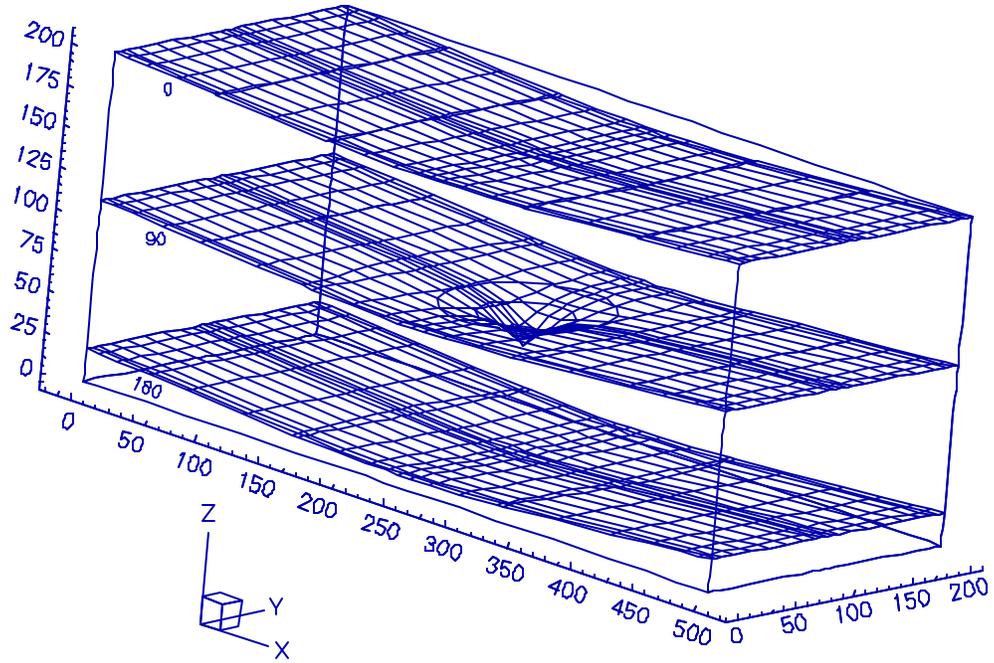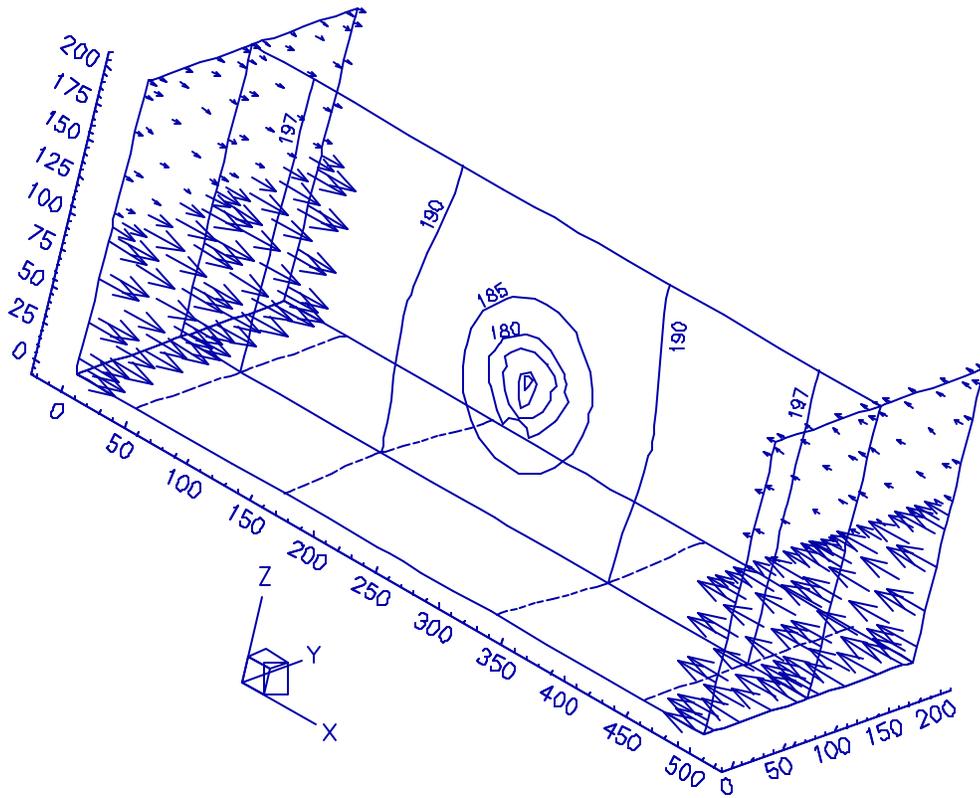
Figure 4.5 **(a) The pressure head distribution and (b) the total head distribution and boundary flow rates at time = 20 days in application 1.**
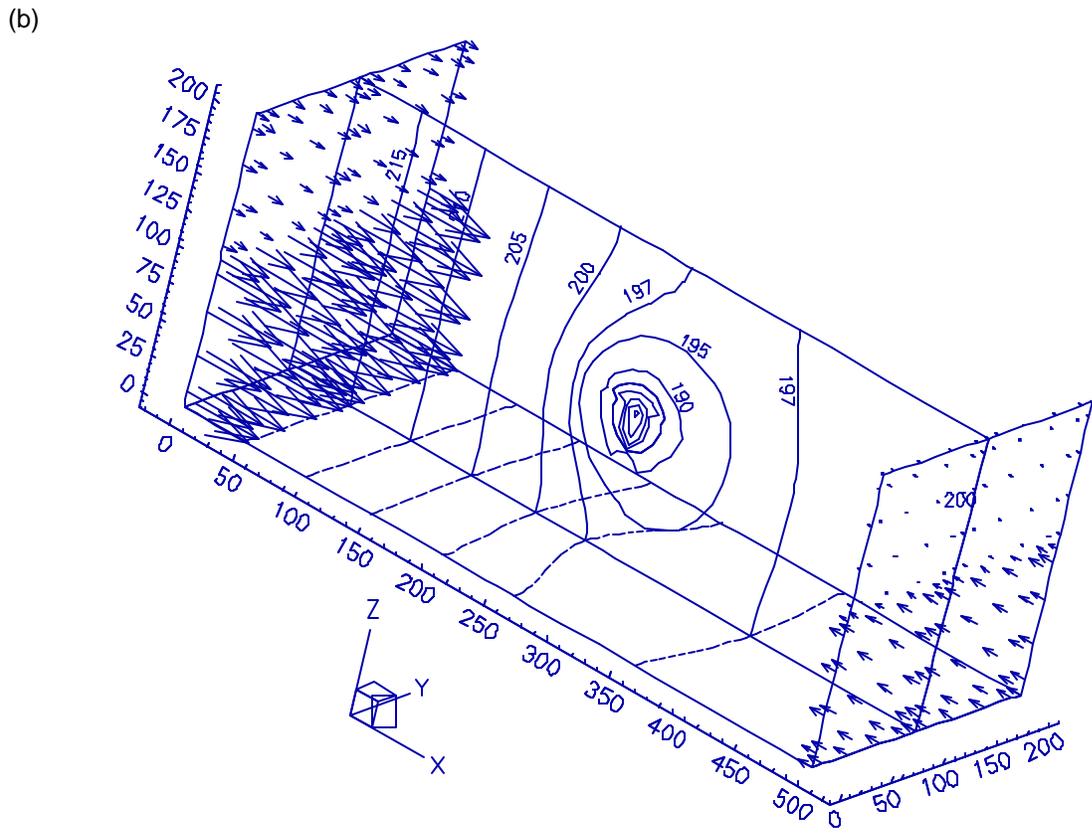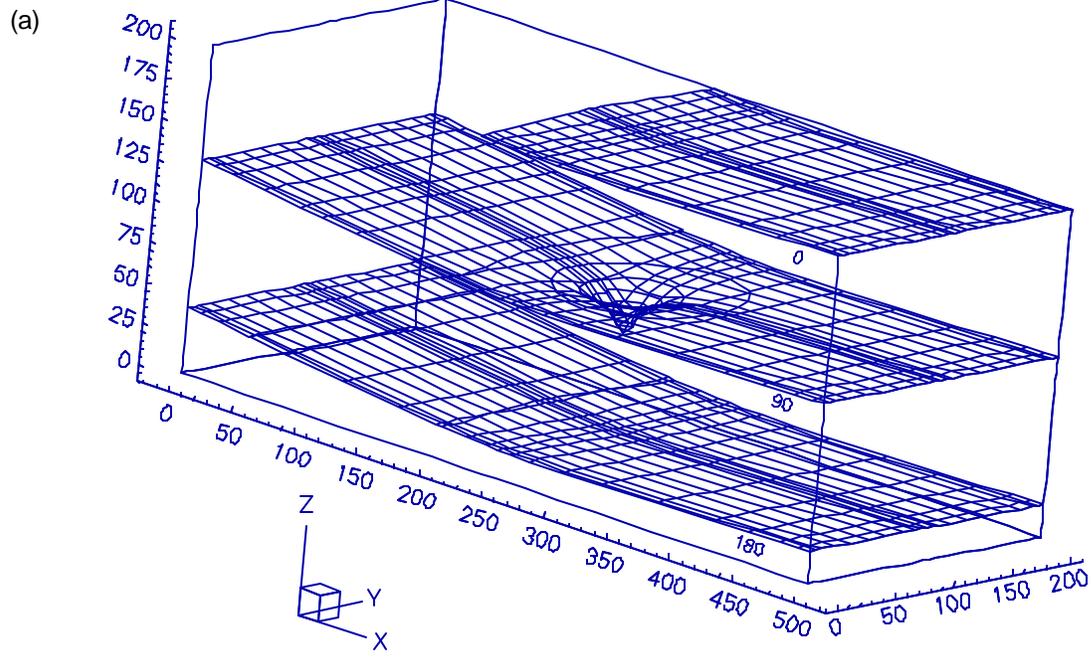
Figure 4.6 (a) The pressure head distribution and (b) the total head distribution and boundary flow rates at time = 40 days in application 1.
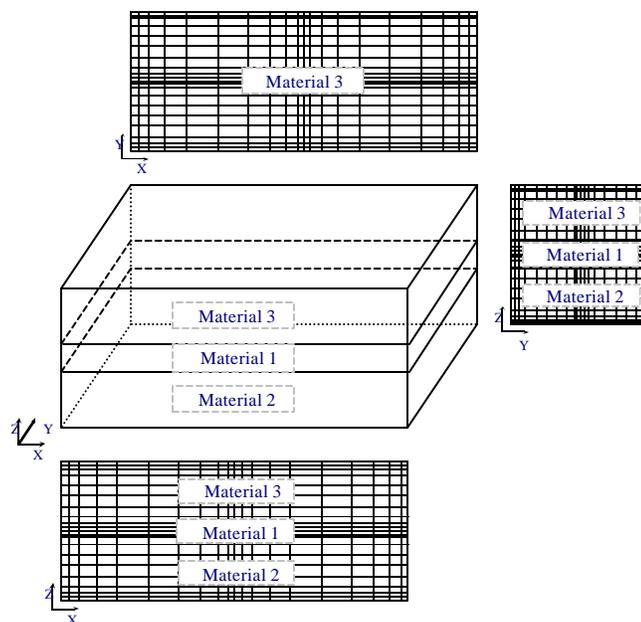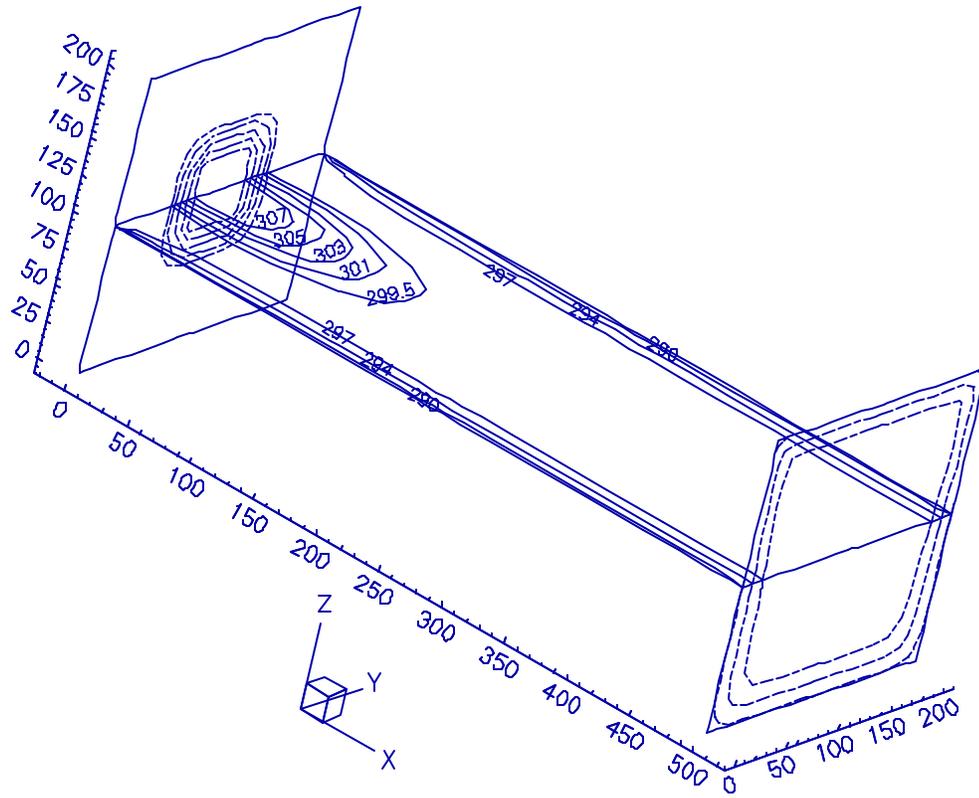
Figure 4.7 A three-material simulated domain in application.

Table 4.3 Material parameters needed for the simulation in application 2.

Material parameter	Material 1	Material 2	Material 3
Effective porosity, 2_{sat} [dimensionless]	0.3	0.3	0.3
Specific heat of dry medium, C_m [dm^2 day^{-1} $^oK^{-1}$]	1.0×10^{18}	1.0×10^{17}	1.0×10^{19}
Apparent thermal conductivity, 8 [g dm day^{-3} $^oK^{-1}$]	1.0×10^{18}	1.0×10^{18}	1.0×10^{18}
Bulk density, D_b [g dm^{-3}]	1200.0	1200.0	1200.0

both figures, the temperature distributions on the planes of z = 100 dm (i.e., Figures 4.12(a) and 4.13(a)) and y = 100 dm (i.e., Figures 4.12(b) and 4.13(b)) are emphasized. The temperature distribution is symmetric in the y-direction (with respect to y = 100 dm) as expected. However, it is asymmetric, even though the prescribed boundary conditions are symmetric, in the z-direction. This is caused by the composition of three materials for the simulated domain. From Figures 4.12(b) and 4.13(b), it is obvious that the heat transfer in material 2 (i.e., the lowest part) is quicker than that in material 3 (i.e., the highest part) due to the deviation in the specific heat of a dry medium (i.e., C_m). Material 3 has the largest C_m, which reflects the largest capacity of storing heat, among the three materials. As a result, the transfer of heat in material 3 is slower than in the other two materials.

4.2.3 Application 3: A Three-Dimensional Example of Chemical Transport

4.2.3.1   Problem Description

Like application 1, the simulated domain is composed of two materials: material 1 for the top half and material 2 for the bottom half. The necessary material parameters for the simulation of chemical transport are tabulated in Table 4.4. There are 7 component species, 19 complexed species, and 9 ion-exchanged species taken into account in this application. All the 28 product species associated with their stoichiometric coefficients and equilibrium constants are listed in Table 4.5. Among the 9 ion-exchanged species, 2 are involved in ion-exchange site 1 while 4 are in site 2 and 3 are in site 3 as marked in Table 4.5.
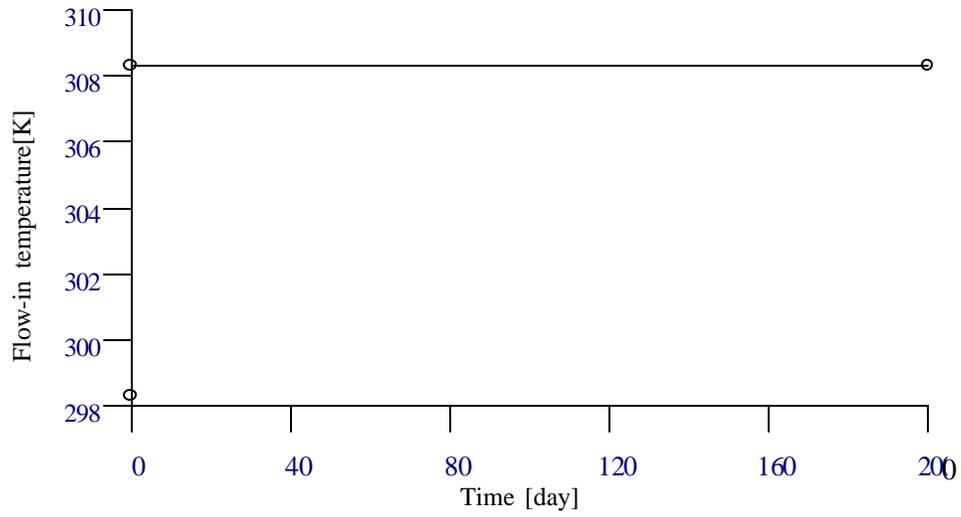
Figure 4.8 The time-dependent flow-in temperature for region A in application 2.
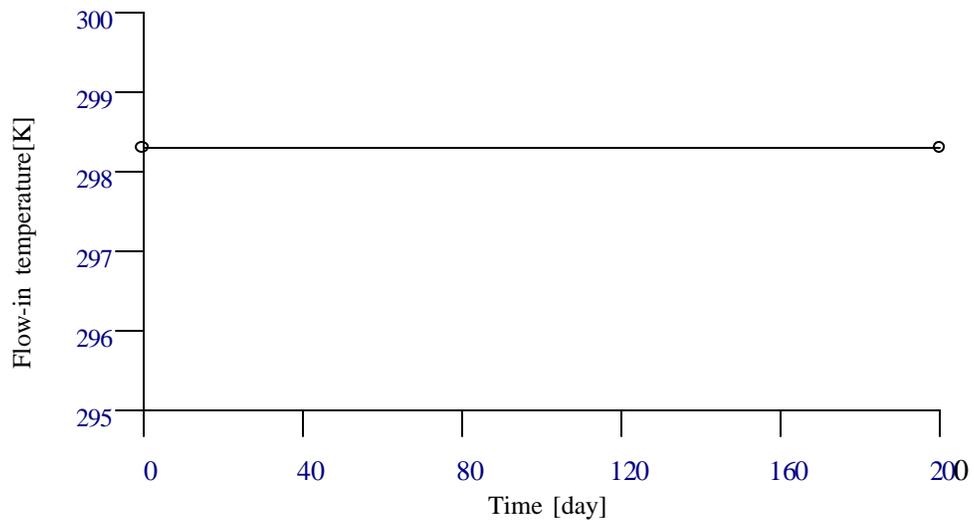
Figure 4.9 The time-dependent flow-in temperature for region B in application 2.
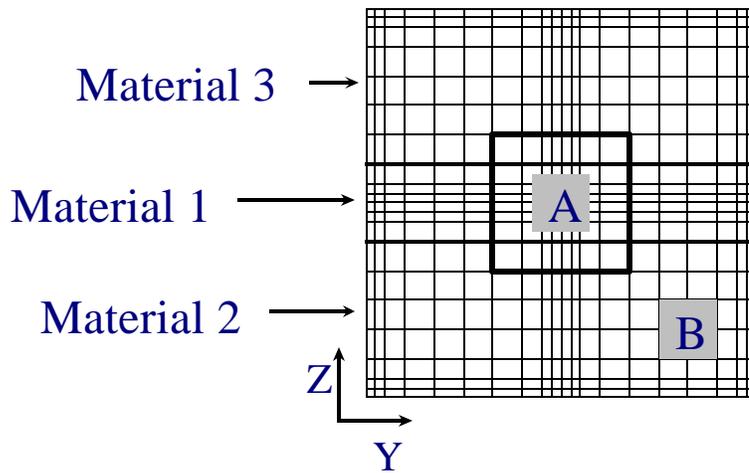
Figure 4.10 The two regions on the flow-in boundary in application 2.
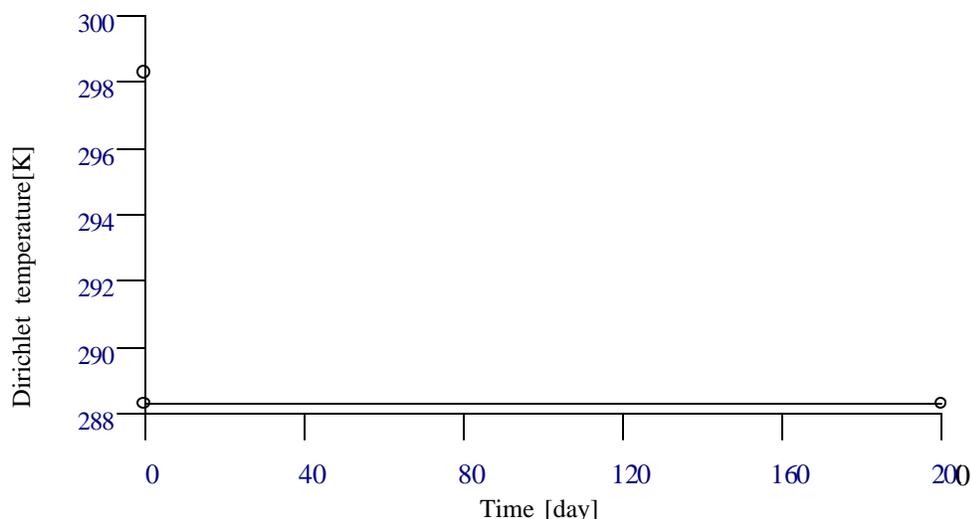
Figure 4.11 The time-dependent Dirichlet temperature in application 2.

The whole domain is assumed water-saturated and a one-dimensional constant subsurface flow of 0.3 dm/day in the x-direction passes through the simulated domain. The total analytical concentrations are specified at 0.001 M for components 1 (i.e., CO_3^{2-}) and 2 (i.e., Cl^-) and at 0.0005 M for components 3 (i.e., Ca^{2+}), 4 (i.e., Mg^{2+}), 5 (i.e., K^+), and 6 (i.e., Na^+) as the pre-initial condition for the steady-state simulation. Meanwhile, the pH-value is controlled at 9 throughout the entire domain during the simulation. The top, bottom, front, and back boundaries of the domain are considered barriers to isolate the simulated domain from the outside world. Hence a Neumann boundary condition of zero flux is implemented for these four boundaries. On the other hand, chemicals can be conducted into the domain through the left boundary and brought out from the domain through the right boundary by subsurface flow. Mathematically, a flow-in Variable boundary condition is applied to the left boundary while a flow-out Variable boundary condition is employed for the right boundary.

To simulate a cleanup process, the flow-in concentrations for components 1 through 6 at time = 0 day are given the same as the total dissolved component concentrations under the pre-initial condition, such that the steady-state solution will be identical to the given pre-initial condition. As soon as the transient-state simulation starts, a relatively clean groundwater is introduced through the left boundary into the simulated domain for a cleanup purpose. Thus, three time points, at time = 0, 10^{-6}, and 160 days (the transient simulation is 160 day long), are needed to describe the profile of the flow-in concentration for each of the first six components. The flow-in concentration for component 7 is not required because the pH-value is assumed to be kept at 9 through the simulation. The flow-in concentration is specified at 10^{-8} M at the second and the third time points (i.e., the total dissolved component concentration is assumed 10^{-8} M in the relatively clean groundwater) in order to investigate a cleanup process in the transient-state simulation herein. At time = 0 day, the flow-in concentrations are 0.001 M for components 1 and 2, 0.00039383252812 M for component 3, 0.00037542367361 M for component 4, 0.000487308065744 M for component 5, and 0.00049417953145 M for component 6. With these given flow-in concentrations, the associated steady-state solution will have the total analytical component concentrations the same as those of the pre-initial condition. This is because these flow-in concentrations are exactly the total dissolved component concentrations under the prescribed pre-initial condition. It is so designed to easily examine if the steady-state solution can be computed within a given error tolerance as well as to provide a simple initial condition for the following transient-state simulation. One thing that should be noted is that components 1 and 2 are actually conservative mobile components because they are not involved in any solid phase. This is why their flow-in concentrations are the same as their pre-initial conditions.

The time-step size is 4 days. Forty time steps are simulated to achieve a transient simulation of 160 days. A relative error tolerance of 5×10^{-4} is utilized to determine convergency in both the steady-state and the transient-state simulations.

4.2.3.2   Simulation Results

As mentioned previously, a convergent steady-state solution, close to the pre-initial condition within a prescribed error tolerance, should be obtained. In other words, the total analytical concentrations are uniformly distributed at 0.001 M for components 1, 2 and at 0.0005 M for components 3 to 6. Figures 4.14 through 4.23 show the transient simulation results, with a focus on the plane of y = 100 dm, by using the convergent steady-state solution as the initial

(a)

(b)
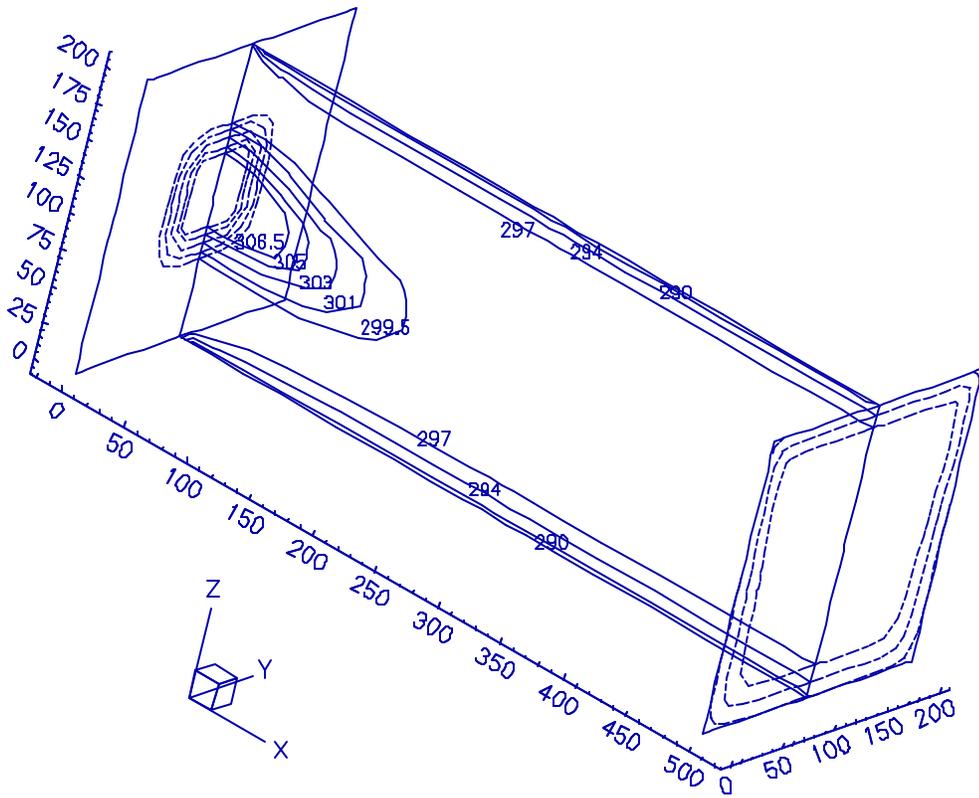
Figure 4.12 The temperature distribution result at time = 100 days in application 2.
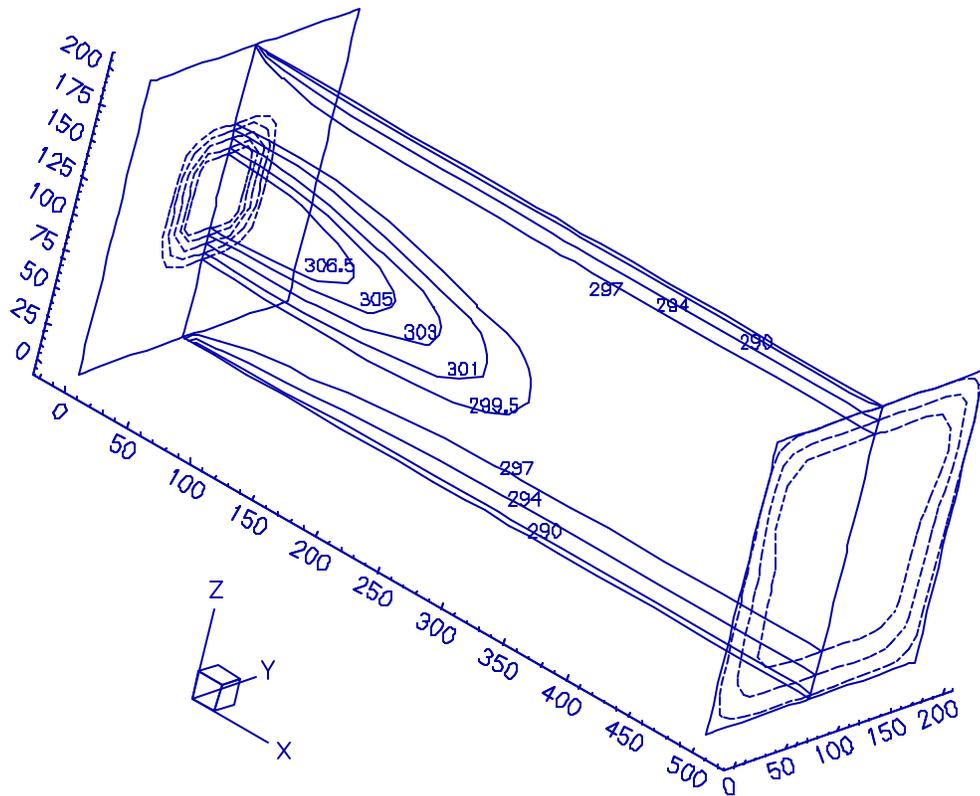
(a)
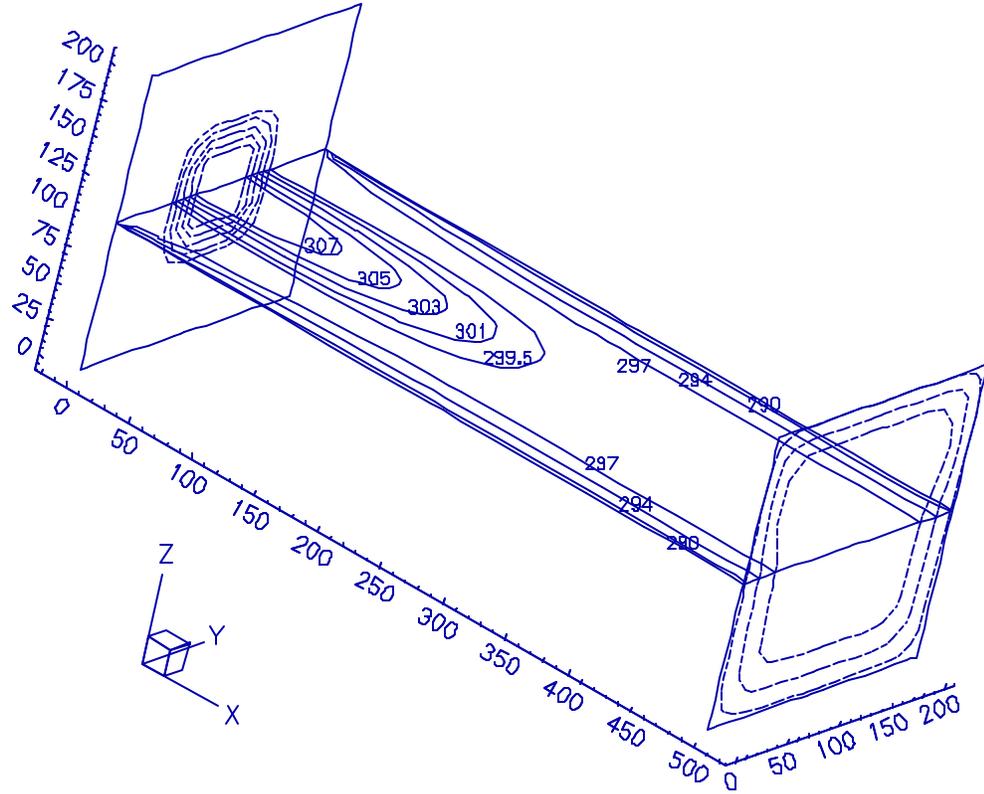
(b)

Figure 4.13 The temperature distribution result at time = 200 days in application 2.

Table 4.4 Material parameters needed for the simulation in application 3.

Material parameter	Material 1	Material 2
Effective porosity, θ_{sat} [dimensionless]	0.3	0.3
Bulk density, D_b [g dm^{-3}]	1200.0	1200.0
Longitudinal dispersivity, α_L [dm]	1.0	10.0
Transverse dispersivity, α_T [dm]	0.1	1.0
Diffusion coefficient, α_m [dm^2 day^{-1}]	0.0001	0.001
Tortuosity, τ [dimensionless]	1.0	1.0
Capacity of ion-exchange site 1 [eq g^{-1}]	1.0×10^{-8}	2.0×10^{-8}
Capacity of ion-exchange site 2 [eq g^{-1}]	2.0×10^{-8}	4.0×10^{-8}
Capacity of ion-exchange site 3 [eq g^{-1}]	3.0×10^{-8}	6.0×10^{-8}

condition. Among these figures, Figures 4.14, 4.16, 4.18, 4.20, and 4.22 display the concentration distributions at time = 80 days, while the others are for time =160 days.

As it can be checked with Table 4.5, both components 1 (i.e., CO_3^{2-}) and 2 (i.e., Cl^-) are involved in the species existing only in the aqueous phase. They are, therefore, considered as conservative mobile components. At any moment and any location, their total analytical concentrations are equal to their total dissolved concentrations. Because an identical setting of the initial and boundary conditions is applied to these two components, the total analytical concentration distribution for these two components will be identical. Figures 4.14 and 4.16 present the distribution of the total analytical (or dissolved) concentrations of component 1 (or 2). It is observed that more dispersion is in the lower half of plane y = 100 dm. This is because the lower half (i.e., material 2) has dispersivities and a diffusion coefficient ten times that of those of the upper half (i.e., material 1). Symmetric concentration distributions in the y-direction, with respect to y = 100 dm can also be observed on the plane of z = 0 dm in both figures due to (1) the uniform initial condition, (2) the symmetric boundary condition in the y-direction, and (3) the invariant material type in the y-direction. For the same reason, the concentration distribution of any component herein is symmetric in the y-direction (Figures 4.16 through 4.23).

In this application, ion-exchanged Na is employed as the referenced ion-exchanged species for the three ion-exchange sites. Three ion-exchanged Na's (one for each site) are taken into account for the convenience in the formulation of computing chemical equilibrium. With a close examination of the data of ion-exchanged species in Table 4.5, it can be seen that ion-exchanged Ca(1) and ion-exchanged Ca(2), in fact, are involved in an identical ion-exchange reaction in this hypothetical example. The same situation happens to the two ion-exchanged Mg's, the two ion-exchanged K's, and the three ion-exchanged Na's. One thing that should be noted is that the setting on those ion-exchanged species herein may not be realistic, however, a problem of multiple ion-exchange sites can be handled mathematically through the demonstration of this application. In practice, one may have a equilibrium constant for ion-exchanged Ca(1) different from that for ion-exchanged Ca(2) when (1) different referenced ion-exchanged species are assigned for sites 1 and 2 or (2) the two sites are naturally different from a chemical point of view.

Two parts, (a) and (b), are included to show the distributions of the total analytical and the total dissolved concentrations, respectively, in each of Figures 4.16 through 4.23, as indicated in those figures. Figures 4.16 and 4.17 are for component 3 (i.e., Ca^{2+}), while Figures 4.18 and 4.19 are for component 4 (i.e., Mg^{2+}), Figures 4.20 and 4.21 are for component 5 (i.e., K^+), and Figures 4.22 and 4.23 are for component 6 (i.e., Na^+). The (b) part of these figures, similar to Figures 4.14 and 4.15, shows the expansion of the total dissolved component concentration mainly due to dispersion/ diffusion. As it might be imagined, if the total dissolved component concentration, rather than the total analytical component concentration, is chosen as the primary dependent variable (PDV), then the associated primary governing equation (PGE) [Yeh and Tripathi, 1990] is a typical advection-dispersion/diffusion equation with the concentration change in the solid phase considered as sources/sinks. Hence a dispersion phenomenon can be expected.

Generally speaking, all the figures of the total analytical concentration (i.e., the (a) part) tell that the upper half can be cleaned up more efficiently than the lower half. This is simply because the upper half (or material 1) contains ion-

Table 4.5 Chemistry data for the simulation in application 3 (to be cont.).

Product species	Stoichiometric coefficient							Log_{10} K
	CO_3^{2-}	Cl^-	Ca^{2+}	Mg^{2+}	K^+	Na^+	H^+	
CaOH	0	0	1	0	0	0	-1	-12.700
$CaCO_3$	1	0	1	0	0	0	0	3.153
$CaHCO_3$	1	0	1	0	0	0	1	11.430
$CaCl^+$	0	1	1	0	0	0	0	0.080
OH^-	0	0	0	0	0	0	-1	-14.000
HCO_3^-	1	0	0	0	0	0	1	10.330
H_2CO_3	1	0	0	0	0	0	2	16.680
$MgOH^+$	0	0	0	1	0	0	-1	-11.200
$MgHCO_3^+$	1	0	0	1	0	0	1	11.340
$MgCO_3$	1	0	0	1	0	0	0	-2.920
$NaCO_3^-$	1	0	0	0	0	1	0	-1.270
$NaHCO_3$	1	0	0	0	0	1	1	-10.080
$Mg_4(OH)_4^{4+}$	0	0	0	4	0	0	-4	-39.660
$MgCl^+$	0	1	0	1	0	0	0	-0.100
NaOH	0	0	0	0	0	1	-1	-13.800
Na_2CO_3	1	0	0	0	0	2	0	0.680
NaCl	0	1	0	0	0	1	0	-0.900
KOH	0	0	0	0	1	0	-1	-14.500
KCl	0	1	0	0	1	0	0	-0.700
Ion-exchanged $Ca(1)^+$	0	0	1	0	0	0	0	0.540
Ion-exchanged $Ca(2)^+$	0	0	1	0	0	0	0	0.540
Ion-exchanged $Mg(2)^+$	0	0	0	1	0	0	0	-0.200
Ion-exchanged $Mg(3)^+$	0	0	0	1	0	0	0	-0.200
Ion-exchanged $K(2)^+$	0	0	0	0	1	0	0	0.390
Ion-exchanged $K(3)^+$	0	0	0	0	1	0	0	0.390
Ion-exchanged $Na(1)^+$	0	0	0	0	0	1	0	0.000
Ion-exchanged $Na(2)^+$	0	0	0	0	0	1	0	0.000
Ion-exchanged $Na(3)^+$	0	0	0	0	0	1	0	0.000

(I) indicates the ion-exchange site that the ion-exchanged species is involved. For instance, Ion-exchanged Ca(2) represents that this ion-exchanged calcium is involved in the second ion-exchange site.
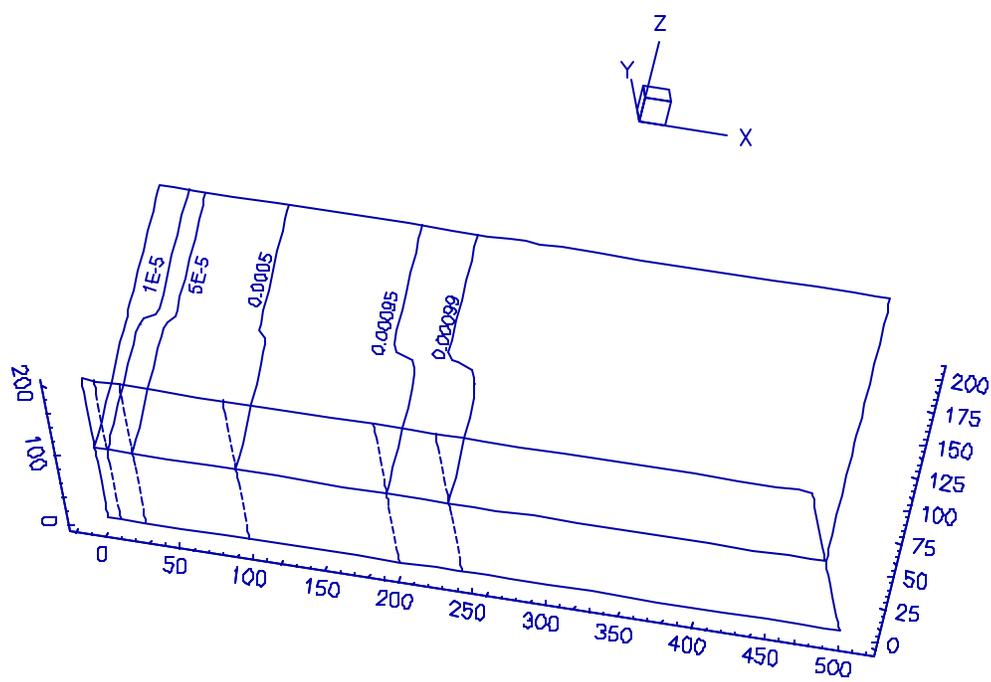
Figure 4.14 The distribution of the total analytical (or dissolved) concentration of component 1 (CO_3^{2-}) or 2 (Cl^-) at time = 80 days in application 3.
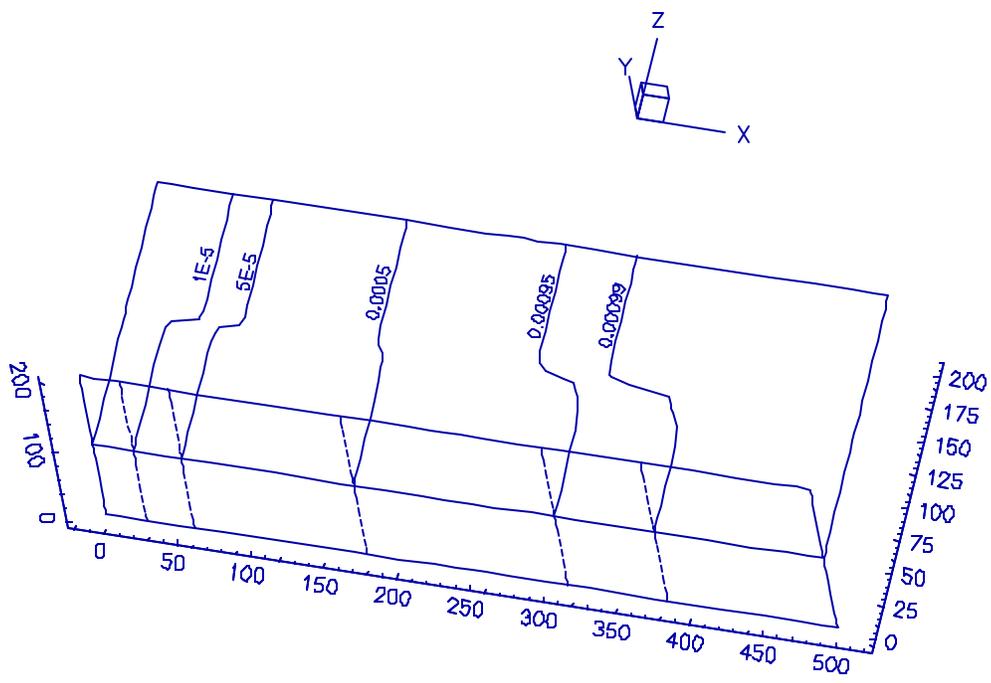
Figure 4.15 The distribution of the total analytical (or dissolved) concentration of component 1 (CO_3^{2-}) or 2 (Cl^-) at time = 160 days in application 3.

(a)
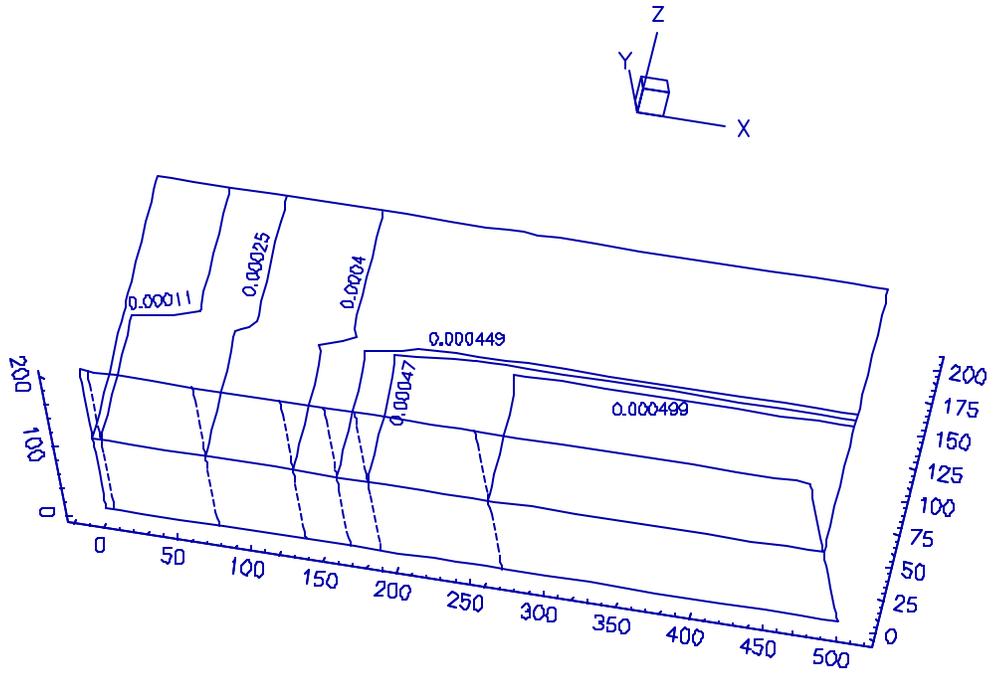
(b)
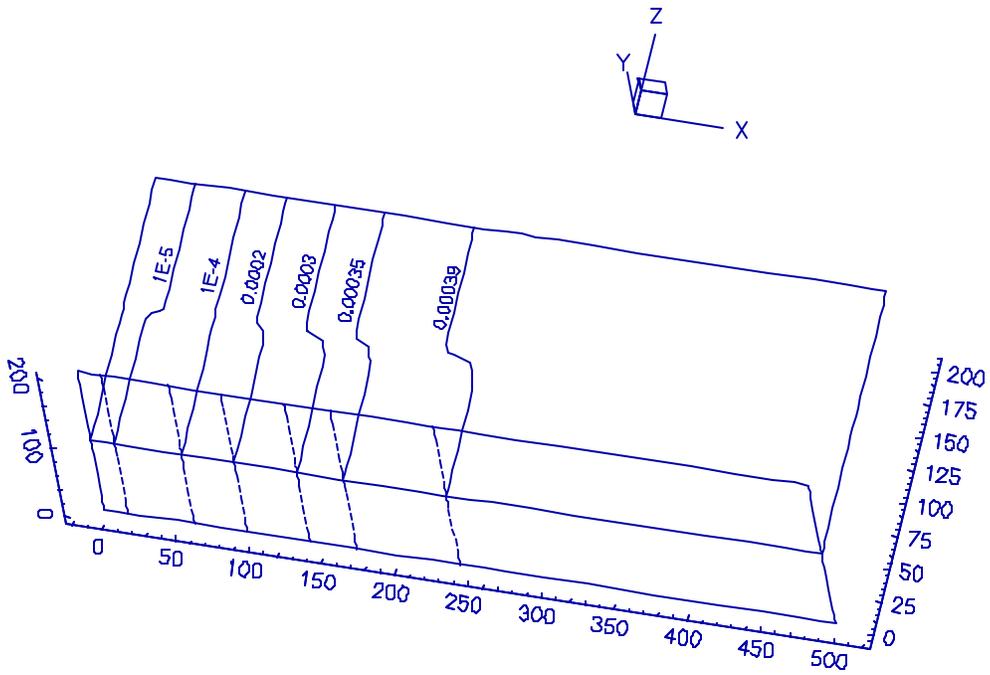
Figure 4.16   The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 3 (Ca^{2+}) at time = 80 days in application 3.
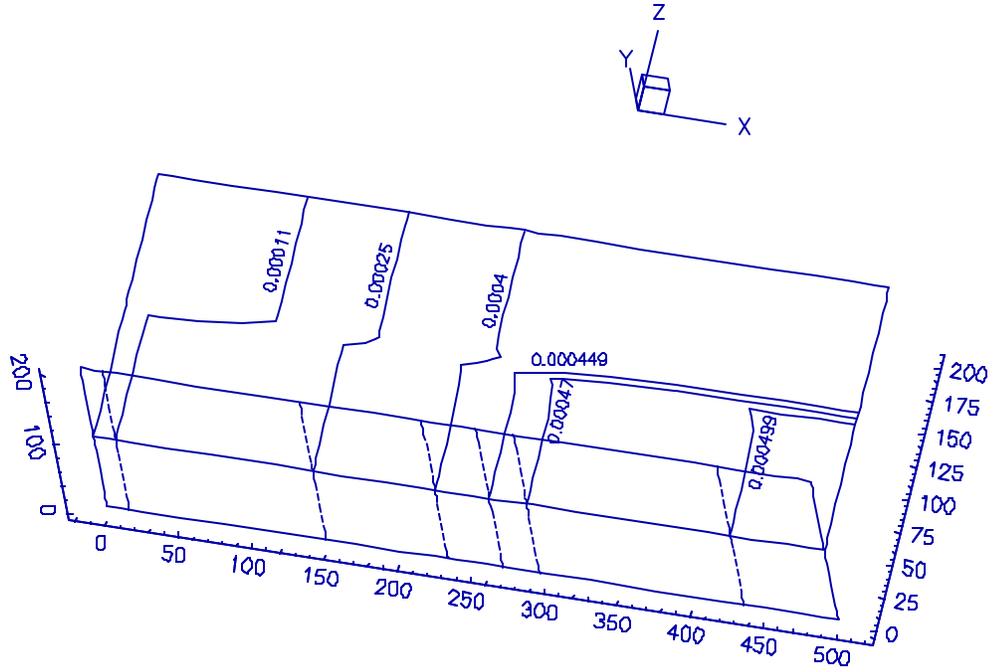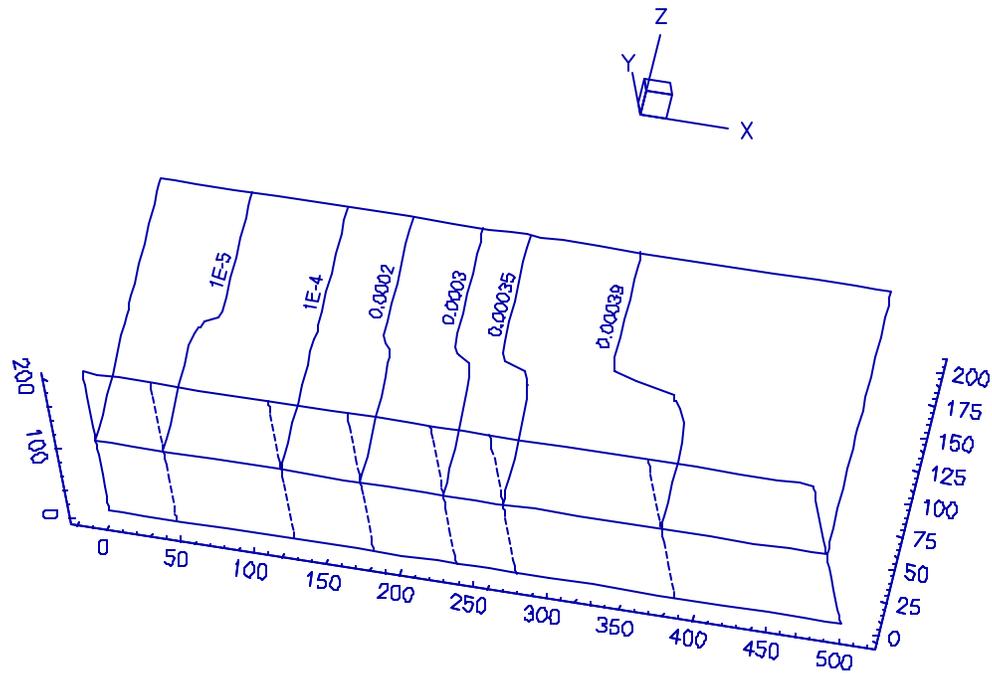
Figure 4.17 The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 3 (Ca^{2+}) at time = 160 days in application 3.

(a)
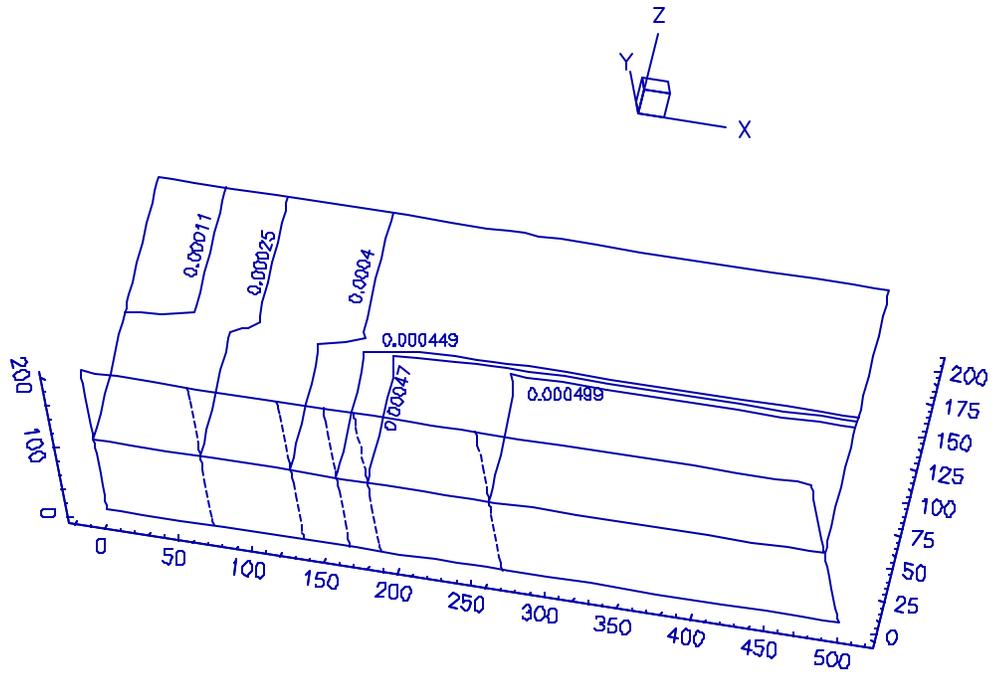
(b)
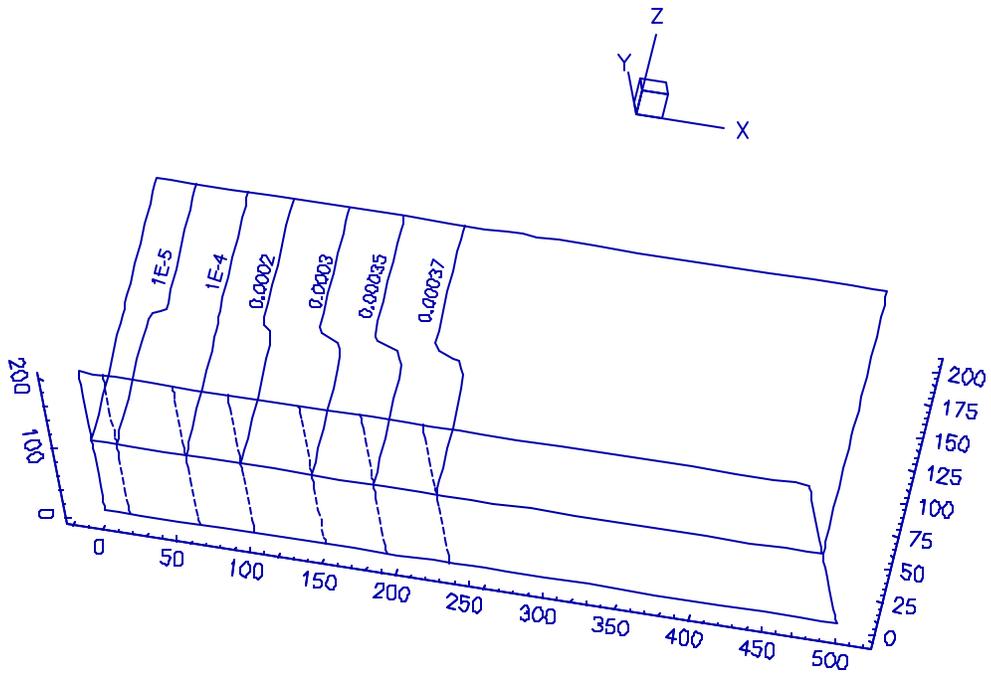
Figure 4.18 The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 4 (Mg^{2+}) at time = 80 days in application 3.

(a)
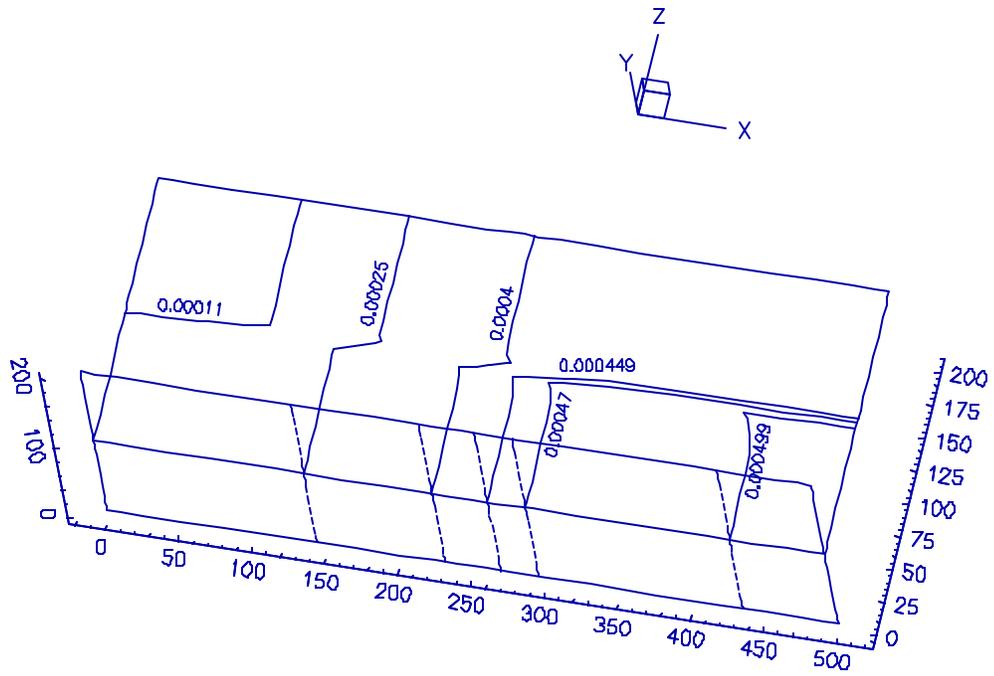
(b)
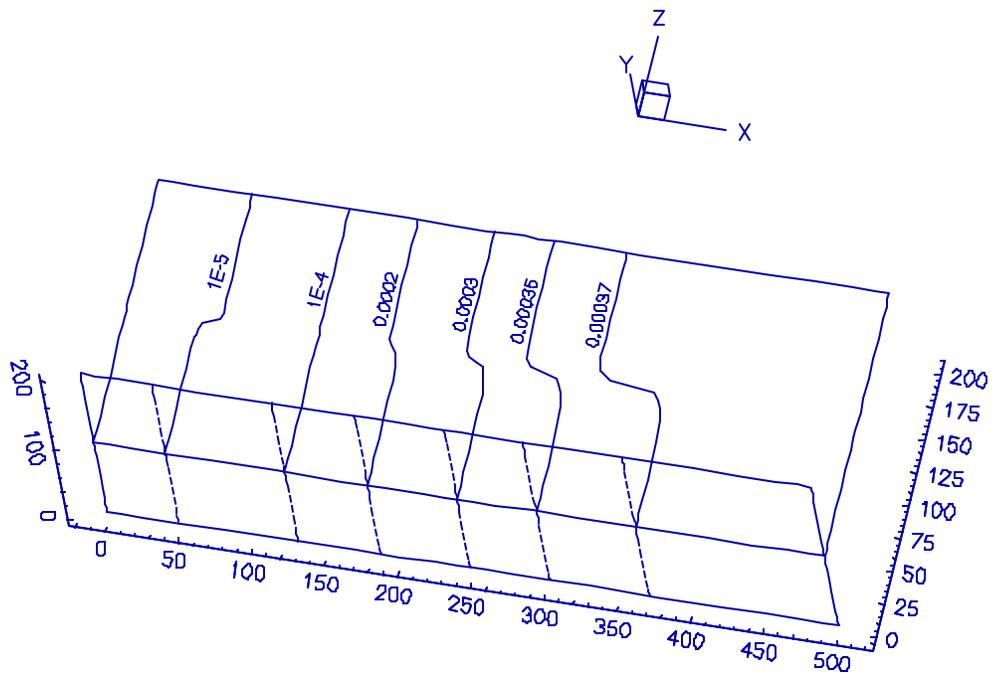
Figure 4.19 The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 4 (Mg^{2+}) at time = 160 days in application 3.
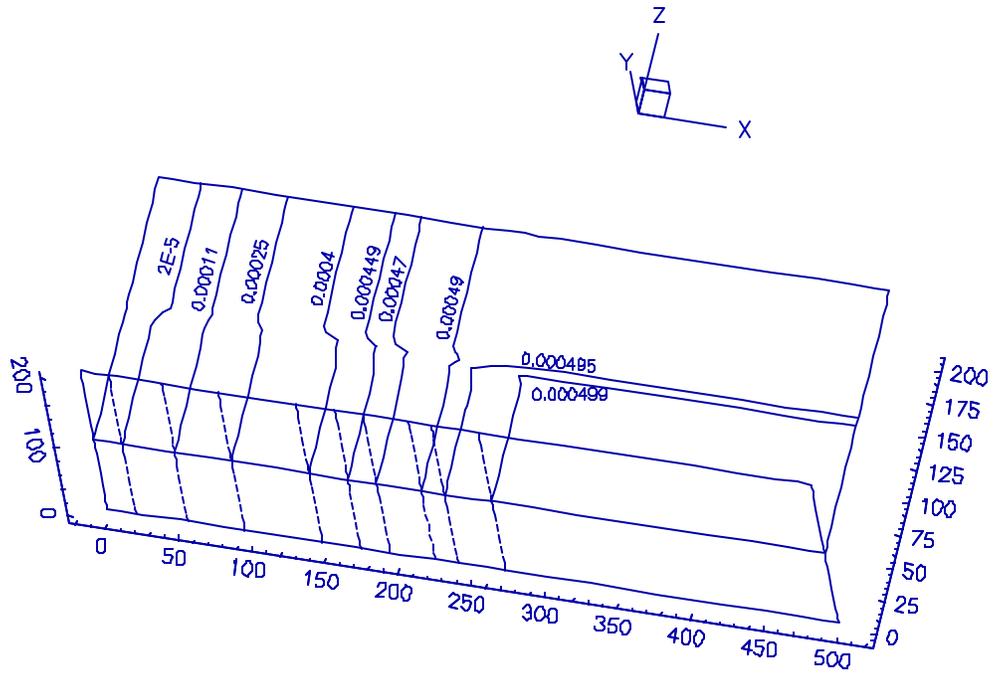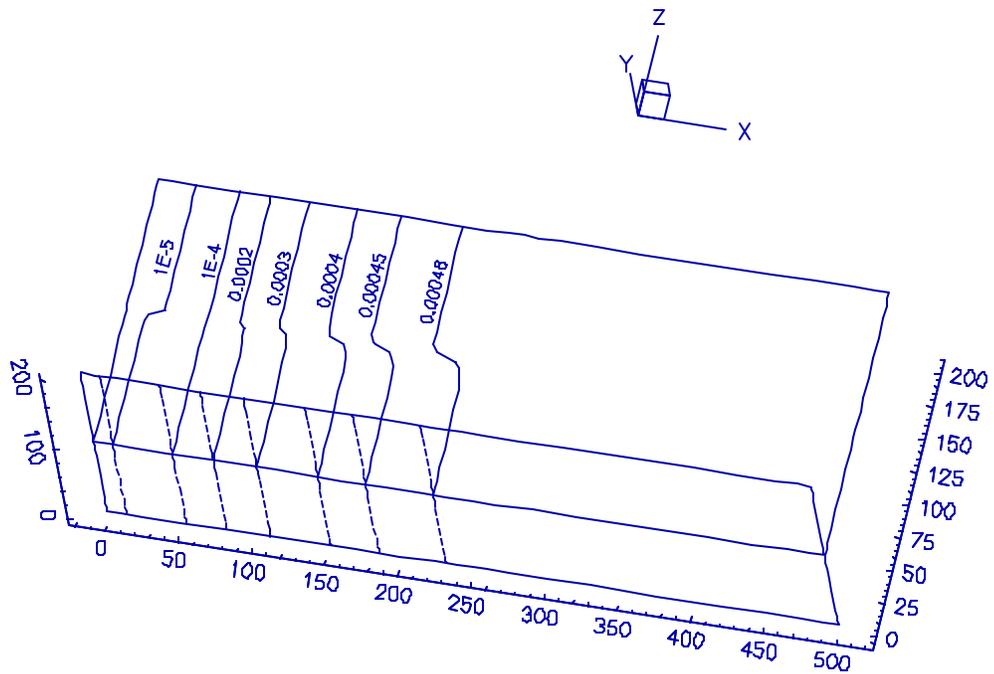
Figure 4.20 The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 5 (K⁺) at time = 80 days in application 3.
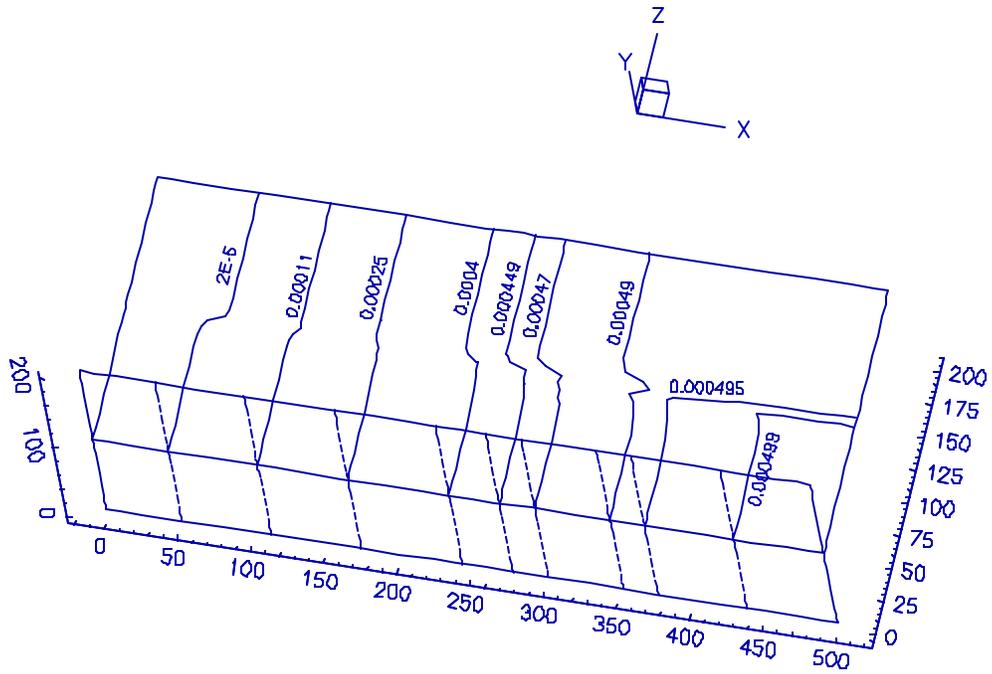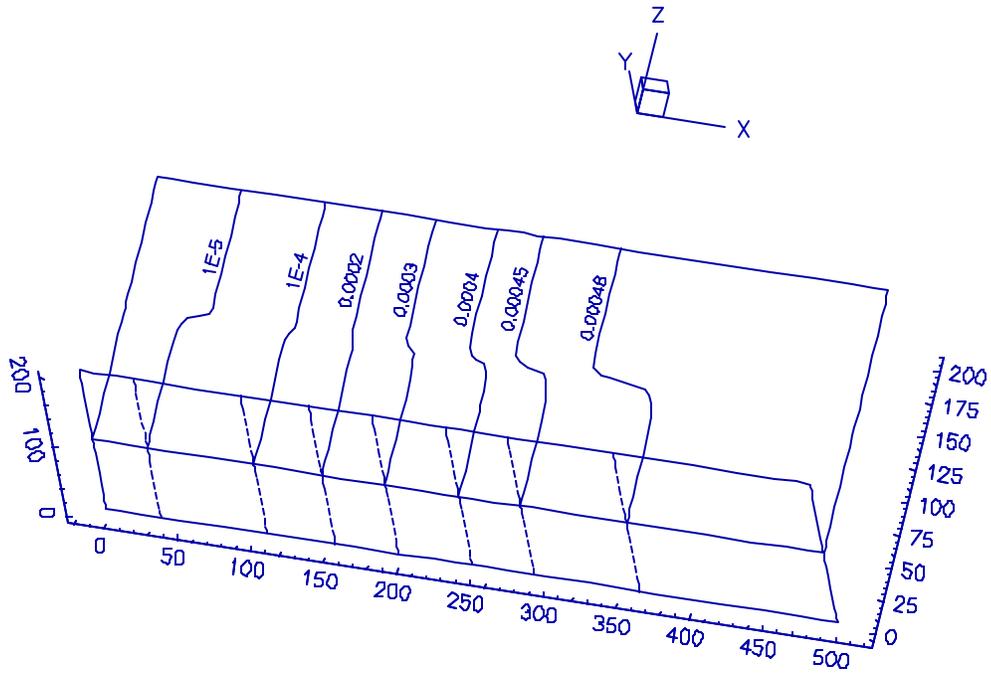
Figure 4.21 The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 5 (K+) at time = 160 days in application 3.

(a)
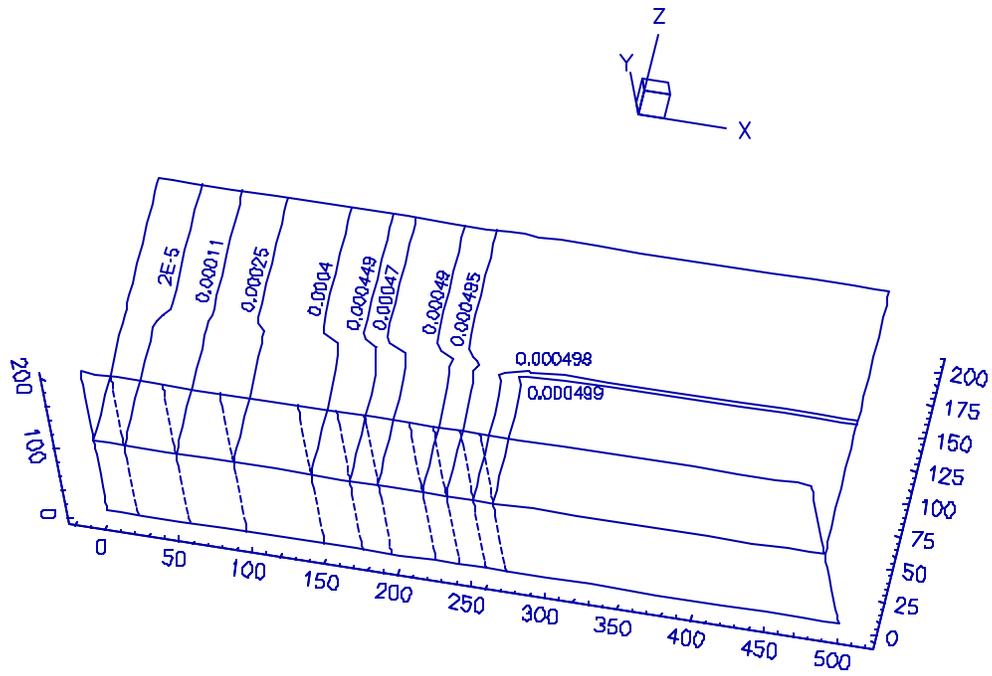
(b)
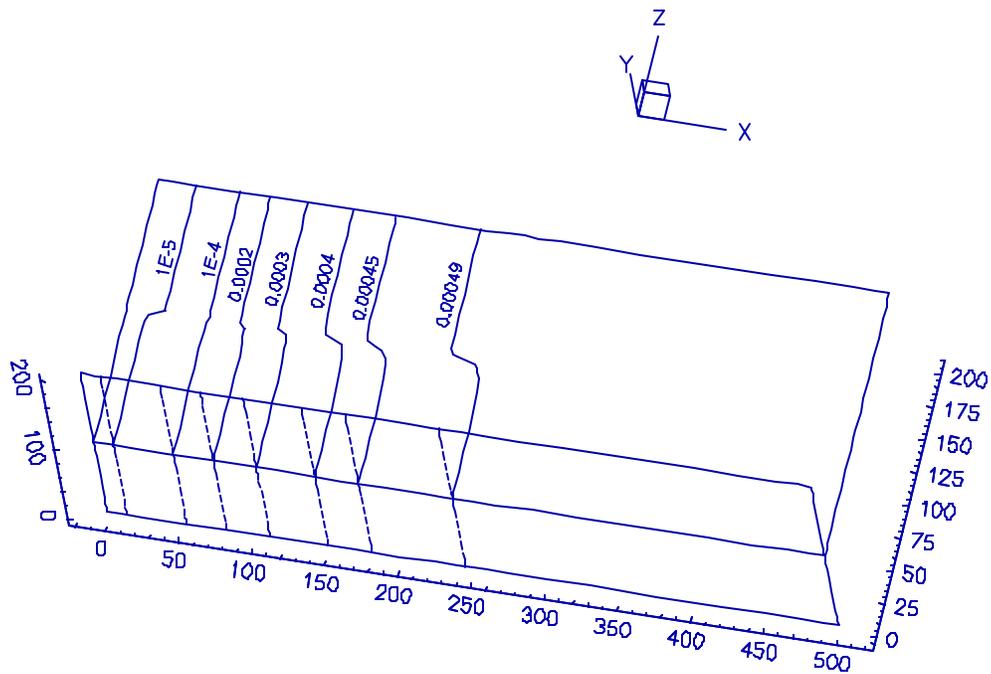
Figure 4.22   **The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 6 (Na⁺)at time = 80 days in application 3.**

(a)
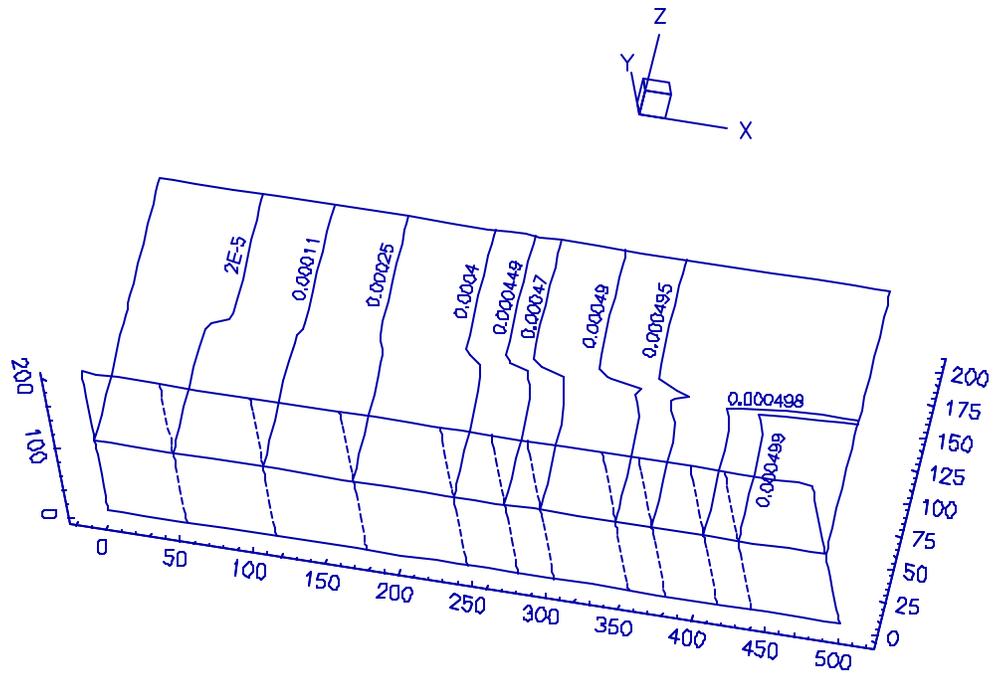
(b)
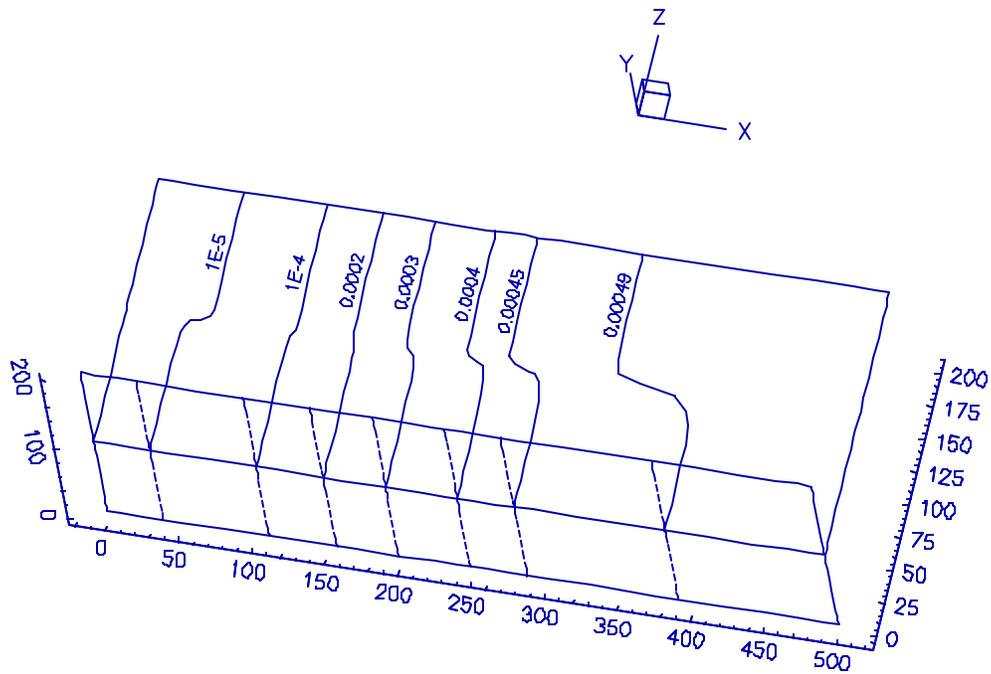
Figure 4.23 The distributions of (a) the total analytical and (b) the total dissolved concentrations of component 6 (Na⁺) time = 160 days in application 3.

exchange capacities just one half of those of the lower half (or material 2). Thus, a smaller amount of ion-exchanged species is stored in the solid phase of the upper half than that of the lower half according to the requirement that an ion-exchange site has to be fully occupied. This requirement also implies that a certain amount of chemicals will still stay in the solid phase no matter how much clean water is used to achieve cleanup. Based on this, a greater amount of ion-exchange species will be found in the lower half than in the upper half after no more chemicals can be washed out from the solid phase.

On the other hand, however, it is observed that the 0.0004 M, 0.000449 M, 0.00047 M, and 0.00049 M concentration lines of component 5 (Figures 4.20(a) and 4.21(a)) in the upper half are behind those in the lower half. The same situation also happens to component 6 (Figure 4.22(a) and 4.23(a)). This is mainly because the chemical system herein allows a relatively small amount of ion-exchanged K and ion-exchanged Na stored in the solid phase within the region where these concentration lines are included. In other words, the total dissolved concentration of components K^+ and Na^+ are much higher than the associated total sorbed concentrations which are contributed by ion-exchanged species. This can be roughly verified by comparing the concentration line of 0.0004 M in parts (a) and (b) of Figures 4.20 through 4.23. As a result, the distributions of the total analytical concentration of components K^+ and Na^+ are similar to the associated total dissolved concentration distributions in that region.

According to the given equilibrium constants of ion-exchanged Ca (i.e., $10^{0.54}$) and ion-exchanged Mg (i.e., $10^{-0.20}$), the contrast between the concentration distributions of components Ca^{2+} and Mg^{2+} should be obvious. However, their concentration difference is limited based on Figures 4.17 to 4.20. To explain this, two main reasons are provided: (1) ion-exchanged Mg is involved in site 3, which has the largest capacity among the three sites, whereas ion-exchanged Ca participates in only sites 1 and 2; (2) this chemical system considers not only ion-exchange reactions but also complexation reactions. The combination of these reactions makes the system difficult to be analyzed by just examining stoichiometric coefficients and equilibrium constants. Reason (1) indicates a significant amount of ion-exchanged Mg stored in the solid phase because of the lack of ion-exchanged Ca to compete in site 3. With a FORTRAN debugger, the concentration of ion-exchanged Mg is determined to be about one fifth of the concentration of ion-exchanged Ca in site 2 (i.e., [ion-exchanged Mg (2)] 0.2 [ion-exchanged Ca (2)]). Hence this reason is supported. Reason (2) points out the need of numerical models to perform quantitative analyses nowadays.

4.2.4 Application 4: A Three-Dimensional Coupled Example of Subsurface Flow, Heat Transfer, and Chemical Transport

4.2.4.1   Problem Description

This application is used to demonstrate the coupling among subsurface flow, heat transfer, and chemical transport. The required universal parameters for the associated simulation are listed in Table 4.6. For computing the temperature-dependent groundwater density, the following equation is employed in this example problem.

$$r_w(T) = a_1 + a_2 T + a_3 T^2 + a_4 T^3 \tag{4.4}$$

where $r_w(T)$ is the temperature-dependent pure water density associated with temperature T; a_1, a_2, a_3, and a_4 are temperature-density relationship parameters. Eq. (4.4) is obtained by curve-fitting the data provided in "CRC Handbook of Chemistry and Physics" [Lide, 1992]. Figure 4.24 displays the curve-fitting result. One thing that should be noted is that the referenced groundwater density r_o (= 997.04882 g/dm³) is associated with the referenced temperature T_o (= 298.3 °K). Two materials are included: material 1 for the upper half and material 2 for the lower half, as pictured earlier by Figure 4.2. Material parameters needed for the simulation are given in Table 4.7.

In computing subsurface flow, Eqs. (4.1) through (4.3) are used to describe the soil properties. The temperature-dependent equilibrium constant is calculated according to the van't Hoff equation. Table 4.8 lists chemistry information needed to compute chemical equilibrium for a simplified closed carbonate system that is considered in this application. The ionic-strength effect is taken into account by utilizing the extended Debye-Hückel formula.

Given well-designed pre-initial and boundary conditions, the steady-state simulation can provide a computational result that presents a uniformly-polluted domain under an isothermal and nearly hydrostatic condition. With this result as the initial condition, the temperature effect on a cleanup process can be investigated by activating a pumping well in the middle of the domain and controlling the flow-in groundwater with different temperatures. The detailed settings are described as follows.

The pre-initial conditions are given 200 dm for the total head, 298.3 °K for the temperature, and 4×10^{-4} M for the total analytical concentrations of both Ca^{2+} and CO_3^{2-} throughout the entire domain. The pH-value is kept at 10. The top, bottom, front, and back boundaries are assumed impermeable for groundwater, heat, and chemicals to pass through. To perform the steady-state solution as mentioned above, a Dirichlet boundary condition of 200 dm is specified for the

Table 4.6 Universal parameters needed for the simulation in application 4.

Universal parameter	
Referenced groundwater density, D_o [g dm^{-3}]	997.04882
Referenced groundwater dynamic viscosity, μ_o [g dm^{-1} day^{-1}]	7.9056×10^5
Referenced temperature, T_o [^{o}K]	298.3
Specific heat of groundwater, C_l [dm^2 day^{-1} ^{o}K^{-1}]	3.12035×10^{15}
Density-temperature relationship parameter 1, a_1 [dimensionless]	-410.174
Density-temperature relationship parameter 2, a_2 [dimensionless]	13.2296
Density-temperature relationship parameter 3, a_3 [dimensionless]	-0.0403921
Density-temperature relationship parameter 4, a_4 [dimensionless]	3.97476×10^{-5}
Ideal gas constant, R [g dm^2 day^{-1} ^{o}K^{-1} mole^{-1}]	6.2067×10^{15}
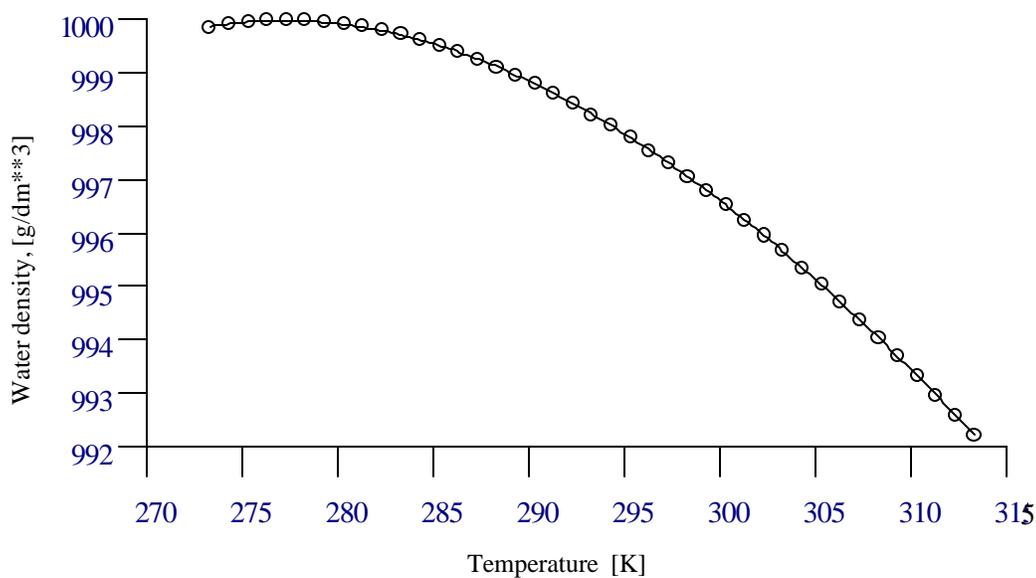
Figure 4.24   The curve-fitting result of the temperature-dependent water density.

Table 4.7 Material parameters needed for the simulation in application 4.

Material parameter	Material 1	Material 2
Saturated K_{xx} [dm day^{-1}]	1.0	5.0
Saturated K_{yy} [dm day^{-1}]	1.0	5.0
Saturated K_{zz} [dm day^{-1}]	1.0	5.0
Saturated K_{xy} (or K_{xy}) [dm day^{-1}]	0.0	0.0
Saturated K_{xz} (or K_{xz}) [dm day^{-1}]	0.0	0.0
Saturated K_{yz} (or K_{yz}) [dm day^{-1}]	0.0	0.0
Effective porosity, 2_{eff} [dimensionless]	0.3	0.2
Soil property parameter 1, 2_a [dimensionless]	0.15	0.1
Soil property parameter 2, h_a [dm]	-1000.0	-1000.0
Specific heat of dry medium, C_m [dm^2 day^{-1} ^{o}K^{-1}]	1.0×10^{14}	2.0×10^{14}
Apparent thermal conductivity, 8 [g dm day^{-3} ^{o}K^{-1}]	2.0×10^{18}	2.0×10^{18}
Bulk density, D_b [g dm^{-3}]	1200.0	1200.0
Longitudinal dispersivity, "$_L$ [dm]	1.0	2.0
Transverse dispersivity, "$_T$ [dm]	0.1	0.2
Diffusion coefficient, "$_m$ [dm^2 day^{-1}]	0.0001	0.0002
Tortuosity, J [dimensionless]	1.0	1.0

Table 4.8 Chemistry data for the simulation in application 4.

Product species	Stoichiometric coefficient			Log$_{10}$ K	Reaction enthalpy*
	Ca^{2+}	CO_3^{2-}	H^+		
OH^-	0	0	-1	-14.000	4.167690×10^{19}
HCO_3^-	0	1	1	10.200	-1.112280×10^{19}
H_2CO_3	0	1	2	16.500	-1.171028×10^{20}
$Ca(OH)_{2(s)}$	1	0	-2	-21.900	1.256353×10^{19}
$CaCO_{3(s)}$	1	1	0	8.300	9.353595×10^{18}

The unit of reaction enthalpy is [g dm^2 day^2 mole^{-1}] ; where 10 [kJ mole^{-1}] = 7.46496×10^{18} [g dm^2 day^2 mole^{-1}]

referenced total head on both the right and left boundaries while Variable boundary conditions are applied to those two boundaries for heat transfer and chemical transport. The flow-in temperature is 298.3 ^{o}K. The flow-in concentration is 1.2247061×10^{-4} M for both Ca^{2+} and CO$_3^{2-}$, where 1.2247061×10^{-4} M is the total dissolved concentration for the two components at equilibrium when 4×10^{-4} M is the total analytical component concentration. Due to the density effect, a truly hydrostatic situation cannot be achieved. However, a nearly hydrostatic condition can be performed because the density effect is not important.

When the transient-state simulation begins, three point sinks are used to simulate a pumping well for a cleanup purpose. These three point sinks, with a pumping rate of 5×10^3 dm^3/day for each, are at (250, 100, 80), (250, 100, 90), and (250, 100, 95). It is thus imaginable that groundwater will flow into the domain through the left and the right boundaries. Relative clean groundwater, with a total dissolved concentration of 10^{-8} M for components 1 and 2, is assumed to supply this flow-in groundwater. A temperature of 308.3 ^{o}K is associated with the flow-in groundwater through the left boundary, whereas the flow-in groundwater through the right boundary is still under the temperature of 298.3 ^{o}K. To put a focus on the temperature effect, the referenced total head of 200 dm is kept as in the steady-state simulation, so that the subsurface flow will be almost symmetric with respect to X = 250 dm (i.e., the middle of the domain in the x-direction). It is thus adequate, directly based on the computational results, to examine how significantly temperature can affect chemical transport.

Forty time steps are included in the transient-state simulation, where 1, 2, and 4 days are set to be time-step sizes of the first, second, and third through fortieth time steps, respectively. 5×10^{-5} dm is assigned as the absolute error tolerance for a convergent solution of subsurface flow. In the meantime, 10^{-3} and 5×10^{-5} are used to be the relative error tolerances in computing for chemical concentrations and temperature, respectively.

4.2.4.2   Simulation Results

As mentioned above, the steady-state solution is controlled to be almost identical to the pre-initial condition. That is, the simulated domain has chemical concentrations uniformly distributed under an isothermal and a nearly hydrostatic condition at the beginning of the transient-state simulation.

With a pumping rate of 5×10^3 dm^3/day assigned to the three point sinks in the middle of the domain, a flow field, symmetric in the x-direction with respect to x = 100 dm, can be observed when identical boundary conditions are applied to both the left and the right boundaries. In this application, however, a temperature of 308.3 ^{o}K is given to the incoming subsurface flow from the left boundary, which is 10 ^{o}K higher than that initially set for the simulated domain or that of the incoming subsurface flow from the right boundary. Thus, the groundwater density will not be symmetrically distributed in the x-direction. Based on the fact that the possible highest temperature is 308.3 ^{o}K and the possible maximum total analytical concentrations are 4×10^{-4} M for component 1 (i.e., Ca^{2+}), 2 (i.e., CO$_3^{2-}$) and less than 10^{-4} M for component 3 (i.e., H$^+$) through the whole simulation, the influence of temperature and chemical concentration on the groundwater density can be as large as 0.35% (according to Figure 4.24) and 0.004% (according to Eq. 2.35(b)) during the transient-state simulation.

Figures 4.25 and 4.26 present the total head distributions at time = 75 days and 155 days, respectively. The (a) part of these two figures puts a focus on planes y = 100 dm, y = 200 dm, and z = 100 dm, while part (b) concentrates on the planes of z = 50 dm, z = 100 dm, z = 150 dm, and y = 100 dm. A symmetry in the y-direction and an asymmetry in the z-direction can be seen as expected. Although the symmetry in the x-direction looks true, it does not exist actually due to the groundwater density effect mainly caused by a higher incoming temperature from the left boundary. Figure 4.27 plots the flow rate on both the left and the right boundaries at time = 75 days (part (a)) and 155 days (part (b)). Again, the magnitudes of the flow rates on the two boundaries cannot be distinguished visually. However, the computational result does provide values to tell the difference. On the other hand, since the groundwater density effect is so minor in this case that the difference mentioned above might be ignored, the subsurface flow could be assumed symmetric in the x-direction. The temperature-dependent cleanup process will be discussed under this assumption.

Figure 4.28 shows the temperature distribution at time = 75 days with emphases on (a) plane z = 100 dm and (b) plane y = 100 dm. Likewise, Figure 4.29 plots the temperature distribution at time = 155 days. From Figure 4.28, only the isothermal line of 299 ^{o}K implicitly indicates a possible existence of pumping wells. The pumping, nevertheless, becomes obvious in Figure 4.29 because a longer simulation time allows heat to be transferred from the left boundary to the neighborhood of pumping wells, where the flow field is greatly dominated by pumping.

Figures 4.30 and 4.31 depict the computational results of simulating a cleanup process by pumping at time = 75 days and 155 days, respectively. In these two figures, part (a) is used to emphasize the total analytical concentration distribution on the planes of z = 65 dm and 135 dm, while part (b) is used to describe the distribution of the total analytical concentration on planes y = 0 dm and y = 100 dm. Due to the fact that Ca(OH)$_{2(s)}$ does not exist under the competition
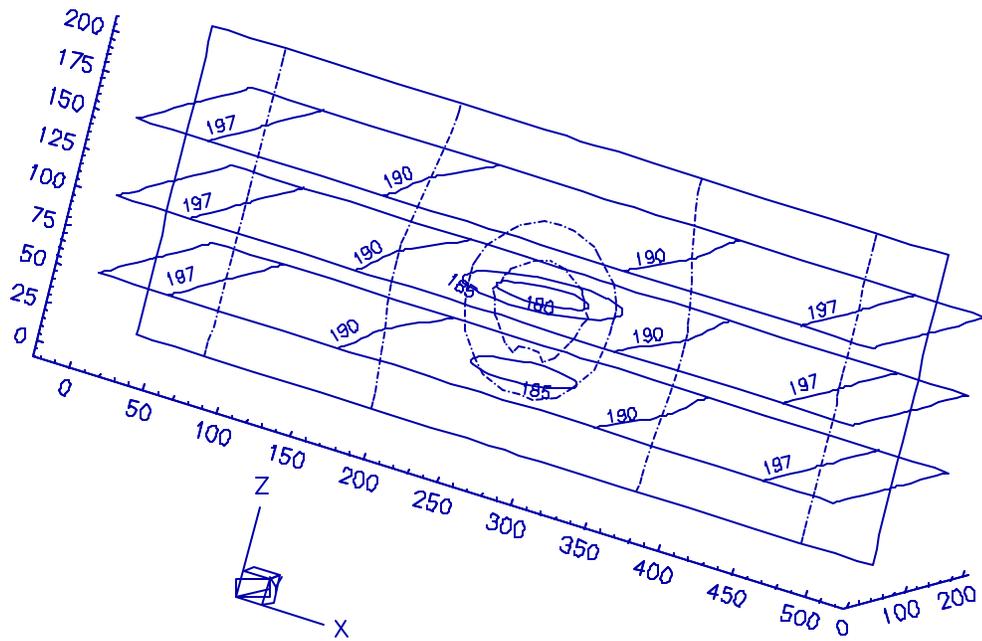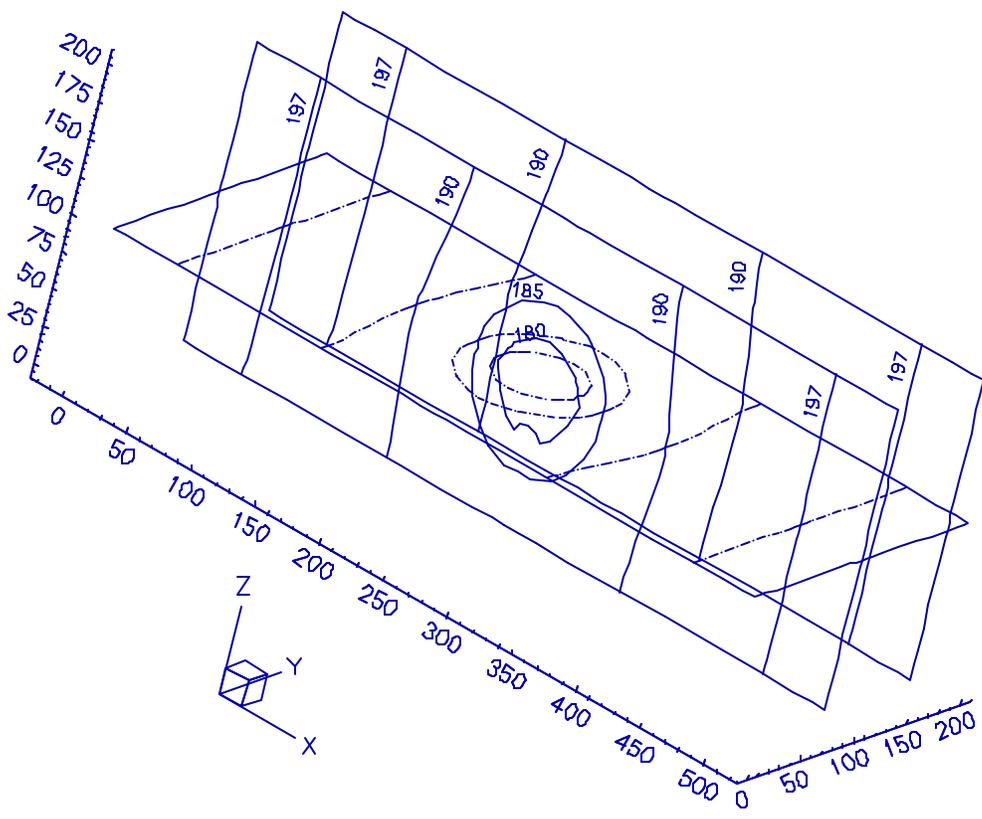
(a)
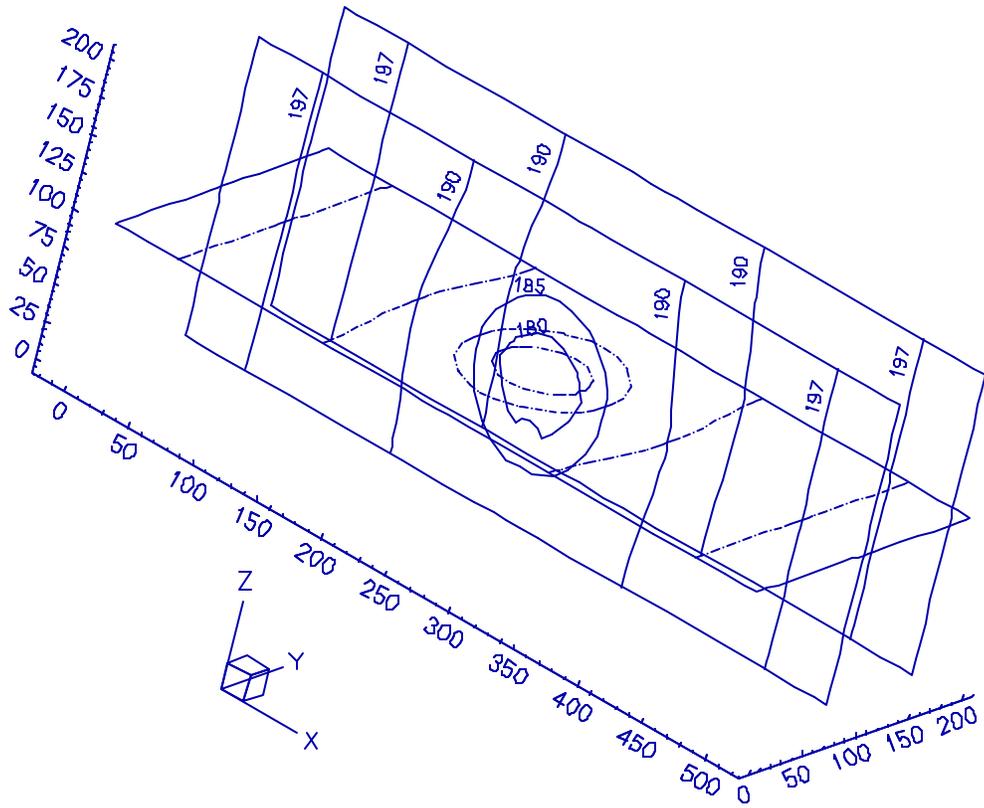
(b)

Figure 4.25   The total head distribution at time = 75 days in application 4.
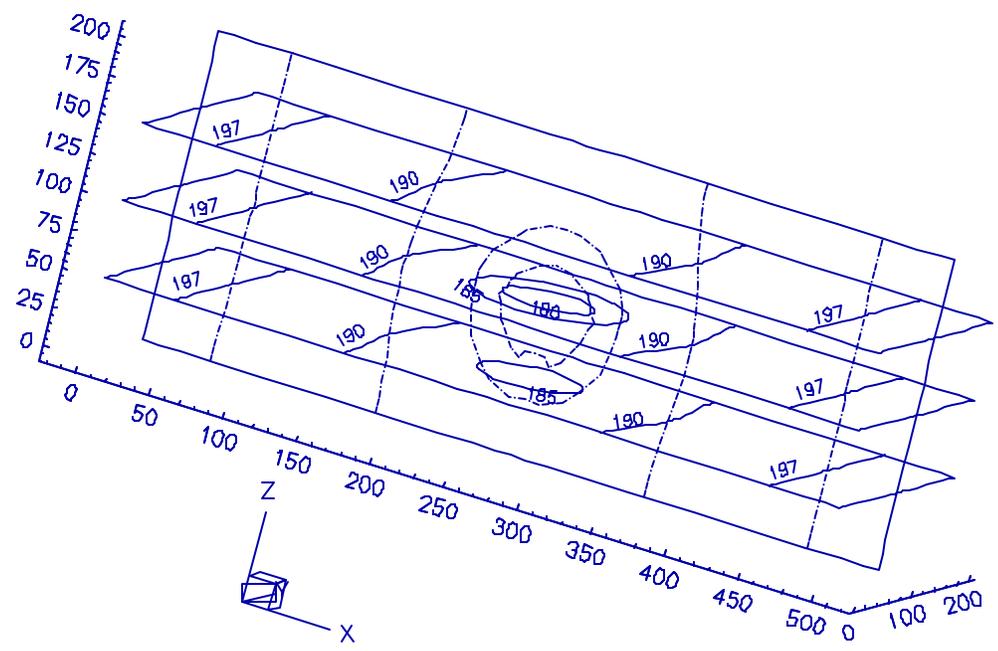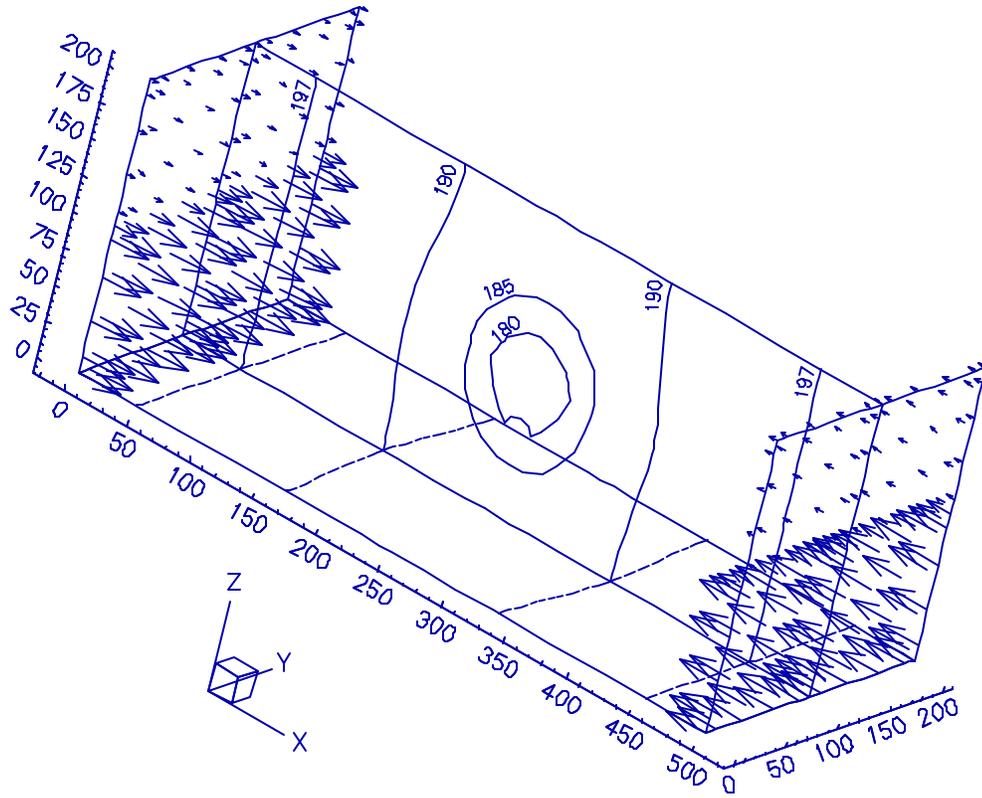
(a)

(b)

Figure 4.26 The total head distribution at time = 155 days in application 4.
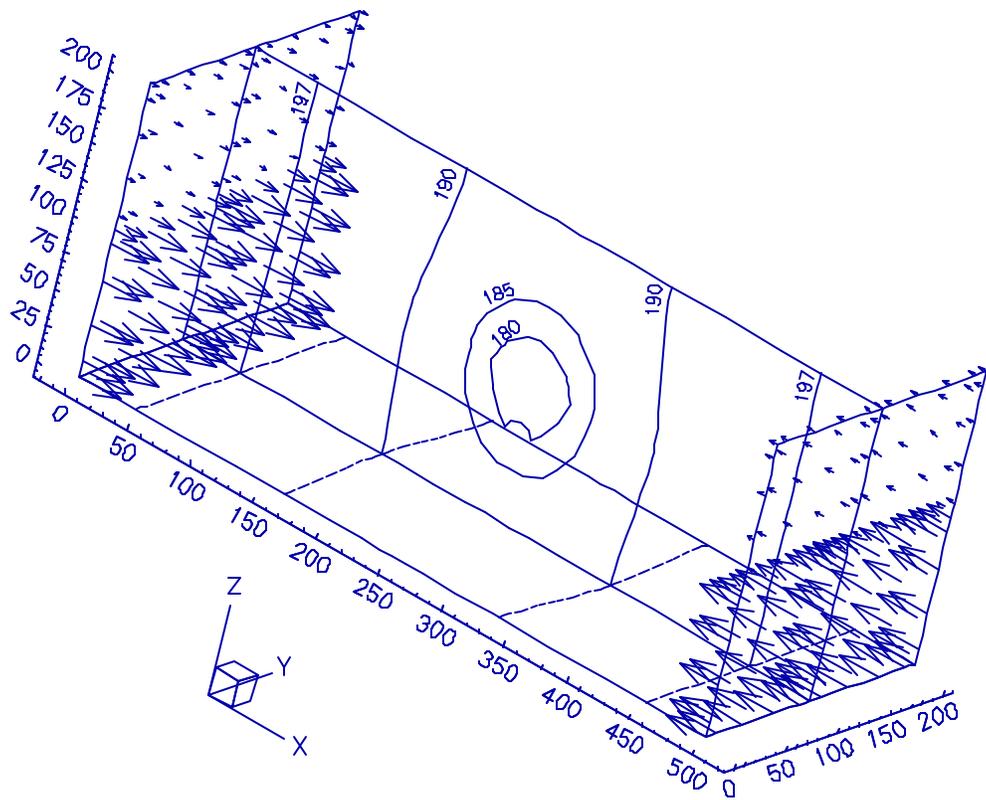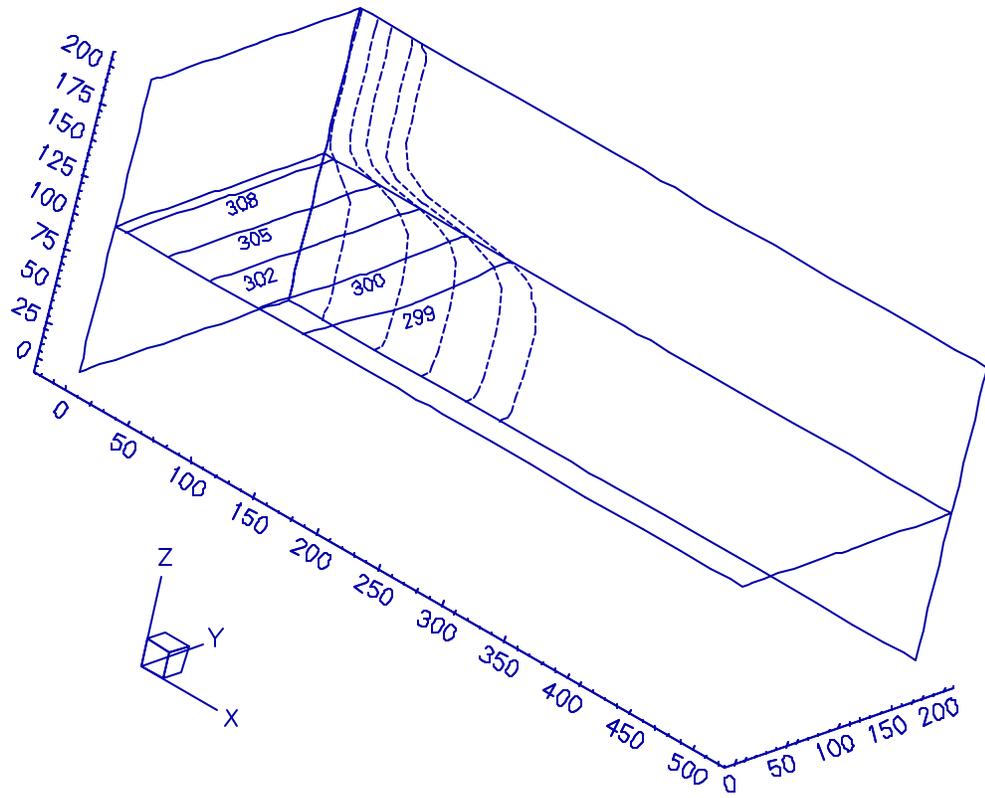
(a)

(b)

Figure 4.27   The total head distribution and boundary flow rates at (a) time = 75 days and (b) time = 155 days in application 4.
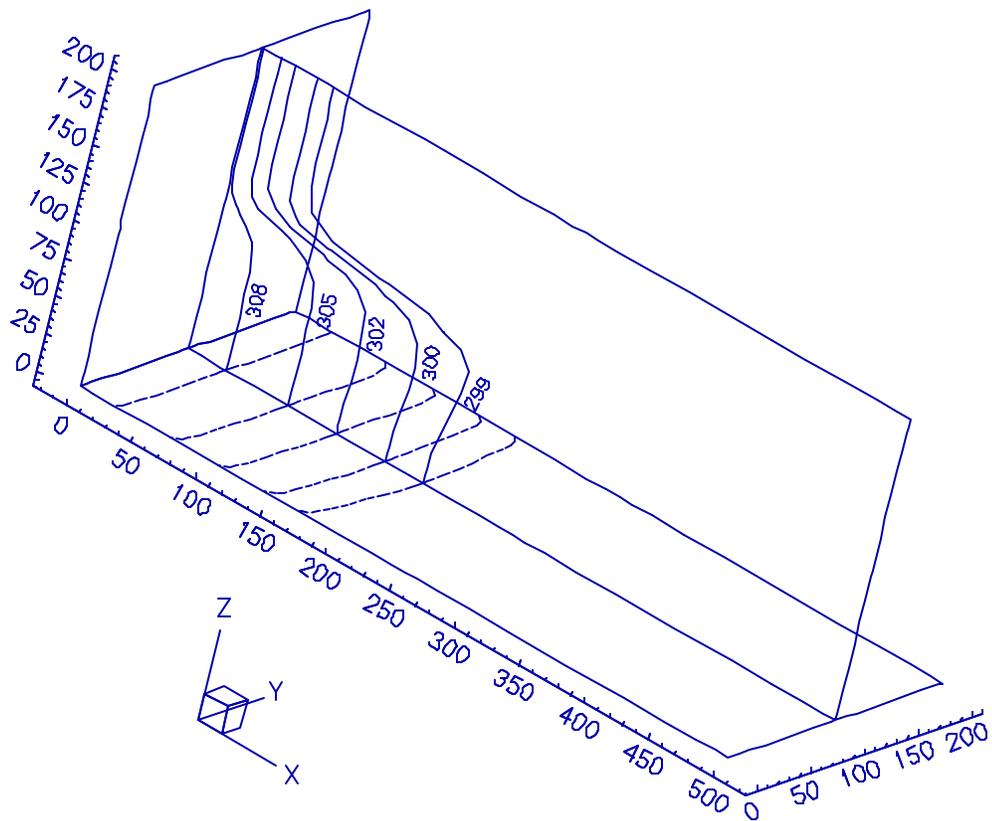
(a)

(b)

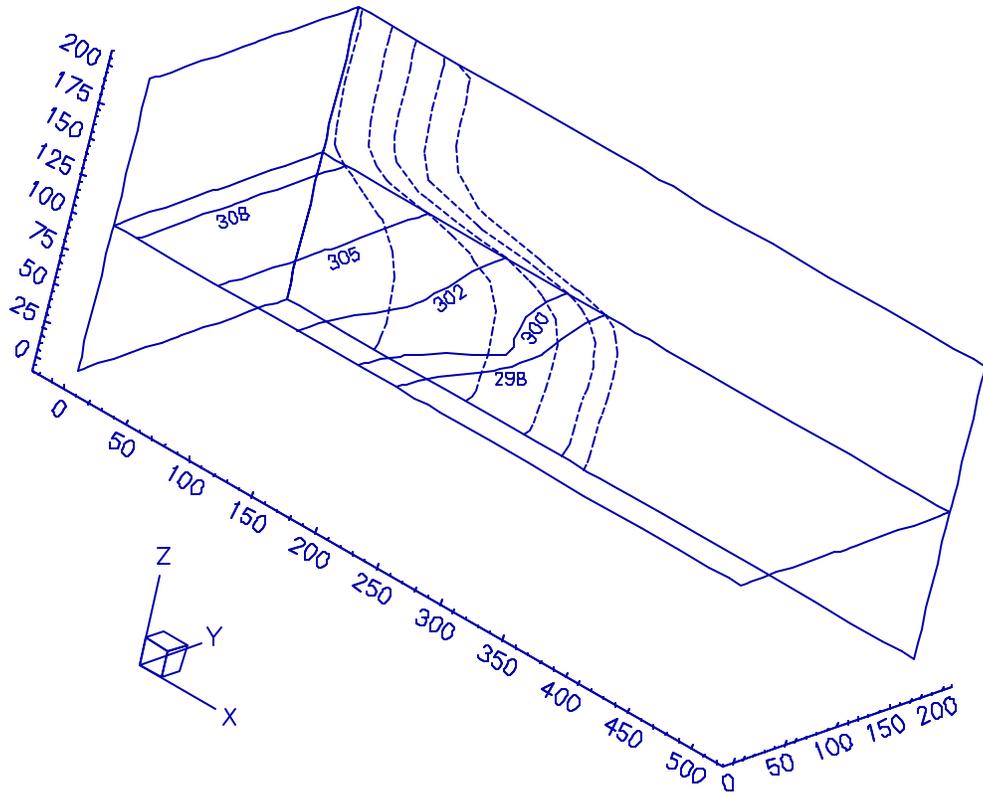**Figure 4.28   The temperature distribution at time = 75 days in application 4.**
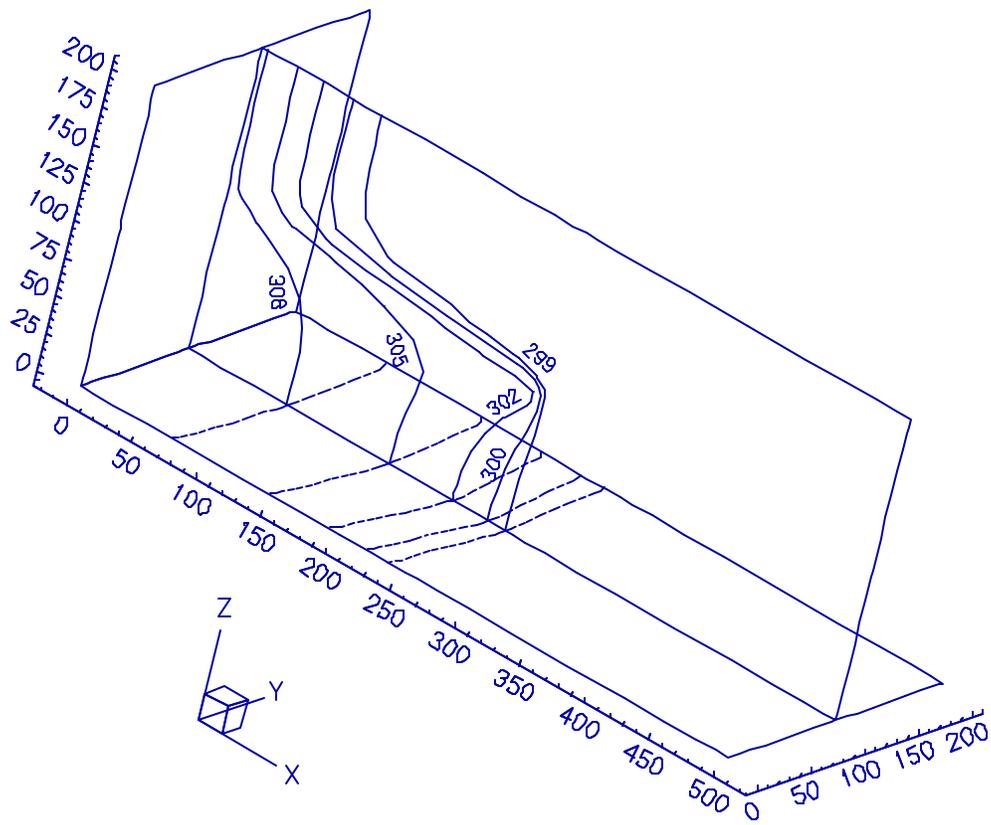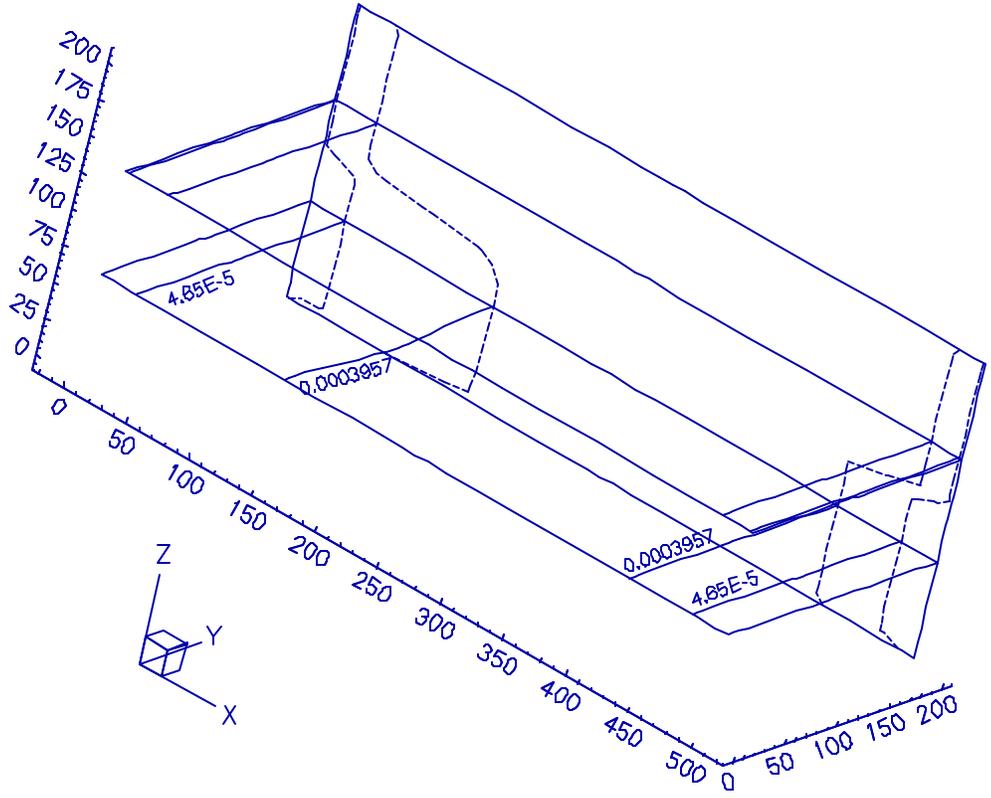
(a)

(b)

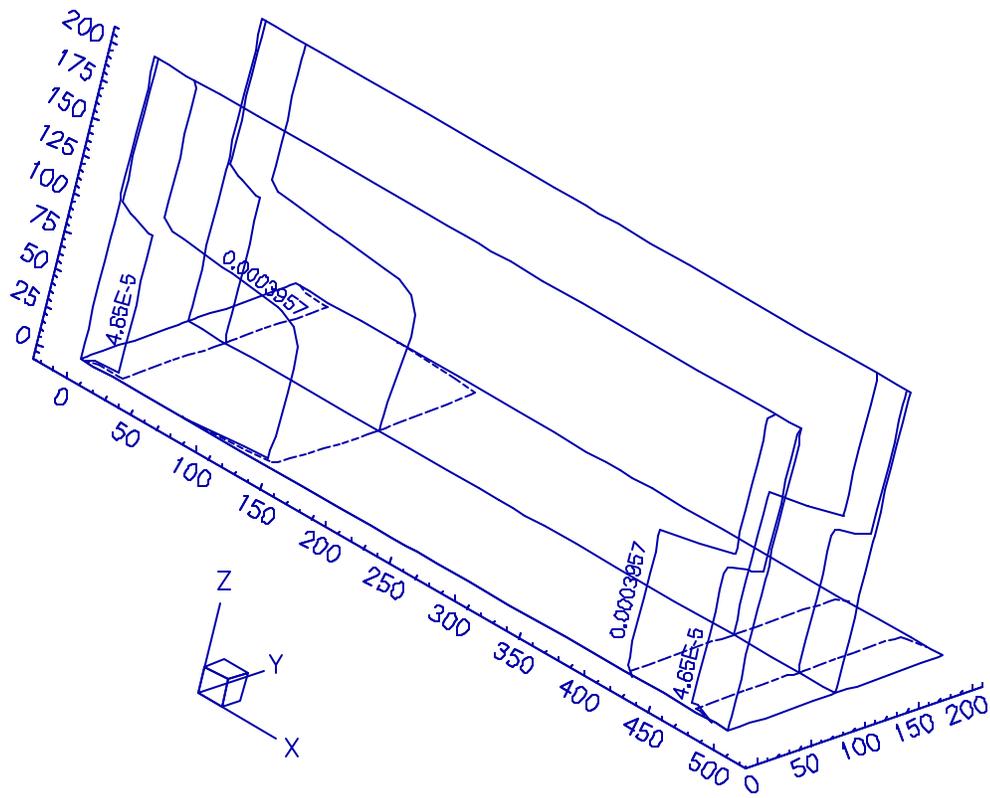**Figure 4.29   The temperature distribution at time = 155 days in application 4.**

73

(a)

(b)

Figure 4.30   The total analytical concentration distribution of component 1 (Ca^{2+}) or 2 (CO_3^{2-}) at time = 75 days in application 4.
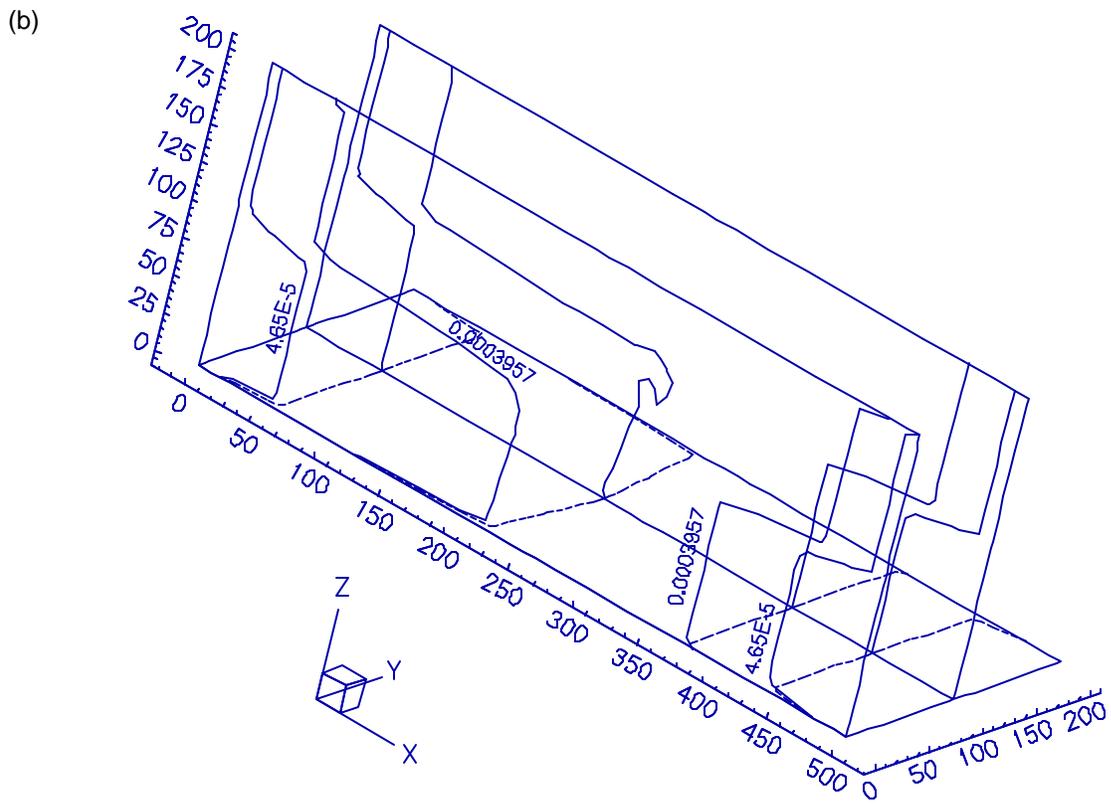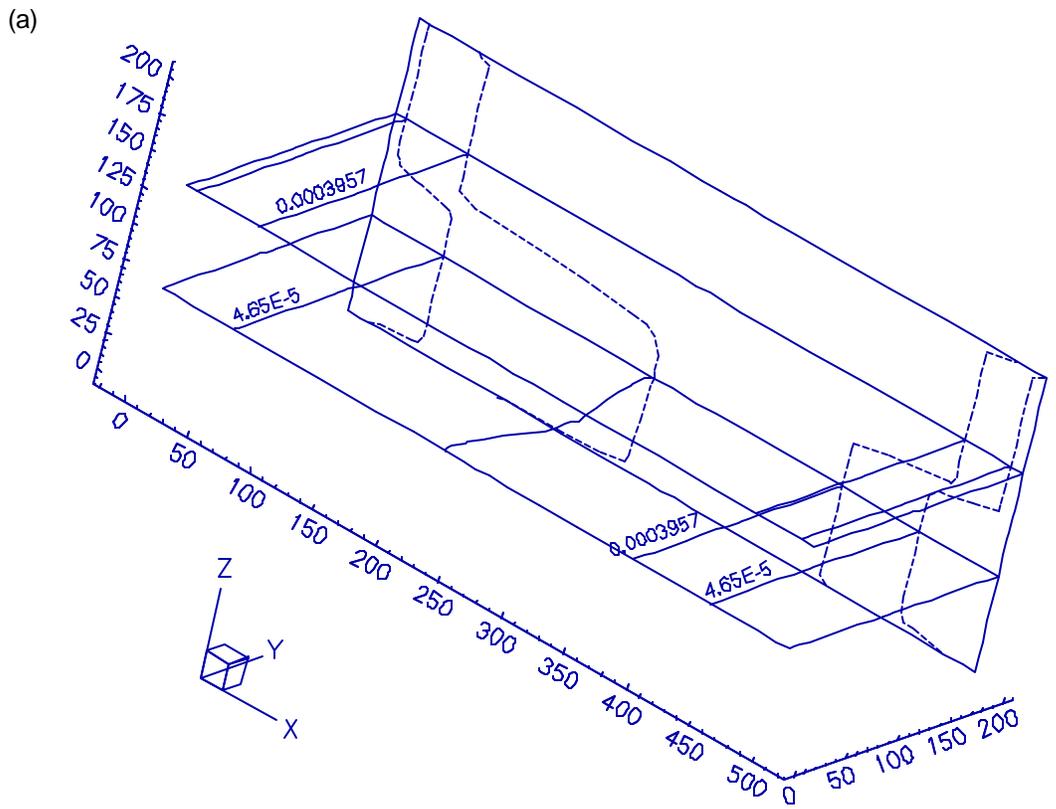
Figure 4.31 The total analytical concentration distribution of component 1 (Ca^{2+}) or 2 (CO_3^{2-}) at time = 155 days in application 4.

with $CaCO_{3(s)}$ and dissolution is the only reaction of mass transfer between the solid and the aqueous phases through the transient simulation, the total analytical concentrations of component Ca^{2+} and CO_3^{2-} are equal. This can be examined based on the prescribed equilibrium constants for such a simplified system. Therefore, the concentration distributions shown in Figures 4.30 and 4.31 can be for component Ca^{2+} or CO_3^{2-}.

Both figures point out that the higher incoming temperature from the left boundary drives the 0.0003957 M concentration line on the left side to move faster than that on the right side. But the 4.65×10^{-5} M concentration lines on both sides seem to move with the same progress toward the locations of sinks. As a matter of fact, the higher temperature makes this 4.65×10^{-5} M concentration line on the left move slower than that on the right. Figure 4.32 provides a two-dimensional concentration contour plot on the plane of y = 100 dm, which gives a clearer view on the two 4.65×10^{-5} M concentration lines at time = 155 days. The contrast between the 0.0003957 M and the 4.65×10^{-5} M concentration lines can be explained as follows.

According to the van't Hoff equation, a higher temperature will increase the equilibrium constant of a chemical reaction if the associated reaction enthalpy is positive. In this application, thus, a higher temperature coming in from the left boundary will help the precipitation of $CaCO_{3(s)}$ and increase the effort of cleanup. This is why the 4.65×10^{-5} M concentration line on the left side migrates slower than that on the right side.

Given the total analytical component concentration, on the other hand, a higher temperature will cause a lower dissolved component concentration achieved at equilibrium. This lower dissolved component concentration at an upstream location and the current time will be brought to a downstream location along with the subsurface flow at the next time. Therefore, the total analytical component concentration at that downstream location will be reduced at the next time if the temperature of the upstream location is higher than that of the downstream location at the current time [Cheng and Yeh, 1997b]. This basically explains what happens to the 0.0003957 M concentration line in this case.

The above discussion, in a way, implicitly signifies the need of numerical models for quantitative analyses: a higher temperature introduced from the upstream boundary will cause more precipitation of $CaCO_{3(s)}$ within the upstream region due to a positive reaction enthalpy, but it also generates a continuous dissolution within the downstream region due to the reduced total analytical concentration (as explained above) at a lower temperature. Under a lower temperature, a downstream location always has less total dissolved concentration coming in than going out, which makes this location undersaturated with the existence of precipitated species. To attain equilibrium, dissolution occurs to decrease the concentration of $CaCO_{3(s)}$ at this downstream location. These opposite situations hence lead to a difficulty in determining whether or not an introduction of a higher temperature from the upstream boundary would reduce the overall cleanup effort. To this point, a qualified numerical model is required. One should be aware that the upstream and the downstream regions mentioned above are actually representing a relatively high-temperature region and a relatively low-temperature region. Therefore, their sizes are to be changed with the transfer of heat. The speed of heat transfer, relative to that of chemical transport, has been found able to significantly affect the cleanup result [Cheng and Yeh, 1997b]. This subject, even though not discussed here, could be interesting to environmental researchers and engineers.
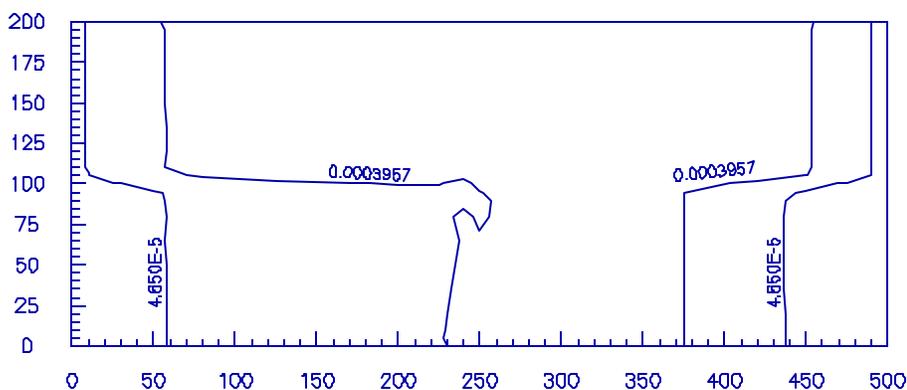

Figure 4.32 The total analytical concentration distribution on the plane of y = 100 dm for component 1 or 2 at time = 155 days in application 4.

Chapter 5
SUMMARY

This report presents a three-dimensional coupled model of subsurface flow, chemical transport, and heat transfer (3DHYDROGEOCHEM). It is developed mainly for dealing with reactive chemical transport problems, under the local equilibrium assumption, in subsurface systems. The capabilities of the model are described in Chapter 1.

Two approaches are included to deal with the chemical transport of nonconservative aqueous components. Although Approach 2 (Eq. (2.10)) is theoretically able to provide more accurate numerical solutions than Approach 1 (Eq. (2.9)), the associated constraint (Eq. (2.11)) has limited its application to many problems (e.g., verification example problem 19, application 3, and application 4). Therefore, Approach 1 is recommended for practical use. If Eq. (2.11) is satisfied, however, not only can more accurate simulation results be obtained, but also less computational time is needed to achieve convergency by using Approach 2 than Approach 1 (e.g., verification example problems 16 and 21).

The numerical problem associated with the existence of dominating precipitated species is resolved by using the modified total analytical component concentration as the primary dependent variable of chemical transport for nonconservative aqueous components that comprise the dominating precipitated species. With the strategy described in Chapter 3, the concentration of a dominating precipitated species is bookkept with two numerical arrays in the computer code, such that chemical transport can be correctly handled from a numerical point of view.

The coupling of subsurface flow, chemical transport, and heat transfer has been demonstrated necessary for many problems (e.g., verification example problem 8 and application 4). The strategy of strong coupling (Chapter 3) is implemented to achieve accurate computational results in steady-state simulations, while weak coupling is employed to perform the transient computation efficiently. The execution of strong coupling in the transient-state simulation is computationally expensive. With the strategy of weak coupling installed in the model (Chapter 3), the difference between the outcomes from strong coupling and weak coupling for the transient-state simulation is minimized. This difference becomes larger when the change of density effect to subsurface flow during a time step becomes more significant. In this case, the difference can be reduced by using smaller time-step sizes which might also be needed for reaching convergency even though strong coupling is considered. Thus, the weak coupling strategy presented in this report is considered capable of both accurately and efficiently performing transient computations.

To further improve the model for practical applications, the following subjects might be studied next.

1. Including the compressibilities of groundwater and soil matrix.
2. Including peak-capturing/valley elevating in subsurface flow and heat transfer.
3. Including the change of porosity due to the change of sorbed and precipitated species.
4. Including colloid (or waterborne) transport that allows the movement of the solid phase (both precipitated and sorbed) along with subsurface flow.
5. Coupling chemical kinetics into the model.
6. Coupling microbiological effects into the model.
7. Searching for more efficient matrix solvers; for example, multigrid solvers.
8. Coupling radioactive decay chains into the model.
9. Coupling the surface water system with the groundwater system.

Appendix A DATA INPUT GUIDE OF 3DHYDROGEOCHEM

DATA SET 1 --- PROBLEM IDENTIFICATION AND DESCRIPTION

Six records with FORMAT(I5,A67,I2,6I1)for the first one and FREE FORMAT for the others are required for this data set. The 1-st record contains the following variables.

1. NPROB = Problem number. If it is given as 0, then the simulation will be stopped.
2. TITLE = Title of the problem. It may contain up to 70 characters.
3. IMOD = Integer indicating the simulation modes to be carried out;
   - 0 = Do the initial variable computation ONLY, for both flow and transport simulations. The purpose for this mode is to verify the input data,
   - 1 = Do the transport simulation only,
   - 10 = Do the flow simulation only,
   - 11 = Do both flow and transport simulations.
4. IGEOM = Integer indicating if
   - 1 = the geometry, boundary and pointer arrays are to be printed,
   - 2 = the boundary and pointer arrays are to be computed or read via logical units. If to be computed, they should be written on logical units.

IGEOM	Arrays to be computed	Arrays to be read	Print?
0	Pointer and Boundary	None	No
1	Pointer and Boundary	None	Yes
2	Pointer	Boundary	No
3	Pointer	Boundary	Yes
4	None	Pointer and Boundary	No

* If IGEOM is an even number, neither pointer nor boundary arrays will be printed.

* If IGEOM is an odd number, both pointer and boundary arrays will be printed.

* If IGEOM is less than or equal to 1, boundary arrays will be computed and written on logical unit 13, but if IGEOM is greater than 1, boundary arrays will be read via logical unit 13.

* If IGEOM is less than or equal to 3, pointer arrays will be computed and written on logical unit 14, but if IGEOM is greater than 3, pointer arrays will be read via logical unit 14.

5. IBUG = Integer indicating if the diagnostic output is desired?
   - 0 = no,
   - 1 = yes.
6. ICHNG = Integer control number indicating if the cyclic change of rainfall-seepage nodes is to be printed;
   - 0 = no,
   - 1= yes.
7. NHGCI = The number shows that in EQMOD, the last iteration results of the inner loop are to be printed for each NHGCI outer loop (precipitation loop) iteration.
8. IQUARF = Index of using quadrature for numerical integration for flow;
   - 1 = Nodal quadrature for surface integration, Gaussian quadrature for element integration,
   - 2 = Nodal quadrature for surface integration, Nodal quadrature for element integration,
   - 3 = Gaussian quadrature for surface integration, Gaussian quadrature for element integration,
   - 4 = Gaussian quadrature for surface integration, Nodal quadrature for element integration.

IQUAR	Surface integration	Element integration
1	Nodal quadrature	Gaussian quadrature
2	Nodal quadrature	Nodal quadrature
3	Gaussian quadrature	Gaussian quadrature
4	Gaussian quadrature	Nodal quadrature

9. IQUART = Index of using quadrature for numerical integration for transport;
   1 = Nodal quadrature for surface integration, Gaussian quadrature for element integration,
   2 = Nodal quadrature for surface integration, Nodal quadrature for element integration,
   3 = Gaussian quadrature for surface integration, Gaussian quadrature for element integration,
   4 = Gaussian quadrature for surface integration, Nodal quadrature for element integration.

The 2-nd record contains only the following variable.

   1. ITEMP = Index of considering temperature effect.
      0 = No, it is not considered,
      1 = Yes, it is taken into consideration.

The 3-rd record is required only when ITEMP = 1. It contains the following variables.

   1. SPHEAT = Specific heat of groundwater. (L**2/T**2/^{o}K, 4.18*10^3 m**2/sec**2/^{o}K)
   2. GASCNT = Ideal gas constant. (M*L**2/T**2/^{o}K/mole, 8.3144 Kg*m**2/sec**2/^{o}K/mole)

The 4-th record is required only when ITEMP =1. It contains the following 8 variables.

   1. RHOC1 = The 1-st coefficient for setting up the temperature-dependent formula for groundwater density,
   2. RHOC2 = The 2-nd coefficient for setting up the temperature-dependent formula for groundwater density,
   3. RHOC3 = The 3-rd coefficient for setting up the temperature-dependent formula for groundwater density,
   4. RHOC4 = The 4-th coefficient for setting up the temperature-dependent formula for groundwater density,
   5. RHOC5 = The 5-th coefficient for setting up the temperature-dependent formula for groundwater density,
   6. RHOC6 = The 6-th coefficient for setting up the temperature-dependent formula for groundwater density,
   7. RHOC7 = The 7-th coefficient for setting up the temperature-dependent formula for groundwater density,
   8. RHOC8 = The 8-th coefficient for setting up the temperature-dependent formula for groundwater density.

The 5-th record is required only when ITEMP =1. It contains the following 8 variables.

   1. AMUC1 = The 1-st coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity,
   2. AMUC2 = The 2-nd coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity,
   3. AMUC3 = The 3-rd coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity,
   4. AMUC4 = The 4-th coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity,
   5. AMUC5 = The 5-th coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity,
   6. AMUC6 = The 6-th coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity,
   7. AMUC7 = The 7-th coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity,
   8. AMUC8 = The 8-th coefficient for setting up the temperature-dependent formula for groundwater dynamic viscosity.

The 6-th record is required only when ITEMP =1. It contains the following 8 variables.

   1. DMBC1 = The 1-st coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity,
   2. DMBC2 = The 2-nd coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity,

3. DMBC3 = The 3-rd coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity,
4. DMBC4 = The 4-th coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity,
5. DMBC5 = The 5-th coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity,
6. DMBC6 = The 6-th coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity,
7. DMBC7 = The 7-th coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity,
8. DMBC8 = The 8-th coefficient for setting up the moisture content-dependent formula for isotropic heat conductivity.

DATA SET 2 --- OPTION PARAMETERS

One line and eight subsets of free-formatted data records are required for this data set.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
A. subset 1:
1.1. NITRFT = Number of iterations allowed for solving steady-state flow-transport iteration.
1.2. OMEFTF = Flow relaxation factor for solving steady-state flow-transport iteration.
1.3. OMEFTT = Transport relaxation factor for solving steady-state flow-transport iteration.
B. subset 2:
2.1. KSSF = Flow steady-state control;
0 = steady-state solution desired,
1 = transient state or transient solutions.
2.2. KSST = Transport steady-state control;
0 = steady-state solution desired,
1 = transient state or transient solutions.
2.3. ILUMP = Is mass lumping to be used?
0 = no,
1 = yes.
2.4. IMID = Is mid-difference?
0 = no,
1 = yes.
2.5. IPNTSF = Is pointwise iterative matrix solver to be used for flow simulations?
0 = no, but block iteration solver is used,
1 = yes,
2 = PPCG is to be used,
3 = ILUCG is to be used.
2.6. IPNTST = Is pointwise iterative matrix solver to be used for transport simulations?
0 = no, but block iteration solver is used,
1 = yes,
2 = PPCG is to be used,
3 = ILUCG is to be used.
2.7. NSTRF = Number of flow records to be read in logical unit 11 for restarting.
2.8. NSTRT = Number of transport records to be read in logical unit 12 for restarting.
C. subset 3:
3.1. KGRAV = Gravity term control;
0 = no gravity term,
1 = with gravity term.
3.2. WF = Time derivative weighting factor for flow simulations;
0.5 = Crank-Nicolson central,
1.0 = backward difference and/or mid-difference.
3.3. OMEF = Iteration parameter for solving the nonlinear flow equation;

0.0 - 1.0 = under-relaxation,
1.0 - 1.0 = exact relaxation,
1.0 - 2.0 = over-relaxation.
3.4. OMIF = Relaxation parameter for solving the linearized flow matrix equation with the block iteration solver or the pointwise iteration solver;
0.0 - 1.0 = under relaxation,
1.0 - 1.0 = exact relaxation,
1.0 - 2.0 = over relaxation.
3.5 CNSTKR = Constrain on the minimum relative conductivity;
0.0 = no constrain,
If CNSTKR is greater than 0.0, then users need to examine if the convergent results are reasonable. Usually, a convergent solution can be obtained when CNSTKR is given greater than 10^{-4}.

D. subset 4:
4.1. KVIT = Velocity input control;
-1 = card input for velocity and moisture content,
1 = steady-state velocity and moisture content will be calculated from steady-state flow simulations,
2 = transient velocity and moisture content will be obtained from transient flow simulations.
4.2. IWET = Weighting function control;
0 = Galerkin weighting,
1 = Upstream weighting.
4.3. IOPTIM = Optimization factor computing indicator;
1 = Optimization factor is to be computed,
0 = Optimization factor is to be set to -1.0 or 0.0 or 1.0.
4.4. APHAG = Upstream weighting factor used if IOPTIM.EQ.0
NOTE: (1) The value is between 0.0 and 1.5,
(2) If APHAG.GT.1.34, the program will choose appropriate values for the weighting factor.

E. subset 5:
5.1. WT = Time derivative weighting factor for transport simulations;
0.5 = Crank-Nicolson central,
1.0 = backward difference and/or mid-difference.
5.2. WVT = Integration factor for velocity; should be between 0.0 to 1.0.
5.3. OMET = Iteration parameter for solving the nonlinear transport matrix equation;
0.0 - 1.0 = under-relaxation,
1.0 - 1.0 = exact relaxation,
1.0 - 2.0 = over-relaxation.
5.4. OMIT = Relaxation parameter for solving the linearized transport matrix equation with the block iteration solver or the pointwise iteration solver;
0.0 - 1.0 = under relaxation,
1.0 - 1.0 = exact relaxation,
1.0 - 2.0 = over relaxation.

F. subset 6:
6.1. NXW = The number of grids for elemental tracking in the x-direction.
6.2. NYW = The number of grids for elemental tracking in the y-direction.
6.3. NEW = The number of grids for elemental tracking in the z-direction.
6.4 IDETQ = Index of particle tracking pattern;
1 = Average velocity is used (more accurate),
2 = Single velocity of the source point is used (less computation).

G. subset 7:
7.1. LGRN = Index of using the Lagrangian approach for transient transport simulation;
0 = no,
1 = yes.
7.2. ICOMP = Index of computing chemical equilibrium at Gaussian points;
0 = no,
1 = yes.
7.3. ICAP = Index of peaks/valleys capturing;
0 = no,
1 = yes.

H. subset 8: (This subset is needed only if IPNTSF.GT.1 or IPNTST.GT.1)
    8.1. IEIGEN = Signal of parameter estimation for GG in the polynomial preconditioned conjugate gradient method;
        0 = Not requested,
        Otherwise = Requested.
    8.2. GG = The upper bound on the maximum eigenvalue of the coefficient matrix used in the polynomial preconditioned conjugate gradient method.
    8.3. ALPHA = Weighting factor for computing the diagonal element of the diagonal matrix used in the modified incomplete Cholesky preconditioned conjugate gradient method.
    8.4. OMEGAG = Relaxation parameter used in the SSOR PCG method.

DATA SET 3 --- ITERATION PARAMETERS

One line and two subsets of free-formatted data records are required for this data set, one for flow simulations, the other for transport simulations.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.

A. subset 1: For flow simulations -
    1.1. NITERF = Number of iterations allowed for solving the nonlinear flow equation.
    1.2. NCYLF = No. of cycles permitted for iterating rainfall-seepage boundary conditions per time step.
    1.3. NPITERF = No. of iterations permitted for solving the linearized flow equation using a block or pointwise iterative matrix solver.
    1.4. TOLAF = Steady-state convergence criteria for flow simulations, (L).
    1.5. TOLBF = Transient-state convergence criteria for flow simulations, (L).
B. subset 2: For transport simulations -
    2.1. NITERT = Number of iterations allowed for solving the nonlinear transport equation.
    2.2. NPITERT = No. of iterations for block or pointwise solution to solve the linearized transport equation.
    2.3. TOLAT = Steady-state convergence criteria for chemical transport simulations.
    2.4. TOLBT = Transient state convergence criteria for chemical transport simulations.
    2.5. ALLOW = The allowed factor for neglecting concentrations in a convergence test.
    2.6. TOLATR = Steady-state convergence criteria for heat transfer simulations.
    2.7. TOLBTR = Transient state convergence criteria for heat transfer simulations.

DATA SET 4 --- TIME CONTROL PARAMETERS

One line and five subsets of data records are required for this data set.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
A. subset 1: free format
    1.1. NTI = Number of time steps or time increments.
    1.2. NDTCHG = No. of times to reset time step size to initial time step size.
B. subset 2: free format
    2.1. DELT = Initial time step size, (T).
    2.2. CHNG = Percentage of change in time step size in each of the subsequent time increments, (dimensionless in decimal point).
    2.3. DELMAX = Maximum value of DELT, (T).
    2.4. TMAX = Maximum simulation time, (T).
C. subset 3: format = 80I1
    3.1. KPR0 = Printer control for steady-state and initial conditions;
        0 = print nothing,
        1 = print FLOW, FRATE, and TFLOW,
        2 = print above (1) plus Darcy velocity, total analytical and total dissolved concentrations,
        3 = print above (2) plus pressure head,

4 = print above (3) plus total head,

5 = print above (4) plus moisture content and material flux.

3.2. KPR(I) = Printer control for the I-th (I = 1, 2, ..., NTI) time step similar to KPR0.

D. subset 4: format = 80I1

4.1. KDSK0 = Auxiliary storage control for steady-state and initial conditions;

0 = no storage,

1 = store on Logical Unit 1.

4.2. KDSK(I) = Auxiliary storage control for the I-th (I = 1, 2, ..., NTI) time step similar to KDSK0.

E. subset 5: free format

5.1. TDTCH(I) = Time when the I-th (I = 1, 2, ..., NDTCHG) step-size-resetting is needed.

**** NOTE: NTI can be computed by NTI = I1 + 1 + I2 + 1, where I1 = largest integer not exceeding Log(DELMAX/DELT)/Log(1+CHNG), I2 = largest integer not exceeding (RTIME-DELT*((1+CHNG)**(I1+1)-1)/CHNG)/DELMAX, RTIME = Real simulation time, DELMAX,DELT,and CHNG are defined in Data Set 3.

DATA SET 5 --- NUMBERS OF COMPONENTS AND SPECIES

There are 2 lines in this data set:

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.

The 2-nd line contains the following 12 variables (FREE FORMAT)

1. NONA = No. of aqueous component species.
2. NONS = No. of adsorbent component species.
3. NOMX = No. of complexed species.
4. NOMY = No. of adsorbed species.
5. NOMZ = No. of ion-exchanged species.
6. NOMP = No. of precipitated species.
7. N0 = No. of conservative mobile components.
8. N1A = No. of type-A non-conservative mobile components; this type of components IS NOT involved dominating precipitated species.
9. N1B = No. of type-B non-conservative mobile components; this type of components IS involved dominating precipitated species.
10. N2 = No. of immobile components.
11. N3 = No. of fixed activity (concentration) components.
12. NODP = No. of precipitation dominating species.

DATA SET 6 --- MATERIAL PROPERTIES

One line and four subsets of free-formatted data records are required for this data set.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.

A. subset 1:

1.1. NMAT = Number of material types.

1.2. NMPPM = No. of material properties per material = 9 for the present version.

B. subset 2: A total of NMAT records are needed per problem, one each for one material. (NOTE: This subset is needed only if IMOD.NE.1)

2.1. PROPF(I,1) = Saturated xx-conductivity or permeability of the medium I, (L/T or L**2).

2.2. PROPF(I,2) = Saturated yy-conductivity or permeability of the medium I, (L/T or L**2).

2.3. PROPF(I,3) = Saturated zz-conductivity or permeability of the medium I, (L/T or L**2).

2.4. PROPF(I,4) = Saturated xy-conductivity or permeability of the medium I, (L/T or L**2).

2.5. PROPF(I,5) = Saturated xz-conductivity or permeability of the medium I, (L/T or L**2).

2.6. PROPF(I,6) = Saturated yz-conductivity or permeability of the medium I, (L/T or L**2).

2.7. PROPF(I,7) = Referenced water density (usually it is clean water density).

2.8. PROPF(I,8) = Referenced water dynamic viscosity (usually it is clean water dynamic viscosity).

2.9. PROPF(I,9) = Saturated moisture content, (L**3/L**3), of medium I.

C. subset 3: (This subset can be modified along with the source code)

3.1. RHOMU(1) = Coefficient for calculating fluid dynamic viscosity as a factor of the 1-st dissolved chemical concentration.

    3.NONA. RHOMU(NONA) = Coefficients for calculating fluid dynamic viscosity as a factor of the NONA-th dissolved chemical concentration.

    3.NONA+1. RHOMU(NONA+1) = Coefficients for calculating fluid dynamic viscosity as a factor of the OH⁻ concentration.

D. subset 4: A total of NMAT records are needed per problem, one each for one material. (NOTE: This subset is needed if IMOD.NE.10 or ITEMP.NE.0)

    4.1. PROPT(I,1) = Bulk density, (M/L**3) for medium I,

    4.2. PROPT(I,2) = Longitudinal dispersivity, (L), for medium I,

    4.3. PROPT(I,3) = Lateral dispersivity, (L), for medium I,

    4.4. PROPT(I,4) = Molecular diffusion coefficient, (L**2/T), for medium I,

    4.5. PROPT(I,5) = Tortuosity, (Dimensionless) for medium I,

    4.6. PROPT(I,6) = Specific Heat, (L**2/T**2/°K), of dry medium I.

    4.7. PROPT(I,7) = Referenced water density (usually it is clean water density).

    4.8. PROPT(I,8) = Apparent thermal conductivity, (M*L/°K/T**3), of medium I.

    4.9. PROPT(I,9) = Saturated moisture content, (L**3/L**3), of medium I.

DATA SET 7 --- SOIL PROPERTIES

One line, three or five subsets of free-formatted data records are required for this data set depending on the forms of the soil property functions.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.

A. subset 1: Soil property control parameters

    1.1. KSP = Soil property input control;

         0 = analytical input,

         1 = tabular data input.

    1.2. NSPPM = Number of points in tabular soil property functions or number of parameters to specify analytical soil functions per material.

    1.3. KCP = Permeability input control;

         0 = input saturated hydraulic conductivity,

         1 = input saturated permeability.

    1.4. GRAV = Acceleration of gravity, (L/T**2).

B. subset 2a: Analytical soil parameters - This sub-data-set is needed if and only if KSP is not 0. Two sets of records are required, one for moisture-content parameters and the other for conductivity (permeability) parameters.

    2.1. SPP(J,I,1) = Analytical moisture-content parameter J of material I, J =1..NSPPM. NMAT sets of these parameters are required for I = 1..NMAT. That is, if SPP(J,I,1) for J = 1..NSPPM can be put on a single line, we need NMAT consecutive lines for the sets of parameters.

    2.2. SPP(J,I,2) = Analytical relative conductivity parameter J of material I. Similar input data setting is required for these parameters as for SPP(J,I,1)

C. subset 2b: Soil properties in tabular form - This sub-data-set is needed if and only if KSP is 0. Four sets of records are needed — for pressure, water-content, relative conductivity (or relative permeability), and water capacity, respectively.

    3.1. SPP(J,I,4) = Tabular value of pressure head of the J-th point for material I. NMAT sets of these parameters are required for I = 1..NMAT. That is, if SPP(J,I,4) for J = 1..NSPPM can be put on a single line, we need NMAT consecutive lines for the sets of parameters.

    3.2. SPP(J,I,1) = Tabular value of moisture-content of the J-th point in material I. Similar input data setting is required for these parameters as for SPP(J,I,4)

    3.3. SPP(J,I,2) = Tabular value of relative conductivity of the J-th point in material I. Similar input data setting is required for these parameters as for SPP(J,I,4)

    3.4. SPP(J,I,3) = Tabular value of moisture-content capacity of the J-th point in material I. Similar input data setting is required for these parameters as for SPP(J,I,4)

DATA SET 8 --- NODAL COORDINATE

One line and two subsets of free-formatted data records are required.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
A. subset 1:
1.1. NNP = Number of nodes.
B. subset 2: nodal coordinates - Coordinates for NNP nodes are needed if KVIT.LE.0. Usually a total of NNP records (KVIT records) are required. However, if a group of subsequent nodes appears in a regular pattern, automatic generation can be made. Each record contains the following variables and is FREE-FORMATTED.
2.1. NI = Node number of the first node in the sequence.
2.2. NSEQ = NSEQ subsequent nodes will be automatically generated.
2.3. NAD = Increment of node number for each of the NSEQ subsequent nodes.
2.4. XNI = x-coordinate of node NI, (L).
2.5. YNI = y-coordinate of node NI, (L).
2.6. ZNI = z-coordinate of node NI, (L).
2.7. XAD = Increment of x-coordinate for each of the NSEQ subsequent nodes, (L).
2.8. YAD = Increment of y-coordinate for each of the NSEQ subsequent nodes, (L).
2.9. ZAD = Increment of z-coordinate for each of the NSEQ subsequent nodes, (L).
**** NOTE: A record with 9 0's must be used to signal the end of this data set.

DATA SET 9 --- SUBREGION DATA

This data set is not required if both or either IPNTSF or IPNTST is 0. One line and three subsets of free-formatted data records are required.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
A. subset 1: One free format data record is needed for this sub-data-set.
1.1. NREGN = No. of subregions.
B. subset 2: No. of Nodes for each Subregion - Normally, NREGN records are required. However, if a regular pattern appears, automatic generation can be made. Each record contains the 5 variables and is FREE-FORMATTED.
2.1. NK = Subregion number of the first subregion in a sequence.
2.2. NSEQ = NSEQ subsequent subregions will have their no. of nodes automatically generated.
2.3. NKAD = Increment of NK in each of the NSEQ subsequent subregions.
2.4. NODES = No. of nodes for the subregion NK.
2.5. NOAD = Increment of NODES in each of the NSEQ subsequent subregions.
**** NOTE: A record with 5 0's must be used to end the input of this subset.

C. subset 3: Mapping between Global nodes and Subregion Nodes - This subset should be repeated NREGN times, one for each subregion. For each subregion, normally, LNNP records are needed. However, automatic generation can be made if subregional node number appears in a regular pattern. Each record contains 5 variables and is FREE-FORMATTED.
3.1. LI = Local node number of the first node in a sequence,
3.2. NSEQ = NSEQ subsequent local nodes will be generated automatically,
3.3. LIAD = Increment of LI for each of the NSEQ subsequent nodes,
3.4. NI = Global node number of local node LI,
3.5. NIAD = Increment of NI for each of the NSEQ subsequent nodes.
**** NOTE: A record with 5 0's must be used to signal the end of this subset.

DATA SET 10 --- ELEMENT DATA

One line and two subsets of free-formatted data records are required for this data set.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
A. subset 1:
    1.1. NEL = Number of elements.
B. subset 2: Element incidences for NEL elements are needed if KVIT.LE.0. Usually, a total of NEL records are needed. However, if a group of elements appears in a regular pattern, automatic generation is made. Each record contains the following variable and is FREE-FORMATTED.
    2.1. MI = Global element number of the first element in a sequence.
    2.2. NSEQ = NSEQ subsequent elements will be automatically generated.
    2.3. MIAD = Increment of MI for each of the NSEQ subsequent elements.
    2.4. IE(MI,1) = Global node number of the first node of element MI.
    2.5. IE(MI,2) = Global node number of the second node of element MI.
    2.6. IE(MI,3) = Global node number of the third node of element MI.
    2.7. IE(MI,4) = Global node number of the fourth node of element MI.
    2.8. IE(MI,5) = Global node number of the fifth node of element MI.
    2.9. IE(MI,6) = Global node number of the sixth node of element MI.
    2.10. IE(MI,7) = Global node number of the seventh node of element MI.
    2.11. IE(MI,8) = Global node number of the eighth node of element MI.
    2.12. IEMAD = Increment of IE(MI,1) through IE(MI,8) for each of the NSEQ elements.

**** Note: IE(MI,1) - IE(MI,8) are numbered according to the convention shown in Figure A.1. The first four nodes start from the front, lower, left corner and progressing around the bottom element surface in a counterclockwise direction. The other four nodes begin from the front, upper, left corner and progressing around the top element surface in a counterclockwise direction.

DATA SET 11 --- MATERIAL TYPE CORRECTION

One line and two subsets of free-formatted data records are required for this data set.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
A. subset 1:
    1.1. NCM = Number of elements with material corrections.
B. subset 2: This set of data records is required only if NCM > 0. Normally, NCM records are required. However, if a group of elements appears in a regular pattern, automatic generation may be made. Each record contains the following variables.
    2.1. MI = Global element number of the first element in the sequence.
    2.2. NSEQ = NSEQ subsequent elements will be generated automatically.
    2.3. MAD = Increment of element number for each of the NSEQ subsequent elements.
    2.4. MITYP = Type of material correction for element MI.
    2.5. MTYPAD = Increment of MITYP for each of the NSEQ subsequent elements.

**** NOTE: A line with 5 0's must be used to signal the end of this data set.

DATA SET 12 --- H+, e-, IONIC STRENGTH, AND ADSORPTION INFORMATION

There are 4 lines and 2 optional data subsets in this data set:

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
The 2-nd line contains 4 variables (FREE FORMAT)
    1. SICOR = User's specified ionic strength.
    2. ICOR = Is ionic strength used to correct activity coefficients?
      0 = No,
1 = Yes, but the fixed ionic strength, SICOR, is used,
2 = Yes, but the variable ionic strength, to be computed in program, is used.
    3. LNH = The location of component H+ in the component list.
    4. LNE = The location of component e- in the component list.
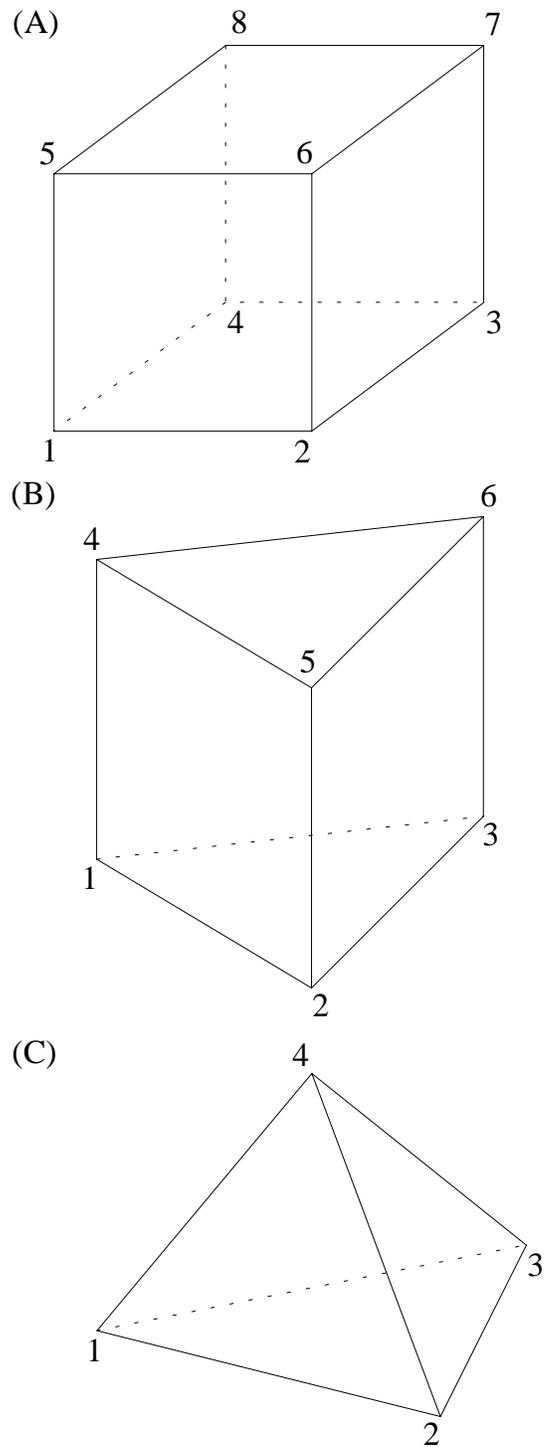The 3-rd line contains the following 2 variables (FREE FORMAT)

Figure A.1 Global node number index of (A) a hexahedral, (B) a triangular prism, and (C) a tetrahedral element.

1. TEMP = Absolute temperature.
2. PRESU = Pressure in atm.
The 4-th line contains the following 4 variables (FREE FORMAT)
    1. PEMN = Expected min. pe-Value.
    2. PEMX = Expected max. pe-Value.
    3. PHMN = Expected min. pH-value.
    4. PHMX = Expected max. pH-value.
The following data subset is needed only if NONS.GT.0. There are (1+NSORB) lines and 1 subset in this subset:

The 1-st line contains the following 2 variables (FREE FORMAT)
    1. NSORB = No. of adsorbing sites.
    2. IADS = Adsorption model index:
        0 = Simple surface complexation model is used,
        1 = Constant capacitance model is used,
        2 = Triple layer model is used.
The 2-nd through the (NSORB+1)-th lines contain the following 2 variables for each line (FREE FORMAT)
    1. LNOA(I) = The location of component exp(-e*psio/kT) of the I-th adsorbing site in the component list.
    2. LNBA(I) = The location of component exp(-e(psib/kT) of the I-th adsorbing site in the component list.
The subset contains (NMAT*NSORB) lines. Each line contains the following 3 variables (FREE FORMAT)
    1. CAP1M(I,J) = The I-th adsorbing site capacitance between the surface and "o" plane for the J-th material.(Faraday/L**2)
    2. CAP2M(I,J) = The I-th adsorbing site capacitance between "o" plane and "b" plane for the J-th material.(Faraday/L**2)
    3. SREAM(I,J) = The I-th adsorbing site surface area for the J-th material. (L**2/M solid)
The following data subset is needed only if NOMZ.GT.0. There are (1+NSITE) lines and 1 subset in this subset:

The 1-st line contains the following variable (FREE FORMAT)
    1. NSITE = No. of ion exchange sites.
The 2-nd through the (NSITE+1)-th lines contain the following 3 variables for each line (FREE FORMAT)
    1. NOMZI(I) = No. of ion-exchanged species participating in the I-th ion exchange site.
    2. LNI(I) = Location of the I-th reference ion-exchanged species in the ion-exchanged species list. Hence, LNI(I) could be the number in between 1 and NOMZI(I).
    3. DCYEC(I) = the decay constant of the I-th ion exchange site.
The subset contains (NMAT*NSITE) lines. Each line contains the following variable (FREE FORMAT)
    1. ECM(I,J) = the ion exchange capacity of the I-th ion exchange site for the J-th material.(Equivalence M solid)

DATA SET 13 --- BASIC REAL AND INTEGER PARAMETERS FOR CHEMICAL COMPUTATION

There are 2 lines in this data set:

The 1-st line contains the following variable (A80)
    1. DATNAM = Description of the data set.
The 2-nd line contains the following 6 variables (FREE FORMAT)
    1. EPS = Error tolerance for iterations.
    2. NITER = No. of iterations allowed.
    3. NPCYL = No. of cycles allowed for iterating precipitation-dissolution. This parameter has to be greater than or equal to 1 even though there is no precipitated species existing in the system of interest.
    4. CNSLT1 = Initial conc. constraint on limiting the conc. of complexed species. This value is set as the initial value of conc. constrain to obtain convergent results. In order to not only get convergent results but also convergence as soon as possible, CNSLT1 =2.0 is recommended (based on experiments).
    5. CNSLT2 = Final conc. constraint on limiting the conc. of complexed species. This value is set as the final value of conc. constrain to obtain convergent results. This value can be greater than, equal to, or less than CNSLT1.
    6. DCNST = Increment to adjust the value of conc. constrain. This value can be greater than, equal to, or less than 0. If CNSLT2 is greater than CNSLT1, then DCNST should be greater than 0. If CNSLT2 is less than CNSLT1, then DCNST should be less than 0. The value of conc. constrain is continuously adjusted until either a convergent solution is obtained or the value is equal to CNSLT2.

DATA SET 14 --- RADIOACTIVE DECAY CONSTANTS OF ALL COMPONENTS

There are (2+NON) lines in this data set:

The 1-st line contains the following variable (A80)
      1. DATNAM = Description of the data set.
The 2-nd through the (NON+1)-th lines contain the following 3 variables for each line (A10,D20.12)
      1. CNAM(J) = Name of the J-th component.
      2. DECAY(J) = Decay constant of the J-th component.
      3. TBGD(J) = Background concentration of the J-th component.
The (NON+2)-th line contains the following (NONA+1) variables (FREE FORMAT)
      1. WTMOL(1) = The molecular weight of the 1-st aqueous component.
        .
        .
      NONA+1. WMOTL(NONA+1) = The molecular weight of OH^-.

DATA SET 15 --- COMPONENT SPECIES AND THEIR ION-EXCHANGED SPECIES

There are 1 line and NON data subsets in this data set:

The 1-st line contains the following variable (A80)
      1. DATNAM = Description of the data set.
The 1-st data subset reads the information about the 1-st component. There are 2 lines and IONEX subsets in the subset:
The 1-st line contains the following 3 variables (A20,I5)
      1. SPECN(I) = Name of the I-th component species
      2. INDC(I) = Transport index of the I-th component species:
        0 = Conservative mobile component species,
        11 = Non-conservative mobile component species that IS NOT involved in dominating precipitated species,
        12 = Non-conservative mobile component species that IS involved in dominating precipitated species,
        2 = Immobile component species,
        3 = Fixed act. (conc.) component species.
      3. ISCN(I) = Chemical index of the I-th component species:
        0 = Conc. (act.) of the I-th component is to be computed.
        2 = A specified quantity, instead of total analytical conc., related to the I-th component is used for the I-th mole balance equation.
        3 = Fixed conc. (act.) of the I-th component is given as the 1-st variable of the next line.
The 2-nd line contains the following 3 variables (FREE FORMAT)
      1. CN(I) = The initial guess of the I-th component species.
        NOTE: This value cannot be zero.
      2. VJ(I) = Valance of the I-th component species.
      3. IONEX = No. of ion exchange sites which the I-th component species participates in.
The following IONEX subsets contain 4 lines for each subset:

The 1-st line contains the following variable (FREE FORMAT)
      1. ISITE = The ISITE-th ion exchange site which the I-th component species participates in.
The 2-nd line contains the following 2 variables (FREE FORMAT)
      1. SPECN(II) = Name of the ion-exchanged species resulted from the I-th component species in the ISITE-th ion exchange site.
      2. ISCN(II) = Index of the ion-exchanged species:
        0 = Species conc. is to be computed,
        3 = Species conc. is fixed.
The 3-rd line contains the following 2+NON*2 variables (FREE FORMAT)
  1. CN(II) = The initial guess of the ion-exchanged species.
        NOTE: (1) This value cannot be zero.
                    (2) If ISCN(II)=3, this value is the fixed conc. of the ion-exchanged species.
      2. PKIPD = Log10 of the selectivity constant of the ion-exchanged species.

3. AXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the ion-exchanged species. For use in mass reaction equations.

.
.

NON+2. AXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the ion-exchanged species. For use in mass reaction equations.

NON+3. BXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the ion-exchanged species. For use in mole balance equations.

.

.

NON*2+2. BXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the ion-exchanged species. For use in mole balance equations.

The 4-th line contains the following variable (FREE FORMAT)

1. ENTHPY(II-NON) = The enthalpy for the associated ion-exchange reaction at the referenced temperature.

The 2-nd data subset reads the information about the 2-nd component.

.
.

The 3-rd data subset reads the information about the 3-rd component.

.
.
.

The NON-th data subset reads the information about the NON-th component.

.
.

DATA SET 16 --- COMPLEXED SPECIES AND THEIR ION-EXCHANGED SPECIES

There are 1 line and NOMX data subsets are in this data set:

The 1-st line contains the following variable (A80)

1. DATNAM = Description of the data set.

The 1-st data subset reads the information about the 1-st complexed species. There are 3 lines and IONEX subsets in the subset:

The 1-st line contains the following 2 variables (A20,I5)

1. SPECN(II) = Name of the complexed species
2. ISCN(II) = Index of the complexed species:
    0 = Species conc. is to be computed,
    3 = Species conc. is fixed.

The 2-nd line contains the following (3+(NON*2)) variables (FREE FORMAT)

1. CN(II) = The initial guess of the complexed species.
    NOTE: (1) This value cannot be zero.
                   (2) If ISCN(II)=3, this value is the fixed conc. of the complexed species.
2. PKIPD = Log10 of the equilibrium constant of the complexed species.
3. AXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the I-th complexed species. For use in mass reaction equations.

.
.

NON+2. AXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the I-th complexed species. For use in mass reaction equations.

NON+3. IONEX = No. of ion exchange sites the complexed species participates in.

NON+4. BXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the I-th complexed species. For use in mole balance equations.

.

.

NON*2+3. BXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the I-th complexed species. For use in mole balance equations.

The 3-rd line contains the following variable (FREE FORMAT)

1. ENTHPY(I) = The enthalpy of the associated complexation reaction at the referenced temperature.
The following IONEX subsets contain 4 lines for each subset:

The 1-st line contains the following variable (FREE FORMAT)
    1. ISITE = The ISITE-th ion exchange site which the complexed species participates in.
The 2-nd line contains the following 2 variables (FREE FORMAT)
    1. SPECN(II) = Name of the ion-exchanged species resulted from the complexed species in the ISITE-th ion exchange site.
    2. ISCN(II) = Index of the ion-exchanged species:
       0 = Species conc. is to be computed,
       3 = Species conc. is fixed.
The 3-rd line contains the following 2+NON*2 variables (FREE FORMAT)
    1. CN(II) = The initial guess of the ion-exchanged species.
       NOTE: (1) This value cannot be zero.
                 (2) If ISCN(II)=3, this value is the fixed conc. of the ion-exchanged   species.
    2. PKIPD = Log10 of the selectivity constant of the ion-exchanged species.
    3. AXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the ion-exchanged species. For use in mass reaction equations.
       .
       .
    NON+2. AXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the ion-exchanged species. For use in mass reaction equations.
    NON+3. BXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the ion-exchanged species. For use in mole balance equations.
       .
       .
    NON*2+2. BXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the   ion-exchanged species. For use in mole balance equations.
The 4-th line contains the following variable (FREE FORMAT)
    1. ENTHPY(II-NON) = The enthalpy for the associated ion-exchange reaction at the referenced temperature.
The 2-nd data subset reads the information about the 2-nd complexed species.
  .
  .
The 3-rd data subset reads the information about the 3-rd complexed species.
  .
  .
  .
The NON-th data subset reads the information about the NON-th complexed species.
  .
  .

DATA SET 17 --- ADSORBED SPECIES

There are 1 line and NOMY data subsets are in this data set:

The 1-st line contains the following variable (A80)
    1. DATNAM = Description of the data set.
The 1-st data subset reads the information about the 1-st adsorbed species. There are 3 lines in the subset:
    The 1-st line contains the following 2 variables (A20,I5)
        1. SPECN(II) = Name of the adsorbed species
        2. ISCN(II) = Index of the adsorbed species:
           0 = Species conc. is to be computed,
           3 = Species conc. is fixed.
    The 2-nd line contains the following (2+(NON*2)) variables (FREE FORMAT)
        1. CN(II) = The initial guess of the adsorbed species.
           NOTE: (1) This value cannot be zero.
                    (2) If ISCN(II)=3, this value is the fixed conc. of the adsorbed species.

2. PKIPD = Log10 of the equilibrium constant of the adsorbed species.
3. AXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the adsorbed species. For use in mass reaction equations.

.

.

NON+2. AXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the adsorbed species. For use in mass reaction equations.

NON+3. BXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the adsorbed species. For use in mole balance equations.

.

.

NON*2+2. BXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the adsorbed species. For use in mole balance equations.

The 3-rd line contains the following variable (FREE FORMAT)
1. ENTHPY(I) = The enthalpy of the associated adsorption reaction at the referenced temperature.
The 2-nd data subset reads the information about the 2-nd adsorbed species.

.

.

The 3-rd data subset reads the information about the 3-rd adsorbed species.

.

.

.

The NON-th data subset reads the information about the NON-th adsorbed species

.

.

DATA SET 18 --- PRECIPITATED SPECIES (NOTE: If there exist NODP precipitation dominating species, then they must be put as the first NODP species in the list of precipitated species)

There are 1 line and NOMP data subsets are in this data set:

The 1-st line contains the following variable (A80)
1. DATNAM(1..8) = Description of the data set.
The 1-st data subset reads the information about the 1-st precipitated species. There are 3 lines in the subset:
The 1-st line contains the following 2 variables (A20,I5)
1. SPECN(II) = Name of the precipitated species
2. ISCN(II) = Index of the precipitated species:
0 = Species conc. is to be computed,
3 = Species conc. is fixed.
The 2-nd line contains the following (2+(NON*2)) variables (FREE FORMAT)
1. CN(II) = The initial guess of the precipitated species.
NOTE: (1) This value cannot be zero.
(2) If ISCN(II)=3, this value is the fixed conc. of the precipitated species.
2. PKIPD = Log10 of the equilibrium constant of the precipitated species.
3. AXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the precipitated species. For use in mass reaction equations.

.

.

NON+2. AXYZP(I,NON) = Stoichiometric coeff. of the NON-th component the precipitated species. For use in mass reaction equations.
NON+3. BXYZP(I,1) = Stoichiometric coeff. of the 1-st component in the precipitated species. For use in mole balance equations.

.

.

NON*2+2. BXYZP(I,NON) = Stoichiometric coeff. of the NON-th component in the precipitated species. For use in mole balance equations.

The 3-rd line contains the following variable (FREE FORMAT)
    1. ENTHPY(I) = The enthalpy of the associated precipitation reaction at the referenced temperature.
The 2-nd data subset reads the information about the 2-nd precipitated species.
  .
  .
  .
 The 3-rd data subset reads the information about the 3-rd precipitated species.
  .
  .
  .
 The NON-th data subset reads the information about the NON-th precipitated species
  .
  .

DATA SET 19 --- CARD INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS

One line and three subsets of free-formatted data records are required for this data set, one for initial pressure head, the other for initial concentration. Generally, for each subset NNP record, one record for each node, is needed. However, if a group of nodes appears in a regular pattern, auto-generation is made.

The 1-st line contains the following variable (A80)
    DATNAM = Description of the data set.
A. subset 1: Initial pressure head - each record contains the following variables.
    (NOTE: This subset is needed only if IMOD.NE.1)
      1.1. NI = Global node number of the first node in the sequence.
      1.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
      1.3. NAD = Increment of node number for each of the NSEQ nodes.
      1.4. HNI = Initial or pre-initial pressure head of node NI, (L).
      1.5. HAD = Increment of initial or pre-initial head for each of the NSEQ nodes, (L).
      1.6. HRD = 0.0
**** NOTE: A line with 6 0's must be used to signal the end of this data set.

  * NOTE ON INITIAL CONDITIONS AND RESTARTING: The initial condition for a transient calculation may be obtained in two different ways: from card input or steady-state calculation using time-invariant boundary conditions that are different from those for the transient computation. In the latter case a card input of the pre-initial conditions is required as the zeroth order iterate of the steady-state solution.
  * NOTE ON STEADY-STATE INPUT: Steady-state option may be used to provide either the final state of a system under study or the initial conditions for a transient state calculation. In the former case, KSSf = 0, KSSt = 0, and NTI = 0, and in the latter case KSSf = 0 or KSSt = 0 and NTI > 0. If KSSf > 0, and KSSt > 0, there will be no steady-state calculation.
B. subset 2: Two parts are included. The 1-st part is to read the initial total analytical concentrations. Therefore, NON groups of data are needed. Each group contains at least 2 records and each record contains the following variables. (NOTE: This subset is needed only if IMOD.NE.10)
      2.1.1. NI = Global node number of the first node in the sequence.
      2.1.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
      2.1.3. NAD = Increment of node number for each of the NSEQ nodes.
      2.1.4. CNI = Initial or pre-initial concentration of node NI, (M/L**3).
      2.1.5. CAD = Increment of CNI for each of the NSEQ nodes, (M/L**3),
      2.1.6. CRD = Geometrical increment of CNI for each of the NSEQ subsequent nodes.
The 2-nd part is to read the initial concentrations of precipitation dominating species. Therefore, NODP groups of data are needed (if NODP is not equal to 0). Each group contains at least 2 records and each record contains the following variables. (NOTE: This subset is needed only if IMOD.NE.10)
      2.2.1. NI = Global node number of the first node in the sequence.
      2.2.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
      2.2.3. NAD = Increment of node number for each of the NSEQ nodes.
      2.2.4. CNI = Initial or pre-initial concentration of node NI, (M/L**3).
      2.2.5. CAD = Increment of CNI for each of the NSEQ nodes, (M/L**3),
      2.2.6. CRD = Geometrical increment of CNI for each of the NSEQ subsequent nodes.
**** NOTE: A record with 6 0's must be used to signal the end of this data set.

NOTE ON INITIAL CONDITIONS: The initial condition for a transient calculation may be obtained in two different ways: from card input or steady-state calculation using time-invariant boundary conditions that are different from those for the transient computation. In the latter case, a card input of the pre-initial conditions is required as the zeroth order iterate of the steady-state solution.

NOTE ON STEADY-STATE INPUT: Steady-state option may be used to provide either the final state of a system under study or the initial conditions for a transient state calculation. In the former case, KSS = 0 and NTI = 0, and in the latter case KSS = 0 and NTI .GT. 0. If KSS .GT. 0, there will be no steady-state calculation.

C. subset 3: Initial temperature distribution (if ITEMP is not equal to 0) - each record contains the following variables. (NOTE: This subset is needed only if IMOD.NE.1)

      3.1. NI = Global node number of the first node in the sequence.

      3.2. NSEQ = NSEQ subsequent nodes will be generated automatically.

      3.3. NAD = Increment of node number for each of the NSEQ nodes.

      3.4. TNI = Initial or pre-initial pressure head of node NI, (L).

      3.5. TAD = Increment of initial or pre-initial head for each of the NSEQ nodes, (L).

      3.6. TRD = 0.0

**** NOTE: A line with 6 0's must be used to signal the end of this data set.

* NOTE ON INITIAL CONDITIONS AND RESTARTING: The initial condition for a transient calculation may be obtained in two different ways: from card input or steady-state calculation using time-invariant boundary conditions that are different from those for the transient computation. In the latter case a card input of the pre-initial conditions is required as the zeroth order iterate of the steady-state solution.

* NOTE ON STEADY-STATE INPUT: Steady-state option may be used to provide either the final state of a system under study or the initial conditions for a transient state calculation. In the former case, KSSf = 0, KSSt = 0, and NTI = 0, and in the latter case KSSf = 0 or KSSt = 0 and NTI > 0. If KSSf > 0, and KSSt > 0, there will be no steady-state calculation.

DATA SET 20 --- ELEMENT (DISTRIBUTED) SOURCE/SINK FOR FLOW SIMULATIONS

(NOTE: This data set is needed only if IMOD.NE.1)

One line and four subsets of free-formatted data records are required in this data set.

The 1-st line contains the following variable (A80)

    DATNAM = Description of the data set.

    A. subset 1: control parameters

      1.1. NSEL = No. of source/sink elements.

      1.2. NSPR = No. of source/sink profiles.

      1.3. NSDP = No. of data points in each of the NSPR source/sink profiles.

      1.4. KSAI = Is element-source/sink profile to be input analytically,

          0 = no,

          1 = yes.

    B. subset 2: source/sink profiles - This group of data is needed if and only if NSEL .GT. 0. For each sub-data-record, NSDP of the data pair (TSOSF(J,I),SOSF(J,I)) are required. If this sub-data-record can be fitted in a line, we will need NSPR lines.

      2.1. TSOSF(J,I) = Time of the J-th data point in the I-th profile, (T).

      2.2. SOSF(J,I) = Source/sink value of the J-th data point in the I-th profile, (L**3/T/L**2/L).

    C. subset 3: global source/sink element number - This group of data is needed if and only if NSEL.GT.0. NSEL data points are required for this record.

      3.1. LES(I) = Global element number of the I-th compressed distributed source/sink element.

    D. subset 4: Source type assigned to each element - Usually one record per element. However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following.

      4.1. MI = Global element number of the first element in the sequence.

      4.2. NSEQ = NSEQ elements will be generated automatically.

      4.3. MAD = Increment of element number for each of the NSEQ elements.

      4.4. MITYP = Source type in element MI.

      4.5. MTYPAD = Increment of MITYP for each of the NSEQ elements.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

DATA SET 21 --- POINT (WELL) SOURCE/SINK DATA FOR FLOW SIMULATION

(NOTE: This data set is needed only if IMOD.NE.1)

One line and four subsets of free-formatted data records are required for this data set.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
   A. subset 1: control parameters
       1.1. NWNP = No. of well or point source/sink nodal points.
       1.2. NWPR = No. of well or point source/sink strength profiles.
       1.3. NWDP = No. of data points in each of the NWPR profiles.
       1.4. KWAI = Is well-source/sink profile to be input analytically,
               0 = no,
               1 = yes.
   B. subset 2: source/sink profiles - This group of data is needed if and only if NWNP .GT. 0. For each sub-data-record, NWDP of the data pair (TWSSF(J,I),WSSF(J,I)) are required. If this sub-data-record can be fitted in a line, we will need NWPR lines.
       2.1. TWSSF(J,I) = Time of the J-th data point in the I-th profile, (T).
       2.2. WSSF(J,I) = Source/sink value of the J-th data point in the I-th profile, (L**3/T).
   C. Record 3: global source/sink element number - This group of data is needed if and only if NWNP.GT.0. NWNP data points are required for this record.
       3.1. NPW(I) = Global node number of the I-th compressed well source/sink node.
   D. subset 4: Source type assigned to each well - Usually one record per element. However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following.
       4.1. NI = Compressed well node number of the first node in the sequence.
       4.2. NSEQ = NSEQ nodes will be generated automatically.
       4.3. NAD = Increment of well node number for each of the NSEQ nodes.
       4.4. NITYP = Source type in node NI.
       4.5. NTYPAD = Increment of NITYP for each of the NSEQ nodes.
**** NOTE: A line with 5 0's is used to signal the end of this data set.

DATA SET 22 --- ELEMENT (DISTRIBUTED) SOURCE/SINK FOR TRANSPORT SIMULATIONS (NOTE: This data set is needed only if IMOD.NE.10 and ITEMP.EQ.0)

Two parts are included. In each part, one line and four subsets of free-formatted data records are required in this data set. The 1-st part is for chemical transport.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.
   A. subset 1: control parameters
       1.1.1. NSEL = No. of source/sink elements.
       1.1.2. NSPR = No. of source profiles, should be .GE. 1.
   1.1.3. NSDP = No. of data points in each profile, should be .GE. 2.
       1.1.4. KSAI = Is element-source/sink profile to be input analytically?
               0 = no,
               1 = yes.
   B. Record subset 2: source/sink profile - This sub-data-set is needed if and only if NSEL.GT.0. For each sub-data-record, NSDP of the data group (TSOSF(J,I), SOSF(J,I,1), SOSF(J,I,2)) is required. If this sub-data-record can be fitted in a line, we will need NSPR lines.
       1.2.1. TSOSF(J,I) = Time of J-th data point in I-th profile, (T).
       1.2.2.   SOSF(J,I,1) = Source/sink flow rate of the J-th data point in the I-th profile, (L**3/T/L**3); positive for source and negative for sink.
       1.2.3. SOSF(J,I,2) = Source/sink concentration of the J-th data point in the I-th profile, (M/L**3).
   C. subset 3: global source/sink element number. NSEL data points are required for this record.
       1.3.1. LES(I) = Global element number of the I-th compressed distributed source/sink element.
   D. subset 4: Source type assigned to each element - Usually one record per element. However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following.

1.4.1. MI = Global element number of the first element in the sequence.
1.4.2. NSEQ = NSEQ elements will be generated automatically.
1.4.3. MAD = Increment of element number for each of the NSEQ elements.
1.4.4. MITYP = Source type in element MI.
1.4.5. MTYPAD = Increment of MITYP for each of the NSEQ elements.

***** NOTE: Subset 4 has to be repeated (N0+N1A+N1B) times.

The 2-nd part is for temperature transport.
The 1-st line contains the following variable (A80)
DATNAM = Description of the data set.
A. subset 1: control parameters
2.1.1. NTSEL = No. of source/sink elements.
2.1.2. NTSPR = No. of source profiles, should be .GE. 1.
2.1.3. NTSDP = No. of data points in each profile, should be .GE. 2.
2.1.4. KTSAI = Is element-source/sink profile to be input analytically?
0 = no,
1 = yes.
B. Record subset 2: source/sink profile - This sub-data-set is needed if and only if NTSEL.GT.0. For each sub-data-record, NTSDP of the data group (TTSOSF(J,I), STOSF(J,I,1), STOSF(J,I,2)) is required. If this sub-data-record can be fitted in a line, we will need NTSPR lines.
2.2.1. TTSOSF(J,I) = Time of J-th data point in I-th profile, (T).
2.2.2. STOSF(J,I,1) = Source/sink flow rate of the J-th data point in the I-th profile, (L**3/T/L**3); positive for source and negative for sink.
2.2.3. STOSF(J,I,2) = Source/sink temperature of the J-th data point in the I-th profile, (°K).
C. subset 3: global source/sink element number. NTSEL data points are required for this record.
2.3.1. LTES(I) = Global element number of the I-th compressed distributed source/sink element.
D. subset 4: Source type assigned to each element - Usually one record per element. However, automatic generation can be made. For I-th (I = 1, 2, ...,) record, it contains the following.
2.4.1. MI = Global element number of the first element in the sequence.
2.4.2. NSEQ = NSEQ elements will be generated automatically.
2.4.3. MAD = Increment of element number for each of the NSEQ elements.
2.4.4. MITYP = Source type in element MI.
2.4.5. MTYPAD = Increment of MITYP for each of the NSEQ elements.

**** NOTE: A line with 5 0's is used to signal the end of every part of this data set.

DATA SET 23 --- POINT (WELL) SOURCE/SINK DATA FOR TRANSPORT SIMULATION

(NOTE: This data set is needed only if IMOD.NE.10)

Two parts are included. In each part, one line and four subsets of data records are required for this data set. The 1-st part is for chemical transport.

The 1-st line contains the following variable (A80)
DATNAM = Description of the data set.
A. subset 1: control parameters
1.1.1. NWNP = No. of well or point source/sink nodes.
1.1.2. NWPR = No. of well or point source/sink strength profiles.
1.1.3. NWDP = No. of data points in each of the NWPR profiles.
1.1.4. KWAI = Is well-source/sink profile to be input analytically?
0 = no,
1 = yes.
B. subset 2: source/sink profiles - This group of data is needed if and only if NWNP .GT. 0. For each sub-data-record, NWDP of the data group (TWSSF(J,I), WSSF(J,I,1), WSSF(J,I,2)) is required. If this sub-data-record can be fitted in a line, we will need NWPR lines.
1.2.1. TWSSF(J,I) = Time of J-th data point in I-th profile, (T).
1.2.2. WSSF(J,I,1) = Source/sink flow rate of the J-th data point in the I-th profile, (L**3/T); positive for source and negative for sink.

1.2.3. WSSF(J,I,2) = Source/sink concentration of the J-th data point in the I-th profile, (M/L**3).
C. subset 3: global source/sink element number - This group of data is needed if and only if NWNP.GT.0. NWNP data points are required for this record.
1.3.1. NPW(I) = Global node number of the I-th compressed point source/sink node.
D. subset 4: Source type assigned to each well - Usually one record per element. However, automatic generation can be made.
1.4.1. NI = Compressed point source/sink node number of the first node in a sequence.
1.4.2. NSEQ = NSEQ nodes will contain the source types that will be automatically generated.
1.4.3. NIAD = Increment of NI for each of the NSEQ nodes.
1.4.4. NITYP = Source type in node NI.
1.4.5. NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.
***** NOTE: Subset 4 has to be repeated (N0+N1A+N1B) times.

The 2-nd part is for temperature transport.
The 1-st line contains the following variable (A80)
DATNAM = Description of the data set.
A. subset 1: control parameters
2.1.1. NTWNP = No. of well or point source/sink nodes.
2.1.2. NTWPR = No. of well or point source/sink strength profiles.
2.1.3. NTWDP = No. of data points in each of the NTWPR profiles.
2.1.4. KTWAI = Is well-source/sink profile to be input analytically?
0 = no,
1 = yes.
B. subset 2: source/sink profiles - This group of data is needed if and only if NTWNP .GT. 0. For each sub-data-record, NTWDP of the data group (TTWSSF(J,I), WTSSF(J,I,1), WTSSF(J,I,2)) is required. If this sub-data-record can be fitted in a line, we will need NTWPR lines.
2.2.1. TTWSSF(J,I) = Time of J-th data point in I-th profile, (T).
2.2.2. WTSSF(J,I,1) = Source/sink flow rate of the J-th data point in the I-th profile, (L**3/T); positive for source and negative for sink.
2.2.3. WTSSF(J,I,2) = Source/sink temperature of the J-th data point in the I-th profile, (^{o}K).
C. subset 3: global source/sink element number - This group of data is needed if and only if NTWNP .GT.0. NTWNP data points are required for this record.
2.3.1. NTPW(I) = Global node number of the I-th compressed point source/sink node.
D. subset 4: Source type assigned to each well - Usually one record per element. However, automatic generation can be made.
2.4.1. NI = Compressed point source/sink node number of the first node in a sequence.
2.4.2. NSEQ = NSEQ nodes will contain the source types that will be automatically generated.
2.4.3. NIAD = Increment of NI for each of the NSEQ nodes.
2.4.4. NITYP = Source type in node NI.
2.4.5. NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.
**** NOTE: A record with 5 0's must be used to signal the end of every part of this data set.

DATA SET 24 --- RAINFALL/EVAPORATION-SEEPAGE BOUNDARY CONDITIONS

(NOTE: This data set is needed only if IMOD.NE.1)

One line and seven subsets of data records are required for this data set.

The 1-st line contains the following variable (A80)
DATNAM = Description of the data set.
A. subset 1: control parameters
1.1. NVES = No. of variable boundary element sides.
1.2. NVNP = No. of variable boundary nodal points.
1.3. NRPR = No. of rainfall profiles.
1.4. NRDP = No. of rainfall data points in each of the NRPR rainfall profiles.
1.5. KRAI = Is rainfall profile to be input analytically?
0 = no,

1 = yes.
B. subset 2: boundary profiles - This subset is required only when NVES is not 0. NRPR profiles are needed. For each profile, NRDP of the data pair (TRF(J,I),RF(J,I)) are required. If these data pairs can fit in a line, we will need NRPR of data lines.
2.1. TRF(J,I) = Time of the J-th data point in the I-th profile, (T).
2.2. RF(J,I) = Rainfall/evaporation rate of the J-th data point in the I-th profile,(L/T).
C. subset 3: boundary profile types assigned to each element. At most NVES records are needed. However, automatic generation can be made. For the I-th (I = 1, 2, ...,) record, it contains the following variables.
3.1. MI = Compressed VB element side of the first side in the sequence.
3.2. NSEQ = NSEQ sides will be generated automatically.
3.3. MIAD = Increment of MI for each of the NSEQ sides.
3.4. MITYP = Type of rainfall/evaporation profiles assigned to side MI.
3.5. MTYPAD = Increment of MITYP for each of the NSEQ sides.
**** NOTE: A line with 5 0's is used to signal the end of this data set.

D. subset 4: Specification of Rainfall/evaporation-seepage Sides. Normally, NVES records are required, one each for a variable boundary (VB) element side. However, if a group of rainfall/evaporation-seepage element sides appears in a regular pattern, automatic generation may be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
4.1. MI = Compressed VB element side number of the first element side in a sequence.
4.2. NSEQ = NSEQ subsequent VB element sides will be generated automatically.
4.3. MIAD = Increment of MI for each of the NSEQ subsequent VB element sides.
4.4. I1 = Global node number of the first node of VB element side MI.
4.5. I2 = Global node number of the second node of VB element side MI.
4.6. I3 = Global node number of the third node of VB element side MI.
4.7. I4 = Global node number of the fourth node of VB element side MI.
4.8. I1AD = Increment of I1 for each of the NSEQ subsequent VB element sides.
4.9. I2AD = Increment of I2 for each of the NSEQ subsequent VB element sides.
4.10. I3AD = Increment of I3 for each of the NSEQ subsequent VB element sides.
4.11. I4AD = Increment of I4 for each of the NSEQ subsequent VB element sides.
**** NOTE: A line with 11 0's must be used to signal the end of this subset.

E. subset 5: Global Node Number of All Compressed Variable Boundary (VB) Nodes. At most, NVNP records are needed for this subset, one each for NVNP variable boundary nodes. For I-th (I = 1, 2, ...,) Record, it contains the following 5 variables.
5.1. NI = Compressed VB node number of the first node in the sequence.
5.2. NSEQ = NSEQ nodes will be generated automatically.
5.3. NIAD = Increment of NI for each of the NSEQ nodes.
5.4. NODE = Global node number of node NI.
5.5. NODEAD = Increment of NODE for each of the NSEQ nodes.
**** NOTE: A line with 5 0's is used to signal the end of this data set.

F. subset 6: Ponding Depth Allowed in Each of NVNP Variable Boundary Nodes. Normally, NVNP records are needed, one for each of the NVNP nodes. However, if a group of nodes has a regular pattern of ponding depth, automatic generation is made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
6.1. NI = Compressed VB node number of the first node in a sequence.
6.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
6.3. NIAD = Increment of NI for each of the NSEQ subsequent nodes.
6.4. HCONNI = Ponding depth of node NI, (L).
6.5. HCONAD = Increment of HCONNI for each of the NSEQ nodes, (L).
**** NOTE: A line with 5 0's must be used to signal the end of this subdata set.

G. subset 7: Minimum Pressure Head Allowed in Each NVNP Variable Boundary Nodes. This subset is read-in similar to the above subdata set. For I-th (I = 1, 2, ...,) record, it contains the following variables.
7.1. NI = Compressed VB node number of the first node in a sequence.
7.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
7.3. NIAD = Increment of NI for each of the NSEQ subsequent nodes.

7.4. HMINNI = Minimum pressure head allow for node NI, (L).
7.5. HMINAD = Increment of HMINNI for each of the NSEQ nodes, (L).
**** NOTE: A line with 5 0's must be used to signal the end of this subdata set.

DATA SET 25 --- DIRICHLET BOUNDARY CONDITIONS FOR FLOW SIMULATION

(NOTE: This data set is needed only if IMOD.NE.1)

One line and four subsets of data records are required for this data set.

The 1-st line contains the following variable (A80)
   DATNAM = Description of the data set.
   A. subset 1: control parameters
       1.1. NDNP = No. of Dirichlet nodal points, should be .GE. 1.
       1.2. NDPR = No. of total Dirichlet-head profiles, should be .GE. 1.
       1.3. NDDP = No. of data points in each total head profiles, should be .GE. 1.
       1.4. KDAI = Is Dirichlet boundary value profile to be input analytically?
               0 = no,
               1 = yes
   B. subset 2: Dirichlet-head profiles - This sub-data-set is required only if NDNP is not 0. NDPR of profiles
       are needed. For each profile, NDDP of the data pair (THDBF(J,I),HDBF(J,I)) are needed. If these
       data pairs can fit in a line, we will need NDPR lines.
       2.1. THDBF(J,I) = Time of the J-th data point in the I-th profile, (T).
       2.2. HDBF(J,I) = Total head of the J-th data point in the I-th profile, (L).
   C. subset 3: Dirichlet nodes - One record is needed for this subdata set. The number of lines in this
       record depends on NDNP.
       3.1. NPDB(I) = Global node number of the I-th compressed Dirichlet node.
   D. subset 4: boundary profile type assign to each Dirichlet node - Normally one record per Dirichlet node,
       i. e. a total of NDNP records. However, if the Dirichlet nodes appear in a regular pattern, automatic
       generation may be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
       1. NI = Compressed Dirichlet node number of the first node in the sequence.
       2. NSEQ = NSEQ subsequent Dirichlet nodes will be generated automatically.
       3. NAD = Increment of NI for each of the NSEQ nodes.
       4. NITYP = Type of total head profile for node NI and NSEQ subsequent nodes.
       5. NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.
**** NOTE: A line with 5 0's must be used to signal the end of this sub-data set.

DATA SET 26 --- CAUCHY BOUNDARY CONDITIONS FOR FLOW SIMULATIONS

(NOTE: This data set is needed only if IMOD.NE.1)

One line and five subsets of data records are required for this data set.

The 1-st line contains the following variable (A80)
   DATNAM = Description of the data set.
   A. subset 1: control parameters
       1.1. NCES = No. of Cauchy boundary element sides.
       1.2. NCNP = No. of Cauchy nodal points.
       1.3. NCPR = No. of Cauchy-flux profiles.
       1.4. NCDP = No. of data points in each of the NCPR Cauchy-flux profiles.
       1.5. KCAI = Is Cauchy flux profile to be input analytically, 0 = no, 1 = yes
   B. subset 2: prescribed Cauchy-flux profiles - This sub-data-set is required only if NCES is not 0. NCPR
       of profiles are needed. For each profile, NCDP of the data pair (TQCBF(J,I),QCBF(J,I)) are needed.
       If these data pairs can fit in a line, we will need NDPR lines.
       2.1. TQCBF(J,I) = Time of the J-th data point in the I-th profile, (T).
       2.2. QCBF(J,I) = Normal Cauchy flux of the J-th data point in the I-th profile, (L**3/T/L**2); positive out
       from the region, negative into the region.

C. subset 3: type of Cauchy flux profiles assigned to each of all NCES sides. At most NCES records are needed. However, automatic generation can be made. For I-th (I = 1, 2,..) record, it contains the following variables.
    3.1. MI = Compressed Cauchy side number of the first side in the sequence.
    3.2. NSEQ = NSEQ sides will be generated automatically.
    3.3. MIAD = Increment of MI for each of the NSEQ sides.
    3.4. MITYP = Type of Cauchy flux profile assigned to side MI.
    3.5. MTYPAD = Increment of MITYP for each of the NSEQ sides.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

D. Record subset 4: Cauchy boundary element sides - Normally, NCES records are required, one each for a Cauchy boundary element side. However, if a group of Cauchy boundary element sides appears in a regular pattern, automatic generation may be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
    4.1. MI = Compressed CB element side number of the first element-side in a sequence.
    4.2. NSEQ = NSEQ subsequent CB element sides will be generated automatically.
    4.3. MIAD = Increment of MI for each of the NSEQ subsequent CB element sides.
    4.4. I1 = Global node number of the first node on the CB element side MI.
    4.5. I2 = Global node number of the second node on the CB element side MI.
    4.6. I3 = Global node number of the third node on the CB element side MI.
    4.7. I4 = Global node number of the fourth node on the CB element side MI.
    4.8. I1AD = Increment of I1 for each of the NSEQ subsequent CB element sides.
    4.9. I2AD = Increment of I2 for each of the NSEQ subsequent CB element sides.
    4.10. I3AD = Increment of I3 for each of the NSEQ subsequent CB element sides.
    4.11. I4AD = Increment of I4 for each of the NSEQ subsequent CB element sides.

**** NOTE: A line with 11 0's is used to end this data set input.

E. subset 5: global nodal number of all Cauchy boundary nodes -Usually NCNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
    5.1. NI = Compressed Cauchy boundary node number of the first node in a sequence.
    5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
    5.3. NIAD = Increment for NI for each of the NSEQ nodes.
    5.4. NODE = Global nodal number of the node NI.
    5.5. NODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's is used to signal end of this subdata set.

DATA SET 27 --- NEUMANN BOUNDARY CONDITIONS FOR FLOW SIMULATIONS

(NOTE: This data set is needed only if IMOD.NE.1)

One line and five subsets of data records are required for this data set.

The 1-st line contains the following variable (A80)
    DATNAM = Description of the data set.
    A. subset 1: control parameters
        1.1. NNES = No. of Neumann boundary element sides.
        1.2. NNNP = No. of Neumann nodal points.
        1.3. NNPR = No. of Neumann flux profiles.
        1.4. NNDP = No. of data points in each of the NNPR Neumann-flux profiles.
        1.5. KNAI = Is Neumann flux profile to be input analytically?
                0 = no,
                1 = yes
    B. subset 2: prescribed Neumann-flux profiles - This sub-data-set is required only if NNES is not 0. NNPR of profiles are needed. For each profile, NNDP of the data pair (TQNBF(J,I), QNBF(J,I)) are needed. If these data pairs can fit in a line, we will need NDPR lines.
        2.1. TQNBF(J,I) = Time of the J-th data point in the I-th profile, (T).
        2.2. QNBF(J,I) = Normal Neumann flux of the J-th data point in the I-th profile, (L**3/T/L**2); positive out from the region, negative into the region.

C. subset 3: type of Neumann flux profiles assigned to each of all NNES sides. At most NNES records are needed. However, automatic generation can be made. For I-th (I = 1, 2,...) record, it contains the following variables.
    3.1. MI = Compressed Neumann side number of the first side in the sequence.
    3.2. NSEQ = NSEQ sides will be generated automatically.
    3.3. MIAD = Increment of MI for each of the NSEQ sides.
    3.4. MITYP = Type of Neumann flux profile assigned to side MI.
    3.5. MTYPAD = Increment of MITYP for each of the NSEQ sides.

**** NOTE: A line with 5 0's is used to signal the end of this data set.

D. subset 4: Neumann boundary element sides - Normally, NNES records are required, one each for a Neumann boundary element side. However, if a group of Neumann boundary element sides appears in a regular pattern, automatic generation may be made. For I-th (I = 1, 2, ...,) record, it contains the following variables.
    4.1. MI = Compressed NB element side number of the first side in sequence.
    4.2. NSEQ = NSEQ subsequent NB element sides will be generated automatically.
    4.3. MIAD = Increment of MI for each of the NSEQ subsequent NB element sides.
    4.4. I1 = Global node number of the first node on the NB element side MI.
    4.5. I2 = Global node number of the second node on NB element side MI.
    4.6. I3 = Global node number of the third node on the NB element side MI.
    4.7. I4 = Global node number of the fourth node on the NB element side MI.
    4.8. I1AD = Increment of I1 for each of the NSEQ subsequent NB element sides.
    4.9. I2AD = Increment of I2 for each of the NSEQ subsequent NB element sides.
    4.10. I3AD = Increment of I3 for each of the NSEQ subsequent NB element sides.
    4.11. I4AD = Increment of I4 for each of the NSEQ subsequent NB element sides.

**** NOTE: A line with 11 0's is used to end this data set input.

E. subset 5: global nodal number of all Neumann boundary nodes -Usually NNNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
    5.1. NI = Compressed Neumann boundary node number of the first node in a sequence.
    5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
    5.3. NIAD = Increment for NI for each of the NSEQ nodes.
    5.4. NODE = Global nodal number of the node NI.
    5.5. NODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.

DATA SET 28 --- RUN-IN/FLOW-OUT (VARIABLE) BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS (NOTE: This data set is needed only if IMOD.NE.10)

Two parts are included. In each part, one line and five subsets of data records are required for this data set. The 1-st part is for chemical transport.

The 1-st line contains the following variable (A80)
    DATNAM = Description of the data set.
A. subset 1: control parameters
    1.1.1. NVES = No. of variable boundary element sides.
    1.1.2. NVNP = No. of variable boundary nodal points.
    1.1.3. NRPR = No. of incoming fluid concentration profiles to be applied to variable boundary element sides.
    1.1.4. NRDP = No. of data points in each of the NRPR profiles.
    1.1.5. KRAI = Is incoming concentration profile to be input analytically?
        0 = no,
        1 = yes.
B. subset 2: variable boundary flux profile - NRPR records are needed. Each record contains NRDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and run-in flow-out concentrations, respectively, as follows:
    1.2.1. TCRSF(J,I) = Time of the J-th data point on the I-th run-in concentration profile, (T),
    1.2.2. CRSF(J,I) = Concentration of the J-th data point on the I-th profile, (M/L**3).

C. subset 3: Run-in concentration type assigned to each of all NVES sides. Usually one record per variable element side. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
    1.3.1. MI = Compressed VB element side of the first side in a sequence.
    1.3.2. NSEQ = NSEQ subsequent sides will be generated automatically.
    1.3.3. MIAD = Increment of MI for each of NSEQ subsequent sides.
    1.3.4. MITYP = Type of concentration profile assigned to side MI.
    1.3.5. MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides.
**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

***** NOTE: Subset 3 has to be repeated (N0+N1A+N1B) times.

D. subset 4: Specification of run-in boundary element sides - Normally, NVES records are required, one each for a VB element side. However, if a group of VB element sides appears in a regular pattern, automatic generation may be made. Each record contains 11 variables and is FREE-FORMATTED.
    1.4.1. MI = Compressed VB element side number of the first side in a sequence.
    1.4.2. NSEQ = NSEQ subsequent VB element sides will be generated automatically.
    1.4.3. MIAD = Increment of MI for each of the NSEQ subsequent VB element sides.
    1.4.4. I1 = Global node number of the first node of VB element side MI.
    1.4.5. I2 = Global node number of the second node of VB element side MI.
    1.4.6. I3 = Global node number of the third node of VB element side MI.
    1.4.7. I4 = Global node number of the fourth node of VB element side MI.
    1.4.8. I1AD = Increment of I1 for each of the NSEQ subsequent VB element sides.
    1.4.9. I2AD = Increment of I2 for each of the NSEQ subsequent VB element sides.
    1.4.10. I3AD = Increment of I3 for each of the NSEQ subsequent VB element sides.
    1.4.11. I4AD = Increment of I4 for each of the NSEQ subsequent VB element sides.
**** NOTE: A record with 11 0's is used to signal the end of this subdata set.

E. subset 5: global nodal number of all run-in flow-out boundary nodes- Usually NVNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
    1.5.1. NI = Compressed VB node number of the first node in a sequence.
    1.5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
    1.5.3. NIAD = Increment for NI for each of the NSEQ nodes.
    1.5.4. NODE = Global nodal number of the node NI.
    1.5.5. NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.
**** NOTE: A record with 5 0's is used to signal end of this subdata set.

The 2-nd part is for temperature transport.
The 1-st line contains the following variable (A80)
DATNAM = Description of the data set.
A. subset 1: control parameters
    2.1.1. NTVES = No. of variable boundary element sides.
    2.1.2. NTVNP = No. of variable boundary nodal points.
    2.1.3. NTRPR = No. of incoming fluid concentration profiles to be applied to variable boundary element sides.
    2.1.4. NTRDP = No. of data points in each of the NTRPR profiles.
    2.1.5. KTRAI = Is incoming concentration profile to be input analytically?
        0 = no,
        1 = yes.
B. subset 2: variable boundary flux profile - NTRPR records are needed. Each record contains NTRDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and run-in flow-out temperature, respectively, as follows:
    2.2.1. TTCRSF(J,I) = Time of the J-th data point on the I-th run-in concentration profile, (T).
    2.2.2. CTRSF(J,I) = Temperature of the J-th data point on the I-th profile, (^{o}K).
C. subset 3: Run-in temperature type assigned to each of all NTVES sides. Usually one record per variables element side. However, automatic generation can be made. Each record contains 5 variable and is FREE-FORMATTED.
    2.3.1. MI = Compressed VB element side of the first side in a sequence.

2.3.2. NSEQ = NSEQ subsequent sides will be generated automatically.
2.3.3. MIAD = Increment of MI for each of NSEQ subsequent sides.
2.3.4. MITYP = Type of concentration profile assigned to side MI.
2.3.5. MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides.
**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

     D. subset 4: Specification of run-in boundary element sides - Normally, NTVES records are required, one each for a VB element side. However, if a group of VB element sides appears in a regular pattern, automatic generation may be made. Each record contains 11 variables and is FREE-FORMATTED.
2.4.1. MI = Compressed VB element side number of the first side in a sequence.
2.4.2. NSEQ = NSEQ subsequent VB element sides will be generated automatically.
2.4.3. MIAD = Increment of MI for each of the NSEQ subsequent VB element sides.
2.4.4. I1 = Global node number of the first node of VB element side MI.
2.4.5. I2 = Global node number of the second node of VB element side MI.
2.4.6. I3 = Global node number of the third node of VB element side MI.
2.4.7. I4 = Global node number of the fourth node of VB element side MI.
2.4.8. I1AD = Increment of I1 for each of the NSEQ subsequent VB element sides.
2.4.9. I2AD = Increment of I2 for each of the NSEQ subsequent VB element sides.
2.4.10. I3AD = Increment of I3 for each of the NSEQ subsequent VB element sides.
2.4.11. I4AD = Increment of I4 for each of the NSEQ subsequent VB element sides.
**** NOTE: A record with 11 0's is used to signal the end of this subdata set.

     E. subset 5: global nodal number of all run-in flow-out boundary nodes- Usually NTVNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
2.5.1. NI = Compressed VB node number of the first node in a sequence.
2.5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
2.5.3. NIAD = Increment for NI for each of the NSEQ nodes.
2.5.4. NODE = Global nodal number of the node NI.
2.5.5. NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.
**** NOTE: A record with 5 0's is used to signal end of this subdata set.

DATA SET 29 --- DIRICHLET BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS (NOTE: This data set is needed only if IMOD.NE.10)

     Two parts are included in this data set. In each part, one line and four subsets of data records are required for this data set. The 1-st part is for chemical transport.

  The 1-st line contains the following variable (A80)
     DATNAM = Description of the data set.
     A. subset 1: control parameters
1.1.1. NDNP = No. of Dirichlet nodes, should be .GE. 1.
1.1.2. NDPR = No. of Dirichlet profiles, should be .GE. 1.
1.1.3. NDDP = No. of data points in each profile, should be .GE. 2.
1.1.4. KDAI = Is Dirichlet boundary value profile to be input analytically?
       0 = no,
       1 = yes.
     B. subset 2: Dirichlet-concentration profiles - NDPR records are needed. Each record contains NDDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Dirichlet concentrations, respectively, as follows:
1.2.1. TCDBF(J,I) = Time of J-th data point in I-th Dirichlet-concentration profile, (T).
1.2.2. CDBF(J,I) = Concentration of J-th data point in I-th Dirichlet-concentration profile, (M/L**3).
     C. subset 3: global node number of compressed Dirichlet nodes - One record is needed for this subdata set, which contains NDNP variables and is FREE-FORMATTED.
1.3.1. NPDB(I) = Global node number of the I-th compressed Dirichlet node.
     D. subset 4: Dirichlet concentration types assigned to Dirichlet nodes - Normally one record per Dirichlet node, i.e., a total of NDNP records, is needed. However, if the Dirichlet nodes appear in a regular

pattern, automatic generation may be made. Each record contains 5 variables and is FREE-FORMATTED.

   1.4.1. NI = Compressed Dirichlet node number of the first node in the sequence.

   1.4.2. NSEQ = NSEQ nodes will contain the Dirichlet concentration types that will be automatically generated.

   1.4.3. NIAD = Increment of NI for each of the NSEQ nodes.

   1.4.4. NITYP = Dirichlet concentration type in node NI.

   1.4.5. NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.

***** NOTE: Subset 4 has to be repeated (N0+N1A+N1B) times.

   The 2-nd part is for temperature transport.

   The 1-st line contains the following variable (A80)

   DATNAM = Description of the data set.

   A.   subset 1: control parameters

   2.1.1. NTDNP = No. of Dirichlet nodes, should be .GE. 1.

   2.1.2. NTDPR = No. of Dirichlet profiles, should be .GE. 1.

   2.1.3. NTDDP = No. of data points in each profile, should be .GE. 2.

   2.1.4. KTDAI = Is Dirichlet boundary value profile to be input analytically?

            0 = no,

            1 = yes.

   B.   subset 2: Dirichlet-temperature profiles - NTDPR records are needed. Each record contains NTDDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Dirichlet temperature, respectively, as follows:

   2.2.1. TTCDBF(J,I) = Time of the J-th data point in the I-th Dirichlet profile, (T).

   2.2.2. CTDBF(J,I) = Temperature of J-th data point in I-th Dirichlet profile, ($^{\circ}$K).

   C.   subset 3: global node number of compressed Dirichlet nodes - One record is needed for this subdata set, which contains NTDNP variables and is FREE-FORMATTED.

   2.3.1. NTPDB(I) = Global node number of the I-th compressed Dirichlet node.

   D.   subset 4: Dirichlet temperature types assigned to Dirichlet nodes - Normally one record per Dirichlet node, i.e., a total of NTDNP records, is needed. However, if the Dirichlet nodes appear in a regular pattern, automatic generation may be made. Each record contains 5 variables and is FREE-FORMATTED.

   2.4.1. NI = Compressed Dirichlet node number of the first node in the sequence.

   2.4.2. NSEQ = NSEQ nodes will contain the Dirichlet concentration types that will be automatically generated.

   2.4.3. NIAD = Increment of NI for each of the NSEQ nodes.

   2.4.4. NITYP = Dirichlet concentration type in node NI.

   2.4.5. NTYPAD = Increment of NITYP for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's must be used to signal the end of every part of this data set.

DATA SET 30 --- CAUCHY BOUNDARY CONDITIONS FOR TRANSPORT SIMULATION

   (NOTE: This data set is needed only if IMOD.NE.10)

   Two parts are included. In each part, one line and five subsets of data records are required for this data set. The 1-st part is for chemical transport.

   The 1-st line contains the following variable (A80)

   DATNAM = Description of the data set.

   A.   subset 1: control parameters

   1.1.1. NCES = No. of Cauchy element sides.

   1.1.2. NCNP = No. of Cauchy nodal points.

   1.1.3. NCPR = No. of Cauchy-flux profiles.

   1.1.4. NCDP = No. of data points on each Cauchy-flux profile.

   1.1.5. KCAI = Is Cauchy flux profile to be input analytically?

            0 = no,

            1 = yes.

B. subset 2: Cauchy flux profiles - NCPR records are needed. Each record contains NCDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Cauchy flux, respectively, as follows:

    1.2.1. TQCBF(J,I) = Time of the J-th data point in the I-th Cauchy flux profile, (T).

    1.2.2. QCBF(J,I) = Value of Cauchy flux of the J-th data point in the I-th Cauchy flux profile, (M/T/L**2).

C. subset 3: Cauchy flux type assigned to each of all NCES sides -Usually one record per Cauchy element side. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.

    1.3.1. MI = Compressed Cauchy boundary element side of the first side in a sequence.

    1.3.2. NSEQ = NSEQ subsequent sides will be generated automatically.

    1.3.3. MIAD = Increment of MI for each of NSEQ subsequent sides.

    1.3.4. MITYP = Type of Cauchy flux profile assigned to side MI.

    1.3.5. MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides.

**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

***** NOTE: Subset 3 has to be repeated (N0+N1A+N1B) times.

D. Record subset 4: specification of Cauchy boundary element sides -Normally, NCES records are required, one each for a Cauchy boundary element side. However, if a group of Cauchy element sides appears in a regular pattern, automatic generation may be made. Each record contains 11 variables and is FREE-FORMATTED.

    1.4.1. MI = Compressed CB element side number of the first element side in a sequence.

    1.4.2. NSEQ = NSEQ subsequent CB element sides will be generated automatically.

    1.4.3. MIAD = Increment of MI for each of the NSEQ subsequent CB element sides.

    1.4.4. I1 = Global node number of the first node of CB element side MI.

    1.4.5. I2 = Global node number of the second node of CB element side MI.

    1.4.6. I3 = Global node number of the third node of CB element side MI.

    1.4.7. I4 = Global node number of the fourth node of CB element side MI.

    1.4.8. I1AD = Increment of I1 for each of the NSEQ subsequent CB element sides.

    1.4.9. I2AD = Increment of I2 for each of the NSEQ subsequent CB element sides.

    1.4.10. I3AD = Increment of I3 for each of the NSEQ subsequent CB element sides.

    1.4.11. I4AD = Increment of I4 for each of the NSEQ subsequent CB element sides.

**** NOTE: A record with 11 0's is used to signal the end of this subdata set.

E. subset 5: global nodal number of all Cauchy boundary nodes -Usually NCNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.

    1.5.1. NI = Compressed Cauchy boundary node number of the first node in a sequence.

    1.5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.

    1.5.3. NIAD = Increment for NI for each of the NSEQ nodes.

    1.5.4. NODE = Global nodal number of the node NI.

    1.5.5. KODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.

**** NOTE: A record with 11 0's is used to signal the end of this subdata set.

  The 2-nd part is for temperature transport.

    The 1-st line contains the following variable (A80)

      DATNAM = Description of the data set.

A. subset 1: control parameters

    2.1.1. NTCES = No. of Cauchy element sides.

    2.1.2. NTCNP = No. of Cauchy nodal points.

    2.1.3. NTCPR = No. of Cauchy-flux profiles.

    2.1.4. NTCDP = No. of data points on each Cauchy-flux profile.

    2.1.5. KTCAI = Is Cauchy flux profile to be input analytically?

        0 = no,

        1 = yes.

B. subset 2: Cauchy flux profiles - NTCPR records are needed. Each record contains NTCDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Cauchy flux, respectively, as follows:

2.2.1. TTQCBF(J,I) = Time of the J-th data point in the I-th Cauchy flux profile, (T).
2.2.2. QTCBF(J,I) = Value of Cauchy flux of the J-th data point in the I-th Cauchy flux profile, (M/T**3).
C. subset 3: Cauchy flux type assigned to each of all NTCES sides -Usually one record per Cauchy element side. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
2.3.1. MI = Compressed Cauchy boundary element side of the first side in a sequence.
2.3.2. NSEQ = NSEQ subsequent sides will be generated automatically.
2.3.3. MIAD = Increment of MI for each of NSEQ subsequent sides.
2.3.4. MITYP = Type of Cauchy flux profile assigned to side MI.
2.3.5. MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides.

**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

D. Record subset 4: specification of Cauchy boundary element sides -Normally, NTCES records are required, one each for a Cauchy boundary element side. However, if a group of Cauchy element sides appears in a regular pattern, automatic generation may be made. Each record contains 11 variables and is FREE-FORMATTED.
2.4.1. MI = Compressed CB element side number of the first element side in a sequence.
2.4.2. NSEQ = NSEQ subsequent CB element sides will be generated automatically.
2.4.3. MIAD = Increment of MI for each of the NSEQ subsequent CB element sides.
2.4.4. I1 = Global node number of the first node of CB element side MI.
2.4.5. I2 = Global node number of the second node of CB element side MI.
2.4.6. I3 = Global node number of the third node of CB element side MI.
2.4.7. I4 = Global node number of the fourth node of CB element side MI.
2.4.8. I1AD = Increment of I1 for each of the NSEQ subsequent CB element sides.
2.4.9. I2AD = Increment of I2 for each of the NSEQ subsequent CB element sides.
2.4.10. I3AD = Increment of I3 for each of the NSEQ subsequent CB element sides.
2.4.11. I4AD = Increment of I4 for each of the NSEQ subsequent CB element sides.

**** NOTE: A record with 11 0's is used to signal the end of this subdata set.

E. subset 5: global nodal number of all Cauchy boundary nodes -Usually NTCNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
2.5.1. NI = Compressed Cauchy boundary node number of the first node in a sequence.
2.5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
2.5.3. NIAD = Increment for NI for each of the NSEQ nodes.
2.5.4. NODE = Global nodal number of the node NI.
2.5.5. NODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's is used to signal end of this subdata set.

DATA SET 31--- NEUMANN BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

(NOTE: This data set is needed only if IMOD.NE.10)

Two parts are included. In each part, one line and five subsets of data records are required for this data set. The 1-st part is for chemical transport.

The 1-st line contains the following variable (A80)
DATNAM = Description of the data set.
A. subset 1: control parameters
1.1.1. NNES = No. of Neumann element sides.
1.1.2. NNNP = No. of Neumann nodal points.
1.1.3. NNPR = No. of Neumann-flux profiles.
1.1.4. NNDP = No. of data points on each Neumann-flux profile.
1.1.5. KSAI = Is Neumann flux profile to be input analytically?
0 = no,
1 = yes.

B. subset 2: Neumann flux profiles - NNPR records are needed. Each record contains NNDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Neumann flux, respectively, as follows:

1.2.1. TQNBF(J,I) = Time of the J-th data point in the I-th Neumann flux profile, (T).

1.2.2. QNBF(J,I) = Value of Neumann flux of the J-th data point in the I-th Neumann flux profile, (M/T/L**2).

C. subset 3: Neumann flux type assigned to each of all NNES sides -Usually one record per Neumann element side. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.

1.3.1. MI = Compressed Neumann boundary element side of the first side in a sequence.

1.3.2. NSEQ = NSEQ subsequent sides will be generated automatically.

1.3.3. MIAD = Increment of MI for each of NSEQ subsequent sides.

1.3.4. MITYP = Type of Neumann flux profile assigned to side MI.

1.3.5. MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides.

**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

***** NOTE: Subset 3 has to be repeated (N0+N1A+N1B) times.

D. subset 4: specification of Neumann boundary element sides -Normally, NNES records are required, one each for a Neumann boundary element side. However, if a group of Neumann element sides appears in a regular pattern, automatic generation may be made. Each record contains 11 variables and is FREE-FORMATTED.

1.4.1. MI = Compressed NB element side number of the first element side in a sequence.

1.4.2. NSEQ = NSEQ subsequent NB element sides will be generated automatically.

1.4.3. MIAD = Increment of MI for each of the NSEQ subsequent NB element sides.

1.4.4. I1 = Global node number of the first node of NB element side MI.

1.4.5. I2 = Global node number of the second node of NB element side MI.

1.4.6. I3 = Global node number of the third node of NB element side MI.

1.4.7. I4 = Global node number of the fourth node of NB element side MI.

1.4.8. I1AD = Increment of I1 for each of the NSEQ subsequent NB element sides.

1.4.9. I2AD = Increment of I2 for each of the NSEQ subsequent NB element sides.

1.4.10. I3AD = Increment of I3 for each of the NSEQ subsequent NB element sides.

1.4.11. I4AD = Increment of I4 for each of the NSEQ subsequent NB element sides.

**** NOTE: A record with 11 0's is used to signal the end of this subdata set.

E. subset 5: global nodal number of all Neumann boundary nodes -Usually NNNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.

1.5.1. NI = Compressed Neumann boundary node number of the first node in a sequence.

1.5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.

1.5.3. NIAD = Increment for NI for each of the NSEQ nodes.

1.5.4. NODE = Global nodal number of the node NI.

1.5.5. NODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.

**** NOTE: A record with 5 0's is used to signal end of this subdata set.

The 2-nd part is for temperature transport.

The 1-st line contains the following variable (A80)

DATNAM = Description of the data set.

A. subset 1: control parameters

2.1.1. NTNES = No. of Neumann element sides.

2.1.2. NTNNP = No. of Neumann nodal points.

2.1.3. NTNPR = No. of Neumann-flux profiles.

2.1.4. NTNDP = No. of data points on each Neumann-flux profile.

2.1.5. KTSAI = Is Neumann flux profile to be input analytically?

0 = no,

1 = yes.

B. subset 2: Neumann flux profiles - NTNPR records are needed. Each record contains NTNDP data points and is FREE-FORMATTED. Each data point has 2 numbers representing the time and Neumann flux, respectively, as follows:

2.2.1. TTQNBF(J,I) = Time of the J-th data point in the I-th Neumann flux profile, (T).
2.2.2. QTNBF(J,I) = Value of Neumann flux of the J-th data point in the I-th Neumann flux profile, (M/T**3).
    C. subset 3: Neumann flux type assigned to each of all NTNES sides -Usually one record per Neumann element side. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
2.3.1. MI = Compressed Neumann boundary element side of the first side in a sequence.
2.3.2. NSEQ = NSEQ subsequent sides will be generated automatically.
2.3.3. MIAD = Increment of MI for each of NSEQ subsequent sides.
2.3.4. MITYP = Type of Neumann flux profile assigned to side MI.
2.3.5. MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides.
**** NOTE: A record with 5 0's must be used to signal the end of this subdata set.

    D. subset 4: specification of Neumann boundary element sides -Normally, NTNES records are required, one each for a Neumann boundary element side. However, if a group of Neumann element sides appears in a regular pattern, automatic generation may be made. Each record contains 11 variables and is FREE-FORMATTED.
2.4.1. MI = Compressed NB element side number of the first element side in a sequence.
2.4.2. NSEQ = NSEQ subsequent NB element sides will be generated automatically.
2.4.3. MIAD = Increment of MI for each of the NSEQ subsequent NB element sides.
2.4.4. I1 = Global node number of the first node of NB element side MI.
2.4.5. I2 = Global node number of the second node of NB element side MI.
2.4.6. I3 = Global node number of the third node of NB element side MI.
2.4.7. I4 = Global node number of the fourth node of NB element side MI.
2.4.8. I1AD = Increment of I1 for each of the NSEQ subsequent NB element sides.
2.4.9. I2AD = Increment of I2 for each of the NSEQ subsequent NB element sides.
2.4.10. I3AD = Increment of I3 for each of the NSEQ subsequent NB element sides.
2.4.11. I4AD = Increment of I4 for each of the NSEQ subsequent NB element sides.
**** NOTE: A record with 11 0's is used to signal the end of this subdata set.

    E. subset 5: global nodal number of all Neumann boundary nodes -Usually NTNNP records are needed for this subdata set. However, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
2.5.1. NI = Compressed Neumann boundary node number of the first node in a sequence.
2.5.2. NSEQ = NSEQ subsequent nodes will be generated automatically.
2.5.3. NIAD = Increment for NI for each of the NSEQ nodes.
2.5.4. NODE = Global nodal number of the node NI.
2.5.5. NODEAD = Increment of the global nodal number for each of the NSEQ subsequent nodes.
**** NOTE: A record with 5 0's is used to signal end of this subdata set.

DATA SET 32 --- NODE CHEMICAL INFORMATION PRINTOUT CONTROL

(NOTE: This data set is needed only if IMOD.NE.10)

There are 3 lines in this data set:

The 1-st line contains the following variable (A80)
    1. DATNAM = Description of the data set.
The 2-nd line contains the following variable (FREE FORMAT)
    1. NCPRT = No. of global nodes of which the chemical information is to be printed.
The 3-rd line contains the following NCPRT variables (FREE FORMAT)
    1. IK(1) = Global node no. of the 1-st node of which the chemical information is to be printed.
    .
    .
    NCPRT. IK(NCPRT) = Global node no. of the NCPRT-th node of which the chemical information is to be printed.

DATA SET 33 --- HYDROLOGICAL VARIABLES

This data set is needed if and only if KVI .LE. 0. When KVI .LE. 0, one line and four groups of data are needed, one group for the velocity field and the other group for the moisture content.

The 1-st line contains the following variable (A80)
1. DATNAM = Description of the data set.
A. subset 1: velocity field in x-direction - Usually NNP records are needed. However, if velocity appears in a regular pattern, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
1.1. NI = Node number of the first node in a sequence.
1.2. NSEQ = NSEQ subsequent nodes will be automatically generated.
1.3. NIAD = Increment of node number in each of the NSEQ subsequent nodes.
1.4. VXNI = x-velocity component at node NI, (L/T).
1.5. VXAD = Increment of VXNI for each of the NSEQ subsequent nodes, (L/T).

**** NOTE: A record with 9 0's is used to signal the end of this subdata set.

B. subset 2: velocity field in y-direction - Usually NNP records are needed. However, if velocity appears in a regular pattern, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
2.1. NI = Node number of the first node in a sequence.
2.2. NSEQ = NSEQ subsequent nodes will be automatically generated.
2.3. NIAD = Increment of node number in each of the NSEQ subsequent nodes.
2.4. VYNI = y-velocity component at node NI, (L/T).
2.5. VYAD = Increment of VYNI for each of the NSEQ subsequent nodes, (L/T).

**** NOTE: A record with 9 0's is used to signal the end of this subdata set.

C. subset 3: velocity field in z-direction - Usually NNP records are needed. However, if velocity appears in a regular pattern, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
3.1. NI = Node number of the first node in a sequence.
3.2. NSEQ = NSEQ subsequent nodes will be automatically generated.
3.3. NIAD = Increment of node number in each of the NSEQ subsequent nodes.
3.4. VZNI = z-velocity component at node NI, (L/T).
3.5. VZAD = Increment of VXNI for each of the NSEQ subsequent nodes, (L/T).

**** NOTE: A record with 9 0's is used to signal the end of this subdata set.

D. subset 4: moisture content field - Usually, NEL records are needed. However, if moisture content appears in a regular pattern, automatic generation can be made. Each record contains 5 variables and is FREE-FORMATTED.
4.1. MI = Element number of the first element in a sequence.
4.2. NSEQ = NSEQ subsequent elements will be automatically generated.
4.3. MIAD = Increment of MI for each of NSEQ subsequent elements.
4.4. THNI = Moisture content of element NI, (Decimal point).
4.5. THNIAD = Increment of THNI for NSEQ subsequent elements, (Decimal point).

**** NOTE: A record with 5 0's is used to signal the end of this subdata set.

DATA SET 34. END OF JOB

If another problem is to be run, then input begins again with input data set 1. If termination of the job is desired, a blank card must be inserted at the end of the data set.

Appendix B
PROGRAM DESCRIPTION

In this appendix, all the subroutines/functions, for both Approach 1 and Approach 2, in 3DHYDROGEOCHEM are described. Figures B.1 through B.12 show the connection among subroutines/functions.
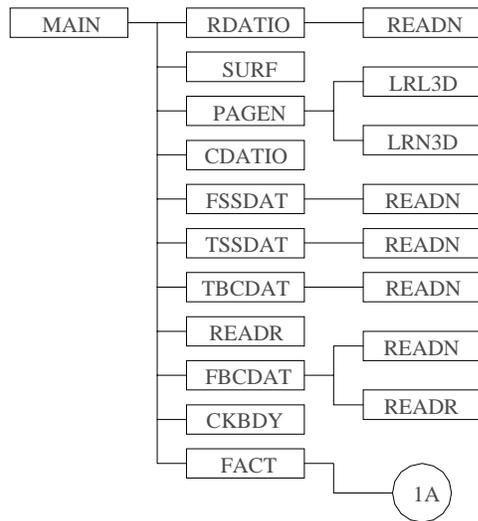

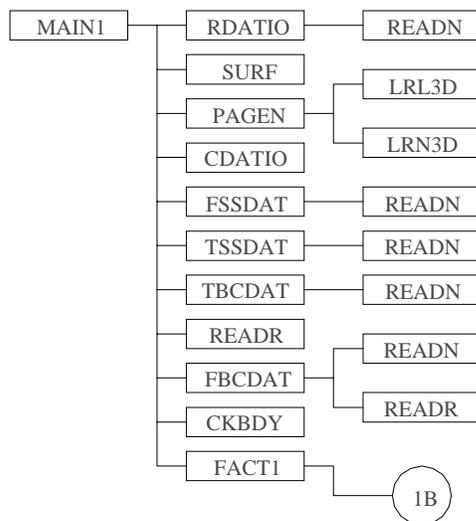**Figure B.1 Subroutine/function connection for Approach 2 (Part 1).**

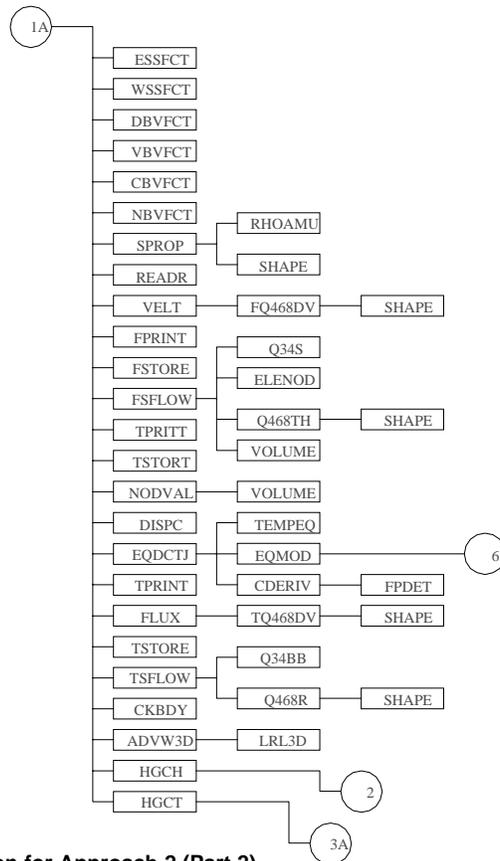**Figure B.2 Subroutine/function connection for Approach 1 (Part 1).**

Figure B.3 Subroutine/function connection for Approach 2 (Part 2).

Figure B.4 Subroutine/function connection forApproach 1 (Part 2).

Figure B.5 Subroutine/function connection for Approaches 1 and 2 (Part 3).

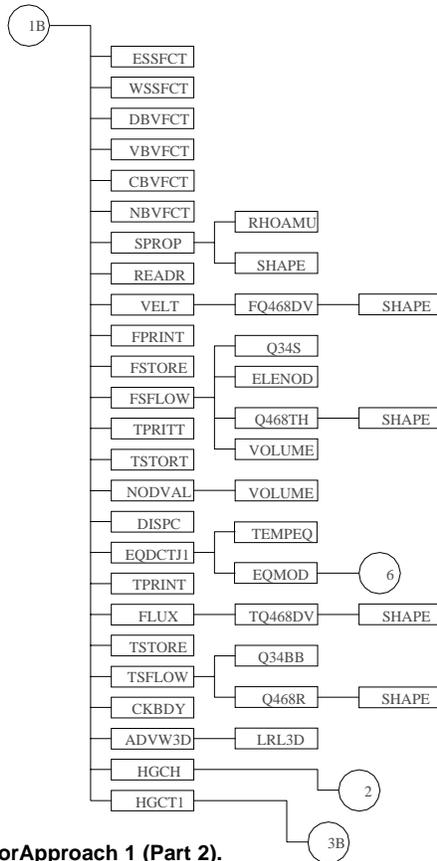**Figure B.6 Subroutine/function connection for Approach 2 (Part 4).**

Figure B.7 Subroutine/function connection for Approach 1 (Part 4).

Figure B.8 Subroutine/function connection for Approach 2 (Part 5).

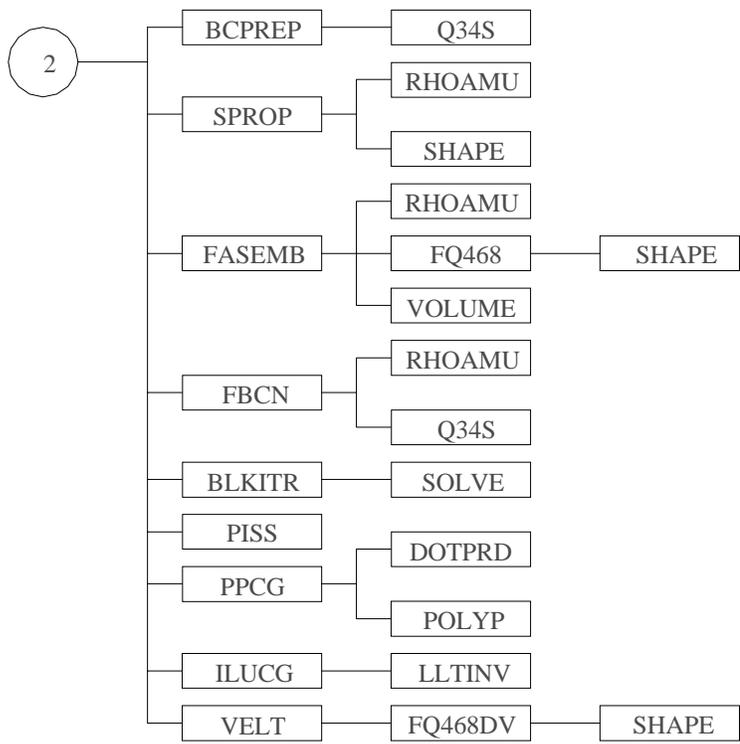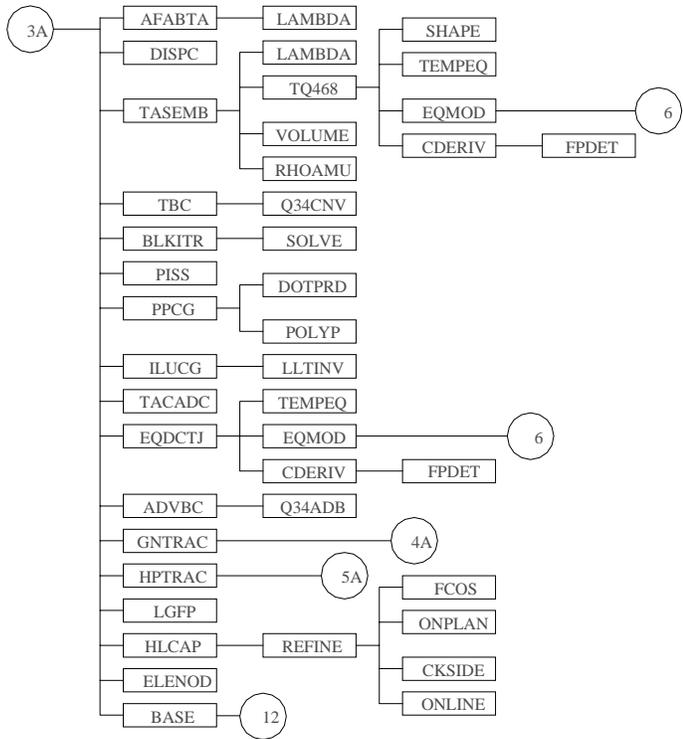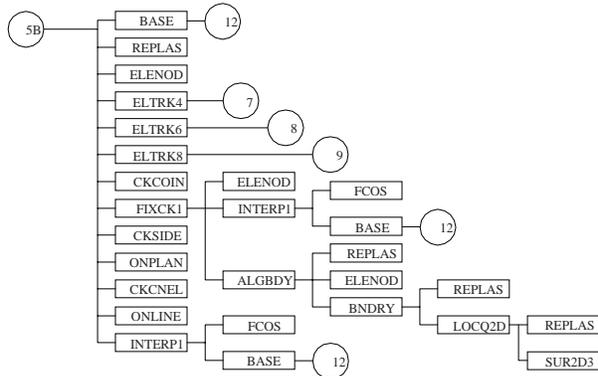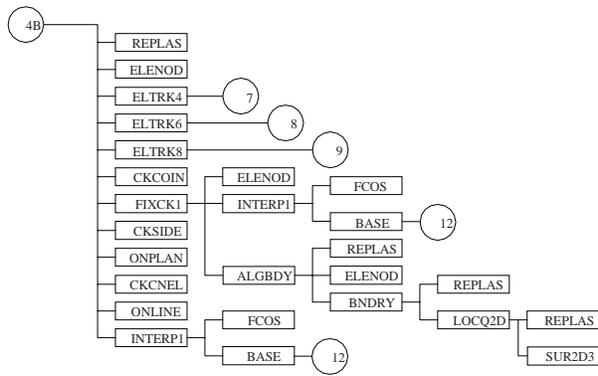**Figure B.9 Subroutine/function connection for Approach 1 (Part 5).**
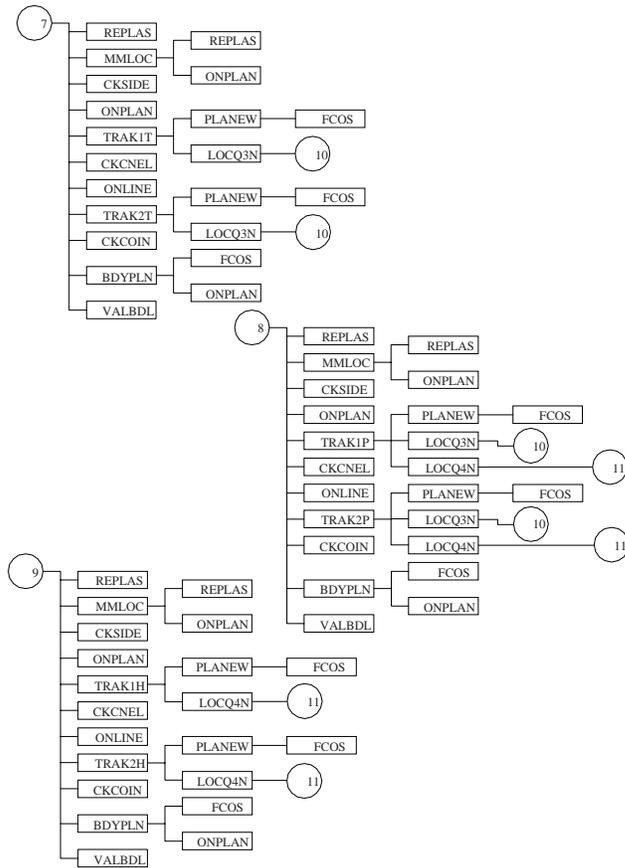

Figure B.10 Subroutine/function connection for Approaches 1 and 2 (Part 6).
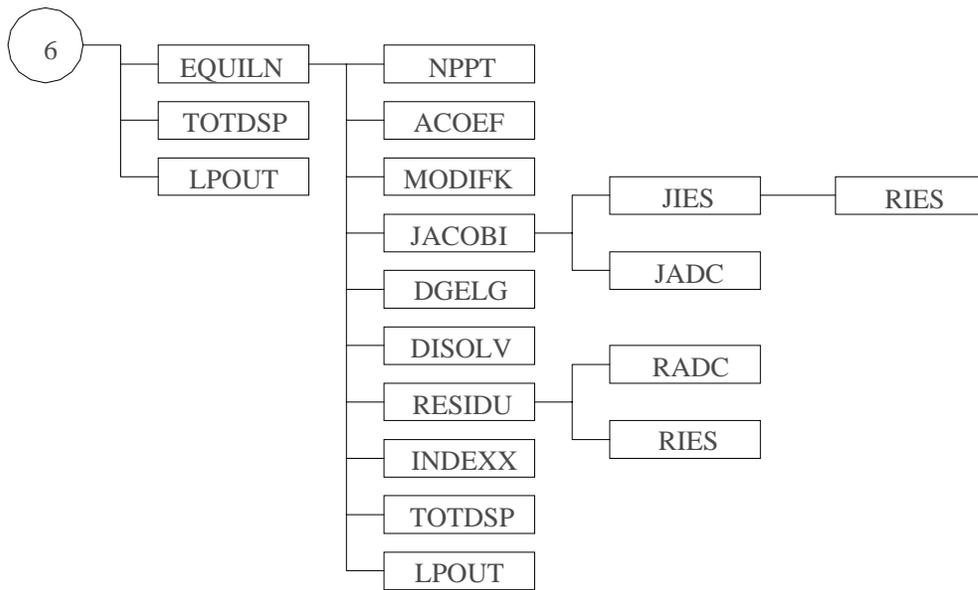
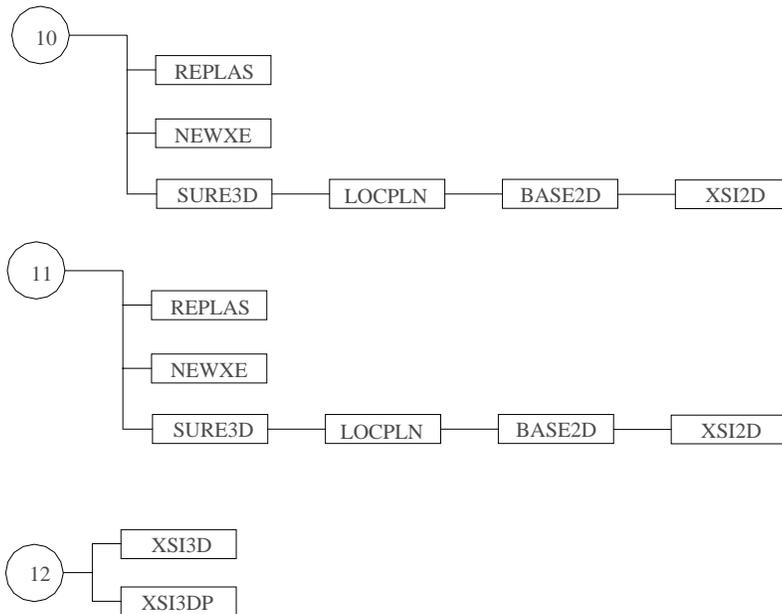**Figure B.11 Subroutine/function connection forApproaches 1 and 2 (Part 7).**

Figure B.12 Subroutine/function connection for Approaches 1 and 2 (Part 8).

In the following, all subroutines/functions in 3DHYDROGEOCHEM are described. There are totally 141 subroutines and 2 functions, including 14 subroutines specifically for Approach 1, 16 subroutines specifically for Approach 2, and 111 subroutines as well as 2 functions in common.

Subroutine **ACOEF**

It calls none.

It is called by EQULIN.

This subroutine is to calculate both the ionic strength and the activity coefficient of each species. One can use zero (ICOR = 0), constant (ICOR = 1), and variable (ICOR = 2) ionic strength to compute the activities of species. For the case of ICOR = 1, one needs to provide a constant ionic strength. For the case of ICOR = 2, 3DHYDROGEOCHEM will compute the ionic strength based on the concentrations of chemical species at the location of interest.

Subroutine **ADVBC**

It calls Q34ADB.

It is called by HGCT.

This subroutine is to implement Cauchy, Variable, and Dirichlet boundary conditions in the Lagrangian step for (1) Approach 2 of chemical transport and (2) heat transfer. For Dirichlet boundaries, the total analytical component concentration (or temperature) is specified. For Variable boundaries, the fictitious particle associated with a downstream boundary node must come from the inside of the domain of interest. The Lagrangian concentration (or temperature) for this boundary node is computed through backward particle tracking and interpolation. Thus, there is no need to implement a Variable boundary condition for a downstream boundary. On the other hand, the concentration (or temperature) of incoming fluid is specified for the upstream Variable boundary. The Lagrangian concentration (or temperature) is calculated according to

$$T^* = \int_{B_V^-} N_i \mathbf{V_{nf}} C_{in} dB \Big/ \int_{B_V^-} N_i \mathbf{V_{nr}} dB$$

where T* is the Lagrangian total analytical concentration of a mobile component (or temperature) due to the boundary source at the boundary node i, $\mathbf{V_n}$ is the normal Darcy flux, $\mathbf{V_{nr}}$ is the normal Darcy flux coupled with the retardation effect, and C_{in} is the total dissolved concentration of the mobile component (or temperature) in the incoming fluid. One should be aware that $\mathbf{V_n}$ and $\mathbf{V_{nf}}$ are identical in computing heat transfer. Cauchy boundary conditions are normally applied to the upstream boundary, on which the material (or heat) flux of incoming fluid is specified. The Lagrangian concentration (or temperature) is calculated according to

$$T^* = \int_{B_c} N_i q_c dB \Big/ \int_{B_c} N_i \mathbf{V_{nr}} dB$$

where T* is the Lagrangian total analytical concentration of a mobile component (or temperature) due to the Cauchy flux at the boundary node i, $\mathbf{V_{nr}}$ is the normal Darcy flux coupled with the retardation effect, and q_c is the Cauchy flux of the mobile component (or heat) in the incoming fluid. Since a Neumann boundary condition describes the physical phenomenon of diffusion, it is bypassed. Generally speaking, all the computations in this subroutine are achieved based on the material mass that should be conserved from an advection point of view.

Subroutine **ADVBC1**

It calls Q34ADB1.

It is called by HGCT1

This subroutine is to implement Cauchy, Variable, and Dirichlet boundary conditions in the Lagrangian step for (1) Approach 1 of chemical transport and (2) heat transfer. For Dirichlet boundaries, the total analytical component concentration is specified. The consideration for executing Variable and Cauchy boundary conditions is the same as those described in ADVBC, except that the Lagrangian concentration here is the total

dissolved component concentration, instead of the total analytical component concentration in ADVBC. The Lagrangian concentration for an upstream Variable boundary is calculated according to

$$C^* = \int_{B_V^-} N_i \mathbf{V_n} C_{in} dB / \int_{B_V^-} N_i \mathbf{V_n} dB$$

where C* is the Lagrangian total dissolved concentration of a mobile component (or temperature) due to the boundary source at the boundary node i, $\mathbf{V_n}$ is the normal Darcy flux, and C_{in} is the total dissolved concentration of the mobile component (or temperature) in the incoming fluid. The Lagrangian concentration for a Cauchy boundary is calculated according to

$$C^* = \int_{B_c} N_i q_c dB / \int_{B_c} N_i \mathbf{V_n} dB$$

where C* is the Lagrangian total dissolved concentration of a mobile component (or temperature) due to the Cauchy flux at the boundary node i, $\mathbf{V_n}$ is the normal Darcy flux, and q_c is the Cauchy flux of the mobile component (or heat) in the incoming fluid. The Neumann boundary condition, describing the physical phenomenon of diffusion, is bypassed.

Subroutine **ADVW3D**
It calls LRL3D.
It is called by FACT and FACT1.
      This subroutine is to generate all the working arrays, including IBW, IEW, NLRLW, and LRLW, for the "in-element" particle tracking in the Lagrangian step. The more sub-elements generated for particle tracking, the more accurate the results obtained. In this subroutine, the working arrays are generated for the following types of elements: (1) tetrahedral elements (if ISHAPE = 4 or 0), (2) triangular prism elements (if ISHAPE = 6 or 0), and (3) hexahedral elements (if ISHAPE = 8 or 0). For the case that the entire domain is discretized with only tetrahedral elements, ISHAPE is defined as 4. ISHAPE = 6 for the case of triangular elements only, while ISHAPE = 8 for hexahedral elements only and ISHAPE = 0 for the case of mixed elements.

Subroutine **AFABTA**
It calls LAMBDA.
It is called by HGCT and HGCT1.
      The subroutine is to calculate the upstream weighting factor along each edge of elements.

Subroutine **ALGBDY**
It calls REPLAS, ELENOD, and BNDRY.
It is called by FIXCK and FIXCK1.
      This subroutine is to control the flow of tracking particles along both the unspecified and the Neumann boundaries. The particle tracking is executed one boundary side by one boundary side based on the projected nodal velocity on the side being considered. The tracking will not be stopped until either the tracking time is totally consumed or the particle encounters a specified boundary side. For accuracy, using the average velocity of both the source point and the target point to locate the target point can be considered in the subroutine. However, if this average velocity approach is not able to deal with complex velocity fields, the single velocity of the source point is used to determine the location of the target point, which is guaranteed to have an answer even though it might be less accurate.

Subroutine **BASE**
It calls XSI3D and XSI3DP.
It is called by FIXCK, GNTRAC, HGCT, HGCT1, HPTRAC, HPTRAC1, INTERP, and INTERP1.
      This subroutine is to calculate all the base functions and their derivatives at a specific point inside the element being considered.

Subroutine **BASE2D**

It calls XSI2D.

It is called by LOCPLN.

> This subroutine is to compute two-dimensional base functions at a specific point inside a two-dimensional element (it is an element side here) being considered based on the given coordinates of the point and the element nodes.

Subroutine **BCPREP**

It calls Q34S.

It is called by HGCH.

> This subroutine is to prepare the infiltration-seepage boundary conditions during a rainfall period or the seepage-evapotranspiration boundary conditions during non-rainfall periods. It determines the number of nodal points on the Variable boundary to be considered as Dirichlet or Cauchy points. It computes the number of points that change boundary conditions from ponding depth (Dirichlet types) to infiltration (Cauchy types), or from infiltration to ponding depth, or from minimum pressure (Dirichlet types) to infiltration during rainfall periods. It also computes the number of points that change boundary conditions from potential evapotranspiration (Cauchy types) to minimum pressure, or from ponding depth to potential evapotranspiration, or from minimum pressure to potential evapotranspiration during non-rainfall periods. The decision for changing boundary conditions is made based on (1) comparing the rainfall rate with the Darcy flux during the rainfall period, (2) comparing the ponding depth with the calculated pressure head during the rainfall period, (3) comparing the evapotranspiration rate with the Darcy flux during the non-rainfall period, and (4) comparing the given minimal pressure head with the pressure head during the non-rainfall period at those Variable boundary nodes.

Subroutine **BDYPLN**

It calls FCOS and ONPLAN.

It is called by ELTRK4, ELTRK6, and ELTRK8.

> This subroutine is to locate the particle on a working element side in the "in-element" particle tracking when a particle encounters the boundary of a working element. In addition, it calls ONPLAN to adjust the coordinate so that the particle and the nodes of the working element side are guaranteed on the same plane, which is required for the successive particle tracking.

Subroutine **BLKITR**

It calls SOLVE.

It is called by HGCH, HGCT, and HGCT1.

> This subroutine is to solve the linearized matrix equations with the block iteration technique. First, the block matrix equation for each subregion is constructed based on the global matrix equation and two pointer arrays GNPLR and LNOJCN. Then the interboundary conditions are implemented. Finally, the block matrix equation is solved with a direct band matrix solver --- SOLVE, and iterations among blocks continue until either the convergence is reached or the given iteration number is completely consumed.

Subroutine **BNDRY**

It calls REPLAS and LOCQ2N.

It is called by ALGBDY.

> This subroutine is to locate the target point of a particle tracking along either an unspecified or a Neumann boundary side. All the computations are made according to either the average velocity approach (IJUDGE = 1) or the single velocity approach (IJUDGE = 2). When the velocity field is very complex, there might be no solution for the average velocity approach. In this case, IJUDGE is temporarily switched to 2. This control is executed in ALGBDY.

Subroutine **CBVFCT**

It calls none.

It is called by FACT and FACT1.

      This subroutine is to find the profile value of Cauchy boundary conditions at time t either by tabular interpolation or by analytical expression. For the latter case, the user must provide the function after "200 CONTINUE" in the subroutine.

Subroutine **CDATIO**

It calls none.

It is called by MAIN and MAIN1.

      This subroutine is to read and print all the information needed for computing chemical equilibrium.

Subroutine **CDERIV**

It calls FGDET.

It is called by EQDCTJ and TQ468.

      This subroutine is to compute the concentration derivatives C_j/T_i and C_j/EC_i for considering retardation effect in chemical transport. In the concentration derivatives, C_j is the total dissolved concentration of the j-th component (only the aqueous components are of interest); T_i is the total analytical concentration of the i-th component; EC_i is the ion-exchange capacity of the i-th ion-exchange site which can be either a cation one or an anion one. This subroutine is used only for Approach 2.

Subroutine **CKBDY**

It calls none.

It is called by FACT, FACT1, MAIN, and MAIN1.

      This subroutine is to check all the boundary sides and generated arrays, including NBDYB and IBDY, for use in the particle tracking on both the unspecified and the Neumann boundaries. NBDYB(I) represents the accumulated number of the unspecified and Neumann boundary sides connecting with the 1-st though the (I-1)-th global node. IBDY(I) indicates the global boundary side to which the I-th unspecified/Neumann boundary side relates.

Subroutine **CKCNEL**

It calls none.

It is called by ELTRK4, ELTRK6, ELTRK8, GNTRAC, GNTRAC1, HPTRAC, and HPTRAC1.

      This subroutine is to check the elements that would connect to a specific side plane.

Subroutine **CKCOIN**

It calls none.

It is called by ELTRK4, ELTRK6, ELTRK8, GNTRAC, GNTRAC1, HPTRAC, and HPTRAC1.

      This subroutine is to check if a specific point coincides with a global node.

Subroutine **CKSIDE**

It calls none.

It is called by ELTRK4, ELTRK6, ELTRK8, GNTRAC, GNTRAC1, HPTRAC, HPTRAC1, and REFINE.

      This subroutine is to check if a specific point is on a side line of a side plane.

Subroutine **DBVFCT**

It calls none.

It is called by FACT and FACT1.

>This subroutine is to find the profile value of Dirichlet boundary conditions at time t either by tabular interpolation or by analytical expression. For the latter case, the user must provide the function after "200 CONTINUE" in the subroutine.

Subroutine **DGELG**

It calls none.

It is called by EQUILN.

>This subroutine is to solve the Jacobian matrix equation by using the technique of Gaussian elimination with full pivoting.

Subroutine **DISOLV**

It calls none.

It is called by EQUILN.

>This subroutine is to dissolve assumed precipitated species that have been found to have negative concentrations for two successive iterations.

Subroutine **DISPC**

It calls none.

It is called by FACT, FACT1, HGCT, and HGCT1.

>This subroutine is to compute dispersion coefficients at either Gaussian points (if IQUAR = 1 or 3) or nodal points (if IQUAR = 2 or 4) for all the elements.

Function **DOTPRD**

It calls none.

It is called by PPCG.

>This function is to calculate the inner product of two vectors.

Subroutine **ELENOD**

It calls none.

It is called by ALGBDY, FIXCK, FIXCK1, FSFLOW, GNTRAC, GNTRAC1, HGCT, HGCT1, HPTRAC, and HPTRAC1.

>This subroutine is to determine the number of nodes, the number of faces, and the elemental shape index of element M by using the IE(M,5) and IE(M,7) information.

Subroutine **ELTRK4**

It calls BDYPLN, CKCNEL, CKCOIN, CKSIDE, MMLOC, ONLINE, ONPLAN, REPLAS, TRAK1T, TRAK2T, and VALBDL.

It is called by GNTRAC, GNTRAC1, HPTRAC, HPTRAC1.

>This subroutine is to compute the particle tracking in a tetrahedral element (Figure B.10). In the subroutine, the subelement in which the starting point is located is first determined. Starting from that subelement, the particle is tracked one subelement by one subelement until either the tracking time is completely consumed or the particle encounters a boundary side of the element being considered. During the particle tracking, this subroutine calls (1) TRAK1T to track a particle in the subelement being considered if that particle coincides with a node of the subelement, and (2) TRAK2T to track a particle if that particle is not on any nodes of the subelement. In the particle tracking process, the average velocity approach is used if IJUDGE=1 and the single velocity approach is used if IJUDGE=2.

Subroutine **ELTRK6**

It calls BDYPLN, CKCNEL, CKCOIN, CKSIDE, MMLOC, ONLINE, ONPLAN, REPLAS, TRAK1P, TRAK2P, and
VALBDL.

It is called by GNTRAC, GNTRAC1, HPTRAC, HPTRAC1.

This subroutine is to compute the particle tracking in a triangular prism element (Figure B.10). The computation
here is the same as that in ELTRK4.

Subroutine **ELTRK8**

It calls BDYPLN, CKCNEL, CKCOIN, CKSIDE, MMLOC, ONLINE, ONPLAN, REPLAS, TRAK1H, TRAK2H, and
VALBDL.

It is called by GNTRAC, GNTRAC1, HPTRAC, HPTRAC1.

This subroutine is to compute the particle tracking in a hexahedral element (Figure B.10). The computation here
is the same as that in ELTRK4.

Subroutine **EQDCTJ**

It calls CDERIV, EQMOD, and TEMPEQ.

It is called by FACT and HGCT.

This subroutine is to control the flow of computing the chemical equilibrium and/or the concentration derivatives
C_j / T_i and C_j / EC_i at all the global nodes. This subroutine is used for Approach 2 only.

Subroutine **EQDCTJ1**

It calls EQMOD and TEMPEQ.

It is called by FACT1 and HGCT1.

This subroutine is to control the flow of computing the chemical equilibrium. This subroutine is employed for
Approach 1 only.

Subroutine **EQNGEN**

It calls none.

It is called by HGCT1.

This subroutine is to compose the coefficient matrix and the load vector of the matrix equations in Approach 1
of chemical transport. Two options of composing the matrix equations can be chosen: ICOMP = 0 or 1. For the
case of ICOMP = 0, the matrix equations are constructed under an assumption that all types of concentrations at
Gaussian points can be interpolated by those at global nodes. For the case of ICOMP = 1, on the other hand, only
the total analytical concentration at Gaussian points can be interpolated whereas the other types of concentrations
(e.g., the total dissolved, sorbed, and precipitated concentrations) need to be computed through chemical
equilibrium. The computation of mass matrices, stiffness matrices, and load vectors associated with this are
implemented in TASEMB1 and TQ4681. Unlike Approach 1, only the case of ICOMP = 1 is considered in
Approach 2 because the concentration derivative needs to be accurately estimated at Gaussian points.

Subroutine **EQMOD**

It calls EQUILN, LPOUT, and TOTDSP.

It is called by EQDCTJ, EQDCTJ1, HGCT1, TQ468, and TQ4681.

This subroutine is to control the flow of solving for chemical equilibrium under a specific condition (Figure
B.11). Basically, it first makes an initial guess based on the given specific condition. Secondly, it calls EQUILN
to solve a set of algebraic equations governing both mole balance and chemical equilibrium. Thirdly, it calls
TOTDSP to compute the total dissolved, the total sorbed, and the total precipitated concentrations after the
concentrations of all the species have been calculated. Finally, it calls LPOUT and STORE to print and to store
the results, respectively, if desired.

Subroutine **EQULIN**

It calls ACOEF, DGELG, DISOLV, INDEXX, JACOBI, LPOUT, MODIFK, NPPT, RESIDU, and TOTDSP.

It is called by EQMOD.

> This subroutine is to solve the system of nonlinear algebraic equations governing both mole balance and chemical equilibrium. The system of nonlinear algebraic equations is solved with the Newton-Ralphson iteration technique. For each iteration, it calls (1) ACOEF to compute the activity coefficients for all species, (2) MODIFK to calculate the modified equilibrium constants for all product species, (3) RESIDU to evaluate the residuals of all the governing equations, (4) JACOBI to compute the Jacobian of all the governing equations, and (5) DGELG to solve the Jacobian matrix equation with full pivoting. New iterations are obtained by adding those differences to the old iterations. In checking convergence, only the relative error is considered. However, after running out of iterations, we check both the absolute error and mole balance to see if a convergent solution has been reached. If the absolute error of each species is less than the specified criteria and mole balance for each component is maintained, the solution is considered convergent.

Subroutine **ESSFCT**

It calls none.

It is called by FACT and FACT1.

> This subroutine is to find the profile value of element (distributed) sources/sinks at time t either by tabular interpolation or by analytical expression. For the latter case, the user must provide the function after "200 CONTINUE" in the subroutine.

Subroutine **FACT**

It calls ADVW3D, CBVFCT, CKBDY, DBVFCT, DISPC, EQDCTJ, ESSFCT, FLUX, FPRINT, FSFLOW, FSTORE, HGCH, HGCT, NBVFCT, NODVAL, READR, SPROP, TPRINT, TPRITT, TSFLOW, TSTORE, TSTORT, VBVFCT, VELT, and WSSFCT.

It is called by MAIN.

> This subroutine is to control the flow of simulations (Figure B.3). It includes three basic sections: an initial/pre-initial block, a steady-state simulation block, and a transient-state simulation block. In the initial/pre-initial block, all the variables required at the starting point of the simulation are calculated. State variables, such as pressure head, temperature, and total analytical concentration, can be printed and/or stored if parameters KPR0 and/or KDSK0 is not 0. In the steady-state simulation block, the iteration among flow, transfer, and transport is designed for computing either the final steady-state solution or the initial steady-state condition used for a further transient-state simulation. State variables can also be printed and/or stored if KPR0 and/or KDSK0 is not 0. In the transient-state simulation block, the results of the previous time step are used to compute the flow at the current time step. The computed flow field is then used for computing the heat transfer. Both the computed flow and temperature are taken into account in computing the chemical transport at the current time step. State variables can also be printed out and/or stored if KPR(ITM) and/or KDSK(ITM) is not 0. In both steady-state and transient-state simulations, this subroutine calls HGCH and HGCT for computing flow and transfer/transport, respectively. This subroutine is used for Approach 2 only.

Subroutine **FACT1**

It calls ADVW3D, CBVFCT, CKBDY, DBVFCT, DISPC, EQDCTJ1, ESSFCT, FLUX, FPRINT, FSFLOW, FSTORE, HGCH, HGCT1, NBVFCT, NODVAL, READR, SPROP, TPRINT, TPRITT, TSFLOW, TSTORE, TSTORT, VBVFCT, VELT, and WSSFCT.

It is called by MAIN1.

> This subroutine is to control the flow of simulation when Approach 1 of chemical transport is considered (Figure B.4). Its structure is basically the same as that of FACT, except that it calls HGCT1 to compute heat transfer and chemical transport.

Subroutine **FASEMB**

It calls FQ468, RHOAMU, and VOLUME.

It is called by HGCH.

>This subroutine is to compose the global matrix equation for computing the flow of groundwater.

Subroutine **FBCDAT**

It calls READN and READR.

It is called by MAIN and MAIN1.

>This subroutine is to read and print all the information needed for implementing the boundary conditions, including Dirichlet, Neumann, Cauchy, and Variable boundary conditions, of subsurface flow.

Subroutine **FBCN**

It calls Q34S and RHOAMU.

It is called by HGCH.

>This subroutine is to incorporate Dirichlet, Neumann, Cauchy, and Variable boundary conditions for subsurface flow. For the Dirichlet boundaries, an identity algebraic equation is generated for each Dirichlet nodes. Any other equations having this nodal variable are modified accordingly to simplify the computation. For the Neumann boundaries, the integrations of both the gradient and gravity fluxes are obtained by calling Q34S. For the Cauchy boundaries, the integration of the surface source is obtained by calling Q34S, and the result is added to the load vector. In implementing Variable boundary conditions, it first checks all the Variable boundary points to identify any of those points that are Dirichlet points. If there are Dirichlet points, the method of incorporating the Dirichlet boundary conditions mentioned above is used. If a given point is not a Dirichlet point, the point is bypassed. Secondly, it checks all the Variable boundary points again to see if any of those points are Cauchy points. If a point is a Cauchy point, then the method incorporating the Cauchy boundary conditions mentioned above is used. If a point is not a Cauchy point, it is bypassed. All the Variable boundary points are taken care of in this manner because a Variable boundary point is either a Dirichlet or a Cauchy point.

Function **FCOS**

It calls none.

It is called by BDYPLN, INTERP, INTERP1, PLANEW, and REFINE.

>This function is to compute the inner product of the normal vector of a given plane with a specified vector whose starting point stands on the plane. The result helps to determine where the end point of the specified vector is located.

Subroutine **FIXCK**

It calls ALGBDY, BASE, ELENOD, and INTERP.

It is called by GNTRAC and HPTRAC.

>This subroutine is to complete the particle tracking after a particle encounters the domain boundary. The backward-tracked concentration (or temperature) is obtained by interpolation if the boundary is specified including Dirichlet, Cauchy, and Variable types. Otherwise, the particle tracking continues along the unspecified and Neumann boundaries until either the specified boundaries is encountered or tracking time is consumed. This is for Approach 2.

Subroutine **FIXCK1**

It calls ALGBDY, ELENOD, and INTERP1.

It is called by GNTRAC1 and HPTRAC1.

>This subroutine is to complete the particle tracking associated with Approach 1 after a particle encounters the domain boundary. Its structure is the same as that of FIXCK.

Subroutine **FLUX**

It calls TQ468DV.

It is called by FACT and FACT1.

> This subroutine is to compute the material fluxes at all the global nodes. In this subroutine, the global matrix equation is composed based on Fick's law and is solved with the mass lumping technique for dispersive fluxes. The total fluxes are then computed by adding the advective fluxes to the dispersive fluxes.

Subroutine **FPDET**

It calls none.

It is called by CDERIV.

> This subroutine is to calculate the determinant of a given matrix.

Subroutine **FPRINT**

It calls none.

It is called by FACT and FACT1.

> This subroutine is to print the simulation results of subsurface flow at each time step. The print-out includes (1) the diagnostic flow information (if KPR.GT.0), (2) the pressure heads (if KPR.GT.1), (3) the total heads (if KPR.GT.2), (4) the moisture contents (if KPR.GT.3), and (5) the Darcy fluxes (if KPR.GT.4).

Subroutine **FQ468**

It calls SHAPE.

It is called by FASEMB.

> This subroutine is to compose the element matrix equation for computing the flow of groundwater. The element matrices include

$$QA(i,j) = \int\limits_{R_e} N_i^e \frac{\rho}{\rho_o} \frac{\partial \theta}{\partial h} N_j^e dR$$

$$QB(i,j) = \int\limits_{R_e} \nabla N_i^e \cdot \mathbf{K} \cdot \nabla N_j^e dR$$

where ρ is the real groundwater density, and ρ_o is the referenced groundwater density, and $\mathbf{K}$ is the hydraulic conductivity tensor. The element load vector is defined as

$$RQ(i) = \int\limits_{R_e} \left[\nabla N_i \cdot \mathbf{K} \cdot \nabla z \frac{\rho}{\rho_o} + \frac{\rho^*}{\rho_o} q \right] dR \qquad \text{if q is a source}$$

$$RQ(i) = \int\limits_{R_e} \left[\nabla N_i \cdot \mathbf{K} \cdot \nabla z \frac{\rho}{\rho_o} - \frac{\rho}{\rho_o} q \right] dR \qquad \text{if q is a sink}$$

The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

Subroutine **FQ468DV**

It calls SHAPE.

It is called by VELT.

> This subroutine is to compute the Darcy fluxes at all the global nodes. The element matrices given by

$$QR(i,j) = \int\limits_{B_e} N_i^e N_j^e dR$$

The element load vector are:

$$QRX(i) = -\int_{R_e} N_i^e\, \mathbf{i}\cdot\mathbf{K}\cdot\frac{\rho_o}{\rho}\left(\nabla N_j^e\right)h_j\, dR - \int_{R_e} N_i^e\, \mathbf{i}\cdot\mathbf{K}\cdot\nabla z\, dR$$

$$QRY(i) = -\int_{R_e} N_i^e\, \mathbf{j}\cdot\mathbf{K}\cdot\frac{\rho_o}{\rho}\left(\nabla N_j^e\right)h_j\, dR - \int_{R_e} N_i^e\, \mathbf{j}\cdot\mathbf{K}\cdot\nabla z\, dR$$

$$QRZ(i) = -\int_{R_e} N_i^e\, \mathbf{k}\cdot\mathbf{K}\cdot\frac{\rho_o}{\rho}\left(\nabla N_j^e\right)h_j\, dR - \int_{R_e} N_i^e\, \mathbf{k}\cdot\mathbf{K}\cdot\nabla z\, dR$$

where h_j is the referenced pressure head at nodal point j, $\mathbf{K}$ is the hydraulic conductivity tensor, and $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the unit vector along the x-, y-, and z- directions. The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

Subroutine **FSFLOW**
It calls ELENOD, Q34S, Q468TH, and VOLUME.
It is called by FACT and FACT1.

This subroutine is to compute the fluxes through various types of boundaries and the increasing rate of water content in the region of interest. The function of FRATE(7) is to store the flux through the whole boundary enclosing the region of interest. It is given by

$$FRATE(7) = \int_B \left(V_x n_x + V_y n_y + V_z n_z\right) dB$$

where B is the global boundary of the region of interest; V_x, V_y, and V_z are Darcy flux components; and n_x, n_y, and n_z are the directional cosines of the outward unit vector normal to the boundary B. FRATE(1) through FRATE(5) store the flux through Dirichlet boundary B_D, Cauchy boundary B_C, Neumann boundary B_N, the seepage-evapotranspiration boundary B_S, and infiltration boundary B_R, respectively, and are given by

$$FRATE(1) = \int_{B_D} \left(V_x n_x + V_y n_y + V_z n_z\right) dB$$

$$FRATE(2) = \int_{B_C} \left(V_x n_x + V_y n_y + V_z n_z\right) dB$$

$$FRATE(3) = \int_{B_N} \left(V_x n_x + V_y n_y + V_z n_z\right) dB$$

$$FRATE(4) = \int_{B_S} \left(V_x n_x + V_y n_y + V_z n_z\right) dB$$

$$FRATE(5) = \int_{B_R} \left(V_x n_x + V_y n_y + V_z n_z\right) dB$$

FRATE(6), which is related to the numerical loss, is given by

$$FRATE(6) = FRATE(7) - \sum_{l=1}^{5} FRATE(l)$$

FRATE(8) and FRATE(9) are used to store the source/sink and increased rate of water within the media, respectively:

$$FRATE(8) = -\int_R \frac{\rho^*}{\rho} q\, dR$$

and

$$FRATE(9) = \int_R \frac{\rho}{\rho_o}\frac{d\theta}{dh}\frac{\partial h}{\partial t} dR$$

If there is no numerical error in the computation, the following equation should be satisfied:

$$FRATE(9) = -[FRATE(7) + FRATE(8)]$$

and FRATE(6) should be equal to zero. Equation (2.28) simply states that the negative rate of water going out from the region through the entire boundary and due to a source/sink is equal to the rate of water accumulated in the region.

Subroutine **FSSDAT**
It calls READN.
It is called by MAIN and MAIN1.

This subroutine is to read and print all the information for implementing the sources/sinks, including element (distributed) and well (point) sources/sinks, of subsurface flow.

Subroutine **FSTORE**
It calls none.
It is called by FACT and FACT1.

This subroutine is to store the flow variables on Logical Unit 11. It is intended for use of plotting. The information stored includes region geometry, subregion data, and hydrological variables, such as pressure head, total head, water content, and Darcy flux components.

Subroutine **GNTRAC**
It calls BASE, CKCNEL, CKCOIN, CKSIDE, ELENOD, ELTRK4, ELTRK6, ELTRK8, FIXCK, INTERP, ONLINE, ONPLAN, and REPLAS.
It is called by HGCT.

This subroutine is to control the flow of particle tracking starting from global nodes (Figure B.8). In the subroutine, each particle is tracked one element by one element until either the tracking time is completely consumed or the particle encounters a specified boundary side. During the particle tracking, this subroutine calls (1) ELTRK4 to track a particle in a tetrahedral element, (2) ELTRK6 to track a particle in a triangular prism element, and/or (3) ELTRK8 to track a particle in a hexahedral element. In order to make the particle tracking complete and remedy the given velocity field error on the unspecified and Neumann boundaries, this subroutine calls ALGBDY to continue tracking particles along the unspecified and Neumann boundaries. When the backward particle tracking is considered, all the total analytical concentrations of mobile components are obtained by interpolation at the end of tracking. One should be aware that each mobile component has its own velocity field, which is determined by taking the retardation effect into account, for particle tracking. The retardation effect is considered based on the concentration derivatives, C_j/T_i and C_j/EC_i, which are computed in CDERIV. This retardation effect is considered only in Approach 2. On the other hand, the particle tracking velocity for heat transfer does not include retardation effect, so does that associated with conservative mobile components in chemical transport.

Subroutine **GNTRAC1**
It calls CKCNEL, CKCOIN, CKSIDE, ELENOD, ELTRK4, ELTRK6, ELTRK8, FIXCK1, INTERP1, ONLINE, ONPLAN, and REPLAS.
It is called by HGCT1.

This subroutine is to control the flow of particle tracking starting from global nodes for Approach 1 (Figure B.9). The structure of this subroutine is the same as that of GNTRAC, except that the pore velocity field is used to achieve particle tracking in this subroutine for all mobile components in chemical transport as well as heat transfer.

Subroutine **HGCH**
It calls BCPREP, BLKITR, FASEMB, FBCN, ILUCG, PISS, PPCG, SPROP, and VELT.

It is called by FACT and FACT1.

This subroutine is to control the flow of computing the subsurface flow in the domain of interest for both the steady-state and transient-state simulations (Figure B.5). It calls SPROP to obtain the hydraulic conductivities, the water capacities, and the water contents of all the elements from the given pressure head at global nodes; VELT to compute the Darcy fluxes at all the global nodes; BCPREP to determine if a change of boundary conditions is required if any boundary conditions exist for the system being simulated; FASEMB to assemble the element matrices over all elements to make the global matrix equation; FBCN to implement the boundary conditions; BLKITR, PISS, PPCG, or ILUCG to solve the linearized matrix equations; FSFLOW to calculate flux through all types of boundaries and water accumulated in the media; FPRINT to print out the results; and FSTORE to store the flow variables in binary format for plotting.

Subroutine **HGCT**

It calls ADVBC, AFABTA, BASE, BLKITR, DISPC, ELENOD, EQDCTJ, GNTRAC, HLCAP, HPTRAC, ILUCG, LGFP, PISS, PPCG, TACADC, TASEMB, and TBC.

It is called by FACT.

This subroutine is to control the flow of computing chemical transport and/or heat transfer for both the steady-state and transient-state simulations. In the steady-state simulation, it solves transport equations for conservative mobile components first, and solves for non-conservative mobile components afterward. There is no need to solve for ion-exchange capacities, immobile components, and specified concentration/activity components in the steady-state simulation. In the transient-state simulation, it solves for both ion-exchange capacities and immobile components, by considering only the first order decay to be the only mechanism to change the amounts of those quantities, on the first step. And then, like that in the steady-state simulation it solves for conservative mobile components before it solves for non-conservative mobile components on the second step. Again, there is no need to solve for the specified concentration/activity components. In both the steady-state and the transient-state simulations, iteration loops are applied to solving for non-conservative mobile components. This strategy is used to deal with the nonlinear behavior associated with those components. The reason why non-conservative mobile components are always solved for at last is that the better initial guesses are made if the concentrations of the other types of components are known at the time being. In other words, the initial guesses would be closer to the final solution if all the component concentrations but the non-conservative ones are already known. In the transient simulation, the Lagrangian step for non-conservative mobile components is located in the nonlinear iteration loop because the retardation effect is changing with iteration until the final solution is reached. To increase accuracy, a peak/valley capturing scheme is implemented in this subroutine for mobile components. This subroutine is used in Approach 2 only (Figure B.6). Temperature is treated the same as the concentration of a conservative mobile component. It is like a non-reactive solute transport.

Subroutine **HGCT1**

It calls ADVBC1, AFABTA, BASE, BLKITR, DISPC, ELENOD, EQDCTJ1, EQMOD, EQNGEN, GNTRAC1, HLCAP, HPTRAC1, ILUCG, PISS, PPCG, TACADC, TASEMB1, TBC1, and TEMPEQ.

It is called by FACT1.

This subroutine is to control the flow of computing chemical transport and/or heat transfer for both the steady-state and transient-state simulations. The basic structure of this subroutine is similar to that of HGCT. However, the Lagrangian step is located outside the nonlinear iteration loop due to the use of the component-independent velocity in both chemical transport and heat transfer. This subroutine is used in Approach 1 only.

Subroutine **HLCAP**

It calls REFINE.

It is called by HGCT and HGCT1.

This subroutine is to capture the maximum peak and the minimum valley of the total analytical concentration for each mobile component.

Subroutine **HPTRAC**

It calls BASE, CKCNEL, CKCOIN, CKSIDE, ELENOD, ELTRK4, ELTRK6, ELTRK8, FIXCK, INTERP, ONLINE, ONPLAN, and REPLAS.

It is called by HGCT.

> This subroutine is to compute the locations and concentrations of all forward-tracked find-grid nodes. Basically, the algorithm of this subroutine is the same as that of GNTRAC. This is for Approach 2 (Figure B.8).

Subroutine **HPTRAC1**

It calls BASE, CKCNEL, CKCOIN, CKSIDE, ELENOD, ELTRK4, ELTRK6, ELTRK8, FIXCK1 INTERP1, ONLINE, ONPLAN, and REPLAS.

It is called by HGCT1.

> This subroutine is to compute the locations and concentrations of all forward-tracked find-grid nodes. Basically, the algorithm of this subroutine is the same as that of GNTRAC1. This is for Approach 1 (Figure B.9).

Subroutine **ILUCG**

It calls LLTINV.

It is called by HGCH, HGCT, and HGCT1.

> This subroutine is to solve the linearized matrix equation, which is sparse asymmetric, with the preconditioned conjugate gradient method using the incomplete Cholesky decomposition as a preconditioner. It calls LLTINV to invert the preconditioner.

Subroutine **INDEXX**

It calls none.

It is called by EQUILN.

> This subroutine is to index the saturation values among all the potential species that are subject to precipitation/dissolution reactions.

Subroutine **INTERP**

It calls BASE and FCOS.

It is called by FIXCK, GNTRAC, and HPTRAC.

> This subroutine is to compute (1) the total analytical component concentration in Approach 2 and (2) temperature by interpolation. In chemical transport, the total analytical concentrations of all the conservative mobile components are interpolated when $N0 = NK$ because they use the same particle tracking velocity field. But when $NK > N0$, only the total analytical concentration of the NK-th component (a non-conservative mobile component) is interpolated because each non-conservative mobile component has its own particle tracking velocity field. The interpolation of temperature is straightforward.

Subroutine **INTERP1**

It calls BASE and FCOS.

It is called by FIXCK1, GNTRAC1, and HPTRAC1.

> This subroutine is to compute (1) the total dissolved component concentration in Approach 1 and (2) temperature by interpolation. The total dissolved concentrations of all the mobile components can be easily interpolated because they use the same particle tracking velocity field. It is trivial to achieve interpolation for temperature.

Subroutine **JACOBI**

It calls JADC and JIES.

It is called by EQUILN.

This subroutine is to evaluate the Jacobians of the nonlinear algebraic equations governing mole balance and chemical equilibrium. In the subroutine, the Jacobians of (1) mole balance equations for aqueous components and (2) precipitation/dissolution reaction equations are directly evaluated. But the Jacobians of adsorption/desorption and ion-exchange reaction equations are calculated by calling JADC and JIES, respectively. Mathematically, the Jacobians can be written as RE_i / CW_j, where RE_i is the i-th residual equation and CW_j is the working concentration of the j-th specific species which is considered to be independent in the system considered.

Subroutine **JADC**
It calls none.
It is called by JACOBI.
> This subroutine is to compute the Jacobians of adsorption/desorption equations with the simple surface complexation model, the constant capacity adsorption model, or the triple-layer adsorption model being considered.

Subroutine **JIES**
It calls RIES.
It is called by JACOBI.
> This subroutine is to compute the Jacobians of the ion-exchange reaction equations. The subroutine is designed to deal with both cation-exchange and anion-exchange reactions. In addition, it is capable of dealing with more than one ion-exchange site.

Subroutine **LAMBDA**
It calls none.
It is called by AFABTA, TASEMB, and TASEMB1.
> This subroutine is to compute moisture content-dependent groundwater density and viscosity based on given empirical formulae.

Subroutine **LGFP**
It calls none.
It is called by HGCT.
> This subroutine is to prepare Lagrangian fine-grid nodes for the backward particle tracking to make up the incomplete total analytical component concentrations at forward-tracked nodes. This subroutine is needed only for Approach 2.

Subroutine **LLTINV**
It calls none.
It is called by ILUCG.
> This subroutine is to solve for a modified residual that will be used in the preconditioned conjugate gradient algorithm.

Subroutine **LOCQ2N**
It calls REPLAS and SUR2D3.
It is called by BNDRY.
> This subroutine is to locate the target point of a particle tracking on a line segment in a specified element. All the computations are made according to the average velocity approach and the single velocity approach, as the index parameter IJUDGE is 1 and 2, respectively. When the average velocity approach is considered, the Newton-Raphson method is used to solve a nonlinear algebraic equation such that the local coordinates of the target point on the pre-determined element segment can be determined. With these local coordinates, the location of the target point can be easily determined based on both the velocity of the source point and the geometrical relationship

between the source point and the pre-determined element side. It calls SUR2D3 to guarantee the solution when the single velocity approach is used.

Subroutine **LOCQ3N**
It calls NEWXE, REPLAS, and SURE3D.
It is called by TRAK1P, TRAK1T, TRAK2P, and TRAK2T.

This subroutine is to locate the target point of a particle tracking on a triangular side of a specified element, which is either a tetrahedral or a triangular prism element. Either the average velocity approach or the single velocity approach can be used. When the average velocity approach is considered, the Newton-Raphson method is used to solve a set of four simultaneous nonlinear algebraic equations such that the natural coordinates of the target point on the pre-determined element side (a triangular side) can be determined. With these natural coordinates, the location of the target point can be easily determined based on both the velocity of the source point and the geometrical relationship between the source point and the pre-determined element side. It calls SURE3D to guarantee the solution when the single velocity approach is used.

Subroutine **LOCQ4N**
It calls NEWXE, REPLAS, and SURE3D.
It is called by TRAK1H, TRAK1P, TRAK2H, and TRAK2P.

This subroutine is to locate the target point of a particle tracking on a quadrilateral side of a specified element, which is either a hexahedral or a triangular prism element. The approach of locating the target point here is the same as that in LOCQ3N, except that the local coordinate system is employed in this subroutine, whereas the natural coordinate system is used in LOCQ3N. This is because the local coordinate system is more appropriate

Subroutine **LOCPLN**
It calls BASE2D.
It is called by SURE3D.

This subroutine is to compute the base function values associated with a specified point on a pre-determined plane. First, it transforms all the point coordinates, including that of the point and those of the plane nodes, from a three-dimensional space to a two-dimensional space. Secondly, it calls BASE2D to compute the base function values based on those transformed two-dimensional coordinates.

Subroutine **LPOUT**
It calls none.
It is called by EQMOD and EQUILN.

This subroutine is to print chemical species distribution at desired nodes at desired times. The information printed includes (1) the total analytical, the total dissolved, the total sorbed, and the total precipitated concentration of all the components (if KPR.GT.0), (2)the species concentrations of all the components species (if KPR.GT.1), and the species concentration, the modified equilibrium constants, and the stoichiometric coefficients of all the product species (if KPR.GT.2).

Subroutine **LRL3D**
It calls none.
It is called by ADVW3D and PAGEN.

This subroutine is to generate the node-element connection pointer arrays NLRL and LRL used in particle tracking.

Subroutine **LRN3D**
It calls none.
It is called by PAGEN.

This subroutine is to generate the node-node connection pointer arrays NRLN and LRN used in composing the linearized matrix equations.

Subroutine **MAIN**

It calls CDATIO, CKBDY, FACT, FBCDAT, FSSDAT, PAGEN, RDATIO, READR, SURF, TBCDAT, and TSSDAT.
It is called by none.

This subroutine is to declare all the array dimensions, common blocks, and arrays (Figure B.1). In addition, it read the information needed for simulations. The information includes (1) problem identification and description, (2) option parameters, (3) iteration parameters, (4) time and print-out control parameters, (5) region's geometric and material data, (6) chemical equilibrium data, (7) initial conditions for flow, transfer, and transport, (8) source/sink conditions for flow, transfer, and transport, and (9) boundary conditions for flow, transfer, and transport. All the information read in this subroutine will be passed to FACT. This subroutine is for Approach 2.

Subroutine **MAIN1**

It calls CDATIO, CKBDY, FACT1, FBCDAT, FSSDAT, PAGEN, RDATIO, READR, SURF, TBCDAT, and TSSDAT.
It is called by none.

This subroutine is to declare all the array dimensions, common blocks, and arrays. In addition, it read the information needed for simulations (Figure B.2). The needed information is the same as that in FACT. This subroutine is for Approach 1.

Subroutine **MMLOC**

It calls ONPLAN and REPLAS.
It is called by ELTRK4, ELTRK6, and ELTRK8.

This subroutine is to determine the subelement where the "in-element" particle tracking begins within an element.

Subroutine **MODIFK**

It calls none.
It is called by EQUILN.

This subroutine is to calculate the modified equilibrium constants for all the product species based on both the zero-activity equilibrium constants and the activity coefficients of all the species.

Subroutine **NBVFCT**

It calls none.
It is called by FACT and FACT1.

This subroutine is to find the profile value of Neumann boundary conditions at time t either by tabular interpolation or by analytical expression. For the latter case, the user must provide the function after "200 CONTINUE" in the subroutine.

Subroutine **NEWXE**

It calls none.
It is called by LOCQ3N and LOCQ4N.

This subroutine is to prepare a new guess of local/natural coordinates used in the iteration loop of the Newton-Raphson scheme.

Subroutine **NODVAL**

It calls VOLUME.
It is called by FACT and FACT1.

This subroutine is to compute (1) the ion-exchange capacities, (2) the adsorption capacities, (3) the surface area for adsorption sites, (3) the bulk densities, (5) the water contents, (6) the changing rate of water contents, (7) the saturated water content, (8) groundwater density, and (9) C_{mb}/C_l at all the global nodes. In the above, (7) through (9) are computed only when heat transfer is considered. In (9), C_m is the specific heat capacity of dry medium, $_b$ is the bulk density, C_l is the specific heat of groundwater, and is the density of groundwater.

Subroutine **NPPT**
It calls none.
It is called by EQUILN.

This subroutine is to determine the number of species allowed to precipitate without violating the phase rule.

Subroutine **ONLINE**
It calls none.
It is called by ELTRK4, ELTRK6, ELTRK8, GNTRAC, GNTRAC1, HPTRAC, HPTRAC1, and REFINE.

This subroutine is to adjust the particle coordinates so that the particle is guaranteed to be on the line determined by two given nodes.

Subroutine **ONPLAN**
It calls none.
It is called by BDYPLN, ELTRK4, ELTRK6, ELTRK8, GNTRAC, GNTRAC1, HPTRAC, HPTRAC1, MMLOC, and REFINE.

This subroutine adjusts the particle coordinates so that the particle is guaranteed to be on the given element side.

Subroutine **PAGEN**
It calls LRL3D and LRN3D.
It is called by MAIN and MAIN1.

This subroutine is to generate nine pointer arrays. It calls LRL3D to generate node-element connection arrays NLRL and LRL. It calls LRN3D to generate node-node connection arrays NLRN and LRN. It also generates arrays, including NTNPLR, NNPLR, GNLR, LNOJCN, and LMAXDF, used for the block iteration solver.

Subroutine **PISS**
It calls none.
It is called by HGCH, HGCT, and HGCT1.

This subroutine is to solve the linearized matrix equation with pointwise iteration solution strategies.

Subroutine **PLANEW**
It calls FCOS.
It is called by TRAK1H, TRAK1P, TRAK1T, TRAK2H, TRAK2P, and TRAK2T.

This subroutine is to determine which one of the two sides, separated by a specified plane, the particle would move into. All the computations are made according to the average velocity approach and the single velocity approach, when the index parameter IJUDGE is 1 and 2, respectively.

Subroutine **POLYP**
It calls none.
It is called by PPCG.

This subroutine is to solve a modified residual that will be used in the preconditioned conjugate gradient algorithm.

Subroutine **PPCG**

It calls POLYP.

It is called by HGCH, HGCT, and HGCT1.

> This subroutine is to solve the linearized matrix equation with the preconditioned conjugate gradient method using a polynomial as a preconditioner. It calls POLYT to invert the preconditioner.

Subroutine **Q34ADB**

It calls none.

It is called by ADVBC.

> This subroutine is to compose the element matrix equation over a boundary surface, either a triangular or a quadrilateral one, for computing both the boundary-surface water fluxes and material/heat fluxes at global nodes in the Lagrangian step. The nodal quadrature is used for surface integration if IQUAR = 1 or 2. The Gaussian quadrature is used for surface integration if IQUAR = 3 or 4. This subroutine is used for Approach 2 of chemical transport and heat transfer.

Subroutine **Q34ADB1**

It calls none.

It is called by ADVBC1.

> This subroutine is to compose the element matrix equation over a boundary surface, either a triangular or a quadrilateral one, for computing both the boundary-surface water fluxes and material/heat fluxes at global nodes in the Lagrangian step. The nodal quadrature is used for surface integration if IQUAR = 1 or 2. The Gaussian quadrature is used for surface integration if IQUAR = 3 or 4. This subroutine is used for Approach 1 of chemical transport and heat transfer.

Subroutine **Q34BB**

It calls none.

It is called by TSFLOW.

> This subroutine is to compose the element matrix equation over a boundary surface for computing the normal flow rates at global nodes. The basic idea of computing the normal flow rates (M/T) is to integrate the normal fluxes (M/L**2/T) over boundary surfaces as described below.

$$RRQ(i) = \int_{R_e} N_i^e \, F \, dB$$

> The nodal quadrature is used for surface integration if IQUAR = 1 or 2. The Gaussian quadrature is used for surface integration if IQUAR = 3 or 4.

Subroutine **Q34CNV**

It calls none.

It is called by TBC and TBC1.

> This subroutine is to compose the element matrix equation over a boundary surface for computing both the boundary surface water flux matrix and material/heat fluxes at global nodes in the Eulerian step. The boundary surface element matrix is defined as

$$BQ(i) = \int_{B_e} N_i^e \, \mathbf{V} N_j^e \, dB$$

> The material/heat flux is defined as

$$RQ(i) = \int_{B_e} N_i^e \, q \, dB$$

> where q is the Cauchy flux, the Neumann flux, or the Variable flux ($\mathbf{n} \cdot \mathbf{V} C_v$). The nodal quadrature is used for surface integration if IQUAR = 1 or 2. The Gaussian quadrature is used for surface integration if IQUAR = 3 or 4.

Subroutine **Q34S**

It calls none.

It is called by BCPREP, FBCN, and FSFLOW.

This subroutine is to compute the surface node flux of the type

$$RQ(\mathbf{I}) = \int_{B_e} N_i^e \frac{\rho}{\rho_o} q \, dB$$

where q is either the Cauchy flux, Neumann flux, or gravity flux. The nodal quadrature is used for surface integration if IQUAR = 1 or 2. The Gaussian quadrature is used for surface integration if IQUAR = 3 or 4.

Subroutine **Q468R**

It calls none.

It is called by TSFLOW.

This subroutine is to compose the element matrix equation over an element for computing both the material and source integrations as described below.

$$QRM = \int_R \theta C \, dR$$

$$QDM = \int_R \theta S \, dR$$

$$SOSM = \int_R \frac{QC_{in}(1 + \text{sign}(Q)) + QC(1 - \text{sign}(Q))}{2} \, dR$$

The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

Subroutine **Q468TH**

It calls SHAPE.

It is called by FSFLOW.

This subroutine is to compute the contribution of the increasing rate of the water content from an element e.

$$QTHP = \int_{R_e} \frac{\rho}{\rho_0} \frac{d\theta}{dh} \frac{\partial h}{\partial t} \, dR$$

The computation of the above integration is straightforward. The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

Subroutine **RADC**

It calls none.

It is called by RESIDU.

This subroutine is to evaluate the residuals of equations governing adsorption reactions with the simple surface complexation model (IADS = 0), the constant capacity adsorption model (IADS = 1), or the triple-layer adsorption model (IADS = 2) being considered.

Subroutine **RDATIO**

It calls READN.

It is called by MAIN and MAIN1.

This subroutine is to read and print material and domain data, including (1) material properties, (2) soil property parameters, (3) global node coordinates, (4) subregion data, (5) element connectivity, and (6) material types of all the elements .

Subroutine **READN**

It calls none.

It is called by FBCDAT, FSSDAT, RDATIO, TBCDAT, and TSSDAT.

This subroutine is to generate integer numbers automatically.

Subroutine **READR**

It calls none.

It is called by FBCDAT, FACT, FACT1, MAIN, and MAIN1.

This subroutine is to generate real numbers automatically.

Subroutine **REFINE**

It calls CKSIDE, FCOS, ONLINE, and ONPLAN.

It is called by HLCAP.

This subroutine is to refine the element where peaks/valleys exist.

Subroutine **REPLAS**

It calls none.

It is called by ALGBDY, BNDRY, ELTRK4, ELTRK6, ELTRK8, GNTRAC, GNTRAC1, HPTRAC, HPTRAC1, LOCQ2N, LOCQ3N, LOCQ4N, and MMLOC.

This subroutine is to replace the last six arguments with the first six arguments orderly.

Subroutine **RESIDU**

It calls RADC and RIES.

It is called by EQUILN.

This subroutine is to evaluate the residuals of all the nonlinear algebraic equations governing both mole balance and chemical equilibrium. It is straightforward to evaluate the residuals of both mole balance and precipitation/dissolution reaction equations. However, the subroutine calls RADC and RIES to give special care when calculating the residuals of adsorption reaction and ion-exchange reaction equations, respectively.

Subroutine **RHOAMU**

It calls none.

It is called by FASEMB, FBCN, SPROP, TASEMB, TASEMB1.

This subroutine is to compute temperature-dependent groundwater density and viscosity based on given empirical formulae.

Subroutine **RIES**

It calls none.

It is called by JIES and RESIDU.

This subroutine is to evaluate the residuals of equations governing ion-exchange reactions.

Subroutine **SHAPE**

It calls none.

It is called by FQ468, FQ468DV, Q468R, Q468TH, SPROP, TQ468, TQ4681, and TQ468DV.

This subroutine is to compute (1) the base functions, (2) the weighting functions, and (3) the derivatives, with respect to x, y, and z, of the base functions at a specific point inside the element being considered. In addition, it calculates the Jacobian for the element.

Subroutine **SOLVE**

It calls none.

It is called by BLKITR.

This subroutine is to solve the linearized matrix equation with a standard banded Gaussian elimination technique.

Subroutine **SPROP**

It calls RHOAMU and SHAPE.

It is called by FACT, FACT1, and HGCH.

This subroutine is to calculate the water contents, the hydraulic conductivities, and the water capacities of all the elements. Either tabular input or analytical functions can be used to represent soil property functions. When analytical functions are used, the users must supply the functional form. The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

Subroutine **SUR2D3**

It calls none.

It is called by LOCQ2N.

This subroutine is to provide a guaranteed solution in LOCQ2N when the single velocity approach is considered.

Subroutine **SURE3D**

It calls LOCPLN.

It is called by LOCQ3N and LOCQ4N.

This subroutine is to provide a guaranteed solution in LOCQ3N and LOCQ4N when the single velocity approach is considered.

Subroutine **SURF**

It calls none.

It is called by MAIN and MAIN1.

This subroutine is to generate three boundary geometry arrays, including DCOSB, ISB, and NPBB, used for implementing boundary conditions.

Subroutine **TACADC**

It calls none.

It is called by HGCT and HGCT1.

This subroutine is to solve the first order ODE (w.r.t. time) with first order decay terms for (1) immobile components, (2) ion-exchange capacities, and (3) adsorption surface areas.

Subroutine **TASEMB**

It calls LAMBDA, RHOAMU, TQ468, and VOLUME.

It is called by HGCT.

This subroutine is to compose the global matrix equation for computing chemical transport and heat transfer in the Eulerian step. To solve for the concentrations of conservative mobile components in chemical transport, this subroutine needs to be called only once. However, it needs to be called in each non-linear iteration in order to solve for the concentrations of non-conservative mobile components. This subroutine is used for Approach 2 of chemical transport and heat transfer.

Subroutine **TASEMB1**

It calls LAMBDA, RHOAMU, TQ4681, and VOLUME.

It is called by HGCT1.

This subroutine is to compose the global matrix equation for computing chemical transport and heat transfer in the Eulerian step. Its structure is similar to that of TASEMB. This subroutine is used for Approach 1 of chemical transport and heat transfer.

Subroutine **TBC**

It calls Q34CNV.

It is called by HGCT.

> This subroutine is to implement Cauchy, Neumann, Variable, and Dirichlet boundary conditions in the Eulerian step for Approach 2 of chemical transport and heat transfer. For a Dirichlet boundary condition, an identity algebraic equation is generated for each Dirichlet nodal point. Any other equations having this nodal variable are modified accordingly to simplify the computation. For the Variable boundaries, the integration of the normal velocity times the incoming concentration is added to the load vector and the integration of normal velocity is added to the matrix. For the Cauchy boundaries, the integration of Cauchy flux is added to the load vector and the integration of normal velocity is added to the matrix. For the Neumann boundaries, the iteration of gradient flux is added to the load vector.

Subroutine **TBC1**

It calls Q34CNV.

It is called by HGCT1.

> This subroutine is to implement Cauchy, Neumann, Variable, and Dirichlet boundary conditions in the Eulerian step for Approach 1 of chemical transport and heat transfer. Its structure is the same as that of TBC.

Subroutine **TBCDAT**

It calls READN.

It is called by MAIN and MAIN1.

> This subroutine is to read and print all the information for implementing the boundary conditions, including Dirichlet, Neumann, Cauchy, and Variable boundary conditions, of chemical transport and/or heat transfer.

Subroutine **TEMPEQ**

It calls none.

It is called by EQDCTJ, EQDCTJ1, TQ468, TQ4681, and HGCT1.

> This subroutine is to compute temperature-dependent chemical equilibrium constants.

Subroutine **TOTDSP**

It calls none.

It is called by EQMOD and EQUILN.

> This subroutine is to calculate (1) the total dissolved, (2) the total sorbed, and (3) the total precipitated concentrations of all the components. The calculation is done based on all the species concentrations previously computed as well as the composition relationship between components and species.

Subroutine **TPRINT**

It calls none.

It is called by FACT and FACT1.

> This subroutine is to print the simulation results of chemical transport at each time step. The print-out includes (1) the material fluxes through all types of boundaries (if KPR.GT.0), (2) the total analytical concentrations as well as the total dissolved concentrations at all the global nodes (if KPR.GT.1), and (3) the material fluxes at all the global nodes (if KPR.GT.3).

Subroutine **TPRITT**

It calls none.

It is called by FACT and FACT1.

> This subroutine is to print the simulation results of heat transfer --- temperature at all nodes.

<u>Subroutine **TQ468**</u>
It calls CDERIV, EQMOD, SHAPE, and TEMPEQ.
It is called by TASEMB.

This subroutine is to compose the element matrix equation over an element for solving (1) the transport equations of mobile components in Approach 2 and (2) the heat transfer equation in the Eulerian step. For chemical transport, it first calls EQMOD and CDERIV to compute chemical equilibrium and concentration derivatives,respectively, at quadrature points. Secondly, it takes the density effect into account if needed. Finally, it composes the element load vector as

$$QR(iq,j) = \int_{R_e} N_{iq}^e \left[\left(\frac{\rho^*}{\rho} + \frac{\rho_o}{\rho} \mathbf{V} \cdot \nabla \left(\frac{\rho}{\rho_o} \right) \right) (S_j + P_j) + q C_{in} \right] dR$$

and all the element matrices as

$$QA(iq,jq) = \int_{R_e} N_{iq}^e \theta N_{jq}^e \, dR$$

$$QD(iq,jq,i,j) = \int_{R_e} \left(\nabla N_{iq}^e \right) \cdot \theta \mathbf{D} \frac{\partial C_j}{\partial T_i} \cdot \left(\nabla N_{jq}^e \right) dR$$

$$QV(iq,jq,i,j) = \int_{R_e} N_{iq}^e \mathbf{V} \frac{\partial C_j}{\partial T_i} \cdot \left(\nabla N_{jq}^e \right) dR$$

$$QC(iq,jq) = \int_{R_e} N_{iq}^e \left[\frac{\partial \theta}{\partial t} + \frac{\rho^*}{\rho} q - \frac{\rho_o}{\rho} \mathbf{V} \cdot \nabla \left(\frac{\rho}{\rho_o} \right) \right] N_{jq}^e dR$$

The computation for heat transfer is straightforward because temperature can be considered as the total analytical concentration of a conservative mobile component. The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

<u>Subroutine **TQ4681**</u>
It calls EQMOD, SHAPE, and TEMPEQ.
It is called by TASEMB1.

This subroutine is to compose the element matrix equation over an element for solving (1) the transport equations of mobile components in Approach 1 and (2) the heat transfer equation in the Eulerian step. For chemical transport, it first calls EQMOD to compute chemical equilibrium at quadrature points. Secondly, it takes the density effect into account if needed. Finally, it composes all the element matrices as

$$QA(iq,jq) = \int_{R_e} N_{iq}^e \theta N_{jq}^e \, dR$$

$$QD(iq,jq) = \int_{R_e} \left(\nabla N_{iq}^e \right) \cdot \theta \mathbf{D} \cdot \left(\nabla N_{jq}^e \right) dR$$

$$QV(iq,jq) = \int_{R_e} N_{iq}^e \mathbf{V} \frac{\partial C_j}{\partial T_i} \cdot \left(\nabla N_{jq}^e \right) dR$$

$$QC(iq,jq) = \int_{R_e} N_{iq}^e \left[\frac{\partial \theta}{\partial t} + \frac{\rho^*}{\rho} q - \frac{\rho_o}{\rho} \mathbf{V} \cdot \nabla \left(\frac{\rho}{\rho_o} \right) \right] N_{jq}^e dR$$

and the element load vector as

$$QR(iq, j) = \int_{R_e} N_{iq}^e \left[\left(\frac{\rho^*}{\rho} + \frac{\rho_o}{\rho} \mathbf{V} \cdot \nabla \left(\frac{\rho}{\rho_o} \right) \right) (S_j + P_j) + q C_{in} + \frac{\partial \theta}{\partial t} C_j \right] dR$$

The computation for heat transfer, as mentioned earlier, is straightforward. The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

Subroutine **TQ468DV**

It calls SHAPE.

It is called by FLUX.

This subroutine is to compose the element matrix equations over an element for computing the material fluxes at all the global nodes. In this subroutine, the element matrix equation is composed based on Fick's law. The element matrices are written as

$$QB(I, J) = \int_{R_e} N_i^e N_j^e \, dR$$

The element load vectors are written as:

$$QRX(I) = - \int_{R_e} N_i^e \, \mathbf{i} \cdot \theta \mathbf{D} \cdot \left(\nabla N_j^e \right) C_j \, dR$$

$$QRY(I) = - \int_{R_e} N_i^e \, \mathbf{j} \cdot \theta \mathbf{D} \cdot \left(\nabla N_j^e \right) C_j \, dR$$

$$QRZ(I) = - \int_{R_e} N_i^e \, \mathbf{k} \cdot \theta \mathbf{D} \cdot \left(\nabla N_j^e \right) C_j \, dR$$

where C_j is the concentration at nodal point j, $\mathbf{i}$ is the unit vector along the x-direction, $\mathbf{j}$ is the unit vector along the y-coordinate, $\mathbf{k}$ is the unit vector along the z-coordinate, is the water content, and $\mathbf{D}$ is the dispersion coefficient tensor. The Gaussian quadrature is used for element integration if IQUAR = 1 or 3. The nodal quadrature is used for volume integration if IQUAR = 2 or 4.

Subroutine **TRAK1H**

It calls LOCQ4N and PLANEW.

It is called by ELTRK8.

This subroutine computes the particle tracking in a specified hexahedral subelement when the source point coincides with a node of the subelement. This subroutine calls PLANEW to determine (1) whether the particle would move into the subelement or not, and (2) which side (a quadrilateral side) of the subelement the particle would head onto if the particle does move into the subelement. After determining which side the particle is going to move onto, this subroutine calls LOCQ4N to compute the exact location of the target point on the side. For accuracy, using the average velocity of both the source point and the target point to locate the target point is firstly considered in the subroutine. However, if this average velocity approach is not able to deal with very complex velocity fields, the single velocity of the source point is used to determine the location of the target point.

Subroutine **TRAK1P**

It calls LOCQ3N, LOCQ4N and PLANEW.

It is called by ELTRK6.

This subroutine is to compute the particle tracking in a specified triangular prism subelement when the source point coincides with a node of the subelement. This subroutine calls PLANEW to determine (1) whether the particle would move into the subelement or not, and (2) which side (either a quadrilateral or a triangular side) of the subelement the particle would head onto if the particle does move into the subelement. After determining

which side the particle is going to move onto, this subroutine calls LOCQ4N (if the side is a quadrilateral one) or LOCQ3N (if the side is a triangular one) to compute the exact location of the target point on the side. For accuracy, using the average velocity of both the source point and the target point to locate the target point is firstly considered in the subroutine. However, if this average velocity approach is not able to deal with very complex velocity fields, the single velocity of the source point is used to determine the location of the target point.

Subroutine **TRAK1T**

It calls LOCQ3N and PLANEW.

It is called by ELTRK4.

This subroutine is to compute the particle tracking in a specified tetrahedral subelement when the source point coincides with a node of the subelement. This subroutine calls PLANEW to determine (1) whether the particle would move into the subelement or not, and (2) which side (a triangular side) of the subelement the particle would head onto if the particle does move into the subelement. After determining which side the particle is going to move onto, this subroutine calls LOCQ3N to compute the exact location of the target point on the side. For accuracy, using the average velocity of both the source point and the target point to locate the target point is firstly considered in the subroutine. However, if this average velocity approach is not able to deal with very complex velocity fields, the single velocity of the source point is used to determine the location of the target point.

Subroutine **TRAK2H**

It calls LOCQ4N and PLANEW.

It is called by ELTRK8.

This subroutine is to compute the particle tracking in a specified hexahedral subelement when the source point dose not coincide with a node of the subelement. It contains computation basically the same as that in TRAK1H.

Subroutine **TRAK2P**

It calls LOCQ3N, LOCQ4N and PLANEW.

It is called by ELTRK6.

This subroutine computes the particle tracking in a specified triangular prism subelement when the source point does not coincide with a node of the subelement. It contains computation basically the same as that in TRAK1P.

Subroutine **TRAK2T**

It calls LOCQ3N and PLANEW.

It is called by ELTRK4.

This subroutine computes the particle tracking in a specified tetrahedral subelement when the source point does not coincide with a node of the subelement. It contains computation basically the same as that in TRAK1T.

Subroutine **TSFLOW**

It calls Q34BB and Q468R.

It is called by FACT and FACT1.

This subroutine is to compute (1) material fluxes, (2) incremental material flow, and (3) accumulated material flow through all types of boundaries. In addition, it computes the amounts of material changes in the region of interest. The items this subroutine computes are described as the following. FRATE(7) is to store the flux through the whole boundary

$$FRATE(7) = \int_B \left(F_x n_x + F_y n_y + F_z n_z \right) dB$$

where B is the global boundary of the region of interest; F_x, F_y, and F_z are the flux components; and n_x, n_y, and n_z are the directional cosines of the outward unit vector normal to the boundary B. FRATE(1), FRATE(2), and

FRATE(3) store the flux rates through the Dirichlet, Cauchy, and Neumann boundaries, respectively. FRATE(4) and FRATE(5) store incoming flux and outgoing flux rates, respectively, through the Variable boundaries B_{V^-} and B_{V^+} , as given by

$$FRATE(1) = \int_{B_d} \left(F_x n_x + F_y n_y + F_z n_z \right) dB$$

$$FRATE(2) = \int_{B_c} \left(F_x n_x + F_y n_y + F_z n_z \right) dB$$

$$FRATE(3) = \int_{B_n} \left(F_x n_x + F_y n_y + F_z n_z \right) dB$$

$$FRATE(4) = \int_{B_{V^-}} \left(F_x n_x + F_y n_y + F_z n_z \right) dB$$

$$FRATE(5) = \int_{B_{V^+}} \left(F_x n_x + F_y n_y + F_z n_z \right) dB$$

where B_{V^-} and B_{V^+} are that part of Variable boundary where the fluxes are directed into the region and out from the region, respectively. FRATE(6) stores the flux rate through unspecified boundaries as

$$FRATE(6) = FRATE(7) - \sum_{I=1}^{5} FRATE(I)$$

FRATE(8) and FRATE(9) storing the accumulate rate in aqueous and solid phases, respectively, are given by

$$FRATE(8) = \int_R \frac{\partial \theta C}{\partial t} dR$$

$$FRATE(9) = \int_R \frac{\partial \theta S}{\partial t} dR$$

FRATE(10) stores the rate loss due to decay and FRATE(11) through FRATE(13) are set to zero as given by

$$FRATE(10) = \int_R \lambda (\theta C + \theta S) dR$$

$$FRATE(11) = FRATE(12) = FRATE(13) = 0$$

FRATE(14) is used to store the source/sink rate as

$$FRATE(14) = \int_R QC_{in} \frac{1 + \text{sign}(Q)}{2} + QC \frac{1 - \text{sign}(Q)}{2} dR$$

If there is no numerical error in the computation, the following equation should be satisfied:

$$\sum_{I=7}^{14} FRATE(I) = 0$$

and FRATE(6) should be equal to zero.

Subroutine **TSSDAT**
It calls READN.
It is called by MAIN and MAIN1.

This subroutine is to read and print all the information for implementing the sources/sinks, including element (distributed), and well (point) sources/sinks, of chemical transport and/or heat transfer.

Subroutine **TSTORE**
It calls none.
It is called by FACT and FACT1.

This subroutine is to store pertinent quantities, including (1) the time steps of interest, the times of interest, and the time step sizes, and (2) the total analytical concentrations, the total dissolved concentrations, and the material fluxes at all the global nodes at the time being.

Subroutine **TSTORT**
It calls none.
It is called by FACT and FACT1.

This subroutine is to store pertinent quantities, including (1) the time steps of interest, the times of interest, and the time step sizes, and (2) temperature at all global nodes at the time being.

Subroutine **VALBDL**
It calls none.
It is called by ELTRK4, ELTRK6, and ELTRK8.

This subroutine is to calculate three interpolated values by the passed working arrays and basis functions.

Subroutine **VBVFCT**
It calls none.
It is called by FACT and FACT1.

This subroutine is to find the profile value, of Variable boundary conditions,at time t either by tabular interpolation or by analytical expression. For the latter case, the user must provide the function after "200 CONTINUE" in the subroutine.

Subroutine **VELT**
It calls FQ468DV.
It is called by FACT, FACT1, and HGCH.

This subroutine is to compute the Darcy fluxes at all the global nodes. It firstly calls FQ468DV to evaluate the element matrices and the derivatives of the total heads. It then sums over all element matrices to form the global matrix equation governing the velocity components at all nodal points. To save computational time, the matrix is diagonalized by lumping. The velocity components can thus be solved point by point.

Subroutine **VOLUME**
It calls none.
It is called by FASEMB, FSFLOW, NODVAL, TASEMB, and TASEMB1.

This subroutine is to calculate the volume of a given element.

Subroutine **WSSFCT**
It calls none.
It is called by FACT and FACT1.

This subroutine is to find the profile value, of well (point) sources/sinks, at time t either by tabular interpolation or by analytical expression. For the latter case, the user must provide the function after "200 CONTINUE" in the subroutine.

Subroutine **XSI2D**

It calls none.

It is called by BASE2D.

>This subroutine is to compute the local coordinates at a specific point that is within a quadrilateral element. The coordinates for both the specified point and element nodes are given.

Subroutine **XSI3D**

It calls none.

It is called by BASE.

>This subroutine is to compute the local coordinates at a specific point that is within a hexahedral element. The coordinates for both the specified point and element nodes are given.

Subroutine **XSI3DP**

It calls none.

It is called by BASE.

>This subroutine is to compute the local coordinate (in the vertical direction) and the natural coordinate (in the horizontal directions) at a specific point that is within a triangular prism element. The global coordinates for both the specified and element nodes are given.

Appendix C PARAMETER SPECIFICATION FOR ARRAY DIMENSIONS

In 3DHYDROGEOCHEM, there are totally 96 parameters, including 10 for chemical reactions, 21 for common use, 21 for subsurface flow, 21 for chemical transport, 21 for heat transfer, and 2 for the "in-element" particle tracking, that must be given values at the beginning of the computer codes such that array dimensions can be specified. They are defined as follows.

For chemical reactions:
MAXNK	=	Maximum number of components.
MAXMK	=	Maximum number of species.
MAXMZK	=	Maximum number of ion-exchanged species.
MAXMPK	=	Maximum number of precipitated species.
MAXPDK	=	Maximum number of product species.
MAXEQK	=	Maximum number of equations to be solved in computing chemical equilibrium.
MXNSTK	=	Maximum number of ion-exchange sites.
MXNSBK	=	Maximum number of adsorption sites.
MAXDPK	=	Maximum number of dominating precipitated species.

For common use:
MAXNPK	=	Maximum number of global nodes.
MAXELK	=	Maximum number of elements.
MXBNPK	=	Maximum number of boundary nodes.
MXBESK	=	Maximum number of boundary sides.
MXTUBK	=	Maximum number of total unspecified boundary sides connected to boundary nodes.
MXNP1K	=	Maximum number of global nodes. (It can be set to only 1 when either Approach 2 or ICOMP=1 in Approach 1 is considered)
MXNPGK	=	Maximum number of fine-grid nodes.
MXKGLK	=	Maximum number of fine grids.
MXPVK	=	Maximum number of peaks/valleys in an element.
MXJBDK	=	Maximum number of global nodes that would connect to a global node.
MXKBDK	=	Maximum number of elements that would connect to a global node.
MXNTIK	=	Maximum number of time intervals.
MXDTCK	=	Maximum number of the reset of time-step sizes.
MXJD1K	=	Maximum number of global nodes that would connect to a global node. (It can be set to 1 when either Approach 2 or ICOMP=1 in Approach 1 is considered)
LTMXNK	=	Maximum number of nodes related to a subregion.
LMXNPK	=	Maximum number of nodes in a subregion.
LMXBWK	=	Maximum number of band width for a subregion.
MXRGNK	=	Maximum number of subregions for block iteration solvers.
MXMATK	=	Maximum number of materials.
MXSPMK	=	Maximum number of soil parameters.
MXMPMK	=	Maximum number of material parameters.

For subsurface flow:
MXSELH	=	Maximum number of element sources/sinks for subsurface flow.
MXSPRH	=	Maximum number of profiles for flow element sources/sinks.
MXSDPH	=	Maximum number of nodes needed to present a profile for flow element sources/sinks.
MXWNPH	=	Maximum number of point sources/sinks for subsurface flow.
MXWPRH	=	Maximum number of profiles for flow point sources/sinks.

MXWDPH	=	Maximum number of nodes needed to present a profile for flow point sources/sinks.
MXCNPH	=	Maximum number of Cauchy boundary nodes for subsurface flow.
MXCESH	=	Maximum number of Cauchy boundary sides for subsurface flow.
MXCPRH	=	Maximum number of profiles for flow Cauchy boundary conditions.
MXCDPH	=	Maximum number of nodes needed to present a profile for flow Cauchy boundary conditions.
MXNNPH	=	Maximum number of Neumann boundary nodes for subsurface flow.
MXNESH	=	Maximum number of Neumann boundary sides for subsurface flow.
MXNPRH	=	Maximum number of profiles for flow Neumann boundary conditions.
MXNDPH	=	Maximum number of nodes needed to present a profile for flow Neumann boundary conditions.
MXVNPH	=	Maximum number of Variable boundary nodes for subsurface flow.
MXVESH	=	Maximum number of Variable boundary sides for subsurface flow.
MXVPRH	=	Maximum number of profiles for flow Variable boundary conditions.
MXVDPH	=	Maximum number of nodes needed to present a profile for flow Variable boundary conditions.
MXDNPH	=	Maximum number of Dirichlet boundary nodes for subsurface flow.
MXDPRH	=	Maximum number of profiles for flow Dirichlet boundary conditions.
MXDDPH	=	Maximum number of nodes needed to present a profile for flow Dirichlet boundary conditions.

For chemical transport:

MXSELC	=	Maximum number of element sources/sinks for chemical transport.
MXSPRC	=	Maximum number of profiles for transport element sources/sinks.
MXSDPC	=	Maximum number of nodes needed to present a profile for transport element sources/sinks.
MXWNPC	=	Maximum number of point sources/sinks for chemical transport.
MXWPRC	=	Maximum number of profiles for transport point sources/sinks.
MXWDPC	=	Maximum number of nodes needed to present a profile for transport point sources/sinks.
MXCNPC	=	Maximum number of Cauchy boundary nodes for chemical transport.
MXCESC	=	Maximum number of Cauchy boundary sides for chemical transport.
MXCPRC	=	Maximum number of profiles for transport Cauchy boundary conditions.
MXCDPC	=	Maximum number of nodes needed to present a profile for transport Cauchy boundary conditions.
MXNNPC	=	Maximum number of Neumann boundary nodes for chemical transport.
MXNESC	=	Maximum number of Neumann boundary sides for chemical transport.
MXNPRC	=	Maximum number of profiles for transport Neumann boundary conditions.
MXNDPC	=	Maximum number of nodes needed to present a profile for transport Neumann boundary conditions.
MXVNPC	=	Maximum number of Variable boundary nodes for chemical transport.
MXVESC	=	Maximum number of Variable boundary sides for chemical transport.
MXVPRC	=	Maximum number of profiles for transport Variable boundary conditions.
MXVDPC	=	Maximum number of nodes needed to present a profile for transport Variable boundary conditions.
MXDNPC	=	Maximum number of Dirichlet boundary nodes for chemical transport.
MXDPRC	=	Maximum number of profiles for transport Dirichlet boundary conditions.
MXDDPC	=	Maximum number of nodes needed to present a profile for transport Dirichlet boundary conditions.

For heat transfer:

MTXSEL	=	Maximum number of element sources/sinks for heat transfer.
MTXSPR	=	Maximum number of profiles for transfer element sources/sinks.
MTXSDP	=	Maximum number of nodes needed to present a profile for transfer element sources/sinks.
MTXWNP	=	Maximum number of point sources/sinks for heat transfer.
MTXWPR	=	Maximum number of profiles for transfer point sources/sinks.
MTXWDP	=	Maximum number of nodes needed to present a profile for transfer point sources/sinks.
MTXCNP	=	Maximum number of Cauchy boundary nodes for heat transfer.
MTXCES	=	Maximum number of Cauchy boundary sides for heat transfer.
MTXCPR	=	Maximum number of profiles for transfer Cauchy boundary conditions.
MTXCDP	=	Maximum number of nodes needed to present a profile for transfer Cauchy boundary conditions.
MTXNNP	=	Maximum number of Neumann boundary nodes for heat transfer.
MTXNES	=	Maximum number of Neumann boundary sides for heat transfer.
MTXNPR	=	Maximum number of profiles for transfer Neumann boundary conditions.
MTXNDP	=	Maximum number of nodes needed to present a profile for transfer Neumann boundary conditions.
MTXVNP	=	Maximum number of Variable boundary nodes for heat transfer.
MTXVES	=	Maximum number of Variable boundary sides for heat transfer.
MTXVPR	=	Maximum number of profiles for transfer Variable boundary conditions.
MTXVDP	=	Maximum number of nodes needed to present a profile for transfer Variable boundary conditions.
MTXDNP	=	Maximum number of Dirichlet boundary nodes for heat transfer.
MTXDPR	=	Maximum number of profiles for transfer Dirichlet boundary conditions.
MTXDDP	=	Maximum number of nodes needed to present a profile for transfer Dirichlet boundary conditions.

For the "in-element" particle tracking:

MXNPWK	=	Maximum number of working fine-grid nodes in an element for the "in-element" particle tracking.
MXELWK	=	Maximum number of working subelements in an element for the "in-element" particle tracking.

To run the four application examples demonstrated in Chapter 4, the 96 parameters must be given values not less than the following.

For chemical reactions:
```
      PARAMETER (MAXNK=8,MAXMK=35,MAXMZK=9,MAXMPK=3,MAXPDK=28,
   >              MAXEQK=16,MXNSTK=3,MXNSBK=1,MAXDPK=3)
```

For common use:
```
      PARAMETER(MAXNPK=9261,MAXELK=8000,MXBNPK=2402,
   >              MXBESK=2400,MXTUBK=6400,MXNP1K=9261)
      PARAMETER(MXNPGK=1,MXKGLK=1,MXPVK=8)
      PARAMETER(MXJBDK=27,MXKBDK= 8,MXNTIK=20,MXDTCK=1,MXJD1K=27)
      PARAMETER(LTMXNK=1,LMXNPK=1,LMXBWK=1,MXRGNK=1)
      PARAMETER(MXMATK=3,MXSPMK=4,MXMPMK=9)
```

For subsurface flow:
```
      PARAMETER(MXSELH=1,MXSPRH=1,MXSDPH=1,MXWNPH=3,MXWPRH=1,
     >            MXWDPH=3)
      PARAMETER(MXCNPH=1,MXCESH=1,MXCPRH=1,MXCDPH=1)
      PARAMETER(MXNNPH=1,MXNESH=1,MXNPRH=1,MXNDPH=1)
      PARAMETER(MXVNPH=1,MXVESH=1,MXVPRH=1,MXVDPH=1)
      PARAMETER(MXDNPH=882,MXDPRH=2,MXDDPH=4)
```

For chemical transport:
```
      PARAMETER(MXSELC=1,MXSPRC=1,MXSDPC=1,MXWNPC=3,
     >            MXWPRC=1,MXWDPC=3)
      PARAMETER(MXCNPC=1,MXCESC=1,MXCPRC=1,MXCDPC=1)
      PARAMETER(MXNNPC=1,MXNESC=1,MXNPRC=1,MXNDPC=1)
      PARAMETER(MXVNPC=882,MXVESC=800,MXVPRC=5,MXVDPC=3)
      PARAMETER(MXDNPC=1,MXDPRC=1,MXDDPC=1)
```

For heat transfer:
```
      PARAMETER(MTXSEL=1,MTXSPR=1,MTXSDP=1,MTXWNP=3,
     >         MTXWPR=1,MTXWDP=3)
      PARAMETER(MTXCNP=1,MTXCES=1,MTXCPR=1,MTXCDP=1)
      PARAMETER(MTXNNP=1,MTXNES=1,MTXNPR=1,MTXNDP=1)
      PARAMETER(MTXVNP=882,MTXVES=800,MTXVPR=2,MTXVDP=3)
      PARAMETER(MTXDNP=1520,MTXDPR=1,MTXDDP=3)
```

For the "in-element particle tracking":
```
      PARAMETER(MXNPWK=36,MXELWK=12)
```

REFERENCES

Cheng, H. P. and G. T. Yeh, 1997a, *On Simulating the Reactive Chemical Transport with Dominating Precipitated Species Controlling Chemical Equilibrium* (submitted).

Cheng, H. P. and G. T. Yeh, 1997b, *On simulating temperature-dependent reactive chemical transport in porous media under chemical equilibrium* (submitted).

Cheng, H. P., J. R. Cheng, and G. T. Yeh, 1996a. *A particle tracking technique for the Lagrangian-Eulerian finite element method in multi-dimensions.* International Journal for Numerical Methods in Engineering, 39(7), 1115-1136.

Cheng, J. R., H. P. Cheng, and G. T. Yeh, 1996b. *A Lagrangian-Eulerian method with adaptively local zooming and peak/valley capturing approach to solve two-dimensional advection-diffusion transport equations.* International Journal for Numerical Methods in Engineering, 39(6), 987-1016.

Cheng H. P., 1995, *Development and application of a three-dimensional finite element model of subsurface flow, heat transfer, and reactive chemical transport*, Ph.D. thesis, Department of Civil Engineering, The Pennsylvania State University, University Park, PA 16802.

Cheng, H. P. and G. T. Yeh, 1994a, *An Approach on Simulating Reactive Chemical Transport with Precipitated Species Dominating Chemical Equilibrium*, X international conference on computational methods in water resources, Heidelberg, Germany, July 19-22, 1994, 597-604.

Cheng, H. P. and G. T. Yeh, 1994b, *An Approach Simulating Reactive Chemical Transport with Dominating Precipitated Species Controlling Chemical Equilibrium*, EOS American Geophysical Union 75(44):209.

Cheng, H. P. and G. T. Yeh, 1993a, *A retardation effect-bases Lagrangian-Eulerian method on solving hydrogeochemical transport equations*, EOS American Geophysical Union 74(16):156.

Cheng, H. P. and G. T. Yeh, 1993b, *A two-dimensional flow and hydrogeochemical transport model*, Proc. 1993 National Conference on Hydraulic Engineering and International Symposium on Engineering Hydrology, San Francisco, July 25-30, 1993, 617-622.

Iskra, G, 1994, *The Verification and Validation of A General Mixed Chemical Kinetic and Equilibrium Model (KEMOD)*, M. S. Thesis, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802.

Lide, D. R., Editor-in-Chief, 1992, *CRC Handbook of Chemistry and Physics*, 73rd edition, CRC Press Inc.

Salvage, K. M. and G. T. Yeh, 1998, *Development and application of a numerical model of kinetic and equilibrium microbiological and geochemical reactions (BIOKEMOD),* J. Hydrol., 209(1-4):27-52.

Yeh, G. T. and H. P. Cheng, 1996, *On diagonalization of coupled hydrologic transport and geochemical reaction equations.* Book Chapter, IMA Volumes in Mathematics and its Applications, Vol. 79 --- Environmental Studies: Mathematical, computational, and statistical analysis. Edited by M. F. Wheeler. Springer-Verlag, New York, 373-398.

Yeh, G. T., G. A. Iskra, J. E. Szecsody, J. M. Zachara, and G. P. Streile, 1995a, *KEMOD: A mixed chemical kinetic and equilibrium model of aqueous and solid phase geochemical reactions*, technical report, PNL-10380, Pacific Northwest Laboratory, Richland, WA 99352.

Yeh, G. T., S. L. Carpenter, P. L. Hopkins, and M. D. Siegel, 1995b, *Users' Manual for LEHGC: A Lagrangian-Eulerian finite element model of HydroGeoChemical transport through saturated-unsaturated media*, technical report, SAND95-1121, Sandia National Laboratories, Albuquerque, NM 87185.

Yeh, G. T., J. R. Cheng, and H. P. Cheng, 1994, *3DFEMFAT: User's Manual of a 3-Dimensional Finite Element Model of Flow and Transport through Saturated-Unsaturated Media*, technical report, Department of Civil and Environmental Engineering, the Pennsylvania State University, University Park, PA 16802.

Yeh, G. T. and J. R. Cheng, 1993a, *3DFEMWATER: A 3-Dimensional Finite Element Model of WATER Flow through Saturated-Unsaturated Media: Version 2.0*, technical report, Department of Civil and Environmental Engineering, the Pennsylvania State University, University Park, PA 16802.

Yeh, G. T. and J. R. Cheng, 1993b, *3DLEWASTE: A 3-Dimensional Hybrid Lagrangian-Eulerian Finite Element Model of WASTE Transport through Saturated-Unsaturated Media: Version 2.0*, technical report, Department of Civil and Environmental Engineering, the Pennsylvania State University, University Park, PA 16802.

Yeh, G. T. and J. R. Cheng, 1993c, *LEWASTE: A 2-Dimensional Hybrid *L*agrangian-*E*ulerian Finite Element Model of *WASTE* Transport through Saturated-Unsaturated Media: Version 2.0*, technical report, Department of Civil and Environmental Engineering, the Pennsylvania State University, University Park, PA 16802.

Yeh, G. T., 1992, *EQMOD: An *EQ*uilibrium *MOD*el* of Aqueous and Solid Phase Geochemical Reactions, technical report, Department of Civil and Environmental Engineering, the Pennsylvania State University, University Park, PA 16802.

Yeh, G. T. and V. S. Tripathi, 1991, *A model for simulating transport of reactive multispecies components: model development and demonstration*, Water Resour. Res., 27(12), 3075-3094.

Yeh, G. T. and V. S. Tripathi, 1990, *HYDROGEOCHEM: A Coupled Model of HYDROlogical Transport and GEOCHEMical Equilibria in Reactive Multicomponent Systems*, ORNL-6371, Oak Ridge National Laboratory, Oak Ridge, TN 37831.

Yeh, G. T., 1990, *A Lagrangian-Eulerian method with zoomable hidden fine-mesh approach to solving advection-dispersion equations*, Water Resour. Res., 26(6),1133-1144.

Yeh, G. T. and R. J. Luxmoore, 1983, *MATTUM: A multidimensional model for simulating *M*oisture *A*nd *T*hermal *T*ransport in *U*nsaturated porous *M*edia*, Report ORNL-5888, Oak Ridge National Laboratory, Oak Ridge, TN 37831.

Yeh, G. T. and D. S. Ward, 1981, *FEMWASTE: A *F*inite-*E*lement *M*odel of *WASTE* transport through saturated/unsaturated porous media*, Report ORNL-5601, Oak Ridge National Laboratory, Oak Ridge, TN 37831.

Yeh, G. T. and D. S. Ward, 1980, *FEMWATER: A *F*inite-*E*lement *M*odel of *WATER* flow through saturated/unsaturated porous media*, Report ORNL-5567, Oak Ridge National Laboratory, Oak Ridge, TN 37831.