DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via the U.S. Department of Energy (DOE) Information Bridge:

Web site: http://www.osti.gov/bridge

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: 703-605-6000 (1-800-553-6847)
TOD: 703-487-4639
Fax: 703-605-6900
E-mail: info@ntis.fedworld.gov
Web site: http://www.ntis.gov/support/ordernowabout.htm

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange (ETDE) representatives, and International Nuclear Information System (INIS) representatives from the following source:

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831
Telephone: 865-576-8401
Fax: 865-576-5728
E-mail: reports@adonis.osti.gov
Web site: http://www.osti.gov/contact.html

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.
HYDROGEOCHEM 5.0: A Three-Dimensional Model of Coupled Fluid Flow, Thermal Transport, and HYDROGEOCHEMical Transport through Variably Saturated Conditions - Version 5.0

Gour-Tsyh (George) Yeh and Jiangtao Sun
Department of Civil and Environmental Engineering
University of Central Florida
Orlando, FL 32816

Philip M. Jardine
Environmental Science Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831

William D. Burgos and Yilin Fang
Department of Civil and Environmental Engineering
The Pennsylvania State University
University Park, PA 16802

Ming-Hsu Li
Institute of Hydrology
National Central University
Chunli, Taiwan 32054

Malcolm D. Siegel
P. O. Box 5800
Sandia National Laboratory
Albuquerque, NM 87185

Date Issued - May 2004
ABSTRACT

The computer program HYDROGEOCHEM 5.0 is a numerical model of fluid flow, thermal transport, hydrologic transport, and biogeochemical kinetic/equilibrium reactions in saturated/unsaturated media. It is a three-dimensional version of the two-dimensional HYDROGEOCHEM 4.0. It iteratively solves three-dimensional fluid flow, heat transfer, and reactive biogeochemical transport equations. The Richards equation is solved for fluid flow, the thermal transport equation is solved for heat transfer, and the advection-dispersion-reactive transport equations are solved for all mobile component equations and kinetic-variable equations.

HYDROGEOCHEM 5.0 is designed for generic applications to reactive transport problems controlled by both kinetic and equilibrium reactions in subsurface media. Input to the program includes the geometry of the system, the spatial distribution of finite elements and nodes, the properties of the media, reaction network, and the initial and boundary conditions. Output includes the spatial distribution of pressure head, total head, velocity fields, moisture contents, temperature, and biogeochemical concentrations as a function of time as well as the distribution of all biogeochemical species at user-specified nodes.

HYDROGEOCHEM 5.0 is the first simulator of coupled fluid flow, hydrologic transport, heat transfer, and biogeochemical transport under variably saturated conditions in three dimensions.
ACKNOWLEDGMENTS

This research was supported, in part, by National Science Foundation (NSF) Grant No.EAR-0196048 with University of Central Florida and, in part, by the Natural and Accelerated Bioremediation Research Program (NABIR), Office of Biological and Environmental Research (OBER), U.S. Department of Energy (DOE) Grant No. ER63181-107643-0007253 with University of Central Florida. Funding for this research was partially provided by the Office of Science Biological and Environment Research, the Natural and Accelerated Bioremediation (NABIR) Program, U.S. Department of Energy, under Contract DE-AC05-00OR22725 with Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC. The authors appreciate the efforts of Paul Bayer, program manager of NABIR.

The development of this latest version reflects the many contributions made by a small army of colleagues. Basing on the original model, HYDROGEOCHEM, Dr. Hwai-Ping (Pearce) Cheng carried out the incorporation of multiple ion-exchange sites and multiple adsorbent components in LEHGC 1.0 and LEHGC 1.1. Dr. Woohee Choi helped in the incorporation of colloid transport in LEGHC 1.0. Dr. Jing-Ru (Ruth) Cheng contributed to and made numerous debugging in LEHGC 1.0 and LEHGC 1.1. Dr. Karen Salvage was heavily involved in the implementation of mixed equilibrium and kinetic reactions in HYDROGEOCHEM 2.0 and HYDROGEOCHEM 2.1. Dr. Ming-Hsu Li performed numerous debugging, made LEHGC 2.0 operational, and verified and applied it to many problems after the code was personally programmed by Dr. Yeh himself. Dr. Yilin Fang incorporated the removal of water due to precipitation-dissolution reactions in HYDROGEOCHEM 3.1 and most importantly implemented the new paradigm of reaction-based biogeochemical processes in HYDROGEOCHEM 3.2. Basing on HYDROGEOCHEM 3.2, Ms. Yuan Li incorporated heat transfer module and made some important modifications in HYDROGEOCHEM 4.0. The authors acknowledge the contribution of these investigators who make the completion of this latest version possible.
CONTENTS

Section                           Page

Abstract ........................................ iii
Acknowledgments ................................ v
List of Figures .............................. xi
List of Tables ............................... xiii

1.0 INTRODUCTION ........................... 1

2.0 MATHEMATICAL FORMULATION ........ 5

2.1 Flow Equations .......................... 5

2.1.1 Governing Equations ................. 5
2.1.2 Initial and Boundary Conditions for Flow Simulations ....... 6
2.1.3 Dependence of Fluid Density and Viscosity on Biogeochemical Concentrations .. 8
2.1.4 Effect of Precipitation/Dissolution on Hydraulic Conductivity and Porosity .... 9

2.2 Reactive Biogeochemical Transport Equations ................. 10

2.2.1 Species Transport Equations ........ 10
2.2.2 Generalized Reactive Biogeochemical Transport in Variably Saturated Media .... 13
2.2.3 Decomposition of the Reactive Transport Equation ........ 14
2.2.4 Reaction Rate Formulation .......... 17
2.2.5 Initial and Boundary Conditions for Reactive Transport ........ 18
2.2.6 Effect of Precipitation/Dissolution on Hydrodynamic Dispersion ........ 20

2.3 Heat Transfer Equations ................. 21

2.3.1 Governing Equations ................. 21
2.3.2 Conversion to Advective Form ........ 22
2.3.3 Initial and Boundary Conditions ........ 23
2.3.4 Effect of temperature on the constants of biogeochemical reactions ......... 24

3.0 NUMERICAL IMPLEMENTATION .......... 25

3.1 Solution of Fluid Flow Equations ......... 26

3.1.1 Spatial Discretization of Flow Equations with the Galerkin Finite Element Method .......... 26
3.1.2 Base and Weighting Functions .......... 28
3.1.3 Numerical Integration ................. 29
Example 2: Transient and Varied Saturated Flow .............................................. 117
Example 3: Steady and Transient Simulations for Varied Saturated Flow ............. 117
Example 4: Horizontal Solute Transport .......................................................... 118
Example 5: Steady Solute Transport ................................................................... 120
Example 6: Steady and Transient Solute Transport .......................................... 121
Example 7: Steady Henry’s Salt Water Intrusion Problem .................................. 122
Example 8: Transient Henry’s Salt Water Intrusion Problem ............................. 122
Example 9: Uranium Milling Tailing Problem .................................................. 125
Example 10: Flow and Advective-Dispersive-Reactive Transport Problem (1) ....... 134
Example 11: Flow and Advective-Dispersive-Reactive Transport Problem (2) ...... 140
Example 12: Flow and Advective-Dispersive-Reactive Transport Problem (3) ...... 144
Example 13: Coupled Fracture Flow and Matrix Diffusive-Reactive Transport ....... 148
Example 14: 1-D Chemical Transport Verification ............................................. 155
Example 15: 3-D Flow and Chemical Transport with Complexation, Adsorption, ion-exchange and mineral dissolution reactions .............................................. 158
Example 16: 3-D Flow and Chemical Transport with Complexation, Adsorption, Ion-exchange and Mineral Dissolution Reactions with Heat Transfer Consideration .......................................................... 173

7.0 REFERENCES ................................................................................................. 179

APPENDIX A: Data Input Guide for HYDROGEOCHEM 5.0 ................................. 185
APPENDIX B: Input and Output Files (in CD) .................................................... (disk)
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>A Hexahedral Element in Local Coordinates</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>A Surface Area and Its Embedded Local Coordinate</td>
<td>31</td>
</tr>
<tr>
<td>3.3</td>
<td>Weighting Factor along a Line Segment</td>
<td>43</td>
</tr>
<tr>
<td>3.4</td>
<td>Upstream Weighting Factors along 12 Sides of a Hexahedral Element</td>
<td>44</td>
</tr>
<tr>
<td>4.1</td>
<td>The Numerical Strategy for Steady State Simulations</td>
<td>61</td>
</tr>
<tr>
<td>4.2</td>
<td>The Numerical Strategy for Transient Simulations</td>
<td>63</td>
</tr>
<tr>
<td>4.3</td>
<td>The Strategy for Solving Fluid Flows</td>
<td>65</td>
</tr>
<tr>
<td>4.4</td>
<td>Program Structure of HYDROGEOCHEM 5.0: 1 of 5</td>
<td>67</td>
</tr>
<tr>
<td>4.4</td>
<td>Program Structure of HYDROGEOCHEM 5.0: 2 of 5</td>
<td>68</td>
</tr>
<tr>
<td>4.4</td>
<td>Program Structure of HYDROGEOCHEM 5.0: 3 of 5</td>
<td>69</td>
</tr>
<tr>
<td>4.4</td>
<td>Program Structure of HYDROGEOCHEM 5.0: 4 of 5</td>
<td>70</td>
</tr>
<tr>
<td>4.4</td>
<td>Program Structure of HYDROGEOCHEM 5.0: 5 of 5</td>
<td>71</td>
</tr>
<tr>
<td>6.1</td>
<td>Problem Definition (top) and Discretization (bottom) for Example 1</td>
<td>116</td>
</tr>
<tr>
<td>6.2</td>
<td>Contours of Pressure Head for Example 1</td>
<td>117</td>
</tr>
<tr>
<td>6.3</td>
<td>The Variations of Water Table at Various Time Steps for Example 2</td>
<td>118</td>
</tr>
<tr>
<td>6.4</td>
<td>The Problem Domain (top) and Discretization (bottom) for Example 4</td>
<td>119</td>
</tr>
<tr>
<td>6.5</td>
<td>Concentration Profiles at Various Time Steps for Example 4</td>
<td>120</td>
</tr>
<tr>
<td>6.6</td>
<td>The 0.5-Concentration Contours at Various Time Steps for Example 5</td>
<td>121</td>
</tr>
<tr>
<td>6.7</td>
<td>Problem Definition (top) and discretization (bottom) for Example 7</td>
<td>123</td>
</tr>
<tr>
<td>6.8</td>
<td>Concentration Contours for Example 7 (the steady Henry’s problem)</td>
<td>124</td>
</tr>
<tr>
<td>6.9</td>
<td>The 0.5-concentration Contour at Various Time Steps for Example 8 (the transient Henry’s problem)</td>
<td>124</td>
</tr>
<tr>
<td>6.10</td>
<td>Problem Definition (top) and Discretization (discretization) of Example 9</td>
<td>126</td>
</tr>
<tr>
<td>6.11</td>
<td>Pressure Head Contours (top) and Velocity Fields (bottom) for Example 9</td>
<td>129</td>
</tr>
<tr>
<td>6.12</td>
<td>Distribution of pH at Various Time Steps for Example 9</td>
<td>130</td>
</tr>
<tr>
<td>6.13</td>
<td>Concentration Contours of Precipitated Carbonate (Calcite) at Various Time Steps for Example 9</td>
<td>131</td>
</tr>
<tr>
<td>6.14</td>
<td>Concentration Contours of Precipitated Sulfate (Gypsum) at Various Time Steps for Example 9</td>
<td>132</td>
</tr>
<tr>
<td>6.15</td>
<td>Concentration Contours of Dissolved Uranium at Various Time Steps for Example 9</td>
<td>133</td>
</tr>
<tr>
<td>6.16</td>
<td>Problem Definition of Example 10: Transport Boundary Conditions, Regions of Different Material Types, and Information of the Adsorbing Site (SOH)</td>
<td>136</td>
</tr>
<tr>
<td>6.17</td>
<td>Steady Flow Results of Example 10: Pressure Head Contours (top) and Velocity Fields (bottom)</td>
<td>137</td>
</tr>
<tr>
<td>6.18</td>
<td>Concentration Contours of Dissolved Neptunium at Various Time Steps for Example 10</td>
<td>138</td>
</tr>
<tr>
<td>6.19</td>
<td>Distributions of the pH Values at Various Time Steps for Example 10</td>
<td>139</td>
</tr>
<tr>
<td>6.20</td>
<td>Variations of the Water Table at Various Times for Example 11</td>
<td>141</td>
</tr>
<tr>
<td>6.21</td>
<td>Concentration Contours of Dissolved Neptunium at Various Time Steps for Example 11</td>
<td>142</td>
</tr>
<tr>
<td>6.22</td>
<td>Distributions of the pH Values at Various Time Steps for Example 11</td>
<td>143</td>
</tr>
<tr>
<td>6.23</td>
<td>Variations of the Water Table at Various Times for Example 12</td>
<td>145</td>
</tr>
<tr>
<td>6.24</td>
<td>Concentration Contours of Dissolved Neptunium at Various Time Steps for Example 12</td>
<td>146</td>
</tr>
<tr>
<td>6.25</td>
<td>Distributions of the pH Values at Various Time Steps for Example 12</td>
<td>147</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Example Problems Simulated with HYDROGEOCHEM 5.0</td>
<td>114</td>
</tr>
<tr>
<td>6.2</td>
<td>Reaction Network for Example 9</td>
<td>127</td>
</tr>
<tr>
<td>6.3</td>
<td>Initial and Boundary Compositions of Recharge Water and Pore Water in the Tailings and Regions outside of the Tailing Pile</td>
<td>129</td>
</tr>
<tr>
<td>6.4</td>
<td>Reaction Network for Example 10</td>
<td>135</td>
</tr>
<tr>
<td>6.5a</td>
<td>Boundary Conditions for Steady Transport in Example 10</td>
<td>135</td>
</tr>
<tr>
<td>6.5b</td>
<td>Boundary Conditions for Transient Transport in Example 10</td>
<td>135</td>
</tr>
<tr>
<td>6.6</td>
<td>Boundary Conditions for Transient Transport in Example 11</td>
<td>141</td>
</tr>
<tr>
<td>6.7a</td>
<td>Boundary Conditions for Steady Transport in Example 12</td>
<td>145</td>
</tr>
<tr>
<td>6.7b</td>
<td>Boundary Conditions for Transient Transport in Example 12</td>
<td>145</td>
</tr>
<tr>
<td>6.8</td>
<td>Reaction Network for Example 13</td>
<td>151</td>
</tr>
<tr>
<td>6.9</td>
<td>Initial and Boundary Conditions for Example 13</td>
<td>151</td>
</tr>
<tr>
<td>6.10</td>
<td>$R^2$ Values Based on the Analytical and Numerical Solutions with Different Values of Dispersivity</td>
<td>156</td>
</tr>
<tr>
<td>6.11</td>
<td>List of Species Definitions for Example 15</td>
<td>161</td>
</tr>
<tr>
<td>6.12</td>
<td>Boundary Conditions for Chemical Transport in Example 15</td>
<td>162</td>
</tr>
<tr>
<td>6.13</td>
<td>Initial Conditions for Chemical Transport in Example 15</td>
<td>163</td>
</tr>
<tr>
<td>6.14</td>
<td>Reaction Network for Example 15</td>
<td>163</td>
</tr>
<tr>
<td>6.15</td>
<td>Matrix Decomposition for the Complex (multiple reaction types) Mixed (equilibrium and kinetic) Reaction System</td>
<td>165</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

This document is the users' guide for the Three-Dimensional Model of Coupled Fluid Flow, Thermal transport, and HYDROGEOCHEMical Transport through Variably Saturated Conditions: Version 5.0 (HYDROGEOCHEM 5.0). HYDROGEOCHEM 5.0 is a three-dimensional version of HYDROGEOCHEM 4.0 that is applicable to two-dimensional problems (Li, 2003). HYDROGEOCHEM 4.0 was evolved from HYDROGEOCHEM 3.2 that was created by extensively modifying HYDROGEOCHEM 3.1. HYDROGEOCHEM 3.1 was a moderate modification of HYDROGEOCHEM 3.0 or LEHGC 2.0 (Yeh et al., 1999) that was developed by combining a flow model with HYDROGEOCHEM 2.1 (Yeh, 1997). HYDROGEOCHEM 2.1 was written in FORTRAN 77 as a general purpose computer program to simulate reactive chemical transport problems subject to mixed geochemical kinetic and equilibrium reactions.

The evolution of HYDROGEOCHEM 2.1 to HYDROGEOCHEM 3.0 included the coupling between fluid flows and reactive chemical transport to deal with the change of porosity, hydraulic conductivity, and diffusion/dispersion due to chemical precipitation or due to the change of minerals resulting from chemical reactions. The evolution from HYDROGEOCHEM 3.0 to HYDROGEOCHEM 3.1 involved moderate changes in which the effects of the removal of water due to precipitation reactions were considered.

The modification of HYDROGEOCHEM 3.1 resulting in HYDROGEOCHEM 3.2 involved a major change in the biogeochemical module. A new paradigm of reaction-based approaches was employed to describe biogeochemical processes in terms of reaction networks. To make the biogeochemical module quite general, any kinetic reaction can be formulated with elementary rate laws, n-th order equations, Monod kinetics or user specified rate equations. Similarly, for any fast reaction, its governing equation is either given with the mass action formulation or specified by the user with an algebraic equation.

The evolution of HYDROGEOCHEM 3.2 to HYDROGEOCHEM 4.0 incorporated heat transfer, which allows the investigation on the effect of thermal transport on both fluid flow and reactive biochemical transport. HYDROGEOCHEM 4.0 is the latest version of two-dimensional reactive chemical transport under variably saturated flow conditions in porous media. The modification of HYDROGEOCHEM 4.0 resulting in HYDROGEOCHEM 5.0 updated the model to three-dimensions. The development history of various versions of HYDROGEOCHEM and/or LEHGC is described below:

(1) HYDROGEOCHEM (Yeh and Tripathi, 1990): HYDROGEOCHEM was the first comprehensive simulator of HYDROlogic Transport and GEOCHEMICAL Reaction in Saturated-Unsaturated Media. It iteratively solves the two-dimensional transport and geochemical equilibrium equations.

(2) LEHGC 1.0 (Yeh et al., 1995): The computer program LEHGC is a Hybrid Lagrangian-Eulerian Finite Element Model of HydroGeoChemical (LEHGC) transport through Saturated-Unsaturated Media. It is a decedent of HYDROGEOCHEM, a strictly Eulerian finite element reactive transport code. The hybrid Lagrangian-Eulerian scheme improves on the Eulerian scheme by allowing larger time steps to be used in advection-dominant transport calculations, and also is more computationally efficient.

(3) LEHGC 1.1 (Yeh et al., 1995): LEHGC version 1.1 is a modification of LEHGC Version 1.0. The modification included: (1) devising a tracking algorithm with the computational effort proportional to N where N is the number of computational grid nodes rather than N^2 as in the LEHGC Version 1.0, (2) including multiple adsorbing sites and multiple ion-exchange sites, (3) using four preconditioned conjugate gradient methods for the solution of matrix equations, and (4) providing
the capacity of colloid transport.

(4) HYDROGEOCHEM 2.0 (Yeh and Salvage, 1997): HYDROGEOCHEM 2.0 expanded the scope of LEHGC1.1 with mixed equilibrium and kinetic geochemical reactions and provided options in conventional finite element and hybrid Lagrangian finite element methods.

(5) HYDROGEOCHEM 2.1 (Yeh and Salvage, 1997): HYDROGEOCHEM 2.1 is a slight modification of HYDROGEOCHEM 2.0. Version 2.1 incorporated a more flexible approach for simulating kinetically controlled reactions. For Version 2.1, the contribution to the residual and Jacobian due to non-basic (non-canonical) kinetic reactions was done by chemical reactions rather than by chemical species done in Version 2.0. This allows for more efficient handling of non-basic reactions.

(6) LEHGC 2.0 (Yeh et al., 1999, 2001): LEHGC 2.0 was developed as a simulator for coupled density dependent fluid flow and reactive geochemical transport. The coupling was achieved by combining the density dependent fluid flow and solute transport model, 2DFEMFAT (Yeh et al., 1993), with the reactive chemical transport model, HYDROGEOCHEM 2.1. It iteratively solved the two-dimensional fluid flow and reactive chemical transport equations. The Richards' equation was solved for fluid flow and the advection-dispersion-reactive transport equations were solved for all chemical components and for kinetically controlled aqueous complexed species. Although this version was initiated at Penn State University, it was successfully completed by Dr. Ming-Hsu Li working with Dr. Malcolm D. Siegel at Sandia National Laboratory.

(7) HYDROGEOCHEM 3.0 (Personal Communication): HYDROGEOCHEM 3.0 was a very slight update of LEGHC 2.0. The update involved mainly the cleanup of the code. In keeping up with the tradition and spirit of HYDROGEOCHEM, the cleaned version was renamed HYDROGEOCHEM 3.0.

(8) HYDROGEOCHEM 3.1 (Personal Communication): HYDROGEOCHEM 3.1 was a moderate modification of HYDROGEOCHEM 3.0. The modification involved the removal of water for the precipitation reactions. This allows simulations of genesis of geologic formations due to dewatering processes.

(9) HYDROGEOCHEM 3.2 (Personal Communication): HYDROGEOCHEM 3.2 is a modified version of HYDROGEOCHEM 3.1. The major modification was the formulation of reactive chemistry, in which a new paradigm of reaction-based approaches to biogeochemical processes (Fang et al., 2003; Yeh 2003a, 2003b) was implemented.

(10) HYDROGEOCHEM 4.0 (Li, 2003): HYDROGEOCHEM 4.0 was the most recent version of the code in two dimensions, which incorporated heat transfer that allows one to investigate the effect of thermal transport on reactive geochemical and biochemical transport. It was developed based on HYDROGEOCHEM 3.2.

(11) HYDROGEOCHEM 5.0 (Sun, 2004): HYDROGEOCHEM 5.0 was the most recent version of the code in three dimensions, which incorporated heat transfer that allows one to investigate the effect of thermal transport on reactive geochemical and biochemical transport. It was developed based on HYDROGEOCHEM 4.0.

This document is intended to assist modelers using HYDROGEOCHEM 5.0 for site-specific applications.
Chapter 2 gives the governing equations and initial and boundary conditions solved by HYDROGEOCHEM 5.0.

Chapter 3 details the numerical method used in solving the equations for flow, thermal transport, and reactive geochemical and biochemical transport.

Chapter 4 describes the program structure of HYDROGEOCHEM 5.0. It is intended to assist users in modifying or extending the code to better meet their needs.

Chapter 5 describes the program's storage and memory requirements and relates it via the parameter statements to the problem size (the finite-element grid size as well as the number of chemical components and species). The chapter also describes the input-output format and specifications of HYDROGEOCHEM 5.0.

Chapter 6 provides sample problems demonstrating the use of the program. It intends to serve as templates in preparing input for their practical applications by users.

Appendix A contains a complete description of all the problem-specific input necessary for running a calculation.

The program must be run with a consistent set of units. Units of mass (M), length (L), temperature (K), and time (T) are indicated in the input description. In this version, the unit of length should be decimeter (dm), and the unit of mass for any chemical species should be mole. The density of water and solid should be expressed in kg/dm³ (liter). The ion-exchange capacity is in equivalents/mass of solid. The corresponding concentration unit of all species (aqueous, sorbed, and precipitated species) is mole/liter of fluid (Molar); the corresponding unit for the sorption distribution coefficient is dm³/kg (= ml/g). The unit of temperature is expressed in Kelvin (K). Any units of time may be used as long as the same unit is used throughout the input file.

HYDROGEOCHEM 5.0 is a flexible and comprehensive package. It is designed to:

- include variably saturated fluid flow, heat transfer, and reactive chemical transport;
- incorporate the effect of precipitation/dissolution on the change of pore sizes, hydraulic conductivity, and diffusion/dispersion;
- deal with equilibrium and kinetic reactions in terms of reaction networks;
- treat heterogeneous and anisotropic media;
- consider spatially and temporally distributed sources/sinks as well as point sources/sinks;
- accept the prescribed initial conditions or obtain initial conditions by simulating the steady-state version of the system under consideration;
- deal with prescribed transient state-variables (pressure head and/or temperature and/or chemical concentrations) distributed over a Dirichlet boundary;
- handle time-dependent fluxes over variable boundaries;
- include the off-diagonal tensor coefficients in the governing equation for dealing with cases in which the coordinate system does not coincide with the principal directions of the tensor;
- provide two options for treating the mass matrix (consistent and lumping);
- give three options for estimating the nonlinear matrix (exact relaxation, under-relaxation, and over-relaxation);
- include six options for solving the linearized matrix equations (block iteration method, successive point iterations, and four preconditioned conjugate gradient methods);
• include hexahedral elements, triangular prisms, and tetrahedral elements to facilitate the discretization of real-world problems;
• reset automatically the time-step size when boundary conditions or source/sinks change abruptly;
• provide three options for numerical approximations for reactive geochemical and biochemical transport and heat transfer equations: conventional finite element methods (FEM), hybrid Lagrangian-Eulerian FEM, or hybrid Lagrangian-Eulerian FEM for interior nodes plus FEM for boundary nodes;
• include (i) fully implicit iteration approach, (ii) operator splitting approach, and (iii) predictor-corrector approach to process the coupling between hydrologic transport and biogeochemical reactions;
• include chemical processes of aqueous complexation, precipitation-dissolution, adsorption-desorption, ion-exchange, redox, and acid-base reactions;
• handle multiple adsorption sites and multiple ion-exchange site;
• consider mixed equilibrium and kinetic reactions; and
• include colloid transport.

Limitations and Recommendations

There are some limitations in HYDROGEOCHEM 5.0. These include (1) applications are limited to single-fluid phase flows and (2) dual-porosity media cannot be effectively dealt with. Further modifications of HYDROGEOCHEM 5.0 to relax these limitations in its design capabilities are needed to improve the model flexibility for practical applications.
2. MATHEMATICAL FORMULATION

A complete mathematical model that describes fluid flow, reactive biogeochemical transport, and heat transfer in variably saturated subsurface media includes governing equations, initial and boundary conditions, and constitutive relationships. This chapter provides a heuristic derivation of the mathematical model. Detailed derivations can be found elsewhere (Yeh, 1999, 2000).

2.1. Flow Equations

The hydrologic variables, including the Darcy's velocity and moisture content, are necessary factors in determining the transport of solutes and heat transfer through saturated-unsaturated subsurface systems. These hydrological variables can be specified by the user as described in the Data Input Guide (Data Set 27 in Appendix A) or can be solved in HYDROGEOCHEM 5.0 The basic governing equation of the saturated-unsaturated flow problems is briefly stated below.

2.1.1. Governing Equations

The governing equations for flow through saturated-unsaturated media can be derived based on (1) continuity of fluid, (2) continuity of solid, (3) motion of fluid (Darcy's law), (4) consolidation of the media, and (5) compressibility of water (Yeh et al., 1994a, 1994b):

\[
\frac{\rho \cdot F \cdot \frac{\partial h}{\partial t}}{\rho_o} = \nabla \cdot [K \cdot (\nabla h + \frac{\rho}{\rho_o} \nabla z)] + \frac{\rho^*}{\rho_o} q
\]  

(2.1.1)

in which \( F \) is the generalized storage coefficient \((1/L)\) defined as

\[
F = \alpha' \frac{\theta}{n_e} + \beta' \theta + n_e \frac{dS}{dh},
\]  

(2.1.2)

\( K \) is the hydraulic conductivity tensor \((L/T)\) defined as

\[
K = \frac{\rho_o \cdot k}{\mu} = \frac{(\rho/\rho_o) \cdot \rho_o \cdot \frac{g}{\mu_o} \cdot k_r \cdot k_{r'}} = \frac{(\rho/\rho_o) \cdot \rho_o \cdot \frac{g}{\mu_o} \cdot K_{so} \cdot k_r},
\]  

(2.1.3)

and \( V \) is Darcy's velocity \((L/T)\) given by

\[
V = -K \cdot \frac{\rho_o \cdot \nabla h + \nabla z}{\rho}
\]  

(2.1.4)

where \( \theta \) is the effective moisture content \((L^3/L^3)\), \( h \) is the pressure head \((L)\), \( t \) is the time \((T)\), \( z \) is the potential head \((L)\), \( q \) is the source or sink representing the artificial injection or withdrawal of fluid \([(L^3/L^3)/T]\), \( \rho_o \) is the referenced fluid density at zero biogeochemical concentration \((M/L^3)\), \( \rho \) is the fluid density with dissolved biogeochemical concentrations \((M/L^3)\), \( \rho^* \) is the fluid density of either injection \((= \rho^*)\) or withdraw \((= \rho)\)
(M/L^3), \( \mu_0 \) is the fluid dynamic viscosity at zero biogeochemical concentration [(M/L)/T], \( \mu \) is the fluid dynamic viscosity with dissolved biogeochemical concentrations [(M/L)/T], \( \alpha' \) is the modified compressibility of the soil matrix (1/L), \( \beta' \) is the modified compressibility of the liquid (1/L), \( n_e \) is the effective porosity (L^3/L^3), \( S \) is the degree of effective saturation of water, \( g \) is the gravity (L/T^2), \( k \) is the permeability tensor (L^2), \( k_s \) is the saturated permeability tensor (L^2), \( K_{so} \) is the referenced saturated hydraulic conductivity tensor (L/T), \( k_r \) is the relative permeability or relative hydraulic conductivity (dimensionless).

Equations (2.1.1) through (2.1.4) and the constitutive relationships among the pressure head, degree of saturation, and hydraulic conductivity tensor, together with appropriate initial conditions and boundary conditions, are used to simulate the temporal-spatial distributions of the hydrological variables, including pressure head, total head, effective moisture content, and Darcy's velocity.

2.1.2. Initial and Boundary Conditions for Flow Simulations

To complete the mathematical formulation of saturated-unsaturated problems, Eq. (2.1.1) must be supplemented with initial and boundary conditions. The initial condition is stated mathematically as

\[
h = h_i(x,y,z) \quad \text{in } \quad R
\]  

where \( R \) is the region of interest and \( h_i \) is the prescribed initial condition, which can be obtained by either field measurements or by solving the steady state version of Eq. (2.1.1) with time-invariant boundary conditions.

Three basic types of boundary conditions and a river boundary condition can be specified for variably saturated flows. In addition, a variable boundary condition normally at the air-media interface can be specified. These general types of boundary conditions are considered in HYDROGEOCHEM 5.0 to handle a variety of physical phenomena that can happen on the boundary. These boundary conditions are stated mathematically as follows:

**Dirichlet Boundary Conditions.** On a Dirichlet boundary, the hydraulic head is prescribed as

\[
h + z = H_d(x_b,y_b,z_b,t) \quad \text{on } \quad B_d
\]  

where \((x_b,y_b,z_b)\) is the spatial coordinate on the boundary, \( H_d \) is the prescribed functional value of the hydraulic head, and \( B_d \) is the Dirichlet boundary. 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.

**Neumann Boundary Conditions.** On a Neumann boundary, which normally is the drainage boundary, the gradient flux is specified as

\[
-n \cdot \mathbf{K} \frac{\rho_o}{\rho} \nabla h = q_n(x_b,y_b,z_b,t) \quad \text{on } \quad B_n
\]  

where \( \mathbf{n} \) is an outward unit vector normal to the boundary, \( q_n \) is the prescribed Neumann flux, and \( B_n \) is the Neumann boundary. On the drainage boundaries, for example, the flux is due to gravity and it can be considered as a Neumann boundary with a zero Neumann flux.
**Cauchy Boundary Conditions.** On a Cauchy boundary, which normally is an infiltration boundary, the volume flux is prescribed as

\[-n \cdot K \left( \frac{\rho}{\rho} \nabla h + \nabla z \right) = q_c(x_b, y_b, z_b, t) \quad \text{on} \quad B_c \quad (2.1.8)\]

where \(q_c\) is the prescribed Cauchy flux and \(B_c\) is the Cauchy boundary. For example, the infiltration rate and ground surface can be considered as a Cauchy flux and a Cauchy boundary, respectively, if the infiltration rate is measured.

**Variable Boundary Conditions - During Precipitation Period.** On a variable boundary, which is normally the air-media interface, either infiltration, ponding, or seepage can occur during precipitation periods. If infiltration occurs, the maximum amount of infiltration rate is the excess precipitation rate. As to which condition is prevalent, it cannot be determined as \textit{a priori}. Rather it must be determined in a cyclic iterative procedure (Yeh, 1987b). These physical conditions lead to the mathematical representations

\[h = h_p(x_b, y_b, z_b, t) \quad \text{iff} \quad -n \cdot K \left( \frac{\rho}{\rho} \nabla h + \nabla z \right) \geq q_p \quad \text{on} \quad B_v \quad (2.1.9)\]

or

\[-n \cdot K \left( \frac{\rho}{\rho} \nabla h + \nabla z \right) = q_p(x_b, y_b, z_b, t) \quad \text{iff} \quad h \leq h_p \quad \text{on} \quad B_v \quad (2.1.10)\]

where \(B_v\) is the variable boundary, \(h_p\) is the allowed ponding depth, and \(q_p\) (numerically negative) is the excess precipitation. Either Eq. (2.1.9) or (2.1.10), but not both, is used at any point on the variable boundary at any time during precipitation periods.

**Variable Conditions - During Nonprecipitation Period.** During nonprecipitation periods, either seepage or evaporation can occur on a variable boundary. If evaporation occurs, the maximum amount of evaporation is the potential evaporation. These physical considerations lead to the following mathematical statements:

\[h = h_p(x_b, y_b, z_b, t) \quad \text{iff} \quad -n \cdot K \left( \frac{\rho}{\rho} \nabla h + \nabla z \right) \geq q_e \quad \text{on} \quad B_v \quad (2.1.11)\]

or

\[h = h_m(x_b, y_b, z_b, t) \quad \text{iff} \quad -n \cdot K \cdot \left( \frac{\rho}{\rho} \nabla h + \nabla z \right) \leq q_e(x_b, y_b, z_b, t) \quad \text{on} \quad B_v \quad (2.1.12)\]

or

\[-n \cdot K \cdot \left( \frac{\rho}{\rho} \nabla h + \nabla z \right) = q_e(x_b, y_b, z_b, t) \quad \text{iff} \quad h \geq h_m \quad \text{on} \quad B_v \quad (2.1.13)\]

where \(h_m\) is the allowed minimum pressure on the variable boundary and \(q_e\) is the allowed maximum evaporation rate on the variable boundary, which is the potential evaporation. Only one of Eqs. (2.1.11) through (2.1.13) is used at any point on the variable boundary at any time during non-precipitation periods.

**River Boundary Conditions.** At surface water-media interfaces, for example, the river-media interfaces, two types of boundary conditions can be specified depending on physical conditions. If there are sediment layers around the wet perimeter of the surface water-media interfaces, then a river boundary condition of radiation
types can be imposed:

$$-n \cdot \mathbf{K} \left( \frac{\rho_0 \nabla h}{\rho} + \nabla z \right) = -\frac{K_R}{b_R} (h_R - h) \quad \text{on} \quad B_r \quad (2.1.14)$$

where $K_R$ is the hydraulic conductivity of the river bottom sediment layer, $b_R$ is the thickness of the river bottom sediment layer, and $h_R$ is the depth of the river bottom measured from the river surface to the top of the sediment layer, and $B_r$ is the segments of river-media interfaces. If sediment layers are not present at the river-media interface (i.e., the surface water is in direct connection with the media), then the Dirichlet boundary conditions described above should be imposed.

### 2.1.3. Dependence of Fluid Density and Viscosity on Biogeochemical Concentrations

The density of groundwater is a function of biogeochemical concentration and can be derived as follows (Cheng, 1995)

$$\rho = \rho_w + \sum_{i=1}^{M_i} c_i m_i \left( 1 - \frac{\rho_w}{\rho_i} \right) \quad (2.1.15)$$

where $\rho_w$ is the density of water, which is a function of temperature; $M_i$ is the number of mobile species or aqueous phase species; $c_i, m_i,$ and $\rho_i$ are the concentration, molecular weight, and intrinsic density of the $i$-th dissolved species, respectively. In order to make the computation easy, 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 this assumption verified, the following equation can be used to compute the groundwater density (Cheng, 1995):

$$\rho = \rho_w + \sum_{j=1}^{N_j} C_j M_j \left( 1 - \frac{\rho_w}{\rho_j} \right) \quad (2.1.16)$$

where $N_j$ is the number of aqueous components or mobile components; $C_j, M_j,$ and $\rho_j$ 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. (2.1.16) can be further simplified to:

$$\rho = \rho_w + \sum_{j=1}^{N_j} C_j M_j \quad (2.1.17)$$

The dynamic viscosity of groundwater is assumed to take the following form, which is similar to Eq. (2.1.17):

$$\mu = \mu_w + \sum_{i=1}^{N_i} a_i C_i M_i \quad (2.1.18)$$

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. (2.1.16) is a simplification of Eq. (2.1.15) which can be derived theoretically, whereas Eq. (2.1.18) is an assumed equation to describe the reality. Thus, Eq. (2.1.18) can be replaced with any other empirical formula to match the situation being considered. The modification in the computer code is straightforward. In HYDROGEOCHEM 5.0, Eq. (2.1.17) is used to describe 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, an empirical formula can be used to characterize the groundwater density.

### 2.1.4. Effect of Precipitation/Dissolution on Hydraulic Conductivity and Porosity

Let \( \theta_{so} \) be the effective saturated moisture content when solid or surface biogeochemical species are not present; \( \theta_s \) the effective saturated moisture content with considering the precipitation effect; \( \theta_r \) the residual moisture content; \( \theta_a \) the volume fraction of air; \( \theta_m \) the volume fraction of solid media excluding solid or surface biogeochemical species; \( S \) the degree of saturation of water; \( \phi_p \) equal to \( \sum P_i V_i \), where \( P_i \) is the precipitated concentration of the i-th mineral \([\text{mole/dm}^3 \text{ of water}]\), \( V_i \) is the mole volume of the i-th mineral \([\text{dm}^3 \text{ of solid/mole}]\), and i stands for the i-th precipitated mineral; and \( h \) the pressure head.

By definition, we have the following identities in a system of reactive biogeochemical transport under saturated-unsaturated flows.

\[
\theta + \theta_a + \theta \phi_p = 1 - \theta_r - \theta_m = \theta_{so} \tag{2.1.19}
\]

\[
\theta = S \theta_s \tag{2.1.20}
\]

\[
\theta_a = (1-S) \theta_s \tag{2.1.21}
\]

Substitution of Eqs. (2.1.20) and (2.1.21) into Eq. (2.1.19) yields

\[
\theta_s + \theta \phi_p = \theta_{so} \tag{2.1.22}
\]

In the following discussion, \( \theta_{so} \) is assumed to be constant with time.

The volume fraction of solid or surface biogeochemical species due to the precipitation is given by

\[
\phi_s = \theta \phi_p \tag{2.1.23}
\]

because we have defined the concentration of solid species as the biogeochemical mass per unit volume of water. Thus

\[
\theta_s = \theta_{so} - \phi_s = \theta_{so} - \theta \phi_p \tag{2.1.24}
\]

Substitution of Eq. (2.1.20) into Eq. (2.1.24) yields

\[
\theta_s = \theta_{so} - S \theta_s \phi_p \quad \therefore \quad \theta_s = \frac{\theta_{so}}{1 + S \phi_p} \tag{2.1.25}
\]

Equation (2.1.25) shows that the effective saturated moisture content is explicitly expressed as a function of both the pressure head (because \( S \) is a function of pressure head) and the concentrations of biogeochemical species. Using Eq. (2.1.25), we can now express the effective moisture content explicitly in terms of the
pressure head (via the degree of saturation $S$) and the concentrations of biogeochemical species

$$\theta = \frac{S \theta_{so}}{1 + S \phi_p}$$  \hspace{1cm} (2.1.26)

Since $K_s \propto \phi^n$, where $\phi$ is the pore volume fraction, $n$ is the fractional exponent for estimating $K_s$ based on particle size and packing structure. Let $K_{so}$ be the saturated hydraulic conductivity without the presence of solid and surface biogeochemical species. We have

$$\frac{K_s}{K_{so}} = \left(\frac{\theta_s}{\theta_{so}}\right)^n$$  \hspace{1cm} (2.1.27)

When the precipitation effect is not included (i.e., when $\phi_p = 0$), $\theta_s = \theta_{so}$ (i.e., $K_s = K_{so}$). Experimental evidence shows that

$$S = f(h), \quad k_r = g(h)$$  \hspace{1cm} (2.1.28)

where $k_r$ is the relative hydraulic conductivity. Under saturated flow conditions, $S = 1$ and $k_r = 1$. Under variably saturated flow conditions, $S$ and $k_r$ are determined from the simulation of Richards' equation governing saturated-unsaturated flows. Since $K = K_s k_r$, and $k_r$ is a function of only the pressure head, the hydraulic conductivity in a saturated-unsaturated flow system is computed as

$$K = K_{so} k_r \left(\frac{\theta_s}{\theta_{so}}\right)^n = K_{so} k_r \left(\frac{1 + S \phi_p}{\theta_{so}}\right)^n = K_{so} k_r \left(\frac{1}{1 + S \phi_p}\right)^n$$  \hspace{1cm} (2.1.29)

The water capacity defined by $\Psi = \theta_s dS/dh$ is a function of pressure and precipitation/dissolution. From the above discussions, it is clear that the hydraulic conductivity, the moisture content, and the water capacity are all depending on the degree of saturation and concentrations of solid and/or surface biogeochemical species.

### 2.2. Reactive Biogeochemical Transport Equations

The governing equations for a reactive system can be derived based on the principles of mass balance and biogeochemical reactions (Yeh, 2000). The governing equations for transport are derived based on the continuity of mass and Fick's flux laws. The major transport processes are advection, dispersion/diffusion, source/sink, and biogeochemical reactions (including radioactive decay).

#### 2.2.1. Species Transport Equations

The general transport equation governing the temporal-spatial distribution of any biogeochemical species in a reactive system is derived below. Let $C_i$ be the concentration of the $i$-th species, the governing equation for $C_i$ can be obtained by applying the principle of mass balance in integral form as follows:
\[
\frac{D}{Dt} \int_v \theta C_i \, dv = -\int_\Gamma \mathbf{n} \cdot (\theta C_i) \mathbf{V}_i \, d\Gamma - \int \mathbf{n} \cdot \mathbf{J}_i \, d\Gamma + \int_\Gamma \theta r_i \, dv + \int_\varepsilon \mathbf{M}_i \, dv, \quad i \in M
\]  

(2.2.1)

where \(v\) is the material volume containing constant amount of media (L\(^3\)), \(C_i\) is the concentration of the \(i\)-th species in molar per unit fluid volume (M/L\(^3\)), \(\Gamma\) is the surface enclosing the material volume \(v\) (L\(^2\)), \(\mathbf{n}\) is the outward unit vector normal to the surface \(\Gamma\), \(\mathbf{J}_i\) is the surface flux of the \(i\)-th species due to dispersion and diffusion with respect to relative fluid velocity [(M/T)/L\(^2\)], \(\theta r_i\) is the production rate of the \(i\)-th species per unit medium volume due to all biogeochemical reactions [(M/L\(^3\))/T], \(\mathbf{M}_i\) is the external source/sink rate of the \(i\)-th species per unit medium volume[(M/L\(^3\))/T], \(M\) is the number of biogeochemical species, and \(\mathbf{V}_i\) is the transporting velocity relative to the solid of the \(i\)-th biogeochemical species (L/T).

By the Reynolds transport theorem (Owczarek, 1964), Eq. (2.2.1) can be written as

\[
\int_v \frac{\partial \theta C_i}{\partial t} \, dv + \int \mathbf{n} \cdot \theta C_i (\mathbf{V}_i + \mathbf{V}_s) \, d\Gamma + \int_\Gamma \mathbf{n} \cdot \mathbf{J}_i \, d\Gamma = \int_\varepsilon \theta r_i \, dv + \int_\varepsilon \mathbf{M}_i \, dv, \quad i \in M
\]  

(2.2.2)

where \(\mathbf{V}_s\) is the velocity of the solid matrix (L/T). Assuming that the relative velocity for all species are the same, then \(\mathbf{V}_s, \mathbf{V}_l\) the fluid velocity \(\mathbf{V}_f\) and the Darcy's velocity \(\mathbf{V}\) are related to each another by

\[
\mathbf{V}_f = \mathbf{V}_i + \mathbf{V}_s, \, \mathbf{V} = \theta \mathbf{V}_i = \theta (\mathbf{V}_f - \mathbf{V}_s)
\]  

(2.2.3)

Applying the Gaussian divergence theorem to Eq. (2.2.2), using the fact that \(v\) is arbitrary, and substituting with Eq. (2.2.3), one can obtain the following continuity equation for the \(i\)-th species:

\[
\frac{\partial \theta C_i}{\partial t} + \nabla \cdot (\theta C_i \mathbf{V}_f) + \nabla \cdot \mathbf{J}_j = \theta r_j + \mathbf{M}_i, \quad i \in M.
\]  

(2.2.4)

Equation (2.2.4) is simply the statement of mass balance over a differential volume. The first term represents the rate of mass accumulation, the second term represents the net rate of mass flux due to advection, the third term is the net mass flux due to dispersion and diffusion, the fourth term is the rate of mass production and reduction due to biogeochemical reactions and radioactive decay, and the last term is source/sink term corresponding to artificial injection and/or withdrawal.

The surface flux \(\mathbf{J}_j\) has been postulated to be proportional to the gradient of \(C_i\) (Nguyen et al., 1982)

\[
\mathbf{J}_i = -\theta \mathbf{D} \cdot \nabla C_i, \quad i \in M
\]  

(2.2.5)

where \(\mathbf{D}\) is the hydrodynamic dispersion coefficient tensor (L\(^2\)/T) (Bear, 1972). According to Millington and Quirk (1961) (From Gerke et al., 1998), \(\mathbf{D}\) can be given in the standard form as

\[
\theta \mathbf{D} = a_T \mathbf{V} \delta + (a_L - a_T) \mathbf{VV}/|\mathbf{V}| + a_m \theta \tau \delta
\]  

(2.2.6)

where \(a_T\) is the transverse diffusivity (L), \(\delta\) is the Kronecker delta tensor, \(|\mathbf{V}|\) is the magnitude of the Darcy's velocity \(\mathbf{V}\) (L/T), \(a_L\) is the longitudinal diffusivity (L), \(a_m\) is the molecular diffusion coefficient (L\(^2\)/T), and \(\tau\) is the tortuosity.
Substituting Eqs. (2.2.3) and (2.2.5) into (2.2.4) and simplify the notation for convenience of derivation, we obtain

\[ \frac{\partial \theta C_i}{\partial t} + \nabla \cdot (\theta C_i V_s) = L(C) + \theta r_j + M_i, \quad i \in M \tag{2.2.7} \]

where L is the transport operator denoting

\[ L(C) = -\nabla \cdot (VC) - \nabla \cdot J = -\nabla \cdot (VC) - \nabla \cdot (\theta D \nabla C) \tag{2.2.8} \]

The second term on the left-hand side of Eq. (2.2.7) can be expressed as

\[ \nabla \cdot (\theta C_i V_s) = \nabla (\theta C_i) \cdot V_s + (\theta C_i) \nabla \cdot V_s \tag{2.2.9} \]

The first term on the right-hand side of Eq. (2.2.9) is the product of two small vectors and will be neglected.

If all the displacement of medium is assumed to be vertical (e.g., vertical consolidation), the solid velocity becomes (Yeh, 1987a)

\[ \nabla \cdot V_s = \alpha \frac{\partial p}{\partial t} = \alpha \frac{\partial \rho g h}{\partial t} = \alpha \frac{\partial h}{\partial t} \tag{2.2.10} \]

where p is the total pressure, \( \alpha \) is the compressibility of the media, and \( \alpha' \) is the modified compressibility of the media.

Substitute Eqs. (2.2.9) and (2.2.10) into Eq. (2.2.7) yields

\[ \frac{\partial \theta C_i}{\partial t} + \alpha' \frac{\partial h}{\partial t} C_i = L(C) + \theta r_j + M_i, \quad i \in M \tag{2.2.11} \]

Equation (2.2.11) is written in conservative form. It has been suggested that using the advective form is sometimes more appropriate, especially if the finite element method is used to simulate the biogeochemical transport equation (Huyakorn et al., 1985). More importantly, an advective form of transport equations allows the use the mixed Lagrangian-Eulerian approach, which can better solve advection-dominant transport problems (Yeh and Tripathi, 1987, 1989). An advective form of the transport equation is derived by expanding the advection term and using the continuity equation for water flow, which is:

\[ \frac{\rho F \partial h}{\rho_0} \frac{\partial t}{\partial t} = -\nabla \left( \frac{\rho F V}{\rho_0} \right) + \frac{\rho^*}{\rho_0} q \tag{2.2.12} \]

which is conservation of fluid mass. Performing the necessary manipulation, we obtain
\[
\frac{\partial C_i}{\partial t} + \left( \theta \frac{\partial h}{\partial t} + \frac{\partial \theta}{\partial t} \right) C_i + \mathbf{V} \cdot \nabla C_i - \nabla \cdot (\theta \mathbf{D} \cdot \nabla C_i) = \theta r_i + M_i - QC_i, \quad i \in M
\]

and

\[
Q = \frac{\rho^*}{\rho} q - \frac{\rho}{\rho} \frac{\partial h}{\partial t} - \frac{\rho}{\rho} \mathbf{V} \cdot \nabla \rho \quad \text{or} \quad Q = \frac{\rho^*}{\rho} q - \frac{\rho}{\rho} \frac{\partial h}{\partial t} - \frac{\rho}{\rho} \mathbf{V} \cdot \nabla \rho
\]

For simplicity of notation and convenience of derivation, let us rewrite Eq. (2.2.13) in the following form:

\[
\frac{\partial C_i}{\partial t} + \left( \theta \frac{\partial h}{\partial t} + \frac{\partial \theta}{\partial t} \right) C_i = \mathscr{L}(C_i) + \theta r_i + M_i - QC_i, \quad i \in M
\]

in which \(\mathscr{L}\) is the advection-dispersion operator denoting

\[
\mathscr{L}(C_i) = -\mathbf{V} \cdot \nabla C_i + \nabla \cdot (\theta \mathbf{D} \cdot \nabla C_i)
\]

### 2.2.2. Generalized Reactive Biogeochemical Transport in Variably Saturated Media

Let us rewrite Eq. (2.2.14) as:

\[
\frac{\partial C_i}{\partial t} + \left( \theta \frac{\partial h}{\partial t} + \frac{\partial \theta}{\partial t} \right) C_i = \mathscr{L}(C_i) + M_i + \theta r_i|_N, \quad i \in M
\]

where \(r_i|_N\) is the rate of production of the i-th species due to N reactions.

Equation (2.2.16) is a statement of material balance of any species i in the system. It states that the rate of change of mass [all terms on the left-hand side of Eq. (2.1.16)] of any species in a bulk volume is due to hydrologic transport [the two terms in the parenthesis on the right-hand side of Eq. (2.2.16)], artificial source/sink, and the production/consumption due to all biogeochemical reactions.

The formulation of \(r_i|_N\) and associated parameters is the central challenge in biogeochemical modeling. Ad hoc and reaction-based formulations are two general means of formulating \(r_i|_N\). In an ad hoc formulation, the production-degradation rate is obtained with empirical functions as:

\[
r_i|_N = f_i(C_1, C_2, \ldots, C_M; p_1, p_2, \ldots)
\]

where \(f_i\) is the empirical function for the i-th species and \(p_1, p_2, \ldots\) are rate parameters used to fit the experimental data. This formulation is purely empirical and does not consider the contribution of every individual reaction; thus it will result in an inadequate parametrization.

Based on the above consideration, we should search for a more reliable approach applicable to diverse geochemical and biochemical conditions. In a reaction-based formulation, the rates of change of M biogeochemical species are described by:

\[
r_i|_N = \sum_{k=1}^{N} (v_{ik} - \mu_{ik}) R_k \quad i \in M; \quad U \theta \left( \frac{dC}{dt} \right)_{\text{reaction}} = \theta v R
\]
where $\nu_{ik}$ and $\mu_{ik}$ are the reaction stoichiometry of the $i$-th species in the $k$-th reaction associated with the products and reactants, respectively; $R_k$ is the rate of the $k$-th reaction; $U$ is a unit matrix; $C$ is a vector with its components representing $M$ species concentrations; $v$ is the reaction stoichiometry matrix with $-\mu_{ik}$ and $\nu_{ik}$ as its components; and $R$ is the reaction rate vector with $N$ reaction rates as its components.

Combined with Eq. (2.2.18), Eq. (2.2.16) can be written in the matrix form as:

$$A\left(\frac{\partial C}{\partial t} + \alpha \frac{\partial h}{\partial t} C + \frac{\partial}{\partial t} C - M\right) = B\left(f(C) - QC\right) + \theta v R$$

where $A$ and $B$ are unit matrices to be decomposed via Gaussian-Jordan elimination.

### 2.2.3. Decomposition of the Reactive Transport Equation

An analytical solution of Eq. (2.2.18) is generally impossible. Numerical integration is an attractive way to solve it. Numerical integration of Eq. (2.2.18) in its primitive form will encounter much difficulty. Since the reaction rates of $N$ reactions can, in general, range over several orders of magnitude. The time-step size used in numerical integration is dictated by the largest reaction rate among $N$ reactions. If at least one of the reactions is infinitely large, the time-step size must be infinitely small, which makes integration impractical. Thus, if more than one equilibrium reactions is involved in any of the $M$ equations in Eq. (2.2.18), there is no way to define the summation (or the difference) of infinity.

In reality, all rates are finite, although they may be quite large. The problem then becomes one of dealing with the stiffness of the reaction matrix, involving rate coefficients varying over many orders of magnitude. For most practical problems the number of independent reactions is less than the number of species. This implies that there are one or more biogeochemical components whose mass must be conserved during the reactions. Biogeochemical components are distinct from biogeochemical species in that the mass of any biogeochemical components must be conserved with respect to reactions (Rubin, 1983). Under such circumstance, the integration of Eq. (2.2.18) may not guarantee mass conservation of biogeochemical components due to numerical errors. Thus, numerical integration of Eq. (2.2.18) will make sense only when the following two conditions are met: (1) all reactions are slow and their rates are comparable within a narrow range, and (2) the rank of $v$ matrix should equal to $M$. For practical problems, these two conditions cannot be met simultaneously. Therefore, Eq. (2.2.18) must be manipulated to decouple fast equilibrium reactions from slow kinetic reactions and to explicitly enforce mass conservation of biogeochemical components.

Equation (2.2.18) in matrix form can be decomposed based on the type of biogeochemical reactions via Gauss-Jordan column reduction of $v$ (Chilakapati, 1995; Steefel and MacQuarrie, 1996; Yeh et al., 2001a). Consider a system involving $N_e$ fast equilibrium reactions and $N_k$ slow kinetic reactions among $M$ biogeochemical species. For a reaction that is fast and instantaneously reaches equilibrium among all species involved, its reaction rate can be conceptually considered infinity. An infinite rate is mathematically represented by a mass action equation. As a result, only linearly independent equilibrium reactions have to be considered, and any kinetic reaction that is linearly dependent on only equilibrium reactions is irrelevant to the system.

From the above discussion, the first step in the reduction is to find out the rank of the matrix made of $N_e$ equilibrium reactions. The rank of a matrix can be determined using any standard matrix operation package (Press et al., 1992). If the rank of the equilibrium reaction matrix is $N_e$, which should be less than or equal...
to $N_e$, then any subset of $N_e$ equilibrium reactions can be chosen as the $N_e$ linearly independent equilibrium reactions and the remaining $(N_e - N_e)$ equilibrium reactions are discarded from further consideration in the reaction network. It should be noted that it does not matter which $N_e$ equations are chosen. Next step is to determine which of the $N_k$ kinetic reactions is linearly dependent on only the chosen $N_e$ equilibrium reactions by checking the rank of the $N_e + 1$ matrix (defined as the matrix made of $N_e$ equilibrium reactions and one kinetic reaction, and one such matrix for each kinetic reaction). As a result, $N_k$ kinetic reactions, which is less than or equal to $N_k$ are left.

Starting with $N_e$ fast reactions and $N_k$ slow reactions, $N_e$ linearly independent equilibrium reactions and $N_k$ kinetic reactions are obtained. Among those $N_k$ kinetic reactions, some are linearly independent reactions while some may be linearly dependent on at least one other kinetic reaction and a number of equilibrium reactions. In other words, a subset, say $N_{ki}$, of $N_k$ kinetic reactions are linearly independent kinetic reactions while the remaining $N_k - N_{ki} = N_{kd}$ reactions are linearly dependent kinetic reactions. Again, the selection of $N_{ki}$ linearly independent kinetic reactions is not unique. The remaining task is to formally decompose the reaction matrix $v$, which is made of $N_e$ linearly independent equilibrium reactions and $N_k$ kinetic reactions. The decomposition of $v$ is also not unique (Yeh et al., 2001a, 2001b).

Denote $N_i$ as the number of linearly independent reactions, i.e., $N_i = N_e + N_{ki}$. After decomposition, the set of component species, the number of which is equal to $M - N_i$, is chosen. The remaining species are the product species (the number of product species is equal to $N_i$), which can be further classified as equilibrium or kinetic species. In order to minimize the number of simultaneous equations in biogeochemical modeling, all fast reactions that are represented by mass action equations should be decoupled from the kinetic reactions. In other words, each fast reaction can be used to eliminate one biogeochemical species from simultaneous consideration. An eliminated species is termed an equilibrium species. Since there are $N_e$ linearly independent equilibrium reactions, there are $N_e$ equilibrium species, each representing one fast reaction. The choice of the set of equilibrium species among $N_i$ product species is not unique when at least one of the fast reactions involves more than one product species. To obtain a suitable set of equilibrium species, a Gauss-Jordan row decomposition of the reaction matrix that is made of equilibrium species is performed, resulting in $N_e$ equilibrium species. The remaining $(N_i - N_e = N_{kd})$ product species are classified as the kinetic species for convenience.

With the above discussion, Eq. (2.2.18) is decomposed into the following equation:

$$
\theta A \frac{d \mathbf{C}}{dt} = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\begin{bmatrix}
\frac{d \mathbf{C}_1}{dt} \\
\frac{d \mathbf{C}_2}{dt}
\end{bmatrix} = \begin{bmatrix}
\mathbf{D} & \mathbf{K}
\end{bmatrix}
\begin{bmatrix}
\mathbf{R}_1 \\
\mathbf{0}_2
\end{bmatrix}
$$

(2.2.20)

where $A$ is the reduced $U$ matrix; $A_{11}$ is a submatrix of $A$ with size of $N_i \times N_i$, $A_{12}$ is a submatrix of $A$ with size of $N_i \times N_c$, $A_{21}$ is a submatrix of $A$ with size of $N_c \times N_i$, and $A_{22}$ is a submatrix of $A$ with size of $N_c \times N_c$; $C_1$ and $C_2$ are subvectors of the vector $C$; $D$ is the diagonal matrix representing a submatrix of the reduced $v$ with size of $N_i \times N_i$ reflecting the effects of $N_i$ linearly independent reactions on the production-consumption rate of all kinetic-variables, $K$ is a submatrix of the reduced $v$ with size of $N_i \times N_{kd}$, reflecting the effects of $N_{kd}$ dependent kinetic reactions, $\mathbf{0}_1$ is a zero matrix representing a submatrix of the reduced $v$ with size $N_c \times N_i$, and $\mathbf{0}_2$ is a zero matrix representing a submatrix of the reduced $v$ with size $N_c \times N_{kd}$; and $R_1$ and $R_2$ are subvectors of the vector $R$. 

15
Perform the same matrix decomposition procedure to Eq. (2.2.19) as to Eq. (2.2.18), the decomposed Eq. (2.2.19) can be expressed as:

\[
\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial C_1}{\partial t} \\
\frac{\partial C_2}{\partial t}
\end{bmatrix}
+ \begin{bmatrix}
\alpha' & \frac{\partial h}{\partial t} & \frac{\partial \theta}{\partial t}
\end{bmatrix}
\begin{bmatrix}
C_1 \\
C_2
\end{bmatrix}
- \begin{bmatrix}
M_1 \\
M_2
\end{bmatrix} =
\begin{bmatrix}
\frac{\partial^2 \theta}{\partial t^2}
\end{bmatrix}
\begin{bmatrix}
B_{11} & B_{12} \\
B_{21} & B_{22}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial \theta}{\partial t} & \frac{\partial \theta}{\partial t} & \frac{\partial \theta}{\partial t}
\end{bmatrix}
\begin{bmatrix}
C_1 \\
C_2
\end{bmatrix}
- \begin{bmatrix}
\theta & \theta & \theta
\end{bmatrix}
\begin{bmatrix}
D_1 & K_1 \\
D_2 & K_2
\end{bmatrix}
\begin{bmatrix}
R_1 \\
R_2
\end{bmatrix}
\tag{2.2.21}
\]

where \( B_{11}, B_{12}, B_{21}, \) and \( B_{22} \) are defined similarly to \( A_{11}, A_{12}, A_{21}, \) and \( A_{22} \), respectively, but with entries of zeros corresponding to immobile species; and \( M_1 \) and \( M_2 \) are sub vector of the vector \( M \), respectively.

The decomposition of Eqs. (2.2.19) to (2.2.21) effectively reduces a set of \( M \) simultaneous partial differential equations (PDEs) to three subsets of equations: the first contains \( N_E \) nonlinear algebraic equations representing the law of mass action for \( N_E \) equilibrium reactions; the second contains \( (N_I-N_E) \) simultaneous partial differential equations representing the transport of \( (N_I-N_E) = N_K \) kinetic variables, and the third contains \( N_C \) partial differential transport equations representing mass balance of \( N_C \) biogeochemical components. These equations are cast as follows:

(1) **Mass Action Equations for \( N_E \) Equilibrium Reactions**

\[
\frac{\partial \theta E_i}{\partial t} + \theta a' \frac{\partial h}{\partial t} E_i - M_i = \left\{ \theta (E_i^a) - Q E_i^a \right\} + \theta D_{kk} R_{kk} + \theta \sum_{j \in N_{KD(k)}} D_{ij} R_{j}, \quad i \in M, \ k \in N_E \tag{2.2.22}
\]

in which \( E_i = \sum_{\ell \in M} a_{\ell} C_{\ell} \) and \( E_i^a = \sum_{\ell \in M_a} b_{\ell} C_{\ell} \). Since the last term on the right-hand side of Eq. (2.2.22) is much smaller then the second last term, Equation (2.2.22) is approximated with

\[
\frac{\partial \theta E_i}{\partial t} + \theta a' \frac{\partial h}{\partial t} E_i - M_i - \left\{ \theta (E_i^a) - Q E_i^a \right\} \approx \theta D_{kk} R_{kk} = \infty, \quad \text{which implies there exists a thermodynamically consistent system of \( N_E \) algebraic equations.}
\]

(2) **Transport Equations for \( N_I-N_E \) Kinetic-Variables**

\[
\theta \frac{\partial E_i}{\partial t} + \left( \theta a' \frac{\partial h}{\partial t} + \frac{\partial \theta}{\partial t} \right) E_i - M_i = \left\{ \theta (E_i^a) - Q E_i^a \right\} + \theta \left( D_{kk} R_{kk} + \sum_{j \in N_{KD(k)}} D_{ij} R_{j} \right), \quad i \in M, \ k \in N_{KI} \tag{2.2.23}
\]

in which \( E_i = \sum_{\ell \in M} a_{\ell} C_{\ell} \) and \( E_i^a = \sum_{\ell \in M_a} b_{\ell} C_{\ell} \).
Transport Equations for $N_C$ Biogeochemical Components

$$\frac{\partial \theta_i}{\partial t} + \left( \theta_i \frac{\partial h}{\partial t} + \frac{\partial \theta_i}{\partial t} \right) E_i = m_i = \omega(E_i^a) - Q(E_i^a), \quad i \in M$$

in which $E_i = \sum_{\ell \in M} a_{\ell} C_{\ell}$ and $E_i^a = \sum_{\ell \in M_a} b_{\ell} C_{\ell}$; (2.2.24)

we normally denote $T_j = E_i$ and $T_j^a = E_i^a$, $j \in N_C$

where $M_i$ is the number of mobile species or aqueous phase species; $E_i$ and $E_i^a$ are the linear combinations of species and mobile species concentration resulting from the matrix decomposition; and $N_{DKD}$ is the subset of linearly dependent kinetic reactions, which depends on the k-th linear independent reaction. For this generic system, the variable $E_i$ is called an equilibrium variable in Eq. (2.2.22), a kinetic variable in Eq. (2.2.23), and a component variable (normally denoted with $T_j$) in Eq. (2.2.24), respectively. In Eqs. (2.2.22) though (2.2.24) $a_i$ and $b_i$ are the stoichiometry of the $\ell$-th species in the i-th equilibrium variable or kinetic variable or component.

### 2.2.4. Reaction Rate Formulation

The key in modeling mixed equilibrium and biogeochemical kinetic reactions is the formulation of rate equations for all $N$ reactions. A rate equation must be specified in order to quantitatively describe a general biogeochemical reaction that is written as:

$$\sum_{i=1}^{M} \mu_{ik} G_i \rightarrow \sum_{i=1}^{M} v_{ik} G_i, \quad k \in N$$

where $G_i$ is the chemical formula of the i-th species involved in the k-th reaction.

For an elementary kinetic reaction the rate law is given by collision theory (Smith, 1981; Atkins, 1986) as

$$R_k = K_{k}^f \prod_{i=1}^{M} \{A_i\}^{^\alpha_{ik}} - K_{k}^b \prod_{i=1}^{M} \{A_i\}^{^\alpha_{ik}}, \quad k \in N_K$$

where $R_k$ is the rate of the k-th kinetic reaction, $A_i$ is the activity of the i-th species, $K_{k}^f$ and $K_{k}^b$ are the activity-based forward and backward rate constant of the k-th kinetic reaction, respectively, and $N_K$ is the number of kinetic reactions. It is seen that the forward and backward rate constants cannot be determined one by one with the measurement of concentration-versus-time curves of all species because $N_K$ equations in Eq. (2.2.26) are coupled with respect to the forward and backward rate constants.

When a kinetic reaction cannot be modeled with an elementary rate, it may be formulated based on either empirical or mechanistic approaches (Steefel and van Cappellen; 1998, Yeh et al., 2001a). To make HYDROGEOCHEM 5.0 completely general, for any non-elementary kinetic reaction, its rate may be users-specified as follows:

$$R_k = R_k(C_1, C_2, ..., C_M; k_1, k_2, ...)$$

(2.2.27)
where $R_k$ is the prescribed rate law by the user written as a function of species concentrations and a number of parameters; $C_i$ is the concentration of the $i$-th species and $k_1$, $k_2$, $\ldots$ are rate parameters used to fit experimental data. For example, all mechanistic-based enzymatic kinetics presented in Segel (Segel, 1975) can be cast in Eq. (2.2.27). Any $n$-th order rate equation can also be cast in Eq. (2.2.27).

If the reaction is an equilibrium reaction, the reaction rate is infinity resulting in the law of mass action as:

$$R_k = \infty \equiv K_k^e = \left( \prod_{i=1}^{M} A_i^{y_{ik}} \right) \left( \prod_{i=1}^{M} A_i^{y_{ik}} \right), \quad k \in N_E$$

(2.2.28)

where $K_k^e$ is the equilibrium constant of the $k$-th reaction, $A_i$ is the activities of the $i$-th species, and $N_E$ is the number of linearly independent equilibrium reactions. The equilibrium constants can be determined one by one with the measurement of the activities of all species using Eq. (2.2.28). For precipitation-dissolution reactions, the activity of solid species is assumed to be constant and equal to unity. Furthermore, only linearly independent equilibrium reactions need to be considered because the mass action equation for any linearly dependent reaction can be obtained by the combinations of the $N_E$ equations of Eq. (2.2.28).

To make HYDROGEOCHEM 5.0 completely general, a users-specified algebraic equation can be used to describe a fast reaction when it cannot be modeled with a mass-action equilibrium equation

$$R_k = \infty \equiv F_k(C_1, \cdots, C_M; p_1, p_2, \cdots) = 0$$

(2.2.29)

where $F_k$ is an implicit function of species concentrations with a number of parameters. For example, the linear (Kd approach) and nonlinear (Freundlich) isotherms describing heterogeneous reactions with partitioning between aqueous and adsorbed chemicals fall into this category.

### 2.2.5 Initial and Boundary Conditions for Reactive Transport

To complete the description of the hydrological transport of biogeochemical, initial and boundary conditions must be specified in accordance with dynamic and physical considerations. It will be assumed initially that the concentration of each biogeochemical species must be given throughout the region of interest, that is

$$T_j = T_{jo} \text{ at } t = 0, \quad j \in N_C$$

(2.2.30)

$$E_i = E_{io} \text{ at } t = 0, \quad i \in M - N_C - N_E$$

(2.2.31)

where $T_{jo}$ is the initial concentration of the $i$-th component species (M/L^3), and $E_{io}$ is the initial concentration of the $i$-th kinetic variable (M/L^3).

Boundary conditions must be specified for all biogeochemical entities that are subject to hydrologic transport. However, the specification of boundary conditions is a difficult and intricate task in multicomponent transport modeling. From the dynamic point of view, a boundary segment may be classified as either flow-through or impervious. From the physical point of view, it is a soil-air interface, a soil-soil interface, or a soil-water interface. From the mathematical point of view, it may be treated as a Dirichlet boundary, on which the concentration of all species are prescribed, a Neumann boundary, on which the flux due to the gradient of species concentrations are known, or a Cauchy boundary, on which the fluxes are given. An even more difficult mathematical boundary is the variable condition, in which the boundary conditions are not known.
as a priori but are themselves part of the solution to be sought. In other words, on the mathematically variable boundary, either Neumann or Cauchy conditions may prevail and change with time. Which condition prevails at a particular time can be determined only in the cyclic processes of solving the governing equations (Freeze 1972a, 1972b; Yeh and Ward 1980, 1981).

Whatever point of view is chosen, all boundary conditions eventually must be transformed into mathematical equations for quantitative simulations. Thus, we can specify the boundary conditions from the mathematical point of view in concert with dynamic as well as physical considerations. The boundary conditions imposed on any segment of the boundary can be either Dirichlet, Neumann, Cauchy, or Variable, and the global boundary may be split into four parts: \( B_d, B_n, B_c, \) and \( B_v \), denoting Dirichlet, Neumann, Cauchy, and Variable boundaries, respectively.

A Dirichlet boundary condition is used to deal with a prescribed concentration as:

\[
T_j = T_{jd} \text{ on } B_d, \ j \in N_C \tag{2.2.32}
\]

\[
E_i = E_{id} \text{ on } B_d, \ i \in M - N_C - N_E \tag{2.2.33}
\]

where \( T_{jd} \) is the prescribed Dirichlet concentration for the \( i \)-th component species \((M/L^3)\), and \( E_{id} \) is the prescribed Dirichlet concentration for the \( i \)-th kinetic variable \((M/L^3)\).

A boundary is often treated as Neumann boundary when the flux due to the gradient of species concentration is known, or as a Cauchy boundary when the total flux is given:

\[
-n \cdot \nabla T_j^a = q_{jn} \text{ on } B_n, \ j \in N_C \tag{2.2.34}
\]

\[
n \left\{ \nabla T_j^a - \theta D \cdot \nabla T_j \right\} = q_{jc} \text{ on } B_c, \ j \in N_C \tag{2.2.35}
\]

and

\[
-n \cdot \nabla E_i^a = q_{in} \text{ on } B_n, \ i \in M - N_C - N_E \tag{2.2.36}
\]

\[
n \left\{ \nabla E_i^a - \theta D \cdot \nabla E_i \right\} = q_{ic} \text{ on } B_c, \ i \in M - N_C - N_E \tag{2.2.37}
\]

where \( q_{jn} \) and \( q_{in} \) are the normal Neumann fluxes of the \( j \)-th component and \( i \)-th kinetic variable, respectively \( [(M/L^2)/T]\); \( q_{jc} \) and \( q_{ic} \) are the normal Cauchy fluxes of the \( j \)-th component and \( i \)-th kinetic variable, respectively, \( [(M/L^2)/T]\); and \( n \) is the outward unit vector normal to the boundary.

A flow-through boundary is normally the soil-air interface or soil-water interface, and the appropriate conditions are variable on the direction of transport across the boundary. For this type of boundary, a variable-type boundary is imposed. When the flow is directed out of the region, the boundary is depicted by the Neumann condition with zero gradient flux; when the flow is directed into the region, the boundary is depicted by the Cauchy condition with given total flux. Written mathematically, the Variable boundary condition can be given by:

\[
-n \cdot \nabla T_j^a = 0 \text{ if } n \cdot V > 0 \text{ on } B_v, \ j \in N_C \tag{2.2.38}
\]
\[ n \{ V T_j^a - \theta D \nabla T_j^a \} = q_{jv} \text{ if } n \cdot V \leq 0 \text{ on } B_v, \ j \in N_C \]  
(2.2.39)

and

\[-n \cdot \theta D \nabla E_i^a = 0 \text{ if } n \cdot V > 0 \text{ on } B_v, \ i \in M - N_C - N_E \]  
(2.2.40)

\[ n \{ V E_i^a - \theta D \nabla E_i^a \} = q_{iv} \text{ if } n \cdot V \leq 0 \text{ on } B_v, \ i \in M - N_C - N_E \]  
(2.2.41)

2.2.6. Effect of Precipitation/Dissolution on Hydrodynamic Dispersion

An intermediate hydrodynamic dispersion coefficient \( AKDC \) can be computed as

\[ (AKDC)_o = D_{dp} + \theta_o D_w \tau \delta = a_f H \delta + (a_L - a_T)VV / |V| + \theta_o D_w \tau \delta \]  
(2.2.42)

where \( D_{dp} \) is the dispersion part of hydrodynamic dispersion, \( D_w \) is the given diffusion coefficient in water, and \( \theta_o \) is either the input moisture content or equal to \( \theta \) which is calculated from the coupling flow simulation. From the Archie's law and the paper of Steefel and Lichtner (1994), we can further derive

\[ \theta D_{dp} = \frac{D_w}{F} = \frac{\theta_o D_w}{\theta_o F} = \theta_o D_w [\theta_o^{-1} (\theta_o - \theta_o \phi_p)^m] = \theta_o D_w [\theta_o^{m-1} (1 - \phi_p)^m] \]  
(2.2.43)

where \( D_{dp} \) is the diffusion part of the hydrodynamic dispersion and \( m \) is the cementation exponent (Dullien, 1979) with the reported values between 1.3 and 2.5. In the above derivation, we have assumed

\[ \theta = \theta_o - \theta_o \phi_p \]  
(2.2.44)

which means the precipitation will only take out the pore spaces from effective pore water no matter whether there is pore air available or not. For both with or without flow simulation, we can always use Eq. (2.2.42) to adjust dispersion coefficient because \( \theta_o \) is fixed and used for computing biogeochemical concentration as well as \( \phi_p \) during transport simulations. Next, if we also assume that the dispersivities \( a_L \) and \( a_T \) are affected by the precipitation/dissolution in a similar way as those for the \( D_{dp} \), the dispersion part \( D_{dp} \) can be adjusted by using the same function as that for the diffusion part. The new hydrodynamic dispersion coefficient \( AKDC \) can be computed as

\[ AKDC = (AKDC)_o [\theta_o^{m-1} (1 - \phi_p)^m] \]  
(2.2.45)

The above equation provides the adjustment on both dispersion and diffusion part in performing the transport simulations. That means even though \( V \) and \( \theta_o \) are both computed with the precipitation effect in the flow module, we still need to adjust the dispersion coefficient (computed based on \( V \) and \( \theta_o \)) to reflect the precipitation effect on the hydrodynamic dispersion. Since the precipitation might be changed in each nonlinear transport loop accounting for precipitation and/or adsorption, this adjustment should be performed during each nonlinear transport loop. For the case without precipitation/dissolution effect on the dispersion coefficient, one should set \( m = 1.0 \) and \( V_i = 0 \) for all possible precipitated minerals.
2.3. Heat Transfer Equations

2.3.1. Governing Equations

The governing equations 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 (Yeh and Luxmoore, 1983). Based on their derivation, the following governing equation in conservative form 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; (2) The medium compressibility is negligible.

\[
\rho C_\theta + \rho_b C_m \frac{\partial T}{\partial t} + \nabla \left[ (\rho C) VT \right] + \left( C \frac{\partial \rho \theta}{\partial t} \right) T - \nabla \cdot (\mathbf{D}^T \nabla T) = \text{SSH} \tag{2.3.1}
\]

where \( T \) is the temperature (K); \( \rho \) is the density of water (M/L\(^3\)); \( \theta \) is effective moisture content (L\(^3\)/L\(^3\)) \( = n_e S \) in which \( n_e \) is the effective porosity (L\(^3\)/L\(^3\)) and \( S \) is degree of effective saturation of water (L\(^3\)/L\(^3\)); \( \rho_b \) is the bulk density of the dry medium (M/L\(^3\)); \( C \) and \( C_m \) are the specific heats of the groundwater and the dry medium in subsurface systems, respectively [(L\(^2\)/T\(^2\))/K]; \( V \) is the Darcy's velocity of the groundwater (L/T); \( \mathbf{D}^T \) is the thermal dispersion/diffusion/conductivity coefficient tensor [(ML/T\(^3\))/K]; and \( \text{SSH} \) is the source/sink term of heat that may be due to artificial injection and withdraw as well as biogeochemical reactions [(M/L)/T\(^3\)].

The thermal dispersion/diffusion/conductivity coefficient tensor \( \mathbf{D}^T \) can be calculated as:

\[
\mathbf{D}^T = \rho C \left[ a_T |V| \delta + (a_L - a_T) \frac{V \cdot V}{|V|^2} \right] + a_m \theta \tau \delta + \lambda \delta \tag{2.3.2}
\]

where \( a_T = \) transverse diffusivity [L], \( \delta = \) Kronecker delta tensor, \( |V| = \) magnitude of Darcy's velocity \( V \) (L/T), \( a_L = \) longitudinal diffusivity [L], \( a_m = \) molecular diffusion coefficient [(ML/T\(^3\))/K], \( \tau = \) tortuosity, \( \lambda = \) the apparent thermal conductivity [(ML/T\(^3\))/K].

For the case of injection, with the given injection rate \( q \) and the temperature of the injecting fluid, the heat injected into the system can be computed. Similarly, for the case of withdraw, we specify the pumping rate \( q \) at the pumping point where the temperature is the same as the resident fluid. Thus the heat withdrawal would be

\[
\text{SSH} = C \rho q T \tag{2.3.3}
\]

For the case of biogeochemical reactions, \( \text{SSH} \) is given by

\[
\text{SSH} = \sum_{k=1}^{N} \Delta H_k \tag{2.3.4}
\]

where \( N \) is the number of reactions and \( \Delta H_k \) is the heat release rate of the \( k \)-th reaction [(M/L)/T\(^3\)].

When the conventional finite element method is employed, we define:

\[
\begin{align*}
C_{RT} &= \rho C_\theta + \rho_b C_m, \\
\mathbf{V}_{CR} &= \rho C V, \\
C_{RH} &= C \frac{\partial \rho \theta}{\partial t}
\end{align*} \tag{2.3.5}
\]

Eq. (2.3.1) can be written as
Equation (2.3.6) is the governing equation for heat transfer in conservative form. It can also be converted to advective form as described in the following section.

2.3.2. Conversion to Advective Form

The continuity equation of flow can be expressed as:

$$
\frac{\partial \rho \theta}{\partial t} = -\nabla \cdot (\rho V) + \rho^* q
$$

(2.3.7)

where $\rho^*$ is the density of source/sink fluid ($= \rho$ if sink/withdraw with negative $q$ or $\rho_{in}$ if source/injection with positive $q$) ($M/L^3$), $q$ is source or sink term due to artificial injection or withdraw of water, unit of volume of water per volume of porous media per time [$(L^3/L^3)/T$].

Multiply both sides of Eq. (2.3.7) by $CT$, we get

$$
CT \frac{\partial \rho \theta}{\partial t} = -CT \nabla \cdot (\rho V) + CTP^* q
$$

(2.3.8)

Substitution of Eq. (2.3.8) into (2.3.1) yields:

$$
\left[ \frac{\partial \rho \theta + \rho b C m}{\partial t} \right] + (\rho C) \nabla \cdot V T - \nabla \cdot (D T \nabla T) = (\rho^* C q)^T s - (\rho^* C q) T
$$

(2.3.9)

where $T^s$ is the source temperature (K). Let:

$$
C_{RQ} = \rho^* C q
$$

(2.3.10)

Equation (2.3.9) can be written as:

$$
C_{RT} \frac{\partial T}{\partial t} + C_{RQ} T + V_{CR} \nabla T - \nabla \cdot (D T \nabla T) = C_{RQ} T^s
$$

(2.3.11)

When the Lagrangian-Eulerian approach is employed to deal with the convection term in the heat transfer equation, Eq. (2.3.11) can be written in the following form

$$
C_{RT} \frac{DV_T}{Dt} + C_{RQ} T - \nabla \cdot (D T \nabla T) = C_{RQ} T^s
$$

(2.3.12)

And the tracking velocity $V_T$ can be expressed as

$$
V_T = \frac{V_{CR}}{C_{RT}} = \frac{\rho C V}{\rho C \theta + \rho b C m} = \frac{V}{\theta + \frac{C_m \times \rho_b}{\rho}}
$$

(2.3.13)

where $V_T$ is the convective velocity used in the particle tracking process within the Lagrangian step. Equation (2.3.12) is a linear partial differential equation and the associated coefficient matrix will be symmetric positive definite (Press et al., 1992).
To perform transient simulations, the temperature must be prescribed initially at all nodes. In addition, appropriate boundary conditions have to be specified for the simulated problem. They are described as follows.

The initial conditions for the simulation of heat transfer can be written as:

$$T = T_i(x, y, z) \text{ in } R$$  \hspace{1cm} (2.3.14)

where $T_i$ represents the initial temperature distribution over the entire domain of interest $R$, it can be obtained by either experimental measurements or solving steady-state version of the governing equation of heat transfer.

Four types of boundary conditions are used to simulate a variety of situations in practice as shown below:

**Dirichlet Boundary Conditions:**

$$T = T_d(x_b, y_b, z_b) \text{ on } B_d$$  \hspace{1cm} (2.3.15)

where $(x_b, y_b, z_b)$ is the spatial coordinate on the boundary and $T_d$ is the prescribed functional value of the temperature on the Dirichlet boundary $B_d$. This boundary condition is used when the boundary temperature can be identified.

**Neumann Boundary Conditions:**

$$-\mathbf{n} \cdot \mathbf{D} T \cdot \nabla T = q_n(x_b, y_b, z_b) \text{ on } B_n$$  \hspace{1cm} (2.3.16)

where $\mathbf{n}$ is the outward unit vector normal to the boundary, and $q_n$ is the prescribed gradient thermal flux through the Neumann boundary $B_n$. This boundary condition 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.

**Cauchy Boundary Conditions:**

$$\mathbf{n} \left( \rho \mathbf{C} \mathbf{V} T - \mathbf{D} T \cdot \nabla T \right) = q_c(x_b, y_b, z_b) \text{ on } B_c$$  \hspace{1cm} (2.3.17)

where $q_c$ 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.

**Variable Boundary Conditions:**

$$-\mathbf{n} \cdot \mathbf{D} T \cdot \nabla T = 0 \text{ if } \mathbf{n} \cdot \mathbf{V} > 0 \text{ on } B_v$$  \hspace{1cm} (2.3.18)

$$\mathbf{n} \left( \rho \mathbf{C} \mathbf{V} T - \mathbf{D} T \cdot \nabla T \right) = \rho \mathbf{C} (\mathbf{n} \cdot \mathbf{V}) T_v(x_b, y_b, z_b) \text{ if } \mathbf{n} \cdot \mathbf{V} < 0 \text{ on } B_v$$  \hspace{1cm} (2.3.19)

where $T_v$ 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. (2.3.18) is considered when the flow is outgoing while Eq. (2.3.19) is for the inward flow.
2.3.4. Effect of temperature on the constants of biogeochemical reactions

Based on the well-known Van't Hoff relationship, temperature and biogeochemical reactions of a system interact with each other. For equilibrium reactions, the equilibrium constant, $K^e$, is temperature-dependent and can be computed with the following equation:

$$\ln \frac{K^e}{K^e_0} = \frac{\Delta H^e_0}{R} \left( \frac{1}{T} - \frac{1}{T_0} \right) \Rightarrow K^e = K^e_0 \exp \left[ \frac{\Delta H^e_0}{R} \left( \frac{T - T_0}{T T_0} \right) \right]$$

(2.3.20)

For kinetic reactions, the forward and backward rate constants, $K^f$ and $K^b$, are also temperature dependent, they are calculated as

$$\ln \frac{K^f}{K^f_0} = \frac{\Delta H^f_0}{R} \left( \frac{1}{T} - \frac{1}{T_0} \right) \Rightarrow K^f = K^f_0 \exp \left[ \frac{\Delta H^f_0}{R} \left( \frac{T - T_0}{T T_0} \right) \right]$$

(2.3.21)

$$\ln \frac{K^b}{K^b_0} = \frac{\Delta H^b_0}{R} \left( \frac{1}{T} - \frac{1}{T_0} \right) \Rightarrow K^b = K^b_0 \exp \left[ \frac{\Delta H^b_0}{R} \left( \frac{T - T_0}{T T_0} \right) \right]$$

(2.3.22)

where $K^e$ and $K^e_0$ are temperature dependent and referenced equilibrium constants for equilibrium reactions, respectively; $K^f$ and $K^f_0$ are the temperature dependent and referenced forward rate constants for kinetic reactions, respectively; $K^b$ and $K^b_0$ are the temperature dependent and referenced backward rate constants for kinetic reactions, respectively; and $\Delta H^e_0$, assumed unchanged with temperature, is the enthalpy of the associated equilibrium reactions. Similarly, $\Delta H^b_0$ and $\Delta H^f_0$ are the forward and the backward enthalpy of the kinetic reaction, respectively; $T$ is the temperature of interest; $T_0$ is the referenced temperature associated with $K^e_0$ or $K^f_0$ and $K^b_0$; $R$ is the ideal gas constant [(M/L)/(T K)]; $\ln$ is the natural logarithm operator; and $\exp$ represents the exponential operator. In Eqs. (2.3.20) through (2.3.22), $K^e_0$ or $K^f_0$ and $K^b_0$, $R$, and $T_0$ are all input by users, and $\Delta H^e_0$, $\Delta H^f_0$ and $\Delta H^b_0$ are calculated from each species' enthalpy input by users. Any consistent unit system can be used to perform the computation.
3. NUMERICAL IMPLEMENTATION

The initial-boundary value problem described by the governing equations of the flow, heat transfer, and reactive biogeochemical transport modules of HYDROGEOCHEM 5.0 along with the boundary conditions cannot, in general, be solved analytically using current applied mathematics. Hence, in order to solve these sets of governing equations, numerical methods are the only mathematical tools capable of handling this task. Although there are many different numerical approximation methods capable of reducing partial differential equations to more simple systems of algebraic equations, there are only two numerical methods that are most common and that can be employed to the most basic form of the governing equations. These two numerical methods are the finite difference and finite element methods. The basic difference between these two methods is that the finite element method is based upon approximating the function, while the finite difference method is founded upon approximating the derivatives of the function. Therefore, the finite difference method only produces solutions at discrete points, while the finite element method yields spatially continuous solutions. Also, the finite element method offers numerous advantages over the finite difference method, such as (1) anisotropy and heterogeneity of aquifers are easily taken care of, (2) there is no need to formulate special formulae to incorporate irregular boundaries, (3) computer storage and computational time can sometimes be saved because often less nodal points are needed to portray the region of interest to the same level of accuracy, (4) irregular grids for handling different levels of spatial discretization in different sections of the region of interest can be incorporated, and lastly, (5) the integral formulation used in this method permits the flux types of the boundary conditions to come about naturally (Yeh, 1987a). Thus, the finite element method is used in this model. The theoretical background as well as numerical procedures of this method can be found in any good finite element method book, such as Istok (1989), and therefore will not be described here. A brief summary of the numerical procedure for applying the finite element method can be found in Yeh (1987a).

The flow module of HYDROGEOCHEM 5.0 includes 4 options for solving the finite element equations. In other words, there are 4 methods (block iteration, successive point iteration, polynomial preconditioned conjugate gradient, incomplete Cholesky preconditioned conjugate gradient methods) for solving the linearized matrix equations, and the reactive biogeochemical transport and heat transfer have 6 options (4 options above and modified incomplete Cholesky preconditioned conjugate gradient methods and symmetrical over-relaxation preconditioned conjugate gradient methods) for solving the linearized matrix equations. Because the Newton-Ralphson method will yield a non-symmetric matrix, the Picard method is used to linearize the matrix equation.

To handle a large variety of possible problem sets, the flow module of HYDROGEOCHEM 5.0 contains 16 optional numerical schemes. Specifically, the mixture of schemes includes the combinations of: (a) the four options for solving the resulting matrix equation as mentioned in the above discussion, (b) two options (lumping and consistent) for handling the mass matrix resulting from the storage term, and (c) two options (time-weighted difference and mid-difference) for approximating the time derivatives. The theoretical background for (b) and (c) may also be found in any respectable matrix computation books and in Yeh (1991).

On the other hand, the reactive biogeochemical transport module for HYDROGEOCHEM 5.0 includes these 16 options, plus more. The thermal transport module is dealt with similarly as the reactive biogeochemical transport. The conventional finite element method is used in the flow module. An option of two conventional finite element methods, either the Galerkin finite element method or the upstream weighting finite element method, and an alternative option of a hybrid Lagrangian-Eulerian finite element method is provided in the
biogeochemical transport module and the heat transfer module. The main difference between the two conventional finite element methods is that while the Galerkin finite element method uses the base functions as the weighting functions, the upstream weighting finite element method uses weighting functions different from the base functions. The advantages of using the upstream weighting finite element method over the Galerkin finite element method become apparent when the advection terms in the transport governing equation are equally important to the dispersion terms (Yeh and Ward, 1981). More details of the two conventional finite element methods may be found in Yeh and Ward (1981).

3.1. Solution of Fluid Flow Equations

3.1.1. Spatial Discretization of Flow Equations with the Galerkin Finite Element Method

When using the finite element method, the referenced pressure head is approximated by:

\[ h = \hat{h} = \sum_{j=1}^{N} h_j(t)N_j(x,y,z) \]  

(3.1.1)

where \( h_j \) and \( N_j \) are the amplitude of \( h \) and the base function, respectively, at nodal point \( j \) and \( N \) is the total number of nodes. After defining a residual and forcing the weighted residual to zero, the flow equation, Eq.(2.1.1), is approximated as:

\[ \left[ \int_{R} \frac{\rho}{\rho_o} F_N dR \right] \frac{dh_j}{dt} + \left[ \int_{R} (\nabla N_j) \cdot K \cdot (\nabla N_j) dR \right] h_j = \int_{R} \frac{\rho}{\rho_o} q dR - \int_{R} (\nabla N_j) \cdot K \cdot \frac{1}{\rho_o} \rho z dR + \int_{B} n \cdot K \left( \nabla h + \frac{1}{\rho_o} \nabla z \right) N_i dB \]  

(3.1.2)

In matrix form, Eq.(3.1.2) is written as:

\[ [M] \left\{ \frac{dh}{dt} \right\} + [S] \{h\} = \{Q\} + \{G\} + \{B\} \]  

(3.1.3)

where \( \{dh/dt\} \) and \( \{h\} \) are the column vectors containing the values of \( dh/dt \) and \( h \), respectively, at all nodes; \([M]\) is the mass matrix resulting from the storage term; \([S]\) is the stiff matrix resulting from the action of conductivity; and \( \{Q\}, \{G\}, \) and \( \{B\} \) are the load vectors from the internal source/sink, gravity force, and boundary conditions, respectively. Furthermore, the mass matrix, \([M]\), and stiff matrix, \([S]\), are described as:

\[ M_{ij} = \sum_{e \in M_e} \int_{R_e} N^e_{\alpha} \frac{\rho}{\rho_o} FN^e_{\beta} dR \]  

(3.1.4)

and

\[ S_{ij} = \sum_{e \in M_e} \int_{R_e} (\nabla N^e_{\alpha}) \cdot K \cdot (\nabla N^e_{\beta}) dR \]  

(3.1.5)

where \( R_e \) is the region of element \( e \), \( M_e \) is the set of elements that have a local side \( \alpha-\beta \) coinciding with the global side \( i-j \); \( N^e_{\alpha} \) and \( N^e_{\beta} \) are the \( \alpha \)-th and \( \beta \)-th local base function of element \( e \), respectively.
In addition, the three load vectors, $\{Q\}$, $\{G\}$, and $\{B\}$, are described as:

$$Q_i = \sum_{e \in M_i} \int_{R_e} N^e \cdot \frac{\rho}{\rho_0} q dR$$  \hspace{1cm} (3.1.6)

$$G_i = - \sum_{e \in M_i} \int_{R_e} (\nabla N^e_i) \cdot \mathbf{K} \cdot \frac{\rho}{\rho_0} \nabla z dR$$  \hspace{1cm} (3.1.7)

and

$$B_i = - \sum_{e \in N_{se}} \int_{R_e} N^e \cdot \mathbf{n} \left[ -\mathbf{K} \left\{ \nabla h + \frac{\rho}{\rho_0} \nabla z \right\} \right] dB$$  \hspace{1cm} (3.1.8)

where $N_{se}$ is the set of boundary segments that have a local node $\alpha$ coinciding with the global node $i$, and $B_e$ is the area of boundary segment $e$.

In most finite element work, the Darcy's velocity components given in Eq. (2.1.4) are calculated numerically by taking the derivatives of the simulated $h$ as

$$\mathbf{V} = -\mathbf{K} \left\{ \frac{\rho}{\rho_0} (\nabla h)_j + \nabla z \right\}$$  \hspace{1cm} (3.1.9)

The above formulation results in velocity field which is not continuous at element boundaries and nodal points if the variation of $h$ is other than linear or constant. The alternative approach would be to apply the Galerkin finite element method to Eq. (2.1.4), thus obtaining

$$[T]\{V_x\} = \{D_x\}$$  \hspace{1cm} (3.1.10)

$$[T]\{V_y\} = \{D_y\}$$  \hspace{1cm} (3.1.11)

$$[T]\{V_z\} = \{D_z\}$$  \hspace{1cm} (3.1.12)

where the matrix $[T]$ and the load vectors $\{D_x\}$ and $\{D_z\}$ are given by

$$D_{xi} = - \sum_{e \in M_i} \int_{R_e} N^e_i \cdot \mathbf{i} \cdot \mathbf{K} \cdot \left\{ \frac{\rho}{\rho_0} \nabla h + \nabla z \right\} dB$$  \hspace{1cm} (3.1.13)

$$D_{yi} = - \sum_{e \in M_i} \int_{R_e} N^e_i \cdot \mathbf{j} \cdot \mathbf{K} \cdot \left\{ \frac{\rho}{\rho_0} \nabla h + \nabla z \right\} dB$$  \hspace{1cm} (3.1.14)
\[ D_{zi} = - \sum_{e \in M_e} \int_{R_e} N_a^i \mathbf{k} \cdot \left\{ \frac{\rho}{\rho} \nabla h + \nabla z \right\} dB \]  

(3.1.15)

\[ T_{ij} = \sum_{e \in M_e} \int_{R_e} N_a^i N_b^j dR \]  

(3.1.16)

where \( V_x \), \( V_y \), and \( V_z \) are the Darcy's velocity components along the x-, y- and z-directions, respectively, and \( \mathbf{i}, \mathbf{j} \) and \( \mathbf{k} \) are the unit vectors along the x-, y- and z-coordinates, respectively.

The reduction of the partial differential equation Eq. (2.1.1) to the set of ordinary differential equations Eq. (3.1.3) simplifies to the evaluation of integrals on the right hand side of Eqs. (3.1.4) through (3.1.8) for every element for boundary surface \( e \). The major tasks that remain to be done are the specification of base functions and the performance of integration to yield the element matrices. Three types of elements are employed in this documentation: hexahedral elements, triangular prisms, and tetrahedral elements.

### 3.1.2. Base and Weighting Functions

The construction of base functions for hexahedral elements is best accomplished using the local coordinates \( (\xi, \eta, \zeta) \). In the local coordinates, the original hexahedral element is mapped into a cubic whose corners are located at \( \xi = \pm 1\), \( \eta = \pm 1\) and \( \zeta = \pm 1\) as shown in Fig. 3.1.

![Fig. 3.1 A Hexahedral Element in Local Coordinates](image-url)
For tri-linear hexahedral elements, the eight base functions are obtained by taking the tensor product of the three base functions of the linear line elements as

\[ N_i(\xi, \eta, \zeta) = \frac{1}{8}(1 + \xi \xi_i)(1 + \eta \eta_i)(1 + \zeta \zeta_i), \quad i = 1, 2, ..., 8 \quad (3.1.17) \]

Because the Galerkin finite element method is used to solve the flow equations, the set of eight weighting functions is taken as the same set of eight base functions for hexahedral elements. The construction of base and weighting functions for triangular prisms and tetrahedral elements can be found elsewhere (Yeh, 1999, 2000).

### 3.1.3. Numerical Integration

To complete the reduction of the partial differential equation Eq. (2.1.1) to the ordinary differential equation Eq. (3.1.3), one has to evaluate the integrals on the right-sides of Eqs. (3.1.4) through (3.1.8) for every element to yield the element mass matrix \([M^e]\) and the element stiff matrix \([S^e]\) as well as the element gravity column vector \([G^e]\), the element source/sink column vector \([Q^e]\), and the element boundary column vector \([B^e]\) as

\[ M^e_{ij} = \int_{R_e} N_i^e \rho \frac{dF}{d\rho} N_j^e \rho dR \quad (3.1.18) \]

\[ S^e_{ij} = \int_{R_e} \left( \nabla N_i^e \cdot \mathbf{K} \cdot \nabla N_j^e \right) dR \quad (3.1.19) \]

\[ Q^e_i = \int_{R_e} N_i^e \frac{\rho}{\rho_o} q dR \quad (3.1.20) \]

\[ G^e_i = -\int_{R_e} \left( \nabla N_i^e \right) \cdot \mathbf{K} \cdot \frac{\rho}{\rho_o} \nabla z dR \quad (3.1.21) \]

and

\[ B^e_i = -\int_{R_e} N_i^e \mathbf{n} \cdot \left( \mathbf{K} \cdot \left[ \nabla h + \frac{\rho}{\rho_o} \nabla z \right] \right) dB \quad (3.1.22) \]

Since Eqs. (3.1.18) through (3.1.22) are written in the global coordinate and the base functions are defined in the local coordinate for hexahedral elements, a transformation between the global and local coordinate is needed. The required transformation from global coordinate to local coordinate is obtained via the base functions as

\[ x = \sum_{j=1}^{8} x_j N_j(\xi, \eta, \zeta) \quad (3.1.23) \]
Because the coordinate transformation uses the base functions, the element is termed the "isoparametric" element.

Using the transformation in Eqs. (3.1.23) through (3.1.25), we convert the differentiation of the base function with respect to the global coordinate to that with respect to the local coordinate by

\[
\begin{bmatrix}
\frac{\partial N_i}{\partial \xi} \\
\frac{\partial N_i}{\partial \eta} \\
\frac{\partial N_i}{\partial \zeta}
\end{bmatrix}
= [J]^{-1}
\begin{bmatrix}
\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\
\frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\
\frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta}
\end{bmatrix}
, \quad [J] = \begin{bmatrix}
\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\
\frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\
\frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta}
\end{bmatrix}
\]

where \([J]\) is the Jacobian of the transformation. In the mean time, a differential volume written in the local coordinate becomes

\[
\int_{e} dR = \int_{-1}^{1} \int_{-1}^{1} J d\xi d\eta d\zeta \quad (3.1.27)
\]

With Eqs. (3.1.26) and (3.1.27), all the integrals in Eqs. (3.1.18) through (3.1.21) can be reduced to the following form

\[
\int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} f(\xi,\eta,\zeta) J d\xi d\eta d\zeta
\]

the integration of which can easily be carried out with a 2 x 2 x 2 = 8 point Gaussian quadrature for hexahedral elements. Numerical integrations of Eqs. (3.1.8) through (3.1.21) for triangular prisms and tetrahedral elements are relatively easy and can be found elsewhere (Yeh, 1999).

The surface integration of Eq. (3.1.22) for a 4-node surface in three dimensions is not as straightforward as in two-dimensional space. This integration requires further elaboration. Any surface integral of a continuous function \(F(x,y,z)\) specified on the surface \(S\) (Fig. 3.2) can be reduced to the area integration. Let \(I\) represent the surface integral:

\[
I = \int_{S} F(x,y,z) dS \quad (3.1.29)
\]
where the surface $S$ is given by the following equation

$$z = f(x, y) \quad (3.1.30)$$

Let $P$ be any point on the surface $S$ with coordinates $(x, y, z)$ or $(\xi, \eta, \zeta)$ (Fig. 3.2). Then the vector $r$ from $O$ to $P$ is given by

$$r = x \mathbf{i} + y \mathbf{j} + z \mathbf{k} \quad (3.1.31)$$

The tangent vectors to the coordinate curves $\xi = \xi_0$ and $\eta = \eta_0$ on the surface $S$ are $\frac{\partial r}{\partial \xi}$ and $\frac{\partial r}{\partial \eta}$, respectively.

![Fig. 3.2 A Surface Area and its Imbedded Local Coordinate](image)

The area $dS$ is given by

$$dS = \left| \frac{\partial r}{\partial \xi} \times \frac{\partial r}{\partial \eta} \right| \, d\xi d\eta \quad (3.1.32)$$

where $\times$ represents vector multiplication. But

$$\frac{\partial r}{\partial \xi} \times \frac{\partial r}{\partial \eta} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{vmatrix} \quad (3.1.33)$$

so that

$$dS = \sqrt{J_x^2 + J_y^2 + J_z^2} \, d\xi d\eta \quad (3.1.34)$$
Substituting Eq. (3.1.34) into Eq. (3.1.29), we obtain

\[
\int_{S} F(x,y,z) dS = \int_{-1}^{1} \int_{-1}^{1} \phi(\xi,\eta) \sqrt{J_x^2 + J_y^2 + J_z^2} d\xi d\eta
\]  

(3.1.36)

where

\[
\phi(\xi,\eta) = F(x(\xi,\eta), y(\xi,\eta), z(\xi,\eta))
\]

(3.1.37)

Surface integrations of Eq. (3.1.36) can easily be computed by Gaussian quadrature for four-node surfaces in three dimensions. The surface integration of Eq. (3.1.22) for a three-node surface in three dimensions is relatively straightforward and can be found elsewhere (Yeh, 1999).

3.1.4. Mass Lumping Option

Referring to [M], one may recall that this is a unit matrix if the finite difference formulation is used in spatial discretization. Hence, by proper scaling, the mass matrix can be reduced to the finite-difference equivalent by lumping (Clough, 1971). In many cases, the lumped mass matrix would result in better solution, in particular, if it is used in conjunction with the central or backward-difference time marching (Yeh and Ward, 1980). Under such circumstances, it is preferred to the consistent mass matrix (mass matrix without lumping). Therefore, options are provided for the lumping of the matrix [M]. More explicitly, [M] will be lumped according to:

\[
M_{ij} = \sum_{\alpha = 1}^{N} \left( \sum_{\beta = 1}^{4} \int_{R_{\alpha}} N_{\alpha}^\beta \frac{P}{P_{0}} dR \right) \text{ if } j = i,
\]

(3.1.38)

and

\[
M_{ij} = 0 \text{ if } j \neq i
\]

(3.1.39)

3.1.5. Finite Difference Approximation in Time

Next, we derive a matrix equation by integrating Eq. (3.1.3). For the time integration of Eq. (3.1.3), the load vector {B} will be ignored. This load vector will be discussed in the next section on the numerical implementation of boundary conditions. An important advantage in finite element approximation over the finite difference approximation is the inherent ability to handle complex boundaries and obtain the normal derivatives therein. In the time dimension, such advantages are not evident. Thus, finite difference methods are typically used in the approximation of the time derivative. Two time-marching methods are adopted in
the present flow model.

The first one is the time weighted method written as:

\[
\frac{\Delta t}{\Delta t}(h_{t+\Delta t} - \{h\}_t) + \omega [S]h_{t+\Delta t} + (1 - \omega) [S]\{h\}_t = \{Q\} + \{G\}
\]  

(3.1.40)

where \([M]\), \([S]\), \{Q\}, and \{G\} are evaluated at \((t + \omega \Delta t)\). In the Crank-Nicolson centered-in-time approach \(\omega = 0.5\), in the backward-difference (implicit difference) \(\omega = 1.0\), and in the forward-difference (explicit scheme) \(\omega = 0.0\). The central-Nicolson algorithm has a truncation error of \(O(\Delta t^2)\), but its propagation-of-error characteristics frequently lead to oscillatory nonlinear instability. Both the backward-difference and forward-difference have a truncation error of \(O(\Delta t)\). The backward-difference is quite resistant to oscillatory nonlinear instability. On the other hand, the forward difference is only conditionally stable even for linear problems, not to mention nonlinear problems.

In the second method, the values of unknown variables are assumed to vary linearly with time during the time interval, \(\Delta t\). In this mid-difference method, the recurrence formula is written as:

\[
\left( \frac{2}{\Delta t} [M] + [S] \right) \{h\}_{t+\Delta t/2} - \frac{2}{\Delta t} [M]\{h\}_t = \{Q\} + \{G\}
\]  

(3.1.41)

and

\[
\{h\}_{t+\Delta t} = 2\{h\}_{t+\Delta t/2} - \{h\}_t
\]  

(3.1.42)

where \([M]\), \([S]\), and \{Q\} are evaluated at \((t+\Delta t/2)\).

Eqs. (3.1.40) and (3.1.41) can be written as a matrix equation

\[
[T]\{h\} = \{Y\}
\]  

(3.1.43)

where \([T]\) is the matrix, \{h\} is the unknown vector to be found and represents the values of discretized pressure field at new time, and \{Y\} is the load vector. Take for example, Eq. (3.1.40) with \(\omega = 1.0\), \([T]\) and \{Y\} represent the following:

\[
[T] = \frac{[M]}{\Delta t} + [S],
\]  

(3.1.44)

and

\[
\{Y\} = \frac{[M]}{\Delta t}\{h\}_t + \{Q\} + \{G\}
\]  

(3.1.45)

where \{h\}_t is the vector of the discretized pressure field at previous time.
3.1.6. Numerical Implementation of Boundary Conditions

The following steps are the incorporation of boundary conditions into matrix equation by finite element method. For the Cauchy boundary condition given by Eq. (2.1.8), we simply substitute Eq. (2.1.8) into Eq. (3.1.22) to yield a boundary-element column vector \( \{B_c^e\} \) for a Cauchy segment:

\[
\{B_c^e\} = \{q_c^e\} \quad (3.1.46)
\]

where \( \{q_c^e\} \) is the Cauchy boundary flux vector given by

\[
q_c^e = -\int_{B_c} N_a^e \frac{\rho}{\rho_o} q_c dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \quad (3.1.47)
\]

The Cauchy boundary flux vector represents the normal fluxes through the four nodal points of the segment \( B_c \). For the Neumann boundary condition given by Eq. (2.1.7), we substitute Eq. (2.1.7) into Eq. (3.1.22) to yield a boundary-element column vector \( \{B_n^e\} \) for a Neumann segment:

\[
\{B_n^e\} = \{q_n^e\} \quad (3.1.48)
\]

where \( \{q_n^e\} \) is the Neumann boundary flux vector given by:

\[
q_n^e = \int_{B_n} N_a^e n \cdot K \cdot \frac{\rho}{\rho_o} \nabla z - N_a^e q_n dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \quad (3.1.49)
\]

which is independent of pressure head in the case of saturated flow, but depends on pressure head via the hydraulic conductivity in the case of unsaturated flows.

The implementation of variable-type boundary conditions is more involved. During the iteration of boundary conditions on the variable boundary, one of Eqs. (2.1.9) through (2.1.13) is used at a node. If either Eq. (2.1.10) or (2.1.13) is used, we substitute it into Eq. (3.1.22) to yield a boundary-element column vector \( \{B_v^e\} \) for a variable boundary segment:

\[
\{B_v^e\} = \{q_v^e\} \quad (3.1.50)
\]

where \( \{q_v^e\} \) is the variable boundary flux given by:

\[
q_v^e = -\int_{B_v} N_a^e \frac{\rho}{\rho_o} q_v dB \quad \text{or} \quad q_v^e = -\int_{B_v} N_a^e \frac{\rho}{\rho_o} q_v dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \quad (3.1.51)
\]

Assembling over all Neumann, Cauchy, and variable boundary segments, we obtain the global boundary column vector \( \{B\} \) as:

\[
\{B\} = \{q\} \quad (3.1.52)
\]

in which

\[
\{q\} = \sum_{e \in N_{nc}} \{q_c^e\} + \sum_{e \in N_{ce}} \{q_c^e\} + \sum_{e \in N_{ve}} \{q_v^e\} \quad (3.1.53)
\]

where \( N_{nc}, N_{ce}, \) and \( N_{ve} \) are the number of Neumann boundary segments, Cauchy boundary segments, and
variable boundary segments with flux conditions imposed on them, respectively. The boundary flux \{B\} given by Eqs.(3.1.52) and (3.1.53) should be added to the right hand side of Eq.(3.1.43).

For the river boundary condition given by Eq.(2.1.14), we simply substitute Eq.(2.1.14) into Eq.(3.1.22) to yield a boundary-element column vector \{B_R^e\} and boundary matrix \[B_R^e\] for a river boundary segment:

\[
\{B_R^e\} = \{q^e\} \quad \text{and} \quad \left[ B_R^e \right] = \left[ b^e \right] \quad (3.1.54)
\]

where \{q^e\} and \[b^e\] are the contributions of the river boundary to the right hand and left hand side of the matrix equation, respectively. They are given by

\[
q^e_{\alpha} = \int_{B_e} N_{\alpha} \frac{\rho}{\rho_o} \frac{K_R}{b^e} h dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \quad \text{and} \quad \alpha = 1, 2, 3, 4
\]

\[
b^e_{\alpha\beta} = \int_{B_e} N_{\alpha} \frac{\rho}{\rho_o} \frac{K_R}{b^e} N_{\beta} dB; \quad \alpha, \beta = 1, 2, 3, 4 \quad \text{or} \quad \alpha, \beta = 1, 2, 3
\]

The boundary flux \{B_R^e\} should be added to the right-hand side of Eq. (3.1.43), and the boundary matrix \[B_R^e\] should be added to the matrix \([T]\) on the left-hand side of Eq. (3.1.43) for all river boundary segments.

At nodes where Dirichlet boundary conditions are applied, an identity equation is generated for each node and included in the matrices of Eq.(3.1.43). The Dirichlet nodes include the nodes on the Dirichlet boundary and the nodes on the variable boundary to which either Eq.(2.1.9), (2.1.11), or (2.1.12) is applied.

After time discretization of Eq. (3.1.3) and incorporation of boundary conditions, we obtain the following matrix equation

\[
[C]\{h\} = \{R\} \quad (3.1.56)
\]

where \([C]\) is the coefficient matrix and \{\{R\}\} is the known vector of the right hand side. For the saturated-unsaturated flow simulation, \([C]\) is a highly nonlinear function of the pressure head \{h\}.

### 3.1.7. Solution of the Matrix Equations

Equation (3.1.56) is, in general, a banded sparse matrix equation. It may be solved numerically by either direct methods or iteration methods. In direct methods, a sequence of operation is performed only once. This would result in an exact solution except for round-off error. In this method, one is concerned with the efficiency and the magnitude of round-off error associated with the sequence of operations. On the other hand, in an iterative method, one attempts to the solution by a process of successive approximations. This involves making an initial guess, then improving the guess by some iterative process until an error criterion is obtained. Therefore, in this technique, one must be concerned with convergence, and the rate of convergence. The round-off errors tend to be self-corrected.

For practical purposes, the greatest advantages of direct method are: (1) the efficient computation when the bandwidth of the matrix \([C]\) is small, and (2) the fact that no problem of convergency is encountered when
the matrix equation is linear or less severity in convergence than iterative methods even when the matrix equation is nonlinear. The greatest disadvantages of direct methods are the excessive requirements on CPU storage and CPU time when a large number of nodes is needed for discretization. On the other hand, the greatest advantages of iterative methods are the efficiencies in terms of CPU storage and CPU time when large problems are encountered. Their greatest disadvantages are the requirement that the matrix, \([C]\), must be well conditioned to guarantee a convergent solution. For three-dimensional problems, the bandwidth of the matrix is usually large, thus the direct elimination method is not practical. Only the iterative methods are implemented in HYDROGEOCHEM 5.0. Four methods are used in solving the linearized matrix equation: (1) block iteration, (2) successive point iteration, (3) polynomial preconditioned conjugate gradient method, and (4) incomplete Cholesky preconditioned conjugate gradient method.

The matrix equation, Eq. (3.1.56), is nonlinear because both the hydraulic conductivity and the water capacity are functions of the pressure head \(h\). To solve the nonlinear matrix equation, two approaches can be taken: (1) the Picard method and (2) the Newton-Ralphson method. The Newton-Ralphson method has a second order of convergent rate and is very robust. However, the Newton-Ralphson method would destroy the symmetrical property of the coefficient matrix resulting from the finite element approximation. As a result the solution of the linearized matrix equation requires extra care. Many of the iterative methods will not warrant a convergent solution for the non-symmetric linearized matrix equation. Thus, the Picard method is used in this report to solve the nonlinear problems.

In the Picard method, an initial estimate is made of the unknown \(\{h\}\). Using this estimate, we then compute the coefficient matrix \([C]\) and solve the linearized matrix equation by the method of linear algebra. The new estimate is now obtained by the weighted average of the new solution and the previous estimate:

\[
\{ h^{(k+1)} \} = \omega \{ h \} + (1-\omega) \{ h^k \},
\]

where \(\{h^{(k+1)}\}\) is the new estimate, \(\{h^k\}\) is the previous estimate, \(\{h\}\) is the new solution, and \(\omega\) is the iteration parameter. The procedure is repeated until the new solution \(\{h\}\) is within tolerance error. If \(\omega\) is greater than or equal to 0 but is less than 1, the iteration is under-relaxation. If \(\omega = 1\), the method is the exact relaxation. If \(\omega\) is greater than 1 but less than or equal to 2, the iteration is termed over-relaxation. The under-relaxation should be used to overcome cases when nonconvergency or the slow convergent rate is due to fluctuation rather than due to "blowup" computations. Over-relaxation should be used to speed up convergent rate when it decreases monotonically.

In summary, there are 16 optional numerical schemes here to deal with as wide a range of problems as possible. These are the combinations of: (1) two ways of treating the mass matrix (lumping and no-lumping); (2) two ways of approximating the time derivatives (time-weighting and mid-difference), and (3) four ways of solving the linearized matrix equation.

### 3.2. Solution of Reactive Biogeochemical Transport Equations

According to custom definition, species in aqueous phase are considered to be mobile and every solid or solid associated species is considered to be immobile. If at least one of the species in a component is an aqueous species, the component is a mobile component. Otherwise the component is an immobile component. Similarly, when at least one of the species in a kinetic variable is an aqueous species, the kinetic variable is a mobile kinetic variable. Otherwise the kinetic variable is an immobile one. Only the mobile components
Rewrite the transport equation for mobile components in advection form:

\[
\frac{\partial T_j}{\partial t} + \frac{\partial}{\partial t}(T_j a_j) - \alpha \frac{\partial h}{\partial t} T_j + M_j^a - Q T_j^a, \quad j \in N_m
\]  

(3.2.1)

Similarly, the transport equations for mobile kinetic variables are:

\[
\frac{\partial E_j}{\partial t} + \frac{\partial}{\partial t}(E_j a_j) - \alpha \frac{\partial h}{\partial t} E_j + M_j^a - Q E_j^a + R_j, \quad j \in N_m
\]  

(3.2.2)

where \( T_j, T_j^a, T_j^s, \) and \( T_j^p \) are the total analytical, total dissolved, total sorbed, and total precipitated concentrations of the \( j \)-th mobile component, respectively; \( E_j, E_j^a, E_j^s, E_j^p \) are the total analytical, total dissolved, total sorbed, and total precipitated concentrations of the \( j \)-th kinetic variable, respectively; \( N_m \) is the number of mobile components; \( M_m \) is the number of mobile kinetic variables; \( a_j \) is the stoichiometry of the \( j \)-th component; \( b_j \) is the stoichiometry of the \( j \)-th kinetic variable; \( M_j^a \) is the total rate of source/sink of the \( j \)-th mobile component or mobile kinetic variable; \( M \) is the number of species presented in the system; \( M_s \) is the number of mobile species; \( M_s \) is the number of sorbent species; \( M_p \) is the number of precipitated species; and \( R_j \) is the production rate for the \( j \)-th mobile kinetic variable due to all biogeochemical reactions [this term should vanish for the component species as shown in Eq. (3.2.1)].

Because the transport equation for component species is mathematically a special case of the transport equations for kinetic variables, only the latter will be discussed here. Three numerical approaches are employed to solve the transport equation. These three options are (1) applying conventional FEM to the advective form of equations, (2) hybrid Lagrangian-Eulerian FEM; and 3) hybrid Lagrangian-Eulerian FEM for interior nodes + applying conventional FEM to the advective form of equation for boundary nodes.

For the case when conventional FEM in advective form approach is used, we rewrite Eq. (3.2.2) as:

\[
\left( \frac{\partial}{\partial t} + \alpha \frac{\partial h}{\partial t} \right) E + V \cdot \nabla E + K(E) =
\]

(3.2.3)

\[
V \cdot \nabla (E^a + E^p) = K(E^a + E^p) + qE^{a_m} - \frac{p^*}{\rho} E^a + R \quad where \quad qE^{a_m} = M^a
\]

For the case when hybrid Lagrangian-Eulerian approach is used, Eq. (3.2.2) becomes:
where \( E^{\text{in}} \) is the input concentration of the mobile kinetic variable if \( q \) is a source or is equal to \( E \) if \( q \) is a sink (withdraw); and \( K \) is an operators denoting:

\[
K(\mid) = \left\{ -\nabla \cdot (\theta \mathbf{D} \nabla) + \frac{\partial \theta}{\partial t} \right\}(\mid)
\]

\[
Q_{d\nu} = -F \frac{\partial h}{\partial t} + \frac{\partial \theta}{\partial t} - \frac{\rho_0}{\rho_0} \mathbf{V} \cdot \nabla \frac{\rho_0}{\rho_0}
\]

The three approaches are detailed below:

(1) **Fully Implicit Sequential Iteration Approach**

In the fully implicit approach, the biogeochemical reaction subsystem of equations and the hydrologic transport subsystems of equations are solved sequentially and iteratively. The concentrations of all immobile PDVs are computed in the biogeochemical subsystem using all species concentrations computed from the prior iteration. The concentrations of mobile PDVs are determined in the hydrologic module using the species concentrations and the reaction term evaluated from species concentrations of the current iteration. Using this approach, Equation (3.2.7) is rearranged for total analytical kinetic variable as:

\[
\frac{\partial E}{\partial t} = \mathcal{A}(E) + \mathcal{R}
\]

where \( \mathcal{A}(\mid) \) is the advective-dispersive transport operator defines as:

\[
\mathcal{A}(\mid) = \mathcal{A}(\mid) - \alpha \frac{\partial h}{\partial t}(\mid)
\]

and \( \mathcal{R} \) is a reaction operator defined as:

\[
\mathcal{R} = R - \frac{\partial (E^s + E^p)}{\partial t} - \alpha \frac{\partial h}{\partial t}(E^s + E^p)
\]
where \( n \) is the time step level and \( r \) is the iteration step.

The solution procedure is iterated until convergence is reached within both the reaction and transport modules. During each iteration, the coefficients in the governing equation are updated to the value of previous iteration. For the fully implicit sequential iteration approach, the solution procedure for every time step is outlined below:

Step 1: Provide guessed value for \( E \), say \( E^W \);
Step 2: With known \( E^W \), solve the biogeochemical problem [which is specified by the definition of \( E^W = E^W(C_1, C_2, \ldots, C_M) \) and a subset of equations in Eq. (2.2.20), including immobile kinetic-variable equations and all equilibrium equations] by to yield the concentrations of all species, from which \( E^s \) and \( E^p \) and \( R \) are calculated;
Step 3: With \( E^s \) and \( E^p \) and \( R \) known, solve Eq. (3.2.10) to update \( E \) and revise the guessed value \( E^W \);
Step 4: Check convergence by comparing the newly obtained \( E \) and \( E^W \);
Step 5: If a convergent solution is obtained, then update the coefficients in the governing equation and proceed to next time step computation; if the solution is not convergent, update \( E^W \) with new \( E \), repeat Step 2 through 4.

(2) Operator Splitting Approach

In this approach, the concentrations of mobile PDVs are solved for in both subsystems of equations. There is no iteration between the transport and biogeochemical reaction subsystems for a given time step with this approach. The mobile PDVs are first subjected to transport without considering the reaction term to yield an intermediate value of their concentrations. These transported concentrations are then subjected to the biogeochemical reactions, yielding final values of the concentrations for the current time step. In this approach, Eq. (3.2.7) is solved in two steps:

\[
\frac{(E^a)^{n+1/2} - (E^a)^n}{\Delta t} - \mathcal{A}[(E^a)^{n+1/2}] = 0
(3.2.11)
\]

\[
\frac{(E^a)^{n+1} - (E^a)^{n+1/2}}{\Delta t} - \mathcal{R}^{n+1} = 0
(3.2.12)
\]

where the superscripts \((n)\), \((n+1/2)\), and \((n+1)\) denote the values at the prior time, at an intermediate point in the current computation, and at the new time, respectively. For the operator-splitting approach, the solution procedure between two continuous time steps is outlined below:

Step 1: Solve Eq. (3.2.11) for intermediate value of \( E^a \);
Step 2: With known intermediate value of \( E^a \), solve Eq. (3.2.12) for the new value of \( E^a \).

(3) Predictor-Corrector Approach

In this approach the concentrations of mobile PDVs are solved for in both subsystems of equations. There is no iteration between the transport and biogeochemical subsystems for a given time step with this approach.
The mobile PDVs are first subjected to transport considering the reaction term for the previous time step to yield an intermediate value of their concentrations. These transported concentrations are then subjected to the biogeochemical reactions, yielding final values of the concentrations for the current time step. In this approach, Eq. (3.2.7) is also solved in two steps:

\[
\frac{(E^{a}_{n+1/2} - (E^{a}_{n})}{\Delta t} - \delta \left[ (E^{a}_{n+1/2}) \right] = R^n \tag{3.2.13}
\]

\[
\frac{(E^{a}_{n+1} - (E^{a}_{n+1/2})}{\Delta t} - R^{n+1} = -R^n \tag{3.2.14}
\]

where the superscripts \((n)\), \((n+1/2)\), and \((n+1)\) denote the values at the prior time, at an intermediate point in the current computation, and at the new time, respectively. For the predictor-corrector approach, the solution procedure between two continuous time steps is outlined below:

Step 1: Compute \(R^n\) based on \((E^{a}_{n})\) and \((E^{a}_{n-1})\);
Step 2: Solve Eq. (3.2.13) for intermediate value of \(E^{a}\);
Step 3: With known intermediate value of \(E^{a}\), solve Eq. (3.2.14) for new value of \(E^{a}\).

Among the three approaches, the fully implicit scheme provides the most accurate solution. The other two schemes provide a more robust, less computationally intensive but less accurate solution for those problems in which convergent solutions are difficult to achieve.

A large number of numerical approximations can be used to reduce the partial differential equations governing the hydrologic transport to a system of algebraic equations. The most common numerical methods used to approximate Eq. (3.2.1) are finite-difference methods (FDMs) and finite-element methods (FEMs) (Forsythe and Wasow, 1960; Huebner, 1975; Lapidus and Pinder, 1982). Many other numerical techniques, such as the integrated finite-difference method (IFDM) (Narasimhan and Witherspoon, 1977), the integrated compartment method (ICM) (Yeh and Luxmoore, 1983), or the method of characteristics (MOC) (Konikow and Bredehoeft, 1978), have been employed to deal with special cases of the hydrologic transport equations. Only the finite differences and finite elements can be applied to the most generalized form of the transport equations.

The advantages of FEMs are their inherent ability to make complex boundaries discrete, to make flux-type boundary conditions easy to deal with, and to allow the flexibility to include cross-derivative terms. Disadvantages of FEMs include the central processing unit (CPU) time required to obtain element matrices and the inflexibility of using iteration methods to solve the resulting matrix equation. The FDM offers great economy because it allows simple interpolation for the derivatives and provides flexibility of solving the resulting matrix equation with various iteration methods. However, it suffers from the following aspects: the regular rectangular grid system has to be used, the flux-type boundary conditions have to be extrapolated, and the cross-derivative terms cannot be consistently approximated.

The most severe limitations of the IFDM are its inability to treat anisotropic media and its use of the Jacobian iteration method, in which the rate of convergency is extremely slow; however, it offers even more flexibility than the FEMs in making the complex boundaries discrete, and the physical representation of the method is clearly understood. The ICM, while retaining the advantage of the IFDM, can deal with anisotropic media by defining new variables but at the expense of having to solve a large number of simultaneous field equations (Yeh and Luxmoore, 1983). In addition, ICM provides options of using the direct elimination
method and iteration methods with the Gauss-Seidel (G-S) or successive over-relaxation (SOR) schemes to solve the matrix equation (Yeh and Luxmoore, 1983). The MOC is best used to solve advection-dominant transport problems. The main limitations of the MOC lay in the fact that computer codes based on the method are problem specific and are very difficult to modify for generic applications.

In light of these discussions, FEMs are the preferred numerical methods. In addition, there has been significant recent progress in using iteration methods to solve finite-element equations (Yeh, 1985, 1986), and influence coefficient methods have been proposed to analytically and economically compute the element matrices (Huyakorn et al., 1985).

3.2.1. Spatial Discretization of Transport Equations

Again, the transport equation for component species is mathematically a special case of the transport equations for kinetic variables; only the latter will be discussed here. Eq. (3.2.3) or (3.2.4) is integrated in the spatial dimensions by the weighted residual method in conjunction with finite elements. Following the procedure of the finite-element weighted-residual method, the approximate formulation of the distribution of the total analytical concentration E is obtained. Let the variables E, E^a, E^s, and E^p be approximated by:

\[ E = \sum_{j=1}^{N} E_j N_j, \quad E^a = \sum_{j=1}^{N} E_j^a N_j, \quad E^s = \sum_{j=1}^{N} E_j^s N_j, \quad E^p = \sum_{j=1}^{N} E_j^p N_j \]  

(3.2.15)

where N_j is the basis function of the spatial coordinate for j-th node, E_j is the value of E at node j, E_j^a is the value of E^a at node j, E_j^s is the value of E^s at node j, E_j^p is the value of E^p at node j, and N is number of finite-element nodes in the region.

Upon substituting Eq. (3.2.15) into Eq. (3.2.3) or (3.2.4) and applying the conventional FEM, we obtain the following matrix equations:

\[ [M] \frac{\partial [E]}{\partial t} + ([V] + [K] + [D]) [E] = ([V] + [K]) ([E^s] + [E^p]) + [Q] + [B] + [R] \]  

(3.2.16)

for the case when the conventional FEM in advective form is used; and

\[ [M] \left( \frac{D [E^s]}{Dt} + \frac{\partial (\{E^s\} + \{E^p\})}{\partial t} \right) + ([K] + [D]) [E] = [K] ([E^s] + [E^p]) + [Q] + [B] + [R] \]  

(3.2.17)

for the case when the hybrid Lagrangian-Eulerian approach is used. In Eqs. (3.2.16) and (3.2.17), \{D [E^s]/Dt\} is the column vector containing the values of D [E^s]/Dt at all nodes; \{\partial (\{E^s\} + \{E^p\})/\partial t\} is the column vector containing \partial (\{E^s\} + \{E^p\})/\partial t at all nodes; \{E\} is the column vector containing the value of E at all nodes; \{E^s\} is the column vector containing the value of E^s at all nodes; \{E^p\} is the column vector containing the value of E^p at all nodes; [M] is the mass matrix resulting from the storage term; [V] is the stiff matrix resulting from the advection; [K] is the modified stiff matrix resulting from the combined action of dispersion, source of water, and the rate of change of moisture content; [D] is the growth matrix representing the effect of the rate of change of moisture content alone; \{Q\} is the load vector from the internal source; \{B\} is the load vector from the boundary.
source; and \{R\} is the reaction vector from all the biogeochemical reactions of the system.

The matrices \([M], [K], [V],\) and \([D]\) are given by:

\[
M_{ij} = \sum_{e \in M_{ij}} \int_{R_e} N^e_{\alpha} \theta N^e_{\beta} dR \\
K_{ij} = \sum_{e \in M_{ij}} \int_{R_e} \left[ \nabla N^e_{\alpha} \cdot \theta D \cdot \nabla N^e_{\beta} + N^e_{\alpha} \left( Q_{\theta \nu} - \frac{\partial \theta}{\partial t} \right) N^e_{\beta} \right] dR \\
V_{ij} = \sum_{e \in M_{ij}} \int_{R_e} W^e_{\alpha} V \cdot \nabla N^e_{\beta} dR \\
D_{ij} = \sum_{e \in M_{ij}} \int_{R_e} N^e_{\alpha} \left( \alpha \frac{\partial \theta}{\partial t} + \frac{\partial \theta}{\partial t} \right) N^e_{\beta} dR
\]

where \(R_e\) is the region of element \(e\), \(M_{ij}\) is the set of elements that have a local side coinciding \(\alpha-\beta\) with the global side \(i-j\), and \(N^e_{\alpha}\) and \(N^e_{\beta}\) are the \(\alpha\)-th and \(\beta\)-th local base functions of element \(e\), respectively, and \(W^e_{\alpha}\) is the \(\alpha\)-th local weighting function of element \(e\).

Similarly, the load vectors \{Q\}, \{B\} and \{R\} are given by:

\[
Q_i = \sum_{e \in M_{ij}} \int_{R_e} N^e_{\alpha} \left( E^a h - \frac{\rho^*}{\rho} E^a \right) dR \\
B_i = -\sum_{e \in N_{\alpha=1}} \int_{R_e} N^e_{\alpha} n \cdot (\theta D \cdot \nabla E^a) dB \\
R_i = \sum_{e \in M_{ij}} \int_{R_e} N^e_{\alpha} R dR
\]

where \(B_e\) is the area of boundary segment \(e\) and \(N_{se}\) is the set of boundary segments that have a local node \(\alpha\) coinciding with global node \(i\).

The reduction of the partial differential equation, Eq. (3.2.3) or (3.2.4), to the set of ordinary differential equations, Eq. (3.2.16) or (3.2.17), simplifies to the evaluation of integrals on the right-hand side of Eqs. (3.2.18) through (3.2.24) for every element or boundary segment \(e\). The major task that remains is the specification of base functions and the performance of integration to yield the element matrices.

This will be carried out for three types of elements: hexahedral elements, triangular prisms, and tetrahedral elements because these types of elements are employed in this report.
3.2.2. Base and Weighting Functions

For the case of flow, the weighting functions are taken as the same set as the base functions. However, in transport formulation using finite element methods with the Eulerian approach, sometimes, it is an advantage to use the weighting functions which are one or two order higher than the base functions: (N+1) or (N+2) upstream weighting. Here, we will only present the N+1 upstream weighting functions. First define, for the line element, the following N+1 upstream weighting functions

\[ F_1(\xi, \alpha) = N_1(\xi) - \frac{\alpha}{4}(1 + \xi)(1 - \xi) \]  
(3.2.25)

\[ F_2(\xi, \alpha) = N_2(\xi) + \frac{\alpha}{4}(1 + \xi)(1 - \xi) \]  
(3.2.26)

where \( \alpha \) is the weighting factor along the line from node 1 to node 2 (Fig. 3.3).

![Weighting Factor along a Line Element](image)

Fig. 3.3  Weighting Factor along a Line Element

Then the weighting functions are obtained by an appropriate tensor product:

\[ W_1 = F_1(\xi, \alpha_1)F_1(\eta, \beta_1)F_1(\zeta, \gamma_1) \]  
(3.2.27)

\[ W_2 = F_2(\xi, \alpha_1)F_1(\eta, \beta_2)F_1(\zeta, \gamma_2) \]  
(3.2.28)

\[ W_3 = F_2(\xi, \alpha_2)F_2(\eta, \beta_2)F_1(\zeta, \gamma_4) \]  
(3.2.29)

\[ W_4 = F_1(\xi, \alpha_2)F_2(\eta, \beta_1)F_1(\zeta, \gamma_3) \]  
(3.2.30)

\[ W_5 = F_1(\xi, \alpha_3)F_1(\eta, \beta_3)F_2(\zeta, \gamma_1) \]  
(3.2.31)

\[ W_6 = F_2(\xi, \alpha_3)F_1(\eta, \beta_4)F_2(\zeta, \gamma_2) \]  
(3.2.32)

\[ W_7 = F_2(\xi, \alpha_4)F_2(\eta, \beta_4)F_2(\zeta, \gamma_4) \]  
(3.2.33)

\[ W_8 = F_1(\xi, \alpha_4)F_2(\eta, \beta_3)F_2(\zeta, \gamma_3) \]  
(3.2.34)

where \( \alpha \)'s, \( \beta \)'s, and \( \gamma \)'s are the weighting factors along the side given in Figure 3.4.
The construction of upstream weighting functions for linear triangular prisms and tetrahedral elements can be found in Yeh (Yeh, 2000)

3.2.3. Evaluation of Element Matrices

To complete the reduction of the partial differential equation [Equation (3.2.1) or (3.2.2)] to the ordinary differential equation [Equation (3.2.16) or (3.2.17)], one has to evaluate the integrals on the right-hand sides of Equations (3.2.18) through (3.2.24) for every element to yield the element mass matrix $[M^e]$, the stiff matrix resulting from the advection $[V^e]$, the modified stiff matrix resulting from the combined action of dispersion, source of water, and the rate of change of moisture of content $[K^e]$, the growth matrix representing the effect of the rate of change of moisture of content alone $[D^e]$, the load vector from the internal source $\{Q^e\}$, the load vector from the boundary source $\{B^e\}$, and the reaction vector from all the biogeochemical reactions of the system $\{R^e\}$:

\[
M_{e\alpha\beta} = \int_{R_e} N^e_\alpha \theta N^e_\beta \, dR
\]  
(3.2.35)

\[
K_{e\alpha\beta} = \int_{R_e} \left( \nabla N^e_\alpha \cdot \theta \mathbf{D} \cdot \nabla N^e_\beta + N^e_\alpha \left( Q_{\alpha\beta} - \frac{\partial \theta}{\partial t} \right) \right) N^e_\beta \, dR
\]  
(3.2.36)

\[
V_{e\alpha\beta} = \int_{R_e} \mathbf{W}^e_\alpha \mathbf{v} \cdot \nabla N^e_\beta \, dR
\]  
(3.2.37)
\[
D_{\alpha\beta}^e = \int_{R_e} N_\alpha^e \left( \alpha \frac{\partial \theta}{\partial t} + \frac{\partial \theta}{\partial t} \right) N_\beta^e \, dR
\]  
(3.2.38)

\[
Q_a = \int_{R_e} N_a^e q \left( E_{ax} n - \frac{\partial}{\partial \rho} E_{ax} \right) \, dR
\]  
(3.2.39)

\[
B_a = -\int_{B_e} N_a^e \mathbf{n} \cdot (-\theta \mathbf{D} \cdot \nabla E_{ax}) \, dB
\]  
(3.2.40)

\[
R_a = \int_{R_e} N_a^e \mathbf{R} \, dR
\]  
(3.2.41)

where the superscript or subscript \( e \) denotes the element, \( \alpha, \beta = 1, 2, \ldots, 4 \) for Tetrahedral elements, \( \alpha, \beta = 1, 2, \ldots, 6 \) for Triangular Prism elements, and \( \alpha, \beta = 1, 2, \ldots, 8 \) for Hexahedral elements.

Following the procedures presented in Section 3.1.3, we first transform Eqs. (3.2.35) through (3.2.41) in terms of local coordinates for hexahedral elements, volume coordinates for tetrahedral elements, and mixed local and area coordinates for triangular prisms (Yeh, 2000). Then we integrate the resulting equations with the Gaussian quadrature. The transformation between the global and local coordinates is also given by Eqs. (3.1.23) through (3.1.25) resulting in isoparametric elements for hexahedral elements. The surface integration from the boundary conditions also follows that presented in Section (3.1.3).

With the element matrices \([M^e], [K^e], [V^e], \) and \([D^e]\) and the element column vector \([Q^e]\) computed, the global matrices \([M], [K], [V] \) and \([D]\) and the global column vector \([Q]\) are then assembled element by element.

### 3.2.4. Mass Lumping Option

Referring to the element mass matrix \([M^e]\) and growth matrix \([D^e]\), one may note that these are the unit matrices if the finite-difference formulation is adopted in the spatial discretization. Hence, by proper scaling, these matrices can be reduced to the finite-difference equivalent by lumping (Clough, 1971). In many cases, the lumped mass and growth matrices result in a better solution. Furthermore, with these lumped element matrices, the problem can be reduced to a beaker system when advection and dispersion-diffusion terms are not involved in the transport equation. Under such circumstances, they are preferred to the nonlumped mass and growth matrices. Therefore, an option is provided for the lumping of these matrices. A systematic and mathematically acceptable procedure for such lumping has been well established (Zienkiewicz, 1977). For example, the element mass matrix can be lumped according to the following:

\[
M_{\alpha\beta}^e = \sum_{\beta=1}^{8, 6, or 4} \int_{R_e} N_\alpha^e \theta N_\beta^e \, dR \quad \text{if} \quad \beta = \alpha \quad \text{and} \quad M_{\alpha\beta}^e = 0 \quad \text{if} \quad \beta \neq \alpha.
\]  
(3.2.42)

### 3.2.5. Time Integration

An important advantage of the finite-element approximation is the inherent ability to handle complex
boundaries and obtain the normal derivatives therein. In the time dimension, such advantages are not evident. Thus, FDMs are typically used in the approximation of the time derivative. Using a time weighting factor, \( w \), we obtain from Equation (3.2.16) or (3.2.17) the following matrix equation:

\[
\begin{align*}
\mathbf{G}\{E\}_{t+\Delta t} &= \mathbf{H}\left(\{E^s\}_{t+\Delta t} + \{E^p\}_{t+\Delta t}\right) + \{L\} + \{B\} + \{R\} \\
\text{(3.2.43)}
\end{align*}
\]

where \( \{E\}_{t+\Delta t} \) is the column vector representing the value of \( \{E\} \) at time \( (t+\Delta t) \), \( \{E^s\}_{t+\Delta t} \) is the column vector representing the value of \( \{E^s\} \) at time \( (t+\Delta t) \), \( \{E^p\}_{t+\Delta t} \) is the column vector representing the values of \( \{E^p\} \) at time \( (t+\Delta t) \), and \( \Delta t \) is the time-step size.

When the conventional FEM in advective form approach is used, the matrices \( \mathbf{G} \) and \( \mathbf{H} \) and the load vector \( \{L\} \) are given as:

\[
\begin{align*}
\mathbf{G} &= \frac{[M]}{\Delta t} + \omega([K] + [D]) + \omega_v[V] \\
\mathbf{H} &= \omega[K] + \omega_v[V] \\
\mathbf{L} &= \frac{[M]}{\Delta t}\{E\}_t + \{Q\} - (1-\omega_v)[V]\{E^s\}_t - (1-\omega)[K]\{E\}_t + [K]\{E^a\}_t \\
\text{(3.2.44)}
\end{align*}
\]

\[
\begin{align*}
\text{(3.2.45)}
\end{align*}
\]

\[
\begin{align*}
\text{(3.2.46)}
\end{align*}
\]

When the hybrid Lagrangian-Eulerian approach is used, \( \mathbf{G} \), \( \mathbf{H} \), and \( \mathbf{L} \) are defined as:

\[
\begin{align*}
\mathbf{G} &= \frac{[M]}{\Delta t} + \omega([K] + [D]) \\
\mathbf{H} &= \frac{[M]}{\Delta t} - \frac{[M]}{\Delta t} + \omega[K] \\
\mathbf{L} &= \frac{[M]}{\Delta t}\{E^a\} + \frac{[M]}{\Delta t}\{E^s\}_t + \{E^p\}_t + \{Q\} - (1-\omega)[D]\{E^a\}_t + \{E^s\}_t + \{E^p\}_t - (1-\omega)[K]\{E^a\} \\
\text{(3.2.47)}
\end{align*}
\]

\[
\begin{align*}
\text{(3.2.48)}
\end{align*}
\]

\[
\begin{align*}
\text{(3.2.49)}
\end{align*}
\]

in which \( \omega_v \) is the time weighting factor for the advection term, \( \omega \) is the time weighting factor for all other spatial derivatives terms, \( \{E\}_t \) is the value of \( \{E\} \) at time \( t \), \( \{E^s\}_t \) is the value of \( \{E^s\} \) at time \( t \), \( \{E^p\}_t \) is the value of \( \{E^p\} \) at time \( t \), \( \{E^a\} \) is the Lagrangian concentration, and \( w \) is the time weighting factors for spatial derivative terms. When \( \omega = 0 \), the time integration is explicit, \( \omega = 0.5 \), it is the Crank-Nicolson central difference, for the implicit (or backward) difference, \( \omega = 1.0 \). Similarly, for the advection term, when \( \omega_v = 0 \), the time integration is explicit, \( \omega_v = 0.5 \), it is the Crank-Nicolson central difference, for the implicit (or backward) difference, \( \omega_v = 1.0 \).

The Lagrangian concentration \( \{E^a\} \) is computed by the backward method of characteristics as follows:

\[
x^+_i = x_i - \int_t^{t+\Delta t} \frac{V}{\theta} \, dt \quad \text{and} \quad E^a_i = \sum_j E^s_j(t)N_j(x^+_i) \, dt \\
\text{(3.2.50)}
\]
where \( x_i^* \) (the Lagrangian point) is the location of the fictitious particle originating at time \( t \), which would arrive at the node \( x_i \) at time \( t + \Delta t \); \( E_j^e(t) \) is the value of concentration at node \( j \) at time \( t \); \( N_j(x_i^*) \) is the interpolation function associated with node \( j \) evaluated at the Lagrangian point \( x_i^* \). If \( x_i^* \) is located within the region of interest, we define \( \Delta \tau \) in Eqs (3.2.47) through (3.2.49) as:

\[
\Delta \tau = \Delta t \quad (3.2.51)
\]

If \( \Delta x^* \) is located outside the region of interest, we must find a \( \Delta \tau (x_i^*) \) such that

\[
x_i^* = x_i - \int_{t}^{t + \Delta \tau(x_i^*)} \frac{V}{\theta} dt \quad (3.2.52)
\]

will locate on the boundary. Thus, \( \Delta \tau \) is less than or equal to \( \Delta t \).

### 3.2.6. Numerical Implementation of Boundary Conditions

To incorporate the boundary conditions, we have to evaluate the right-hand side of Eq. (3.2.23) for every boundary segment \( B_e \) to yield the load vector \( \{B^e\} \):

\[
B^e_\alpha = - \int_{B_e} N^e_\alpha n \cdot (-\theta \nabla \cdot \text{VE}^a) dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \quad (3.2.53)
\]

For the Neumann boundary condition given by Eq. (2.2.34) or (2.2.36), we simply substitute Eq. (2.2.34) or (2.2.36) into Eq. (3.2.53) to yield a boundary-element column vector \( \{B^e_n\} \) for a Neumann segment:

\[
\{B^e_n\} = \{q^e_n\} \quad (3.2.54)
\]

where \( \{q^e_n\} \) is the Neumann boundary flux vector given by:

\[
q^e_n = - \int_{B_e} N^e_\alpha q_n dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \quad (3.2.55)
\]

The Neumann boundary flux vector represents the normal fluxes through the four nodal points of the segment \( B_e \) on \( B_n \).

For the Cauchy boundary condition given by Eq. (2.2.35) or (2.2.37), we may rewrite Eq. (3.2.53) in the following form:

\[
\begin{align*}
B^e_\alpha &= - \int_{B_e} N^e_\alpha n \cdot (\text{VE}^a - \theta \nabla \cdot \text{VE}^a) dB + \\
&\int_{B_e} N^e_\alpha n \cdot \text{VE} dB - \int_{B_e} N^e_\alpha n \cdot \text{VE}(a + E^a) dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3
\end{align*}
\]

The total analytical concentration on the boundary segment \( B_e \) can be approximated by:

\[
E = \sum_{\beta = 1}^{4 \text{ or } 3} E^e_\beta N^e_\beta \quad (3.2.57)
\]
Similarly, the total sorbed concentration, $E^s$, and the total precipitated concentration, $E^p$, on the same boundary segment can be approximated by:

$$E^s = \sum_{\beta = 1}^{4} E^s_{\beta} N^e_{\beta} \quad \text{and} \quad E^p = \sum_{\beta = 1}^{3} E^p_{\beta} N^e_{\beta}$$

(3.2.58)

Substituting Eqs. (3.2.57) and (3.2.58) into (3.2.56), we obtain the boundary-element column vector $\{B^e_c\}$ for a Cauchy segment:

$$\{B^e_c\} = \{q^e_c\} + [V^e_c]\{E\} - [V^e_c]\{(E^s) + (E^p)\}$$

(3.2.59)

in which the Cauchy boundary flux vector $\{q^e_c\}$ and the Cauchy boundary matrix $[V^e_c]$ from the normal velocity component are given by:

$$q^e_{ca} = -\int_{B^e_c} N^e_{a} q^e_{c} dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3$$

(3.2.60)

$$V^e_{ca\beta} = -\int_{B^e_c} N^e_{a} (n \cdot V) N^e_{\beta} dB; \quad \alpha, \beta = 1, 2, 3, 4 \quad \text{or} \quad \alpha, \beta = 1, 2, 3$$

(3.2.61)

Segments on which the variable boundary conditions are imposed are the flow-through boundaries, where the flow direction is not known a priori. When the flow is directed into the region, Cauchy boundary conditions will be used. The boundary-element column vector $\{B^e_v\}$ for a variable-boundary segment can be obtained similar to $\{B^e_c\}$:

$$\{B^e_v\} = \{q^e_v\} + [V^e_v]\{E\} - [V^e_v]\{(E^s) + (E^p)\}$$

(3.2.62)

in which the variable-boundary flux vector $\{q^e_v\}$ and the variable-boundary matrix $[V^e_v]$ from the normal velocity component are given by:

$$q^e_{va} = -\int_{B^e_v} N^e_{a} (n \cdot V) E^a dV; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3$$

(3.2.63)

$$V^e_{va\beta} = -\int_{B^e_v} N^e_{a} (n \cdot V) N^e_{\beta} dB; \quad \alpha, \beta = 1, 2, 3, 4 \quad \text{or} \quad \alpha, \beta = 1, 2, 3$$

(3.2.64)

where $E^a$ is the total dissolved concentration of the incoming fluid. When the flow is directed out from the region, both $\{q^e_v\}$ and $[V^e_v]$ are set equal to 0.

Assemble over all Neumann, Cauchy, and variable-boundary segments, we obtain the global boundary column vector $\{B\}$ as:

$$\{B\} = \{q\} + [V]\{E\} - [V]\{(E^s) + (E^p)\}$$

(3.2.65)

in which

$$\{q\} = \sum_{e \in N^e_n} \{q^e_n\} + \sum_{e \in N^e_c} \{q^e_c\} + \sum_{e \in N^e_v} \{q^e_v\} \quad \text{and} \quad [V] = \sum_{e \in N^e_n} [V^e_n] + \sum_{e \in N^e_c} [V^e_c] + \sum_{e \in N^e_v} [V^e_v]$$

(3.2.66)

where $N^e_n$ is the number of Neumann segments, $N^e_c$ is the number of Cauchy segments, and $N^e_v$ is the number of variable-boundary segments.
Substitute Eq. (3.2.65) into Eq. (3.2.43) and drop the subscript \((t + \Delta t)\) to simplify the notation, we obtain:

\[
[U]{E} = [W]\{E^*\} + \{E^p\} + \{R'\}
\]  

(3.2.67)

where \([U], [W],\) and \(\{R'\}\), after using Eqs. (3.2.44) through (3.2.49), are given by the following equations:

\[
[U] = [G] - \omega_v[V] \quad \text{and} \quad [W] = [H] - \omega_v[V]
\]  

(3.2.68)

and

\[
\{R'\} = \{L\} + \{R\} + \{q\} + (1 - \omega_v)[V]\{E^*\}
\]  

(3.2.69)

for conventional FEM approach, or

\[
\{R'\} = \{L\} + \{R\} + \{q\} + (1 - \omega_v)[V]\{E^{**}\}
\]  

(3.2.70)

for hybrid Lagrangian-Eulerian approach.

At nodes where Dirichlet boundary conditions are applied, an identity equation is generated for each node and included in the matrices of Eq. (3.2.67). The detailed method of applying this type of boundary condition can be found elsewhere (Wang and Connor, 1975).

Boundary conditions need to be implemented in the computation of the Lagrangian concentrations \(\{E^{**}\}\). Neumann boundary conditions normally apply to the boundary when flow is directed out from the region of interest. On the Neumann boundary, the backtracking would locate \(x_i^*\) in the interior of the domain; hence, the Lagrangian concentration at the i-th Neumann boundary node is simply computed via interpolation. Cauchy boundary conditions normally apply to the boundary segment when the flow is directed into the region of interest. The Lagrangian concentration on the i-th Cauchy node \(E_a^*\) is computed by the following formula:

\[
(E^*_{a})_{i}^* = \frac{\int_{B_e} N_i q_c dB}{\int_{B_e} N_i (n \cdot V) dB}
\]  

(3.2.71)

On the variable boundary, boundary conditions need not be implemented if the flow is directed out from the region. If the flow is directed into the region, then the Lagrangian concentration on the i-th variable boundary node \(E^{a}_{v_i}\) is computed by:

\[
(E^a_{v_{i}})^* = \frac{\int_{B_e} N_i n \cdot V E^{ain} dB}{\int_{B_e} N_i n \cdot V dB}
\]  

(3.2.72)

where \(E^{ain}\) is the concentration of the incoming fluid through the variable boundary. On the Dirichlet boundary nodes, the Lagrangian concentration is simply set to the specified value.
3.2.7. Solution of the Matrix Equations

Although both the matrices \([U]\) and \([W]\) are not functions of the unknown \(\{E\}\), Eq. (3.2.67) still represents a system of nonlinear algebraic equations because both \(\{E^r\}\) and \(\{E^p\}\) are functions of \(\{E\}\). The solution of this system requires some type of iterative procedure. The approach taken here is to make an initial estimate of the unknown \(\{E\}\). Using this estimate, we then compute the total sorbed concentration \(\{E^r\}\) and total precipitated concentration \(\{E^p\}\) from the biogeochemical reactions. Substituting the computed \(\{E^r\}\) and \(\{E^p\}\) vectors into the right-hand side of Eq. (3.2.67) and combining \([W]\)(\(\{E^r\}+\{E^p\}\)) with \(\{R^*\}\) to form a new right-hand side load vector, we linearize the matrix equation, which is then solved by the method of linear algebra to obtain the new solution \(\{E\}\). The new estimate is now obtained by the weighted average of the new solution and the previous estimate,

\[
\{E^{k+1}\} = \omega \{E\} + (1 - \omega) \{E^k\},
\]

where \(\{E^{k+1}\}\) = the new estimate, \(\{E^k\}\) = the previous estimate, \(\{E\}\) = the new solution, and \(\omega\) = the iteration parameter. The procedure is repeated until the new solution \(\{E\}\) is reached within a prescribed tolerance of error. When the iteration parameter is greater than or equal to 0 but is less than 1, the iteration is termed under-relaxation. If \(\omega = 1\), the method is the exact-relaxation. For the cases in which \(\omega\) is greater than 1 but less than or equal to 2, the iteration is termed over-relaxation.

Three options are employed to solve the linearized matrix equation: the block iteration method, the successive point iteration method, and the preconditioned conjugate gradient method (PCG). When these three methods are used, a double loop is required: the inner loop to solve the linearized equation and the outer loop to iterate the nonlinearity. Three choices can be made when the successive point iteration method is used to solve the linearized matrix equation. These are the successive under-relaxation (SUR), Gauss-Seidel (G-S), and successive over-relaxation (SOR) iteration methods. These three choices are unified by a relaxation parameter, \(\Omega\). When \(\Omega\) is less than 1 but greater than or equal to 0, the method is termed SUR iteration. When \(\Omega\) equals 1, the method is termed G-S iteration. If \(\Omega\) is greater than 1 but less than or equal to 2, the method is termed SOR iteration. When the PCG methods are used to solve the linearized equations, four alternative preconditioners are included in HYDROGEOCHEM 5.0: the polynomial, the incomplete Cholesky decomposition, the modified incomplete Cholesky decomposition, and the symmetrical over-relaxation.

3.3. Solution of Heat Transfer Equations

Similarly, heat transfer equations are solved by either the conventional Finite Element Method or the Lagrangian-Eulerian approach in HYDROGEOCHEM 5.0. In this report, there are three options for solving the transport equations. These three options are (1) application of the conventional FEM to the conservative form of equation, (2) Hybrid Lagrangian-Eulerian FEM, (3) Hybrid Lagrangian-Eulerian FEM for interior nodes + application of the conventional FEM to the advective form of equation for boundary nodes. They are briefly stated in the following three sections, respectively.

3.3.1. Solutions by Applying the Conventional FEM to the Conservative Form

For the Conventional FEM in Conservative Form, the governing equation is Eq. (2.3.6), as we have derived in Chapter 2:
Apply the weighted residual method to Equation (2.3.6) and integrate it in the spatial dimensions as follows:

\[ \int_R \left( C_{RT} \frac{\partial T}{\partial t} + C_{RH} T + \nabla \cdot (V_{CRT}) - \nabla \cdot (D \cdot \nabla T) \right) dR = \text{SSH} \tag{3.3.1} \]

Let \( T \) be approximated by:

\[ T = \sum_{j=1}^{N} T_j(0) N_j(x,y,z) \tag{3.3.2} \]

where \( N_j \) is the basis function of spatial coordinate for \( j \)-node, \( T_j \) is the value of temperature at node \( j \), and \( N \) is the number of finite-element nodes in the region.

Substitute Equation (3.3.2) and integrate Equation (3.3.1) by part, we obtain:

\[ \int_R \left[ \sum_{j=1}^{N} N_j C_{RT} N_j dR \right] \left( \frac{dT}{dt} \right) + \int_R \left[ \sum_{j=1}^{N} C_{RH} N_j dR \right] T + \int_R \left[ \sum_{j=1}^{N} \nabla W_i \cdot V_{CR} N_j dR \right] T + \int_R \sum_{j=1}^{N} \nabla N_j \cdot (D \cdot \nabla T) dR + \int_R N_j \text{SSH} dB \tag{3.3.3} \]

Equation (3.3.3) can be written in the matrix form as:

\[ [M] \left( \frac{dT}{dt} \right) + ([C] + [L] + [V])T = \{S\} + \{B\} \tag{3.3.4} \]

where the matrices \([M]\), \([C]\), \([L]\), and \([V]\) are given by:

\[ M_{ij} = \sum_{e \in M_e} \int_{R_e} N_{i}^e C_{RT} N_{j}^e dR \tag{3.3.5} \]

\[ C_{ij} = \sum_{e \in M_e} \int_{R_e} N_{i}^e C_{RH} N_{j}^e dR \tag{3.3.6} \]

\[ L_{ij} = \sum_{e \in M_e} \int_{R_e} \nabla N_{i}^e \cdot D \cdot \nabla N_{j}^e dR \tag{3.3.7} \]

\[ V_{ij} = -\sum_{e \in M_e} \int_{R_e} \nabla W_{i}^e \cdot V_{CR} N_{j}^e dR \tag{3.3.8} \]

where \( R_e \) is the region of element \( e \); \( M_e \) is the set of elements that have a local side \( \alpha-\beta \) coinciding with the global side \( i-j \); and \( N_{i}^e \) and \( N_{j}^e \) are the \( \alpha \)-th and \( \beta \)-th local base function of element \( e \), respectively; and \( W_{i}^e \)
is the $\alpha$-th local weighting function of element $e$.

In addition, the two load vectors, $\{S\}$ and $\{B\}$, are described as

$$S_i = \sum_{e \in M_e} \int_{R_e} N^e_\alpha SSH \, dR \quad (3.3.9)$$

$$B_i = -\sum_{e \in N_e} \int_{B_e} N^e_\alpha n \cdot (V_{CR} T - D^T \cdot \nabla T) \, dB \quad (3.3.10)$$

Similarly, with the same base and weighting functions, Equations (3.3.5) through (3.3.10) are evaluated as:

$$M^e_{\alpha\beta} = \int_{R_e} N^e_\alpha C_{RT} N^e_\beta \, dR \quad (3.3.11)$$

$$C^e_{\alpha\beta} = \int_{R_e} N^e_\alpha C_{RH} N^e_\beta \, dR \quad (3.3.12)$$

$$L^e_{\alpha\beta} = \int_{R_e} (\nabla N^e_\alpha) \cdot D^T \cdot (\nabla N^e_\beta) \, dR \quad (3.3.13)$$

$$V^e_{\alpha\beta} = \int_{R_e} (\nabla W_{\alpha}) \cdot V_{RT} N^e_\beta \, dR \quad (3.3.14)$$

and

$$S^e_{\alpha} = \int_{R_e} N^e_\alpha SSH \, dR \quad (3.3.15)$$

$$B^e_{\alpha} = -\int_{B_e} N^e_\alpha n \cdot (V_{CR} T - D^T \cdot \nabla T) \, dB \quad (3.3.16)$$

where the superscript or subscript $e$ denotes the element, $\alpha, \beta = 1, 2, .., 4$ for tetrahedral elements, $\alpha, \beta = 1, 2, .., 6$ for triangular prism elements, and $\alpha, \beta = 1, 2, .., 8$ for hexahedral elements.

### 3.3.2. Solutions by Applying the Conventional FEM to the Advective Form

For the conventional FEM in Advective Form, the governing equation is Eq. (2.3.11), as we have derived in Chapter 2:

$$C_{RT} \frac{\partial T}{\partial t} + C_{RQ} T + V_{CR} \cdot \nabla T - \nabla \cdot (D^T \cdot \nabla T) = C_{RQ} T^t \quad (2.3.11)$$

Approximate $T$ by a finite element interpolation as
\[ T = \sum_{j=1}^{N} T_j(t) N_j(x,y,z) \]  

(3.3.17)

Apply the weighted residual method to Eq. (2.3.13), the following ordinary differential equation can be obtained:

\[
\begin{align*}
\int_{R} N_i C_{RT} T_j \, dR \left( \frac{dT}{dt} \right) &+ \int_{R} N_i C_{RQ} N_j \, dR \{T\} + \int_{R} W_i V_{CR} \cdot \nabla N_j \, dR \{T\} + \\
\int_{R} \nabla N_i \cdot D \cdot \nabla N_j \, dR \{T\} &+ \int_{B} \nabla N_i \cdot D \cdot \nabla \mathbf{n} \, d\mathbf{n}
\end{align*}
\]

(3.3.18)

Equation (3.3.18) can be written in matrix form as:

\[
[M] \left( \frac{dT}{dt} \right) + \{V\} + \{D\} + \{C\} \{T\} = \{R\} + \{B\}
\]

(3.3.19)

where \{T\} is a vector composed of the temperature; \{dT/dt\} is a vector composed of total time derivatives of the temperature; [M] is the mass matrix related to the total derivative term; [V] is the matrix related to the velocity term; [D] is the stiff matrix related to the conduction term; [C] is the stiff matrix related to source/sink effect; \{R\} is the load vector of the temperature associated with the source/sink term; and \{B\} is the load vector associated with boundary conditions.

The matrices [M], [V], [D], [C], [R], and [B] are given by:

\[
M_{ij} = \sum_{e \in M_i} \int_{R_e} N^e_i C_{RT} N^e_j \, dR
\]

(3.3.20)

\[
V_{ij} = \sum_{e \in M_i} \int_{R_e} W^e_i V_{CR} \cdot \nabla N^e_j \, dR
\]

(3.3.21)

\[
D_{ij} = \sum_{e \in M_i} \int_{R_e} \nabla N^e_i \cdot \mathbf{D} \cdot \nabla N^e_j \, dR
\]

(3.3.22)

\[
C_{ij} = \sum_{e \in M_i} \int_{R_e} N^e_i C_{RQ} N^e_j \, dR
\]

(3.3.23)

and

\[
R_i = \sum_{e \in R_i} \int_{R_e} N^e_i C_{RQ} T \, dR
\]

(3.3.24)

\[
B_i = \sum_{e \in B_i} \int_{B_e} N^e_i \mathbf{n} \cdot (- \mathbf{D} \cdot \nabla T) \, d\mathbf{n}
\]

(3.3.25)
where $R_e$ is the region of element $e$, $M_e$ is the set of elements that have a local side $\alpha$-$\beta$ coinciding with the global side $i$-$j$, $N_{\alpha}^e$ is the $\alpha$-th local base function of element $e$, $N_{\beta}^e$ is the $\beta$-th local base function of element $e$ and $W_{\beta}^e$ is the $\alpha$-th local weighting function of element $e$.

Similarly, with the same base and weighting functions, Eqs. (3.3.20) through (3.3.25) are evaluated as:

\[ M_{\alpha\beta}^e = \int_{R_e} N_{\alpha}^e C_{RT} N_{\beta}^e \, dR \]  

\[ V_{\alpha\beta}^e = \int_{R_e} W_{\alpha}^e V_{CR} \cdot \nabla N_{\beta}^e \, dR \]  

\[ D_{\alpha\beta}^e = \int_{R_e} \left( \nabla N_{\alpha}^e \right) \cdot D^T \cdot \left( \nabla N_{\beta}^e \right) \, dR \]  

\[ C_{\alpha\beta}^e = \int_{R_e} N_{\alpha}^e C_{RQ} N_{\beta}^e \, dR \]  

and

\[ R_{\alpha}^e = \int_{R_e} N_{\alpha}^e C_{RQT} \, dR \]  

\[ B_{\alpha}^e = -\int_{B_e} N_{\alpha}^e \mathbf{n} \cdot ( -D^T \cdot \nabla T ) \, dB \]

where the superscript or subscript $e$ denotes the element, $\alpha, \beta = 1, 2, .., 4$ for Tetrahedral elements, $\alpha, \beta = 1, 2, .., 6$ for Triangular Prism elements, and $\alpha, \beta = 1, 2, .., 8$ for Hexahedral elements.

### 3.3.3. Solution by with the hybrid Lagrangian-Eulerian Finite Element Method

To overcome the inaccuracy caused by tracking time at the boundary node, we will first approximate the time derivative with finite difference method and then multiply the governing equation with the time interval. When the hybrid Lagrangian-Eulerian Finite Element Method is employed, the governing equation is Eq. (2.3.12), as we have derived in Chapter 2:

\[ C_{RT} \frac{D_T \mathbf{T}}{D\tau} + C_{RQ} \mathbf{T} = \nabla ( \mathbf{D}^T \cdot \nabla \mathbf{T} ) = C_{RQT} \mathbf{T} \]  

Approximate $T$ by a finite element interpolation as:

\[ T = \sum_{j=1}^{N} T_j(t) N_j(x,y,z) \]  

54
Apply the weighted residual method to Eq. (2.3.14), the following ordinary differential equation can be obtained:

\[
\left[ \int_R N_i C_{RT} N_j dR \right] \begin{Bmatrix} D_{VT} T \\ \frac{D}{Dt} T \end{Bmatrix} + \left[ \int_R N_i C_{RQ} N_j dR \right] T + \left[ \int_R \nabla N_i \cdot D^T \cdot \nabla N_j dR \right] T =
\int_R N_i C_{RQ} T \cdot dB + \int_B N_i n \cdot D^T \cdot \nabla T dB
\]

Equation (3.3.33) can be written in matrix form as:

\[
[M] \begin{Bmatrix} D_{VT} T \\ \frac{D}{Dt} T \end{Bmatrix} + ([D] + [C]) T = \{R\} + \{B\}
\]

where \( \{T\} \) is a vector composed of the temperature; \( \{D_{VT}/Dt\} \) is a vector composed of total time derivatives of the temperature; \( [M] \) is the mass matrix related to the total derivative term; \( [D] \) is the stiff matrix related to the conduction term; \( [C] \) is the stiff matrix related to source/sink effect; \( \{R\} \) is the load vector of the temperature associated with the source/sink term; and \( \{B\} \) is the load vector associated with boundary conditions.

The matrices \([M], [D], [C], [R], \) and \([B]\) are given by:

\[
M_{ij} = \sum_{e \in M_e} \int_{R_e} N^e_\alpha C_{RT} N^e_\beta dR
\]

\[
C_{ij} = \sum_{e \in M_e} \int_{R_e} N^e_\alpha C_{RQ} N^e_\beta dR
\]

\[
D_{ij} = \sum_{e \in M_e} \int_{R_e} \left( \nabla N^e_\alpha \right) \cdot D^T \cdot \nabla N^e_\beta dR
\]

and

\[
R_i = \sum_{e \in B_e \cap B_i} \int_{R_e} N^e_\alpha C_{RQ} T \cdot dB
\]

\[
B_i = - \sum_{e \in B_e \cap B_i} \int_{R_e} N^e_\alpha n \cdot (-D^T) \cdot \nabla T dB
\]

where \( R_e \) is the region of element \( e \), \( M_e \) is the set of elements that have a local side \( \alpha-\beta \) coinciding with the global side \( i-j \), \( N^e_\alpha \) is the \( \alpha \)-th local base function of element \( e \) and \( N^e_\beta \) is the \( \beta \)-th local base function of element \( e \).
Similarly, with the same base and weighting functions, Eqs. (3.3.35) through (3.3.39) are evaluated as:

\[
\begin{align*}
M_{\alpha\beta}^e &= \int_{R_e} N_\alpha^e C_{RT} N_\beta^e \, dR \\
C_{\alpha\beta}^e &= \int_{R_e} N_\alpha^e C_{RQ} N_\beta^e \, dR \\
D_{\alpha\beta}^e &= \int_{R_e} \left[ \nabla N_\alpha^e \right]^T \mathbf{D}^T \left[ \nabla N_\beta^e \right] \, dR \\
\end{align*}
\]

(3.3.40)

(3.3.41)

(3.3.42)

and

\[
\begin{align*}
R_\alpha^e &= \int_{R_e} N_\alpha^e C_{RQ} T_s \, dR \\
B_\alpha^e &= -\int_{B_e} N_\alpha^e \mathbf{n} \cdot (-\mathbf{D}^T) \cdot \nabla T \, dB
\end{align*}
\]

(3.3.43)

(3.3.44)

where the superscript or subscript \( e \) denotes the element, \( \alpha, \beta = 1, 2, \ldots, 4 \) for Tetrahedral elements, \( \alpha, \beta = 1, 2, \ldots, 6 \) for Triangular Prism elements, and \( \alpha, \beta = 1, 2, \ldots, 8 \) for Hexahedral elements.

3.3.4. Numerical Implementation of Boundary Conditions

To incorporate the boundary conditions, Eq. (3.3.10) derived from conservative form and Eq. (3.3.44) derived from advective form should be evaluated. Applying weighted residual method to Neumann, Cauchy or variable boundary conditions yields the expression for the global boundary column vector \( \{ \mathbf{B} \} \):

\[
\{ \mathbf{B} \} = \{ \mathbf{q} \} + [\mathbf{V}] \{ \mathbf{T} \}
\]

(3.3.45)

where \([\mathbf{V}]\) and \([\mathbf{q}]\) are different when different approaches are used.

Application of the conventional FEM to the Conservative Form

\[
\{ \mathbf{q} \} = \sum_{e \in N_{ne}} \{ q_{Te}^e \} + \sum_{e \in N_{ce}} \{ q_{Te}^e \} + \sum_{e \in N_{ve}} \{ q_{Tv}^e \}
\]

(3.3.46)

\[
[\mathbf{V}] = \sum_{e \in N_{ne}} [V^e_{\alpha}] + \sum_{e \in N_{ve}} [V^e_{Tv}]
\]

(3.3.47)

in which \( N_{ne}, N_{ce}, N_{ve} \) are the number of Neumann, Cauchy, and Variable boundary segments, respectively.

For Neumann Boundary conditions, \( \{ q_{Te}^e \} \) is the Neumann boundary flux vector, and \([V_{Tv}^e]\) is the Neumann boundary matrix from the normal velocity component, and they are given by:
\[ q_{Tn}^{e} = - \int_{B_e}^{e} N_a^e q_{Tn} \, dB; \quad \alpha = 1, 2, 3, 4 \text{ or } \alpha = 1, 2, 3 \]

(3.3.48)

and

\[ V_{n\beta}^{e} = - \int_{B_e}^{e} (n \cdot V_{CR}) N_{\beta}^{e} \, dB; \quad \alpha, \beta = 1, 2, 3, 4 \text{ or } \alpha, \beta = 1, 2, 3 \]

(3.3.49)

For Cauchy Boundary conditions, \( \{q_{Tc}^{e}\} \) is the Cauchy boundary flux vector given by:

\[ q_{Tc}^{e} = - \int_{B_e}^{e} N_a^e q_{Tc} \, dB; \quad \alpha = 1, 2, 3, 4 \text{ or } \alpha = 1, 2, 3 \]

(3.3.50)

For variable boundary conditions, \( \{q_{Tv}^{e}\} \) is the Neumann boundary flux vector, and \( [V_v^{e}] \) is the Neumann boundary matrix from the normal velocity component, and they are given by:

If \( V_p \cdot n > 0 \),

\[ V_{v\beta}^{e} = - \int_{B_e}^{e} (n \cdot V_{CR}) N_{\beta}^{e} \, dB; \quad \alpha, \beta = 1, 2, 3, 4 \text{ or } \alpha, \beta = 1, 2, 3 \]

(3.3.51)

If \( V_p \cdot n < 0 \),

\[ q_{Tv}^{e} = - \int_{B_e}^{e} (n \cdot V_{CR}) T_v \, dB; \quad \alpha = 1, 2, 3, 4 \text{ or } \alpha = 1, 2, 3 \]

(3.3.52)

where \( T_v \) is the temperature in the incoming fluid.

**Application of the Conventional FEM to the Advective Form**

\[ \{q\} = \sum_{e \in N_{ce}} \{q_{Tn}^{e}\} + \sum_{e \in N_{ce}} \{q_{Tc}^{e}\} + \sum_{e \in N_{ve}} \{q_{Tv}^{e}\} \]

(3.3.53)

\[ [V] = \sum_{e \in N_{ce}} [V_{c}^{e}] + \sum_{e \in N_{ve}} [V_{v}^{e}] \]

(3.3.54)

in which \( N_{ne}, N_{ce}, N_{ve} \) are the number of Neumann, Cauchy, and Variable boundary segments, respectively.

For Neumann Boundary conditions, \( \{q_{Tn}^{e}\} \) is the Neumann boundary flux vector given by:

\[ q_{Tn}^{e} = - \int_{B_e}^{e} N_a^e q_{Tn} \, dB; \quad \alpha = 1, 2, 3, 4 \text{ or } \alpha = 1, 2, 3 \]

(3.3.55)

For the Cauchy boundary conditions, \( \{q_{Tc}^{e}\} \) is the Cauchy boundary flux vector, and \( [V_{c}^{e}] \) is the Cauchy boundary matrix from the normal velocity component, and they are given by:
\[ q_{\text{Tca}}^e = - \int_{B_e} N_a q_{\text{Tc}} dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \] (3.3.56)

and

\[ V_{\text{ca}\beta}^e = - \int_{B_e} N_a \langle n \cdot V_{\text{CR}} \rangle N_{\beta}^e dB; \quad \alpha, \beta = 1, 2, 3, 4 \quad \text{or} \quad \alpha, \beta = 1, 2, 3 \] (3.3.57)

For Variable boundary conditions, segments on which the variable boundary conditions are imposed are the flow-through boundaries and the flow direction is not known as a priori. Two situations are to be considered. When the flow is directed into the region, Cauchy boundary conditions will be applied. \{q_{\text{Tva}}^e\} is the variable boundary flux vector, and \[V_{\text{va}\beta}^e\] is the variable boundary matrix from the normal velocity component, and they are given by:

\[ q_{\text{Tva}}^e = - \int_{B_e} N_a \langle n \cdot V_{\text{CR}} \rangle T_v dB; \quad \alpha = 1, 2, 3, 4 \quad \text{or} \quad \alpha = 1, 2, 3 \] (3.3.58)

\[ V_{\text{va}\beta}^e = - \int_{B_e} N_a \langle n \cdot V_{\text{CR}} \rangle N_{\beta}^e dB; \quad \alpha, \beta = 1, 2, 3, 4 \quad \text{or} \quad \alpha, \beta = 1, 2, 3 \] (3.3.59)

where \( T_v \) is the temperature in the incoming fluid. When the flow is directed out from the region, both the variable-boundary flux vector and the variable-boundary matrix from the normal velocity component are set equal to 0.

3.3.5. Finite Difference Approximation in Time

When the Lagrangian-Eulerian approach is used in the simulation, the time weighted difference is applied to Eq. (3.3.34) and yields:

\[ \frac{[M]}{\Delta t} \{ T^{t+\Delta t} - T^* \} + \omega ([D] + [C]) \{ T^{t+\Delta t} \} + (1 - \omega) ([D] + [C]) \{ T_t \} = \{ R \} + \{ B \} \] (3.3.60)

where \( \Delta t \) is the Lagrangian tracking time; \( \{ T^{t+\Delta t} \} \) is the vector containing the temperature at all nodes at new time \( t+\Delta t \); \( \{ T_t \} \) is the vector containing the temperature at all nodes at previous time \( t \); \( \{ T^* \} \) is the Lagrangian temperature vector and it is computed by the backward tracking method of characteristics as follows:

\[ x_i^* = x_i - \int_t^{t+\Delta t} V_T dt \quad \text{and} \quad T_i^* = \sum_j T_j(t) N_j(x_i^*) dt \] (3.3.61)

where \( x_i^* \) (the Lagrangian point) is the location of the fictitious particle originating at time \( t+\Delta t \) that would arrive at node \( i \) at time \( t \); \( T_j(t) \) is the temperature at node \( j \) at time \( t \); \( T_i^* \) is the Lagrangian temperature at node \( i \); \( N_j(x_i^*) \) is the base function of node \( j \) evaluated at the Lagrangian point \( x_i^* \).
4. THE HYDROGEOCHEM 5.0 PROGRAM STRUCTURE

HYDROGEOCHEM 5.0 is designed to solve a system of equations describing fluid flows, hydrologic transport, biogeochemical reactions, and heat transfer in a reactive multicomponent system under saturated-unsaturated conditions. The fluid flow equations are derived from the continuity of fluid mass and Darcy's law for volume flux. The transport equations are derived from the continuity of mass and Fick’s law of mass flux. The biogeochemical equations are mole balances coupled with the mass action and kinetic rate equations. The heat transfer equation is derived based on the principle of conservation of energy and Fourier’s law of thermal flux. The Richards equation along with initial and boundary conditions govern the fluid flow in variably saturated media. The major factors controlling fluid flows are slight deformation of the media, pressure and gravity forces, capillary tension, evaporation, transpiration, infiltration, and artificial injections and pumping. The transport equations, along with initial and boundary conditions and the biogeochemical equations, govern the migration and biogeochemical transformation of multicomponent species in saturated-unsaturated media. The major transport processes are advection, dispersion/diffusion, and source/sinks. The major biogeochemical processes are aqueous complexation, adsorption, ion-exchange, precipitation/dissolution, redox, acid-base reactions, and microbial-mediated biochemical reactions. The heat transfer equation, along with initial and boundary conditions and Van't Hoff equation, govern the thermal transport in variably saturated media. The main factors that affect the heat transfer are standard reaction enthalpy, temperature, specific heats of the fluid and the dry medium, artificial heat source/sink, and heat source due to reactions.

4.1. The General Solution Strategy of HYDROGEOCHEM 5.0

This section describes the numerical strategies involved in HYDROGEOCHEM 5.0. These 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, reactive biogeochemical transport as well as heat transfer, and (4) the strategy for coupling hydrologic transport and biogeochemical reactions.

4.1.1. Steady-State Simulation with Strong Coupling

The strong coupling of subsurface flow, reactive biogeochemical transport and heat transfer is considered for steady-state simulations in HYDROGEOCHEM 5.0. As mentioned in chapter 2, a strong coupling is needed because (1) the groundwater density and dynamic viscosity are functions of dissolved concentrations of biogeochemical species; (2) the equilibrium constants of biogeochemical reactions are temperature-dependent; and (3) the subsurface flow velocity affects the temporal-spatial distributions of biogeochemical concentrations. Therefore, all the governing equations should, in general, be solved simultaneously.

Fig. 4.1 shows how strong coupling is implemented in the computer codes. During a coupling iteration, the governing equations are solved in the order of subsurface flow, heat transfer, and biogeochemical transport. The new iteration values of the variables of interest are computed based on the working iteration values of the variables of interest that are determined by relaxing the previous 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 Fig. 4.1, the working iteration values are updated after all the governing equations are solved and the nonconvergence is detected.
These are explained in details in the following steps:

- **Step 0:** Provide the pre-initial condition as the values of the zeroth iteration (or the initial guess) to begin the steady-state simulation.

- **Step 1:** The subsurface flow governing equation is solved based on the working values of system variables (e.g., pressure head, temperature and biogeochemical concentrations) from the previous iteration. The distribution of pressure head in the domain of interest is then obtained. The distributions of moisture content, water capacity, and groundwater velocity are calculated based on the currently computed pressure head.

- **Step 2:** The computed pressure head is compared with that from the previous iteration to examine convergency. All the variables computed in the subsurface flow module are applied to computing heat transfer and biogeochemical transport in the current iteration (i.e., the subsurface flow variables are updated).

- **Step 3:** The steady-state version of the heat transfer governing equation is solved next to obtain the distribution of temperature in the domain of interest.

- **Step 4:** Update the previous iteration temperature with the currently computed value in Step 3, the latter is to be used in modifying the equilibrium constants for the computation of biogeochemical transport in the current iteration (i.e., temperature is updated).

- **Step 5:** The steady-state governing equations of reactive biogeochemical transport are solved for the distribution of total analytical component concentrations as well as the total analytical concentrations of kinetic variables in the domain of interest.

- **Step 6:** The currently computed and previous iteration values are compared for checking convergency.

- **Step 7:** At the end of each iteration, determine if another iteration is needed. If convergency has been reached for both flow and transport 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 update. A steady-state simulation is finished when a convergent solution is reached.

Figure 4.1 shows the complete structure of the strong coupling loop, with the above seven steps executed in the model. The structure is designed for enhancing the convergency associated with the coupling loop.
Prepare for steady-state simulations

Start a new coupling iteration

Implement the steady-state version of the fluid flow module

Check the convergence of fluid flow

Update subsurface flow

Implement the steady-state version of the heat transfer module

Update heat transfer

Implement the steady-state version of the chemical transport module

Check the convergence of chemical transport

Update chemical transport

Convergence for all?

Yes
End steady-state simulations

No

The coupling loop

Fig. 4.1. The Numerical Strategy for Steady State Simulations
4.1.2. Transient Simulation With Strong Coupling

The strong coupling of subsurface flow, reactive biogeochemical transport, and heat transfer is also used for transient simulations in HYDROGEOCHEM 5.0. Figure 4.2 shows how the strong coupling is implemented in the computer code.

For each time step, a coupling loop is implemented. Initial guesses must be made for the coupling loop for each time step. For the first time step, the initial guesses are obtained either from the pre-initial conditions when the steady-state simulation is not desired or from the steady-state simulations. From the second time step on, the initial guesses are obtained from the previous time step. In each time step, the coupling is performed similar to that in the steady-state simulation. The transient version of subsurface flow governing equation is first solved based on the working values of system variables (e.g., pressure head and biogeochemical concentrations) from the previous iteration. The distribution of pressure head in the domain of interest is obtained. The distributions of moisture content, water capacity, and groundwater velocity are calculated 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 biogeochemical transport in the current iteration (i.e., the subsurface flow variables are updated). The transient version of the heat transfer governing equation is solved next to obtain the distribution of temperature in the domain of interest. The currently computed temperature is used in modifying the equilibrium constants for the computation of biogeochemical transport coming up in the current iteration (i.e., temperature is updated). The transient governing equations of biogeochemical transport are then solved for the distribution of the total analytical component concentrations as well as the total analytical concentrations of kinetic variables in the domain of interest. The currently computed and previous iteration values are then compared for convergency.

At the end of each iteration, overall convergency is checked. If both the flow and biogeochemical transport modules are convergent for the current iteration, then the convergent solution represents the solution for the current time step. 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 guessed values (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 geochemical reactions, so that the total dissolved component concentrations can be obtained to calculate the estimated fluid density and viscosity for the next iteration. In the meantime, the concentrations of all precipitated species are used to compute the effective porosity, effective moisture content, and hydraulic conductivity. The solution at the present time step is completed when a convergent solution is reached. When a convergent solution for the coupling loop is achieved, a check is made to see if the number of time steps or the maximum simulation time is exceeded. If either condition is satisfied, the transient simulation is terminated. If both conditions are not satisfied, the flow, heat and biogeochemical variables are updated by replacing the values of the previous time step with the new solutions. These new solutions serve as the initial conditions for the next time-step simulation.
Fig 4.2. The Numerical Strategy for Transient Simulations
4.1.3. The Strategy for Solving Fluid Flows

In HYDROGEOCHEM 5.0, the subsurface flow module is basically the modified version of the 3DFEMWATER model (Yeh, 1987b) where neither the density nor precipitation effect is taken into account. The effect of density and precipitation/dissolution on fluid flows is described in Chapter 2. In addition to the Dirichlet, Neumann, and Cauchy boundary conditions, 3DFEMWATER 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 2. Since the variable boundary condition is not determined a priori, it needs to be correctly addressed by using the technique of cyclic iteration. Only when the variable boundary conditions for two consecutive cycles are exactly the same, a correct boundary condition and its associated convergent solution represent the situation being considered. Figure 4.3 illustrates the flow chart of how the subsurface flow equation is solved numerically in HYDROGEOCHEM 5.0.

In Fig. 4.3, two iteration loops are included: one to address the variable boundary condition, the other one to deal with the nonlinearity of unsaturated media. During a variable-boundary-condition iteration cycle, the boundary condition for the flow-in/flow-out boundary is guessed based on either the given initial 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 boundary condition is the same as that of the previous cycle, then boundary-condition consistency is reached and the convergent solution from the nonlinear iteration loop is the numerical solution for the time step. Otherwise, the boundary condition needs to be updated for use in the next cycle. Both iterations continue until consistent/convergent results are obtained.
Prepare the initial guess of variable boundary conditions based on either the initial guess or the initial condition given for simulations

Start a new variable-boundary-condition cycle

Start a new nonlinear iteration

Calculate soil properties and moisture content and fluid density and viscosity based on the working values of pressure head from nonlinear loop and chemical concentrations from coupling loop

Compose the linearized matrix equations

Solve the linearized matrix equations

Check the convergence for the nonlinear iteration loop

If not convergent
Update the working values of pressure head
Go to the next nonlinear iteration

Check the consistence of the variable boundary conditions
If not consistent
Update boundary conditions applied on variable boundary sides
Go to the next variable-boundary-condition cycle

If convergent
End the simulation of subsurface flow

Fig 4.3. The Strategy for Solving Fluid Flow
4.2 Description of HYDROGEOCHEM 5.0 Subroutines

HYDROGEOCHEM 5.0 consists of a short MAIN routine, 191 subroutines, and 3 functions (EQ, DOTPRD, and FCOS). The MAIN module is used to specify the sizes of all arrays, read data file names, and open data files. The control and coordination activities are performed with Subroutine HGC50 where the coupling of flow, heat transfer and biogeochem transport is also handled. The fluid flow are solved in Subroutines HYDROS and HYDROT for the steady state and transient simulations, respectively. The heat transfer are solved in Subroutines HEATS and HEATT for the steady state and transient simulations, respectively. The linkage between the hydrologic transport model and reactive biogeochemical model is performed with the Subroutines CHEMIS and CHEMIT for the steady state and transient simulations, respectively. The remaining 184 subroutines can be classified into seven major categories:

- 14 subroutines are used to perform fluid flow: RHOMU0, SPROP, VELT, FQ468D, FBCMTRX, FSFLOWPRE, FSFLOW, FQ468TH, BCPREP, FASEMB, FQ468, FBC, FQ34R, FQ34S.
- 18 subroutines and one function are used to perform hydrologic transport: SROSTR, GRDROV, AFABTA, DISPC, MVELT, Q468D, SFLOW, Q34S, TACADC, WKTOTC, ASEMBL, Q468, TRANSP, EQNGEN_F, EQNGEN_P, BC, Q34CNV, CHKVBC.
- 15 subroutines and one function (EQ) are used to perform heat transfer: DISPT, LANGVT, TASEMB, Q468T, KMATRIX, MATRIX, VMATRIX, VMATRIX1, TBC, Q34TB, TBC_1, Q34TB_1, ENTHA, ENTHABF, EKCONST.
- 11 subroutines and one function (DOTPRD) are to solve matrix equations for flow / transport /heat transfer simulations: BLKITR, SOLVE, PISS, PPCG, POLYP, ILUCG, LILTINV, MICPCG, MICP, SSORCG, SSORP, DOTPRD.
- 44 subroutines and one function (FCOS) are used to compute the Lagrangian particle tracking: GNTRAK, REPLAS, ELENOD, WRKARY, ELTRK4, ELTRK6, ELTRK8, MMLOC, ONLINE, TRAK1T, TRAK2T, TRAK1P, TRAK2P, TRAK1H, TRAK2H, PLANEW, LOCQ3N, LOCQ4N, CHNGSN, SURE3D, LOCPLN, BASE2D, XSI2D, NEWXE, CKSIDE, CKCNEL, CKCOIN, ONPLAN, BDYPLN, BASE1, FIXCK, ALGBDY, INTERP1, BNDRY, LOCQ2N, SUR2D3, BASE, XSI3D, XSI3DP, CKBDY, ADVBC, Q34ADV, TADVBC, Q34TADV, FCOS.
- 46 subroutines are used to simulate biochemical equilibrium and kinetics: OCSPIT, ARRVA, DERIV, LUDCMP, FUNCT, JACKR, BIOGEOCHEM, KIVRAT, KINEQL, TOTDNP, NPPT, MSP2EQ, ACOEF, MODIFK, MODBFK, RESIDU, RADC, USERMA, RIES, RKRXNS, RUSRLW, RATEL0, PRODBF, JACOB, JADC, JACKV1, JACKV2, JARXN, FTERM, JOLAW, DERDRXN, DERXN, DERDEQ, DEREQ, DGELG, INDEXX, DISOLV, BASWIT, INVERSE, PREPOS, ERANK1, ERANK, KRANK, EKRANK, CPEQ, SPTYP.
- 36 subroutines are to perform utility functions: RDATIO, DATAIO, READR, READN, CONECT, SURF, PAGEN, LRL3D, LRN3D, FIDAT, IDAT, TIDAT, FSSDAT, SSDAT, TSSDAT, FBCDAT, BCDAT, TCBDAT, NODVA0, NODVA1, NODVA2, NODVA3, INTERPADV3D, CALKD, FPRINT, PRINTT, TPRINT, FSTORE, STORE, TSTORE, LPOUT, TOTAL0, TOTAL1, TOTAL2, SHAPE.
Fig 4.4 shows the structure of the program. The subroutines are described below.

Fig. 4.4 Program Structure of HYDROGEOCHEM 5.0: 1 of 5
Fig. 4.4. Program Structure of HYDROGEOCHEM 5.0: 2of 5
Fig. 4.4., Program Structure of HYDROGEOCHEM 5.0: 3 of 5
Fig 4.4. Program Structure of HYDROGEOCHEM 5.0: 5 of 5
PROGRAM MAIN

Before calling the Subroutine HGC50 to perform simulations, the MAIN routine calls the subroutines, as shown in the program structure in Fig 4.4, to read and print required information as well as to prepare some utility functions for computations.

- Subroutine RDATIO - to read and print input data required for defining the domain of interest and perform decomposition of reaction network.
- Subroutine PAGEN - to generate point arrays to store the global matrix in compressed form and to construct the sub-regional block matrices.
- Subroutine CONECT - to rearrange the pointer array of node connections in the ascending order and to identify the location of the diagonal entity for each node.
- Subroutine SURF - to identify the boundary sides, sequences the boundary nodes, and compute the directional cosine of each boundary side.
- Subroutine FIDAT - to read and print initial conditions required for flow computations.
- Subroutine IDAT - to read and print initial conditions required for computing the reactive biogeochemical transport.
- Subroutine TIDAT - to read and print initial conditions required for computing the heat transfer.
- Subroutine FSSDAT - to read and print source/sink information for flow computations.
- Subroutine SSDAT - to read and print source/sink information for computing the reactive biogeochemical transport.
- Subroutine TSSDAT - to read and print source/sink information for computing the heat transfer.
- Subroutine FBCDAT - to read and print boundary conditions required for flow computations.
- Subroutine BCDAT - to read and print boundary conditions required for computing the reactive biogeochemical transport.
- Subroutine TBCDAT - to read and print boundary conditions required for computing the heat transfer.
- Subroutine NODVA0 - to evaluate the fluid density and specific heat of dry medium at each node for transport and heat transfer simulations.
- Subroutine NODVA1 - to evaluate the reference fluid density and the media bulk density at
each node for coupled flow, heat transfer and transport simulations.

- Subroutine NODVA2 - to evaluate the moisture content and its time derivative at each node for coupled flow, heat transfer and transport simulations.

- Subroutine NODVA3 - to evaluate the ion-exchange capacity, the surface area of the adsorbing site, the capacitance between the surface and “o” plane, and the capacitance between the “o” plane and “b” plane at each node for biogeochemical calculations.

- Subroutine ENTHA to - calculate the enthalpy of equilibrium reactions.

- Subroutine ENTHABF - to calculate the enthalpy of kinetic reactions.

**SUBROUTINES**

1. **Subroutine HGC50**: Subroutine HGC50 controls the main sequence of operations, a function generally performed by the MAIN program. It is preferable, however, to keep a short MAIN routine and several subroutines with variable storage allocation. This makes it possible to deal with a site-specific problem without making changes in array dimensions throughout all subroutines.

Subroutine HGC50 will perform

- The flow only simulations (IMID = 0, ITEMP = 0 and IMIDF = 1).

- The transport only simulations (IMID = 1, ITEMP = 0 and IMIDF = 0).

- The coupled heat transfer and transport simulations (IMID = 1, ITEMP = 1 and IMIDF = 0).

- The coupled flow and transport simulations (IMID = 1, ITEMP = 0 and IMIDF = 1).

- The coupled flow, heat transfer and transport simulations (IMID = 1, ITEMP = 1 and IMIDF = 1).

For each of the above described simulations, the following three options can be chosen for computations

- The steady-state solutions (NTI = 0 and KSSF and/or KSS = 0 for flow and/or transport).

- The transient solutions using the steady-state solution as the initial conditions (NTI > 0 and KSSF and/or KSS = 0 for flow and/or transport).

- The transient solutions using user-supplied initial conditions (NTI > 0 and KSSF and/or KSS = 1 for flow and/or transport).

Subroutine HGC50 calls:

- Subroutine HYDROS to perform steady-state flow simulations.
• Subroutine HYDROT to perform transient flow simulations.
• Subroutine HEATS to perform steady-state heat transfer simulations.
• Subroutine HEATT to perform transient heat transfer simulations.
• Subroutine CHEMIS to perform steady-state reactive transport simulations.
• Subroutine CHEMIT to perform transient reactive transport simulations.
• Subroutine OCSPIT to initialize the system to biogeochemical equilibrium if the steady-state solutions of the reactive biogeochemical transport are desired.
• Subroutine INTERP to obtain sources/sinks and boundary values.
• Subroutine RHOMU0 to compute the ratio of fluid density and dynamic viscosity over the reference fluid density and dynamic viscosity at Gaussian points of each element, respectively.
• Subroutine SROSTR to compute the ratio of fluid density over its reference value for each element and well source/sink.
• Subroutine GRDROV to compute the gradient term of the ratio of fluid density over its reference value at the Gaussian points of each element.
• Subroutine SPROP to compute the moisture content, the relative hydraulic conductivity, and the water capacity.
• Subroutine VELT to compute the Darcy's velocities.
• Subroutine AFABTA to compute the upstream weighting factor.
• Subroutine DISPC to compute the hydrodynamic dispersion coefficients of transport.
• Subroutine DISPT to compute the thermal conductivity of heat transfer.
• Subroutine MVELT to compute the material flux.
• Subroutine NODVA2 to evaluate the moisture content and its time derivative at each node for coupled flow, heat transfer and transport simulations.
• Subroutine ADVW3D to prepare working arrays for Lagrangian integration.
• Subroutine CALKD to compute equivalent K_d values.
• Subroutine FSFLOWPRE to compute the boundary flux based on pressure head.
• Subroutine FSFLOW to compute the net rates of flow through open boundaries.
• Subroutine SFLOW to compute the net rates of biogeochemicals through open boundaries.
• Subroutine FPRINT to print the flow results.
• Subroutine PRINTT to print the transport results.
• Subroutine TPRINT to print the heat transfer results.
• Subroutine FSTORE to store the flow results for plotting.
• Subroutine STORE to store the transport results for plotting.
• Subroutine TSTORE to store the heat transfer results for plotting.

It also calls Subroutines TOTAL0, TOTAL1, TOTAL2,

2. **Subroutine RDATIO**: Subroutine RDATIO, called by the MAIN route, reads and prints Data Sets 5 through 9 as described in Appendix A. It calls Subroutines READR and READN to automatically generate real and integer numbers, respectively. It also calls Subroutine DATAIO to input transport and biogeochemical reaction information.

3. **Subroutine DATAIO**: This subroutine, called by the Subroutine RDATIO, reads and print Data Sets 10 through 16 as described in Appendix A.

4. **Subroutine READR**: This subroutine is called by Subroutines RDATIO, FIDAT, IDAT, TIDAT, and FBCDAT to generate real numbers for Data Sets 7, 19, 22, 28 and 35 as described in Appendix A. Automatic generation of regularly patterned data is built into this subroutine.

5. **Subroutine READN**: This subroutine is called by Subroutine RDATIO, FSSDAT, SSDAT, TSSDAT, FBCDAT, BCDAT and TBCDAT to generate integers for Data Sets 9, 20, 21, 22, 23, 24, 25, 26, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40 and 41 as described in Appendix A. Automatic generation of regularly patterned data is built into this subroutine.

6. **Subroutine SURF**: Subroutine SURF identifies the boundary sides, sequences the boundary nodes, and computes the directional cosine of the surface sides. The mapping from boundary nodes to global nodes are stored in NPBB(I) (where NPBB(I) is the global node number of I-th boundary node). The boundary node numbers of the four nodes for each boundary side are stored in ISB(I,J) (where ISB(I,J) is the boundary node number of I-th node of J-th side, I = 1 to 4). There are six sides for each element. Which of these six sides is the boundary side is determined automatically in the Subroutine SURF and is stored in ISB(5,J). The global element number, to which the J-th boundary side belongs, is also preprocessed in the Subroutine SURF and is stored in ISB(6,J). The directional cosines of the J-th boundary side are computed and stored in DCOSB(I,J) (where DCOSB(I,J) is the directional cosine of the J-th surface with I-th coordinate, I = 1 to 3). The information contained in NPBB, ISB, and DCOSB, along with the number of boundary nodes and the number of boundary sides, is returned to routine MAIN for use in other subroutines.

7. **Subroutine PAGEN**: Subroutine PAGEN generates pre-process pointer arrays that are needed to store the global matrix in compressed form and to construct the sub-regional block matrices. The pointer arrays automatically generated in this subroutine include the global node connectivity (stencil) LRN(J,N), global
elements connected to nodes $\text{LRL}(J,N)$, regional node connectivity $\text{LNOJC}(J,I,K)$, total node number for each subregion $\text{NTNPLR}(K)$, bandwidth indicator for each subregion $\text{LMAXDF}(K)$, and partial fill-up for the mapping array between global node number and local subregion node number $\text{GNPLR}(I,K)$ with $I = \text{NNPLR}(K) + 1$ to $\text{NTNPLR}(K)$. Here $\text{LRN}(J,N)$ is the global node number of $J$-th node connected to the global node $N$; $\text{LRL}(J,N)$ is the global element number of the $J$-th elements connected to the global node $N$, $\text{LNOJC}(J,I,K)$ is the local node number of the $J$-th node connected to the local node $I$ in $K$-th subregion; $\text{NTNPLR}(K)$ is the total number of nodes in the $K$-th subregion, including the interior nodes, the global boundary nodes, and intra-boundary nodes; $\text{LMAXDF}(K)$ is the maximum difference between any two nodes of any element in $K$-th subregion; and $\text{GNPLR}(I,K)$ is the global node number of $I$-th local-region node in the $K$-th subregion. These pointer arrays are generated based on the element connectivity $\text{IE}(M,J)$, the number of node for each subregion $\text{NNPLR}(K)$, and the mapping between global node and local-region node $\text{GNLR}(I,K)$ with $I = 1$, $\text{NNPLR}(K)$. Here $\text{IE}(M,J)$ is the global node number of $J$-th node of element $M$; $\text{NNPLR}(K)$ is the number of nodes in the $K$-th subregion including the interior nodes and the global boundary nodes but not the intra-boundary nodes. If a preconditioned conjugate gradient solver is selected to solve linear matrix equations, this subroutine will re-arrange $\text{LRN}(J,N)$ such that $\text{LRN}(1,N) = N$.

8. **Subroutine LRL3D**: This subroutine is called by Subroutines PAGEN and ADVW3D. This subroutine generates the node-element connection pointer arrays $\text{NLRL}$, $\text{LRL}$, $\text{NLRLW}$, and $\text{LRLW}$ used in particle tracking.

9. **Subroutine LRN3D**: This subroutine is called by Subroutine PAGEN. This subroutine generates the node-node connection pointer arrays $\text{NLRN}$ and $\text{LRN}$ used in composing the linearized matrix equations.

10. **Subroutine CONECT**: This subroutine is called by the MAIN program to rearrange the point array of node connectivity $\text{LRN}(J,N)$ according to the ascending order of nodes connecting to the node $N$ and to store the diagonal entity of $\text{LRN}(J,N)$ in the array $\text{ND}(N)$.

11. **Subroutine FIDAT**: This subroutine is called by the MAIN route to read and print the flow initial conditions via the input file assigned for logic unit 16 or the binary files assigned for logic units 11 and 14.

12. **Subroutine IDAT**: This subroutine is called by the MAIN route to read and print the transport initial conditions via the input file assigned for logic unit 16 or the binary files assigned for logic units 12 and 13.

13. **Subroutine TIDAT**: This subroutine is called by the MAIN route to read and print the heat transfer initial conditions via the input file assigned for logic unit 16 or the binary files assigned for logic units 22 and 23.

14. **Subroutine FSSDAT**: This subroutine is called by the MAIN route to read and print the element and well source/sink for flow simulations as described in Data Sets 20 and 21 in Appendix A.

15. **Subroutine SSDAT**: This subroutine is called by the MAIN route to read and print the element and well source/sink for transport simulations as described in Data Sets 29 and 30 in Appendix A.

16. **Subroutine TSSDAT**: This subroutine is called by the MAIN route to read and print the element and well source/sink for heat transfer simulations as described in Data Sets 36 and 37 in Appendix A.

17. **Subroutine FBCDAT**: This subroutine is called by the MAIN route to read and print the boundary conditions for flow simulations as described in Data Sets 22 through 26 in Appendix A.
18. **Subroutine BCDAT**: This subroutine is called by the MAIN route to read and print the boundary conditions for transport simulations as described in Data Sets 31 through 34 in Appendix A.

19. **Subroutine TBCDAT**: This subroutine is called by the MAIN route to read and print the boundary conditions for heat transfer simulations as described in Data Sets 38 through 41 in Appendix A.

20. **Subroutine NODVA0**: This subroutine is called by MAIN to convert the values of fluid density and specific heat capacity at elements into those at nodal points.

21. **Subroutine NODVA1**: This subroutine is called by MAIN to convert the values of reference fluid density and media bulk density at elements into those at nodal points.

22. **Subroutine NODVA2**: This subroutine is called by MAIN and HGC50 to convert the values of the moisture content and its time derivative at elements into those at nodal points.

23. **Subroutine NODVA3**: This subroutine is called by MAIN to convert the values of the initial values of exchange sites and the surface area and capacitance of adsorbing sites at elements into those at nodal points.

24. **Subroutine ENTHA**: This subroutine is called by MAIN to calculate the enthalpy of equilibrium reactions.

25. **Subroutine ENTHABF**: This subroutine is called by MAIN to calculate the enthalpy of kinetic reactions.

26. **Subroutine INTERP**: This subroutine is called by HGC50 to compute the functional values (such as the Dirichlet concentrations, element source/sinks, point source/sinks, and incoming concentrations of fluids through variable boundary segments) at a particular time for all profiles. It uses linear interpolation of tabular data.

27. **Subroutine RHOMU0**: This subroutine is called by HGC50 to compute the ratio of fluid density and dynamic viscosity over the reference fluid density and dynamic viscosity at Gaussian points of each element, respectively.

28. **Subroutine SROSTR**: This subroutine is called by HGC50 to compute the ratio of fluid density over its reference value for each element and well source/sink.

29. **Subroutine GRDROV**: This subroutine is called by HGC50 to compute GRHOVM which accounts for the gradient term of Q described in Eq. (2.2.13). GRHOVM is used for the integration computed in Subroutine Q468 and is computed at Gaussian points of each element as

\[
\text{GRHOVM}(I,M) = \frac{\rho_0 \mathbf{V} \cdot \nabla \rho}{\rho_0} \quad (4.2.1)
\]

where I is the I-th Gaussian point of the M-th element.

30. **Subroutine SPROP**: This subroutine is called by HGC50, HYDROS, and HYDROT to calculate the values of moisture content, relative hydraulic conductivity, and the water capacity. 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.
31. **Subroutine VELT**: This subroutine calls FQ468D to evaluate the element matrices and the derivatives of the total head. It then sums over all element matrices to form a 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. The computed velocity field is then returned to HGC50, HYDROS, or HYDROT through the argument. This velocity field is also passed to Subroutine BCPREP to evaluate Darcy's flux across the seepage-infiltration-evapotranspiration surfaces.

32. **Subroutine FQ468D**: Subroutine FQ468D is called by the Subroutine VELT to compute the element matrices given by

\[
QB(I,J) = \int_{r_e} N_i^e N_j^e dR \quad (4.2.2)
\]

where \(N_i^e\) and \(N_j^e\) are the basis functions for nodal point \(i\) and \(j\) of element \(e\), respectively. Subroutines FQ468D also evaluates the element load vector:

\[
QRX(I) = -\int_{r_e} N_i^e i \cdot \mathbf{K} \cdot \frac{\rho_o}{\rho} (\nabla N_j^e) dR - \int_{r_e} N_i^e i \cdot \mathbf{K} (1 - \frac{\rho_o}{\rho}) \nabla z dR \quad (4.2.3)
\]

\[
QRY(I) = -\int_{r_e} N_i^e j \cdot \mathbf{K} \cdot \frac{\rho_o}{\rho} (\nabla N_j^e) dR - \int_{r_e} N_i^e j \cdot \mathbf{K} (1 - \frac{\rho_o}{\rho}) \nabla y dR \quad (4.2.4)
\]

and

\[
QRZ(I) = -\int_{r_e} N_i^e k \cdot \mathbf{K} \cdot \frac{\rho_o}{\rho} (\nabla N_j^e) dR - \int_{r_e} N_i^e k \cdot \mathbf{K} (1 - \frac{\rho_o}{\rho}) \nabla z dR \quad (4.2.5)
\]

where \(H_j\) = the total head at nodal point \(j\), \(i\) = the unit vector along the x-coordinate, \(j\) = the unit vector along the y-coordinate, and \(k\) = the unit vector along the z-coordinate.

33. **Subroutine SHAPE**: This subroutine is called by Subroutines FQ468D, FQ468TH, FQ468, Q468, Q468T and GRDROV to evaluate the value and their derivatives of the base function and/or weighting function at a Gaussian point.

34. **Subroutine AFABTA**: This subroutine is called by HGC50 to compute the optimum weighting factors for all sides of an element. The results are stored in array WETAB(J,M), where \(M = 1, 2, ..., \text{NEL}\) and \(J = 1, 2, ..., 12\) for Hexahedral elements, \(J = 1, 2, ..., 9\) for Triangular Prism elements, and \(J = 1, 2, ..., 6\) for Tetrahedral elements.

35. **Subroutine DISPC**: This subroutine is called by the Subroutine HGC50 to compute the hydrodynamic dispersion coefficients as described in Eq. (2.2.6) for Gaussian points of each element.

36. **Subroutine DISPT**: This subroutine is called by the Subroutine HGC50 to compute the thermal diffusion/dispersion/conductivity coefficients as described in Eq. (2.3.2) for Gaussian points of each element.

37. **Subroutine ADVW3D**: This subroutine is called by the Subroutine HGC50 to generate all the working arrays including IBW, IEW, NLRLW, and LRLW, for in-element particle tracking in the Lagrangian step computation used in the transient-state simulation. The more sub-elements generated for particle tracking, the more accurate result obtained. In this subroutine, the working arrays 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).

38. Subroutine FSFLOWPRE: This subroutine is used to compute the boundary fluxes by the known pressure head of Subroutine FBCMTRX, and save the result in the array BFLXW(I), I=1, NNP. If IBF() is 6 (which is the imperial boundary), boundary flux will be zero. BFLXW() will be sent to Subroutine FSFLOW to replace old value which is calculated by Darcy's Velocity.

39. Subroutine FSFLOW: This subroutine is used to compute the fluxes through various types of boundaries and the increasing rate of water content in the region of interest. Only at Steady-State flow and/or initial condition, the boundary flux will be calculated by Darcy's Velocity. At Transient flow simulation, boundary flux will use the results calculated by Subroutine FSFLOWPRE.

FRATE(7) stores the flux through the whole boundary that encloses the region of interest. It is given by

\[
FRATE(7) = \int_B (V_x n_x + V_y n_y + V_z n_z) dB
\]  

(4.2.6)

where B is the global boundary of the region of interest; \(V_x, V_y\) and \(V_z\) are Darcy's velocity 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 \(B_{S}\), and infiltration boundary \(B_{R}\), respectively, and FRATE(10) stores the flux through River boundary \(B_{Riv}\). These are given by:

\[
FRATE(1) = \int_{B_{D}} (V_x n_x + V_y n_y + V_z n_z) dB
\]  

(4.2.7)

\[
FRATE(2) = \int_{B_{C}} (V_x n_x + V_y n_y + V_z n_z) dB
\]  

(4.2.8)

\[
FRATE(3) = \int_{B_{N}} (V_x n_x + V_y n_y + V_z n_z) dB
\]  

(4.2.9)

\[
FRATE(4) = \int_{B_{S}} (V_x n_x + V_y n_y + V_z n_z) dB
\]  

(4.2.10)

\[
FRATE(5) = \int_{B_{R}} (V_x n_x + V_y n_y + V_z n_z) dB
\]  

(4.2.11)

\[
FRATE(10) = \int_{B_{Riv}} (V_x n_x + V_y n_y + V_z n_z) dB
\]  

(4.2.12)

FRATE(6), which is related to the numerical loss, is given by

\[
FRATE(6) = FRATE(7) - FRATE(10) - \sum_{i=1}^{5} FRATE(i)
\]  

(4.2.13)
FRATE(8) and FRATE(9) are used to store the source/sink and increased rate of water within the media, respectively:

\[ \text{FRATE(8)} = - \int_{R} \frac{\rho^*}{\rho_0} q \, dR \]  
(4.2.14)

and

\[ \text{FRATE(9)} = \int_{R} \frac{\rho}{\rho_0} F \frac{\partial H}{\partial t} \, dR \]  
(4.2.15)

If there are no numerical errors in the computation, the following equation should be satisfied:

\[ \text{FRATE(9)} = - [\text{FRATE(7)} + \text{FRATE(8)}] \]  
(4.2.16)

and FRATE(6) should be equal to zero. Equation (4.2.16) simply states that the negative rate of water going out from the region through the entire boundary and the rate due to source/sink are equal to the rate of water accumulated in the region.

40. Subroutine FQ468TH: This subroutine is used to compute the contribution to the increasing rate of water content from an element e:

\[ QTHP = \int_{R_e} \frac{\rho_o^*}{\rho_o} \left( \alpha \frac{\theta}{n_e} + \beta \theta + \frac{n_e}{dh} \right) \frac{\partial H}{\partial t} \, dR \]  
(4.2.17)

where \( \alpha^* \) and \( \beta^* \) are the modified compressibility of the soil matrix and liquid fluid, respectively, \( n_e \) is the effective porosity, and \( \theta \) is the moisture content. The computation of the above integral is straightforward.

41. Subroutine MVELT: This subroutine calls Q468D to evaluate the element matrices and the derivatives of concentrations. It then sums over all element matrices to form a matrix equation governing the flux components at all nodal points. To save computational time, the matrix is diagonalized by lumping. The flux components due to dispersion can thus be solved point by point. The flux due to the velocity is then added to the computed flux due to dispersion. The computed total flux field is then returned to HGC50 through the argument.

42. Subroutine Q468D: Subroutine Q468D is called by the Subroutine MVELT to compute the element matrices given by

\[ Q_{B(i,j)} = \int_{R_e} N_i^e N_j^e dR \]  
(4.2.18)

where \( N_i^e \) and \( N_j^e \) are the basis functions for nodal point i and j of element e, respectively. Subroutine Q468D also evaluates the element load vector:
\[ QRX(I) = -\int_{R_x} N_j \cdot i \cdot \theta D \cdot (\nabla N_j) C_{k,j} dR, \quad (4.2.19) \]

\[ QRY(I) = -\int_{R_y} N_j \cdot j \cdot \theta D \cdot (\nabla N_j) C_{k,j} dR, \quad (4.2.20) \]

and

\[ QRZ(I) = -\int_{R_z} N_j \cdot k \cdot \theta D \cdot (\nabla N_j) C_{k,j} dR, \quad (4.2.21) \]

where \( C_{k,j} \) is the concentration of the \( k \)-th component at nodal point \( j \), \( i \) is the unit vector along the \( x \)-direction, \( j \) is the unit vector along the \( x \)-direction, \( k \) is the unit vector along the \( z \)-coordinate, \( \theta \) is the moisture content, and \( D \) is the dispersion coefficient tensor.

43. **Subroutine SFLOW**: This subroutine is used to compute the flux rates through various types of boundaries and the increasing rate of material in the region of interest. \( FRATE(I,K) \) stores the flux through the variable-boundary for each biogeochemical component

\[ FRATE(I,K) = \int_{B_v} (F_{k,x} n_x + F_{k,y} n_y + F_{k,z} n_z) dB, \quad (4.2.22) \]

where \( B_v \) is the variable boundary of the region of interest; \( F_{k,x}, F_{k,y}, \) and \( F_{k,z} \) are the vertically integrated flux of the \( k \)-th components; and \( n_x, n_y, \) and \( n_z \) are the directional cosines of the outward unit vector normal to the boundary \( B_v \).

44. **Subroutine Q34S**: This subroutine is called by the Subroutine SFLOW to perform surface integration of the following type

\[ RQ(I,K) = \int_{R_e} N_j \cdot F_k dB, \quad (4.2.23) \]

where \( F_k \) is the normal flux of the \( k \)-th components.

45. **Subroutine CALKD**: This subroutine is called by HGC50 to calculate the equivalent \( K_d \) for all components.

46. **Subroutine FPRINT**: This subroutine, called by Subroutine HGC50, is used to print the flow variables. These include the diagnostic flow information, pressure heads, total heads, water contents, and Darcy's velocities.

47. **Subroutine PRINTT**: This subroutine, called by Subroutine HGC50, is used to print the transport variables. These include the total analytical concentrations, total dissolved concentrations, total sorbed concentrations, total precipitated concentrations of all components, and the negative logarithm of concentration for all component species.
48. **Subroutine TPRINT**: This subroutine, called by Subroutine HGC50, is used to print the temperature distribution.

49. **Subroutine FSTORE**: This subroutine, called by the Subroutine HGC50, is used to store the flow variables on Logical Unit 11. It is intended for the use of plotting purposes. The information stored includes region geometry, subregion data, and hydrological variables such as pressure head, total head, moisture content, and Darcy's velocity components.

50. **Subroutine STORE**: This subroutine, called by the Subroutine HGC50, is used to store the transport variables on Logical Unit 12. This data is intended for plotting and includes region geometry and transport variables.

51. **Subroutine TSTORE**: This subroutine, called by the Subroutine HGC50, is used to store the heat transfer variables on Logical Unit 22. This data is intended for plotting and includes region geometry and heat transfer variables.

52. **Subroutine TOTAL0**: This subroutine computes total component concentration and total kinetic variable concentration given individual species concentration.

53. **Subroutine TOTAL1**: This subroutine computes total component concentration and total kinetic variable concentration given individual species concentration at each Dirichlet node.

54. **Subroutine TOTAL2**: This subroutine computes total quantity for PDVs given individual species concentration, material flux or source/sink at each boundary element side, or source/sink side/node.

55. **Subroutine HYDROS**: Subroutines HYDROS is called by HGC50 to perform steady-state flow simulations.

56. **Subroutine HYDROT**: Subroutine HYDROT is called by HGC50 to perform transient flow simulations.

Both Subroutines HYDROS and HYSROT call

- Subroutine SPROP to compute the moisture content, the relative hydraulic conductivity, and the water capacity.

- Subroutine VELT to compute the Darcy's velocities.

- Subroutine BCPREP to determine the boundary conditions applied for the flow variable boundary sides.

- Subroutine FASEMB to compose the matrix equations and to implement the element and well source/sink.

- Subroutine FBCMTRX to save the matrix equations before implementing flow boundary condition.

- Subroutine FBC to implement flow boundary conditions.
• Subroutine BLKITR to solve matrix equations with block iteration method.
• Subroutine PISS to solve matrix equations with the pointwise Gauss-Seidel iteration strategy.
• Subroutine PPCG to solve matrix equations with the preconditioned conjugate gradient method using the polynomial as a preconditioner.
• Subroutine ILUCG to solve matrix equations with the preconditioned conjugate gradient method using the incomplete Cholesky decomposition as a preconditioner.

57. Subroutine BCPREP: This subroutine is called by HYDROS and HYDROT to prepare the infiltration-seepage boundary conditions during a rainfall period or the seepage-evapotranspiration boundary conditions during non-rainfall periods. It decides 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. Upon completion, this subroutine returns the Darcy’s flux (RSVAB(N, 4), where N is the N-th variable boundary node), infiltration/potential evapotranspiration rate (RSVAB(N, 3)), the ponding depth nodal index (INDRS(N, 1)), the flux-type nodal index (INDRS(N, 3)), the minimum pressure nodal index (INDRS(N, 2)), and the number of nodal points (NCHG) that have changed boundary conditions.

58. Subroutine FASEMB: This subroutine calls FQ468 to evaluate the element matrices. It then sums over all element matrices to form a global matrix equation governing the pressure head at all nodes.

59. Subroutines FQ468: These subroutine is called by the Subroutine FASEMB to compute the element matrix given by

$$QA(I,J) = \int_{E} N_i^e \frac{\rho}{\rho_o} F N_j^e dR , \quad (4.2.24)$$

and

$$QB(I,J) = \int_{E} (\nabla N_i^e) \cdot K (\nabla N_j^e) dR , \quad (4.2.25)$$

where $F$ is the soil property function. Subroutine FQ468 also calculates the element load vector given by

$$RQ(I) = \int_{E} [(\nabla N_i^e) \cdot K \frac{\rho}{\rho_o} (\nabla z) - N_i^e \frac{\rho}{\rho_o} q ] dR , \quad (4.2.26)$$

where $q$ is the source/sink.

60. Subroutine FBCMTRX: This subroutine abstracts the part of global matrices and global load vector, which is the boundary nodes before incorporating boundary condition. It also stores them into two arrays,
BCMTRX and BCRLD, which will be passed to Subroutine FSFLOWPRE to calculate the boundary flux.

61. **Subroutine FBC**: This subroutine incorporates Dirichlet, Cauchy, Neumann, and variable boundary conditions. For a Dirichlet boundary condition, an identity algebraic equation is generated for each Dirichlet nodal point. Any other equation having this nodal variable is modified accordingly to simplify the computation. For a Cauchy surface, the integration of the surface source is obtained by calling the Subroutine Q34S, and the result is added to the load vector. For a Neumann surface, the integration of gravity fluxes is obtained by calling the Subroutine FQ34S. Both the gradient flow and gravity fluxes are added to the load vector. The Subroutine FBC also implements the variable boundary conditions. First, it checks all infiltration-evapotranspiration-seepage points, identifying any that are Dirichlet points. If there are Dirichlet points, the method of incorporating Dirichlet boundary conditions mentioned above is used. If a given point is not the Dirichlet point, the point is bypassed. Second, it checks all rainfall-evaporation-seepage points again to see if any of them is a Cauchy point. If it is a Cauchy point, then the computed flux by infiltration or potential evapotranspiration is added to the load vector. If a given point is not a Cauchy point, it is bypassed. Because the infiltration-evaporation-seepage points are either Dirichlet or Cauchy points, all points are taken care of in this manner.

62. **Subroutine FQ34S**: This subroutine is called by the Subroutines BCPREP, FBC and FSFLOW to compute the surface node flux of the type

\[ RQ(I) = \int_{b_e}^{b_o} \rho q dB \]

(4.2.27)

where \( q \) is determined by the boundary type. For a Cauchy boundary condition, \( q \) represents the Cauchy flux. For a Neumann boundary condition, \( q \) represents the sum of Neumann flux and gravity flux. For the flux part of a variable boundary condition, \( q \) is the flux.

63. **Subroutine FQ34R**: This subroutine is called by Subroutines FBC to implement river boundary conditions, hence to compute the surface node flux of the type

\[ RQ(I) = \int_{b_e}^{b_o} \rho \frac{K_R}{b_R} N_j dB , \]

(4.2.28)

\[ BQ(I_j) = \int_{b_e}^{b_o} \rho \frac{K_R}{b_R} N_j dB , \]

(4.2.29)

Where \( K_R \) is the hydraulic conductivity of the river bottom sediment layer, \( b_R \) is the thickness of the river bottom sediment layer, and \( h_R \) is the depth of the river bottom measured from the river surface.

64. **Subroutine HEATS**: This routine is called by Subroutine HBGC to simulate the steady state heat transfer.

65. **Subroutine HEATT**: This routine is called by Subroutine HBGC to simulate the transient state heat transfer.

66. **Subroutine TASEMB**: This subroutine is called by Subroutines HEATS and HEATT to evaluate the element matrices for conservative and advective form FEM. It then sums over all element matrices to form
the compressed global matrices MMTRX, SMTRX, VMTRX and DMTRX and the global load vector RLD.

67. Subroutine Q468T: This subroutine is called by Subroutine TASEMB to compute matrices QA, QD, QC, QV, QH, and QR as follows:

For conservative form FEM method:

\[
QA(I,J) = \int_{R_e} N_a^e C_{RT} N_\beta^e dR \quad (4.2.30)
\]

\[
QD(I,J) = \int_{R_e} (\nabla N_a^e) \cdot D (\nabla N_\beta^e) dR \quad (4.2.31)
\]

\[
QV(I,J) = - \int_{R_e} (\nabla W_a^e) \cdot V_{CR} N_\beta^e dR \quad (4.2.32)
\]

\[
QH(I,J) = \int_{R_e} N_a^e C_{RH} N_\beta^e dR \quad (4.2.33)
\]

\[
QR(I) = \int_{R_e} N_a^e SSH dR \quad (4.2.34)
\]

For L-E and Advective form FEM:

\[
QA(I,J) = \int_{R_e} N_a^e C_{RT} N_\beta^e dR \quad (4.2.35)
\]

\[
QD(I,J) = \int_{R_e} (\nabla N_a^e) \cdot D (\nabla N_\beta^e) dR \quad (4.2.36)
\]

\[
QV(I,J) = \int_{R_e} W_a^e V_{CR} \nabla N_\beta^e dR \quad (4.2.37)
\]

\[
QC(I,J) = \int_{R_e} N_a^e C_{RQ} N_\beta^e dR \quad (4.2.38)
\]

\[
QR(I) = \int_{R_e} N_a^e C_{RQ} T^e dR \quad (4.2.39)
\]

68. Subroutine KMATRIX: This subroutine performs the following integration:

\[
QA(I,J) = \int_{R_e} N_a^e A N_\beta^e dR \quad (4.2.40)
\]

69. Subroutine VMATRIX: This subroutine performs the following integration: (For conservative form)

\[
QV(I,J) = - \int_{R_e} (\nabla W_a^e) \cdot V N_\beta^e dR \quad (4.2.41)
\]
70. **Subroutine VMATRIX1**: This subroutine performs the following integration: (For advective form)

\[ QV(I,J) = \int_{R_e} W \cdot \nabla N^e \, dR \]  
(4.2.42)

71. **Subroutine MATRIX**: This subroutine computes the following integration:

\[ Q(I,J) = \int_{R_e} \nabla N^e \cdot B \cdot \nabla N^e \, dR \]  
(4.2.43)

72. **Subroutine TBC**: This subroutine, called by Subroutines HEATS and HEATT for conservative form FEM, controls Dirichlet, Cauchy, Neumann and variable-boundary condition incorporation into the system of equations. For a Dirichlet boundary condition, a flux at each Dirichlet nodal point is solved as unknown. Any other equation having this nodal variable is modified accordingly to simplify computations. Subroutine BC also implements the Cauchy, Neumann and variable-boundary conditions, calling Subroutine Q34TB to compute the contributions of a cauchy, neumann, or variable-boundary segment to the global matrix equation.

73. **Subroutine TBC_1**: This subroutine, called by Subroutines HEATS and HEATT for advective form FEM, controls Dirichlet, Cauchy, Neumann and variable-boundary condition incorporation into the system of equations. For a Dirichlet boundary condition, a flux at each Dirichlet nodal point is solved as unknown. Any other equation having this nodal variable is modified accordingly to simplify computations. Subroutine BC also implements the Cauchy, Neumann and variable-boundary conditions, calling Subroutine Q34TB_1 to compute the contributions of a Cauchy, Neumann, or variable-boundary segment to the global matrix equation.

74. **Subroutine Q34TB**: This subroutine is called by Subroutine TBC for conservative form FEM to compute a four-by-four matrix, BQ, and a load vector, RQ, for each variable-boundary segment as follows:

\[ BQ(I,J) = 0, \quad RQ(I) = -\int_{B_e} N^e q_c \, dB, \quad \text{if} \quad V_i \cdot n < 0 \]  
(4.2.44)

and

\[ BQ(I,J) = -\int_{B_e} N^e n \cdot V_{CR} N^e \, dB, \quad RQ(I) = 0, \quad \text{if} \quad V_i \cdot n > 0 \]  
(4.2.45)

For each Cauchy-boundary segment as follows:

\[ BQ(I,J) = 0, \quad RQ(I) = -\int_{B_e} N^e q_c \, dB \]  
(4.2.46)

For each Neumann-boundary segment as follows:

\[ BQ(I,J) = -\int_{B_e} N^e n \cdot V_{CR} N^e \, dB, \quad RQ(I) = -\int_{B_e} N^e q_n \, dB \]  
(4.2.47)

75. **Subroutine Q34TB_1**: This subroutine is called by Subroutine TBC_1 for advective FEM to compute
a four-by-four matrix, $BQ$, and a load vector, $RQ$, for each variable-boundary segment as follows:

$$BQ(I,J) = \int_{B_e} N_a^e n \cdot C_p V N_e^d dB, \quad RQ(I) = -\int_{B_e} N_a^e (n \cdot C_p V) T_i dB, \quad \text{if } V_i \cdot n < 0$$  \hspace{1cm} (4.2.48)

and

$$BQ(I,J) = 0, \quad RQ(I) = 0, \quad \text{if } V_i \cdot n > 0$$  \hspace{1cm} (4.2.49)

For each Cauchy-boundary segment as follows:

$$BQ(I,J) = \int_{B_e} N_a^e n \cdot V_C N_e^d dB, \quad RQ(I) = -\int_{B_e} N_a^e q_n dB$$  \hspace{1cm} (4.2.50)

For each Neumann-boundary segment as follows:

$$BQ(I,J) = 0, \quad RQ(I) = -\int_{B_e} N_a^e q_n dB$$  \hspace{1cm} (4.2.51)

76. **Subroutine LANGVT**: This subroutine is called by HEATT to compute the Lagrangian tracking velocity for heat transfer when transient simulation is performed.

77. **Subroutine CKBDY**: This subroutine is called by HEATT and CHMIT to check all the boundary sides and generate the arrays, including NBDYB and IBDY, for later use in along-boundary tracking on both the unspecified and the Neumann boundary sides. NBDYB(I) represents the accumulated number of unspecified/Neumann boundary sides connecting with the 1-st through the (I-1)-th global node. IBDY(I) indicates the global boundary side to which the I-th unspecified/Neumann boundary side relates.

78. **Subroutine TADVBC**: This subroutine is called by HEATT to implement the boundary conditions. For Dirichlet boundaries, the Lagrangian temperature is specified. For variable boundaries, the fictitious particle associated with the boundary node must come from the interior nodes if the flow is directed out of the region. This requires Subroutine GNTRAK to be called first. If the flow is directed into the region, the temperature of incoming fluid is specified for variable boundaries. An intermediate Lagrangian temperature is calculated according to

$$T_{vi}^* = \frac{\int_{B_e} N_a^e p Cn \cdot VT_{in} dB}{\int_{B_v} N_a^e p Cn \cdot V dB}$$  \hspace{1cm} (4.2.52)

where $T_{vi}^*$ is the temperature due to the boundary source at the boundary node i, $V$ is Darcy's velocity, and $T_{in}$ is the Lagrangian temperature due to the boundary source at the boundary node i.

Cauchy boundary conditions are normally applied to the boundary where flow is directed into the region, and the material flux of incoming fluid is specified. The intermediate Lagrangian temperature is thus calculated according to
\[ T_{ci}^* = \frac{\int_{B_v} N_v^e q_{cdB}}{\int_{B_v} N_v^e \rho C_n V dB} \]  

(4.2.53)

where \( T_{ci}^* \) is the temperature due to the boundary source at the boundary node \( i \), \( V \) is Darcy's velocity, and \( q_c \) is the Cauchy flux of temperature in the incoming fluid.

79. **Subroutine Q34TADVB**: This subroutine is called by Subroutine TADVBC to evaluate the following integration for a boundary segment, for variable boundary.

\[ \text{RIQ}(I) = \int_{B_v} N_i V_n T \text{in dB}, \quad \text{and} \quad \text{RLQ}(I) = \int_{B_v} N_i V_n dB \]  

(4.2.54)

For Cauchy boundary,

\[ \text{RIQ}(I) = \int_{B_c} N_i q_c dB, \quad \text{and} \quad \text{RLQ}(I) = \int_{B_c} N_i V_n dB \]  

(4.2.55)

where \( \text{RIQ}(i) \) is the rate of heat passing through node \( i \) and \( \text{RLQ}(i) \) is the rate of water passing through node \( i \).

80. **Subroutine CHEMIS**: This subroutine is called by HGC50 to perform steady-state transport simulations. It calls Subroutine ASMBL to generate compressed global matrices and the global load vector; Subroutine TRANSP to solve hydrologic transport; and Subroutine OCSPIT to solve biogeochemical reactions.

81. **Subroutine CHEMIT**: This subroutine is called by HGC50 to perform transient transport simulations. It calls Subroutine ADVBC to implement the boundary conditions for the Lagrangian step; Subroutine GNTRAK to compute the Lagrangian concentrations; Subroutine TACDAC to compute the total concentrations of all adsorbent components and the number of equivalents; Subroutine ASMBL to generate compressed global matrices and the global load vector; Subroutine TRANSP to solve hydrologic transport; and Subroutine OCSPIT to solve biogeochemical reactions.

82. **Subroutine TACADC**: This subroutine is called by Subroutine CHEMIT to compute total concentrations of all adsorbent components and the number of equivalents.

83. **Subroutine WKTOTC**: This subroutine is called by Subroutine CHEMIT to prepare the right hand side of transport equations for mobile kinetic variables.

84. **Subroutine ASEMBL**: This subroutine is called by Subroutines CHEMIS and CHEMIT, and calls Subroutine Q468 to evaluate the element matrices. It then sums over all element matrices to form compressed global matrices MMTRX, SMTRX, VMTRX, and DMTRX and the global load vector RLD.

85. **Subroutines Q468**: This subroutine is called by Subroutine ASEMBL to compute the element matrix given by
\[
\begin{align*}
QA(I,J) &= \int_{R_e} N_i^e \theta N_j^e \, dR \\
QB(I,J) &= \int_{R_e} (\nabla N_i^e \cdot \theta D) (\nabla N_j^e) \, dR \\
QV(I,J) &= \int_{R_e} W_i^e \cdot \nabla N_j^e \, dR \\
QQ(I,J) &= \int_{R_e} N_i^e Q N_j^e \, dR \\
QC(I,J) &= \int_{R_e} N_i^e \frac{\partial \theta}{\partial t} N_j^e \, dR
\end{align*}
\]

and
\[
QE(I,J) = -\int_{R_e} N_i^e \{GRHOVM\} N_j^e \, dR.
\]

The term, GRHOVM, shown in the bracket of Eq. (4.2.61) for the integration is pre-computed at Gaussian points of each element in Subroutine GRDROV. Subroutine Q468 also calculates the element load vector given by
\[
QR(I,K) = \int_{R_e} N_i^e QC_{k}^{in} \, dR, \quad k = 1,2,...,N_k
\]

where \(C_{k}^{in}\) is the source concentration of the k-th component and kinetic variable.

86. **Subroutine TRANSP**: This subroutine is called by Subroutines CHEMIS and CHEMIT to perform hydrologic transport computations. TRANSP is called three times by CHEMIT for every time step. It simulates the transport of mobile components and mobile kinetic variables. Subroutine TRANSP also calls Subroutine EQNGEN_F or EQNGEN_P to produce the matrix equation from the compressed matrix and load vector for each primary dependent variables (PDVs); Subroutine BC to implement the boundary conditions; and Subroutines BLKITR, PISS, PPCG, ILUCG, MICPCG, or SSORCG to solve the resulting matrix equations.

87. **Subroutine EQNGEN_F**: This subroutine, called by Subroutine TRANSP, generates the matrix equation for PDVs from the compressed global matrices and the global load vectors when the fully implicit method is used to simulate reactive biogeochemical transport. It uses the array of the node connectivity produced in Subroutine PAGEN to form a banded matrix and load vector for all PDVs.

88. **Subroutine EQNGEN_P**: This subroutine, called by Subroutine TRANSP, generates the matrix equation for PDVs from the compressed global matrices and the global load vectors when the predictor-corrector method or the operator splitting method is used to simulate reactive biogeochemical transport. It uses the array of the node connectivity produced in Subroutine PAGEN to form a banded matrix and load vector for all PDVs.

89. **Subroutine BC**: This subroutine, called by Subroutine TRANSP, controls Dirichlet, Cauchy, Neumann
and variable-boundary conditions incorporation into the system of equations. For a Dirichlet boundary condition, an identity algebraic equation is generated for each Dirichlet nodal point. Any other equation having this nodal variable is modified accordingly to simplify computations. Subroutine BC also implements Cauchy, Neumann, and variable-boundary conditions, calling Subroutine Q34CNV to compute the contributions of a Cauchy, Neumann, or variable-boundary segment to the global matrix equation.

90. Subroutine CHKVBC: This subroutine is called by Subroutine TRANSP to perform the checking on flow directions over the variable boundaries. If the flow is directed into the region, then the FEM approach is used for boundary nodes. Otherwise, the LE approached is used. This subroutine is invoked only when the option of using LE FEM for interior nodes and conventional FEM for boundary nodes is used.

91. Subroutine Q34CNV: This subroutine is called by Subroutine BC to compute a four-by-four matrix, BQ, and a load vector, RQ, for each variable-boundary segment as follows:

\[
BQ(i,j) = \begin{cases} 
- \int_{R_e} N_i^e (\mathbf{n} \cdot \mathbf{V}) N_j^e dB, & \text{if } (\mathbf{n} \cdot \mathbf{V}) < 0 \ , \\
0, & \text{if } (\mathbf{n} \cdot \mathbf{V}) \geq 0 , 
\end{cases} 
\] (4.2.63)

and

\[
RQ(i) = - \int_{R_e} N_i^e (\mathbf{n} \cdot \mathbf{V}) E_{\text{ain}} dB, \text{ if } (\mathbf{n} \cdot \mathbf{V}) < 0 \ , \\
0, \text{ if } (\mathbf{n} \cdot \mathbf{V}) \geq 0 . 
\] (4.2.64)

where \( E_{\text{ain}} \) is the total dissolved concentration of the incoming fluid.

For each Cauchy boundary segment as follows:

\[
BQ(i,j) = - \int_{R_e} N_i^e (\mathbf{n} \cdot \mathbf{V}) N_j^e dB \quad (4.2.67)
\]

\[
RQ(i) = - \int_{R_e} N_i^e q_c dB \quad (4.2.68)
\]

For each Neumann boundary segment as follows:

\[
RQ(i) = - \int_{R_e} N_i^e q_n dB \quad (4.2.69)
\]

92. Subroutine ADVBC: This subroutine, called by Subroutine CHEMIT, implements the boundary conditions. For a Dirichlet boundary, the Lagrangian concentration is specified. For variable boundaries, the fictitious particle associated with the boundary node must come from the interior nodes if the flow is directed out of the region. Hence the Lagrangian concentration for the boundary node has already been computed in Subroutine GNTRAK and the implementation for such a boundary segment is bypassed. Thus, Subroutine GNTRAK must be called first. For variable boundaries, the concentration of incoming fluid is specified if the flow is directed into the region. The Lagrangian concentration is then calculated according to
\[ E_{vi}^{a*} = \int_{B_v}^{B_v} N_i V_n E^{ain} dB / \int_{B_v}^{B_v} N_i V_n dB, \quad (4.2.70) \]

where \( E_{vi}^{a*} \) is the Lagrangian concentration at the variable boundary node \( i \), \( V_n \) is the normal component of Darcy’s velocity, and \( E^{ain} \) is the concentration of incoming fluid.

Cauchy boundary conditions are normally applied to the boundary where flow is directed into the region, and the material flux of incoming fluid is specified. The intermediate Lagrangian concentration is thus calculated according to

\[ E_{ci}^{a*} = \int_{B_c}^{B_c} N_i q_c dB / \int_{B_c}^{B_c} N_i V_n dB, \quad (4.2.71) \]

where \( E_{ci}^{a*} \) is the Lagrangian concentration at the boundary node \( i \), \( V_n \) is the normal component of Darcy's velocity, and \( q_c \) is the Cauchy flux of the incoming fluid.

This subroutine is called twice to restore the value of Dirichlet boundaries.

**93. Subroutine Q34ADVB:** This subroutine, called by Subroutine ADVBC, evaluates the following integration for a boundary segment, for variable boundary.

\[ RIQ(i) = \int_{B_v}^{B_v} N_i V_n E^{ain} dB, \quad RLQ(i) = \int_{B_v}^{B_v} N_i V_n dB \quad (4.2.72) \]

For Cauchy boundary,

\[ RIQ(i) = \int_{B_c}^{B_c} N_i q_c dB, \quad RLQ(i) = \int_{B_c}^{B_c} N_i V_n dB \quad (4.2.73) \]

where \( RIQ(i) \) is the rate of biogeochemical passing through node \( i \) and \( RLQ(i) \) is the rate of water passing through node \( i \).

**94. Subroutine OCSPIT:** This subroutine, called by Subroutines HGC50, CHEMIS, and CHEMIT, calculates the total dissolved concentrations, total sorbed concentrations, and total precipitated concentrations of all components and also calculates the negative logarithm of the concentrations of all component species by calling Subroutine BIOGEOCHEM. The input to Subroutine OCSPIT are the total analytical concentrations of all components and their identification numbers.

**95. Subroutine BIOGEOCHEM:** Subroutine BIOGEOCHEM is called by Subroutine OCSPIT and will perform either the steady-state computation alone (KSS = 0 and NTI = 0), a transient state computation using the steady-state solution as the initial conditions (KSS = 0, NTI > 0), or a transient computation using user-supplied initial conditions (KSS = 1, NTI > 0). It initializes the concentrations of all product species given the estimate of component concentrations from Subroutine OCSPIT. BIOGEOCHEM then calls to Subroutine KINEQL to solve a set of mixed ordinary differential and algebraic equations governing mole balance, and biogeochemical kinetic and equilibrium reactions. It calls to Subroutine TOTDNP to compute total dissolved, total sorbed, and total precipitated concentrations of all components after concentrations of all species have been found. Finally, it calls to LPOUT to print biogeochemical species distributions.

**96. Subroutine PREPOS:** This subroutine is called by Subroutine DATAIO to perform preprocessing of
chemistry simulation. It calls Subroutines ERANK, KRank, EKRANK, ERANK1, CPEQK, and SPTYE to perform the diagonalization of reaction networks and pre-processes all arrays required for biogeochemical simulations.

97. Subroutine ERANK: This subroutine is called by Subroutine PREPOS to calculate the rank of the matrix made up of the stoichiometric coefficients of the equilibrium reactions and remove redundant fast reactions.

98. Subroutine KRANK: This subroutine is called by Subroutine PREPOS to compute the rank of a matrix made up of the stoichiometric coefficients of all equilibrium reactions plus one kinetic reaction. Its function is to find out and remove the irrelevant kinetic reactions.

99. Subroutine EKRANK: This subroutine is called by Subroutine PREPOS to find out the rank of the matrix made up of linearly independent equilibrium reactions and all relevant kinetic reactions.

100. Subroutine CPEQK: This subroutine is called by Subroutine PREPOS to find new equilibrium reaction constants after decomposition.

101. Subroutine ERANK1: This subroutine is called by Subroutine PREPOS to decompose the matrix made up of the stoichiometric coefficients of the equilibrium reactions represented by the mass action equations.

102. Subroutine SPTYPE: This subroutine is called by Subroutine PREPOS to obtain species type, particularly to indicate and accumulate fictitious species SIGMAO and SIGMAB (refer to ISCN(I,1)).

103. Subroutine INVERSE: This subroutine is called by Subroutine DATAIO and Subroutine BASWIT to produce the mapping from local to global species numbers. For example, the array ISCMI(I), it produces, denotes the global species number of the I-th component species.

104. Subroutine ARRV A: This subroutine is called by OCSPIT to prepare arrays for the calculation of Jacobian matrix with respect to total component and total kinetic variables's concentrations.

105. Subroutine EKCONST: This subroutine is called by OCSPIT. It calculates temperature dependent reaction constants.

106. Subroutine DERIV: This subroutine is called by OCSPIT. It is called to form JD and JN Matrices and calculate the partial derivatives of the reaction terms in the transport governing equations with respect to primary variables, i.e., \( \frac{\partial R_R}{\partial T} \) and \( \frac{\partial R_R}{\partial E} \), for fully implicit sequential iteration schemes when coupling the transport and reactions.

107. Subroutine FUNCT: This subroutine is called by Subroutine DERIV to form different functions required to compute the derivatives of reaction terms to primary variables.

108. Subroutine JACKR: This subroutine is called by Subroutine FUNCT to compute the contribution to the kinetic rows in the Jacobian matrix by forward and backward rate terms.

109. Subroutine LUDCMP: This subroutine is called by Subroutine DERIV to compute the matrix
determinant.

110. **Subroutine KIVRAT**: This subroutine is called by BIOGEOCHEM to compute reaction rate terms on the right hand side of governing equations for kinetic variables.

111. **Subroutine KINEQL**: This subroutine is called by BIOGEOCHEM to solve the system of mixed ordinary differential and nonlinear algebraic equations. The solution is done with the Newton-Raphson iteration method. For each iteration, Subroutine KINEQL calls:

- Subroutine ACOEF to compute the activity coefficients for all species.
- Subroutine MODIFK to calculate the equilibrium constants for all equilibrium product species and the modified forward and backward rate constants for any basic kinetic reactions.
- Subroutine MODBFK to calculate the modified forward and backward rate constants for any mixed kinetic reactions.
- Subroutine RESIDU to evaluate the residuals of all governing equations.
- Subroutine JACOB to compute the Jacobian of all governing equations.
- Subroutine DGELG to decompose the Jacobian matrix with full pivoting and for back substitution to obtain the differences between new iteration and previous iterations of all unknowns. New iterations are obtained by adding these differences to the old iterations.
- Subroutines MSP2EQ, TOTDNP, LPOUT, DISOLV, INDEXX, NPPT and BASWIT are also called by KINEQL.

112. **Subroutine ACOEF**: This subroutine is called by Subroutine KINEQL to compute ionic strength and activity coefficients of all species.

113. **Subroutine MODIFK**: This subroutine is called by Subroutine KINEQL to calculate the modified equilibrium constants for all equilibrium product species.

114. **Subroutine MODBFK**: This subroutine is called by Subroutine KINEQL to calculate the modified forward and backward rate constants for all mixed kinetic reactions.

115. **Subroutine RESIDU**: This subroutine is called to evaluate residuals of discretized ordinary differential and nonlinear algebraic equations governing biogeochemical kinetics and equilibrium. Residuals are evaluated in the Subroutine RESIDU for the following equations: (1) mole balance equations for aqueous components, (2) kinetic variable equations; (3) ion-exchange equations; (4) user specified empirical equations; and (5) equilibrium precipitation reaction equations. Residuals for equations governing the mole balance of adsorbent components are obtained by calling Subroutine RADC. RKRXNS computes production/consumption rates of kinetic variables. Residuals for user's empirical equations are obtained by calling USERMA. Residuals for ion-exchange reactions are obtained by calling Subroutine RIES.

116. **Subroutine RADC**: This subroutine is called by Subroutine RESIDU to evaluate residuals for equations governing the mole balance for adsorbent components.
117. **Subroutine USERMA**: This subroutine is called by Subroutines RESIDU and DEREQ to form equations specified by the user for equilibrium reactions.

118. **Subroutine RIES**: This subroutine is called by Subroutines RESIDU and DEREQ to evaluate residuals for ion-exchange reactions and cation ion exchange capacity constraint.

119. **Subroutine RKRXNS**: This subroutine is called by Subroutine RESIDU to evaluate the reaction rates of all kinetic reactions.

120. **Subroutine RUSRLW**: This subroutine is called by Subroutines RKRXNS and DERXN to compute user specified reaction rates.

121. **Subroutine RATEL0**: This subroutine is called by Subroutine RKRXNS to compute elements reaction rate.

122. **Subroutine PRODBF**: This subroutine is called by Subroutine RATEL0 to compute the species concentration product in the backward rate and forward rate formulation.

123. **Subroutine JACOB**: This subroutine is called by Subroutine KINEQL to evaluate Jacobians of the discretized ordinary differential and nonlinear algebraic equations governing biogeochemical kinetics and equilibrium. Subroutine JACOB calls DERDEQ and DERDRXN to prepare for the computation of jacobians of reactions. Jacobians are evaluated in Subroutine JACOB for the mole balance equations for aqueous components. Jacobians for the equations governing the mole balance of adsorbent components are obtained by calling Subroutine JADC. JACOB calls JARXN to calculate the contribution to all Jacobians from reactions to kinetic rows.

124. **Subroutine JADC**: This subroutine is called by Subroutine JACOB to compute the Jacobian for mole balance equations of adsorbent components.

125. **Subroutine JACKV1**: This subroutine is called by Subroutine JACOB to compute the contribution of variables in the kinetic variables to the kinetic rows in the Jacobian matrix. This subroutine is called for immobile kinetic variables and for mobile kinetic variables also when predictor corrector approach is used (Dirichlet boundary nodes are exception), where total kinetic variable concentration is not fixed.

126. **Subroutine JACKV2**: This subroutine is called by Subroutine JACOB to compute the contribution of variables in the kinetic variables to the kinetic rows in the Jacobian matrix. This subroutine is called for mobile kinetic variables where total kinetic variable concentration is fixed.

127. **Subroutine JARXN**: This subroutine is called by Subroutine JACOB to evaluate Jacobian contributions to all kinetic rows.

128. **Subroutine JOLAW**: This subroutine is called by Subroutine JARXN to compute contribution to the kinetic row in the Jacobian matrix from a user's specified rate law.

129. **Subroutine FBTERM**: This subroutine is called by Subroutine JARXN to calculate the contribution of forward and backward rates to the kinetic rows of Jacobian matrix.

130. **Subroutine DERDEQ**: This subroutine is called by Subroutine JACOB to calculate the derivatives.
of equilibrium species with respect to primary variables.

131. Subroutine DERDRXN: This subroutine is called by Subroutine JACOB to calculate the derivatives of kinetic reactions with respect to primary variables.

132. Subroutine DERDEQ: This subroutine is called by Subroutine DERDEQ to calculate the derivatives of each equilibrium species with respect to primary variables.

133. Subroutine DERXN: This subroutine is called by Subroutine DERDRXN to calculate the derivatives of each kinetic reactions with respect to primary variables.

134. Subroutine DGELG: This subroutine is called by Subroutine KINEQL to solve the Jacobian matrix equation. Gaussian elimination with full pivoting is used in the algorithm.

135. Subroutine TOTDNP: This subroutine is called by Subroutines BIOGEOCHEM and KINEQL to evaluate the log of free species concentrations, total dissolved concentrations, total sorbed concentrations, and total precipitated concentrations for all components or kinetic variable.

136. Subroutine LPOUT: This subroutine is called by the Subroutines BIOGEOCHEM and KINEQL to line print biogeochemical species distribution at desired nodes at desired time intervals. The information printed includes concentrations, modified equilibrium constants, and stoichiometric coefficients of all species.

137. Subroutine INDEXX: This subroutine is used to index the saturation value among all potential species that are subject to precipitation/dissolution reactions.

138. Subroutine NPPT: This subroutine is called by Subroutine KINEQL to determine the number of species allowed to precipitate without violating the phase rule.

139. Subroutine DISOLV: This subroutine is called by KINEQL to dissolve an assumed precipitated species that has shown negative concentrations during two successive iterations.

140. Subroutine MSP2EQ: This subroutine is called by the Subroutine KINEQL to count the number of mixed ordinary differential and nonlinear algebraic equations to be solved in the current precipitation cycle.

141. Subroutine BASWIT: This subroutine is called by Subroutine KINEQL to perform basis switching as described in chapter 2 of this report. In the forward switching, this subroutine calls Subroutine INDEXX to find the order of the species concentrations for determining the basis to be switched. In the backward switching, the same procedures for switching the information are performed again to restore the information which have been switched in the forward switching. In the DO 890 Loop of this subroutine, the switching is performed one species at each time for the required NSCH times.

142. Subroutine GNTRAK: This subroutine is called by CHEMIT and HEATT to control the process of backward particle tracking starting from global nodes. 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) Subroutine ELTRK4 to track a particle in a tetrahedral element, (2) Subroutine ELTRK6 to track a particle in a triangular prism element, and (3) Subroutine ELTRK8 to track a particle in a hexahedral element. When the particle cannot be tracked by normally elemental tracking, it calls Subroutine FIXCK to check if it hits specified or unspecified boundaries.
In order to make the particle tracking complete and remedy the given velocity field error on the unspecified boundaries, Subroutine FIXCK calls Subroutine ALGBDY to continue tracking particles along the unspecified/Neumann boundaries. At the end of backward particle tracking, the concentrations/temperature are obtained by interpolation executed in Subroutine INTERP1.

143. **Subroutine REPLAS**: This subroutine replaces the last six arguments with the first six arguments orderly.

144. **Subroutine ELENOD**: This subroutine determines 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.

145. **Subroutine WRKARY**: This subroutine prepares six working arrays for later usage.

146. **Subroutine ELTRK4**: This subroutine counts the particle tracking in a tetrahedral element. In the subroutine, the subelement in which the starting point is located is dug out firstly. 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) Subroutine TRAK1T to track a particle in the considered subelement if that particle is standing right on a node of the subelement, and (2) Subroutine 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; the single velocity approach is used if IJUDGE = 2.

147. **Subroutine ELTRK6**: This subroutine counts the particle tracking in a triangular prism element. In the subroutine, the subelement in which the starting point is located is dug out firstly. 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) Subroutine TRAK1P to track a particle in the considered subelement if that particle is standing right on a node of the subelement, and (2) Subroutine TRAK2P 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; the single velocity approach is used if IJUDGE = 2.

148. **Subroutine ELTRK8**: This subroutine counts the particle tracking in a hexahedral element. In the subroutine, the subelement in which the starting point is located is dug out firstly. 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) Subroutine TRAK1H to track a particle in the considered subelement if that particle is standing right on a node of the subelement, and (2) Subroutine TRAK2H 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; the single velocity approach is used if IJUDGE = 2.

149. **Subroutine MMLOC**: This subroutine is called by ELTRK4, ELTRK6, and ELTRK8 to locate the particle associated with a specific subelement for subsequent elemental tracking. If this particle coincides with the nodes of a subelement, ICODE = 0 is returned. In addition, the information of the particle location with respect to each surface of this element is also registered.

150. **Subroutine BDYPLN**: This subroutine locates the four global nodal numbers for returning to GNTRAK when the particle hits a boundary of the working element. In addition, it calls Subroutine ONPLAN to adjust the coordinate so that these five points are really on the same plane.
151. **Subroutine FIXCK**: This is a control panel to check the ongoing process when a particle hits the boundary of the region of interest. The backward tracked concentrations are obtained by interpolation if the boundary is specified including Dirichlet, Cauchy, and Variable types. Otherwise, the particle tracking continues along the unspecified boundary till either the specified boundaries is encountered or tracking time is consumed up.

152. **Subroutine TRAK1T**: This subroutine computes the particle tracking in a specified tetrahedral subelement when the starting point coincides with a node of the subelement. This subroutine calls Subroutine 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 Subroutine LOCQ3N to compute the exact location of the target point on the side. For accuracy, using the average velocity of both the starting 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 starting point is used to determine the location of the target point.

153. **Subroutine TRAK2T**: This subroutine computes the particle tracking in a specified tetrahedral subelement when the starting point does not coincide with a node of the subelement. This subroutine calls Subroutine 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 Subroutine LOCQ3N to compute the exact location of the target point on the side. For accuracy, using the average velocity of both the starting 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 starting point is used to determine the location of the target point.

154. **Subroutine CKCNEL**: This subroutine checks the elements connected to a specific side plane.

155. **Subroutine CKCOIN**: This subroutine checks if a specific point coincides with a global node.

156. **Subroutine ONPLAN**: This subroutine adjusts the particle coordinates to be on the same plane with the element side.

157. **Subroutine CKSIDE**: This subroutine checks if a specific point is on a side line of a side plane.

158. **Subroutine ONLINE**: This subroutine adjusts the particle coordinates to be on the same line with the other two points.

159. **Subroutine PLANEW**: This subroutine determine which one of the two sides, separated by a specified plane, the particle would move onto. 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.

160. **Subroutine LOCQ3N**: This subroutine locates the target point of a particle tracking in a specified element, which is either a tetrahedral or a triangular prism element. All the computations are made according to either the average velocity approach or the single velocity approach as the index parameter IJUDGE is 1 and 2, respectively. The Newton-Ralphson method is used to solve a set of two 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. This subroutine also calls Subroutine NEWXE to compute the new guess of this pair of natural coordinate.

161. **Subroutine NEWXE**: This subroutine is called by Subroutines LOCQ3N and LOCQ4N for taking a new guess of local coordinates within the iteration loop built with Newton-Ralphson scheme.

162. **Subroutine CHNGSN**: This subroutine is called by the LOCQ2N, LOCQ3N, LOCQ4N to change the sign of velocities when performing the backward particle tracking because the computation performed in LOCQ2N, LOCQ3N and LOCQ4N is achieved based on the forward particle tracking.

163. **Subroutine SURE3D**: This subroutine is called by LOCQ3N, LOCQ4N to locate the target point on a side in a single velocity tracking. The coordinates and associated velocities of a given point and of the element nodes on the projected side are passed to this subroutine for the calculation of target position and velocity as well as the elapsed time of this particle tracking.

164. **Subroutine LOCPLN**: This subroutine computes the base function values by calling the Subroutine BASE2D.

165. **Subroutine BASE2D**: This subroutine is called by LOCPLN to compute the base function values associated with a specified point based on the given two-dimensional global coordinates. For the cases of quadrilateral elements, it calls XS12D to calculate the local coordinates, and computes base functions with these determined local coordinates. For the cases of triangular elements, the base functions can be analytically determined based on the given global coordinates.

166. **Subroutine XS12D**: This subroutine is called by BASE2D to compute the local coordinate of a quadrilateral element given the global coordinate within that element.

167. **Subroutine TRAK1P**: This subroutine computes the particle tracking in a specified triangular prism subelement when the starting point coincides with a node of the subelement. This subroutine calls subroutine 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 Subroutine LOCQ4N (if the side is a quadrilateral one) or Subroutine 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 starting 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 starting point is used to determine the location of the target point.

168. **Subroutine TRAK2P**: This subroutine computes the particle tracking in a specified triangular prism subelement when the starting point does not coincide with a node of the subelement. This subroutine calls subroutine 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 Subroutine LOCQ4N (if the side is a quadrilateral one) or Subroutine 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 starting 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 starting point is used to determine the location of the target point.

169. **Subroutine LOCQ4N**: This subroutine locates the target point of a particle tracking in a specified element, which is either a hexahedral or a triangular prism 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-Ralphson method is used to solve a set of two simultaneous nonlinear algebraic equations such that the local coordinates of the target point on the pre-determined element side (a quadrilateral side) 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. This subroutine also calls Subroutine NEWXE to compute the new guess of this pair of natural coordinate.

170. **Subroutine BASE1**: This subroutine is called by ELTRK8 to compute the base functions for hexahedral elements.

171. **Subroutine TRAK1H**: This subroutine computes the particle tracking in a specified hexahedral subelement when the starting point coincides with a node of the subelement. This subroutine calls subroutine 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 Subroutine LOCQ4N to compute the exact location of the target point on the side. For accuracy, using the average velocity of both the starting 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 starting point is used to determine the location of the target point.

172. **Subroutine TRAK2H**: This subroutine computes the particle tracking in a specified hexahedral subelement when the starting point does not coincide with a node of the subelement. This subroutine calls Subroutine 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 Subroutine LOCQ4N to compute the exact location of the target point on the side. For accuracy, using the average velocity of both the starting 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 starting point is used to determine the location of the target point.

173. **Subroutine ALGBDY**: This subroutine is called by FIXCK to control the process of backward particle tracking along the unspecified boundaries. In the subroutine, the particle tracking is executed one boundary side by one boundary side based on the nodal velocity component along the side being considered. The tracking will not be stopped until either the tracking time is completely consumed or the particle encounters a specified boundary side. This subroutine calls BNDRY to track a particle along a predetermined boundary 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.

174. **Subroutine BNDRY**: This subroutine is called by ALGBDY to locate the target point of a particle tracking along a specified boundary side. 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. For
both approaches, the location of the target point can be determined by calling Subroutine LOCQ2N. However, when the velocity field is very complex, there might be no solution with the average approach. Thus, IJUDGE is originally set to 1 and is changed to 2 if the average approach fails. This control is executed in ALGBDY.

175. **Subroutine LOCQ2N**: This subroutine locates 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-Ralphson method is used to solve a nonlinear algebraic equations 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.

176. **Subroutine SUR2D3**: This subroutine is called by LOCQ2N to locate the target point on a line in a single velocity tracking. The coordinates and associated velocities of a given point and of the element nodes on the projected line are passed to this subroutine for the calculation of target position and velocity as well as the elapsed time of this particle tracking.

177. **Subroutine INTERP1**: This subroutine computes the contaminant concentrations by interpolation with the basis functions calculated by Subroutine BASE.

178. **Subroutine BASE**: This subroutine is called by subroutine INTERP1 to compute the base function values associated with a specified point based on the given three-dimensional global coordinates. For the cases of hexahedral elements, it calls XSI2D to calculate the local coordinates, and computes base functions with these determined local coordinates. For the cases of triangular prism elements, it calls XSI2DP to calculate the local coordinates, and computes base functions with these determined local coordinates. For the cases of tetrahedral elements, the base functions can be analytically determined based on the given global coordinates.

179. **Subroutine XSI3DP**: This subroutine computes the local coordinate (in the vertical direction) and the natural coordinate (in the horizontal directions) of a triangular prism element given the global coordinates for both the specified and element nodes.

180. **Subroutine XSI3D**: This subroutine computes the local coordinates of a hexahedral element given the global coordinates for both the specified point and element nodes.

181. **Subroutine BLKITR**: This subroutine is called by Subroutines HYDROS, HYDROT, TRANSP, HEATS and HEATT to solve the matrix equation with block iteration methods. For each subregion, a block matrix equation is constructed based on the global matrix equation and two pointer arrays GNPLR and LNOJCN (see Subroutine PAGEN), and the resulting block matrix equation is solved with the direct band matrix solver by calling Subroutine SOLVE. This is done for all subregions for each iteration until a convergent solution is obtained. This subroutine and the Subroutine SOLVE, to be described in the next paragraph, are needed only for when the block iteration option is used.

182. **Subroutine SOLVE**: This subroutine is called by Subroutine BLKITR to solve for the matrix equations solved of the type
where \([C]\) is the coefficient matrix and \(\{x\}\) and \(\{y\}\) are two vectors. \(\{x\}\) is the unknown to be solved, and \(\{y\}\) is the known load vector. The computer returns the solution \(\{y\}\) and stores it in \(\{y\}\). The computation is a standard banded Gaussian direct elimination procedure.

**183. Subroutine PISS:** Subroutine PISS is called by Subroutines HYDROS, HYDROT, TRANSP, HEATS and HEATT to solve matrix equations. The pointwise Gauss-Seidel iteration strategy is implemented in Subroutine PISS.

**184. Subroutine PPCG:** Subroutine PPCG is called by Subroutines HYDROS, HYDROT, TRANSP, HEATS and HEATT to solve matrix equations. The preconditioned conjugate gradient method using the polynomial as a preconditioner is implemented in PPCG. PPCG calls Subroutine POLYP to invert the preconditioner.

**185. Subroutine POLYP:** Subroutine POLYP is called by Subroutine PPCG to solve a modified residual that will be used in the preconditioned conjugate gradient algorithm.

**186. Subroutine ILUCG:** Subroutine ILUCG is called by Subroutines HYDROS, HYDROT, TRANSP, HEATS and HEATT to solve matrix equations. The preconditioned conjugate gradient method using the incomplete Cholesky decomposition as a preconditioner is implemented in ILUCG in which Subroutine LLTINV is called to invert the preconditioner.

**187. Subroutine LLTINV:** Subroutine LLTINV is called by Subroutine ILUCG to solve a modified residual that will be used in the preconditioned conjugate gradient algorithm.

**188. Subroutine MICPCG:** This subroutine is called by Subroutines TRANSP, HEATS and HEATT to solve matrix equations with the preconditioned conjugate gradient method using the modified incomplete Cholesky decomposition as a preconditioner. It calls to MICP to invert the preconditioner.

**189. Subroutine MICP:** This subroutine is called by Subroutine MICPCG to solve for a modified residual that will be used in the preconditioned conjugate gradient algorithm.

**190. Subroutine SSORCG:** This subroutine is called by Subroutines TRANSP, HEATS and HEATT, if necessary, to solve the linearized matrix equation with the preconditioned conjugate gradient method using the symmetric successive over-relaxation as a preconditioner. It calls to SSORP to invert the preconditioner.

**191. Subroutine SSORP:** This Subroutine is called by Subroutine SSORCG to solve for a modified residual that will be used in the preconditioned conjugate gradient algorithm.

**FUNCTIONS**

1. **Function FCOS:** This function computes 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 endpoint of the specified vector is located.

2. **Function EQ:** This function is called by Subroutine EKCONST to compute the temperature dependent
reaction constants by Van't Hoff equation.

3. **Function DOTPRD**: This function is to perform the product of two vectors. It is used by Subroutine PPCG, Subroutine SSORCG, and Subroutine MICPCG.
5. ADAPTATION OF HYDROGEOCHEM 5.0 TO SITE SPECIFIC APPLICATIONS

The following describes the maximum control integers that must be defined for each site-specific application and the data files that should be prepared.

5.1. Specification of Maximum Control Integers

For each site-specific application, 109 maximum control integers must be assigned with the PARAMETER statements in the MAIN program to specify the size of the problem. The listing and definitions of these parameters are given below.

**Maximum Control Integers for the Spatial Domain**

- MAXNPK = maximum number of nodes.
- MAXELK = maximum number of elements.
- MXBPNK = maximum number of boundary nodal points.
- MXBESK = maximum number of boundary-element surfaces.
- MXJBDK = maximum number of nodes connecting to any node.
- MXKBKD = maximum number of elements connecting to any node.

**Maximum Control Integers for the Time Domain**

- MXNTIK = maximum number of time steps.
- MXDTCK = maximum number of DELT changes.

**Maximum Control Integers for Subregions**

- LTMXNK = maximum number of total nodal points in any subregion, including interior nodes, global boundary nodes, and intraboundary nodes.
- LMXNPK = maximum number of nodal points in any subregion, including interior nodes and global boundary nodes.
- LMXBWK = maximum no. of the bandwidth in any subregion.
- MXRGNK = maximum no. of subregions.

**Maximum Control Integers for Material and Soil Properties**

- MXMATK = maximum number of material types.
- MXPPMK = maximum number of soil parameters per material to describe soil characteristic curves.
- MXMPMH = maximum number of flow material properties per material (= 14, for this code version).
- MXMPMK = maximum number of transport material properties per material (= 9, for this code version).
version).
MXMPMTK = maximum number of heat transfer material properties per material (= 7, for this code version).

Maximum Control-Integers and integer indices for Sub-element Tracking

MXNPWK = maximum number of nodes for a sub-element tracking.
MXELWK = maximum number of sub-elements for a sub-element tracking.
MXKGLK = maximum number of sub-elements in the Lagrangian step.
MXNPFK = maximum number of forward tracked nodal points.
MXADNPK = MXNPK
MXTUBK = maximum accumulated number of unspecified boundary sides that all the global node connect for transport.
MXTUBK = maximum accumulated number of unspecified boundary sides that all the global node connect for heat transfer.

Maximum Control Integers for Source/Sinks for Flow

MXSELK = maximum number of source elements (note: same as for transport).
MXSPRH = maximum number of source profiles.
MXSDPH = maximum number of data points on each element source/sink profile.
MXWNPK = maximum number of well nodal points (note: same as for transport).
MXWPRH = maximum number of well source/sink profiles.
MXWDPH = maximum number of data points on each well source/sink profile.

Maximum Control-Integers for Cauchy Boundary Conditions for Flow

MXCNPH = maximum number of Cauchy nodal points.
MXCESH = maximum number of Cauchy element surfaces.
MXCPRH = maximum number of Cauchy-flux profiles.
MXCDPH = maximum number of data points on each Cauchy-flux profile.

Maximum Control Integers for Neumann Boundary Conditions for Flow

MXNNPH = maximum number of Neumann nodal points.
MXNESH = maximum number of Neumann element surfaces.
MXNPRH = maximum number of Neumann-flux profiles.
MXNDPH = maximum number of data points on each Neumann-flux profile.

Maximum Control Integers for Rainfall-Seepage Boundary Conditions for Flow

MXVNPH = maximum number of variable nodal points.
MXVESH = maximum number of variable element surfaces.
MXVPRH = maximum number of rainfall profiles.
MXVDPH = maximum number of data point on each rainfall profile.

**Maximum Control Integers for Dirichlet Boundary Conditions for Flow**

MXDNPH = maximum number of Dirichlet nodal points.
MXDPRH = maximum number of Dirichlet total head profiles.
MXDDPH = maximum number of data points on each Dirichlet profile.

**Maximum Control Integers for River Boundary Conditions (Radiation Type) for Flow**

MXRNPH = maximum number of river boundary nodal points.
MXRESH = maximum number of river element surfaces.
MXRPRH = maximum number of river head profiles.
MXRDPH = maximum number of data point on each river-head profile.
MXRMAH = maximum number of river bottom materials.

**Maximum Control Integers for Source/Sinks for Transport**

MXSELK = maximum number of source elements (note: same as for flow).
MXSPRK = maximum number of source profiles.
MXSDPK = maximum number of data points on each element source/sink profile.
MXWNPK = maximum number of well nodal points (note: same as for flow).
MXWPRK = maximum number of well source/sink profiles.
MXWDPK = maximum number of data points on each well source/sink profile.

**Maximum Control Integers for Cauchy Boundary Conditions for Transport**

MXCNPK = maximum number of Cauchy nodal points.
MXCESK = maximum number of Cauchy element surfaces.
MXCPRK = maximum number of Cauchy profiles.
MXCDPK = maximum number of data points on each Cauchy profile.

**Maximum Control-Integers for Neumann Boundary Conditions for Transport**

MXNNPK = maximum number of Neumann nodal points.
MXNESK = maximum number of Neumann element surfaces.
MXNPRK = maximum number of Neumann profiles.
MXNDPK = maximum number of data points on each Neumann profile.

**Maximum Control Integers for Variable Boundary Conditions for Transport**

MXVNPK = maximum number of variable nodal points.
MXVESK = maximum number of variable element surfaces.
MXVPRK = maximum number of rainfall profiles.
MXVDPK = maximum number of data points on each rainfall profile.

**Maximum Control Integers for Dirichlet Boundary Conditions for Transport**

MXDNPK = maximum number of Dirichlet nodal points.
MXDPRK = maximum number of Dirichlet total head profiles.
MXDDPK = maximum number of data points on each Dirichlet profile.

**Maximum Control Integers for Source/Sinks for Heat Transfer**

MXTSEL = maximum number of source elements, (MXTSEL = MXSELK).
MXTSPR = maximum number of source profiles.
MXTSMDP = maximum number of data points on each element source/sink profile.
MXTWNP = maximum number of well nodal points, (MXTWNP = MXWNPK).
MXTWPR = maximum number of well source/sink profiles.
MXTWDP = maximum number of data points on each well source/sink profile.

**Maximum Control Integers for Cauchy Boundary Conditions for Heat Transfer**

MXTCNP = maximum number of Cauchy nodal points.
MXTCES = maximum number of Cauchy element surfaces.
MXTCPR = maximum number of Cauchy profiles.
MXTCDP = maximum number of data points on each Cauchy profile.

**Maximum Control Integers for Neumann Boundary Conditions for Heat Transfer**

MXTNKP = maximum number of Neumann nodal points.
MXTNES = maximum number of Neumann element surfaces.
MXTNPR = maximum number of Neumann profiles.
MXTNDP = maximum number of data points on each Neumann profile.

**Maximum Control Integers for Variable Boundary Conditions for Heat Transfer**

MXTVNP = maximum number of variable nodal points.
MXTVES = maximum number of variable element surfaces.
MXTVPR = maximum number of rainfall profiles.
MXTVDP = maximum number of data points on each rainfall profile.

**Maximum Control Integers for Dirichlet Boundary Conditions for Heat Transfer**

MXTDNP = maximum number of Dirichlet nodal points.
MXTDPR = maximum number of Dirichlet total head profiles.
MXTDDP = maximum number of data points on each Dirichlet profile.

**Maximum Control Integers for Biogeochemical Reactions**

- MAXCMK = maximum number of components.
- MXEQVK = maximum number of normal equilibrium reactions.
- MXEQXK = maximum number of equilibrium ion-exchanging species.
- MXEQMK = maximum number of equilibrium precipitation species.
- MXKIVK = maximum number of kinetic variables.
- MXNSBK = maximum number of adsorbing sites.
- MSITEK = maximum number of ion-exchange sites.
- MXEQSK = maximum number of user-specified mass action.
- MXEQSRK = maximum number of equilibrium reactions, (= MXEQVK + MXEQXK + MXEQMK + MXEQSK).
- MXSPK = maximum number of species, (= MAXCMK + MXEQVK + MXEQXK + MXEQMK + MXEQSK).
- MXHX1K = maximum number of species +flow information, (=MXSPK + 2).
- MAXEQK = maximum number of equations to be solved in reaction module, (= MAXCMK + MXEQXK + MXEQMK + MXEQSK).
- MAXHKK = maximum number of transport components+kinetic variables, (= MAXCMK + MXKIVK).
- MXKIRK = maximum number of kinetic reactions.
- MXRXNK = maximum number of reactions, (= MXEQSRK + MXKIRK).
- MPARK1K= maximum number of constituent species for an equilibrium species.
- MPARK2K= maximum number of constituent species for a forward/backward reaction.
- MPARK3K= maximum number of constituent species for a kinetic variable.
- MPARK4K= maximum number of constituent species for a component.

5.2. Specification of Maximum Control Integers with PARAMETER Statements

Let us assume that a region of interest is discretized into 30 x 20 x 10 nodes and 29 x 19 x 9 rectangular elements, i.e., 30 nodes along the longitudinal or x-direction, 20 nodes along the lateral or y-direction, and 10 nodes along the vertical or z-direction. Because we have a total of 30 x 20 x 10 = 6,000 nodes, the maximum number of nodes is MAXNPK = 6000. The total number of elements is 29 x 19 x 9 = 4,959, i.e., MAXELK = 4959. For this simple discretization problem, the maximum connecting number to any of the 6,000 nodes in the region of interest is 27, i.e., MXJBDK = 27, and the maximum connecting number of elements to any of the 6,000 nodes is 8, i.e. MXKBDK = 8. There will be 29 x 19 = 551 element surfaces each on the bottom and top faces of the region, 29 x 9 = 261 element-surfaces each on the front and back faces of the region, and 19 x 9 = 171 element-surfaces each on the left and right faces of the region. Thus, there will be a total of 1,966 element-surfaces, i.e., MXBESK = 1966. Similarly, we can compute the surface-boundary nodes to be 1,968, i.e., MXBNPK = 1968.

In order to specify maximum control-integers related to subregion data, we have to know how the region of interest is subdivided into subregions. Let us assume we have subdivided the region of interest into 20 subregions, each subregion has 30 x 10 nodes. It is seen, in fact, we are taking a vertical slice as a subregion. For this subregionalization, we have MXRGNK = 20. Each subregion has 30 x 10 = 300 nodes, resulting LMXNPK = 300. It is also seen that there will be 600 intraboundary nodes, 300 nodes each on the two
neighboring slices of a subregion. Thus, we have $LTMXNK = 900$. For each subregion, the maximum bandwidth can be computed as $LMXBWK = 23$ if the nodes are labelled along the $z$-directions consecutively.

In this version of the code, there are fourteen flow material properties, nine transport material properties, and seven heat transfer material properties for each material, so $MXMPMH = 14$, $MXMPMK = 9$ and $MXMPMTK = 7$. If there are nine soil parameters used to describe soil characteristic curves, we have $MXPPMK = 9$. If we assume that the whole region of interest is composed of three different kinds of materials, then we have $MXMATK = 3$. Assuming we will complete a 500-time-step simulation and change the time-step size 20 times during our simulation, we will have $MXNTIK = 500$ and $MXDTCK = 20$. To accurately perform node tracking, let us assume that for each element we need to divide it into $3 \times 3 \times 3$ sub-elements and use the average velocity approach. Thus, we have $MXNPWK = 64$, $MXELWK = 27$ and set $MXKGLK = 5000$, $MXNPFK = 5000$, $MXADNPK = MXNPK$.

Assume there are a maximum of 11 elements with distributed sources/sinks (i.e., $MXSELK = 11$) and a maximum of 10 nodes that can be considered well sources/sinks (i.e., $MXWNPK = 10$). For flow source/sink, we assume there are three different distributed source/sink profiles (i.e., $MXSPRH = 3$) and five distinct point source/sink profiles (i.e., $MXWPRH = 5$). Let us further assume that four data points are needed to describe the flow distributed source/sink profiles as a function of time and that 8 data points are required to described flow point source/sink profiles (i.e., $MXSDPH = 4$ and $MXWDPH = 8$). For transport source/sink, we assume there are three different distributed source/sink profiles (i.e., $MXSPRK = 3$) and five point source/sink profiles (i.e., $MXWPRK = 5$). Let us further assume four data points are needed to describe the transport distributed source/sink profiles as functions of time, and eight data points are required to described the transport point source/sink profiles (i.e., $MXSDPK = 4$ and $MXWDPK = 8$). For heat transfer source/sink, we assume there are three different distributed source/sink profiles (i.e., $MXTSPR = 3$) and five point source/sink profiles (i.e., $MXTWPR = 5$). Let us further assume four data points are needed to describe the heat transfer distributed source/sink profiles as functions of time, and eight data points are required to described the heat transfer point source/sink profiles (i.e., $MXTSDP = 4$ and $MXTWDP = 8$).

To specify maximum control-integers for flow boundary conditions, let us assume that the top surface is a variable boundary (i.e., on the air-soil interface, either ponding, infiltration, or evapotranspiration may take place). On the left face, fluxes from the adjacent aquifer are known. On the right face, the total head is assumed known. On the bottom face, natural drainage is assumed to occur (i.e., the gradient of the pressure head can be assumed zero).

There are $20 \times 10 = 200$ nodes and $19 \times 9 = 171$ element surfaces on the left face; thus $MXCNPH = 200$ and $MXCESH = 171$. It is further assumed that there two different fluxes going into the region through the left face and that each flux can be described by four data points as a function of time (i.e., $MXCPRH = 2$, and $MXCDPH = 4$). On the bottom surface, there are $30 \times 20 = 600$ nodes and $29 \times 19 = 551$ surface elements. Since the gradient of pressure head on the bottom surface is zero, there is only one Neumann flux profile, and two data points, one at zero time and the other at infinite time, are sufficient to describe the constant value of zero. Hence, we have $MXNNPH = 600$, $MXNESH = 551$, $MXNPRH = 1$, and $MXNDPH = 2$. On the top face, there will be $30 \times 20 = 600$ nodes and $29 \times 19 = 551$ elements. Let us assume that there are three different rainfall intensities that might fall on the air-soil interface, and that each rainfall intensity is a function of time and can be described by 24 data points. With these descriptions, we have $MXVNPH = 600$, $MXVESH = 551$, $MXVPRH = 3$, and $MXVDPH = 24$. On the right face, there are $20 \times 10 = 200$ nodes. Let us assume that here are twenty different values of the total head, one each on a vertical line of the right face. We further assume that each of these twenty values can be described by 8 data points as a function of time. We then have $MXDNPH = 200$, $MXDPRH = 20$, and $MXDDPH = 8$. 

108
To specify maximum control integers for transport boundary conditions, let us assume that the total analytical concentrations on the left face of the domain and the left half of the top face are known. The right half of the top face, bottom face, and right face are specified as variable boundaries. On the left face there are 20 x 10 = 200 nodes, and on the left half of the top face there are 15 x 20 = 300 nodes. Thus, we have a total of 500 transport Dirichlet nodes, so $\text{MXDNPK} = 500$. Let us assume that the middle 80 nodes on the left face have one specified total analytical concentration, the bottom 60 nodes on the left face have a different specified concentration, and the other 360 nodes have yet another specified total analytical concentration (i.e., $\text{MXDPBK} = 3$). We further assume that these analytical concentrations can be described as a function of time using eight data points, so $\text{MXDDPK} = 8$. There are 1,120 nodes and 1,007 element sides on the right half of the top face, bottom face, and right face: $\text{MXVNPK} = 1120$ and $\text{MXTVES} = 1007$. We can also assume there are three different incoming fluid concentrations going into the region through the top, bottom, and right faces, respectively, and each of these three concentrations is a function of time that can be described by eight data points. With these descriptions, we have $\text{MXVPRK} = 3$ and $\text{MXVDPK} = 8$.

To specify maximum control integers for heat transfer boundary conditions, let us assume that the temperature on the left face of the domain and the left half of the top face are known. The right half of the top face, bottom face, and right face are specified as variable boundaries. On the left face there are 20 x 10 = 200 nodes, and on the left half of the top face there are 15 x 20 = 300 nodes. Thus, we have a total of 500 heat transfer Dirichlet nodes, so $\text{MXTDNP} = 500$. Let us assume that the middle 80 nodes on the left face have one specified temperature, the bottom 60 nodes on the left face have a different specified temperature, and the other 360 nodes have yet another specified temperature (i.e., $\text{MXTDPR} = 3$). We can further assume that these temperatures can be described as a function of time using eight data points, so $\text{MXTDDP} = 8$. There are 1,120 nodes and 1,007 element sides on the right half of the top face, bottom face, and right face: $\text{MXTVNP} = 1120$ and $\text{MXTVES} = 1007$. We can also assume there are three different incoming fluid temperatures going into the region through the top, bottom, and right faces, respectively, and each of these three temperatures is a function of time that can be described by eight data points. With these descriptions, we have $\text{MXTVPR} = 3$ and $\text{MXTVDP} = 8$.

Assume that we are dealing with 12 components (i.e., $\text{MAXCMK} = 12$). Let us further assume that we will deal with 15 normal equilibrium reactions, 10 equilibrium ion-exchange reactions, 5 equilibrium precipitation reactions, 15 kinetic variables and 20 kinetic reactions (i.e., $\text{MXEQVK} = 15$, $\text{MXEQXK} = 10$, $\text{MXEQQM} = 5$, $\text{MXKIVK} = 15$, $\text{MXKIRK} = 20$). Also we assume there are two adsorbing sites, five ion-exchange sites, one user-specified mass action (i.e., $\text{MXNSBK} = 2$, $\text{MSITEK} = 5$, $\text{MXEQSK} = 1$). Thus, the number of equilibrium reactions will be $\text{MXEQRK} = \text{MXEQVK} + \text{MXEQXK} + \text{MXEQQM} + \text{MXEQSK} = 45$. The number of species will be $\text{MXSPK} = \text{MAXCMK} + \text{MXEQVK} + \text{MXEQXK} + \text{MXEQQM} + \text{MXEQSK} + \text{MXIVK} + \text{MXEQSK} = 58$, and $\text{MXHH1K} = \text{MXSPK} + 2 = 60$. The number of equations to be solved in the reaction module will be $\text{MAXEQK} = \text{MAXCMK} + \text{MXEQVK} + \text{MXEQXK} + \text{MXEQQM} + \text{MXIVK} + \text{MXEQSK} + \text{MXKIVK} = 43$. The number of transport components & kinetic variables will be $\text{MAXHKK} = \text{MAXCMK} + \text{MXKIVK} = 27$. Finally, we set $\text{MPAR1K} = 9, \text{MPAR2K} = 10, \text{MPAR3K} = 30, \text{MPAR4K} = 50$ for the number of constituent species for an equilibrium species, forward/backward reaction, kinetic variable and component, respectively.

From the above discussion, the following PARAMETER statements can be used to specify the maximum control integers in the MAIN program for the problem at hand:

```
PARAMETER (MAXELK=4959,MAXNPK=6000,MXBESK=1966,MXBNPK=1968)
PARAMETER (MXJBDK=27,MXKBKD=8)
PARAMETER (LTMXNK=900,LXNPK=300,LMXBWK=23,MXRGNK=20)
PARAMETER (MXNTIK=500,MXTCK=20)
```
PARAMETER (MXMATK=3,MXPPMK=9,MXMPMH=14,MXMPMK=9,MXMPMTK=7)
PARAMETER (MXSELP=11,MXWNPK=10)
PARAMETER (MNPPWK=64, MXELWK=27)
PARAMETER (MXKGLK=5000,MXNPFK=5000,MXADNP=MAXNP)
PARAMETER (MXTUBK=4*MXBNPK, MXTUBTK=4*MXBNPK)
PARAMETER (MXSPRH=3,MXSDPH=4,MXWPRH=5,MXWDPH=8)
PARAMETER (MXCNPH=200,MXCESH=171,MXCPPR=2,MXCDPH=4)
PARAMETER (MXNNPH=600,MXNESH=551,MXNPRH=1,MXNDPH=2)
PARAMETER (MXVEPK=600,MXVESK=551,MXVPRH=3,MXVDPH=24)
PARAMETER (MXTUDB=200,MXDPRH=20,MXDDPH=8)
PARAMETER (MXRNPH=1,MXRESH=1,MXRPKR=1,MXRDPH=1,MXRMK=1)

PARAMETER (MXSPRK=3,MXSDPK=4,MXWPRK=5,MXWDPK=8)
PARAMETER (MXCNPK=500,MXCPPK=3,MXCDPK=8)
PARAMETER (MXCESK=1,MXCNPK=1,MXCPPK=1,MXCDPK=1)
PARAMETER (MXNPK=1,MXNPPK=1,MXNPRIK=1,MXNPK=1)
PARAMETER (MXVESP=1007,MXVNPK=1120,MXVRPK=3,MXVDPK=8)

PARAMETER (MXTSPR=3,MXTSDP=4,MXTWPR=5,MXTWDP=8)
PARAMETER (MXTDNP=500,MXTDPK=3,MXTDPK=8)
PARAMETER (MXTCES=1,MXTCNP=1,MXTCPP=1,MXTCDP=1)
PARAMETER (MXTNES=1,MXTNPP=1,MXTNPRI=1,MXTNPK=1)
PARAMETER (MXTVES=1007,MXTVNP=1120,MXTVPR=3,MXTVDP=8)

PARAMETER(MAXCMK=12,MXEQVK=15,MXEQXK=10,MXEQMK=5,MXKIVK=15)
PARAMETER(MAXNSBK=2,MSITEK=5,MXEQSK=1)
PARAMETER(MAXKESK=MXEQV+MXEQXK+MXEQMK+MXEQSK)
PARAMETER(MAXCMK+MXEQK=MXKIVK+MXEQSK)
PARAMETER(MAXHK=MAXCMK+MXKIVK)
PARAMETER(MXKIRK=20)
PARAMETER(MAXRXNK=MAXQRK+MXKIRK)
PARAMETER(MPAR1K=9,MPAR2K=10,MPAR3K=30,MPAR4K=50)

5.3. Soil Property Function Specifications

Analytical functions can be used to describe the relationships of water content, water capacity, and relative hydraulic conductivity with pressure head if KSP = 0 is chosen in the DATA SET 6. The user must supply analytic functions to compute the water content, water capacity, and relative hydraulic conductivity based on the current value of pressure head. The parameters needed to specify the functional form are read and stored in SPP arrays. Modifications are needed in the SUBROUTINE SPROP to represent the desired analytic functions. One example is given below for clarity. Assume the water content and relative hydraulic conductivity are given as:
\[ \theta = \theta_r + \left( \frac{\theta_s - \theta_r}{A} \right) \left( \frac{A}{h - h_a} \right)^B \]  
(5.3.1)

and

\[ K_r = \left( \frac{\theta - \theta_r}{\theta_s - \theta_r} \right)^n \]  
(5.3.2)

Form Eq. (5.3.1), the water capacity can be obtained

\[ \frac{\partial \theta}{\partial h} = \frac{AB(\theta_s - \theta_r)(h - h_a)^{B-1}}{(A + |h - h_a|^B)^2} \]  
(5.3.3)

To apply the above three equations for computing soil properties, two blocks of statements in the SUBROUTINE SPROP should be modified. They have been pre-written in the SUBROUTINE SPROP of HYDROGEOCHEM 5.0 as listed in the following:

**BLOCK ONE:**

\[ WCR = SPP(1,MTYP,1) \]
\[ WCS = SPP(2,MTYP,1) \]
\[ HAA = SPP(3,MTYP,1) \]
\[ THAA = SPP(4,MTYP,1) \]
\[ THBB = SPP(5,MTYP,1) \]
\[ POWER = SPP(1,MTYP,2) \]

**BLOCK TWO:**

700 \[ DGSAT = THAA/(THAA + (DABS(-HNP-HAA))**THBB) \]
\[ PREEFF = 1.0D0/(1.0D0 + DGSAT*PHIMRG(KG)) \]
\[ EFFPOR(KG,M) = THETSE*PREEFF \]
\[ TH(KG,M) = DGSAT*EFFPOR(KG,M) \]
\[ USKFCT = (DGSAT)**POWER \]
\[ DNOM = THAA + (DABS(-HNP-HAA))**THBB \]
\[ DTH(KG,M) = (WCS-WCR)*THAA*(DABS(-HNP-THAA))**((THBB-1.0D0)/DNOM)**2 \]

Block one is to specify the variables for computing soil properties based on the given parameters stored in the SPP arrays. Block two is to compute the values of water content, relative hydraulic conductivity, and water capacity according to the desired analytic functions. After this modification is completed, the user need to recompile HYDROGEOCHEM 5.0 to generate the new execution code for simulations. Users can specify any desired analytic function for simulations as long as the SPP arrays are given and the SUBROUTINE SPROP is appropriately modified. On the other hand, the user can also use the tabular input to specify soil properties as described in DATA SET 6. For the tabular input, no modification is required in the SUBROUTINE SPROP. With these two options available for describing soil properties, HYDROGEOCHEM 5.0 is able to handle any site specific flow problem without the effort of major recoding.
5.4. Input and Output Devices

Nine logical units are needed to execute HYDROGEOCHEM 5.0. Logical Units 15 and 16 are used for data input and line printer output, respectively. If KVI \geq 0, logical Unit 17 is used to read flow variables produced by the previous runs of HYDROGEOCHEM 5.0. Logical Units 11, 12 and 22 must be specified to store the flow, transport and heat transfer simulation results, respectively, in binary form, which can be used for plotting purposes or for initial conditions of a restarting job. Logical Units 14, 13 and 23 are used to read flow, transport and heat transfer initial conditions, respectively, if the restart option is used. If the restart option for transport is active, the file associated with Logical Unit 13 for the current job should be the same file associated with Logical Unit 12 of the previous job. The same consistency should also be maintained between Logical Units 11 and 14, and 22 and 23. if the restart option for flow is active. The warning message will be printed to the file assigned to Logical Unit 34.

Upon executing HYDROGEOCHEM 5.0, the code will prompt eight times to input file names as follows:

PLEASE TYPE IN YOUR INPUT DATA FILE NAME. (assigned to Logical Unit 15)

THEN YOUR OUTPUT FILE NAME. (assigned to Logical Unit 16)

THEN YOUR BINARY STORAGE FILE NAME FOR FLOW. (assigned to Logical Unit 11)

THEN YOUR BINARY RESTART FILE NAME FOR FLOW. (assigned to Logical Unit 14)

THEN YOUR BINARY STORAGE FILE NAME FOR TRANSPORT. (assigned to Logical Unit 12)

THEN YOUR BINARY RESTART FILE NAME FOR TRANSPORT. (assigned to Logical Unit 13)

THEN YOUR BINARY STORAGE FILE NAME FOR HEAT TRANSFER. (assigned to Logical Unit 22)

THEN YOUR BINARY RESTART FILE NAME FOR HEAT TRANSFER. (assigned to Logical Unit 23)

THEN YOUR BINARY VELOCITY FILE NAME. (assigned to Logical Unit 17)

THEN YOUR WARNING FILE NAME. (assigned to Logical Unit 34)

A file name must be entered after each prompt. If an existing file name is chosen for the output file, binary storage files, or the warning file, the existing file will be overwritten.
Sixteen example problems are used to demonstrate the ability of HYDROGEOCHEM 5.0 to solve a variety of mixed subsurface flow, heat transfer, solute transport, and chemical reaction problems. Table 6.1 gives a brief description of each example simulated in this report. Since chemical reaction is incorporated with solute transport in the transport module, we use S-F (steady flow), T-F (transient flow), S-T (steady transport), T-T (transient transport), and T-H (transient heat transfer) to distinguish the involvement of each example among flow, heat transfer and/or transport associated with steady and/or transient state simulations. The input file name of each example is also listed in Table 6.1, and these files can be found in the complementary disk of this report.

Examples 1 is the steady-state flow problems, originally designed for 3DFEMWATER (Yeh, 1987b). Examples 2 and 3 are the transient and the combined steady-state/transient flow problems, respectively, of example 1. Examples 4 is the steady-state solute transport problems, originally designed for LEWASTE (Yeh and Chang, 1993). Examples 5 and 6 are the transient and the combined steady-state and transient versions, respectively, of example 4. Examples 7 and 8 are the steady and transient versions of Henry's salt water intrusion problems which involve coupled density-dependent flow and solute transport in a confined aquifer and were previously presented elsewhere (Cheng et al., 1998). These eight problems verify the adequacy of a variety of flow only, transport only, and coupled flow and solute transport.

Example 9 is the uranium mill tailing problem as previously presented in LEHGC 1.1 (Yeh et al., 1995b).

Examples 10 through 12 exemplify the capability of HYDROGEOCHEM 5.0 to deal with various combinations of the flow and biogeochemical transport problems. Example 10 simulates steady-state flow and both steady and transient biogeochemical transport. Here the initial conditions for the transient transport are obtained by its steady state solution and the steady-state flow is used for both steady-state and transient transport. Example 11 simulates both the steady-state and transient flows and the transient transport. In this example, the simulation of transient biogeochemical transport requires a transient flow field which is obtained with the transient flow simulation using the simulated steady state flow as the initial conditions of flows. Example 12 simulates the steady-state flow and transport and the transient flow and transport. Initial conditions for the transient flow are obtained from the steady-state flow simulation. Similarly, the initial conditions for the transient reactive transport are obtained from its steady-state simulation.

Example 13 is a coupled flow and biogeochemical transport problem. It is intended to exhibit the capability of HYDROGEOCHEM 5.0 to simulate the interactions and feedback between the precipitation/dissolution and the fluid flows and advection-diffusion/dispersion. The effects of precipitation/dissolution on both transport and fluid flow are elucidated.

Example 14 simulates 1-D transient advective-dispersive-reactive transport problem, in which three different schemes are compared in solving the governing transport equation: (1) FEM in advective form; (2) hybrid Lagrangian-Eulerian FEM for interior nodes plus FEM in advective form for boundary nodes; (3) hybrid Lagrangian-Eulerian FEM. Also explored is the accuracy of these schemes by comparing the simulated results and the analytical solutions.

Example 15 simulates steady flow and transient reactive biogeochemical transport. It was designed to demonstrate the capability of HYDROGEOCHEM 5.0 to deal with various combinations of the flow, reactive biogeochemical transport problems. A fictitious reaction system, which involves totally 41 species.
and 33 reactions (including aqueous complexation, adsorption, ion-exchange, and mineral dissolution reactions) has been designed to demonstrate the generic flexibility of BIOGEOCHEM module of HYDROGEOCHEM 5.0.

Example 16 is designed to explore the significance of the heat transfer on reactive biogeochemical transport described in Example 15. This example will demonstrate how temperature affects the biogeochemical transport as a consequence of changing the equilibrium constants and kinetic rate constants.

Table 6.1 Example Problems Simulated with HYDROGEOCHEM 5.0

<table>
<thead>
<tr>
<th>Ex.</th>
<th>Problem Description / Input File Name</th>
<th>Flow</th>
<th>Transport</th>
<th>Heat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Saturated-unsaturated flow / EX-f2.INP</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>T-F version of Example 2 / EX-f3.INP</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>S-F &amp; T-F versions of Example 2 / EX-f4.INP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>biogeochemical transport / EX-t2.INP</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>T-T version of Example 6 / EX-t3.INP</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>S-T and T-T versions of Example 6 / EX-t4.INP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>Steady state Henry's problem / Henry1.INP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>Transient state Henry's problem / Henry2.INP</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>9</td>
<td>Uranium mill tailing problem of LEHGC1.1 / EX-ft1.INP</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>Flow and biogeochemical transport with equilibrium complexation and adsorption reactions (1) / EX-ft2.INP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>Flow and biogeochemical transport with equilibrium complexation and adsorption reactions (2) / EX-ft3.INP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>12</td>
<td>Flow and biogeochemical transport with equilibrium complexation and adsorption reactions (3) / EX-ft4.INP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>13</td>
<td>(Coupled) fracture flow and matrix diffusion with equilibrium complexation and kinetic precipitation reactions / (EX-ft5-1.INP, EX-ft5-2.INP, EX-ft5-3.INP)</td>
<td>X</td>
<td>(X)</td>
<td>X</td>
</tr>
<tr>
<td>14</td>
<td>1-D biogeochemical transport verification/EX-t1.INP</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>15</td>
<td>3-D Flow and biogeochemical transport with complexation, adsorption, ion-exchange and mineral dissolution reactions / EX-fth-1.INP</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>16</td>
<td>3-D Flow and biogeochemical transport with complexation, adsorption, ion-exchange and mineral dissolution reactions with heat transfer consideration / EX-fth-2.INP, EX-fth-3.INP</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
6.1. Example 1: Saturated-Unsaturated Flow

Example 1 is the steady state simulation of a varied saturated flow problem. The region of interest is bounded on the left and right by two symmetric drains (20 m apart), on the bottom by an impervious aquifuge, and on the top by an air-soil interface (top of Fig. 6.1). The medium is assumed to have a saturated hydraulic conductivity of \( K_{xx} = K_{zz} = 0.01 \text{ m/day} \) and \( K_{yy} = K_{xy} = K_{xz} = K_{yz} = 0 \text{ m/day} \), a porosity of 0.25 and a field capacity of 0.05. The unsaturated characteristic hydraulic properties of the medium are given as:

\[
\theta = \theta_r + \left( \frac{\theta_s - \theta_r}{A} \right) \frac{A}{\theta_s - \theta_r - |h - h_a|^{B}} \tag{6.1}
\]

and

\[
K_r = \left( \frac{\theta - \theta_r}{\theta_s - \theta_r} \right)^n \tag{6.2}
\]

where \( h_a = 0 \text{ m}, A = 10 \text{ m}^2, B = 2, \theta_s = 0.25, \) and \( \theta_r = 0.05 \) are the parameters used to compute the water content and \( n = 2 \) is the parameter to compute the relative hydraulic conductivity.

To reflect the soil property function given by Eqs. (6.1) and (6.2), two blocks of statements associated with Example 1 in the SUBROUTINE SPROP should be changed to be available. They have been written in the SUBROUTINE SPROP of HYDROGEOCHEM 5.0 as listed in the following:

**block 1:**

\[
\begin{align*}
WCR &= SPP(1, MTYP, 1) \\
WCS &= SPP(2, MTYP, 1) \\
HAA &= SPP(3, MTYP, 1) \\
THAA &= SPP(4, MTYP, 1) \\
THBB &= SPP(5, MTYP, 1) \\
POWER &= SPP(1, MTYP, 2)
\end{align*}
\]

**block 2:**

\[
\begin{align*}
DGSAT &= THAA/(THAA+(DABS(-HNP-HAA))**THBB) \\
PREEFF &= 1.0D0/(1.0D0+DGSAT*PHIMRG(KG)) \\
EFFPOR(KG,M) &= \text{THETSE*PREEFF} \\
TH(KG,M) &= DGSAT*EFFPOR(KG,M) \\
USKFCT &= (DGSAT)**POWER \\
DNOM &= THAA+(DABS(-HNP-HAA))**THBB \\
DTH(KG,M) &= (WCS-WCR)*THAA*(DABS(-HNP-THAA))**(THBB-1.0D0)/DNOM**2 \\
IF &= (USKFCT.LT.CNSTKR) USKFCT=CNSTKR**2
\end{align*}
\]

Because of the symmetry, the region for numerical simulation will be taken as \( 0 \text{ m} < x < 10 \text{ m} \) and \( 0 \text{ m} < z < 10 \text{ m} \), and 10m wide along the \( y \)-direction will be assumed. The boundary conditions are given as: no flux is imposed on the left (\( x = 0 \text{ m} \)), front (\( y = 0 \text{ m} \)), back (\( y = 10 \text{ m} \)) and bottom (\( z = 0 \text{ m} \)) sides of the region; pressure head is assumed to vary from zero at the water surface (\( z = 2 \text{ m} \)) to 2 m at the bottom (\( z = 0 \text{ m} \)) on the right side (\( x = 10 \text{ m} \)); and variable boundary conditions are used elsewhere. Ponding depth is assumed to be \( 0 \text{ m} \) for all variable boundary surfaces. The incoming fluxes of variable boundaries are assumed equal to 0.006 m/day for the top surfaces and are equal to zero for the right side above the water surface. A steady state solution will be sought. A pre-initial condition is set as \( h = 2 - z \) which means the pre-initial water table
is located at \( z = 2 \) m. The simulation region is discretized with \( 10 \times 11 \times 1 = 110 \) elements and \( 11 \times 12 \times 2 = 264 \) nodes with the regular element size of \( 1 \) m \( \times \) \( 1 \) m \( \times \) \( 1 \) m for interior element and the refined element size of \( 1 \) m \( \times \) \( 0.5 \) m \( \times \) \( 1 \) m for right boundary element (bottom of Fig. 6.1). The pressure head tolerance for nonlinear iteration is \( 2 \times 10^{-3} \) m. The pressure head distribution is shown by contours in Fig. 6.2.

![Diagram of problem definition and discretization](image)

**Fig. 6.1. Problem Definition (top) and Discretization (bottom) for Example 1**
6.2. **Example 2: Transient and Varied Saturated Flow**

Example 2 is the transient version of the previous example. The initial condition is set as a hydrostatic pressure head distribution of $h = 2 - z$ through the domain of interest. The same boundary conditions as those of Example 2 are imposed on all boundary sides. The initial time step size is 0.25 day, and each subsequent time step size is increased with a multiplier of 2.0 but not greater than a maximum time step size of 64.0 day. A total of 40 time steps and 2111.75 days is simulated in this example. Figure 6.3 shows the rising of the water table at various times caused by infiltrations on the top surface. Since the only difference between Examples 1 and 2 is the steady or transient approach, the simulated water table of Example 2 at the last time step should be similar to that of Example 1 as long as the total simulation time is long enough for transient simulations to reach quasi-steady state. Example 2 is designed to meet this purpose as we compare the water table at $t = 2111.75$ days in Fig. 6.3 with that (i.e., $h = 0$) in Fig. 6.2.

6.3. **Example 3: Steady and Transient Simulations for Varied Saturated Flow**

Example 3 is the combined steady and transient version of Example 1. To perform the steady state simulation, a pre-initial condition of $h = 2 - z$ is assumed throughout the domain and variable boundary conditions with zero flux are applied on the top surface when $t = 0$ day. After the steady state results were obtained, 40 time steps of transient simulations as those of Example 2 were performed subsequently. The only difference between Examples 2 and 3 is the initial conditions, Example 3 used the results from steady state simulations while Example 2 assumed a hydrostatic pressure head distribution. By comparing the output files from both examples, the differences of computed and assumed initial conditions are small. Example 3 shows similar results as those of Example 2 in Fig. 6.3, as expected.
6.4. Example 4: Steady Solute Transport

Example 4 involves steady solute transport in a rectangular domain having a size of 20 cm x 1 cm x 10 cm (top of Fig. 6.4). The simulation region is assumed to contain the soil with a bulk density of 1.2 gm/cm$^3$, a longitudinal dispersivity of 1 cm, and a lateral dispersivity of 0.1 cm. A pre-initial condition of zero concentration is assumed for a conservative tracer throughout the domain of interest. Steady state simulation is performed with the following boundary conditions (top of Fig. 6.4): on the left side ($x = 0$ cm) of the domain, a concentration of 1.0 gm/ml is maintained at $4 \text{ cm} < z < 6 \text{ cm}$ and 0 gm/ml for others; and a natural condition is imposed on the right side ($x = 20$ cm). A steady flow field is assumed with a specific discharge (Darcy's velocity) of 1.0 cm/day and a constant moisture content of 0.8. The region is discretized with 40 x 10 x 1 = 400 elements and 41 x 11 x 2 = 902 nodes. Each element has the same size of 0.5 cm x 1 cm x 1 cm (bottom of Fig. 6.4). Figure 6.5 shows the simulated concentration contours, and how the tracer distributes in the $z$-direction due to the existence of the lateral dispersivity. However, the simulation results are still symmetric to the axis of $z = 5$ cm due to the homogeneous material properties and symmetric boundary conditions (with respect to $z = 5$ axis).
Fig. 6.4. The Problem Definition (top) and Discretization (bottom) for Example 4
6.5. Example 5: Transient Solute Transport

Example 5 is the transient version of the previous example. An initial condition of zero concentration is assumed through the entire domain. The same boundary conditions as those of Example 4 are also imposed in this example. The initial time step size is 0.05 day, and each subsequent time step size is increased with a multiplier of 1.02 until a maximum time step size of 0.13 day is reached. A total of 250 time steps resulting in a total simulation time of 30.23 days is performed in this example. Figure 6.6 depicts the 0.5 concentration profiles at various time steps. The 0.5-concentration contour at $t = 30.23$ days in Fig. 6.6 is similar to that in Fig. 6.5 as the total simulation time of 30.27 days is long enough to reach quasi-steady state.
6.6. Example 6: Steady and Transient Solute Transport

Example 6 is the combined steady and transient version of Example 4. The results from the steady state simulation will be used as initial conditions for the subsequent transient simulation. Example 6 is designed to reproduce identical transient results to those of Example 5. A pre-initial condition of 0 concentration is assumed through the entire domain for steady state simulation. In order to obtain the same initial concentrations as those of Example 5, boundary conditions imposed for $z = 4$ cm to 6 cm at the left side ($x = 0$ cm) are set to zero while performing the steady state simulation. For transient simulations, boundary conditions are set as the same as those of Example 5. By comparing output files of both examples, the transient results of this example are identical to those of Example 5 as shown in Fig. 6.6.
6.7. Example 7: Steady Henry's Salt Water Intrusion Problem

This example is widely known as Henry's salt water intrusion problem (Henry, 1959). It involves subsurface flow and solute transport of salt water in a confined aquifer. The aquifer under consideration is a uniform isotropic aquifer that is bounded below and above by impermeable strata. The aquifer is exposed on the right side by a stationary seawater body and is recharged on the left side by freshwater with a constant influx. The top of Fig. 6.7 shows the problem domain (200 m x 100 m) and boundary conditions applied for this example. On the right (coastal) side, variable boundary conditions are used to permit convective salt mass out of the domain if flow is going out, or to allow the salt mass add to the domain via both advection and dispersion fluxes if flow is coming in. Therefore, the interface between sea water and fresh water on the right side is determined automatically during simulations.

Steady simulation is performed in this example. The region of interest is discretized into 176 nodes and 150 rectangular elements (bottom of Fig. 6.7). Both the longitudinal and lateral dispersivities are set to zero, only a molecular diffusion coefficient of $6.6 \times 10^{-2} \text{m}^2/\text{day}$ is assumed throughout the domain of interest. The relationship between fluid density and concentration (under isothermal conditions) is expressed as $\rho = \rho_0 \left(1 + \varepsilon C\right)$, where $\varepsilon = 0.0245$, $\rho_0 \text{(M/L}^3\text{)}$ is the density of water at salt water concentration $C$, and $\rho_0$ is the density of water at zero concentration. The rest of the parameters for the problem are: $U = 6.6 \times 10^{-3} \text{m/day}$, $K = 1 \text{m/day}$, and $\varphi = 0.35$ (where $\varphi$ is the porosity of the aquifer and $K$ is the hydraulic conductivity (L/T)). As described in the section 2.1.3 of this report, $\rho = \rho_0 \left(1 + \sum w_i C_i\right)$ is the equation used to estimate the density effect, where $w_i$ is the molecular weight of the $i$-th biogeochemical component and $C_i$ is the concentration of the $i$-th biogeochemical component. In order to reflect the same $\varepsilon = 0.0245$ as that in Henry's problem, we use only one conservative tracer having a molecular weight of $24500 \text{g/m}^3$ and $\rho_0 = 10^6 \text{g/m}^3$ to perform simulation.

A pre-initial zero concentration and a hydrostatic pressure head of 0 m (the total head datum is set at $z = 100 \text{m}$) are assumed throughout the domain of interest. Figure 6.8 shows the simulated concentration contours which can be found similar to other researchers' results (Cheng et al., 1998).

6.8. Example 8: Transient Henry's Salt Water Intrusion Problem

Example 8 is the transient version of the Henry's problem. The initial time step size is 12 days, and each subsequent time step size is increased with a multiplier of $1.17169$ until a maximum time step size of 600 days is reached. Based on the same setting in the previous example, 29 time steps are performed in this example. As a result, the total simulation time is 6000.85 days. Figure 6.9 shows the 0.5-concentration contours at various time steps.
V_n = dC/dn = 0

K = 1 m/d
U = 6.6 x 10^{-3} m/d
e = 0.0245
f = 0.35

H = e (b-z)

V_n = U
C = 0

V_n = dC/dn = 0

if flow going out
if flow coming in

where C_{in} = 1

Fig. 6.7. Problem Definition (top) and Discretization (bottom) for Example 7
Fig. 6.8. Concentration Contours for Example 7 (the Steady Henry's Problem)

Fig. 6.9. The 0.5-Concentration Contour at Various Time Steps for Example 8 (the Transient Henry's Problem)
6.9. Example 9: Uranium Milling Tailing Problem

Example 9 is the Uranium milling tailing problem previously presented in LEHGC1.1. This problem considers the release and migration of uranium from a simplified uranium mill tailings pile. A schematic two-dimensional vertical cross section of the hypothetical site is shown in the top of Fig. 6.10. The mill tailings pile is located adjacent to a surface that slopes down to a river. The region is discretized with 158 elements and $192 \times 2 = 384$ nodes (bottom of Fig. 6.10). The vertical left edge, front edge, back edge and horizontal bottom of the region are impermeable no-flow boundaries. The sloping region on the top right is a variable flow boundary with either zero ponding depth or a net rainfall rate of 0.0139 dm/day. The horizontal region on the top of the mill tailings pile is a Cauchy flow boundary with an infiltration rate of 0.139 dm/day. The ten nodes on the vertical line on the right side and the four nodes on the river bottom are specified as known-head conditions (Dirichlet flow-boundary conditions). Total head at the ten vertical nodes is 39 dm. The total head on the left and right nodes of the river bottom are 45 dm and 40 dm, respectively. Only advective transport is considered in this example.

The problem considers the hydrogeochemical transport of seven components: calcium, carbonate, uranium(VI), sulfate, phosphate, ferrous iron, and hydronium (pH). The pH is computed for each point on the basis of the total excess hydronium designated by TOTH (total analytical excess or deficit of protons, a concept similar to the proton condition). A total of 35 species and 14 minerals is defined for the problem; redox reactions were not considered. Table 6.2 lists biogeochemical reactions involved in this example.

A total of 300 days were simulated with 150 time steps and a constant step size of 2 days. The composition of the pore water in the tailings, pore water outside of the tailing pile and the recharge water are given in Table 6.3. The mill tailings pile is represented by Dirichlet nodes in order to hold the total concentrations of the biogeochemical components constant throughout the simulation. The sloping area to the right of the tailings pile, the river bank, the river and right-hand side of the domain below the river are variable boundary nodes.

From the steady flow results, pressure head contours and velocity fields are depicted in Fig. 6.11. Figure 6.12 shows contour plots of computed pH at various time steps. Initially, the acidity is confined to the tailing pile, and the pH is greater than 7 outside. Simulations at 100 and 300 days illustrate transport of the acidity outside the tailings; however, due to reaction with CaCO₃ the pH is neutralized rapidly in the reaction zone.

Figures 6.13 and 6.14 depict contour plots of precipitated carbonate and sulfate, respectively. The two figures show the displacement of calcite by gypsum as the simulation proceeded. CaCO₃, which was initially present everywhere outside of the tailings ($t = 0$ day, in Fig. 6.13), disappears as acidity is transported out of the tailings. At $t = 300$ days (Fig. 6.13), CaCO₃ is present only in a small zone near the right edge of the cross section. Gypsum concentrations spread from the tailings pile due to transport of sulfate and calcium (Fig. 6.14). However, below the hill slope, gypsum is dissolved by incoming rainwater with low sulfate concentration (Fig. 6.14). The release of calcium by dissolution of the gypsum leads to some precipitation of calcite by reaction with the carbonate transported from the tailing pile.

The distribution of uranium is significantly different from that of sulfate and calcium. As shown in Fig. 6.15, the concentration of uranium steadily increases with time in the region surrounding the tailings due to advective transport. Unlike calcium and sulfate that are present in high concentrations through the cross section, the difference between the uranium concentration with and outside the tailings is more than three orders of magnitude.
Fig. 6.10. Problem Definition (top) and Discretization (bottom) for Example 9
Table 6.2. Reaction Network for Example 9

<table>
<thead>
<tr>
<th>Reaction</th>
<th>#</th>
<th>Reaction Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>H$_2$O(aq) $\leftrightarrow$ H$^+$ + OH$^-$</td>
<td>R1</td>
<td>Log$K_e$ = -13.99</td>
</tr>
<tr>
<td>Ca$^{2+}$ + CO$_3^{2-}$ $\leftrightarrow$ CaCO$_3$ (aq)</td>
<td>R2</td>
<td>Log$K_e$ = 3.22</td>
</tr>
<tr>
<td>Ca$^{2+}$ + H$^+$ + CO$_3^{2-}$ $\leftrightarrow$ CaHCO$_3^+$</td>
<td>R3</td>
<td>Log$K_e$ = 11.43</td>
</tr>
<tr>
<td>Ca$^{2+}$ + SO$_4^{2-}$ $\leftrightarrow$ CaSO$_4$(aq)</td>
<td>R4</td>
<td>Log$K_e$ = 2.31</td>
</tr>
<tr>
<td>Ca$^{2+}$ + 2H$^+$ + PO$_4^{3-}$ $\leftrightarrow$ CaH$_2$PO$_4^-$</td>
<td>R5</td>
<td>Log$K_e$ = 20.96</td>
</tr>
<tr>
<td>Ca$^{2+}$ + PO$_4^{3-}$ $\leftrightarrow$ CaPO$_4$</td>
<td>R6</td>
<td>Log$K_e$ = 6.46</td>
</tr>
<tr>
<td>Ca$^{2+}$ + H$^+$ + PO$_4^{3-}$ $\leftrightarrow$ CaHPO$_4$ (aq)</td>
<td>R7</td>
<td>Log$K_e$ = 15.08</td>
</tr>
<tr>
<td>Ca$^{2+}$ + H$_2$O $\leftrightarrow$ H$^+$ + CaOH$^+$</td>
<td>R8</td>
<td>Log$K_e$ = -12.58</td>
</tr>
<tr>
<td>Fe$^{2+}$ + SO$_4^{2-}$ $\leftrightarrow$ FeSO$_4$(aq)</td>
<td>R9</td>
<td>Log$K_e$ = 2.20</td>
</tr>
<tr>
<td>Fe$^{2+}$ + H$_2$O $\leftrightarrow$ H$^+$ + FeOH$^+$</td>
<td>R10</td>
<td>Log$K_e$ = -9.50</td>
</tr>
<tr>
<td>Fe$^{2+}$ + 2H$_2$O $\leftrightarrow$ 2H$^+$ + Fe(OH)$_2$ (aq)</td>
<td>R11</td>
<td>Log$K_e$ = -20.57</td>
</tr>
<tr>
<td>Fe$^{2+}$ + 3H$_2$O $\leftrightarrow$ 3H$^+$ + Fe(OH)$_3^-$</td>
<td>R12</td>
<td>Log$K_e$ = -31.00</td>
</tr>
<tr>
<td>Fe$^{2+}$ + 4H$_2$O $\leftrightarrow$ 4H$^+$ + Fe(OH)$_4^{2-}$</td>
<td>R13</td>
<td>Log$K_e$ = -46.00</td>
</tr>
<tr>
<td>UO$_2^{2+}$ + H$_2$O $\leftrightarrow$ H$^+$ + (UO$_2$)(OH)$^+$</td>
<td>R14</td>
<td>Log$K_e$ = -5.30</td>
</tr>
<tr>
<td>2UO$_2^{2+}$ + 2H$_2$O $\leftrightarrow$ 2H$^+$ + (UO$_2$)$_2$(OH)$_2^{2+}$</td>
<td>R15</td>
<td>Log$K_e$ = -5.68</td>
</tr>
<tr>
<td>3UO$_2^{2+}$ + 4H$_2$O $\leftrightarrow$ 4H$^+$ + (UO$_2$)$_3$(OH)$_4^{2+}$</td>
<td>R16</td>
<td>Log$K_e$ = -11.88</td>
</tr>
<tr>
<td>3UO$_2^{2+}$ + 5H$_2$O $\leftrightarrow$ 5H$^+$ + (UO$_2$)$_3$(OH)$_5^-$</td>
<td>R17</td>
<td>Log$K_e$ = -15.82</td>
</tr>
<tr>
<td>4UO$_2^{2+}$ + 7H$_2$O $\leftrightarrow$ 7H$^+$ + (UO$_2$)$_4$(OH)$_7^-$</td>
<td>R18</td>
<td>Log$K_e$ = -21.90</td>
</tr>
<tr>
<td>3UO$_2^{2+}$ + 7H$_2$O $\leftrightarrow$ 7H$^+$ + (UO$_2$)$_3$(OH)$_3^-$</td>
<td>R19</td>
<td>Log$K_e$ = -28.34</td>
</tr>
<tr>
<td>UO$_2^{2+}$ + CO$_3^{2-}$ $\leftrightarrow$ (UO$_2$)(CO$_3$)(aq)</td>
<td>R20</td>
<td>Log$K_e$ = 9.65</td>
</tr>
<tr>
<td>UO$_2^{2+}$ + 2CO$_3^{2-}$ $\leftrightarrow$ (UO$_2$)(CO$_3$)$_2^{2-}$</td>
<td>R21</td>
<td>Log$K_e$ = 17.08</td>
</tr>
<tr>
<td>UO$_2^{2+}$ + 3CO$_3^{2-}$ $\leftrightarrow$ (UO$_2$)(CO$_3$)$_3^{4-}$</td>
<td>R22</td>
<td>Log$K_e$ = 21.70</td>
</tr>
<tr>
<td>2UO$_2^{2+}$ + CO$_3^{2-}$ + 3H$_2$O $\leftrightarrow$ 3H$^+$ + (UO$_2$)$_2$(CO$_3$)(OH)$_3^-$</td>
<td>R23</td>
<td>Log$K_e$ = -1.18</td>
</tr>
<tr>
<td>UO$_2^{2+}$ + SO$_4^{2-}$ $\leftrightarrow$ (UO$_2$)(SO$_4$)(aq)</td>
<td>R24</td>
<td>Log$K_e$ = 2.95</td>
</tr>
<tr>
<td>UO$_2^{2+}$ + 2SO$_4^{2-}$ $\leftrightarrow$ (UO$_2$)(SO$_4$)$_2^{2-}$</td>
<td>R25</td>
<td>Log$K_e$ = 4.00</td>
</tr>
<tr>
<td>Reaction</td>
<td>#</td>
<td>Reaction Constants</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------</td>
<td>--------------------</td>
</tr>
<tr>
<td><strong>Aqueous Complexation Reactions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2H^+ + UO_2^{2+} + PO_4^{3-} \leftrightarrow H_2(UO_2)(PO_4)^+$</td>
<td>(R26)</td>
<td>$LogK_e = 23.20$</td>
</tr>
<tr>
<td>$3H^+ + UO_2^{2+} + PO_4^{3-} \leftrightarrow H_3(UO_2)(PO_4)^{3+}$</td>
<td>(R27)</td>
<td>$LogK_e = 22.90$</td>
</tr>
<tr>
<td>$Ca^{2+} + 4H^+ + UO_2^{2+} + 2PO_4^{3-} \leftrightarrow CaH_4(UO_2)(PO_4)_2^{2+}$</td>
<td>(R28)</td>
<td>$LogK_e = 45.24$</td>
</tr>
<tr>
<td>$Ca^{2+} + 5H^+ + UO_2^{2+} + 2PO_4^{3-} \leftrightarrow CaH_5(UO_2)(PO_4)_2^{3+}$</td>
<td>(R29)</td>
<td>$LogK_e = 46.00$</td>
</tr>
<tr>
<td>$H^+ + CO_3^{2-} \leftrightarrow HCO_3^-$</td>
<td>(R30)</td>
<td>$LogK_e = 10.32$</td>
</tr>
<tr>
<td>$2H^+ + CO_3^{2-} \leftrightarrow H_2CO_3$</td>
<td>(R31)</td>
<td>$LogK_e = 16.67$</td>
</tr>
<tr>
<td>$H^+ + SO_4^{2-} \leftrightarrow HSO_4^-$</td>
<td>(R32)</td>
<td>$LogK_e = 1.99$</td>
</tr>
<tr>
<td>$H^+ + PO_4^{3-} \leftrightarrow HPO_4^{2-}$</td>
<td>(R33)</td>
<td>$LogK_e = 12.35$</td>
</tr>
<tr>
<td>$2H^+ + PO_4^{3-} \leftrightarrow H_2PO_4^-$</td>
<td>(R34)</td>
<td>$LogK_e = 19.55$</td>
</tr>
<tr>
<td>$3H^+ + PO_4^{3-} \leftrightarrow H_3PO_4$</td>
<td>(R35)</td>
<td>$LogK_e = 21.75$</td>
</tr>
<tr>
<td><strong>Precipitation/Dissolution Reactions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Ca^{2+} + SO_4^{2-} \leftrightarrow CaSO_4\ (s)$</td>
<td>(R36)</td>
<td>$LogK_e = 4.62$</td>
</tr>
<tr>
<td>$Ca^{2+} + CO_3^{2-} \leftrightarrow CaCO_3\ (s)$</td>
<td>(R37)</td>
<td>$LogK_e = 8.48$</td>
</tr>
<tr>
<td>$5Ca^{2+} + 3PO_4^{3-} + H_2O \leftrightarrow H^+ + Ca_3(OH)(PO_4)_3\ (s)$</td>
<td>(R38)</td>
<td>$LogK_e = 40.47$</td>
</tr>
<tr>
<td>$Fe^{2+} + CO_3^{2-} \leftrightarrow FeCO_3\ (s)$</td>
<td>(R39)</td>
<td>$LogK_e = 10.50$</td>
</tr>
<tr>
<td>$Ca^{2+} + 2UO_2^{2+} + 2PO_4^{3-} \leftrightarrow Ca(UO_2)(PO_4)_2\ (s)$</td>
<td>(R40)</td>
<td>$LogK_e = 48.61$</td>
</tr>
<tr>
<td>$4Ca^{2+} + H^+ + 3PO_4^{3-} \leftrightarrow CaH_2(PO_4)_3\ (s)$</td>
<td>(R41)</td>
<td>$LogK_e = 48.20$</td>
</tr>
<tr>
<td>$Ca^{2+} + H^+ + PO_4^{3-} \leftrightarrow CaH(PO_4)\ (s)$</td>
<td>(R42)</td>
<td>$LogK_e = 19.30$</td>
</tr>
<tr>
<td>$Ca^{2+} + 2H_2O \leftrightarrow 2H^+ + Ca(OH)_2\ (s)$</td>
<td>(R43)</td>
<td>$LogK_e = -21.90$</td>
</tr>
<tr>
<td>$3Fe^{2+} + 2PO_4^{3-} \leftrightarrow Fe_3(PO_4)_2\ (s)$</td>
<td>(R44)</td>
<td>$LogK_e = 33.30$</td>
</tr>
<tr>
<td>$Fe^{2+} + 2H_2O \leftrightarrow 2H^+ + Fe(OH)_2\ (s)$</td>
<td>(R45)</td>
<td>$LogK_e = -12.10$</td>
</tr>
<tr>
<td>$UO_2^{2+} + 2H_2O \leftrightarrow 2H^+ + (UO_2)(OH)_2\ (s)$</td>
<td>(R46)</td>
<td>$LogK_e = -5.40$</td>
</tr>
<tr>
<td>$UO_2^{2+} + CO_3^{2-} \leftrightarrow (UO_2)(CO_3)\ (s)$</td>
<td>(R47)</td>
<td>$LogK_e = 14.11$</td>
</tr>
<tr>
<td>$Fe^{2+} + 2UO_2^{2+} + 2PO_4^{3-} \leftrightarrow Fe(UO_2)(PO_4)_2\ (s)$</td>
<td>(R48)</td>
<td>$LogK_e = 46.00$</td>
</tr>
<tr>
<td>$H^+ + UO_2^{2+} + PO_4^{3-} \leftrightarrow H(UO_2)(PO_4)\ (s)$</td>
<td>(R49)</td>
<td>$LogK_e = 25.00$</td>
</tr>
</tbody>
</table>
Table 6.3. Initial and Boundary Compositions of Recharge Water and Pore Water in the Tailings and Regions outside of the Tailing Pile

<table>
<thead>
<tr>
<th>Species</th>
<th>Outside Tailing Pile</th>
<th>Inside Tailing Pile</th>
<th>Recharge Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ca(^{2+})</td>
<td>10(^{-2})</td>
<td>10(^{-2})</td>
<td>10(^{-3})</td>
</tr>
<tr>
<td>CO(_3^{2-})</td>
<td>1.5 x 10(^{-3})</td>
<td>10(^{-2})</td>
<td>1.5 x 10(^{-3})</td>
</tr>
<tr>
<td>UO(_2^{2+})</td>
<td>10(^{-7})</td>
<td>5.0 x 10(^{-4})</td>
<td>10(^{-8})</td>
</tr>
<tr>
<td>PO(_4^{3-})</td>
<td>10(^{-6})</td>
<td>10(^{-6})</td>
<td>10(^{-6})</td>
</tr>
<tr>
<td>SO(_4^{2-})</td>
<td>2.0 x 10(^{-2})</td>
<td>2.0 x 10(^{-1})</td>
<td>10(^{-4})</td>
</tr>
<tr>
<td>H(^+)</td>
<td>10(^{-7})</td>
<td>3.5 x 10(^{-2})</td>
<td>10(^{-7})</td>
</tr>
<tr>
<td>Fe(^{2+})</td>
<td>10(^{-3})</td>
<td>2.0 x 10(^{-1})</td>
<td>10(^{-3})</td>
</tr>
</tbody>
</table>

Fig. 6.11. Pressure Head Contours (top) and Velocity Fields (bottom) for Example 9
Fig. 6.12. Distribution of pH at Various Time Steps for Example 9
Fig. 6.13. Concentration Contours of Precipitated Carbonate (calcite) at Various Time Steps for Example 9
Fig. 6.14. Concentration Contours of Precipitated Sulfate (gypsum) at Various Time Steps for
Example 9
Fig. 6.15. Concentration Contours of Dissolved Uranium at Various Time Steps for Example 9
6.10. Example 10: Flow and Advective-Dispersive-Reactive Transport Problem (1)

Example 10 involves steady flow and combined steady and transient transport in a rectangular domain which has the same size and discretization as that of Example 1 (bottom of Fig. 6.1). The length unit in this example is "dm". The pre-initial pressure head is assumed as a hydrostatic distribution with the water table located at \( z = 20 \) dm. The flow boundary conditions are identical as those applied in Example 1: the rainfall rate on the top surface is 0.06 dm/day and the Dirichlet boundary for \( 0 \) dm < \( z < 20 \) dm on the right side is the water table held at \( z = 20 \) dm.

The problem domain contains two types of materials as shown in Fig. 6.16. The unsaturated characteristic hydraulic properties of both materials are described by the same equations associated with the identical values of parameters as in Example 1 (Eqs. 6.1 and 6.2). Material 1 has a saturated hydraulic conductivity of \( K_{xx} = K_{zz} = 0.1 \) dm/day and \( K_{yy} = K_{xy} = K_{xz} = K_{yz} = 0 \) dm/day. Material 2 has a saturated hydraulic conductivity of \( K_{xx} = K_{zz} = 0.085 \) dm/day and \( K_{yy} = K_{xy} = K_{xz} = K_{yz} = 0 \) dm/day. Figure 6.17 shows the pressure head contours and velocity fields of steady state results. Based on flow results, both steady and transient transport, including complexation and adsorption/desorption reactions, will be simulated in this example.

The biogeochemical reactions are given in Table 6.4. Figure 6.16 shows the transport boundary conditions designed in this example: Dirichlet B.C. on the top surface from A to B; variable B.C. - I on the top surface from B to C; variable B.C. - II on the right side; and no flux conditions on the bottom, front, back and left sides. Boundary conditions for both steady and transient transport simulations can be found in Tables 6.5a and 6.5b, respectively. Material 1 has a longitudinal dispersivity of 6 dm and a lateral dispersivity of 3 dm. Material 2 has a longitudinal dispersivity of 5 dm and a lateral dispersivity of 2.5 dm. No molecular diffusion is considered for both materials. The pre-initial conditions for steady transport are assumed as: \( Ca^{2+} = 10^{-3} \) M, \( CO_{3}^{2-} = 10^{-4} \) M, \( NpO_{2}^{+} = 0 \) M, and \( H^{+} = 10^{-8} \) M through the entire domain. One adsorbing site is located between \( z = 50 \) dm and \( z = 70 \) dm, and its concentration is 0.001 M for both right and left regions except the central part has a high one of 0.025 M (Fig. 6.16). The initial time step size is 0.25 day, and each subsequent time step size is increased with a multiplier of 1.5 until a maximum time step size of 32.0 day is reached. A total of 100 time steps resulting in a total simulation time of 2880.37 days is performed in this example.

This example is designed to present a hypothetic clean-up process of a contaminated domain by continuously recharging clean water through the top surface (steady flow). The steady transport simulation is performed first to obtain the initial status of contaminants distributions based on the boundary conditions described in Table 6.5a. Then, the transient transport simulation is performed with no contaminant on each boundary side (Table 6.5b). Figures 6.18 and 6.19 show the dissolved neptunium distributions and pH values at different time steps. Before recharging the clean water, only the region with the higher adsorbing-site concentration has a lower dissolved neptunium concentration due to the adsorption effect. Most of the regions are contaminated with a high concentration of neptunium. As clean water continuously recharges through the top surface, the dissolved neptunium concentration becomes lower as \( t \) increases (Fig. 6.18). On the other hand, the higher pH value is initially throughout the domain due to the higher concentrations of calcium. Only the region with the higher adsorbing-site concentration has the lower pH value because of the adsorption effect. As clean water is continuously recharged through the top surface, the pH values become lower as \( t \) increases larger (Fig. 6.19).
Table 6.4. Reaction Network for Example 10

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aqueous Complexation Reactions</strong></td>
<td></td>
</tr>
<tr>
<td>(\text{H}_2\text{O(aq)} \leftrightarrow \text{H}^+ + \text{OH}^-) (R1)</td>
<td>(\log K_e = -14.00)</td>
</tr>
<tr>
<td>(\text{Ca}^{2+} + \text{CO}_3^{2-} \leftrightarrow \text{CaCO}_3) (aq) (R2)</td>
<td>(\log K_e = 3.22)</td>
</tr>
<tr>
<td>(\text{Ca}^{2+} + \text{H}^+ + \text{CO}_3^{2-} \leftrightarrow \text{CaHCO}_3^+) (R3)</td>
<td>(\log K_e = 11.43)</td>
</tr>
<tr>
<td>(\text{Ca}^{2+} + 2\text{H}_2\text{O} \leftrightarrow 2\text{H}^+ + \text{Ca(OH)}_2) (aq) (R4)</td>
<td>(\log K_e = -12.58)</td>
</tr>
<tr>
<td>(\text{H}^+ + \text{CO}_3^{2-} \leftrightarrow \text{HCO}_3^-) (R5)</td>
<td>(\log K_e = 10.32)</td>
</tr>
<tr>
<td>(2\text{H}^+ + \text{CO}_3^{2-} \leftrightarrow \text{H}_2\text{CO}_3) (R6)</td>
<td>(\log K_e = 16.67)</td>
</tr>
<tr>
<td>(\text{NpO}_2^+ + \text{H}_2\text{O} \leftrightarrow \text{H}^+ + \text{NpO}_2\text{(OH)}) (aq) (R7)</td>
<td>(\log K_e = -8.85)</td>
</tr>
<tr>
<td>(\text{NpO}_2^+ + \text{CO}_3^{2-} \leftrightarrow \text{NpO}_2\text{(CO}_3^-)) (R8)</td>
<td>(\log K_e = 5.60)</td>
</tr>
<tr>
<td>(\text{NpO}_2^+ + 2\text{CO}_3^{2-} \leftrightarrow \text{NpO}_2\text{(CO}_3^-)_2) (R9)</td>
<td>(\log K_e = -3.50)</td>
</tr>
<tr>
<td><strong>Adsorption/Desorption Reactions</strong></td>
<td></td>
</tr>
<tr>
<td>(=\text{SOH} \leftrightarrow \text{H}^+ + =\text{SO}^-) (R10)</td>
<td>(\log K_e = -10.30)</td>
</tr>
<tr>
<td>(=\text{SOH} + \text{H}^+ \leftrightarrow =\text{SOH}_2^+) (R11)</td>
<td>(\log K_e = 5.40)</td>
</tr>
<tr>
<td>(\text{NpO}_2^+ + \text{H}_2\text{O} + =\text{SOH} \leftrightarrow \text{H}^+ + (\text{NpO}_2\text{(OH)})(=\text{SOH})) (R12)</td>
<td>(\log K_e = -3.50)</td>
</tr>
</tbody>
</table>

Note: "=" indicates sorbed species, for example, =SOH.

---

Table 6.5a. Boundary Conditions for Steady Transport in Example 10

<table>
<thead>
<tr>
<th>Unit: M</th>
<th>Ca(^{2+})</th>
<th>CO(_3^{2-})</th>
<th>NpO(_2^+)</th>
<th>H(^+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirichlet B.C.</td>
<td>(10^{-3})</td>
<td>(10^{-4})</td>
<td>(10^{-5})</td>
<td>(10^{-8})</td>
</tr>
<tr>
<td>Variable B.C. - I</td>
<td>(10^{-3})</td>
<td>(10^{-4})</td>
<td>(10^{-5})</td>
<td>(10^{-8})</td>
</tr>
<tr>
<td>Variable B.C. - II</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 6.5b. Boundary Conditions for Transient Transport in Example 10

<table>
<thead>
<tr>
<th>Unit: M</th>
<th>Ca(^{2+})</th>
<th>CO(_3^{2-})</th>
<th>NpO(_2^+)</th>
<th>H(^+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirichlet B.C.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Variable B.C. - I</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Variable B.C. - II</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Fig. 6.16. Problem Definition of Example 10: Transport Boundary Conditions, Regions of Different Material Types, and Information of the Adsorbing Site (SOH)
Fig. 6.17. Steady Flow Results of Example 10: Pressure Head contours (top) and Velocity Fields (bottom)
Fig. 6.18. Concentration Contours of Dissolved Neptunium at Various Time Steps for Example 10
Fig. 6.19. Distribution of the pH Values at Various Time Steps for Example 10
6.11. Example 11: Flow and Advective-Dispersive-Reactive Transport Problem (2)

Example 11 is a modification of the previous example to perform steady/transient flow and transient transport. The major difference between these two examples is the water table is now initially located at z = 20 dm and will change due to recharging on the top surface. This initial condition for transient simulation is determined from steady flow simulations with zero recharge. With the initial condition determined, transient flow simulations are carried out using the same recharge as that in Example 10. The problem domain of this example is the same as that used in the previous example as shown in Fig. 6.16. The material types and information, hydraulic conductivities, and dispersivities are also retained unchanged in this example.

The pre-initial pressure head is assumed hydrostatic with the water table located at z = 20 dm. The rainfall rate on the top surface is 0.0 dm/day for steady flow and changes to 0.06 dm/day for the subsequent transient flow. The initial time step size is 0.25 day, and each subsequent time step size is increased with a multiplier of 2.0 until a maximum time step size of 128.0 day is reached. A total of 200 time steps resulting in a total simulation time of 24575.75 days is performed in this example. Figure 6.20 depicts the simulated water table at various time steps. While the water table is changing, transient transport is performed in this example. The biogeochemical reactions considered in this example are identical to those studied in the previous example (Table 6.4). Boundary conditions for each corresponding type in Fig. 6.16 can be found in Table 6.6. The initial conditions for each component species are Ca^{2+} = 10^{-3} M, CO_3^{2-} = 10^{-4} M, NpO_2^{+} = 10^{-8} M, and H^+ = 0 M through the entire domain.

This example is designed to present an originally clean domain to be contaminated due to the continuous recharging of contaminated water through the top surface. Initially, the water table is located at z = 20 dm, transient flow simulations show the gradual rising of the water table (Fig. 6.20). Meanwhile, neptunium, calcium, and carbonate will enter the domain along with the recharging through the top surface. Figures 6.21 and 6.22 show the dissolved neptunium distributions and pH values at different time steps. The dissolved neptunium concentration becomes higher as t gets larger. The adsorbing site provides the buffer to adsorb neptunium which makes the central region (higher SOH) have the lowest value of neptunium concentration. Due to the existence of the higher concentration calcium, the high pH values are present only initially throughout the domain, and only the region with the higher adsorbing-site concentration has lower pH value due to the adsorption effect. Since calcium is continuously recharged into the system, the pH values become higher as t increases (Fig. 6.22).
Table 6.6. Boundary Conditions for Transient Transport in Example 11

<table>
<thead>
<tr>
<th>Unit: M</th>
<th>Ca^{2+}</th>
<th>CO_3^{2-}</th>
<th>NpO_2^{+}</th>
<th>H^+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirichlet B.C.</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
<td>$10^{-5}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>Variable B.C. - I</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
<td>$10^{-5}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>Variable B.C. - II</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Fig. 6.20. Variations of the Water Table at Various Times for Example 11
Fig. 6.21. Concentration Contours of Dissolved Neptunium at Various Time Steps for Example 11
Fig. 6.22. Distribution of the pH Values at Various Time Steps for Example 11

Example 12 is modified from Example 10 to perform both steady and transient simulations for coupled flow and transport. The computed results of steady flow and transport will be used as initial conditions for the subsequent transient flow and transport, respectively. The biogeochemical reactions considered in this example are identical to that of Example 10. The problem domain of this example is the same as that used in the previous example as shown in Fig. 6.16. The material types and information, hydraulic conductivities, and dispersivities are also retained unchanged in this example.

The pre-initial pressure head is assumed hydrostatic with the water table located at z = 20 dm. The rainfall rate on the top surface is 0.06 dm/day for steady flow and changes to 0.0 dm/day, i.e., no recharge, for the subsequent transient flow simulations. Therefore, the transient flow performed in this example presents the descending of the water table due to the ceasing of rainfall. A constant time step size of 0.25 day is used for 10000 time steps resulting in a total simulation time of 2500 days. As shown in Fig. 6.23, the water table is falling down faster in the early time steps, such as t from 0 day to 250 days, than the later time steps, such as t from 750 days to 1000 days. The reason is the hydraulic gradient is decreasing as the water table falls down, which impedes the movement of water.

The biogeochemical reactions considered in this example is identical to those studied in Example 10 (Table 6.4). As the boundary conditions listed in Table 6.7a, the steady transport is to obtain the initial distributions of contaminants based on the steady flow results. The pre-initial conditions for each component species are Ca²⁺ = 0 M, CO₃²⁻ = 0 M, NpO₂⁺ = 0 M, and H⁺ = 0 M. Without recharging of clean water on the top surface, transient transport is performed by assuming zero incoming contaminant concentration on every specified boundary side (Table 6.7b).

Figures 6.24 and 6.25 show the dissolved neptunium distributions and pH values at different time steps. Initially, most regions are contaminated with the high concentration of neptunium (10⁻⁵ M) due to the continuous recharging of contaminated water, and only the region of the adsorbing site has a low value of neptunium because of the adsorption effect. When rainfall stops, the concentration of neptunium becomes lower as t gets larger (Fig. 6.24) because the contaminated water is moving out of the domain associated with the dropping water table (Fig. 6.23). As one can imagine, without the recharging of clean water into the system, the clean-up process is much slower than that shown in Example 10 with the recharging of clean water. Initially, the high pH values were found throughout the domain due to the high calcium concentration. As shown in Fig. 6.25, only the region with the higher adsorbing-site concentration has a lower pH value due to the adsorption effect. Without the recharging of clean water, the processing time needed to neutralize a system is much longer than that shown in Example 10 (at t = 1920 days in Fig. 6.19 vs. at t= 2500 days in Fig. 6.25).
Table 6.7a. Boundary Conditions for Steady Transport in Example 12

<table>
<thead>
<tr>
<th></th>
<th>Ca\textsuperscript{2+}</th>
<th>CO\textsubscript{3}\textsuperscript{2-}</th>
<th>NpO\textsubscript{2}\textsuperscript{+}</th>
<th>H\textsuperscript{+}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirichlet B.C.</td>
<td>10\textsuperscript{-3}</td>
<td>10\textsuperscript{-4}</td>
<td>10\textsuperscript{-5}</td>
<td>10\textsuperscript{-8}</td>
</tr>
<tr>
<td>Variable B.C. - I</td>
<td>10\textsuperscript{-3}</td>
<td>10\textsuperscript{-4}</td>
<td>10\textsuperscript{-5}</td>
<td>10\textsuperscript{-8}</td>
</tr>
<tr>
<td>Variable B.C. - II</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 6.7b. Boundary Conditions for Transient Transport in Example 12

<table>
<thead>
<tr>
<th></th>
<th>Ca\textsuperscript{2+}</th>
<th>CO\textsubscript{3}\textsuperscript{2-}</th>
<th>NpO\textsubscript{2}\textsuperscript{+}</th>
<th>H\textsuperscript{+}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirichlet B.C.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Variable B.C. - I</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Variable B.C. - II</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Fig. 6.23. Variations of the Water Table at Various Times for Example 12
Fig. 6.24. Concentration Contours of Dissolved Neptunium at Various Time Steps for Example 12
Fig. 6.25. Distributions of the pH Values at Various Time Steps for Example 12

This problem involves matrix diffusion-reactive transport through fractured media. Three cases are simulated in this example, the first one is identical to the setup of problem 3 in LEHG01.1 (EX-f5-3.INP as the input file) in which the flow was not simulated, the second one considers the effect of precipitation/dissolution on this coupled system EX-f5-1.INP as the input file), and the third one does consider this effect (EX-f5-2.INP as the input file). The first case is used to verify HYDROGEOCHEM 5.0 with LEHG01.1. In the second case, the flow and reactive transport are coupled due to the effect of precipitation on hydraulic conductivity, porosity, and hydrodynamic dispersion. To reflect the effect of mineral precipitations on this coupled system, the value of the fractal exponent for adjusting the hydraulic conductivity is set as 2.5 (PROPF(14,I) of data set 6), the value of the power in the Archie's law for adjusting the dispersion coefficient is set as 2.0 (PROP(9,I) of data set 6), and the molar volumes of portlandite, Ca(OH)$_2$ (s), and calcite, CaCO$_3$ (s), are set as 0.03 dm$^3$/mol and 0.037 dm$^3$/mol (data set 11), respectively. In the third case, although the effects of precipitation/dissolution on conductivity and dispersion are not considered, the solute transport and biogeochemical reaction still need to be coupled because the precipitation/dissolution reactions, which are non-homogeneous, are involved in the system.

The domain of interest was based on a longitudinal cross-section of a 0.76 meter high column with a 120 μm wide fracture in the center, running the length of the column. Due to symmetry, one-half of the cross-section can be used; the resulting domain (top of Fig. 6.26) is 7.6 dm high by 0.325 dm wide by 1.0 dm deep with a 6 x 10$^{-4}$ dm wide fracture on the left hand side. The fracture is simulated by using specific elements having a moisture content, $\theta$, of 1.0 and a bulk density $\rho_b$ of 1.0 x 10$^{-20}$ kg/dm$^3$ (non-zero bulk density is required for any material in HYDROGEOCHEM 5.0). The matrix elements have a moisture content and bulk density of 0.35 and 1.7 kg/dm$^3$, respectively.

The grid is made up of 21 rows by 17 columns of rectangular elements, with the three left-hand columns of elements representing the fracture and the remaining elements representing the matrix (bottom of Fig. 6.26). The fracture element widths(x-direction) are 1.2 x 10$^{-4}$ dm in the first column and 2.4 x 10$^{-4}$ dm in the second and third columns. The matrix element widths gradually increase from being equal to the fracture element width, 2.4 x 10$^{-4}$ dm, at the fracture side to 0.1 dm at the opposite side. The heights(z-direction) of rows of elements gradually increase from 6 x 10$^{-4}$ dm to 0.2 dm, from the top to the bottom.

For flow simulations in the second and third cases, both top and bottom surfaces are held by the Dirichlet head of 7.615 dm and 7.6 dm, respectively. One material type is used to describe the matrix zone. In order to have parabolic velocity distributions in the fracture, three material types are applied for the three columns used to describe the fracture zone (bottom of Fig. 6.26). The saturated hydraulic conductivity in the z-direction for each material are $K_{zz} = 2.16 \times 10^{-6}$ dm/hr for material 1, $K_{zz} = 237.0$ dm/hr for material 2, $K_{zz} = 222.0$ dm/hr for material 3, and $K_{zz} = 90.0$ dm/hr for material 4 (assuming $K_{xx} = K_{yy} = K_{xy} = K_{xz} = K_{yz} = 0$ for all four material types).

The upper boundary of the domain is assumed to maintain a constant concentration on the Dirichlet boundary. Neumann boundaries are used for the other three boundaries with $\partial C/\partial x = 0$ for the two vertical boundaries and $\partial C/\partial z = 0$ for the bottom boundary. The molecular diffusion coefficient is 3.6 x 10$^{-4}$ dm$^2$/hr. The tortuosity of the matrix cells is assumed as 0.1. Longitudinal dispersivity ($\alpha_L$) for all elements is set to 7.6 dm while lateral dispersivity ($\alpha_T$) is set to 0.0 dm for all elements. Only one conservative tracer is considered in the first cases. The biogeochemical reactions in the second and third cases are given in Table 6.8. Table 6.9 lists initial and boundary conditions assumed for simulations. A time step size of 0.24 hour was used; a total of six days (144 hours) is simulated using 600 time steps.
Figure 6.27 depicts the concentration profiles of the first case simulated by HYDROGEOCHEM 5.0. Compared with the output and figure obtained from LEHG 1.1, identical results were found between both codes. For the second and third cases, Figs. 6.28 and 6.29 show the concentration profiles of calcite and conservative tracer, respectively, vs. the horizontal distance from the center of fracture at different elevations for t = 144 hours. There is a difference between Figs. 6.27 and Fig. 6.28(b) because the flow field was given as fixed in the first case rather than computed from the flow setup in the third case.

In the second case the precipitation effect is considered, the diffusion transport is diminished due to the calcite precipitation which impedes the diffusion of calcium and carbonate into the matrix zone, resulting less calcite precipitations. This is confirmed by comparing Figs. 6.28(a) and 6.28(b). The concentration profiles of calcite with the precipitation/dissolution effect in the matrix zone (Fig. 6.28(a)) are much smaller than those without the precipitation/dissolution effect (Fig. 6.28(b)). On the other hand, since smaller amounts of contaminant can diffuse into the matrix zone, a larger amount of contaminant will tarry in the fracture zone. This can be found as the concentrations of the conservative tracer in the fracture zone with the precipitation effect (data points on the vertical-axis of Fig. 6.29(b)) are much higher than those without the effect (the corresponding data points on the vertical-axis of Fig. 6.29(a)). As explained before, the concentrations of tracer in the matrix zone with the precipitation effect (data points not on the vertical-axis of Fig. 6.29(b)) are lower than those without the precipitation effect (the corresponding data points not on the vertical-axis of Fig. 6.29(a)). On the effect of precipitation on the flow, the change of velocity fields is not significant after comparing output files of both cases.

The velocity at node 43 is only changed from 0.44803 dm/hr to 0.44799 dm/hr due to the hydraulic conductivity diminished by this precipitation effect. This tiny change of flow velocity is due to the precipitation developed in the fracture zone being only a relative small portion of the entire pore space in the fracture zone.

Although the flow fields are not significantly affected by the precipitation, the contaminant distributions show a major impact from this effect. The mineral precipitation therefore plays an important role in solving non-homogeneous reactive transport problems. HYDROGEOCHEM 5.0 has demonstrated the ability to successfully handle this precipitation/dissolution effect on both flow and reactive transport.
Fig. 6.26. Problem definition (top) and discretization (bottom) for Example 13
Table 6.8. Reaction Network for Example 13

<table>
<thead>
<tr>
<th>Reaction</th>
<th>#</th>
<th>Reaction Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aqueous Complexation Reactions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H$_2$O(aq) ↔ H$^+$ + OH$^-$</td>
<td>(R1)</td>
<td>$\log K_e = -14.00$</td>
</tr>
<tr>
<td>Ca$^{2+}$ + CO$_3^{2-}$ ↔ CaCO$_3$ (aq)</td>
<td>(R2)</td>
<td>$\log K_e = 3.00$</td>
</tr>
<tr>
<td>Ca$^{2+}$ + H$^+$ + CO$_3^{2-}$ ↔ CaHCO$_3^+$</td>
<td>(R3)</td>
<td>$\log K_e = 11.60$</td>
</tr>
<tr>
<td>Ca$^{2+}$ + H$_2$O ↔ H$^+$ + CaOH$^+$</td>
<td>(R4)</td>
<td>$\log K_e = -12.20$</td>
</tr>
<tr>
<td>H$^+$ + CO$_3^{2-}$ ↔ HCO$_3^-$</td>
<td>(R5)</td>
<td>$\log K_e = 10.20$</td>
</tr>
<tr>
<td>2H$^+$ + CO$_3^{2-}$ ↔ H$_2$CO$_3$</td>
<td>(R6)</td>
<td>$\log K_e = 16.50$</td>
</tr>
<tr>
<td><strong>Precipitation/Dissolution Reactions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ca$^{2+}$ + 2H$_2$O ↔ 2H$^+$ + Ca(OH)$_2$ (s)</td>
<td>(R7)</td>
<td>$\log K_e = -21.90$</td>
</tr>
<tr>
<td>Ca$^{2+}$ + CO$_3^{2-}$ ↔ CaCO$_3$ (s)</td>
<td>(R8)</td>
<td>$\log K_f = 3.50$; $\log K_b = -5.00$</td>
</tr>
</tbody>
</table>

Table 6.9. Initial and Boundary Conditions for Example 13

<table>
<thead>
<tr>
<th>Unit: M</th>
<th>Ca$^{2+}$</th>
<th>CO$_3^{2-}$</th>
<th>Tracer</th>
<th>H$^+$</th>
<th>CaCO$_3$(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Conditions</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>1.0</td>
<td>0.0</td>
<td>$8.642 \times 10^{-5}$</td>
</tr>
<tr>
<td>Dirichlet B.C. (top surface)</td>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
<td>1.0</td>
<td>0.0</td>
<td>-----</td>
</tr>
<tr>
<td>variable B.C. (others)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-----</td>
</tr>
</tbody>
</table>
Fig. 6.27. Concentration profiles of Conservative Tracer in the First Case for Example 13 at $t = 144$ hours (to compare with Fig. 6.10 in the report of LEHGC 1.1)
Fig. 6.28. Concentration Profiles of Calcite: (a) Case Two with Considering the Precipitation Effect and (b) Case Three without Considering the Precipitation Effect for Example 13 at $t = 144$ hours
Fig. 6.29. Concentration Profiles of Conservative tracer: (a) Case Two with Considering the Precipitation Effect and (b) Case Three without Considering the Precipitation Effect for Example 13 at $t = 144$ hours
Example 14: 1-D Biogeochemical Transport Verification

Example 14 involves the transient simulation of biogeochemical transport in horizontal soil column shown in Fig. 6.30. The domain of interest has a size of 150 cm x 1 cm x 1 cm. The flow column contains water with an initial concentration of aqueous species \( C(x,0) = 0 \) mmol/cm³ of water volume. It also contains soil with a bulk density of 1.5 g/cm³ and sorbed species \( S(x,0) = 0 \) mmol/cm³ of water volume. A steady flow field is assumed with a constant Darcy’s velocity \( V = 1 \) cm/day and an effective moisture content of 0.2. The molecular diffusion coefficient is assumed to be zero. The boundary condition at \( x = 0 \) is flow-in boundary condition and incoming concentration of species \( C \) is 1.0 mmol/cm³. Natural boundary condition is put at \( x = 150 \) cm. The domain is discretized into 100 equal size elements (1.5 cm x 1 cm x 1 cm each). A constant time step size of 0.1 day is used for the simulation. The simple equilibrium reaction considered is

\[
C \leftrightarrow \text{S}, \quad K_d = 0.1333 \tag{6.3}
\]

where \( K_d \) is the partition coefficient.

Simulations with three schemes: (a) FEM (Finite Element Method) in advective form, (b) Hybrid Lagrangian-Eulerian (LE) FEM for interior nodes plus FEM in advective form for boundary nodes, and (c) Hybrid Lagrangian-Eulerian FEM, were performed individually. Simulation results are compared with the analytical solutions given by Lindstorm and Freed (1967) in Table 6.10 and Fig. 6.31. Figure 6.31 illustrates the concentration profiles for biogeochemical \( C \) after 8 days.

The grid Peclet number \( P_e = V\Delta x/D = V\Delta x/(\alpha_L V) = \Delta x/\alpha_L \) in the simulation corresponds to the three longitudinal dispersivities, \( \alpha_L = 0.1 \) cm., \( \alpha_L = 10 \) cm, and \( \alpha_L = 50 \) cm, are \( P_e = 15 \), \( P_e = 0.15 \), and \( P_e = 0.03 \), respectively. Grid Courant number \( (\Delta t/\Delta x) \) is less than 1. Discrepancies among the three schemes are apparent as shown in Fig. 6.31.

It is seem from Table 6.10 that for advection dominant problem (large Peclet number), the Hybrid Lagrangian-Eulerian schemes gives more accurate results even though it shows some amount of numerical diffusion. However, for dispersion dominant problem (small Peclet number), accuracy of the Hybrid Lagrangian-Eulerian scheme is bad because of the inaccurate treatment of boundary fluxes. Although conventional FEM presents high accuracy, it is susceptible to numerical diffusion and limitations on the size of the Courant number. Therefore, for dispersion dominant problems with flux boundary conditions, it is recommended to apply the Hybrid LE scheme for interior nodes plus FEM in advective form for boundary nodes. For advection dominant problem, the LE plus FEM scheme is also recommended because it ensured mass conservation (Yeh et al., 2003)
Fig. 6.30. The Problem Domain and the Discretization for Example 14

Table 6.10. $R^2$ Values Based on the Analytical and Numerical Solutions with Different Values of Dispersivity

<table>
<thead>
<tr>
<th>$\alpha_L$ (cm)</th>
<th>Peclet number</th>
<th>$R^2$ Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FEM</td>
</tr>
<tr>
<td>0.1</td>
<td>15</td>
<td>0.9528</td>
</tr>
<tr>
<td>10</td>
<td>0.15</td>
<td>0.9997</td>
</tr>
<tr>
<td>50</td>
<td>0.03</td>
<td>0.9998</td>
</tr>
</tbody>
</table>
Fig. 6.31. Concentration Profiles of Biogeochemical C at $t = 8$ days for Example 14:
Upper: $\alpha_L = 0.1$ cm; Middle: $\alpha_L = 10$ cm; Lower: $\alpha_L = 50$ cm.
6.15. Example 15: 3-D Flow and Biogeochemical Transport with Complexation, Adsorption, Ion-exchange and Mineral Dissolution Reactions

Example 15 involves steady flow, transient reactive biogeochemical transport within a regular region with size 200 dm x 200 dm x 100 dm. The simulation region is discretized into 20 x 20 x 10 = 4000 elements connected at 21 x 21 x 11 = 4851 nodes. Each element has the same size of 10 x 10 x 10 = 1000 dm³ (Fig. 6.32).

Fig. 6.32. The Problem Domain and the Discretization for Example 15 (unit: dm)

The flow boundary conditions are given as: no flux is imposed on the left (x = 0), on the back (y = 200 dm), and at the bottom (z = 0) side of the region; pressure head is assumed to vary from zero at z = 30 dm to 30 dm at the bottom (z = 0) on the right (x = 200 dm) and front (y = 0) sides; variable boundary conditions are used elsewhere. Ponding depth is assumed zero for all variable boundary surfaces. The incoming fluxes of variable boundary are assumed equal to 1 dm/day for the top surface (variable B.C.-I in Fig. 6.33) and equal to zero for the front and right sides above the water surface (variable B.C.-II) (Fig. 6.33). The steady-state solution is to be sought. A pre-initial condition is set as h = 30-z which means the pre-initial water table is located at z = 30 dm.

The problem domain contains one type of material. The material has the saturated hydraulic conductivities of $K_{xx} = K_{yy} = K_{zz} = 1$ dm/day and $K_{xy} = K_{xz} = K_{yz} = 0$ dm/day. The unsaturated characteristic hydraulic properties of the material are described by the same equations associated with the identical values of parameters as in Example 1 (Eqs. 6.1 and 6.2).

The steady-state pressure head contours are presented in Fig. 6.34, and the velocity fields are plotted in Fig. 6.35 along a 2-D cross-section (x = 0) of the domain.
Fig. 6.33. Problem Definition for Example 15: Flow Boundary Conditions and Initial conditions

Fig. 6.34. Steady-state Pressure Head Contours for Example 15
Given the flow fields, transient biogeochemical transport, incorporating complexation, adsorption/desorption, ion-exchange and mineral dissolution reactions in a system of mixed equilibrium and kinetic reactions, is simulated in this example. Table 6.11 defines the biogeochemical formula for each species involved in the reaction system. These species are classified into aqueous species, solid species, adsorbed species, or ion-exchange site.

Figure 6.36 shows the reactive biogeochemical transport boundary conditions. No flux is imposed on the left (x = 0), on the back (y = 200 dm), and on the bottom (z = 0) side of the region; Dirichlet boundary is located on the top surface of the domain within 0 dm < x < 90 dm and 130 dm < y < 150 dm; variable B.C.-I is imposed on the top surface excluding the domain of Dirichlet boundary; variable B.C.-II is cast on the right side (x = 200 dm) and front side (y = 0). Also indicated in Fig 6.36 is the high concentration region, which is located within 0 dm < x < 100 dm, 110 dm < y < 140 dm and 80 dm < z < 90 dm.

The species concentrations on the boundary are listed in Table 6.12. The pre-initial conditions are listed in Table 6.13 for the species whose concentration is not zero. A time step size of 2 days is used and a total 100 time steps resulting in a total simulation time of 200 days is performed in this example.

The reaction network, which is listed in Table 6.14, is complex and involves a total of 33 reactions. Initially, the metal-hydroxide mineral M becomes solubilized and produces two aqueous species C₁ and C₂ through an irreversible reaction (reaction R1 in Table 6.14). In addition, adsorbing sites are formed on the surface of the mineral M (R2). The surface site commonly undergoes ionization reactions upon contact with water (R25 and R26). Species C₃ dissolves into C₄ and C₅ (R3). Then C₄, C₅, and dissolved species C₁ react to form various complexed species (R4 through R23). In turn, some of these complexed species are adsorbed on the surface of mineral M (R27 through R31). In addition, three ion-exchange sites and three ions (C₆, C₂₉, and C₃₀) are involved in the reaction system (R32 and R33). The reaction system is solved with BIOGEOCHEM, the module of HYDROGEOCHEM 5.0 to deal with the biogeochemical reaction network. Some reactions are created to make up the reaction system, for example, R2 (partition between bulk and
surface metal ions) is used to reflect the mass balance of surface sites. Some of the reaction constants are obtained from Szecsody, et al. (1994) and Yeh, et al. (1995a). To demonstrate the capability of BIOGEOCHEM, some reactions (R1, R3, R7, R28-R31, R32) are treated as kinetic reactions. For example, the dissolution of the mineral (R1) is assumed as a slow kinetic reaction.

Table 6.11. List of Species Definitions for Example 15

<table>
<thead>
<tr>
<th>Species Name</th>
<th>Chemical Formula</th>
<th>Species Name</th>
<th>Chemical Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Fe$^{3+}$</td>
<td>C22</td>
<td>Co(OH)$_3^-$</td>
</tr>
<tr>
<td>C2</td>
<td>H$^+$</td>
<td>C23</td>
<td>HEDTA$^{3-}$</td>
</tr>
<tr>
<td>C3</td>
<td>CoEDTA$^{2-}$</td>
<td>C24</td>
<td>H$_3$EDTA$^{2-}$</td>
</tr>
<tr>
<td>C4</td>
<td>Co$^{2+}$</td>
<td>C25</td>
<td>H$_3$EDTA$^-$</td>
</tr>
<tr>
<td>C5</td>
<td>EDTA$^{4+}$</td>
<td>C26</td>
<td>H$_4$EDTA</td>
</tr>
<tr>
<td>C6</td>
<td>Ca$^{2+}$</td>
<td>C27</td>
<td>OH$^-$</td>
</tr>
<tr>
<td>C7</td>
<td>CaEDTA$^{2-}$</td>
<td>C28</td>
<td>H$_2$O(l)</td>
</tr>
<tr>
<td>C8</td>
<td>CaHEDTA$^-$</td>
<td>C29</td>
<td>Mg$^{2+}$</td>
</tr>
<tr>
<td>C9</td>
<td>Ca(OH)$^+$</td>
<td>C30</td>
<td>Na$^+$</td>
</tr>
<tr>
<td>C10</td>
<td>FeEDTA$^-$</td>
<td>M</td>
<td>Fe(OH)$_3$(s)</td>
</tr>
<tr>
<td>C11</td>
<td>FeHEDTA</td>
<td>S$_1$</td>
<td>=FeOH</td>
</tr>
<tr>
<td>C12</td>
<td>FeEDTA(OH)$_2$</td>
<td>S$_2$</td>
<td>=FeO$^-$</td>
</tr>
<tr>
<td>C13</td>
<td>FeEDTA(OH)$_3$</td>
<td>S$_3$</td>
<td>=FeOH$_2$</td>
</tr>
<tr>
<td>C14</td>
<td>Fe(OH)$^{2+}$</td>
<td>S$_4$</td>
<td>=FeOH$_2$-H$_2$EDTA$^-$</td>
</tr>
<tr>
<td>C15</td>
<td>Fe(OH)$_2^+$</td>
<td>S$_5$</td>
<td>=FeOH$_2$-FeEDTA$^-$</td>
</tr>
<tr>
<td>C16</td>
<td>Fe(OH)$_3^+$</td>
<td>S$_6$</td>
<td>=FeOH$_2$-CoEDTA$^-$</td>
</tr>
<tr>
<td>C17</td>
<td>Fe(OH)$_4^-$</td>
<td>S$_7$</td>
<td>=FeO-Co$^+$</td>
</tr>
<tr>
<td>C18</td>
<td>Fe$_2$(OH)$_4$</td>
<td>S$_8$</td>
<td>=FeOH$_2$-CaEDTA$^-$</td>
</tr>
<tr>
<td>C19</td>
<td>CoHEDTA$^-$</td>
<td>site-C$_6$</td>
<td>Ca$^{2+}$</td>
</tr>
<tr>
<td>C20</td>
<td>Co(OH)$^+$</td>
<td>site-C$_{29}$</td>
<td>Mg$^{2+}$</td>
</tr>
<tr>
<td>C21</td>
<td>Co(OH)$_2$</td>
<td>site-C$_{30}$</td>
<td>Na$^{2+}$</td>
</tr>
</tbody>
</table>

Note: 1) C$_1$ ~ C$_{30}$: Aqueous Species; 2) M: Solid Species; 3) S$_1$ ~ S$_8$: Adsorbed Species; 4) site-C$_{6}$, site-C$_{29}$ and site-C$_{30}$: Ion-exchange Sites.
Table 6.12. Boundary Conditions for Biogeochemical Transport in Example 15

<table>
<thead>
<tr>
<th>Species</th>
<th>Dirichlet B.C</th>
<th>variable B.C-I</th>
<th>variable B.C-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₆</td>
<td>1.0e-3</td>
<td>1.0e-5</td>
<td>0</td>
</tr>
<tr>
<td>C₁</td>
<td>1.0e-7</td>
<td>1.0e-7</td>
<td>0</td>
</tr>
<tr>
<td>C₄</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
<td>0</td>
</tr>
<tr>
<td>C₂</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
<td>0</td>
</tr>
<tr>
<td>C₅</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
<td>0</td>
</tr>
<tr>
<td>C₂₈</td>
<td>1.0e-4</td>
<td>1.0e-4</td>
<td>0</td>
</tr>
<tr>
<td>C₁₀</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
<td>0</td>
</tr>
<tr>
<td>C₃</td>
<td>1.0e-3</td>
<td>1.0e-7</td>
<td>0</td>
</tr>
<tr>
<td>C₂₉</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
<td>0</td>
</tr>
<tr>
<td>C₃₀</td>
<td>1.0e-4</td>
<td>1.0e-5</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: The unit of species concentration is mole/dm³.
Table 6.13. Initial Conditions for Biogeochemical Transport in Example 15

<table>
<thead>
<tr>
<th>Species</th>
<th>High Concentration Region</th>
<th>Low Concentration Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₆</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>C₁</td>
<td>1.0e-7</td>
<td>1.0e-7</td>
</tr>
<tr>
<td>C₄</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>C₂</td>
<td>3.0e-5</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>C₅</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>C₂₈</td>
<td>1.0e-4</td>
<td>1.0e-4</td>
</tr>
<tr>
<td>C₁₀</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>M</td>
<td>2.0e-5</td>
<td>1.0e-7</td>
</tr>
<tr>
<td>C₃</td>
<td>1.0e-6</td>
<td>1.0e-7</td>
</tr>
<tr>
<td>C₂₉</td>
<td>1.0e-5</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>C₃₀</td>
<td>1.0e-4</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>Site-C₂₉</td>
<td>1.4e-4</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>Site-C₃₀</td>
<td>7.0e-4</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>Site-C₆</td>
<td>1.5e-4</td>
<td>1.0e-5</td>
</tr>
</tbody>
</table>

Note: The unit of species concentration is mole/dm³

Table 6.14. Reaction Network for Example 15

<table>
<thead>
<tr>
<th>Reaction</th>
<th>#</th>
<th>Reaction Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mineral Dissolution and Surface Site Formation Reactions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M ← C₁ - 3C₂</td>
<td>(R1)</td>
<td>kᵢ = 0.05</td>
</tr>
<tr>
<td>M ← S₁</td>
<td>(R2)</td>
<td>User specified partition between bulk and surface metal ions as expressed by equation (M1) in the Table 6.15</td>
</tr>
<tr>
<td><strong>Aqueous Complexation Reactions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C₃ ← C₄ + C₅</td>
<td>(R3)</td>
<td>Log k₅ = 2.03, Log k₃ = -2.00</td>
</tr>
<tr>
<td>C₆ + C₅ ← C₇</td>
<td>(R4)</td>
<td>Log K₄ = 12.32</td>
</tr>
<tr>
<td>C₅ + C₆ + C₆ ← C₈</td>
<td>(R5)</td>
<td>Log K₅ = 15.93</td>
</tr>
<tr>
<td>C₆ ← C₂ + C₉</td>
<td>(R6)</td>
<td>Log K₆ = 12.60</td>
</tr>
<tr>
<td>C₁ + C₅ ← C₁₀</td>
<td>(R7)</td>
<td>Log k₅ = 25.00, Log k₇ = -2.57</td>
</tr>
<tr>
<td>C₁ + C₂ + C₄ ← C₁₁</td>
<td>(R8)</td>
<td>Log K₈ = 29.08</td>
</tr>
</tbody>
</table>
\[
\begin{align*}
C_1 + C_5 & \rightleftharpoons C_3 + C_{12} \quad & \text{(R9)} & \log K_9^e = 19.65 \\
C_1 + C_3 & \rightleftharpoons 2C_2 + C_{13} \quad & \text{(R10)} & \log K_{10}^e = -36.30 \\
C_1 & \rightleftharpoons C_2 + C_{14} \quad & \text{(R11)} & \log K_{11}^e = -2.19 \\
C_1 & \rightleftharpoons 2C_2 + C_{15} \quad & \text{(R12)} & \log K_{12}^e = -5.67 \\
C_1 & \rightleftharpoons 3C_2 + C_{16} \quad & \text{(R13)} & \log K_{13}^e = -13.60 \\
C_1 & \rightleftharpoons 4C_2 + C_{17} \quad & \text{(R14)} & \log K_{14}^e = -21.60 \\
2C_1 & \rightleftharpoons 2C_2 + C_{18} \quad & \text{(R15)} & \log K_{15}^e = -2.95 \\
C_2 + C_4 + C_5 & \rightleftharpoons C_{19} \quad & \text{(R16)} & \log K_{16}^e = 21.40 \\
C_4 & \rightleftharpoons C_2 + C_{20} \quad & \text{(R17)} & \log K_{17}^e = -9.67 \\
C_4 & \rightleftharpoons 2C_2 + C_{21} \quad & \text{(R18)} & \log K_{18}^e = -18.76 \\
C_4 & \rightleftharpoons 3C_2 + C_{22} \quad & \text{(R19)} & \log K_{19}^e = -32.23 \\
C_2 + C_5 & \rightleftharpoons C_{23} \quad & \text{(R20)} & \log K_{20}^e = 11.03 \\
2C_2 + C_5 & \rightleftharpoons C_{24} \quad & \text{(R21)} & \log K_{21}^e = 17.78 \\
3C_2 + C_5 & \rightleftharpoons C_{25} \quad & \text{(R22)} & \log K_{22}^e = 20.89 \\
4C_2 + C_5 & \rightleftharpoons C_{26} \quad & \text{(R23)} & \log K_{23}^e = 23.10 \\
C_2 + C_{27} & \rightleftharpoons C_{28} \quad & \text{(R24)} & \log K_{24}^e = 14.00 \\
\end{align*}
\]

**Adsorption-Desorption Reactions**

\[
\begin{align*}
S_1 & \rightleftharpoons S_2 + C_2 \quad & \text{(R25)} & \log K_{25}^e = -11.60 \\
S_1 + C_2 & \rightleftharpoons S_3 \quad & \text{(R26)} & \log K_{26}^e = 5.60 \\
S_1 + 3C_2 + C_5 & \rightleftharpoons S_4 \quad & \text{(R27)} & \log K_{27}^e = 30.48 \\
S_1 + C_1 + C_2 + C_5 & \rightleftharpoons S_5 \quad & \text{(R28)} & \log k_{28}^f = 40.00, \log k_{28}^b = 2.37 \\
S_1 + C_2 + C_4 + C_5 & \rightleftharpoons S_6 \quad & \text{(R29)} & \log k_{29}^f = 30.00, \log k_{29}^b = 1.51 \\
S_1 - C_2 + C_4 & \rightleftharpoons S_7 \quad & \text{(R30)} & \log k_{30}^f = -0.99, \log k_{30}^b = 1.70 \\
S_1 + C_2 + C_5 + C_6 & \rightleftharpoons S_8 \quad & \text{(R31)} & \log k_{31}^f = 25.0, \log k_{31}^b = 1.19 \\
\end{align*}
\]

**Ion-Exchange Reactions**

\[
\begin{align*}
C_{29} + 2 \text{site-C}_{30} & \rightleftharpoons \text{site-C}_{29} + 2C_{30} \quad & \text{(R32)} & \log k_{32}^f = -0.75, \log k_{32}^b = -0.5 \\
C_6 + 2 \text{site-C}_{30} & \rightleftharpoons \text{site-C}_6 + 2C_{30} \quad & \text{(R33)} & \log K_{33}^e = 0.6 \\
\end{align*}
\]

164
Table 6.15. Matrix Decomposition for the Complex (multiple reaction types) Mixed (equilibrium and kinetic) Reaction System

<table>
<thead>
<tr>
<th>Users-Specified Algebraic Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_A N_A M_{\text{mineral}}[M] = [S_1] + [S_2] + [S_3] + [S_4] + [S_5] + [S_6] + [S_7] + [S_8] ) (M1)</td>
</tr>
<tr>
<td>The users’ specified equation is modified from Stumm and Morgan, 1996, in which ( S_A ) is the unit surface area (m(^2) g(^{-1})) of mineral, ( N_S ) is the surface site density (mol sites m(^{-2})), ( N_A ) is Avogadro’s number (mol sites per mol), ( M_{\text{mineral}} ) is the molecular weight of mineral (g mol(^{-1})).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mass-Action Equilibrium Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C(_7)) = K(_i) ( C_2 ) ( C_6 ) ( \text{(M2)} )</td>
</tr>
<tr>
<td>(C(_{10})) = K(_i) ( C_2 ) ( C_3 ) ( C_6 ) ( \text{(M3)} )</td>
</tr>
<tr>
<td>(C(_9)) = K(<em>i) ( C</em>{\frac{6}{2}} ) ( C_2 ) ( \text{(M4)} )</td>
</tr>
<tr>
<td>(C(_{11})) = K(_i) ( C_1 ) ( C_2 ) ( C_3 ) ( \text{(M5)} )</td>
</tr>
<tr>
<td>(C(_{12})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M6)} )</td>
</tr>
<tr>
<td>(C(_{13})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M7)} )</td>
</tr>
<tr>
<td>(C(_{14})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M8)} )</td>
</tr>
<tr>
<td>(C(_{15})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M9)} )</td>
</tr>
<tr>
<td>(C(_{16})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M10)} )</td>
</tr>
<tr>
<td>(C(_{17})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M11)} )</td>
</tr>
<tr>
<td>(C(_{18})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M12)} )</td>
</tr>
<tr>
<td>(C(_{19})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M13)} )</td>
</tr>
<tr>
<td>(C(_{20})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M14)} )</td>
</tr>
<tr>
<td>(C(_{21})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M15)} )</td>
</tr>
<tr>
<td>(C(_{22})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M16)} )</td>
</tr>
<tr>
<td>(C(_{23})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M17)} )</td>
</tr>
<tr>
<td>(C(_{24})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M18)} )</td>
</tr>
<tr>
<td>(C(_{25})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M19)} )</td>
</tr>
<tr>
<td>(C(_{26})) = K(_i) ( \frac{C_1}{C_2} ) ( \text{(M20)} )</td>
</tr>
<tr>
<td>(S(_2)) = K(_i) ( \frac{S_1}{C_2} ) ( \text{(M22)} )</td>
</tr>
<tr>
<td>(S(_3)) = K(_i) ( \frac{S_1}{C_2} ) ( \text{(M23)} )</td>
</tr>
<tr>
<td>(S(_4)) = K(_i) ( \frac{S_1}{C_2} ) ( \text{(M24)} )</td>
</tr>
<tr>
<td>(site- ( C(<em>{30}) ) = ( \frac{1}{K</em>{\text{site}}^{0.5}} \frac{(C_{30}) (site-C_0)^{0.5}}{(C_0)^{0.5}} ) ( \text{(M25)} )</td>
</tr>
</tbody>
</table>
Kinetic-Variable Transport Equations

\[ \frac{\partial E_1}{\partial t} = P(E_1^+ - QE_1^+ + R_1); \quad R_1 = k_1^f[C_1][C_3] - k_1^b[C_{10}] \]
where \( E_1 = C_{10}, E_1^* = C_{10} \) \hspace{1cm} (M26)

\[ \frac{\partial E_2}{\partial t} = -R_1; \quad R_1 = k_1^f[M] \]
where \( E_2 = [S_1] + [M] + [S_2] + [S_3] + [S_4] + [S_5] + [S_6] + [S_7] + [S_8] \) \hspace{1cm} (M27)

\[ \frac{\partial E_3}{\partial t} = P(E_3^+ - QE_3^+ - R_3); \quad R_3 = k_3^f[C_3] - k_3^b[C_4][C_3] \]
where \( E_3 = C_3, E_3^* = C_3 \) \hspace{1cm} (M28)

\[ \frac{\partial E_4}{\partial t} = R_{28}; \quad R_{28} = k_6^f[S_6][C_1][C_2][C_3] - k_6^b[S_7] \]
where \( E_4 = S_6 \) \hspace{1cm} (M29)

\[ \frac{\partial E_5}{\partial t} = R_{29}; \quad R_{29} = k_6^f[S_7][C_1][C_2][C_3] - k_6^b[S_8] \]
where \( E_5 = S_7 \) \hspace{1cm} (M30)

\[ \frac{\partial E_6}{\partial t} = R_{30}; \quad R_{30} = k_{30}^f \frac{[S_5][C_1]}{[C_2]} - k_{30}^b[S_5] \]
where \( E_6 = S_8 \) \hspace{1cm} (M31)

\[ \frac{\partial E_7}{\partial t} = R_{31}; \quad R_{31} = k_{31}^f[S_7][C_3][C_4] - k_{31}^b[S_9] \]
where \( E_7 = S_9 \) \hspace{1cm} (M32)

\[ \frac{\partial E_8}{\partial t} = P(E_8^+ - QE_8^+ - R_{32}); \quad R_{32} = k_{32}^f[C_{30}][site-C_{30}]^2 - k_{32}^b[C_{30}][C_{30}]^2 \]
where \( E_8 = C_{30}, E_8^* = C_{30} \) \hspace{1cm} (M33)

Component Transport Equations

\[ \frac{\partial T_1}{\partial t} = P(T_1^+ - QT_1^+) \]
where \( T_1 = [C_4] + [C_5] + [C_6] + [C_8] + [S_6] + 0.5 \text{[site-C}_{30}] \)
\[ T_1^* = [C_4] + [C_5] + [C_6] + [C_8] + [C_{20}] \] \hspace{1cm} (M34)

\[ \frac{\partial T_2}{\partial t} = P(T_2^+ - QT_2^+) \]
where \( T_2 = [C_3] + [C_4] + [C_{19}] + [C_{20}] + [C_{21}] + [C_{22}] + [S_6] + [S_7] \)
\[ T_2^* = [C_3] + [C_4] + [C_{19}] + [C_{20}] + [C_{21}] + [C_{22}] \] \hspace{1cm} (M35)
\[ \theta \frac{\partial T_i}{\partial t} = P(T_i^*) - QT_i^* \]

where \[ T_i = [C_{i3} + C_{i4} + [C_{i5}] + [C_{i6}] - 2[C_{i7}] + 2[C_{i9}] - 3[C_{i10}] - 4[C_{i11}] + 2[C_{i12}] + 2[C_{i13}] + 2[C_{i14}] - 2[C_{i15}] - 3[C_{i16}] + 2[C_{i17}] + 2[C_{i18}] + [C_{i19}] + [C_{i20}] - 2[C_{i21}] + 3[C_{i22}] + 2[C_{i23}] + 3[C_{i25}] + 4[C_{i26}] - [C_{i27}] ] \]

(M36)

\[ T_i^* = [C_{i1} + C_{i3} + C_{i4} + C_{i5} + C_{i6} + C_{i7} + C_{i9} + C_{i10} + C_{i12} + C_{i13} + C_{i14} + C_{i15} + C_{i16} + C_{i17} + C_{i18} + C_{i20} + C_{i21} + C_{i22} + C_{i23} + C_{i25} - [C_{i27}] ] \]

(M37)

\[ T_i^* = [C_{i1} + C_{i2} + C_{i3} + C_{i4} + C_{i5} + C_{i6} + C_{i7} + C_{i8} + C_{i9} + C_{i10} + C_{i12} + C_{i13} + C_{i14} + C_{i15} + C_{i16} + C_{i17} + C_{i18} + C_{i20} + C_{i21} + C_{i22} + C_{i23} + C_{i24} + C_{i25} ] \]

(M38)

\[ T_i^* = [C_{i1} + C_{i2} + C_{i3} + C_{i4} + C_{i5} + C_{i6} + C_{i7} + C_{i8} + C_{i9} + C_{i10} + C_{i12} + C_{i13} + C_{i14} + C_{i15} + C_{i16} + C_{i17} + C_{i18} + C_{i20} + C_{i21} + C_{i22} + C_{i23} + C_{i24} + C_{i25} ] \]

(M39)

\[ T_i^* = [C_{i1} + C_{i2} + C_{i3} + C_{i4} + C_{i5} + C_{i6} + C_{i7} + C_{i8} + C_{i9} + C_{i10} + C_{i12} + C_{i13} + C_{i14} + C_{i15} + C_{i16} + C_{i17} + C_{i18} + C_{i20} + C_{i21} + C_{i22} + C_{i23} + C_{i24} + C_{i25} ] \]

(M40)

\[ T_i^* = [C_{i1} + C_{i2} + C_{i3} + C_{i4} + C_{i5} + C_{i6} + C_{i7} + C_{i8} + C_{i9} + C_{i10} + C_{i12} + C_{i13} + C_{i14} + C_{i15} + C_{i16} + C_{i17} + C_{i18} + C_{i20} + C_{i21} + C_{i22} + C_{i23} + C_{i24} + C_{i25} ] \]

(M41)

Note: 1) A species inside a parenthesis in Table 6.15 denotes the activity of the species.
2) A species inside a bracket in Table 6.15 denotes the concentration of the species.
3) Activity = activity coefficient \times concentration.

The simulation involves a total 41 species, excluding the species C_{28} (H_2O) whose activity is set as 1.0, 25 equilibrium reactions (N_e = 25) and 8 kinetic reactions (N_k = 8). The BIOGEOCHEM processor will perform the formal matrix decomposition of the reaction network (Fang et al., 2003), as which is shown in Table 6.15.

Finally, the concentrations of 8 biogeochemical components and 8 kinetic variables are introduced into the transport module to form the governing equation of reactive transport.
There are three schemes to couple hydrologic transport and biogeochemical reactions in HYDROGEOCHEM 5.0: (1) fully implicit iteration approach, (2) operator splitting approach, and (3) Predictor-Corrector approach. Since this example is a three-dimensional problem containing 4000 elements, 4851 nodes and 41 species, it is presumed more efficient to solve this “big” system by using operator splitting approach or Predictor-Corrector approach. In this example, the operator splitting approach has been selected to perform the coupling between transport and biogeochemical reactions.

Initially, all species were set at low concentrations in the low concentration region; in the high concentration region, mineral M, aqueous species C₃, C₃₀, and three ion-exchange sites were set at relatively higher concentrations to highlight the changes of concentrations due to transport and biogeochemical reactions. For example, mineral M was given an initial concentration of 2.0e-5 mol/dm³ and 1.0e-7 mol/dm³ in high concentration region and low concentration region, respectively. Mineral M would gradually dissolve into aqueous species C₁ and C₂ in an irreversible reaction (R₁ in Table 6.14), the concentration curves of mineral M in the low and high concentration regions are plotted in Fig. 6.37. Along with mineral M desolating, the surface area of mineral M reduces. Therefore, the concentrations of sorbed species S₁ through S₈ decrease since they are adsorbed on the surface of mineral M. To show the tendency of decreasing for all sorbed species, the concentration contours of the logarithm value for sorbed species S₁ are plotted along one cross-section z = 90 dm of domain (Fig. 6.40). The details of the cross-section z = 90 dm, which goes through the high and low concentration region, is plotted in 3-D view and 2-D view individually (Fig. 6.39).

The Dirichlet boundary condition on the top surface (the top of Fig. 6.39) provides a source of high concentration for aqueous species C₆, C₅, C₃ and C₃₀, which are transported into the subsurface along with fluid flow; while the concentrations of C₆, C₅, C₃ and C₃₀ are lower on the variable boundary on the top surface. After several time steps, these species enter the high concentration region in subsurface, driving the reaction system to produce/consume species. The red rectangle, which is drawn on the cross-section z = 90 dm (the bottom of Fig. 6.39), indicates the region affected by the Dirichlet boundary; while regions elsewhere are affected by the variable boundary. The simulation shows contrast results within the high concentration region due to different affects of the Dirichlet and the variable boundary conditions on the top surface. For instance, the concentration contours of the logarithm value for S₁ along the cross-section z = 90 dm are plotted in Fig. 6.40. It’s observed that the concentration of S₁ within the high concentration region affected by the Dirichlet boundary is much lower than that within the region affected by the variable boundary, and this is caused by the aqueous species C₆, C₅, C₃ and C₃₀, which drive the reaction system to consume more sorbed species S₁.

R₁ in Table 6.14, the product of mineral M dissolution, becomes slower than that at the beginning. The thin solid line in Fig. 6.38 shows the decreasing tendency of species C₁. As a result, the production of C₁₀ will decrease as well as the production of C₁ by R₁. The concentration of C₁₀ increases up to 1.47e-5 mol/dm³ at the end of the first 10 days and converges downwards to the value of 1.0e-5 mol/dm³.

The mass of C₁₀ generated in the high concentration region is transported into subsurface along with fluid flux. The concentration contours for species C₁₀ are plotted in Fig. 6.41 at various time steps.
Fig. 6.37. Concentration Curves of Mineral M within High and Low Concentration Regions of Example 15

Fig. 6.38. Concentration Curves for Species Involved in Reaction 7 of Example 15
Fig. 6.39. Cross-section at $z = 90$ dm of the Domain (top: 3D view, bottom: 2D view)
Fig. 6.40. Concentration Contours of Logarithm Values for Adsorbed Species S₁ along Cross-section z = 90 dm of the Domain at Various Time Steps
Fig. 6.41. 3-D Concentration Contours for Species C_{10} at Various Time Steps
Example 16: 3-D Flow and Biogeochemical Transport with Complexation, Adsorption, Ion-exchange and Mineral Dissolution Reactions with Heat Transfer Consideration

Example 16 is designed to study the significance of heat transfer on reactive biogeochemical transport simulated in Example 15. The initial temperature of the entire region is set 298 K. The regions of boundary conditions for heat transfer are identical with those of reactive biogeochemical transport in Example 15 (Fig. 6.42). A fixed high temperature 308 K is set on Dirichlet boundary while the temperature is forced 298 K on variable boundaries.

The specific heats of the water and dry medium are assumed 1.0E20 dm²/day²/K and 1.0E19 dm²/day²/K, respectively. The apparent thermal conductivity is set 2.0E19 dm²/day²/K. A time step size of 2 days is used and a total 100 time-step transient simulation is performed in this example. The heat transports from the high temperature region on Dirichlet boundary into the low temperature region through conduction and convection along with fluid flux, and the temperature distributions at various time steps are presented in Fig. 6.43.

In the subsurface system, heat transfer is coupled with flow and biogeochemical transport during the simulation. It can be either considered or neglected. When heat transfer is not to be simulated, or the reaction enthalpy for each reaction is set zero, the distributions of temperature will not affect the reactive biogeochemical transport. For that case, concentration contours in logarithm for sorbed species $S_1$, which is the reactant in the desorption reactions (R27 of Table 6.14), along cross-section $z = 90$ dm were shown in Fig. 6.39. For the present case the high temperature condition is applied, the concentration contours in logarithm for reactant $S_1$ with positive reaction enthalpy are shown in Fig. 6.44 (EX-fth-2.INP as the input file). Figure 6.45 shows the concentration contours in logarithm for reactant $S_1$ with negative reaction enthalpy at various time steps (EX-fth-3.INP as the input file). By comparing these two figures, the concentration of reactant $S_1$ with positive reaction enthalpy is observed to decrease faster than that with negative reaction enthalpy. Therefore, a high temperature increases the extent of a reaction with a positive reaction enthalpy and diminishes the extent of a reaction with a negative reaction enthalpy, and vice versa.

In practical cases, such as contaminant clean-up process, high temperature will introduce less retardation to the pollutant dissolution or desorption reactions under positive reaction enthalpy in reactive biogeochemical transport. Practically, less retardation implies that more of the pollutant will transform to the aqueous phase and therefore less effort is needed in cleaning up. Thus, temperature is an important factor in the subsurface simulation so that proper means can be dealt with to treat the removal of pollutants.
Fig. 6.42. The Problem Definition for Example 16: Heat Transfer Boundary Conditions
Fig. 6.43. Distributions of Temperature at Various Time Steps for Example 16
Fig. 6.44. Concentration Contours of Logarithm Value for Adsorbed Species $S_1$ along Cross-section $z = 90$ dm with Positive Reaction Enthalpy at Various Time Steps.
Fig. 6.45. Concentration Contours of Logarithm Values for Adsorbed Species $S_1$ along Cross-section $z = 90 \text{ dm}$ with Negative Reaction Enthalpy at Various Time Steps.
7. REFERENCES


Yeh, G.T., 1987a. FEMWATER: A Finite Element Model of WATER Flow through Saturated-Unsaturated Porous Media -- First Revision. ORNL-5567/R1, Oak Ridge National Laboratory, Oak Ridge, TN.

Yeh, G.T., 1987b. 3DFEMWATER: A Three-dimensional Finite Element Model of WATER Flow through Saturated-Unsaturated Porous Media. ORNL-6386, Oak Ridge National Laboratory, Oak Ridge, TN.


on simulation of Subsurface Flow and Contaminant Transport by Finite Element and Analytic Methods.


Yeh, G.T., Y.L. Fang and W.D. Burgos. 2003b. BIOGEOCHEM 2.0: A Numerical Model to Simulate


APPENDIX A: DATA INPUT GUIDE FOR HYDROGEOCHEM 5.0

In this version, the unit of length should be the decimeter (dm) and the unit of mass for any chemical species should be the mole. The density of water and solid should be expressed in kg/dm$^3$ (liter). The ion-exchange capacity is in equivalents/M of solid. The corresponding concentration unit of all species (aqueous, sorbed, and precipitated species) is mole/liter of fluid (Molar); the corresponding unit for the sorption distribution coefficient is dm$^3$/kg (= ml/g). The unit of temperature should be expressed in the Kelvin scale (K). Any units of time may be used as long as the same unit is used throughout the input file.

All data sets except for DATA SET 1 must be preceded by a data set name.

DATA SET 1: TITLE

Three records are used per problem

Record 1: FORMAT(I5,7A10)

1.1 NPROB = Problem number.
1.2 TITLE = Array for the title of the problem. It may contain up to 70 characters from column 6 to column 75.

Record 2: Free Format

2.1 IMODF = Is flow to be modeled? 0 = No, 1 = Yes.
2.2 IMOD = Is transport to be modeled? 0 = No, 1 = Yes.
2.3 IITR = Integer indicating if iteration table of convergence information to be printed:
0 = no.
1 = transport iteration table be printed.
2 = both transport and chemical iteration tables printed.

2.4 INTER = Integer indicating if concentration field is to be printed for each hydrological-chemical interaction:
1 = yes.
0 = no.

2.5 ICOND = Integer indicating if the condition number of the Jacobian matrix in chemical equilibrium computation to be printed:
1 = yes.
0 = no.

2.6 NHGCI = Integer indicating if chemical information is to be printed:
0 = no.
> 0 = print every NHGC inter-hydro-geochem iterations.

2.7 IGEOM = Integer signifying if the geometric data are to be printed:
0 = no.
1 = yes.

2.8 IBUG = Integer indicating if the diagnostic output for coupling loop and flow simulation is desired? 0 = No, 1 = Yes.

2.9 ICHNG = Integer control number indicating if the cyclic change of rainfall-seepage nodes is to be printed, =0, no = 1, yes.

2.10 ITEMP = Integer indicating if heat transfer need to be modeled:
0 = no.
1 = yes

If ITEMP = 1, then the following record is needed for heat transfer simulations:

Record 3:

3.1 SPHTW = Specific heat of groundwater, \( [(L^2/T^3)/K] \).

3.2 GASCNT = Ideal gas constant \( [(M/L)/(T^3K)] \).

DATA SET 2: COUPLING ITERATION PARAMETERS

One Record is needed for this data set.

1.1 NITFTS = Iteration numbers allowed for solving the coupled nonlinear equations for the steady-state solutions.

1.2 NITFTT = Iteration numbers allowed for solving the coupled nonlinear equations for the transient or transient-state solutions.

1.3 OMEFTS = Iteration parameter for solving the coupled nonlinear equations for the steady-state solutions.

1.4 OMEFTT = Iteration parameters for solving the coupled nonlinear equations for the transient or transient-state solutions.

DATA SET 3: ITERATION AND OPTIONAL PARAMETERS

At most eight records are needed for this data set.

If IMODF = 1, then the following three records are needed for flow simulations:

Flow Record 1 -
### Flow Record 1

1.1 **NCYLF** = No. of cycles permitted for iterating rainfall-seepage boundary conditions per time step.

1.2 **NITERF** = Number of iterations allowed for solving the non-linear flow equation.

1.3 **NPITERF** = No. of iterations permitted for solving the linearized flow equation using iterative matrix solver.

1.4 **KGRAV** = Gravity term control; 0 = no gravity term, 1 = with gravity term.

1.5 **IPNTSF** = Matrix solution indicator for flow simulations;
   - 0 = block iteration methods,
   - 1 = successive point iteration methods,
   - 2 = polynomial preconditioned conjugate gradient methods,
   - 3 = incomplete Cholesky preconditioned conjugate gradient methods.

### Flow Record 2

2.1 **ILUMPF** = Mass matrix lumping control for solving flow equation:
   - 0 = No lump.
   - 1 = Lump.

2.2 **IMIDF** = Mid-difference control for solving flow equation:
   - 0 = No mid-difference.
   - 1 = Mid-difference, and WF in the Flow Record 3 should be 1.

2.3 **KSSF** = Flow steady state control;
   - 0 = steady state solution desired,
   - 1 = transient state or transient solutions.

2.4 **KSTRF** = Auxiliary storage output control for flow variables;
   - 0 = no storage,
   - 1 = flow output stored on Logical Unit 11.

2.5 **NSTRF** = Number of logical records to be read for flow variables via auxiliary storage device Logical Unit 14 for restarting flow calculation, 0 = no restart.

2.6 **IQUARF** = Indicator for quadrature integration in flow simulations:
   - 11 = nodal quadrature for both surface integration and volume integration.
   - 12 = nodal and Gaussian quadrature for surface and volume integration, respectively.
   - 21 = Gaussian and nodal quadrature for surface and volume integration, respectively.
   - 22 = Gaussian quadrature for both surface and volume integration.

### Flow Record 3

3.1 **TOLAF** = Steady-state convergence criteria for flow simulations, (L).
3.2 TOLBF = Transient-state convergence criteria for flow simulations, (L).

3.3 WF = Time derivative weighting factor for flow simulations;
   0.5 = Crank-Nicolson central,
   1.0 = backward difference and/or mid-difference.

3.4 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.

3.5 OMIF = Relaxation parameter for iterative solution of linearized matrix equation;
   0.0 - 1.0 = under relaxation,
   1.0 - 1.0 = exact relaxation,
   1.0 - 2.0 = over relaxation.

3.6 CNSTKR = Constraint on relative hydraulic conductivity,
   0 = no constraint,
   0.0001, 0.001, or 0.01 should be tried if non-convergence occurs.

If IMOD = 1, then the following three records are needed for transport and heat transfer simulations:

Transport Record 1 -

1.1 NSTR = Number of logical records to be read for transport variables via auxiliary storage device Logical Unit 13 for restarting transport calculation, and heat transfer variables via auxiliary storage device Logical Unit 23 for restarting heat transfer calculation.
   0 = no restart.

1.2 KSTR = Auxiliary storage output control for transport and/or heat transfer variables;
   0 = no storage,
   1 = transport output stored on Logical Unit 12, heat transfer output stored on Logical Unit 22.

1.3 KSS = Transport and heat transfer steady state control;
   0 = steady state solution desired,
   1 = transient state or transient solutions.

1.4 NITER = Number of iterations allowed for solving the non-linear transport and heat transfer equation.

1.5 NPITER = No. of iterations permitted for solving the linearized transport and heat transfer equation using iterative matrix solver.

1.6 KVI = Velocity input control when transport simulations only:
   - 1 = Velocity and moisture content defined by Data Set 27.
   - 1 = Steady-state velocity and moisture content read from Logical Unit 17.
   - 2 = Transient velocity and moisture content read from Logical Unit 17.
Transport Record 2 -

2.1 IMID = Mid-difference control for solving transport and heat transfer equation:
0 = No mid-difference.
1 = Mid-difference, and W in the Transport Record 3 should be 1.

2.2 IWET = Weighting function control for advection term of the transport and heat transfer equation:
0 = Galerkin weighting.
1 = Full upstream weighting.

2.3 ILUMP = Mass matrix lumping control for solving transport and heat transfer equation:
0 = No lump.
1 = Lump.

2.4 IOPTIM = Optimization factor computing indicator:
1 = Optimization factor is to be computed.
0 = Optimization factor is given by APHAG in line 3.

2.5 IPNTS = Matrix solution indicator for transport and heat transfer simulations;
0 = block iteration methods,
1 = successive point iteration methods,
2 = polynomial preconditioned conjugate gradient methods,
3 = incomplete Cholesky preconditioned conjugate gradient methods.
4 = modified incomplete Cholesky PCG methods.
5 = symmetric successive over-relaxation PCG methods.

2.6 LGRN = Is the Lagrangian-Eulerian approach to be used to solve transport and heat transfer equation:
1 = Yes.
0 = No.

2.7 IQUAD = Indicator for quadrature integration in transport and heat transfer simulations:
11 = nodal quadrature for both surface integration and volume integration.
12 = nodal and Gaussian quadrature for surface and volume integration, respectively.
21 = Gaussian and nodal quadrature for surface and volume integration, respectively.
22 = Gaussian quadrature for both surface and volume integration.

2.8 IDTI = Time step option for the Eulerian step when the LE approach is applied;
0 = 1.0/DELT,
1 = Computed from tracking.

2.9 IALT = Indicator of solution technique of kinetically controlled complexed species:
0 = use direct iteration method. This option can be selected only when all kinetic reactions are "basic" kinetic reactions.
1. Use either the operator splitting or the predictor-corrector method, as specified by APC.

2.10 **ILEBF** = Indicator of FEM scheme applied to boundary elements when LGRN = 1
   - 0 = hybrid Lagrangian-Eulerian for whole domain.
   - 1 = hybrid Lagrangian-Eulerian for interior & advective FEM for boundary.

### Transport Record 3 -

3.1 **W** = Time derivative weighting factor for transport and heat transfer simulations;
   - 0.5 = Crank-Nicolson central,
   - 1.0 = backward difference and/or mid-difference.

3.2 **WV** = Time derivative weighting factor for velocity term in transport and heat transfer simulations;
   - 0.5 = Crank-Nicolson central,
   - 1.0 = backward difference and/or mid-difference.

3.3 **OME** = Iteration parameter for solving the nonlinear transport and heat transfer equation;
   - 0.0 - 1.0 = under-relaxation,
   - 1.0 - 1.0 = exact relaxation,
   - 1.0 - 2.0 = over-relaxation.

3.4 **OMI** = Relaxation parameter for iterative solution of linearized matrix equation;
   - 0.0 - 1.0 = under relaxation,
   - 1.0 - 1.0 = exact relaxation,
   - 1.0 - 2.0 = over relaxation.

3.5 **TOLA** = Steady state convergence criteria for nonlinear transport equations. Half of its value is used as the tolerance for solving the linearized matrix equation with successive point iteration.

3.6 **TOLB** = Transient convergence criteria for nonlinear transport equations. Half of its value is used as the tolerance for solving the linearized matrix equation with successive point iteration.

3.7 **APHAG** = Upstream weighting factor if IOPTIM = 0:
   - Values are between 0.0 and 1.50 when the advection form of the equation is used.
   - If APHAG > 1.34D0, the program will choose appropriate values of weighting factor. When the conservative form of the equations is used or when IOPTIM = 1, this value is not used by the program.

3.8 **APC** = Indicator for solution technique for kinetically controlled complexed species when IALT = 1 is chosen.
   - 0.0 = use the operator splitting method,
   - 1.0 = use the predictor-corrector method.

3.9 **TOLAT** = Steady state convergence criteria for heat transfer equations. Half of its
value is used as the tolerance for solving the linearized matrix equation with successive point iteration.

3.10 TOLBT = Transient convergence criteria for heat transfer equations. Half of its value is used as the tolerance for solving the linearized matrix equation with successive point iteration.

Transport Record 4 -

This record is needed in this data set if and only if LGRN = 1

4.1 NXW = number of subelements in x-direction for tracking
4.2 NYW = number of subelements in y-direction for tracking
4.3 NZW = number of subelements in z-direction for tracking
4.4 IDETQ = tracking velocity option:
            1 = average velocity is used.
            2 = single velocity is used.

Transport Record 5 -

This record is needed in this data set if and only if IPNTSF or IPNTS > 1.

5.1 IEIGEN = signal of parameter estimation for GG in the polynomial preconditioned conjugate gradient method; 0 = not requested, 1 = requested.
5.2 GG = the upper bound on the maximum eigenvalue of the coefficient matrix used in the polynomial preconditioned conjugate gradient method.
5.3 ALPHA = weighting factor for computing the diagonal element of the diagonal matrix used in the modified incomplete Cholesky preconditioned conjugate gradient method.
5.4 OMEGA = relaxation parameter used in the SSORCG method.

DATA SET 4: TIME STEP AND PRINTOUT AS WELL AS STORAGE CONTROL

The following six records are needed for this data set.

Record 1-

1.1 NTIF = Number of time steps or time increments for flow simulations.
1.2 NTI = Number of time steps or time increments for transport and heat transfer simulations.

191
1.3 \textbf{DELT} = Initial time-step size (T).
1.4 \textbf{CHNG} = Percent change in the time-step size in each of the subsequent time increments, (dimensionless in decimal fraction).
1.5 \textbf{DELMAX}= Maximum allowable time-step size (T).
1.6 \textbf{TMAX} = Maximum simulation time (T).

Record 2-

2.1 \textbf{NTIPRT} = No. of time steps to line-printout flow, transport and heat transfer simulations

Record 3 -

3.1 \textbf{ITMPRT} = Time step number when the first time the line-printout is needed.
3.2 \textbf{ITYPPRT} = Type of information needed at \textbf{ITMPRT}:
0= Print nothing.
1= Print FLOW, FRATE, and TFLOW for flow and transport, and temperature for heat transfer
2= Print above (1) plus pressure head for and concentration for transport.
3= Print above (2) plus total head for flow and material flux for transport.
4= Print above (3) plus Darcy velocity for flow.
5= Print above (4) plus moisture content at quadrature points.

Repeat 3.1 and 3.2 for \textbf{NTIPRT} times.

Record 4-

4.1 \textbf{NTISTO} = No. of time-steps to store flow, transport and heat transfer simulations in auxiliary storage device.

Record 5 -

5.1 \textbf{ITMPRT} = Time step number when the first time the storage is needed.

Repeat 5.1 for \textbf{NTISTO} times.

Record 6 -

6.1 \textbf{NDTCH} = Number of times to reset the time-step size to the initial value.
6.2 \textbf{TDTCH}(1) = The first time at which the time-step size is reset to its initial value.
6.3 \textbf{TDTCH}(2) = The second time at which the time-step size is reset to its initial value.
6.N+1 TDTCH(N) = The N-th (N=NDTCH) time at which the time-step size is reset to its initial values.

**DATA SET 5: CHEMICAL PRINTOUT INDICATOR**

This data set is needed if **IMOD = 1**. Two records are needed.

Record 1: Free-field format input containing one integer.

1.1 NCPRT = Number of nodes where detailed chemical information will be printed.

Record 2: Free-field format input contains 2*NCPRT integers.

2.1 NODE = Global node number of the first node where detailed chemical information will be printed.

2.2 ITYP = Indicator to show the type of geochemical information to be printed:
- 0 = Print nothing.
- 1 = Chemical component concentrations
- 2 = Print above (1) plus component species concentrations
- 3 = Print above (2) plus product species concentrations.

Repeat 2.1 and 2.2 for NCPRT times.

**DATA SET 6: MATERIAL PROPERTIES**

Record 1 -

1.1 NMAT = No. of material types.

The following records reading saturated material properties and unsaturated material characteristics (soil properties) are needed if **IMODF = 1**

Record 2 - Integers and real numbers required to specify the materials

2.1 NMPPMF = No. of material properties per material for flow, = 14 in the current version.

2.2 KCP = Permeability input control;
- 0 = input saturated hydraulic conductivity,
- 1 = input saturated permeability.

2.3 NSPPM = Number of points in tabular soil property functions or number of parameters to specify analytical soil functions per material.
2.4 KSP = Soil property input control; 0 = analytical input; 1 = Tabular data input.

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

Records 3 to Records (NMAT+2) - A total of NMAT records reading saturated material properties are needed, one each for one material.

<table>
<thead>
<tr>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>PROPF(1,I)  = Modified coefficient of compressibility of the medium I, (1/L); 0.0 for no compressibility.</td>
</tr>
<tr>
<td>3.2</td>
<td>PROPF(2,I)  = Modified coefficient of compressibility of water in medium I, (1/L); 0.0 for no compressibility.</td>
</tr>
<tr>
<td>3.3</td>
<td>PROPF(3,I)  = Saturated moisture content of medium I without mineral precipitations within, (dimensionless in decimal point).</td>
</tr>
<tr>
<td>3.4</td>
<td>PROPF(4,I)  = xx-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).</td>
</tr>
<tr>
<td>3.5</td>
<td>PROPF(5,I)  = yy-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).</td>
</tr>
<tr>
<td>3.6</td>
<td>PROPF(6,I)  = zz-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).</td>
</tr>
<tr>
<td>3.7</td>
<td>PROPF(7,I)  = xy-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).</td>
</tr>
<tr>
<td>3.8</td>
<td>PROPF(8,I)  = xz-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).</td>
</tr>
<tr>
<td>3.9</td>
<td>PROPF(9,I)  = yz-component of the saturated hydraulic conductivity tensor, (L/T) or saturated permeability, (L**2).</td>
</tr>
<tr>
<td>3.10</td>
<td>PROPF(10,I) = Referenced density of fluid in material I, (M/L**3).</td>
</tr>
<tr>
<td>3.11</td>
<td>PROPF(11,I) = Referenced dynamic viscosity of fluid in material I, (M/LT).</td>
</tr>
<tr>
<td>3.12</td>
<td>PROPF(12,I) = Bulk density of the media in material I, (M/L**3).</td>
</tr>
<tr>
<td>3.13</td>
<td>PROPF(13,I) = Residual moisture content of medium I (dimensionless in decimal point).</td>
</tr>
<tr>
<td>3.14</td>
<td>PROPF(14,I) = The fractal exponent to compute the effect of mineral precipitations on permeability; documentary rage is 1.0 ~ 5.5; set this parameter as 0.0 and the mineral mole volume of each precipitated species (in Data Set 18) as 0.0 for the case of no precipitation effect on flow simulations.</td>
</tr>
</tbody>
</table>
***** NOTE: 3.1 through 3.14 must be repeated for NMAT times.

Records (NMAT +3) through (2*NMAT + 2) - Analytical soil parameter for computing moisture content and relative conductivity (relative permeability) as functions of pressure head. These records are needed if and only if $KSP = 0$ and modifications of the SUBROUTINE SPROP are needed to represent the desired analytic soil property function as described in chapter 4.

4.1  SPP(1,I,1) = Analytical moisture-content parameter 1 of material I.
4.2  SPP(1,I,2) = Analytical relative conductivity parameter 1 of material I.

\begin{align*}
4.2*J-1 & \quad SPP(J,I,1) = \text{Analytical moisture-content parameter } J \text{ of material I.} \\
4.2*J & \quad SPP(J,I,2) = \text{Analytical relative conductivity parameter } J \text{ of material I.}
\end{align*}

4.2*N-1  SPP(N,I,1) = Analytical moisture-content parameter J of material I.
4.2*N  SPP(N,I,2) = Analytical relative conductivity parameter J of material I.

*****NOTE: 4.1 through 4.2*N (where N = NSPPM) must be repeated for NMAT times.

Records (NMAT +3) through (2*NMAT + 2) - Tabular soil parameter for interpreting degree of saturation ($S$), relative conductivity (relative permeability; $k_r$), and water capacity ($d\theta/dh$) based on pressure head ($h$). These records are needed if and only if $KSP = 1$

4.1  SPP(1,I,4) = Tabular value of $h$ of the first point for material I
4.2  SPP(1,I,1) = Tabular value of $S$ of the first point for material I
4.3  SPP(1,I,2) = Tabular value of $k_r$ of the first point for material I
4.4  SPP(1,I,3) = Tabular value of $d\theta/dh$ of the first point for material I

\begin{align*}
4.4*J-3 & \quad SPP(J,I,4) = \text{Tabular value of } h \text{ of the J-th point for material I} \\
4.4*J-2 & \quad SPP(J,I,1) = \text{Tabular value of } S \text{ of the J-th point for material I} \\
4.4*J-1 & \quad SPP(J,I,2) = \text{Tabular value of } k_r \text{ of the J-th point for material I} \\
4.4*J & \quad SPP(J,I,3) = \text{Tabular value of } d\theta/dh \text{ of the J-th point for material I}
\end{align*}

4.4*N-3  SPP(N,I,4) = Tabular value of $h$ of the N-th point for material I
4.4*N-2  SPP(N,I,1) = Tabular value of \( S \) of the \( N \)-th point for material I
4.4*N-1  SPP(N,I,2) = Tabular value of \( k_r \) of the \( N \)-th point for material I
4.4*N  SPP(N,I,3) = Tabular value of \( \theta/dh \) of the \( N \)-th point for material I

*****NOTE: 4.1 through 4.4*N (where \( N = NSPPM \)) must be repeated for \( NMAT \) times.

The following records reading material properties for transport simulations are needed if \textbf{IMOD} = 1

Records (\( 2*NMAT + 3 \)) - Integer required to specify the material properties for transport simulations

5.1  NMPPM = No. of material properties per material for transport, = 9

Records (\( 2*NMAT+4 \)) to Records (\( 3*NMAT+3 \)) - A total of \( NMAT \) records reading material properties for transport are needed, one each for one material.

6.1  PROP(1,I) = Longitudinal dispersivity (L).
6.2  PROP(2,I) = Lateral dispersivity (L).
6.3  PROP(3,I) = Molecular diffusion coefficient of pure liquid in the I-th medium \( (L^{**2}/T) \).
6.4  PROP(4,I) = Bulk density of medium I \( (M/L^{**3}) \).
6.5  PROP(5,I) = Modified compressibility of medium I \( (1/L) \); 0.0 for no compressibility.
6.6  PROP(6,I) = Modified compressibility of water in medium I \( (1/L) \); 0.0 for no compressibility.
6.7  PROP(7,I) = Saturated moisture content of medium I without mineral precipitations within (dimensionless).
6.8  PROP(8,I) = Tortuosity of medium I (dimensionless), which is used to adjust diffusion coefficient due to the tortuosity effect; value should be given as \( \leq 1.0 \); 1.0 for no adjustment. Tortuosity is computed as \( (dl/dx)^2 \) where \( dl \) is the actual sinuous path in a length interval \( dx \).
6.9  PROP(9,I) = Power in the Archie's law to compute the effect of mineral precipitations on dispersion coefficients (dimensionless); documentary range is 1.3 to 2.5, set this parameter as 1.0 and the mineral mole volume of each precipitated species (in Data Set 18) as 0.0 for the case of no precipitation effect on transport simulations.

The following records reading material properties for heat transfer simulations are needed if \textbf{ITEMP} = 1

Records (\( 3*NMAT + 4 \)) - Integer required to specify the material properties for heat transfer simulations
7.1 NMPPMT = No. of material properties per material for heat transfer, = 7

Records (3*NMAT+5) to Records (4*NMAT+4) - A total of NMAT records reading material properties for heat transfer are needed, one each for one material.

8.1 PROPT(1,I) = Longitudinal thermal dispersivity [L].
8.2 PROPT(2,I) = Lateral thermal dispersivity [L].
8.3 PROPT(3,I) = Molecular thermal diffusion coefficient of pure liquid in the I-th medium [(ML/T^3)/K].
8.4 PROPT(4,I) = Specific heat of the dry medium I [(L^2/T^2)/K].
8.5 PROPT(5,I) = xx-component of the apparent thermal conductivity [(ML/T^3)/K].
8.6 PROPT(6,I) = yy-component of the apparent thermal conductivity [(ML/T^3)/K].
8.7 PROPT(7,I) = zz-component of the apparent thermal conductivity [(ML/T^3)/K].

*****NOTE: Data Sets 7, 7B and 8 described below are not needed if Logical Unit 17 is used as input (i.e., IMOD = 1 and IMODF = 0 and KVI > 0).

DATA SET 7: NODAL POINT COORDINATES

Typically, a total of (1+NNP) records are required: One record for the controlling numbers, NNP records for nodal coordinate. However, if a group of nodes appear in regular pattern, automatic generation can be used.

Record 1 - Controlling numbers

1.1 NNP = Total number of global nodes in the region of interest.

Records 2 to (1+NNP): Each record is free-field format input containing:

2.1 NI = Node number of the first node in the sequence.
2.2 NSEQ = NSEQ subsequent nodes to 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 line with 9 0's must be used to signal the end of this data set.

DATA SET 7B: SUBREGION DATA

This data set is required if both or either IPNTSF and IPNTS is 0.

Record 1 - Controlling number

1.1 NREGN = No. of subregions.

Record 2 to (1+NREGN) - No. of Nodes for each Subregion - Normally, NREGN records are required. However, if 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 NODE = Number of nodes for the subregion NK.
2.5 NODEAD= Increment of NODE in each of the NSEQ subsequent subregions.

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

Record (2+NREGN) to (2+NREGN*(1+LNNP)) - Mapping between Global nodes and Subregion Nodes - This subdata set 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 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.
DATA SET 8: ELEMENT INCIDENCES

Typically, a total of (1+NEL) records are needed: the first record is to read controlling integers and the next NEL records are to read element incidences, one for each element. However, automatic generation is made if a group of elements appears in a regular pattern.

Record 1 - Controlling numbers

1.1 NEL = Total number of elements in the region of interest.

Records 2 to (1+ NEL): Each record is free-field format input containing:

2.1 MI = Global element number.
2.2 NESQ = NESQ subsequent element will be automatically generated.
2.3 MIAD = Increment of MI for each of the NSEQ subsequent element.
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: A record with 12 0's must be used to signal the end of this subdata set.

IE(MI,1) to IE(MI,8) are numbered according the convention shown in following diagram. 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.
NOTE: IE(MI,5) through IE(MI,8) are set to zero for a tetrahedral element. Likewise, IE(MI,7) and IE(MI,8) are set to zero for a triangular prism element.

**DATA SET 9: MATERIAL TYPE CORRECTION**

Normally 1+NCM records are required. However, if the elements to be corrected with different material properties appear in regular patterns, automatic generation can be used.

Record 1 - Controlling numbers

1.1 NCM = Total number of elements in the region of interest.

Records 2 to (1+NCM): This subdata is required only if NCM > 0. Each record is free-field format input containing:

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 MIAD = Increment of element number for each of the NSEQ subsequent elements.
2.4 MITYP = Material type to be applied to element MI.
2.5 MTYPAD = Increment of the type of material 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.
Fig. A.1 Element Indices in (A) Hexahedral Element; (B) Triangular Prism; (C) Tetrahedral Element
DATA SET 10: NUMBER OF SPECIES AND REACTIONS

Record 1:

1.1 NOM = Number of species
1.2 NER = Number of equilibrium reactions
1.3 NKR = Number of kinetic reactions

DATA SET 11: SPECIES INFORMATION

Record 1:

[SPECN(i), ISCN(i,1), ISCN(i,2), ISCN(i,3), VJ(i), VM(i)] where:

1.1 SPECN(i) = species name
1.2 ISCN(i,1) = Index of phase for the i-th species

1 = Aqueous phase species
2 = NAPL phase species
3 = Air phase species
4 = Solid phase species

5xx = Adsorbed species. A 3 digit number is required with the last two digits representing the site where the species is sorbed. For example, 501 means this species is sorbed to sorption site 1. This index designation can be used for surface complexed species (more information required below), and sorbed species whose liquid-solid distribution will be described using a conventional isotherm equation (e.g., Freundlich, Langmuir).

6xx = Ion-exchange species. A 3 digit number is required with the last two digits representing the site where the species is ion-exchanged. For example, 601 means this species is ion-exchanged to ion-exchange site 1.

7xx = Fictitious species representing \(\exp(-e\Psi_o/kT)\) (where \(e\) is the electronic charge = 1.6(10 \(^{-19}\) C/electron, \(k\) is the Boltzman constant = 1.38(10 \(^{-23}\) J K\(^{-1}\)), and \(\Psi_o\) is the electrostatic potential at the 'o' plane of the solid). This species is required if the constant capacitance model or the triple-layer model is used to model adsorption. A 3 digit number is required with the last two digits representing the site where the species is sorbed. For example, 701 means this species is sorbed to the 'o' plane of surface complex site 1.

8xx = Fictitious species representing \(\exp(-e\Psi_\beta/kT)\) (where \(e\) is the electronic charge, \(k\) is the Boltzman constant, \(T\) is the absolute temperature, and \(\Psi_\beta\) is the electrostatic potential at the '\(\beta\)' plane in the triple-layer model). This species is required if the triple-layer model is used to model adsorption. A 3 digit number is required with the last two digits representing the site where the species is sorbed. For example, 801 means this species is sorbed to the '\(\beta\)'
plane of surface complex site 1.

1.3 ISCN(i,2) = Index of mobility for the i-th species
   0 = immobile species (default value)
   1 = mobile species

1.4 ISCN(i,3) = Index of species concentration for the i-th species
   0 = species concentration is simulated
   3 = species concentration is fixed

1.5 VJ(i) = Charge of the i-th species

1.6 VM(i) = Molar volume of the i-th species

******NOTE: Repeat record 1.1 through 1.6 for NOM times.

DATA SET 11A ADDITIONAL INFORMATION FOR ADSORBED SPECIES

This data set is needed if there are adsorbed species (i.e., if ISCN(i,1) = 5xx)

Record 1:

1.1 IADS = Index of surface complexation model
   0 = diffuse layer (i.e., double-layer) model
   1 = constant capacitance model
   2 = triple-layer model

1.2 NSORB = Number of adsorption sites

Record 2: (this record is needed when IADS > 0)

For I = 1, NSORB

2.1 CAP1A(I,J) = Capacitance between the 'o' plane and 'β' plane, (F/m²) for the I-th adsorbing site

2.2 CAP2A(I,J) = Capacitance between the 'β' plane and 'd' plane, (F/m²) for the I-th adsorbing site

2.3 SREAA(I,J) = Surface area of adsorption site, for each material type J, this is repeated for NSORB times.

******NOTE: Repeat record 2.1 through 2.3 repeated for NMAT times.

DATA SET 11B ADDITIONAL INFORMATION FOR ION-EXCHANGED SPECIES
This data set is needed only if there are ion-exchanged species (i.e., if $\text{ISCN(i,1)} = 6xx$)

Record 1:

1.1 NSITE = Number of ion-exchange sites

Record 2:
For each NSITE, this record contains 2 entries [$\text{NOMZI(i), MGSZI(j,i)}$] where:

2.1 NOMZI(i) = Number of ion-exchanged species on the i-th site.
2.2 MGSZI(j,i) = Global species number of the j-th ion-exchanged species on the i-th site, and this is repeated NOMZI(i) times.

**DATA SET 12: REACTION INFORMATION** (this data set is needed only if NRXN > 0)

Record 1:

1.1 NRS = Number of species involved in the k-th reaction.
1.2 IGS(i) = Global species number of the i-th species in the k-th reaction.
1.3 RPNU(i) = Stoichiometric coefficient of the i-th species in the k-th reaction (+values for production/products, −values for consumption/reactants). IGS(i+1), RPNU(i+1) are repeated in this line until all information for all species is provided (i.e., NRS times).
1.4 KTYP = Reaction type of the k-th reaction
   - 1 = Equilibrium reactions in-phase complexation (e.g., aqueous complexation) or surface complexation
   - 2 = Equilibrium precipitation-dissolution reaction
   - 3 = Equilibrium ion-exchange reaction
   - 4 = User-specified mass action equation derived for Example #3 and programmed into the accompanying version of the BIOGEOCHEM executable file
   <−4 = Equilibrium reaction represented by user-specified mass action equation, where the user must write their own equilibrium equations and modify the code (contact G.-T. Yeh for code)
0 = Rate is given by the elementary rate
1 = Rate of the k-th reaction is given by "single" Monod kinetics
2 = Rate of the k-th reaction is given by single Monod kinetics with time lag
3 = Rate of the k-th reaction is given by dual Monod kinetics with time lag
4 = Rate of the k-th reaction is given by triple Monod kinetics with time lag
5 = Rate of the k-th reaction is given by n-th-order reaction
N = Rate of the k-th reaction is given by empirical function N

Records 2 and 3 depend on the KTYP selected:
If KTYP = -1, -2 or -3, then record 2 (free format) will contain one entry [EKLOG] where:

2.1 EKLOG = log10 of the equilibrium constant for this reaction

If KTYP \leq -4, then record 2 (free format) will contain several entries [NSP, MGSUS(i), NPR, PARUS(m)] where:

2.1 NSP = Number of species involved in the user specified mass action
2.2 MGSUS(i) = Global species number of the i-th species in this mass action equation. This entry is repeated NSP +1 times.
2.3 NPR = Number of parameters in this mass action equation
2.4 PARUS(m) = the value of the m-th parameter, and this is repeated NPR times (i and m for these entries are iteration counters and have no special meaning)

If KTYP = 0, then record 2 (free format) will contain two entries (NREAC, NPROD) where:

2.1 NREAC = Number of reactants in this reaction
2.2 NPROD = Number of products in this reaction

If KTYP = 0, then record 3 (free format) will contain three entries (FFK1, BBK1, EKLOG) where:

3.1 FFK1 = log10 of the forward rate constant for this kinetic reaction
3.2 BBK1 = log10 of the backward rate constant for this kinetic reaction
3.3 EKLOG = log10 of the equilibrium constant for this reaction

If KTYP > 0, then record 2 (free format) will contain several entries [NSP, IGSLAW(i), NPR, PARLAW(m)] where:

2.1 NSP = Number of species in the user-specified rate law
2.2 IGSLAW(i) = Global species number of the i-th species used in the rate law, and this is repeated NSP times
2.3 NPR = Number of parameters used in the rate law
2.4 PARLAW(m) = The value of the m-th parameter, and this is repeated NPR times.

DATA SET 13: SUSPECTED COMPONENTS
Record 1:

1.1 NR = Number of suspected components

Record 2:

2.1 IW(I) = global species number of the I-th suspected component, and this is repeated NR times

Record 3:

3.1 NCOMP = decomposition option
0 = perform full decomposition of reaction matrix
1 = perform partial decomposition of reaction matrix.

DATA SET 14: IONIC STRENGTH INFORMATION

Record 1:

1.1 SICOR = User's specified ionic strength for computing activity coefficients. (This will be used as a constant ionic strength if the user sets ICOR = 1 below).

1.2 ICOR = Is ionic strength used to correct activity coefficient?
0 = no
1 = constant ionic strength is used (value specified by the user as above)
2 = variable ionic strength is used (as calculated by the program)

1.3 LNH = Location of H⁺ among global species list

1.4 LNE = Location of e⁻ among global species list

1.5 LNW = Location of H₂O among global species list

1.6 LNOH = Location of OH⁻ among global species list

DATA SET 15: TEMP, PRESSURE, AND EXPECTED pE AND pH

Record 1:

1.1 TEMP = absolute temperature, K

1.2 PRESU = pressure, atm

Record 2:

2.1 PEMN = expected minimum pE
2.2 $\text{PEMX} = \text{expected maximum pE}$
2.3 $\text{PHMN} = \text{expected minimum pH}$
2.4 $\text{PHMX} = \text{expected maximum pH}$

**DATA SET 16: INFORMATION ON NUMERICS OF THE SIMULATION**

Record 1:

1.1 $\text{OMEGAC} =$ relaxation parameter for iteration:
   0 ~ 1 = under-relaxation
   1 = exact relaxation
   1 ~ 2 = over relaxation

1.2 $\text{EPS} =$ relative error tolerance for iteration of total concentration solver loop
   (usually $10^{-5} - 10^{-6}$)

1.3 $\text{NITERC} =$ number of iterations allowed for species concentration solver loop
   (usually 100 - 500)

1.4 $\text{NPCYL} =$ number of cycles allowed for iterating precipitation-dissolution
   (set NPCYL = 1 if precipitation-dissolution is not in the simulation, otherwise usually
   $\sim 50$)

1.5 $\text{CNSTRN} =$ a factor for the constraint on complex species concentration. No complex
   species concentration would yield a total component concentration greater
   than CNSTRN times of the total component concentration (usually 2)

1.6 $\text{NSWCH} =$ maximum species switching cycles allowed in each precipitation cycle. (set
   = 0 for no basis switching; set = 2 to do basis switching twice; usually = 3
   is enough)

**DATA SET 17: SPECIES ENTHALPY**

This data set is needed only if $\text{ITEMP}=1$

Record 1:

1.1 $\text{ENTHP(I)} =$ Enthalpy of the I-th species, $[(M/L)/T^3]$

*****NOTE: Repeat record 1.1 for NOCM times.

**DATA SET 18: DEPENDENCE INFORMATION OF FLOW ON CHEMICAL TRANSPORT**
This data set is needed if IMODF = 1 and IMOD = 1

Record 1: FREE FORMAT contains the number of H₂O molecules in the molecular formula of mineral species. This record is needed only if there is mineral species (i.e., ISCN(I,1) = 4) to consider the water removing effects of mineral.

1.1 STWK(1) = number of H₂O molecules in the first mineral species' molecular formula

1.N STWK(N) = number of H₂O molecules in the N-th mineral species' molecular formula

*****NOTE: this input should follow the sequence of mineral species' appearance in the global species list. If no water removing effects are considered, their default value is 0.0.

Record 2: FREE FORMAT contains the molecular weights of all transport components

2.1 WTMOL(1) = The molecular weight of the first species.

2.N WTMOL(N) = The molecular weight of the N-th species (N = NOM)

Record 3: FREE FORMAT contains coefficient to compute density-dependent viscosity.

3.1 AMUWT(1) = Coefficient of the first species to calculate viscosity.

3.N AMUWT(N) = Coefficient of the N-th species to calculate viscosity (N=NOM).

*****NOTE: The values of WTMOL(I) and AMUWT(I) are used to estimate the dependence of fluid density and dynamic viscosity, respectively, on the concentration of the transport component as described in section 2.1.2. For the case without considering the dependence of density and viscosity on the component, both WTMOL(I) and AMUWT(I) should be set as zero.

DATA SET 19: INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS FOR FLOW

This data set is needed only if NSTRF = 0. When this data set is needed, typically a total of NNP lines is required, one each for each node. However, if the initial or pre-initial conditions appear in regular patterns, automatic generation may be used.

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 NIAD = 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 batch input, auxiliary storage input, or steady-state calculation using time-invariant boundary conditions. In the latter case, a batch input of the pre-initial conditions is required as the zero-th order iterate of the steady state solution. Auxiliary storage input is necessary whenever the restarting facility is being used. That is, pressure head for NSTRF different times have been generated and written on disk. If NSTRF > 0, these pressure heads will be read from Logical Unit 14, and the NSTRF-th pressure head will be used as the initial condition for current calculation. If KSTRF > 0, the pressure head values will be written on a different device as they are read from Logical Unit 14 so that a complete record of calculations may be kept on one device, Logical Unit 11. If either the first (batch input) or the last (steady-state) option is used, then NSTRF = 0.

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

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

This data set is needed if IMODF = 1. Four subsets of free-formatted data records are required.

(1). 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.

Subdata sets (2) through (4) are required only when NSEL > 0.

(2). Subset 2: Source/Sink Profiles - For each record, NSDP of the data pair (TSOSF(J,I),SOSF(J,I)) are required. If this 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).

(3). Subset 3: Global Source/Sink Element Number - Usually one record per element. However, automatic
generation can be made. For I-th (I = 1, 2, ..., ) record, it contains the following.

3.1 MI = Compressed element number of the first element in the sequence.
3.2 NSEQ = NSEQ elements will be generated automatically.
3.3 MIAD = Increment of element number for each of the NSEQ elements.
3.4 M = Global element number of element MI.
3.5 MAD = Increment number of M for each of the NSEQ elements.

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

(4). 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 = Compressed element number of the first element in the sequence.
4.2 NSEQ = NSEQ elements will be generated automatically.
4.3 MIAD = Increment of element number for each of the NSEQ elements.
4.4 MITYP = Source type in element MI.
4.5 MITYPAD = 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

This data set is needed if IMODF = 1. Four subsets of free-formatted data records are required.

(1). 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.

Subdata sets (2) through (4) are required only when NWNP > 0.

(2). Subset 2: Source/Sink Profiles - For each record, NWDP of the data pair (TWSSF(J,I),WSSF(J,I)) are required. If this 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, \((L^{**3/T}).\)

(3). Subset 3: Global Source/Sink Node Number - Usually one record per node. However, automatic generation can be made. For I-th (I = 1, 2, ..., ) record, it contains the following.

3.1 NI = Compressed well node number of the first node in the sequence.
3.2 NSEQ = NSEQ nodes will be generated automatically.
3.3 NIAD = Increment of well node number for each of the NSEQ nodes.
3.4 NODE = Global node number of node NI.
3.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

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

(4). 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 point source/sink node number of the first node in a sequence.
4.2 NSEQ = NSEQ nodes will be automatically generated.
4.3 NIAD = Increment of NI for each of the NSEQ nodes.
4.4 NITYP = Source type of node NI.
4.5 NITYPAD = 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 data set.

DATA SET 22: VARIABLE RAINFALL/EVAPORATION-SEEPAGE BOUNDARY CONDITION

This data set is needed if \( \textbf{IMODF} = 1 \). Seven subsets of data records may be required for this data set.

(1). Subset 1: Control Parameters

1.1 NVES = No. of variable boundary element sides.
1.2 NVNP = No. of variable boundary nodal points.
1.3 NVPR = No. of rainfall profiles.
1.4 NVDP = No. of rainfall data points in each of the NVPR rainfall profiles.
Subdata sets (2) through (7) are required only when NVES > 0.

(2). Subset 2: Rainfall Profiles - NVPR profiles are needed. For each profile, NVDP of the data pair \((TRFLF(J,I), RFLF(J,I))\) are required. If these data pairs can fit in a line, we will need NRPR of data lines.

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

2.2 RFLF(J,I) = Rainfall/evaporation rate of the J-th data point in the I-th profile, (L/T).

(3). Subset 3: Boundary Profile Types Assigned to Each VB side. At most NVES 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 VB 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 rainfall/evaporation profiles assigned to MI.

3.5 MITYPAD = 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.

(4). 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 element side MI.

4.5 I2 = global node number of the second node of element side MI.

4.6 I3 = global node number of the third node of element side MI.

4.7 I4 = global node number of the fourth node of 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 subdata set.

(5). Subset 5: Global Node Number of All Compressed Variable Boundary (VB) Nodes. At most, NVNP records are required for this subdata set.

5.1 NI = Compressed VB node number of the first node in a sequence.

5.2 NSEQ = NSEQ subsequent nodes will be generated automatically.

5.3 NIAD = Increment of NI for each of the subsequent 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 should be used to signal the end of this subdata set.

(6). 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).

6.6 0.0

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

(7). Subset 7: Minimum Pressure Head Allowed in Each NVNP Variable Boundary Nodes. This subdata set 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.
DATA SET 23: DIRICHLET BOUNDARY CONDITIONS FOR FLOW SIMULATION

This data set is needed if IMODF = 1. Four subsets of data records are required for this data set.

(1). 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.

Subdata sets (2) through (4) are required only if NDNP > 0.

(2). Subset 2: Dirichlet-head Profiles - 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).

(3). Subset 3: Dirichlet Nodes - Normally one record per Dirichlet node, i.e. a total of NDNP records. However, if the Dirichlet nodes appear in regular pattern, automatic generation may be made. For I-th (I = 1, 2, ..., ) record, it contains the following variables.

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

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

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

3.4 NODE = global node number for NI.

3.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

*****NOTE: A line with 6 0's must be used to signal the end of this subdata set.
*****NOTE: A line with 5 0's should be used to signal the end of this subdata set.

(4). Subset 4: Boundary Profile Type Assigned to Each Dirichlet Node - Normally one record per Dirichlet node, i.e. a total of NDNP records. However, if the Dirichlet nodes appear in regular pattern, automatic generation may be made. For I-th (I = 1, 2, ...) record, it contains the following variables.

4.1 NI = Compressed Dirichlet node number of the first node in the sequence.
4.2 NSEQ = NSEQ subsequent Dirichlet nodes will be generated automatically.
4.3 NIAD = Increment of NI for each of the NSEQ nodes.
4.4 NITYP = Type of total head profile for node NI.
4.5 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 24: CAUCHY BOUNDARY CONDITIONS FOR FLOW SIMULATIONS

This data set is needed if IMODF = 1. Five subsets of data records are required for this data set.

(1). 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.

Subdata sets (2) through (5) are required only if NCES > 0.

(2). Subset 2: Prescribed Cauchy-flux Profiles - 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 NCPR 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.

(3). Subset 3: Type of Cauchy Flux Profiles Assigned to Each of All NCES Cauchy Boundary 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 element 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 MI.
3.5 MTYPAD  =  Increment of MITYP for each of the NSEQ subsequent sides.

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

(4). 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 Cauchy element side number of the first element side in a sequence.
4.2 NSEQ  =  NSEQ subsequent Cauchy element sides will be generated automatically.
4.3 MIAD  =  Increment of MI for each of the NSEQ subsequent Cauchy element sides.
4.4 I1  =  global node number of the first node of element side MI.
4.5 I2  =  global node number of the second node of element side MI.
4.6 I3  =  global node number of the third node of element side MI.
4.7 I4  =  global node number of the fourth node of element side MI.
4.8 I1AD  =  Increment of I1 for each of the NSEQ subsequent Cauchy element sides.
4.9 I2AD  =  Increment of I2 for each of the NSEQ subsequent Cauchy element sides.
4.10 I3AD  =  Increment of I3 for each of the NSEQ subsequent Cauchy element sides.
4.11 I4AD  =  Increment of I4 for each of the NSEQ subsequent Cauchy element sides.

*****NOTE: A line with 11 0's must be used to signal the end of this subdata set.

(5). Subset 5: Global Node Number of All Compressed Cauchy Nodes - At most NCNP records are needed. However, automatic generation can be made. For I-th (I = 1, 2, ..., ) record, it contains the following variables.

5.1 NI  =  Compressed Cauchy node number of the first node in a sequence.
5.2 NSEQ  =  NSEQ subsequent Cauchy nodes will be generated automatically.
5.3 NIAD = Increment of NI for each of the NSEQ nodes.

5.4 NODE = Global node number for node NI.

5.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

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

DATA SET 25: NEUMANN BOUNDARY CONDITIONS FOR FLOW SIMULATIONS

This data set is needed if IMODF = 1. Five subsets of data records are required for this data set.

(1). 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.

Subdata sets (2) through (5) are required only if NNES > 0.

(2). Subset 2: Prescribed Neumann-flux Profiles - 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.

(3). Subset 3: Type of Neumann flux Profiles Assigned to Each of All NNES Neumann boundary 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 element 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 MI.

3.5 MTYPAD = Increment of MITYP for each of the NSEQ subsequent sides
(4). 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 Neumann element side number of the first element side in a sequence.

4.2 NSEQ = NSEQ subsequent Neumann element sides will be generated automatically.

4.3 MIAD = Increment of MI for each of the NSEQ subsequent Neumann element sides.

4.4 I1 = global node number of the first node of Neumann element side MI.

4.5 I2 = global node number of the second node of Neumann element side MI.

4.6 I3 = global node number of the third node of Neumann element side MI.

4.7 I4 = global node number of the fourth node of Neumann element side MI.

4.8 I1AD = Increment of I1 for each of the NSEQ subsequent Neumann element sides.

4.9 I2AD = Increment of I2 for each of the NSEQ subsequent Neumann element sides.

4.10 I3AD = Increment of I3 for each of the NSEQ subsequent Neumann element sides.

4.11 I4AD = Increment of I4 for each of the NSEQ subsequent Neumann element sides.

*****NOTE: A line with 11 0's must be used to signal the end of this subdata set.

(5). Subset 5: Global Node Number of All Compressed Neumann Nodes - At most NNNP records are needed. However, automatic generation can be made. For I-th (I = 1, 2, ..., ) record, it contains the following variables.

5.1 NI = Compressed Neumann node number of the first node in a sequence.

5.2 NSEQ = NSEQ subsequent Neumann nodes will be generated automatically.

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

5.4 NODE = global node number for node NI.

5.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

*****NOTE: A line with 5 0's should be used to signal the end of this subdata set.
DATA SET 26: RIVER BOUNDARY CONDITIONS FOR FLOW SIMULATIONS

This data set is needed if \textbf{IMODF} = 1. Seven subsets of data records are required for this data set.

(1). Subset 1: Control Parameters

1.1 \textbf{NRNP} = No. of river nodal points.

1.2 \textbf{NRPR} = No. of river-head profiles.

1.3 \textbf{NRDP} = No. of data points in each of the \textbf{NRPR} river-head profiles.

1.4 \textbf{NRES} = No. of river boundary element sides.

1.5 \textbf{NRMA} = No. of river bottom material types.

Subdata sets (2) through (7) are required only if \textbf{NRNP} > 0.

(2). Subset 2: Prescribed River-head Profiles - \textbf{NRPR} of profiles are needed. For each profile, \textbf{NRDP} of the data pair (\textbf{THRBF}(_J,I),\textbf{HRBF}(_J,I)) are needed. If these data pairs can fit in a line, we will need \textbf{NRPR} lines.

2.1 \textbf{THRBF}(_J,I) = Time of the \textbf{J}-th data point in the \textbf{I}-th profile, (T).

2.2 \textbf{HRBF}(_J,I) = River head of the \textbf{J}-th data point in the \textbf{I}-th profile, (L).

(3). Subset 3: Global Node Number of All Compressed River Nodes - At most \textbf{NrNP} records are needed. However, automatic generation can be made. For \textbf{I}-th (I = 1, 2, ...) record, it contains the following variables.

3.1 \textbf{NI} = Compressed river node number of the first node in a sequence.

3.2 \textbf{NSEQ} = NSEQ subsequent river nodes will be generated automatically.

3.3 \textbf{NIAD} = Increment of \textbf{NI} for each of the NSEQ nodes.

3.4 \textbf{NODE} = global node number for node \textbf{NI}.

3.5 \textbf{NODEAD} = Increment of \textbf{NODE} for each of the NSEQ subsequent nodes.

*****\textbf{NOTE}: A line with 5 0's should be used to signal the end of this subdata set.

(4). Subset 4: Type of River-head Profiles Assigned to Each of all \textbf{NRNP} Nodes. At most \textbf{NRNP} records are needed. However, automatic generation can be made. For \textbf{I}-th (I = 1, 2, ...) record, it contains the following variables.

4.1 \textbf{NI} = Compressed river node number of the first node in the sequence.
4.2 NSEQ = NSEQ subsequent river nodes will be generated automatically.

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

4.4 NITYP = Type of river head profile for node NI.

4.5 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.

(5). Subset 5: River Boundary Element Sides - Normally, NRES records are required, one each for a river boundary element side. However, if a group of river 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.

5.1 MI = Compressed river element side number of the first element side in a sequence.

5.2 NSEQ = NSEQ subsequent river element sides will be generated automatically.

5.3 MIAD = Increment of MI for each of the NSEQ subsequent river element sides.

5.4 I1 = global node number of the first node of river element side MI.

5.5 I2 = global node number of the second node of river element side MI.

5.6 I3 = global node number of the third node of river element side MI.

5.7 I4 = global node number of the fourth node of river element side MI.

5.8 I1AD = Increment of I1 for each of the NSEQ subsequent river element sides.

5.9 I2AD = Increment of I2 for each of the NSEQ subsequent river element sides.

5.10 I3AD = Increment of I3 for each of the NSEQ subsequent river element sides.

5.11 I4AD = Increment of I4 for each of the NSEQ subsequent river element sides.

*****NOTE: A line with 11 0's must be used to signal the end of this subdata set.

(6). Subset 6: River Bottom Material Types - NRMA data points are needed for this data set. Each data point contains two number: the first number is the river bottom hydraulic conductivity and the second number is the river bottom thickness. FREE FORMAT

6.1 PROR(1,1) = river bottom hydraulic conductivity of the first data point.

6.2 PROR(2,1) = river bottom thickness of the first data point.

6.3 PROR(1,2) = river bottom hydraulic conductivity of the second data point.
6.4 \( \text{PROR(2,2)} \) = river bottom thickness of the second data point.

6.2*I-1 \( \text{PROR(1,I)} \) = river bottom hydraulic conductivity of the I-th data point.

6.2*I \( \text{PROR(2,I)} \) = river bottom thickness of the I-th data point.

(7). Subset 7: River Bottom Material Types Assigned to Each of the NRES Boundary Sides - This subdata set contains the following variables: FREE FORMAT

7.1 \( \text{ISR(5,1)} \) = river bottom material type assigned to the first river boundary side.

7.2 \( \text{ISR(5,2)} \) = river bottom material type assigned to the second river boundary side.

7.1 \( \text{ISR(5,1)} \) = river bottom material type assigned to the I-th river boundary side.

7.N \( \text{ISR(5, N)} \) = river bottom material type assigned to the NRES-th river boundary side.

**DATA SET 27: VELOCITY AND MOISTURE CONTENT**

This data set is needed either (IMODF= 0 and IMOD = 1) or ITEMP = 1, and If KVI > 0, this data set is not needed because these will be read via Logical Unit 17.

Three subsets of free-formatted data records are required.

(1). Subset 1: Velocity Field - Normally, one line per node is needed. However, automatic generation can be used. Each line is free-field format input containing seven variables:

1.1 NI = Node number of the first node in the sequence.

1.2 NSEQ = NSEQ subsequent nodes will be automatically generated.

1.3 NIAD = Increment of NI in each of the NSEQ subsequent nodes.

1.4 VXNI = \( x \)-velocity component of node NI (L/T).

1.5 VYNI = \( y \)-velocity component of node NI (L/T).

1.6 VZNI = \( z \)-velocity component of node NI (L/T).

1.7 VXAD = Increment of \( x \)-velocity for each of the NSEQ subsequent nodes (L/T).

1.8 VYAD = Increment of \( y \)-velocity for each of the NSEQ subsequent nodes (L/T).
1.9 VZAD = Increment of z-velocity for each of the NSEQ subsequent nodes (L/T).

*****NOTE: A line with 9 0's is used to signal the end of this subdata set.

(2). Subset 2: Moisture Content Field - Typically, one line per element is needed. However, automatic generation can be used. The input values should be able to represent the efficient moisture content with the effect of mineral precipitations within the simulation site. Each line is free-field format input containing five variables:

2.1 MI = Element number of the first element in the 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 THMI = Moisture content of element MI (decimal point).
2.5 THMIAD = Increment of THMI for each of the NSEQ subsequent elements.

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

(3). Subset 3 Fluid Density Field - Typically, one line per element is needed. However, automatic generation can be used. Each line is free-field format input containing five variables:

3.1 MI = Element number of the first element in the sequence.
3.2 NSEQ = NSEQ subsequent elements will be automatically generated.
3.3 MIAD = Increment of MI for each of the NSEQ subsequent elements.
3.4 RHOMI = Fluid density of element MI (decimal point).
3.5 RHOMIAD = Increment of RHOMI for each of the NSEQ subsequent elements.

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

DATA SET 28: INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS FOR TRANSPORT

This data set is needed only if NSTR = 0. When this data set is needed, typically a total of NSP*NNP records are required for each chemical species, one for each node. However, if the initial or pre-initial conditions appear in regular patterns, automatic generation may be used.

(1). Subset 1: Species Number That Need to Input Initial or Pre-initial Concentration

1.1 NSP = total number of species' concentration to be input
1.2 IWORK(1) = global species number of the first species in the series
1. $N+1$ \text{IWORK}(N) \quad = \quad \text{global species number of the } N\text{-th species in the series } (N = \text{NSP})$

(2). Subset 2: Each Record Is Free-field Format Input Containing the Following Information (needed if \text{NSP} \geq 0):

2.1 \text{NI} \quad = \quad \text{Global node number of the first node in the sequence.}

2.2 \text{NSEQ} \quad = \quad \text{NSEQ subsequent nodes will be generated automatically.}

2.3 \text{NIAD} \quad = \quad \text{Increment of the node number for each of the NSEQ nodes.}

2.4 \text{RNI} \quad = \quad \text{Initial or pre-initial total concentration at node NI (moles/liter).}

2.5 \text{RAD} \quad = \quad \text{Increment of initial or pre-initial total concentration for each of the NSEQ nodes (moles/liter).}

2.6 \text{RRD} \quad = \quad \text{Percent increase of the increment over its preceding increment (decimal fraction): }\nonumber
0 = \text{All increments, XAD's, are the same.} 
> 0 = \text{The first increment is XAD*(1+XRD), the second increment is XAD*(1+XRD)**2, the third increment is XAD*(1+XRD)**3, and so on.}

*****NOTE: A line with 6 0's is used to signal the end of this subdata set.

This record is repeated for \text{NSP} times.

*****NOTE ON INITIAL CONDITIONS: The initial conditions for a transient calculation may be obtained in three different ways: from batch input, auxiliary storage input, or steady-state calculation using time-invariant boundary conditions. In the latter case, a batch input of the pre-initial conditions is required as the zero-th order iterate of the steady state solution. Auxiliary storage input is necessary whenever the restarting facility is being used. That is, concentration distribution for \text{NSTR} different times have been generated and written on disk or magnetic tape. If \text{NSTR} > 0, these distributions will be read from Logical Unit 13, and the \text{NSTR}-th distribution will be used as the initial condition for current calculation. If \text{KSTR} > 0, the concentration values will be written on a different device as they are read from Logical Unit 13 so that a complete record of calculations may be kept on one device, Logical Unit 12. If either the first (batch input) or the last (steady-state) option is used, then \text{NSTR} = 0.

*****NOTE ON AUXILIARY UNITS: Logical Unit 17 is used to input hydrodynamic variables to HYDROGEOCHEM if KVI > 0. Logical Unit 12 is used to store output of HYDROGEOCHEM if KSTR > 0. Logical Unit 13 is used to input initial condition if \text{NSTR} > 0. Proper identification of these three units must be made if either of these options is used. The file name for Logical Unit 13 of the current job should be the same as that for Logical Unit 12 of the previous job.

*****NOTE ON STEADY-STATE INPUT: A 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 former case KSS = 0 and NTI = 0, and in the latter case KSS = 0 and NTI > 0. If KSS > 0, there will be no steady-state calculation.
DATA SET 29: ELEMENT (DISTRIBUTED) SOURCE/SINK FOR TRANSPORT

This data set is needed if \( \text{IMOD} = 1 \). Five subsets of free-formatted data records are required.

(1). 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.

Subdata sets (2) through (5) are needed if and only if \( \text{NSEL} > 0 \).

(2). Subset 2: Global Source/Sink Element Number - Usually one record per element. However, automatic generation can be made. For I-th (\( I = 1, 2, \ldots \) ) record, it contains the following.

2.1 MI  =  Compressed element number of the first element in the sequence.
2.2 NSEQ  =  NSEQ elements will be generated automatically.
2.3 MIAD  =  Increment of element number for each of the NSEQ elements.
2.4 M  =  global element of element MI.
2.5 MAD  =  Increment number of M for each of the NSEQ elements.

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

(3). Subset 3: Species Number That Need to Input Source/Sink Profile

3.1 NSP  =  total number of species' source/sink profile to be input
3.2 IWORK(1)  =  global species number of the first species in the series
. N+1  IWORK(N)  =  global species number of the N-th species in the series (\( N = \text{NSP} \))

(4). Subset 4: Element Source/Sink Profile: Number of lines depends on NSPR, NSDP and NSP. Each record contains a number of dat points = 2*NSDP numerical numbers.

For \( K=1,\text{NSP}+2 \)
For \( I=1, \text{NSPR} \)

4.1 TSOSF(1,I,K)  =  Time of first data point in I-th profile for K-th species (T).
4.2 SOSF(1,I,K)  =  Source/sink value of first data point in I-th profile for K-th species.
4.3 TSOSF(2,I,K)  =  Time of second data point in I-th profile for K-th species (T).
4.4 \( \text{SOSF}(2,I,K) \) = Source/sink value of second data point in I-th profile for K-th species.

\[
\text{TSOSF}(N,I,K)= \text{Time of } N\text{-th data point in I-th profile for } K\text{-th species (T).}
\]

4.2N-1 \( \text{SOSF}(N,I,K) \) = Source/sink value of N-th data point in I-th profile for K-th species:

*** NOTE: Value of \( \text{SOSF}(*,*,*) \) is concentration (moles/liter) if \( K \leq \text{NSP} \); Values beyond \( K>\text{NSP} \) are: flow rate of water \( (L^3/T) \) if \( K=(\text{NSP}+1) \); water density \( (M/L^3) \) if \( K=(\text{NSP}+2) \).

(5). Subset 5: Source/Sink Profile Type in Each Source/Sink Element: Typically, one line per element is needed. However, automatic generation can be used. Each line is free-field format input containing the following:

5.1 \( \text{MI} \) = Compressed source/sink element number of first element in the sequence.

5.2 \( \text{NSEQ} \) = NSEQ source/sink elements will have profile type \( \text{MITYP} \).

5.3 \( \text{MIAD} \) = Increment of \( \text{MI} \) for each of the NSEQ elements.

5.4 \( \text{MITYP} \) = Source/sink profile type for element \( \text{MI} \).

5.5 \( \text{MTYPAD} = \) Increment of source profile type in 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 30: WELL (POINT) SOURCE/SINK FOR TRANSPORT**

This data set is needed if \( \text{IMOD} = 1 \). Five subsets of free-formatted data records are required.

(1). Subset 1: Control Parameters

1.1 \( \text{NWNP} \) = No. of well or point source/sink nodal points.

1.2 \( \text{NWPR} \) = No. of well or point source/sink strength profiles.

1.3 \( \text{NWDP} \) = No. of data points in each of the NWPR profiles.

Subdata sets (2) through (5) are needed if and only if \( \text{NWNP} > 0 \).

(2). Subset 2: Global Node Number of Compressed Point Source/Sink Number: Free-field format input contains the following information:

2.1 \( \text{NI} \) = Compressed point source/sink node number of the first node in a sequence.
2.2 NSEQ = NSEQ subsequent nodes will be generated automatically.

2.3 NIAD = Increment of NI for each of the NSEQ subsequent nodes.

2.4 NODE = global node of the compressed node NI.

2.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

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

(3). Subset 3: Species Number That Need to Input Source/Sink Profile

3.1 NSP = total number of species' source/sink profile to be input

3.2 IWORK(1) = global species number of the first species in the series

3.N+1 IWORK(N) = global species number of the N-th species in the series (N = NSP)

(4). Subset 4: Point Source/Sink Profile: Number of lines depends on NWPR, NWDP and NSP. Each record contains a number of dat points = 2*NWDP numerical numbers.

For K=1,NSP+2
For I = 1, NWPR

4.1 TWSSF(1,I,K) = Time of first data point in I-th profile for K-th species (T).

4.2 WSSF(1,I,K) = Source/sink value of first data point in I-th profile for K-th species:

4.3 TWSSF(2,I,K) = Time of second data point in I-th profile for K-th species (T).

4.4 WSSF(2,I,K) = Source/sink value of second data point in I-th profile for K-th species:


4.2N WSSF(N,I,K) = Source/sink value of N-th data point in I-th profile for K-th species:

*** NOTE: Value of WSSF(*,*,*) is concentration (moles/liter) if K<= NSP;
Values beyond K>NSP are: flow rate of water(L**3/T) if K=(NSP+1);
water density (M/L**3) if K=(NSP+2).

(5). Subset 5: Type of Point Source/Sink Nodes: Typically, one line per well node is needed. However, automatic generation may be used. Each line is free-field format input containing the following:

5.1 NI = Compressed well node number of the first node in a sequence.
5.2 NSEQ = NSEQ subsequent well nodes will be generated automatically.
5.3 NIAD = Increment of NI for each of the NSEQ subsequent nodes.
5.4 NITYP = Type of well source/sink profile assigned to NI-th well node.
5.5 NITYPAD = 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 data set.

DATA SET 31.: VARIABLE BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

This data set is needed if IMOD = 1. Six subsets of data records may be required for this data set.

(1). Subset 1: Control Parameters

1.1 NVES = No. of variable boundary element sides.
1.2 NVNP = No. of variable boundary nodal points.
1.3 NVPR = No. of incoming concentration profiles,
1.4 NVDP = No. of concentration data points in each of the NVPR incoming concentration profiles.

Subdata sets (2) through (6) are needed if and only if NVES > 0.

(2). Subset 2: Species Number That Need to Input Incoming Concentration Profile

2.1 NSP = total number of species' Cauchy flux profile to be input
2.2 IWORK(1) = global species number of the first species in the series
2.N+1 IWORK(N) = global species number of the N-th species in the series (N = NSP)

(3). Subset 3: Incoming Concentration Profiles.

For K=1,NSP
For I = 1, NVPR

3.1 TCVBF(1,I,K) = Time of first data point in I-th incoming concentration versus time profile for K-th species (T).
3.2 CVBF(1,I,K) = Concentration of first data point in I-th incoming concentration versus time profile for K-th species: (moles/liter).
3.3 TCVBF(2,I,K) = Time of second data point in I-th incoming concentration versus
time profile for K-th species (T).

3.4 $CVBF(2,I,K)$ = Concentration of second data point in I-th incoming concentration versus time profile for K-th species: (moles/liter).

3.2*N-1 $TCVBF(N,I,K)$ = Time of the N-th data point in I-th incoming concentration versus time profile for K-th species (T).

3.2*N $CVBF(N,I,K)$ = Concentration of the N-th data point in I-th incoming concentration versus time profile for K-th species: (moles/liter).

(4). Subset 4: Incoming Concentration Profile Type Assigned to Variable-boundary (VB) Sides.

4.1 MI = Compressed VB side number of the first side in a sequence.

4.2 NSEQ = NSEQ subsequent sides will be generated automatically.

4.3 MIAD = Increment of the MI for each of the NSEQ subsequent sides.

4.4 MITYP = Type of incoming concentration profile assigned to side MI.

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

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

(5). Subset 5: Global Nodal Number of NVNP Variable-boundary (VB) Condition Nodes: Typically, NVNP lines are needed. However, automatic generation can be used. Each line is free-field format input containing the following:

5.1 NI = Compressed VB node number of the first node in a sequence.

5.2 NSEQ = NSEQ subsequent nodes will be generated automatically.

5.3 NIAD = Increment of NI for each of the NSEQ subsequent nodes.

5.4 NODE = global node of the compressed node NI.

5.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

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

(6). Subset 6: Specification of Variable-boundary (VB) Sides: Typically, NVES lines are required, one each for a variable-boundary element side. However, if a group of variable-boundary element sides appears in a regular pattern, automatic generation may be used. Each line is free-field format input containing the following:

6.1 MI = Compressed VB element-side number of the first element side in a
6.2 NSEQ = NSEQ subsequent VB element sides will be generated automatically.
6.3 MIAD = Increment of MI for each of the NSEQ subsequent VB element sides.
6.4 I1 = global VB nodal number of the first node of element side MI.
6.5 I2 = global VB nodal number of the second node of element side MI.
6.6 I3 = global VB nodal number of the third node of element side MI.
6.7 I4 = global VB nodal number of the fourth node of element side MI.
6.8 I1AD = Increment of I1 for each of the NSEQ subsequent VB element sides.
6.9 I2AD = Increment of I2 for each of the NSEQ subsequent VB element sides.
6.10 I3AD = Increment of I3 for each of the NSEQ subsequent VB element sides.
6.11 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 data set.

DATA SET 32: CAUCHY BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

This data set is needed if IMOD = 1. Six subsets of data records may be required for this data set.

(1). Subset 1: Control Parameters

1.1 NCES = number of Cauchy boundary element sides
1.2 NCNP = number of Cauchy boundary nodal points
1.3 NCPR = number of Cauchy-flux profiles
1.4 NCDP = number of concentration data points in each of the NCPR Cauchy-flux profiles

Subdata sets (2) through (6) are needed if and only if NCES > 0.

(2). Subset 2: Species Number that Need to Input Cauchy-flux Profile

2.1 NSP = total number of species' Neumann flux profile to be input
2.2 IWORK(1) = global species number of the first species in the series
2.N+1 IWORK(N) = global species number of the N-th species in the series (N = NSP)
(3). Subset 3: Cauchy Flux Profile

For K=1,NSP
For I=1,NCPR

3.1 \( TCCBF(1,I,K) \) = time of first data in I-th data point in the I-th profile for IWORK(K)-th species

3.2 \( CCBF(1,I,K) \) = Normal Cauchy flux of first data point in I-th profile for IWORK(K)-th species

3.3 \( TCCBF(2,I,K) \) = time of second data in I-th data point in the I-th profile for IWORK(K)-th species

3.4 \( CCBF(2,I,K) \) = Normal Cauchy flux of second data point in I-th profile for IWORK(K)-th species

3.N-1 \( TCCBF(N,I,K) \) = time of N-th data in I-th data point in the I-th profile for IWORK(K)-th species

3.N \( CCBF(N,I,K) \) = Normal Cauchy flux of N-th data point in I-th profile for IWORK(K)-th species

(4). Subset 4: Type of Cauchy Flux Profile Assigned to Cauchy-boundary Sides

4.1 \( MI \) = Compressed Cauchy element side number of the first side in a sequence

4.2 \( NSEQ \) = NSEQ subsequent sides will be generated automatically

4.3 \( MIAD \) = increment of MI for each of the NSEQ subsequent sides

4.4 \( MITYP \) = Type of Cauchy flux profile assigned to side MI

4.5 \( MITYPAD \) = increment of MITYP for each of the NSEQ subsequent sides

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

(5). Subset 5: Global Nodal Number of NCNP Cauchy Boundary Condition Nodes: Typically, NCNP lines are needed. However, automatic generation can be used. Each line is free-field format input containing the following:

5.1 \( NI \) = compressed Cauchy flux node number of the first node in a sequence

5.2 \( NSEQ \) = NSEQ subsequent nodes will be generated automatically

5.3 \( NIAD \) = increment of NI for each of the NSEQ subsequent nodes

230
5.4 \( \text{NODE} \) = global node of the compressed node \( NI \)

5.5 \( \text{NODEAD} \) = increment of \( \text{NODE} \) for each of the \( \text{NSEQ} \) subsequent nodes

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

(6). Subset 6: Specification of Cauchy Boundary Element Sides: Typically, NCES lines 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 used. Each line is free-field format input containing the following:

6.1 \( \text{MI} \) = compressed Cauchy element side number of the first element side in a sequence

6.2 \( \text{NSEQ} \) = \( \text{NSEQ} \) subsequent Cauchy element sides will be generated automatically

6.3 \( \text{MIAD} \) = increment of \( \text{MI} \) for each of the \( \text{NSEQ} \) subsequent Cauchy element sides

6.4 \( \text{I1} \) = global node number of the first node of element side \( \text{MI} \)

6.5 \( \text{I2} \) = global node number of the second node of element side \( \text{MI} \)

6.6 \( \text{I3} \) = global node number of the third node of element side \( \text{MI} \)

6.7 \( \text{I4} \) = global node number of the fourth node of element side \( \text{MI} \)

6.8 \( \text{I1AD} \) = increment of \( \text{I1} \) for each of the \( \text{NSEQ} \) subsequent Cauchy element sides

6.9 \( \text{I2AD} \) = increment of \( \text{I2} \) for each of the \( \text{NSEQ} \) subsequent Cauchy element sides

6.10 \( \text{I3AD} \) = increment of \( \text{I3} \) for each of the \( \text{NSEQ} \) subsequent Cauchy element sides

6.11 \( \text{I4AD} \) = increment of \( \text{I4} \) for each of the \( \text{NSEQ} \) subsequent Cauchy element sides

**** NOTE: A line with 11 0's must be used to signal the end of this data set.

DATA SET 33: NEUMANN BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

This data set is needed if \( \text{IMOD} = 1 \). Six subsets of data records may be required for this data set.

(1). Subset 1: Control Parameters

1.1 \( \text{NNES} \) = number of Neumann boundary element sides

1.2 \( \text{NNNP} \) = number of Neumann boundary nodal points

1.3 \( \text{NNPR} \) = number of Neumann flux profiles
1.4 NNDP = number of concentration data points in each of the NNPR Neumann flux profiles

Subdata sets (2) through (6) are needed if and only if \( \text{NDES} > 0 \).

(2). Subset 2: Species Number That Need to Input Neumann Flux Profile

2.1 NSP = total number of species' Neumann flux profile to be input

2.2 IWORK(1) = global species number of the first species in the series

2.N+1 IWORK(N) = global species number of the N-th species in the series (N = NSP)

(3). Subset 3: Neumann flux profile

For \( K=1,\text{NSP} \)

For \( I=1,\text{NNPR} \)

3.1 TCNBF(1,I,K) = time of first data in I-th data point in the I-th profile for IWORK(K)-th species

3.2 CNBF(1,I,K) = Normal Neumann flux of first data point in I-th profile for IWORK(K)-th species

3.3 TCNBF(2,I,K) = time of second data in I-th data point in the I-th profile for IWORK(K)-th species

3.4 CNBF(2,I,K) = Normal Neumann flux of second data point in I-th profile for IWORK(K)-th species

3.N-1 TCNBF(N,I,K) = time of N-th data in I-th data point in the I-th profile for IWORK(K)-th species

3.N CNBF(N,I,K) = Normal Neumann flux of N-th data point in I-th profile for IWORK(K)-th species

(4). Subset 4: Type of Neumann Flux Profile Assigned to Neumann-boundary Sides

4.1 MI = Compressed Neumann element side number of the first side in a sequence

4.2 NSEQ = NSEQ subsequent sides will be generated automatically

4.3 MIAD = increment of MI for each of the NSEQ subsequent sides

4.4 MITYP = Type of Neumann flux profile assigned to side MI

4.5 MTYPAD = increment of MITYP for each of the NSEQ subsequent sides

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

232
(5). Subset 5: Global Nodal Number of NNNP Neumann Boundary Condition Nodes: Typically, NNNP lines are needed. However, automatic generation can be used. Each line is free-field format input containing the following:

5.1 NI = compressed Neumann flux node number of the first node in a sequence
5.2 NSEQ = NSEQ subsequent nodes will be generated automatically
5.3 NIAD = increment of NI for each of the NSEQ subsequent nodes
5.4 NODE = global node of the compressed node NI
5.5 NODEAD = increment of NODE for each of the NSEQ subsequent nodes

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

(6). Subset 6: Specification of Neumann Boundary Element Sides: Typically, NNES lines 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 used. Each line is free-field format input containing the following:

6.1 MI = compressed Neumann element side number of the first element side in a sequence
6.2 NSEQ = NSEQ subsequent Neumann element sides will be generated automatically
6.3 MIAD = increment of MI for each of the NSEQ subsequent Neumann element sides
6.4 I1 = global node number of the first node of element side MI
6.5 I2 = global node number of the second node of element side MI
6.6 I3 = global node number of the first node of element side MI
6.7 I4 = global node number of the second node of element side MI
6.8 I1AD = increment of I1 for each of the NSEQ subsequent Neumann element sides
6.9 I2AD = increment of I2 for each of the NSEQ subsequent Neumann element sides
6.10 I3AD = increment of I3 for each of the NSEQ subsequent Neumann element sides
6.11 I4AD = increment of I4 for each of the NSEQ subsequent Neumann element sides

**** NOTE: A line with 11 0's must be used to signal the end of this data set.
DATA SET 34: DIRICHLET BOUNDARY CONDITIONS FOR TRANSPORT SIMULATIONS

This data set is needed if IMOD = 1. Five subsets of data records may be required for this data set.

(1). Subset 1: Control Parameters

1.1 NDNP = No. of Dirichlet nodal points, should be .GE. 1.
1.2 NDPR = No. of total Dirichlet-concentration profiles, should be .GE. 1.
1.3 NDDP = No. of data points in each total concentration profiles, should be .GE. 1.

Subdata sets (2) through (5) are needed if and only if NDNP > 0.

(2). Subset 2: Species Number that Need to Input Neumann Flux Profile

2.1 NSP = total number of species' Dirichlet concentration profile to be input
2.2 IWORK(1) = global species number of the first species in the series
\[ \text{2.N+1 IWORK(N) = global species number of the N-th species in the series (} N = \text{NSP)} \]

(3). Subset 3: Dirichlet Concentration Profile

For K=1,NSP
For I = 1, NDPR

3.1 TCDBF(1,I,K) = Time of first data point in I-th Dirichlet concentration versus time profile for K-th species (T).
3.2 CDBF(1,I,K) = Concentration of first data point in the I-th Dirichlet concentration versus time profile for K-th species: (moles/liter).
3.3 TCDBF(2,I,K) = Time of second data point in the I-th Dirichlet concentration versus time profile for K-th species (T).
3.4 CDBF(2,I,K) = Concentration of second data point in the I-th Dirichlet concentration versus time profile for K-th species: (moles/liter).
\[ \text{.} \]
\[ \text{.} \]
\[ \text{.} \]
3.2*N-1 TCDBF(N,I,K) = Time of the N-th data point in the I-th Dirichlet concentration versus time profile for K-th species (T).
3.2*N CDBF(N,I,K) = Concentration of the N-th data point in the I-th Dirichlet concentration versus time profile for K-th species: (moles/liter).
(4). Subset 4: Type of Dirichlet Node.

4.1 NI = Compressed Dirichlet node number of the first node in the sequence.
4.2 NSEQ = NSEQ subsequent Dirichlet nodes will be generated automatically.
4.3 NIAD = Increment of NI for each of the NSEQ nodes.
4.4 NITYP = Type of Dirichlet concentration profile for node NI.
4.5 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 data set.

(5). Subset 5: Dirichlet Nodes:

5.1 NI = Compressed Dirichlet boundary node number of the first node in a sequence.
5.2 NSEQ = NSEQ subsequent nodes will be generated automatically.
5.3 NIAD = Increment of NI for each of the NSEQ subsequent nodes.
5.4 NODE = Global node of the compressed node NI.
5.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes.

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

DATA SET 35: INPUT FOR INITIAL OR PRE-INITIAL CONDITIONS FOR HEAT TRANSFER

This data set is needed only if ITEMP = 1 and NSTR = 0.

Each record is free-field format input containing the following information.

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 NIAD = increment of the node number for each of the NSEQ nodes.
1.4 RNI = initial or pre-initial temperature for each of the NSEQ nodes (K).
1.5 RAD = increment of initial or pre-initial temperature for each of the NSEQ nodes (K).
1.6 RRD = percent increase of the increment over its preceding increment (decimal fraction):
0 = all increments, RADs, are the same;
> 0 = the first increment is RAD*(1+RRD), the second increment is RAD*(1+RRD)**2, the third is RAD*(1+RRD)**3, and so on.

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

DATA SET 36: ELEMENT (DISTRIBUTED) SOURCE/SINK FOR HEAT TRANSFER

This data set is needed if ITEMP = 1. Four subsets of free-formatted data records are required.

(1). Subset 1: Control Parameters

1.1 NSELT = number of source/sink elements
1.2 NSPRTR = number of source/sink profiles
1.3 NSDPT = number of data points in each of the NSPRTR source/sink profile

Subdata sets (2) through (4) are needed if and only if NSELT > 0.

(2). Subset 2: Source/sink Profiles - For each record, NSDPT of the data pair (TSOST(J,I),SOSFT(J,I)) are required. If this record can be fitted in a line, we will need NSPRTR lines.

2.1 TSOST(J,I) = Time of the J-th data point in the I-th profile, (T)
2.2 SOSFT(J,I) = Source/sink value of the J-th data point in the I-th profile, (K)

(3). Subset 3: Global Source/sink Element Number - Usually one record per element. However, automatic generation can be made. For I-th (I = 1, 2, ..., ) record, it contains the following.

3.1 MI = Compressed element number of the first element in the sequence.
3.2 NSEQ = NSEQ source/sink elements will be generated automatically.
3.3 MIAD = Increment of MI for each of the NSEQ elements.
3.4 M = global element number of element MI.
3.5 MAD = Increment number of M 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.

(4). Subset 4: Source/sink Profile Type Assigned to Each Source/sink 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 = Compressed source/sink element number of first element in the sequence.
4.2 NSEQ = NSEQ source/sink elements will be generated automatically.

4.3 MIAD = Increment of MI for each of the NSEQ elements.

4.4 MITYP = Source/sink profile type in element MI.

4.5 MITYPAD = 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 37: WELL(POINT) SOURCE/SINK FOR HEAT TRANSFER**

This data set is needed if **ITEMP = 1**. Four subsets of free-formatted data records are required.

(1). Subset 1: Control Parameters

1.1 NWNPT = number of well or point source/sink nodal points

1.2 NWPRT = number of well or point source/sink strength profiles

1.3 NWDPT = number of data points in each of the NWPRT source/sink profile

Subdata sets (2) through (4) are needed if and only if **NWNPT > 0**.

(2). Subset 2: Point Source/sink Profile: For each record, NWDPT of the data pair (TWSST(J,I),WSSFT(J,I)) are required. If this record can be fitted in a line, we will need NWPRT lines.

2.1 TWSST(J,I) = time of J-th data in I-th profile, (T)

2.2 WSSFT(J,I) = source/sink value of J-th data point in I-th profile, (K)

(3). Subset 3: Global Source/sink Node Number - Usually one record per node. However, automatic generation can be made. For I-th (I = 1, 2, ..., ) record, it contains the following.

3.1 NI = Compressed point source/sink node number of the first node in a sequence.

3.2 NSEQ = NSEQ subsequent nodes will be generated automatically

3.3 NIAD = Increment of NI for each of the NSEQ subsequent nodes

3.4 NODE = Global node of the compressed node NI

3.5 NODEAD = Increment of NODE for each of the NSEQ subsequent nodes

**** NOTE: A line with 5 0's must be used to signal the end of this data set.
(4). Subset 4: Type of Source/sink Node - Usually one record per node. However, automatic generation can be made. For I-th (I = 1, 2, ..., ) record, it contains the following.

4.1 NI = Compressed source/sink node number of first node in the sequence
4.2 NSEQ = NSEQ source/sink nodes will be generated automatically
4.3 NIAD = Increment of NI for each of the NSEQ nodes
4.4 NITYP = Source/sink profile type in node NI
4.5 NITYPAD = 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 data set.

DATA SET 38: VARIABLE BOUNDARY CONDITIONS FOR HEAT TRANSFER SIMULATIONS

This data set is needed if ITEMP = 1. Five subsets of free-formatted data records are required.

(1). Subset 1: Control Parameters

1.1 NVEST = number of variable boundary element sides
1.2 NVNPT = number of variable boundary nodal points
1.3 NVPRT = number of incoming temperature profiles
1.4 NVDPT = number of temperature data points in each of the NVPRT incoming temperature profiles

Subdata sets (2) through (5) are needed if and only if NVEST > 0.

(2). Subset 2: Incoming Temperature Profiles. For each record, NVDPT of the data pair (TTVBF(J,I), TVBF(J,I)) are required. If this record can be fitted in a line, we will need NVPRT lines.

2.1 TTVBF(J,I) = time of J-th data point in the I-th incoming temperature profile, (T)
2.2 TVBF(J,I) = temperature of J-th data point in I-th profile, (K)

(3). Subset 3: Type of Incoming Temperature Profile Assigned to Variable-boundary Sides

3.1 MI = Compressed VB side number of the first side in a sequence
3.2 NSEQ = NSEQ subsequent sides will be generated automatically
3.3 MIAD = increment of MI for each of the NSEQ subsequent sides
3.4 MITYP = Type of incoming temperature profile assigned to side MI
3.5 MTYPAD= increment of MITYP for each of the NSEQ subsequent sides

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

(4). Subset 4: Global Nodal Number of NVNPT Variable Boundary(VB) Condition Nodes: Typically, NVNPT lines are needed. However, automatic generation can be used.

4.1 NI = compressed VB node number of the first node in a sequence
4.2 NSEQ = NSEQ subsequent nodes will be generated automatically
4.3 NIAD = increment of NI for each of the NSEQ subsequent nodes
4.4 NODE = global node of the compressed node NI
4.5 NODEAD= increment of NODE for each of the NSEQ subsequent nodes

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

(5). Subset 5: Specification of Variable Boundary (VB) Sides: Typically, NVEST lines are required, one each for a variable-boundary element side. However, if a group of variable-boundary element sides appears in a regular pattern, automatic generation may be used.

5.1 MI = compressed VB element side number of the first element side in a sequence
5.2 NSEQ = NSEQ subsequent VB element sides will be generated automatically
5.3 MIAD = increment of MI for each of the NSEQ subsequent VB element sides
5.4 I1 = global VB nodal number of the first node of element side MI
5.5 I2 = global VB nodal number of the second node of element side MI
5.6 I3 = global VB nodal number of the third node of element side MI
5.7 I4 = global VB nodal number of the fourth node of element side MI
5.8 I1AD = increment of I1 for each of the NSEQ subsequent VB element sides
5.9 I2AD = increment of I2 for each of the NSEQ subsequent VB element sides
5.10 I3AD = increment of I3 for each of the NSEQ subsequent VB element sides
5.11 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 data set.
DATA SET 39: CAUCHY BOUNDARY CONDITIONS FOR HEAT TRANSFER SIMULATIONS

This data set is needed if ITEMP = 1. Five subsets of free-formatted data records are required.

(1). Subset 1: Control Parameters
   1.1 NCEST = number of Cauchy boundary element sides
   1.2 NCNPT = number of Cauchy boundary nodal points
   1.3 NCPRT = number of Cauchy-flux profiles
   1.4 NCDPT = number of temperature data points in each of the NCPRT Cauchy-flux profiles

Subdata sets (2) through (5) are needed if and only if NCEST > 0.

(2). Subset 2: Cauchy Flux Profile: For each record, NCDPT of the data pair (TTCBF(J,I), TCBF(J,I)) are required. If this record can be fitted in a line, we will need NCPRT lines.
   2.1 TTCBF(J,I) = time of J-th data in I-th data point in the I-th profile, (T)
   2.2 TCBF(J,I) = temperature of J-th data point in I-th profile, (K)

(3). Subset 3: Type of Cauchy Flux Profile Assigned to Each Cauchy Boundary Sides
   3.1 MI = Compressed Cauchy element side number of the first side in a sequence
   3.2 NSEQ = NSEQ subsequent sides will be generated automatically
   3.3 MIAD = increment of MI for each of the NSEQ subsequent sides
   3.4 MITYP = Type of Cauchy flux profile assigned to side MI
   3.5 MTYPAD = increment of MITYP for each of the NSEQ subsequent sides

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

(4). Subset 4: Global Nodal Number of NCNPT Cauchy Boundary Condition Nodes: Typically, NCNPT lines are needed. However, automatic generation can be used.
   4.1 NI = compressed Cauchy flux node number of the first node in a sequence
   4.2 NSEQ = NSEQ subsequent nodes will be generated automatically
   4.3 NIAD = increment of NI for each of the NSEQ subsequent nodes
   4.4 NODE = global node of the compressed node NI
4.5 NODEAD= increment of NODE for each of the NSEQ subsequent nodes

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

(5). Subset 5: Specification of Cauchy Boundary Element Sides: Typically, NCEST lines 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 used.

5.1 MI = compressed Cauchy element side number of the first element side in a sequence

5.2 NSEQ = NSEQ subsequent Cauchy element sides will be generated automatically

5.3 MIAD = increment of MI for each of the NSEQ subsequent Cauchy element sides

5.4 I1 = global node number of the first node of element side MI

5.5 I2 = global node number of the second node of element side MI

5.6 I3 = global node number of the third node of element side MI

5.7 I4 = global node number of the fourth node of element side MI

5.8 I1AD = increment of I1 for each of the NSEQ subsequent Cauchy element sides

5.9 I2AD = increment of I2 for each of the NSEQ subsequent Cauchy element sides

5.10 I3AD = increment of I3 for each of the NSEQ subsequent Cauchy element sides

5.11 I4AD = increment of I4 for each of the NSEQ subsequent Cauchy element sides

**** NOTE: A line with 11 0's must be used to signal the end of this data set.

DATA SET 40: NEUMANN BOUNDARY CONDITIONS FOR HEAT TRANSFER SIMULATIONS

This data set is needed if ITEMP = 1. Five subsets of free-formatted data records are required.

(1). Subset 1: Control Parameters

1.1 NNEST = number of Neumann boundary element sides

1.2 NNNPT = number of Neumann boundary nodal points

1.3 NNPRT = number of Neumann flux profiles

1.4 NNDPT = number of temperature data points in each of the NNPRT Neumann flux profiles
Subdata sets (2) through (5) are needed if and only if $\text{NNEST} > 0$.

(2). Subset 2: Neumann Flux Profile: For each record, NNDPT of the data pair $(\text{TTNBF}(j,i), \text{TNBF}(j,i))$ are required. If this record can be fitted in a line, we will need NNPRT lines.

2.1 $\text{TTNBF}(j,i) = \text{time of } j\text{-th data in } i\text{-th data point in the } i\text{-th profile, (T)}$

2.2 $\text{TNBF}(j,i) = \text{temperature of } j\text{-th data point in } i\text{-th profile, (K)}$

(3). Subset 3: Type of Neumann Flux Profile Assigned to Neumann Boundary Sides

3.1 $\text{MI} = \text{Compressed Neumann element side number of the first side in a sequence}$

3.2 $\text{NSEQ} = \text{NSEQ subsequent sides will be generated automatically}$

3.3 $\text{MIAD} = \text{increment of MI for each of the NSEQ subsequent sides}$

3.4 $\text{MITYP} = \text{Type of Neumann flux profile assigned to side MI}$

3.5 $\text{MTYPAD} = \text{increment of MITYP for each of the NSEQ subsequent sides}$

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

(4). Subset 4: Global Nodal Number of NNNPT Neumann Boundary Condition Nodes: typically, NNNPT lines are needed. However, automatic generation can be used.

4.1 $\text{NI} = \text{compressed Neumann flux node number of the first node in a sequence}$

4.2 $\text{NSEQ} = \text{NSEQ subsequent nodes will be generated automatically}$

4.3 $\text{NIAD} = \text{increment of NI for each of the NSEQ subsequent nodes}$

4.4 $\text{NODE} = \text{global node of the compressed node NI}$

4.5 $\text{NODEAD} = \text{increment of NODE for each of the NSEQ subsequent nodes}$

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

(5). Subset 5: Specification of Neumann Boundary Element Sides: Typically, NNEST lines 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 used.

5.1 $\text{MI} = \text{compressed Neumann element side number of the first element side in a sequence}$

5.2 $\text{NSEQ} = \text{NSEQ subsequent Neumann element sides will be generated automatically}$

5.3 $\text{MIAD} = \text{increment of MI for each of the NSEQ subsequent Neumann element sides}$
5.4 I1 = global node number of the first node of element side MI
5.5 I2 = global node number of the second node of element side MI
5.6 I3 = global node number of the third node of element side MI
5.7 I4 = global node number of the fourth node of element side MI
5.8 I1AD = increment of I1 for each of the NSEQ subsequent Neumann element sides
5.9 I2AD = increment of I2 for each of the NSEQ subsequent Neumann element sides
5.10 I3AD = increment of I3 for each of the NSEQ subsequent Neumann element sides
5.11 I4AD = increment of I4 for each of the NSEQ subsequent Neumann element sides

**** NOTE: A line with 11 0's must be used to signal the end of this data set.

DATA SET 41: DIRICHLET BOUNDARY CONDITIONS FOR HEAT TRANSFER SIMULATIONS

This data set is needed if ITEMP = 1. Four subsets of free-formatted data records are required.

(1). Subset 1: Control Parameters

1.1 NDNPT = number of Dirichlet nodal points
1.2 NDPRT = number of Dirichlet temperature profiles
1.3 NDDPT = number of temperature data points in each of the NDPRT temperature profiles

Subdata sets (2) through (4) are needed if and only if NDNPT > 0.

(2). Subset 2: Dirichlet Temperature Profile: For each record, NDDPT of the data pair (TTDBF(J,I), TDBF(J,I)) are required. If this record can be fitted in a line, we will need NDPRT lines.

2.1 TTDBF(J,I) = time of J-th data in J-th data point in the I-th profile, (T)
2.2 TDBF(J,I) = temperature of J-th data point in I-th profile, (K)

(3). Subset 3: Type of Dirichlet Temperature Profile Assigned to Dirichlet Boundary Nodes.

3.1 NI = Compressed Dirichlet node number of the first node in a sequence
3.2 NSEQ = NSEQ subsequent Dirichlet nodes will be generated automatically
3.3 NIAD = increment of NI for each of the NSEQ subsequent nodes
3.4 NITYP = Type of Dirichlet temperature profile assigned to node NI

3.5 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 data set.

(4). Subset 4: Dirichlet Nodes

4.1 NI = compressed Dirichlet node number of the first node in a sequence

4.2 NSEQ = NSEQ subsequent nodes will be generated automatically

4.3 NIAD = increment of NI for each of the NSEQ subsequent nodes

4.4 NODE = global node of the compressed node NI

4.5 NODEAD= increment of NODE for each of the NSEQ subsequent nodes

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