FEMWATER

A Three-Dimensional Finite Element Computer Model for Simulating Density-Dependent Flow and Transport in Variably Saturated Media

Version 3.0

by Hsin-Chi J. Lin, David R. Richards and Cary A. Talbot
U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Gour-Tsyh (George) Yeh
Department of Civil and Environmental Engineering
University of Central Florida
Orlando, FL 32816-2450

Jing-Ru (Ruth) Cheng
Department of Computer Science
Pennsylvania State University
University Park, PA 16802

Hwai-Ping (Pearce) Cheng
Department of Nuclear Science
National Tsing Hua University
Hsinchu 30013, Taiwan

Norman L. Jones
Environmental Modeling Research Laboratory
Department of Civil and Environmental Engineering
Brigham Young University
Provo, UT 84602

Contents

List of Figures

List of Tables

1 Introduction

Purpose

The purpose of this report is to provide a complete set of documentation for the FEMWATER groundwater model. The intended users of the manual will have a wide range of technical experience and have widely different needs that the model and documentation will fulfill. While it is impossible to address all needs in the body of one document, this report has been written to provide useful information for all users. Sophisticated users will find descriptions of the numerical techniques and a complete set of governing equations that form the theoretical basis of the model. The casual user will find examples of several types of problems that it is hoped will include their type of application. With little effort the casual modeler will be able to follow the examples provided and spend little time with problem setup.

FEMWATER Origins

In the early 1990's, the Athens Laboratory of the U.S. Environmental Protection Agency (AERL) and the U.S. Army Engineer Waterways Experiment Station (WES) conducted independent evaluations of a wide variety of groundwater models to determine which existing groundwater models could be adopted for use in their in-house applications. AERL was interested in adopting a three-dimensional (3-D) variably saturated model for wellhead protection use that could model irregular geometries. WES was interested in the same capabilities but for the purposes of conducting groundwater remediation studies at contaminated Department of Defense sites and for salinity intrusion applications in U.S. Army Corps of Engineers navigation projects. The independent evaluations by both agencies resulted in the selection of the 3DFEMWATER (Yeh 1987b) and 3DLEWASTE (Yeh 1990) models for future development and implementation within their agencies. Once it became known to the agencies that they had similar interests and research responsibilities, a cooperative research agreement was reached and work began on a single groundwater modeling system to support both agencies. FEMWATER is the name of the developed model.

FEMWATER is a modern implementation of the two older models, 3DFEMWATER (flow) and 3DLEWASTE (transport). FEMWATER was formed by combining the two codes into a single coupled flow and transport model. The 3DFEMWATER and 3DLEWASTE models were originally written by Dr. Gour-Tsyh (George) Yeh at Pennsylvania State University while FEMWATER was written as a collaborative effort between Dr. Yeh and Dr. Hsin-Chi (Jerry) Lin at WES.

The improvements implemented in FEMWATER are numerous. First, the entire program structure was changed to allow its integration into the Department of Defense Groundwater Modeling System (GMS). The GMS contains a state-of-the-art graphical user environment that allows efficient model setup and visualization (Engineering Computer Graphics Laboratory (ECGL) 1996). This was a particularly onerous task in the older implementation of the model since it suffered from the common limitations of older FORTRAN codes. Second, a series of new solvers were added to replace the previously used block iterative solver. The new solvers allow an arbitrary node numbering scheme that allows easier graphical user interface connections and still enjoy improved computational efficiency. Third, density-driven (salinity) transport capability was added to allow salinity intrusion studies in coastal aquifers. This required the coupling of flow and transport within a common model. Previous versions separated the flow and transport calculations.

Improvements in Version 3.0 of FEMWATER over previous versions include the addition of a dynamic relaxation parameter, dynamic time-step size reduction in cases of difficult convergence, a new output file giving fluid flux at mesh nodes, revisions to the preconditioned-conjugate gradient solver packages previously added to FEMWATER, the option to use B-spline functions in obtaining values from the user-input soil moisture curves and the ability to specify effective porosity separately from total porosity for use in transport calculations.

Formulation of FEMWATER

FEMWATER is designed to solve the following system of governing equations along with initial and boundary conditions, which describe flow and transport through saturated-unsaturated porous media. The governing equations for flow are basically the modified Richards equation, which is derived in Appendix A. The equation is as follows:

Governing equations for flow

$$\frac{\rho}{\rho_o} F \frac{\partial h}{\partial t} = \nabla \cdot \left[\mathbf{K} \cdot \left(\nabla h + \frac{\rho}{\rho_o} \nabla z \right) \right] + \frac{\rho^*}{\rho_o} q \qquad (1)$$

$$F = \alpha' \frac{\theta}{n} + \beta'\theta + n\frac{dS}{dh} \tag{2}$$

where

F = storage coefficient

h = pressure head

t = time

K = hydraulic conductivity tensor

z = potential head

q = source and/or sink

ρ = water density at chemical concentration C

ρ_o = referenced water density at zero chemical concentration

ρ^* = density of either the injection fluid or the withdrawn water

θ = moisture content

α' = modified compressibility of the medium

β' = modified compressibility of the water

n = porosity of the medium

S = saturation

The hydraulic conductivity **K** is given by

$$\mathbf{K} = \frac{\rho g}{\mu}\mathbf{k} = \frac{(\rho/\rho_o)}{(\mu/\mu_o)}\frac{\rho_o g}{\mu_o}\mathbf{k_s}k_r = \frac{\rho/\rho_o}{\mu/\mu_o}\mathbf{K_{so}}k_r \tag{3}$$

where

μ = dynamic viscosity of water at chemical concentration C

μ_o = referenced dynamic viscosity at zero chemical concentration

k = permeability tensor

$\mathbf{k}_s$ = saturated permeability tensor

$\mathbf{k}_r$ = relative permeability or relative hydraulic conductivity

$\mathbf{K}_{so}$ = referenced saturated hydraulic conductivity tensor

The referenced value is usually taken at zero chemical concentration. The density and dynamic viscosity of water are functions of chemical concentration and are assumed to take the following form

$$\frac{\rho}{\rho_o} = a_1 + a_2 C + a_3 C^2 + a_4 C^3 \tag{4}$$

and

$$\frac{\mu}{\mu_o} = a_5 + a_6 C + a_7 C^2 + a_8 C^3 \tag{5}$$

where a_1, a_2, ..., a_8 are the parameters used to define concentration dependence of water density and viscosity and C is the chemical concentration.

The Darcy velocity is calculated as follows

$$\mathbf{V} = -\mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) \tag{6}$$

Initial conditions for flow equation. The initial conditions for the flow equation are given by Equation (7):

$$h = h_i(x, y, z) \quad \text{in}\, R, \tag{7}$$

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 Equation (1).

Boundary conditions for flow equation. The boundary conditions for the flow equation are given in the following equations.

a. Dirichlet conditions:

$$h = h_d(x_b, y_b, z_b, t) \quad \text{on } B_d, \tag{8}$$

b. Gradient flux conditions:

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h\right) = q_n(x_b, y_b, z_b, t) \quad \text{on } B_n, \tag{9}$$

c. Flux conditions:

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h + \nabla z\right) = q_c(x_b, y_b, z_b, t) \quad \text{on } B_c, \tag{10}$$

d. Variable conditions during precipitation period:

$$h = h_p(x_b, y_b, z_b, t) \quad \text{on } B_v, \tag{11}$$

or

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h + \nabla z\right) = q_p(x_b, y_b, z_b, t) \quad \text{on } B_v, \tag{12}$$

e. Variable conditions during nonprecipitation period:

$$h = h_p(x_b, y_b, z_b, t) \quad \text{on } B_v, \tag{13}$$

or

$$h = h_m(x_b, y_b, z_b, t) \quad \text{on } B_v, \tag{14}$$

or

$$-\mathbf{n} \cdot \mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) = q_e(x_b, y_b, z_b, t) \quad \text{on } B_v, \tag{15}$$

where

(x_b,y_b,z_b) = spatial coordinate on the boundary

$\mathbf{n}$ = outward unit vector normal to the boundary

h_d = Dirichlet functional value

q_n = Gradient flux value

q_c = Flux value

B_d = Dirichlet boundary

B_n = Gradient flux boundary

B_c = Flux boundary

B_v = variable boundary

h_p = ponding depth

q_p = throughfall of precipitation on the variable boundary

h_m = minimum pressure on the variable boundary

q_e = evaporation rate on the variable boundary

Only one of Equations (11)-(15) is used at any point on the variable boundary at any time.

Governing equations for transport

The governing equations for transport are derived based on the continuity of mass and flux laws as given in Appendix A. The major processes are advection, dispersion/diffusion, adsorption, decay, biodegradation, and source/sink.

$$\begin{aligned}&\theta\frac{\partial C}{\partial t}+\rho_b\frac{\partial S}{\partial t}+\mathbf{V}\cdot\nabla C-\nabla\cdot\left(\theta\mathbf{D}\cdot\nabla C\right)=\\&-\left(\alpha'\frac{\partial h}{\partial t}+\lambda\right)\left(\theta C+\rho_b S\right)-\left(\theta K_w C+\rho_b K_s S\right)+\\&m-\frac{\rho^*}{\rho}qC+\left(F\frac{\partial h}{\partial t}+\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)-\frac{\partial\theta}{\partial t}\right)C\end{aligned}\tag{16}$$

$$S = K_d C \quad \text{for linear isotherm} \tag{17}$$

$$S = \frac{S_{max}KC}{1+KC} \quad \text{for Langmuir isotherm} \tag{18}$$

$$S = KC^n \quad \text{for Freundlich isotherm} \tag{19}$$

where

θ = moisture concentration

ρ_b = bulk density of the medium (M/L^3)

C = material concentration in aqueous phase (M/L^3)

S = material concentration in adsorbed phase (M/M)

t = time

V = discharge

∇ = del operator

D = dispersion coefficient tensor

α' = compressibility of the medium

h = pressure head

λ = decay constant

m = q C_{in} = artificial mass rate

q = source rate of water

C_{in} = material concentration in the source

K_w = first order biodegradation rate constant through dissolved phase

K_s = first order biodegradation rate through adsorbed phase

F = storage coefficient

K_d = distribution coefficient

S_{max} = maximum concentration of medium in the Langmuir nonlinear isotherm

n = power index in the Freundlich nonlinear isotherm

K = coefficient in the Langmuir or Freundlich nonlinear isotherm.

The dispersion coefficient tensor **D** in Equation (16) is given by

$$\theta \mathbf{D} = a_T |\mathbf{V}| \delta + (a_L - a_T) \frac{\mathbf{VV}}{|\mathbf{V}|} + a_m \theta \tau \delta \qquad (20)$$

where

$|\mathbf{V}|$ = magnitude of **V**

δ = Kronecker delta tensor

a_T = lateral dispersivity

a_L = longitudinal dispersivity

a_m = molecular diffusion coefficient

τ = tortuosity

Initial conditions for transport equation. The initial conditions for the transport equation are given by Equation (16):

$$C = C_i(x, y, z) \quad \text{in}\, R \qquad (21)$$

where R is the region of interest and C_i is the prescribed initial condition, which can be obtained by either field measurements.

Boundary conditions for transport equation. The boundary conditions for the transport equation are given in the following equations.

a. Dirichlet conditions:

$$C = C_d(x_b, y_b, z_b) \quad \text{on } B_d \tag{22}$$

b. Variable conditions:

$$\mathbf{n} \cdot \left(\mathbf{V}C - \theta \mathbf{D} \cdot \nabla C \right) = \mathbf{n} \cdot \mathbf{V} C_v \left(x_b, y_b, z_b, t \right) \quad \text{if } \mathbf{n} \cdot \mathbf{V} \leq 0 \tag{23}$$

$$\mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C \right) = 0 \quad \text{if } \mathbf{n} \cdot \mathbf{V} > 0 \tag{24}$$

c. Flux conditions:

$$\mathbf{n} \cdot \left(\mathbf{V}C - \theta \mathbf{D} \cdot \nabla C \right) = q_c \left(x_b, y_b, z_b, t \right) \quad \text{on } B_c \tag{25}$$

d. Gradient flux conditions:

$$\mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C \right) = q_n \left(x_b, y_b, z_b, t \right) \quad \text{on } B_n \tag{26}$$

where

(x_b,y_b,z_b) = spatial coordinate on the boundary

n = outward unit vector normal to the boundary

C_d = concentration on the Dirichlet boundary

C_v = concentration of water through the variable boundary

B_d = Dirichlet boundary

B_v = variable boundary

q_c = total flux through the boundary B_c

q_n = total gradient flux through the boundaries B_n

Since the hybrid Lagrangian-Eulerian approach is used to simulate Equation (16), it is written in the Lagrangian-Eulerian form as

$$\begin{aligned}\left(\theta+\rho_b K_d\right)\frac{D_{v_d}C}{Dt} &= \nabla\cdot\left(\theta\mathbf{D}\cdot\nabla C\right)-\left(\alpha'\frac{\partial h}{\partial t}+\lambda\right)\left(\theta C+\rho_b S\right)- \\ &\quad \left(\theta K_w C+\rho_b K_s S\right)+m-\frac{\rho^*}{\rho}qC+ \\ &\quad \left(F\frac{\partial h}{\partial t}+\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)-\frac{\partial\theta}{\partial t}\right)C\end{aligned} \tag{27}$$

$$\mathbf{V}_d=\frac{\mathbf{V}}{\theta+\rho_b K_d}\ \text{ for linear isotherm model} \tag{28}$$

$$\begin{aligned}\theta\frac{D_{v_f}C}{Dt}+\rho_b\frac{dS}{dC}\frac{\partial C}{\partial t} &= \nabla\cdot\left(\theta\mathbf{D}\cdot\nabla C\right)- \\ &\quad \left(\alpha'\frac{\partial h}{\partial t}+\lambda\right)\left(\theta C+\rho_b S\right)-\left(\theta K_w C+\rho_b K_s S\right)+ \\ &\quad m-\frac{\rho^*}{\rho}qC+\left(F\frac{\partial h}{\partial t}+\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)-\frac{\partial\theta}{\partial t}\right)C\end{aligned} \tag{29}$$

$$\mathbf{V}_f=\frac{\mathbf{V}}{\theta}\quad \text{for Freundlich and Langmuir models} \tag{30}$$

where $\mathbf{V}_d$ and $\mathbf{V}_f$ are the retarded and fluid pore velocities, respectively; and $D_{Vd}()/Dt$ and $D_{Vf}()/Dt$ denote the material derivative of ( ) with respect to time using the retarded and fluid pore velocities, respectively.

The flow equation, Equation (1), subject to initial and boundary conditions, Equations (8)-(15), is solved with the Galerkin finite element method. The transport equations, Equations (27) and (28) or (29) and (30), subject to initial and boundary conditions, Equations (21)-(26), are solved with the hybrid Lagrangian-Eulerian finite element methods. Detailed implementation of the numerical approximation of flow and transport problems is given in Appendix B.

2 Running FEMWATER

File Organization

FEMWATER was designed to be operated in batch mode. The input for FEMWATER is organized into a set of input files. The output from FEMWATER is a combination of screen and file output. A summary of the input and output files is shown in Table 1 and Table 2.

Super File

When FEMWATER is launched, the user is prompted for the name of a single input file. This file is called the “super file” and contains a list of all of the appropriate input and output files used in a particular simulation. Grouping the file names in a super file simplifies file management and eliminates the need to type the names of all of the files each time a simulation is performed. The format of the super file is shown in Figure 1.

```
FEMSUP              /* File type identifier */
GEOM  filename      /* Geometry file */
BCFT  filename      /* Model file */
PRTF  filename      /* Printed output file */
ICHD  filename      /* Pressure head initial condition file */
ICMC  filename      /* Moisture content (nodal)  initial condition file */
ICVL  filename      /* Velocity initial condition file */
ICCN  filename      /* Concentration initial condition file */
FLVL  filename      /* Velocity flow file (for transient only simulations) */
FLPH  filename      /* Pressure head file (for transient only simulations) */
PSOL  filename      /* Pressure head solution file */
MSOL  filename      /* Moisture content (nodal) solution file */
VSOL  filename      /* Velocity solution file */
CSOL  filename      /* Concentration solution file */
BFXF  filename      /* Nodal boundary fluid flux solution file */
```

Figure 1. Super file format

Table 1
Input Files

File Name	Description
Super File	Text file containing a list of all of the input and output files used in a FEMWATER simulation.
Geometry File	Text file containing the data describing the finite element mesh, i.e., nodal coordinates and element topology.
Model File	Text containing analysis parameters and options, material properties , boundary conditions, and initial condition options.
Initial Condition Files	Text or binary files containing concentration, pressure head, velocity, moisture content initial conditions.
Flow Files	Text or binary files containing a previously computed flow solution (pressure head and velocity) which are used to define a 3-D flow field for a transport only simulation.

Table 2
Output Files

File Name	Description
Printed Output	Text file containing a summary of the output.
Pressure Head	Text or binary file containing the computed pressure heads. Used for post-processing or as initial conditions for a subsequent analysis.
Moisture Content	Text or binary file containing the computed moisture content at nodes. Used for post-processing.
Velocity	Text or binary file containing the computed Darcian velocities. Used for post-processing.
Concentration	Text or binary file containing the computed concentrations. Used for post-processing or as initial conditions for a subsequent analysis.
Nodal Boundary Fluid Flux	Text or binary file containing the computed nodal fluid flux passing through each node of the mesh. Internal nodes, unless assigned a point source or sink, will all be zero. Nodes with boundary conditions applied can have non-zero fluid flux rates.
Nodal Boundary Concentration Flux	Text or binary file containing the computed nodal concentration flux passing through each node of the mesh. Internal nodes, unless assigned a point source or sink, will all be zero. Nodes with boundary conditions applied can have non-zero fluid flux rates.

The first record in the file is the file type identifier. Each of the subsequent records represents an input or an output file. The first field in each record is a four-character string identifying the type of the file listed in the record. The second field in each record is the name of the corresponding file. The files should be in the same directory as the super file.

Not all of the files shown in Figure 1 are required for every simulation. Some of the initial condition files are not required depending on the initial condition options specified in the model file. Also, the user can also specify in the model file not to output some of the solution files.

Card Style Format

The records used in the super file format shown in Figure 1 are representative of the formatting style used for all of the FEMWATER input files. This format is often referred to as the “card style” format. With this format, the components of the file are grouped into logical groups called “cards.” Typically, each card is a single line or record; however, some cards extend to multiple lines. The first component of each card is a short name that serves as the card identifier. The remaining fields on the line contain the information associated with the card. In some cases, such as lists, a card can use multiple lines. All of the cards are assumed to be free-format.

Several advantages are associated with the card type approach to formatting files:

a. Card identifiers make the file easier to read. Each input line has a label, which helps to identify the data on the line.

b. The card names are useful as text strings for searching in a large file. All input lines of a particular type can be located quickly in a large input file.

c. Cards allow the data to be input in any order in many cases; i.e., the order that the cards appear in the file is usually not important.

d. Cards make it easy to modify a file format. New data can be included simply by defining a new card type. If the new card is optional (which is typically the case for new cards) old files are still compatible. If an old card type is no longer used, the card can simply be ignored without causing input errors.

Other Files

Each of the files listed in the super file are described in more detail in subsequent chapters. The geometry file is described in Chapter 3, the contents of the model file are described in Chapters 3, 4, 5, and 6, the initial condition files are described in Chapter 7, and the solution files are described in Appendix C.

3 Meshes

Introduction

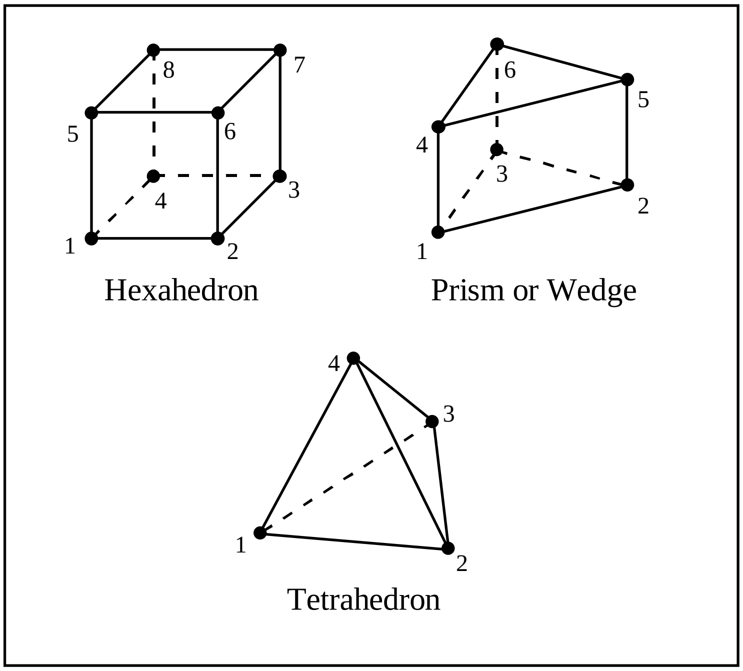
The computational discretization utilized by FEMWATER is a three-dimensional finite element mesh. The volumetric domain to be modeled by FEMWATER must be idealized and discretized into hexahedra, prisms, and/or tetrahedra. Elements are typically grouped into zones representing different stratigraphic units. Each element is assigned a material ID representing the zone to which the elements belongs. When constructing a mesh, care should be taken to ensure that elements do not cross or straddle stratigraphic boundaries.

Elements Supported

The types of elements supported by FEMWATER are shown in Figure 2. Each of the elements are linear; quadratic elements are not supported. Although all three element types are supported, tetrahedra do not perform as well as the other types and should be avoided if possible. The numbering sequence shown in Figure 2 should be used when describing the elements in the geometry file.

Geometry File Format

The coordinates of the mesh nodes and the element topology are input to FEMWATER through the geometry file. The format of the geometry file is shown in Figure 3.



Hexahedron

Prism or Wedge

Tetrahedron

Figure 2. The element types supported by FEMWATER

```
3DFEMGEO                                    /* File type identifier */
T1 text                                     /* Title, line 1 */
T2 text                                     /* Title, line 2 */
T3 text                                     /* Title, line 3 */
GN id x y z                                 /* Nodal coordinates */
GE8 id n1 n2 n3 n4 n5 n6 n7 n8 matid        /* Hex. Element */
GE6 id n1 n2 n3 n4 n5 n6 matid              /* Prism element */
GE4 id n1 n2 n3 n4 matid                    /* Tetrahedral element */
END                                         /* End of input data */
```

Figure 3. Geometry file format

The cards used in the geometry file are as follows:

Card Type	**3DFEMGEO**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**T1, T2, T3**
Description	Three title lines. Any text can be entered on these lines. The text is used as a banner to the printed output.
Required	NO
Format	T1 text T2 text T3 text
Sample	T1 Salinity intrusion problem T2 ACR Environmental Services Inc. T3 June 19, 1995

Card Type	**GE8**		
Description	Hexahedral element. One entry for each hexahedral element in the mesh. The ordering of the nodes should correspond to the diagram in Figure 2.		
Required	YES		
Format	GE8 id n1 n2 n3 n4 n5 n6 n7 n8 matid		
Sample	GE8 10 847 938 943 928 380 942 835 655 1		
Field	*Variable*	*Value*	*Description*
id	NEL	+	The ID of the node.
N1	IE(NEL,1)	+	The index of node number 1.
N2	IE(NEL,2)	+	The index of node number 2.
N3	IE(NEL,3)	+	The index of node number 3.
N4	IE(NEL,4)	+	The index of node number 4.
N5	IE(NEL,5)	+	The index of node number 5.
N6	IE(NEL,6)	+	The index of node number 6.
N7	IE(NEL,7)	+	The index of node number 7.
N8	IE(NEL,8)	+	The index of node number 8.
Matid	IE(NEL,9)	+	The element material index.

Card Type	**GE6**		
Description	Wedge or prism element. One entry for each wedge element in the mesh. The ordering of the nodes should correspond to the diagram in Figure 2.		
Required	YES		
Format	GE6 id n1 n2 n3 n4 n5 n6 matid		
Sample	GE6 10 847 938 943 928 380 942 1		
Field	*Variable*	*Value*	*Description*
id	NEL	+	The ID of the node.
N1	IE(NEL,1)	+	The index of node number 1.
N2	IE(NEL,2)	+	The index of node number 2.
N3	IE(NEL,3)	+	The index of node number 3.
N4	IE(NEL,4)	+	The index of node number 4.
N5	IE(NEL,5)	+	The index of node number 5.
N6	IE(NEL,6)	+	The index of node number 6.
Matid	IE(NEL,9)	+	The element material index.

Card Type	**GE4**		
Description	Tetrahedral element. One entry for each tetrahedral element in the mesh. The ordering of the nodes should correspond to the diagram in Figure 2.		
Required	YES		
Format	GE4 id n1 n2 n3 n4 matid		
Sample	GE4 10 847 938 943 928 1		
Field	*Variable*	*Value*	*Description*
id	NEL	+	The ID of the node.
N1	IE(NEL,1)	+	The index of node number 1.
N2	IE(NEL,2)	+	The index of node number 2.
N3	IE(NEL,3)	+	The index of node number 3.
N4	IE(NEL,4)	+	The index of node number 4.
Matid	IE(NEL,9)	+	The element material index.

Card Type	**GN**		
Description	Mesh node. One entry for each node in the mesh.		
Required	YES		
Format	GN id x y z		
Sample	GN 83 3482.4 4389.3 34.6		
Field	*Variable*	*Value*	*Description*
id	NNP	+	The ID of the node.
X	X(NNP,1)	±	The x coordinate of the node [L].
y	X(NNP,2)	±	The y coordinate of the node [L].
z	X(NNP,3)	±	The z coordinate of the node [L].

Card Type	**END**
Description	End of input data marker.
Required	YES

Mesh Generation Guidelines

As with any finite element problem, special care should be used in the construction of the mesh to avoid numerical instability. Numerical accuracy and stability are often competing interests. Procedures that assist in achieving model stability often do so to the detriment of accuracy. When the need for computational efficiency is added to accuracy and stability concerns, mesh generation can be a difficult task. Indeed, many consider it an art form although numerical efforts are under way to minimize the artistic requirements.

Prior to mesh generation, the hydrogeologic conceptual model should be well defined. The stratigraphy and associated variations in hydraulic conductivity should be well understood. Boundary conditions, whether they are surface recharge, salinity concentrations, or wells, should be known in detail. The locations of remediation structures or other mitigative treatments should be mapped. Once all of this information is known, it is possible to start the mesh generation process. Guidelines are now presented to assist in the process.

The computational domain required for FEMWATER is an unstructured, three-dimensional finite element domain. The recommended method for construction of such a numerical domain requires first that a two-dimensional projection mesh be constructed. This projection mesh will be extruded into layers of three-dimensional finite elements, forming the computational domain for FEMWATER. The two-dimensional mesh will determine the horizontal position of all nodes in the three-dimensional mesh while the vertical position of these nodes will be determined by the stratigraphic conceptual model.

The first step in setting up a two-dimensional projection mesh is to take the conceptual model generated in the subsurface conceptualization phase and determine where the external boundary conditions will be located. These will essentially coincide with locations where water table information around the area of interest is available. The external edges of the model should be located where this information is well known. The next step is to determine the number of elements and their best distribution to solve the flow and transport problem. Simply put, fine mesh spacing should be located where head or concentration gradients are expected to be maximum. This will certainly be near wells that have caused significant cones of depression. However, care must be given to gradually vary the size of the elements to avoid numerical errors. For example, the 50 percent rule should be followed whenever possible: the size of an element should not differ from the size of an adjacent element by more than 50 percent. Additionally, every effort should be made to avoid highly skewed or irregularly shaped elements.

Once a projection mesh has been developed, interest should shift to the proper vertical element spacing. Issues with regard to element spacing are applicable to the vertical as well as horizontal dimensions. Fine resolution should be placed in vertical regions where head or concentration gradients are greatest and most particularly in the unsaturated zone. For example, if a highly conductive aquifer is adjacent to a highly impermeable aquiclude, fine mesh resolution is required in the vicinity of the interface. In general, there should be a minimum of three layers of elements vertically for each distinct stratigraphic unit particularly if large variations of hydraulic conductivity occur in adjacent layers. As an aside, if large variations in hydraulic conductivity are required, no two adjacent layers should vary by more than three orders of magnitude. If this rule is violated, the solutions will likely be inaccurate and probably slow to converge. If there are indeed sharp variations in conductivity, then many layers should be used and the values varied gently over the short distance in which they change.

When setting up a mesh for a transport analysis, all of the previous issues are germane as well as one other. First, if hexahedral elements are used, each element should be constructed such that all of the element faces are planar. This is a particular requirement because the particle tracking algorithm used by FEMWATER may break down if a particle crosses an element face that is not planar. Triangular faces are always planar, but quadrilateral faces may or may not be planar.

On the subject of computational efficiency, it is important to note that the smallest number of elements does not always provide the fastest simulation. It is quite possible to construct a model with insufficient numbers of elements to characterize the problem adequately, thereby creating a simulation that is slow to converge. In short, fewer elements will be used in the calculation, but greater numbers of iterations will be required. In general, if sufficient numbers of elements are used, fewer iterations will be required to converge on a solution. It is better to use large numbers of elements for few iterations to get accurate answers than to use few elements for many iterations to get inaccurate and possibly divergent answers.

4 Analysis Options

Introduction

One of the primary FEMWATER input files is the model file, which consists of analysis options, material properties, boundary, and initial conditions. The first of these groups, the analysis options, are described in this Chapter. The material properties are described in Chapter 5, the boundary conditions are described in Chapter 6, and the initial conditions are described in Chapter 7.

File Format

The set of cards in the model file corresponding to analysis options is shown in Figure 4. The file type identifier and the title cards are similar to the corresponding cards found in the geometry file and described in Chapter 0. Each of the remaining cards is described in more detail in the following sections.

Run Option Parameters

The run option parameters designated on cards OP1, OP2, OP3, OP4, and OP5 include options for specifying the type of simulation, the solver, relaxation parameters, and sorption options.

Type of simulation (OP1)

The OP1 card is used to specify the type of simulation to be performed by FEMWATER. The parameter KMOD indicates the type of simulation to be conducted.

```
3DFEMWBC                                       /* File type identifier */
T1 text                                        /* Title, line 1 */
T2 text                                        /* Title, line 2 */
T3 text                                        /* Title, line 3 */.
OP1 kmod                                       /* Simulation type */
OP2 kssf ksst ilump imid ipntsf iquar          /* Solver options */
OP3 wf omef omif omemin omemax omeadd omered   /* Weight factor and relaxation parameters */
OP4 ksorp                                      /* Sorption */
OP5 gg                                         /* Upper bound for eigen value */
IP1 niterf ncycle npiterf tolaf tolbf ibspl    /* Iteration opt.s, flow, b-spline option flag */
IP2 nitert npitert tolbt                       /* Iteration opt.s, trans. */
IP3 nitfit omeftt epss epst                    /* Iteration opt.s, coupled */
PT1 nxw nyw nzw idetq                          /* Particle tracking opts */
TC1 tmax                                       /* Total simulation time */
TC2 idt delt idtxy                             /* Time-step definition */
OC1 ibug ichng jopt kprt/nprint                /* Print options */
OC2 nselt kpro()                               /* Print options */
OC3 ifile kopt kdsk/npost                      /* Format and interval */
OC4 kselt ksave()                              /* Solution file opts */
```

Figure 4. The analysis option cards in the model file

Card Type	**OP1**		
Description	Type of simulation.		
Required	YES		
Format	OP1 kmod		
Sample	OP1 10		
Field	*Variable*	*Value*	*Description*
1	KMOD	10 1 11	Flow simulation only. Transport simulation only. Coupled flow and transport.

The following options are available:

a. Perform a flow simulation only (KMOD=10).

b. Perform a transport simulation only(KMOD=1). For this case, a steady-state or transient flow simulation must be performed prior to the transport simulation. The resulting pressure heads of this simulation are then input to FEMWATER as flow variables. While only pressure heads are required as input, previous versions of FEMWATER required that the velocity field also be input for a transport only simulation. To ensure backward compatibility, the velocity field may also be input as a flow variable but it is not required. The steps involved in setting up the proper initial conditions are described in Chapter 7.

c. Perform a coupled flow and transport simulation (KMOD=11). With a coupled flow and transport simulation, the user has the option of simulating either density-dependent flow or density-independent flow. This option is controlled by entering the appropriate parameters defining the relationship between concentration and density and concentration and viscosity. These parameters are entered on the MP4 card described in the section “Concentration Dependence Coefficients (MP4)” in Chapter 5.

(1) Density-dependent flow. In this case, the flow and transport simulations are performed simultaneously. The concentration of the solute changes the density of the solution, thus changing the flow solution. This option should be chosen only for density dependent flow problems such as salinity intrusion in coastal aquifers. The proper setting for this case is shown in sample problem 5 of Chapter 8.

(2) Density-independent flow. In this case, the flow and transport simulation are performed sequentially for every time-step. The proper setting for this case is shown in sample problem 3 of Chapter 8.

Solver options (OP2)

The OP2 card is used to select the type of time mode (steady-state or transient) to be used and to set various solver options.

Card Type	**OP2**		
Description	Solver options.		
Required	YES		
Format	OP2 kssf ksst ilump imid ipntsf iquar		
Sample	OP2 1 1 1 0 1 1		
Field	*Variable*	*Value*	*Description*
1	KSSF	0 1	Steady-state flow simulation. Transient flow simulation.
2	KSST	0 1	Steady-state transport simulation. Transient transport simulation. (Note: KSSF and KSST must be set to the same value)
3	ILUMP	0 1	No mass lumping. Mass lumping.
4	IMID	0 1	No mid-difference. Mid-difference.
5	IPNTSF	1 2 3	The pointwise iterative matrix solver. P.C.G. method (polynomial). P.C.G. method (incomplete Cholesky).
6	IQUAR	11 12 21 22	Nodal/nodal quadrature. Nodal/ gaussian quadrature. Gaussian/nodal quadrature. Gaussian/gaussian quadrature.

Steady-state versus transient. FEMWATER can be run in either a steady-state or transient mode. The steady state mode is allowed only when the flow-simulation-only option has been selected with the OP1 card. FEMWATER uses the Lagrangian-Eulerian finite element method to solve the transport equation. Therefore, the steady-state mode of transport simulation is not allowed in this option. The transient mode must be used when a transport simulation is being performed.

Mass lumping (ILUMP). This parameter indicates whether or not the mass matrix is to be lumped. With lumping (ILUMP=1), the solution is less accurate but potentially more stable. For saturated-unsaturated flow computations, or if negative concentrations or oscillating solutions occur, this parameter should be set to 1. If the computations are quite stable, particularly in largely saturated flow simulations, the parameter should be set to zero.

Mid-difference (IMID). This parameter indicates if the mid-difference method should be used in both the flow and transport computations. If IMID =1, the mid-difference method is used. Setting IMID=1 is reserved for research purposes so IMID=0 is the preferred setting.

Solver Selection (IPNTSF). The following three solvers are provided in FEMWATER:

a. Pointwise iterative matrix solver. The pointwise iterative matrix solver employs the basic successive iterative method to solve the matrix equation, including the Gauss-Seidel method, successive underrelaxation, and successive overrelaxation. When the resulting matrix is diagonally dominant, the pointwise iterative solver provides a convergent solution. This solver is preferred because it is more robust than the other two solvers. However, when the speed of convergence is too slow, one may wish to choose one of the other two solvers.

b. Preconditioned conjugate gradient method (polynomial). This solver employs the conjugate gradient method to solve the matrix equation. It uses a polynomial as a preconditioner. This matrix solver provides a convergent solution when the resulting matrix is symmetric positive definite (SPD). Theoretically, the convergence speed is faster than the pointwise iterative solver. This solver should be used only when the pointwise iterative solver is too slow.

c. Preconditioned conjugate gradient method (incomplete Choleski). This solver employs the conjugate gradient method using the incomplete Choleski decomposition as the preconditioner. A convergent solution is provided when the matrix is SPD. However, when the matrix is slightly non-symmetric, the solver could also give convergent solutions. Its speed of convergence is theoretically faster than the pointwise iterative solver and is comparable to the polynomial preconditioned conjugate gradient method. This solver should be used only when the pointwise iterative solver is too slow. This solver is generally but not always preferred over the polynomial preconditioned conjugate gradient method.

Quadrature selection (IQUAR). This parameter is an indicator of the type of quadrature used in the numerical integration. The following four quadrature options are provided:

a. Nodal/nodal quadrature. Nodal quadrature is used for surface and element integration.

b. Nodal/gaussian quadrature. Nodal quadrature is used for surface integration, and gaussian quadrature is used for element integration.

c. Gaussian/nodal quadrature. Gaussian quadrature is used for surface integration, and nodal quadrature is used for element integration.

d. Gaussian/gaussian quadrature. Gaussian quadrature is used for both surface and element integration.

Gaussian/gaussian quadrature yields the most accurate solution and should be used as the default value. However, this option may provide oscillations or divergence in highly nonlinear problems. When this occurs, the user should try to use the nodal/nodal quadrature. Once these options have been used unsuccessfully, the remaining options can be tried.

Weighting factor options (OP3)

The OP3 card is used to select the type of time derivative and relaxation weighting factors to be used and to set several parameters associated with the weighting factor.

Card Type	**OP3**		
Description	Weighting factor options.		
Required	YES		
Format	OP3 wf omef omif omemax omemin omeadd omered		
Sample	OP3 1.0 1.0 1.0 1.5 0.001 0.005 0.667		
Field	*Variable*	*Value*	*Description*
1	WF	0.5 1.0	Crank-Nicolson. Backward difference.
2	OMEF	 0.0-1.0 1.0 1.0-2.0	Iteration param. For nonlinear flow and transport. Underrelaxation. Exact relaxation. Overrelaxation.
3	OMIF	 0.0-1.0 1.0 1.0-2.0	Iteration param. For linearized flow and transport. Underrelaxation. Exact relaxation. Overrelaxation.
4	OMEMAX	1.0-2.0	Upper limit for dynamic relaxation, typically 1.5
5	OMEMIN	0.0-1.0	Lower limit for dynamic relaxation, typically 0.001
6	OMEADD	+	Amount added when relaxation parameter is dynamically increased, typically 0.005
7	OMERED	+	Reduction factor applied when relaxation parameter is dynamically reduced, typically 0.667

Weighting factor type (WF). This parameter determines how one would evaluate the time derivative terms associated with the velocity in the flow equation. Two types of weighting factors are available:

a. Crank-Nicolson central (WF=0.5). When new time derivatives are determined by averaging the previous time derivative and an estimated time derivative, the process is called Crank-Nicolson central weighting.

b. Backward difference (WF=1.0). When the time derivatives are evaluated only at the new time, the process is called backward difference weighting.

A value of WF equal to 1.0 (an implicit numerical scheme) should be used for most practical problems. Setting WF equal to 0.5 is normally done for research purposes to assess the accuracy of the Crank-Nicolson scheme.

Relaxation parameter for solving nonlinear flow and transport equations (OMEF). When the flow and transport equations are nonlinear, an estimate of the pressure head and the concentration is needed to compose the matrix equation. There are three options to estimate the pressure head and concentration based on previous guesses and newly obtained values: underrelaxation, exact relaxation, and overrelaxation. OMEF is a weighting factor that is applied to the newly obtained values, and a weighting factor of 1.0 minus OMEF is applied to the previous guesses. For underrelaxation, a value of OMEF between 0.0 and 1.0 is used for the newly obtained values. For exact relaxation OMEF is set equal to 1.0 and the newly obtained values are used as the new guesses. For overrelaxation, a value of OMEF between 1.0 and 2.0 is used for the newly obtained values.

FEMWATER now employs a dynamic relaxation parameter for non-linear flow and transport. What the user specifies in the OMEF field is the starting relaxation parameter value. The actual OMEF value will be adjusted by FEMWATER as needed to achieve convergence, as dictated by the OMEMAX, OMEMIN, OMEADD, and OMERED values.

Relaxation parameter for solving linearized flow and transport equations (OMIF). In order to solve the linearized matrix equations using the iteration method, an estimate of the solution is needed prior to taking the next iteration. There are three options to estimate the solution based on previous guesses and the newly obtained solution: underrelaxation, exact relaxation, and overrelaxation. This is accomplished with an OMIF weighting factor that is similar to the OMEF factor described in the previous section.

Normally OMIF should be set to 1.0. If the convergence history shows signs of oscillation, then a value less than 1.0 should be used. If the convergence history shows a slow but monotonic decrease, OMIF should be set to a value between 1.0 and 2.0.

Upper limit for dynamic non-linear flow and transport relaxation parameter (OMEMAX). FEMWATER will adjust the non-linear flow and transport relaxation parameter (OMEF) either upward or downward to help achieve convergence as the simulation progresses. The user specifies the upper limit of the range of relaxation parameter values with the OMEMAX card.

FEMWATER will not use relaxation parameter values above this limit. The default value for OMEMAX is 1.5. For backward compatibility, this card is not required. If omitted, the default value will be used.

Lower limit for dynamic non-linear flow and transport relaxation parameter (OMEMIN). FEMWATER will adjust the non-linear flow and transport relaxation parameter (OMEF) either upward or downward to help achieve convergence as the simulation progresses. The user specifies the lower limit of the range of relaxation parameter values with the OMEMIN card. FEMWATER will not use relaxation parameter values below this limit. The default value for OMEMIN is 0.001. For backward compatibility, this card is not required. If omitted, the default value will be used.

Dynamic non-linear flow and transport relaxation parameter increase increment (OMEADD). When FEMWATER determines that the non-linear flow and transport relaxation parameter needs to be increased to achieve faster convergence, the OMEADD card specifies the amount of increase that is added to the current relaxation parameter value. The default value for OMEADD is 0.005. For backward compatibility, this card is not required. If omitted, the default value will be used.

Dynamic non-linear flow and transport relaxation parameter decrease factor (OMERED). When FEMWATER determines that the non-linear flow and transport relaxation parameter needs to be decreased to achieve convergence, the OMERED card specifies the reduction factor that is applied to the current relaxation parameter value. This calculation is performed by the equation OMEF = OMEF * OMERED, thus this card should have a value between 0.0 and 1.0. The default value for OMERED is 0.6667. For backward compatibility, this card is not required. If omitted, the default value will be used.

Sorption options (OP4)

The OP4 card is used to designate which model will be used for the sorption isotherm. The selection of the sorption model should be dictated by experimental evidence, and it depends highly on the type of chemicals and subsurface media.

Card Type	**OP4**		
Description	Sorption options.		
Required	YES		
Format	OP4 ksorp		
Sample	OP4 1		
Field	*Variable*	*Value*	*Description*
1	KSORP	1 2 3	Linear isotherm. Freundlich isotherm. Langmuir isotherm.

The following three models are available for modeling the sorption isotherm:

a. Linear. A linear isotherm is used for the adsorption model. For salinity intrusion simulations, a linear model is sufficient.

b. Freundlich. A nonlinear isotherm (Freundlich isotherm) is used for the adsorption model.

c. Langmuir. A nonlinear isotherm (Langmuir isotherm) is used for the adsorption model.

Although the Freundlich isotherm option can be used to simulate a linear isotherm by setting the value of the exponent (n = 1), it is recommended that the linear isotherm be simulated by using only the linear isotherm option. This is because the linear isotherm option makes use of retarded seepage velocities, which result in a more accurate solution for the particle tracking scheme than the pore velocities used in conjunction with the nonlinear adsorption models. Sorption constants for the different isotherm options are entered on the MP5 card discussed in Chapter 5.

Preconditioned conjugate gradient method (OP5)

The OP5 card is used to provide an estimator for GG in the preconditioned conjugate gradient method. If the OP5 card is not present in the input, GG is not read but will be computed by the solver itself. If the OP5 card is present, GG is read and it will be the upper bound of the maximum eigenvalue of the coefficient matrix using the preconditioned conjugate gradient method. The default value is 1.0.

Card Type	**OP5**		
Description	Maximum eigenvalue for preconditioned conjugate gradient method.		
Required	NO		
Format	OP5 gg		
Sample	OP5 1.0		
Field	*Variable*	*Value*	*Description*
1	GG		Upper bound for eigenvalue.

Iteration Parameters

The cards IP1, IP2, and IP3 are used to specify the number of iterations for the flow simulation, the transport simulation, and the coupled simulation.

Flow simulation (IP1)

The IP1 card is used to specify the number of iterations for solving the flow equations, and the number of cycles used to update rainfall-seepage boundary conditions, and in determining convergence criteria in both steady-state and transient simulations. This card is required for both flow-only and coupled simulations.

Card Type	**IP1**		
Description	Iteration parameters for the flow simulation.		
Required	YES		
Format	IP1 niterf ncylf npiterf tolaf tolbf		
Sample	IP1 40 10 400 0.0001 0.0001		
Field	*Variable*	*Value*	*Description*
1	NITERF	+	Number of iterations allowed for solving the nonlinear flow equation.
2	NCYLF	+	Number of cycles permitted for iterating rainfall seepage boundary condition per time-step.
3	NPITERF	+	Number of iterations allowed for solving linearized flow equations by pointwise iterative solver.
4	TOLAF	+	Steady state convergence criterion for flow simulation [L].
5	TOLBF	+	Transient convergence criterion for flow simulation [L].
6	IBSPL	0 or 1	Flag (1=on, 0=off) indicating use of B-spline function on soil moisture curves in unsaturated zone calculations.

The following iteration parameters must be designated for the flow simulation.

a. Number of iterations for solving the nonlinear flow equation, NITERF. Normally a value of 40 is necessary for solving the nonlinear flow equations. However, if this number is exceeded, a warning message is issued.

b. Number of iterations used per time-step to check if rainfall-seepage boundary conditions are properly used and converged, NCYLF. If no rainfall-seepage boundary conditions are used, the value should be 1. When they are used, a value of 10 should be adequate. If 10 is not adequate for convergence, a warning message is issued and the user should increase the number of iterations.

c. Number of iterations allowed for solving the linearized flow equation by pointwise iterative solver, NPITERF. A value of 400 is sufficient for most problems. If this number is exceeded and the solution does not converge, the program issues a warning message.

d. Steady-state convergence criterion for the flow simulation, TOLAF. This is the absolute error allowed to determine if a steady-state solution has reached convergence for hydraulic heads. A value of 0.00001 for the maximum disturbance is usually sufficient for most problems. However, the user may wish to determine the convergence criterion as part of the calibration of model parameters.

e. Transient convergence criterion for the flow simulation, TOLBF. This is the absolute error allowed to determine if hydraulic heads have converged. A value of 0.0001 for the maximum disturbance is usually sufficient for most problems. However, the user may wish to determine the convergence criterion as part of the calibration of model parameters.

f. Use of B-spline function to obtain values for the soil moisture curves in unsaturated zone calculations. When this flag is set (1), FEMWATER will fit a B-spline through the points defining the soil moisture curves when obtaining value from the curves for the particular negative pressure head computed. This will guarantee smooth curves with no sharp breaks in slope. Convergence times can be reduced by up to 20% using this option.

Transport simulation (IP2)

The IP2 card is used to specify the number of iterations for solving the transport equation, the transport equation by pointwise solver, and convergence criteria for transient transport simulations. This card is required for both transport only and coupled simulations.

Card Type	**IP2**		
Description	Iteration parameters for the transport simulation.		
Required	YES		
Format	IP2 nitert npitert tolbt		
Sample	IP2 40 10 400 0.001		
Field	*Variable*	*Value*	*Description*
1	NITERT	+	Number of iterations allowed for solving the nonlinear transport equation.
2	NPITERT	+	Number of iterations allowed for solving linearized transport equation.
3	TOLBT	+	Transient convergence criterion for transport simulation [M/L^3].

The following iteration parameters must be designated for transport simulations.

a. Number of iterations allowed for solving the nonlinear transport equation, NITERT. Normally, a value of 40 is sufficient. If this number is exceeded and the solution does not converge, a warning message is displayed.

b. Number of iterations allowed for solving the linearized transport equation, NPITERT. Normally, a value of 400 is sufficient for solving the linearized transport equation with the successive point iterative solver. Exceeding this number will indicate a nonconvergent solution and cause a message to be displayed. When this occurs, a larger value should be used.

c. Transient convergence criterion for transport simulation, TOLBT. This is the relative error for determining if concentrations have converged during transient simulations. A value of 0.001 is recommended.

Coupled simulation (IP3)

The IP3 card is used to specify number of iterations used in solving the coupled flow and transport equations. This card is used in addition to the IP1 and IP2 cards.

Card Type	**IP3**		
Description	Iteration parameters for the coupled simulation.		
Required	YES		
Format	IP3 nitfit omeftt epss epst		
Sample	IP3 10 0.5 0.01 0.05		
Field	*Variable*	*Value*	*Description*
1	NITFIT	+	Number of iterations allowed for solving the coupled nonlinear equations for transient solutions.
2	OMEFTT	+	Iteration parameter for solving the coupled nonlinear equations for transient solutions.
3	EPSS	+	Convergence criterion for head for solving the coupled nonlinear equations for transient solutions [L].
4	EPST	+	Convergence criterion for concentration for solving the coupled nonlinear equations for transient solutions [M/L^3].

The following iteration parameters must be designated for coupled simulation.

a. Number of iterations allowed for solving the coupled nonlinear equations for transient solutions, NITFTT. Normally, a value of 10 should be sufficient. If this number is exceeded and the solution does not converge, a warning message will be issued.

b. Iteration parameter for solving the coupled nonlinear equations for transient solutions, OMEFTT. This parameter is the weighting factor used with the present and previous results for solving the coupled nonlinear equation of the transient solution. A value of 0.5 should be used for most problems.

c. Convergence criterion for head for solving the coupled nonlinear equations for transient solutions, EPSS. This is the absolute error allowed for determining if a coupled flow and transport solution for hydraulic head has converged. A value of 0.01 for the maximum disturbance should be sufficient.

d. Convergence criterion for concentration for solving the coupled nonlinear equations for transient solutions, EPST. This is the relative error allowed for determining if a coupled flow and transport solution for concentration has converged. A value of 0.05 for the maximum disturbance should be sufficient.

Particle Tracking Parameters

In order to simulate the advective transport of particles through the model domain, FEMWATER employs a particle tracking technique. Particle tracking, as its name implies, is a means of using numerical results to track fictitious individual particles across a numerical model mesh, approximating the advection of the contaminant front. In order to accurately track particles over large elements with large velocity gradients, it sometimes necessary to subdivide the individual elements into smaller subelements.

Card Type	**PT1**		
Description	Particle tracking parameters.		
Required	YES		
Format	PT1 nxw nyw nzw idetq		
Sample	PT1 1 1 1 1		
Field	*Variable*	*Value*	*Description*
1	NXW	+	The number of grids for element tracking in the x-direction.
2	NYW	+	The number of grids for element tracking in the y-direction.
3	NZW	+	The number of grids for element tracking in the z-direction.
4	IDETQ	 1 2	Index of particle tracking pattern. Average velocity is used. Single velocity of the starting point is used.

The particle tracking parameters are entered on the PT1 card. The following parameters must be specified:

a. The number of grids for element tracking in the x-direction, NXW. This parameter specifies how many subelements are needed in the x-direction. The higher the velocity variation in the x-direction, the more subelements are needed.

b. The number of grids for element tracking in the y-direction, NYW. This parameter specifies how many subelements are needed in the y-direction. The higher the velocity variation in the y-direction, the more subelements are needed.

c. The number of grids for element tracking in the z-direction, NZW. This parameter specifies how many subelements are needed in the z-direction. The higher the velocity variation in the z-direction, the more subelements are needed.

d. The particle tracking pattern, IDETQ. Two options are available:

   (1) Average velocity is used (IDETQ=1). The use of average velocity is more accurate, and it requires fewer subelements.

   (2) Single velocity of the starting point is used (IDETQ=2). This option should be used when the velocity pattern is so complicated that the use of the average velocity would fail to locate a fictitious particle. It should be used when a quick tracking is needed.

Time Control Parameters

The TC1 and TC2 cards are used to specify the total simulation time and the time-step interval.

Maximum simulation time (TC1)

This is actual length of time to be simulated. Once the simulation time reaches the maximum simulation time, the simulation will be terminated. In steady-state simulations, this parameter is ignored.

Card Type	**TC1**		
Description	Maximum simulation time.		
Required	YES		
Format	TC1 tmax		
Sample	TC1 1000		
Field	*Variable*	*Value*	*Description*
1	TMAX	+	The maximum simulation time [T].

Time-step interval (TC2)

The computational time-step can be specified either as a constant value or a series of time-step sizes. The variable time-step should be used when the boundary conditions are changing rapidly.

Card Type	**TC2**		
Description	Time-step size.		
Required	YES		
Format	TC2 idt delt/idtxy		
Sample	TC2 0 1.0		
Field	*Variable*	*Value*	*Description*
1	IDT	 0 1	Time-step type. Constant. Variable.
2	DELT IDTXY	+ +	If IDT=0, constant time-step. If IDT=1, index of xy card for variable time-step values. The x values in the series represent times at which the time-step size will change. The y values represent the time-step sizes.

The variable time-step option in FEMWATER is defined by specifying both the time-step length and the time-step values at which these lengths occur. This is specified using an XY Series card (XY1) described in the next section. An example specification of these two parameters is given in Table 3. The resulting time-steps that this specification will produce in FEMWATER are given in Table 4. Note that in a transient FEMWATER simulation a time-step value of zero represents the initial conditions specified for the simulation and is saved by default to the resulting output files. The value of the first computational time-step will be determined by the specified length. The time-step values specified in the first column of Table 3 indicate the time-steps to which the corresponding time-step lengths in the second column of Table 3 are valid. In other words, one can read the two columns of data in Table 3 as, "until time-step X, the time-step lengths will be Y", etc. until the maximum simulation time is reached.

Table 3
Time-Step Interval Specification

Time-Step Value	Time-Step Length
10	2
20	5
50	10

Table 4
Resulting Time-Step Lengths

Time-Step	Δt
2	2
4	2
6	2
8	2
10	2
15	5
20	5
30	10
40	10
50	10

The XY Series format (XY1)

The TC2 card in the previous section uses the XY Series format (XY1 card) to specify how the time-step size changes with time. The XY1 card is used by many of the FEMWATER input cards as a consistent means of designating a list of pairs of numbers. This can be thought of as the x- and y-coordinates of a curve, although the values do not necessarily have to correspond to a curve. In many cases, the x-value represents a time and the y-value represents some parameter that is changing with time. However, the series can represent any relationship. Each XY1 card has an index that is referenced by other cards. XY1 cards can appear anywhere in the input file and are often listed at the end of the file.

Card Type	**XY1**		
Description	XY Series. Used to define a sequence of pairs of numbers.		
Required	YES		
Format	XY1 i n dx dy rep begc tname x_1 y_1 x_2 y_2 . . x_n y_n		
Sample	XY1 1 5 0 0 0 0 head 4.0 0.0 4.1 2.0 4.2 7.0 4.3 8.0 4.4 9.5		
Field	*Variable*	*Value*	*Description*
1	I	+	Index of xy series.
2	N	+	The number of xy pairs in the series.

3	DX		Value is ignored by FEMWATER.
4	DY		Value is ignored by FEMWATER.
5	REP		Value is ignored by FEMWATER.
6	BEGC		Value is ignored by FEMWATER.
7	TNAME	text	A character string representing the name of the XY series.
8	X,Y	±, ±	After the XY1 card, the xy values of the series are listed, one pair per line, up to N pairs.

Output Control Parameters

There are two basic types of output generated by FEMWATER. The first type consists of printed text output summarizing the input data, the progress of the simulation (convergence criteria, etc.), and a summary of the results. This type of output is printed to the screen and to the printed output file described in Chapter 2. The other type of output consists of a series of binary or text solution files that can be used for graphical post-processing. The format of the solution files is described in Appendix C. The output control card group (OC1, OC2, OC3, and OC4) is used to specify what information is to be written to the printed output file and saved to the solution files. The cards also control the interval at which the output is printed or saved.

Print interval (OC1)

The OC1 card controls the frequency at which data are written to the printed output file and whether or not diagnostic output and the cyclic change of rainfall seepage nodes are printed.

Card Type	**OC1**		
Description	Interval for writing to printed output file.		
Required	YES		
Format	OC1 ibug ichng jopt kprt/nprint		
Sample	OC1 0 0 0 1 /* Print at every time-step */ OC1 0 0 1 5 /* Print at specified times */ 1.0 2.0 4.0 8.0 16.0		
Field	*Variable*	*Value*	*Description*
1	IBUG	0 1	Do not print diagnostic output. Print diagnostic output.
2	ICHNG	0 1	Do not print cyclic change of rainfall/seep. nodes. Print cyclic change of rainfall/seep. nodes.
3	JOPT	0 1	Print at specified interval. Print at specified set of time values.
4	KPRT KPRINT	+ +	Print interval if JOPT=0, or Total number of specified times if JOPT=1.
5		+	If JOPT=1, after the OC1 card, list the specified print times, one per line, up to KPRINT times.

Diagnostic output (IBUG). The integer, IBUG, is an indicator for diagnostic output control. The diagnostic output will help users locate errors in the input data.

Cyclic change of rainfall-seepage nodes (ICHNG). The integer, ICHNG, is an indicator for controlling the printout of the convergence behavior at rainfall-seepage nodes.

Print frequency (JOPT). The information that is written to the printed output file can either be written out at a regular interval or at a specified set of time values. The time values in the specified set of time values should correspond to computational time-steps.

Print options (OC2)

The OC2 card is used to specify what information is written to the printed output file. The information is written at the interval specified in the OC1 card.

Card Type	**OC2**		
Description	Options for writing to printed output file.		
Required	YES		
Format	OC2 nselt kpro(i)		
Sample	OC2 3 1 4 7		
Field	*Variable*	*Value*	*Description*
1	NSELT	+	Total number of print options selected. This field should by followed by NSELT values, each of which ranges from 0-7 as explained below.
2+	KPRO(I)	0 1 2 3 4 5 6 7	Print nothing. Print flow and/or mass information at the boundary. Print total head. Print pressure head. Print concentration. Print flux. Print nodal moisture content. Print Darcy velocity.

The following items can be printed:

a. Flow and/or mass information at the boundary. The rate of change, incremental, and total fluid and/or mass flow through the boundary are printed (depending on whether simulation is flow only, transport only or coupled flow and transport).

b. Total head. The total head at each node is printed.

c. Pressure head. The pressure head at each node is printed.

d. Concentration. The concentration at each node is printed.

e. Flux. The flux at each node is printed.

f. Moisture content. The moisture content at each node is printed.

g. Darcy velocity. The Darcy velocity is printed at each node.

Save interval (OC3)

The OC3 card controls the frequency at which the computed solution is saved to output files for post-processing. The solution can be saved at a regular interval or at a specified set of time values. The time values in the specified set of time values should correspond to computational time-steps.

The OC3 is also used to specify whether the files should be saved in text or binary format. The binary format results in much smaller file sizes, an issue which may be very important when running transient analyses on large meshes. The text format requires more disk space, but it can be viewed with a text editor and transferred between different computer platforms. Binary files, however, cannot be transferred between different computer platforms.

Card Type	**OC3**		
Description	Interval for saving solution files for post-processing.		
Required	YES		
Format	OC3 ifile kopt kdsk/npost		
Sample	OC3 0 0 1 /* Save at every time-step */ OC3 0 1 5 /* Save at specified times */ 1.0 2.0 4.0 8.0 16.0		
Field	*Variable*	*Value*	*Description*
1	IFILE	0 1	Save in text format. Save in binary format.
2	KOPT	0 1	Save at specified interval. Save at specified set of time values.
3	KDSK KPOST	+ +	Save interval if KOPT=0, or Total number of specified times if KOPT=1.
4+		+	If KOPT=1, after the OC3 card, list the specified save times, one per line, up to KPOST times.

Save options (OC4)

The OC4 card is used to specify what information is saved to the solution files for post-processing or for use as initial conditions for subsequent simulations. Each set of information is written to a separate file, which is either binary or text depending on the option selected in the OC3 card. The formats of the solution files are described in Appendix C. The solutions are saved one time-step at a time at the interval specified in the OC3 card. At each time-step, one value is written for each node.

Card Type	**OC4**		
Description	Options for saving solution files for post-processing.		
Required	YES		
Format	OC4 kselt ksave(i)		
Sample	OC4 4 1 2 3 4		
Field	*Variable*	*Value*	*Description*
1	KSELT	+	Total number of solution files selected. This field should be followed by KSELT values, each of which ranges from 0-5 as explained below.
2+	KSAVE(I)	0	Save nothing.
		1	Save pressure head.
		2	Save nodal boundary fluid flux.
		3	Save nodal moisture content.
		4	Save velocity.
		5	Save concentration.

The following types of solution files can be saved:

a. Pressure head.

b. Nodal boundary fluid flux. Flux values at nodes without boundary conditions or point source/sinks assigned will be zero.

c. Nodal moisture content computed at nodes as an average of surrounding elements.

d. Darcy velocity.

e. Concentration.

5 Material Properties

Introduction

The second of the four primary groups of information in the model file is the material properties. The material properties include fluid properties and soil properties. One set of fluid properties is defined for the entire mesh, and one set of soil properties is defined for each of the soil or aquifer types referenced by the element material ID's.

File Format

The set of cards in the model file corresponding to material properties is shown in Figure 5.

```
.
MP1 kcp                                          /* Cond. Or perm. Flag*/
MP2 i kxx kxy kz kxy kxz kyz alpha por           /* Cond. Or perm. Values */
MP3 rho visc grav betap                          /* Dens. And visc. Of water */
MP4 a1 a2 a3 a4 a5 a6 a7 a8                      /* Dens. And visc. Coeff.s */
MP5 I k gamma al at am t decay n deckw decks     /* Disp./diffusion coeff.s */
SP1 I ihm ihc ihw                                /* Soil prop.For unsat. zone */
.
```

Figure 5. The material properties cards in the model file

Fluid Properties

The fluid property cards input to the model file are used to specify the density, the viscosity and the compressibility of fluid, and the acceleration of gravity. The acceleration of gravity is not a fluid property, but it is input on the same line as the density and viscosity for convenience. Specifying the acceleration of gravity allows the user to use any desired units for the fluid properties, soil properties, and all other parameters input to FEMWATER. All parameters should be consistent with the units of the specified acceleration of gravity. For example, if the gravity constant is specified in units of m/day^2, all length units, including mesh coordinates, must be in meters and all time units, including time-step data, must be specified in units of days. However, since the most common mass concentration unit used in groundwater is milligrams per liter (mg/ ℓ) in SI units, the concentrations in FEMWATER should be specified in units of mg/ ℓ (ppm).

FEMWATER can be used to model density-driven flow and transport. Thus, relationships must be defined between concentration, density, and viscosity. The relationships used by FEMWATER are:

$$\frac{\rho}{\rho_o} = a_1 + a_2C + a_3C^2 + a_4C^3 \tag{31}$$

and

$$\frac{\mu}{\mu_o} = a_5 + a_6C + a_7C^2 + a_8C^3 \tag{32}$$

where

ρ_o, μ_o = density and viscosity of fresh water

$a_1 \ldots a_8$ = parameters used to define concentration dependence of water density and viscosity

C = chemical concentration

Thus, values of ρ_o, μ_o, and $a_1 \ldots a_8$ must be specified by the user.

Density, viscosity and compressibility of fresh water and acceleration of gravity (MP3)

The density, viscosity, and compressibility of fresh water , ρ_o, μ_o, β, and the acceleration of gravity are specified with the MP3 card. Note that FEMWATER requires the user to enter the compressibility of fresh water, not the modified compressibility term used in Equation (2). Modified compressibility is computed by FEMWATER according to Equation (A23). Typically the compressibility of water is a small number, generally on the order of 10^{-15} - 10^{-17}, depending on the units used. Note that the units of compressibility are LT^2/M. Compressibility values are sometimes given in English units of ft^2/lb. The reader is reminded that when using English units, pounds are units of force, not mass and the appropriate conversion to slugs must be made. As with all parameters in FEMWATER, it falls to the user to ensure that all values are specified in units consistent with the model dimensions of length, mass and time.

Card Type	**MP3**		
Description	Density, viscosity and compressibility of fresh water.		
Required	YES		
Format	MP3 rho visc grav beta		
Sample	MP3 1000.0 4.68 1.3e+8 0.0		
Field	*Variable*	*Value*	*Description*
1	RHO	+	Density of fresh water [M/L^3].
2	VISC	+	Dynamic viscosity of water [M/L^2/T].
3	GRAV	+	Acceleration of gravity [L/T^2].
4	BETA	+	Compressibility of water [LT^2/M].

Concentration dependence coefficients (MP4)

The coefficients, $a_1 \ldots a_8$, which are used to define the concentration dependence of water density and viscosity, are specified with the MP4 card. The parameters a_1 and a_5 should always set to 1.0 and for density-independent flow the parameters a_2, a_3, a_4, a_6, a_7, and a_8 should also be set to zero.

For density-dependent flow, some or all of the parameters a_2-a_4 and a_6-a_8 should be nonzero. For example, typical salt water has a salt concentration of around 1 kilogram per liter (1000 ppt). For a first-order density-dependent relationship, the parameter a_1 is set to 1.0, a_2 is set to 0.001 (the inverse units of concentration), and the parameters a_3 and a_4 are set to zero.

Card Type	**MP4**		
Description	Concentration dependence coefficients.		
Required	YES		
Format	MP4 a1 a2 a3 a4 a5 a6 a7 a8		
Sample	MP4 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0		
Field	*Variable*	*Value*	*Description*
1	RHOMU(1)	1	Coefficient a1.
2	RHOMU(2)	±	Coefficient a2. The units are the inverse of concentration units.
3	RHOMU(3)	±	Coefficient a3. The units are the inverse of concentration units squared.
4	RHOMU(4)	±	Coefficient a4. The units are the inverse of concentration units cubed.
5	RHOMU(5)	1	Coefficient a5.
6	RHOMU(6)	±	Coefficient a6. The units are the inverse of concentration units squared.
7	RHOMU(7)	±	Coefficient a7. The units are the inverse of concentration units cubed.
8	RHOMU(8)	±	Coefficient a8.

Soil Properties

Each element in the mesh is assigned a material corresponding to the zone or aquifer in which the element is located. The material ID is an index to a list of soil properties. The list of soil properties is input to the model file. The soil properties that must be specified are hydraulic conductivity, compressibility, dispersion, diffusion, radioactive decay coefficient, the first order biodegradation rate of dissolved and absorbed phase, and three water retention curves describing how moisture content, relative conductivity, and water capacity vary with pressure head in the unsaturated zone.

Hydraulic conductivity (MP1, MP2)

The hydraulic conductivity for each soil type is specified using the MP1 and MP2 cards. The MP1 card is used to specify whether the conductivity values should be interpreted as hydraulic conductivity or as permeability.

Card Type	**MP1**		
Description	Hydraulic conductivity versus permeability.		
Required	YES		
Format	MP1 kcp		
Sample	MP1 0		
Field	*Variable*	*Value*	*Description*
1	KCP	0 1	Hydraulic conductivity. Permeability.

The hydraulic conductivity **K** is a symmetric tensor of second order. It relates the flux **F** to the gradient of total head, H, in a rectangular system (x,y,z) as

$$\begin{aligned} F_x &= -\left(K_{xx}\frac{\partial H}{\partial x} + K_{xy}\frac{\partial H}{\partial y} + K_{xz}\frac{\partial H}{\partial z} \right) \\ F_y &= -\left(K_{yx}\frac{\partial H}{\partial x} + K_{yy}\frac{\partial H}{\partial y} + K_{yz}\frac{\partial H}{\partial z} \right) \\ F_z &= -\left(K_{zx}\frac{\partial H}{\partial x} + K_{zy}\frac{\partial H}{\partial y} + K_{zz}\frac{\partial H}{\partial z} \right) \end{aligned} \tag{33}$$

Since the tensor is symmetric, a coordinate system (ξ,ψ,ζ) can be found such that (Long 1961):

$$\begin{aligned} F_\xi &= -K_{\xi\xi}\frac{\partial H}{\partial \xi} \\ F_\psi &= -K_{\psi\psi}\frac{\partial H}{\partial \psi} \\ F_\zeta &= -K_{\zeta\zeta}\frac{\partial H}{\partial \zeta} \end{aligned} \tag{34}$$

Therefore, the hydraulic conductivity tensor **K′** in the (ξ,ψ,ζ) coordinate system can be written as

$$\mathbf{K} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} \quad \mathbf{K'} = \begin{bmatrix} K_{\xi\xi} & 0 & 0 \\ 0 & K_{\psi\psi} & 0 \\ 0 & 0 & K_{\zeta\zeta} \end{bmatrix} \tag{35}$$

The transformation of K between the coordinate system (ξ,ψ,ζ) and (x,y,z) is given as follows.

Let θ, ϕ, and φ be the Euler angles between the (ξ,ψ,ζ) and (x,y,z) coordinates: θ is the angle between the ζ- and z-axis on the plane perpendicular to the y-axis, ϕ is the angle between ξ and x-axis on the plane perpendicular to z-axis, and φ is the third angle (Morse and Feshbach 1978). The relationship between (x,y,z) and (ξ,ψ,ζ) is given by

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \begin{Bmatrix} \xi \\ \psi \\ \zeta \end{Bmatrix} \tag{36}$$

in which

$\alpha_1 = \sin\varphi \sin\phi + \cos\varphi \cos\phi \cos\theta$
$\alpha_2 = \cos\varphi \sin\phi - \sin\varphi \cos\phi \cos\theta$
$\alpha_3 = \cos\phi \cos\theta$
$\beta_1 = \sin\varphi \cos\phi - \sin\varphi \sin\phi \cos\theta$
$\beta_2 = \cos\varphi \cos\phi + \sin\varphi \sin\phi \cos\theta$
$\beta_3 = -\sin\phi \sin\theta$
$\gamma_1 = -\cos\varphi \sin\theta$
$\gamma_2 = \sin\varphi \sin\theta$
$\gamma_3 = \cos\theta$

where α's, β's, and γ's are the directional cosines between two coordinate systems. These directional cosines are symmetrical with respect to the two coordinate systems. The transformation of the flux vector between two coordinate systems is similar to that between the coordinates, i.e.,

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \begin{Bmatrix} F_\xi \\ F_\psi \\ F_\zeta \end{Bmatrix} \tag{37}$$

Substituting Equations (33) and (34) into Equation (37) gives

$$\begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} \begin{Bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial y} \\ \frac{\partial H}{\partial z} \end{Bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \begin{bmatrix} K_{\xi\xi} & 0 & 0 \\ 0 & K_{\psi\psi} & 0 \\ 0 & 0 & K_{\zeta\zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial H}{\partial \xi} \\ \frac{\partial H}{\partial \psi} \\ \frac{\partial H}{\partial \zeta} \end{Bmatrix} \quad (38)$$

By the chain rule,

$$\begin{Bmatrix} \frac{\partial H}{\partial \xi} \\ \frac{\partial H}{\partial \psi} \\ \frac{\partial H}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \psi} & \frac{\partial y}{\partial \psi} & \frac{\partial z}{\partial \psi} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial y} \\ \frac{\partial H}{\partial z} \end{Bmatrix} \quad (39)$$

The Jacobian can be obtained from Equation (37):

$$\begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \psi} & \frac{\partial y}{\partial \psi} & \frac{\partial z}{\partial \psi} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{bmatrix} \quad (40)$$

Substituting Equation (40) into Equation (39), then the resulting equation into Equation (38) gives

$$\begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \begin{bmatrix} K_{\xi\xi} & 0 & 0 \\ 0 & K_{\psi\psi} & 0 \\ 0 & 0 & K_{\zeta\zeta} \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{bmatrix} \quad (41)$$

Relabeling the directional cosines in Equation (41) gives

$$\begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{21} & \gamma_{31} \\ \gamma_{12} & \gamma_{22} & \gamma_{32} \\ \gamma_{13} & \gamma_{23} & \gamma_{33} \end{bmatrix} \begin{bmatrix} K_{\xi\xi} & 0 & 0 \\ 0 & K_{\psi\psi} & 0 \\ 0 & 0 & K_{\zeta\zeta} \end{bmatrix} \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix} \quad (42)$$

in which

$$\begin{aligned}
\gamma_{11} &= \sin\varphi \sin\phi + \cos\varphi \cos\phi \cos\theta \\
\gamma_{12} &= \sin\varphi \cos\phi - \cos\varphi \sin\phi \cos\theta \\
\gamma_{13} &= -\cos\varphi \sin\theta \\
\gamma_{21} &= \cos\varphi \sin\phi - \sin\varphi \cos\phi \cos\theta \\
\gamma_{22} &= \cos\varphi \cos\phi + \sin\varphi \sin\phi \cos\theta \\
\gamma_{23} &= \sin\varphi \sin \theta \\
\gamma_{31} &= \sin\theta \cos\phi \\
\gamma_{32} &= -\sin\theta \sin\phi \\
\gamma_{33} &= \cos\theta
\end{aligned}$$

Finally the working equations to compute the hydraulic conductivity components in the (x,y,z) coordinate are given by the following equation, when the principal components in the (ξ,ψ,ζ) coordinate and the three Euler angles between the (x,y,z) and (ξ,ψ,ζ) coordinates are known:

$$\begin{aligned}
K_{xx} &= \gamma_{11}^2 K_{\xi\xi} + \gamma_{21}^2 K_{\psi\psi} + \gamma_{31}^2 K_{\zeta\zeta} \\
K_{xy} &= \gamma_{11}\gamma_{12} K_{\xi\xi} + \gamma_{21}\gamma_{22} K_{\psi\psi} + \gamma_{31}\gamma_{32} K_{\zeta\zeta} \\
K_{xz} &= \gamma_{11}\gamma_{13} K_{\xi\xi} + \gamma_{21}\gamma_{23} K_{\psi\psi} + \gamma_{31}\gamma_{33} K_{\zeta\zeta} \\
K_{yx} &= \gamma_{12}\gamma_{11} K_{\xi\xi} + \gamma_{22}\gamma_{21} K_{\psi\psi} + \gamma_{32}\gamma_{31} K_{\zeta\zeta} \\
K_{yy} &= \gamma_{12}^2 K_{\xi\xi} + \gamma_{22}^2 K_{\psi\psi} + \gamma_{32}^2 K_{\zeta\zeta} \qquad (43) \\
K_{yz} &= \gamma_{12}\gamma_{13} K_{\xi\xi} + \gamma_{22}\gamma_{23} K_{\psi\psi} + \gamma_{32}\gamma_{33} K_{\zeta\zeta} \\
K_{zx} &= \gamma_{13}\gamma_{11} K_{\xi\xi} + \gamma_{23}\gamma_{21} K_{\psi\psi} + \gamma_{33}\gamma_{31} K_{\zeta\zeta} \\
K_{zy} &= \gamma_{13}\gamma_{12} K_{\xi\xi} + \gamma_{23}\gamma_{22} K_{\psi\psi} + \gamma_{33}\gamma_{32} K_{\zeta\zeta} \\
K_{zz} &= \gamma_{13}^2 K_{\xi\xi} + \gamma_{23}^2 K_{\psi\psi} + \gamma_{33}^2 K_{\zeta\zeta}
\end{aligned}$$

For anisotropic cases, the three principal components should not all be equal, i.e., $K_{\xi\xi} \neq K_{\psi\psi} \neq K_{\zeta\zeta}$. The nine components of hydraulic conductivity tensor in the (x,y,z) coordinate system must be computed given the three principal components and three Euler angles.

a. For isotropic cases, the three principal components should be all equal, i.e., $K_{xx} = K_{yy} = K_{zz} = K$. All off-diagonal terms of the hydraulic conductivity tensor must be zero no matter what coordinate system is used.

b. For horizontal isotropic cases, all off-diagonal terms must be zero, K_{xx} and K_{yy} must be equal to K, and K_{zz} must be different from K.

c. For the vertical isotropic cases, all off-diagonal terms must be zero, K_{yy} and K_{zz} must be equal to K, and K_{xx} must be different from K.

The hydraulic conductivity values, and compressibility, α, of the medium are entered on the MP2 card. The K values should correspond to either hydraulic conductivity or permeability depending on the status of the KCP variable on the MP1 card. Note that as with the compressibility of the fluid, FEMWATER requires the user to enter the compressibility of the medium, not the modified compressibility used to determine the storage coefficient in Equation (2). Modified compressibilty is obtained from Equation (A22) by multiplying the user-specified density and gravity values by the compressibility value entered here. Soil compressibility values are also typically quite small, on the order of $10^{-15} – 10^{-17}$, depending on the units of the simulation. The reader is again reminded that when using reported compressibility values in English units, these are often given in units of ft^2/lb. FEMWATER requires compressibility to be entered in units of LT^2/M. Since pounds are units of force, they must first be converted to slugs (units of mass). It is the responsibility of the user to ensure that all values used in a simulation are in consistent units of length, mass and time.

The user may also note that porosity is used in Equation (2). Here, “porosity” means total porosity and is defined according to Equation (A9) in Appendix A. At fully saturated conditions, total porosity is equal to moisture content thus the porosity value on the MP2 card should be the same as the appropriate value on the moisture content vs. pressure head curve (see the section on the SP1 card later in this chapter).

Card Type	**MP2**		
Description	Hydraulic conductivity tensor and soil compressibility and porosity.		
Required	YES		
Format	MP2 I kxx kyy kzz kxy kxz kyz alpha por		
Sample	MP1 10 0.003 0.003 0.003 0.0 0.0 0.0 0.0 0.0		
Field	*Variable*	*Value*	*Description*
I	I	+	Material ID.
Kxx	PROPF(1,I)	+	Saturated xx hydraulic conductivity [L/T], or saturated permeability [L^2] of the medium i.
Kyy	PROPF(2,I)	+	Saturated yy hydraulic conductivity [L/T], or saturated permeability [L^2] of the medium i.
Kzz	PROPF(3,I)	+	Saturated zz hydraulic conductivity [L/T], or saturated permeability [L^2] of the medium i.
Kxy	PROPF(4,I)	+	Saturated xy hydraulic conductivity [L/T], or saturated permeability [L^2] of the medium i.
Kxz	PROPF(5,I)	+	Saturated xz hydraulic conductivity [L/T], or saturated permeability [L^2] of the medium i.
Kyz	PROPF(6,I)	+	Saturated yz hydraulic conductivity [L/T], or saturated permeability [L^2] of the medium i.
ALPHA	PROPF(7,I)	+	The compressibility of the medium i [LT^2/M].
POR	PROPF(8,I)	+	The total porosity of the medium i [dimensionless].

Dispersion/diffusion coefficients (MP5)

The MP5 card is used to specify the bulk density, the parameters required to define the isotherm relationship, the dispersion coefficients, the tortuosity, radioactive decay constant, and the first-order biodegradation rate constants for the dissolved and adsorbed phases for each soil type.

Card Type	**MP5**		
Description	Dispersion/diffusion coefficients.		
Required	YES		
Format	MP5 i k gamma al at am t decay n deckw decks		
Sample	MP5 10 0.0 1200.0 5.0 5.0 0.0 1.0 0.0 0.0 0.0 0.0		
Field	*Variable*	*Value*	*Description*
i	I	+	Material type ID.
k	PROPT(1,I)	+	Distribution coefficient, K_d, Freundlich K, or Langmuir K for medium i [L^3/M].
gamma	PROPT(2,I)	+	Bulk density for medium i [M/L^3].
al	PROPT(3,I)	+	Longitudinal dispersion for medium i [L].
at	PROPT(4,I)	+	Transverse dispersion for medium i [L].
am	PROPT(5,I)	+	Molecular diffusion coefficient for medium i [L^2/T].
t	PROPT(6,I)	+	Tortuosity for medium i [dimensionless].
decay	PROPT(7,I)	+	Radioactive decay constant for medium i [1/T].
n	PROPT(8,I)	+	Freundlich n or Langmuir S_{max} for medium i [dimensionless].
deckw	PROPT(11,I)	+	The first-order biodegradation rate constant through dissolved phase for medium i [1/T].
decks	PROPT(12,I)	+	The first-order biodegradation rate constant through adsorbed phase for medium i [1/T].
fm	PROPT(13,I)	+	The ratio of effective to total porosity, n_e/n [dimensionless].

The density is specified as a bulk density, in units of [M/L^3]. The bulk density of a soil corresponds to its oven-dried mass divided by its in situ volume.

The isotherm is used to define a relationship between the amount of contaminant adsorbed by the soil, S (mass of contaminant / mass of soil) and the concentration of the contaminant, C. Three types of isotherms can be used in FEMWATER:

a. Linear isotherm:

$$S = K_d C \tag{44}$$

b. Langmuir isotherm:

$$S = \frac{S_{max} KC}{1 + KC} \tag{45}$$

c. Freundlich isotherm:

$$S = KC^n \tag{46}$$

where

K_d = distribution coefficient [L^3/M]

S_{max} = maximum concentration of medium in the Langmuir nonlinear isotherm

K = coefficient in the Langmuir or Freundlich isotherm

n = power index in the Freundlich nonlinear isotherm.

Note that a linear isotherm can be obtained using Equation (44) by setting n equal to 1. The type of isotherm to be used is specified with the KSORP variable on the OP3 card. However, one set of coefficients, K_d, S_{max}, K, and n, is specified for each material type on the MP5 card.

Diffusion and dispersion are modeled in FEMWATER using the following relationship:

$$\theta \mathbf{D} = a_T |\mathbf{V}| \delta + (a_L - a_T) \frac{\mathbf{VV}}{|\mathbf{V}|} + a_m \theta \tau \delta \tag{47}$$

where

θ = moisture content

$\mathbf{D}$ = dispersion coefficient tensor

a_T = transverse dispersion [L]

δ = Kronecker delta tensor

$|\mathbf{V}|$ = the magnitude of the Darcy velocity $\mathbf{V}$ [L/T]

a_L = longitudinal dispersion [L]

a_m = molecular diffusion coefficient [L^2/T]

τ = tortuosity

Thus, the parameters, a_T, a_L, a_m, and t must be specified by the user. They can be obtained by experiment or literature review.

A radioactive decay constant, λ [1/T], may be specified for each material if the contaminant decays according to radioactive decay behavior. First-order biodegredation of the contaminant can also be simulated by entering biodegradation constants for both the dissolved and adsorbed phases of the contaminant in each material.

FEMWATER v3.0 now includes the ability to specify effective porosity for use in the transport calculations separately from the total porosity specified on the moisture content curve. Effective porosity is defined as the ratio of interconnected pore space to the total soil volume, or in other words, the ratio of pore space that is available for flow and transport. For some media, effective porosity can be greater than one order of magnitude smaller than total porosity. FEMWATER requires that the user input the ratio of effective to total porosity, n_e/n. If this parameter is omitted, the ratio is assumed to be 1 and the same value for both total and effective porosity will be used for all calculations.

Soil properties for unsaturated zone (SP1)

The final set of parameters that must be specified for each medium is a sequence of three curves defining how the moisture content, relative conductivity, and water capacity vary as a function of pressure head in the unsaturated zone.

Card Type	**SP1**
Description	Soil properties for unsaturated zone.
Required	YES
Format	SP1 i ihm, ihc, ihw

Sample	SP1 10 3 4 5		
Field	*Variable*	*Value*	*Description*
1	I	+	Material type index.
2	IHM(I)	+	Index of moisture content versus pressure head XY series.
3	IHC(I)	+	Index of relative conductivity versus pressure head XY series.
4	IHW(I)	+	Index of water capacity versus pressure head XY series.

Relative conductivity. The governing equation for flow of water through a variably saturated porous medium has effective hydraulic conductivity and storage terms. The effective hydraulic conductivity can be rewritten as the product of nonlinear and constant terms in the form:

$$K(h) = K_r K_S \qquad (48)$$

where K_r is the relative conductivity, ranging in value from 0.0 to 1.0, and K_S is the saturated hydraulic conductivity. The change of relative hydraulic conductivity is caused by changes in moisture content, resulting in the preferential movement of water through certain pathways, due to the influence of capillary forces. As the soil becomes less saturated, the flow of water becomes restricted to the pore sequences of smaller radii. This results in a reduction in the spatially averaged effective hydraulic conductivity.

Moisture content. Moisture content in the unsaturated zone is a function of the pressure head. The more negative the pressure head, the lower the moisture content. The curve defining the relationship between moisture content and pressure head should vary between the saturated moisture content, θ_s, and the residual moisture content, θ_r. The saturated moisture content is equal to the total porosity of the medium since all of the void space is filled with fluid and as mentioned previously, FEMWATER uses this value in all flow calculations where porosity is required. Effective porosity can be entered separately for use in transport calculations. See the section on the MP5 card in this chapter for further details on how to specify effective porosity. Under unsaturated conditions some of the void space is filled with air; thus, the moisture content is less than the medium's porosity and this curve is used to define how moisture content varies with negative pressure head. The residual moisture content represents the amount of water that cannot be removed from a soil by gravity drainage (even under large suction pressure) because it adheres to the grains of the soil.

Water capacity. The water capacity curve is equal to the slope of the moisture content versus pressure head curve. Although this curve could be determined automatically by FEMWATER from the moisture content curve, it is input by the user to avoid errors resulting from the approximate piecewise linear nature of the moisture content curve.

Generating the curves. Ideally, the relative conductivity, moisture content, and water capacity curves are determined directly by performing a series of tests on the soils involved in the study. However, in many cases they can be approximated using a set of measured or approximated constants and a set of empirical relationships. For example, one option for generating the curves is to use the van Genuchten functions (van Genuchten 1980):

$$K_r = \theta_e^{0.5}\left[1-\left(1-\theta_e^{1/\gamma}\right)^{\gamma}\right]^2 \tag{49}$$

and

$$\theta_e = \left[1+\left(|\alpha h|\right)^{\beta}\right]^{-\gamma} \quad \text{for } h < 0 \tag{50}$$

$$\theta_e = 1 \quad \text{for } h \geq 0 \tag{51}$$

where

$$\theta_w = \theta_r + \theta_e\left(\theta_s - \theta_r\right) \tag{52}$$

$$\gamma = 1-\frac{1}{\beta} \tag{53}$$

and

θ_w = moisture content (dimensionless)

θ_e = effective moisture content (dimensionless)

θ_s = saturation moisture content (dimensionless)

θ_r = residual moisture content (dimensionless)

β, γ = soil-specific exponents (dimensionless)

α = soil-specific coefficient (1/L)

Table 5 lists a set of saturated and residual moisture contents and the van Genuchten α and β terms for a variety of common soil types. When applying the α term, care should be taken to convert it to the proper units.

Table 5
Representative Soil Parameters

Soil Type	Saturated Moisture Content, θ_s	Residual Moisture Content, θ_r	$\alpha[cm^{-1}]$	β
Clay[1]	0.38	0.068	0.008	1.09
Clay loam	0.41	0.095	0.019	1.31
Loam	0.43	0.078	0.036	1.56
Loam sand	0.41	0.057	0.124	2.28
Silt	0.46	0.034	0.106	1.37
Silt loam	0.45	0.067	0.020	1.41
Silty clay	0.36	0.070	0.005	1.09
Silty clay loam	0.43	0.089	0.010	1.23
Sand	0.43	0.045	0.145	2.68
Sandy clay	0.38	0.100	0.027	1.23
Sandy clay Loam	0.39	0.100	0.059	1.48
Sandy loam	0.41	0.065	0.075	1.89

Note: [1] Agricultural soil, less than 60% clay
Source: Carsel and Parrish (1988)

Model convergence. The unsaturated zone soil property curves can have a significant effect on model convergence. If any portion of a curve has a sharp change in gradient (extreme curvature), FEMWATER may have a difficult time converging during the iterative solution process. To aid in convergence, the user is given the option of using a B-spline function to help guarantee smooth curves over all pressures. When this option is enabled, a B-spline will be fit through the points of the curves input by the user, eliminating the possibility for sharp gradient changes in the curves. Tests indicate up to 20% faster convergence by using this option on some simulations. This flag is set on the IP1 card and is discussed in the previous chapter under the IP1 card heading.

Entering the curves. The user must supply the moisture content, relative conductivity, and water capacity versus pressure head curves in tabular form. The curves are specified at the end of the model file using a set of xy series cards. The curves are referred to on the SP1 card by the ID’s of the xy series. The format of the xy series card is described in Chapter 4, “The XY Series format.”

6 Boundary Conditions

Introduction

The boundary conditions supported by FEMWATER include point sources and sinks[1] , Dirichlet (specified head), flux, flux gradient, and variable boundaries (rainfall/evaporation and seepage). In each case, the prescribed values can be either constant or vary with time. Not assigning boundary conditions to a mesh boundary is equivalent to a no-flux boundary in a flow simulation or to a no-diffusion flux boundary in a transport simulation.

Choosing Appropriate Boundary Conditions

FEMWATER offers the user several types of boundary conditions in an attempt to cover the range of valid boundary conditions that can be encountered in finite element groundwater modeling. Some confusion over which boundary conditions are appropriate for various situations will undoubtedly occur. The following discussion is intended to help in choosing appropriate boundary conditions in FEMWATER simulations.

The most common boundary condition used in groundwater modeling is the specified head (Dirichlet) boundary condition. If heads are known at a given location over the length of the simulation, this is the most appropriate boundary condition to use.

As a general rule, the variable boundary condition should be used when modeling rainfall and evaporation. This boundary condition is the most robust because it allows FEMWATER the freedom to change the boundary condition to a specified head boundary condition for both over- and undersaturated conditions (see the discussion of this boundary condition later in this chapter). This feature is especially applicable in transient simulations where the water table may fluctuate over the simulation time span.

[1] Point sources/sinks are technically not boundary conditions. However, they are included in this chapter for convenience.

Some modelers may be more accustomed to using the specified flux boundary condition for infiltration/evaporation and inflow/outflow flux conditions. Indeed, in some instances it is possible to know the flux on element faces with a reasonable degree of assurance. However, the user is cautioned that the specified flux is always taken in a direction parallel to the normal of the element face to which this boundary condition is applied (see the discussion of the flux boundary condition later in this chapter). Therefore, in a simulation where a known flux condition exists, the specified flux value must be adjusted for all element faces whose normals deviate from the known flux direction. In most groundwater modeling scenarios, and particularly in transient simulations, the model is best suited either by using specified head boundary conditions (for inflow/outflow conditions on model boundaries) or by allowing the numerical model to change a flux boundary condition to a constant head condition if conditions warrant (for a infiltration/evaporation flux condition). The variable boundary condition in FEMWATER provides the user with the latter capability.

Gradient flux boundary conditions, as indicated in the discussion of this type of boundary condition later in this chapter, are rarely encountered in modeling natural systems but are included in FEMWATER for completeness and for research applications.

File Format

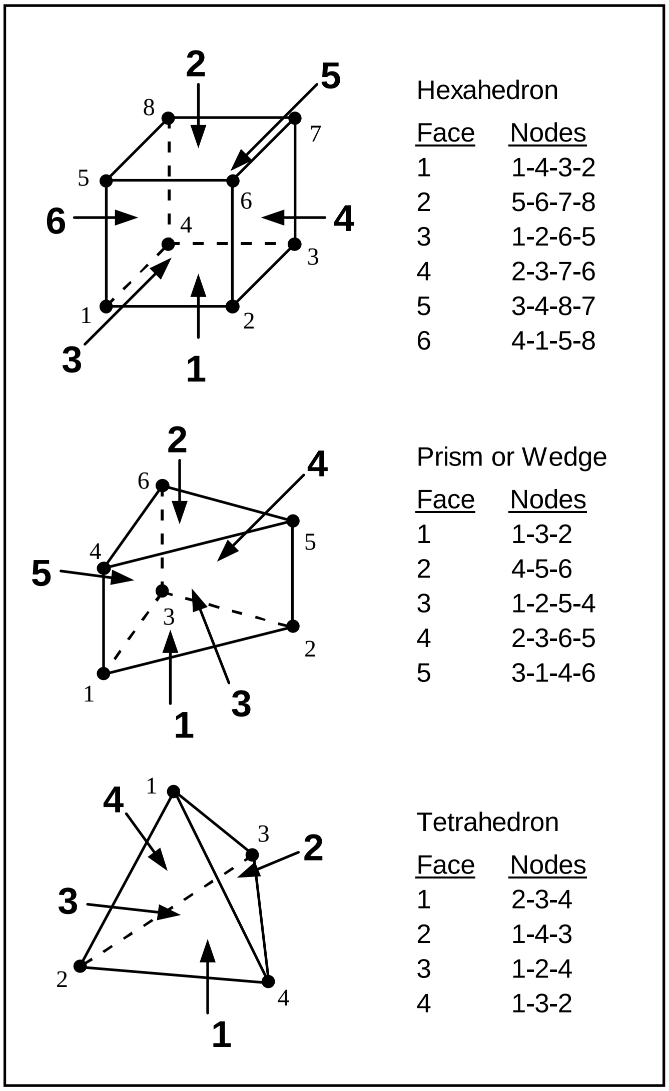
Boundary conditions are stored in the model file. The set of model file cards corresponding to boundary conditions is shown in Figure 6.

```
.
PS1 nodeid flowseries                /* Point source, flow rate */
PS2 nodeid concseries                /* Point source, concentration */
DB1 nodeid headseries                /* Dirichlet, head */
DB2 nodeid concseries                /* Dirichlet, concentration */
CB1 elemid faceid fluxseries         /* Flux, flux rate */
CB2 elemid faceid concseries         /* Flux, chemical flux rate */
NB1 elemid faceid fluxseries         /* Gradient Flux, gradient flux rate */
NB2 elemid faceid concseries         /* Gradient Flux, chemical gradient flux rate */
RS1 elemid faceid fluxseries         /* Variable, flux */
RS2 elemid faceid concseries         /* Variable, concentration */
RS3 hcon hmin                        /* Variable, pond depth, min hd. */
.
```

Figure 6. The boundary condition cards in the model file

Element Faces

The flux, gradient flux, and variable boundary conditions described in later sections are all “flux” type boundary conditions and are assigned to element faces. An element face is referenced by two indices, the element number and a face index. The element number is simply the element ID The numbering scheme used to determine face indices is shown in Figure 7.



Hexahedron

Face	Nodes
1	1-4-3-2
2	5-6-7-8
3	1-2-6-5
4	2-3-7-6
5	3-4-8-7
6	4-1-5-8

Prism or Wedge

Face	Nodes
1	1-3-2
2	4-5-6
3	1-2-5-4
4	2-3-6-5
5	3-1-4-6

Tetrahedron

Face	Nodes
1	2-3-4
2	1-4-3
3	1-2-4
4	1-3-2

Figure 7. Numbering scheme for element faces

Point Sources/Sinks (PS)

Point source/sink type boundary conditions consist of a volumetric flux or flow rate assigned to nodes in the mesh. Sources and sinks typically correspond to injection and extraction wells. For transport simulations, the concentration of the fluid can also be specified. The sign convention for point sources and sinks in FEMWATER is that a positive flow rate represents an injection well and a negative value represents an extraction well.

Point sources/sinks are specified with the PS1 and PS2 cards. The PS1 card is used to specify the flow rate (L^3/T), and the PS2 card is used to specify the concentration (M/ L^3). Both cards contain a field which is an index to an XY Series card. The XY Series card (specified elsewhere in the file) describes how the flow rate and concentration vary with time. If the value is constant (does not vary with time), only one value should be given in the XY Series. The XY Series card is described in Chapter 4.

It should be noted that the PS2 card is required for all transport simulation point sources and sinks. Thus a concentration must be specified for both injection and extraction wells. The value specified at the extraction well will not, however, affect the simulation results in any way.

A common practice in well installation is to inject and/or extract the fluid over a screened interval of the aquifer. This interval can be any distance from several centimeters to many meters. Since FEMWATER requires that well flow rates be specified at nodes, it is good practice to designate as many nodes of the finite element mesh as lie within the screened interval of the well as point sources and/or sinks. The flow rate for the well should be divided among the nodes used to simulate the screened interval of the well. It is not required that these flow rates be divided evenly among the well screen nodes thus allowing the user to weight rates of flow from different levels of the screened interval.

It should also be noted that FEMWATER requires point sources and sinks to be defined only in the saturated zone. Thus all wells, whether injection or extraction, must lie in the saturated zone of the aquifer and may not exist in the vadose zone. If a point source or sink is located in a region that becomes unsaturated during the course of the simulation, the simulation will halt at the timestep where this occurs, regardless of whether the well is injecting or extracting.

Card Type	**PS1**		
Description	Point source/sink flow rate for both flow and transport simulation.		
Required	NO		
Format	PS1 nodeid flowseries		
Sample	PS1 324 1		
Field	*Variable*	*Value*	*Description*
Nodeid	NPWF(I)	+	Node number.
Flowseries	IWTYPF(I)	+	Index of the flow rate XY Series.

Card Type	**PS2**		
Description	Point source/sink concentration for transport simulation.		
Required	NO		
Format	PS2 nodeid concseries		
Sample	PS2 324 2		
Field	*Variable*	*Value*	*Description*
Nodeid	NPWT(I)	+	Node number.
Concseries	IWTYPT(I)	+	Index of the concentration XY Series.

Dirichlet Boundary Conditions (DB)

A Dirichlet boundary condition consists of a prescribed total head at a node. A concentration may also be specified. The Dirichlet boundary condition is usually applied to soil-water interface such as stream, artificial impoundments, lakes, and coastal lines.

The governing flow equations for FEMWATER are written in terms of pressure head. Thus, the formal definition of a Dirichlet boundary condition for head is:

$$h = h_d(x_b, y_b, z_b, t) \quad \text{on } B_d, \tag{54}$$

where (x_b,y_b,z_b) is the spatial coordinate on the boundary, B_d is the Dirichlet boundary, and h_d is the prescribed pressure head. Since it is more convenient to specify total head rather than pressure head, Dirichlet boundary conditions are input as total head values, and they are converted internally by FEMWATER to pressure heads by subtracting the elevation of the nodes where the boundary conditions are assigned.

Thus, the formal definition of a Dirichlet concentration boundary condition for head is:

$$C = C_d(x_b, y_b, z_b) \quad \text{on } B_d, \tag{55}$$

where C_d is the prescribed concentration at the Dirichlet boundary. It should be noted that this is a fixed concentration and is independent of fluid flow direction.

There are situations in transport modeling where one desires to solve for the fluid concentration as the fluid is exiting through a specified head boundary. FEMWATER provides a means of accomplishing this through the use of two types of boundary conditions. The specified head values should be assigned to the nodes as described above however no specified concentration value should be assigned. The variable boundary condition (described later in this chapter) should be assigned to the element faces on this boundary. This will allow the user to assign a concentration to the fluid when flow is entering the system through the constant head boundary and when flow leaves the system through the boundary, a concentration will be computed by FEMWATER. The reader is referred to the discussion of the variable boundary condition later in this chapter for a more detailed description of how to implement this special combination of boundary conditions.

Dirichlet boundary conditions are specified with the DB1 and DB2 cards. The total head (L) is specified with the DB1 card for flow simulation, and the concentration (M/L^3) is specified with the DB2 card for transport simulation. In both cases, the values are specified with XY Series cards. If the head or concentration does not change with time, only one value should be listed in the XY Series.

Card Type	**DB1**		
Description	Dirichlet boundary conditions, prescribed total head for flow simulation.		
Required	NO		
Format	DB1 nodeid headseries		
Sample	DB1 324 2		
Field	*Variable*	*Value*	*Description*
Nodeid	NPDBF(I)	+	Node number.
Headseries	IDTYPF(I)	+	Index of the head XY Series.

Card Type	**DB2**		
Description	Dirichlet boundary conditions, prescribed concentration for transport simulation.		
Required	NO		
Format	DB2 nodeid concseries		
Sample	DB2 324 3		
Field	*Variable*	*Value*	*Description*
Nodeid	NPDBT(I)	+	Node number.
Concseries	JDTYPT(I)	+	Index of the concentration XY Series.

Flux Boundary Conditions (CB)

A flux boundary condition consists of a fluid flux prescribed at a boundary element face. A contaminant mass flux may also be specified. Formally, the flux boundary condition for flow can be stated as:

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h+\nabla z\right)=q_c(x_b,y_b,z_b,t)\quad \text{on } B_c, \tag{56}$$

where

n = outward unit vector normal to the boundary

(x_b, y_b, z_b) = spatial coordinate on the boundary

K = hydraulic conductivity tensor

ρ_o = reference (clean) fluid density

ρ = solution density

h = pressure head

z = elevation

q_c = flux rate

B_c = flux boundary

The flux boundary for transport can be stated as:

$$\mathbf{n} \cdot \left(\mathbf{V}C - \theta \mathbf{D} \cdot \nabla C \right) = q_c \left(x_b , y_b , z_b , t \right) \quad \text{on } B_c , \tag{57}$$

where

$\mathbf{V}$ = Darcy velocity

C = concentration

θ = moisture content

$\mathbf{D}$ = dispersion coefficient tensor

q_c = prescribed flux rate

B_c = flux boundary

The flux boundary condition is typically applied to surface water bodies with known infiltration rates through the layers of the bottom sediments or liners into the subsurface media. The flux rate for flow and transport is a negative value when it leaves the system and is a positive value when it enters the system.

Flux boundary conditions are specified with the CB1 and CB2 cards. The flux rate (L/T) is specified with the CB1 card for flow simulation, and the contaminant mass flux rate (mass per unit area per time, $M/L^2/T$) is specified with the CB2 card for transport simulation. In both cases, the values are specified with XY Series cards. If the flux rate does not change with time, only one value should be listed in the XY Series. Note that concentration flux is not specified with the flux boundary condition but rather contaminant mass flux. It is possible in FEMWATER to prescribe a fluid flux and a concentration flux at the same element faces by prescribing both the flux and Variable boundary conditions at the element faces. Fluid flux is specified at the element faces with the flux boundary condition with no contaminant mass flux. The Variable boundary condition, described later in this chapter, is then used to prescribe the concentration flux (with no fluid flux) at the same element faces. This combination of boundary condition types is very useful in modeling boundaries with known inflow concentrations and unknown outflow concentrations, such as in tidal saltwater boundaries.

Card Type	**CB1**		
Description	Flux boundary conditions, prescribed flux rate for flow simulation.		
Required	NO		
Format	CB1 elemid faceid fluxseries		
Sample	CB1 353 3 8		
Field	*Variable*	*Value*	*Description*
Elemid	NPCBF(I)	+	Element number.
Faceid	IDCF(I)	+	Element face ID.
Fluxseries	ICTYPF(I)	+	Index of the flux XY Series.

Card Type	**CB2**		
Description	Flux boundary conditions, prescribed flux rate for transport simulation.		
Required	NO		
Format	CB2 elemid faceid concseries		
Sample	CB2 353 3 9		
Field	*Variable*	*Value*	*Description*
Elemid	NPCBT(I)	+	Element number.
Faceid	IDCT(I)	+	Element face ID.
Concseries	ICTYPT(I)	+	Index of the contaminant mass flux XY Series.

Gradient Flux Boundary Conditions (NB)

The gradient flux boundary condition rarely occurs in nature, but is included for completeness. It consists of a fluid flux gradient prescribed at a boundary element face. The concentration of the fluid may also be specified. Formally, the gradient flux boundary condition for flow can be stated as:

$$-\mathbf{n}\cdot\mathbf{K}\cdot\frac{\rho_o}{\rho}\nabla h = q_n(x_b, y_b, z_b, t) \quad \text{on } B_n, \tag{58}$$

where

$\mathbf{n}$ = outward unit vector normal to the boundary

(x_b, y_b, z_b) = spatial coordinate on the boundary

$\mathbf{K}$ = hydraulic conductivity tensor

ρ_o = reference (clean) fluid density

ρ = solution density

h = pressure head

q_n = prescribed flux rate

B_n = Gradient flux boundary

The gradient flux boundary for transport can be stated as:

$$\mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C \right) = q_n \left(x_b, y_b, z_b, t \right) \quad \text{on}\, B_n, \tag{59}$$

where

C = concentration

θ = moisture content

D = dispersion coefficient tensor

q_n = prescribed flux rate

B_n = Gradient flux boundary

The flux gradient for flow and transport is a negative value when it leaves the system and is a positive value when it enters the system.

Gradient flux boundary conditions are specified with the NB1 and NB2 cards. The flux gradient (L/T) is specified with the NB1 card for flow simulation, and the flux rate (mass per unit area per time, $M/L^2/T$) is specified with the NB2 card for transport simulation. In both cases, the values are specified with XY Series cards. If the flux gradient does not change with time, only one value should be listed in the XY Series.

Card Type	**NB1**		
Description	Gradient flux boundary conditions, prescribed flux gradient for flow simulation.		
Required	NO		
Format	NB1 elemid faceid fluxseries		
Sample	NB1 353 3 8		
Field	*Variable*	*Value*	*Description*
Elemid	NPNBF(I)	+	Element number.
Faceid	IDNF(I)	+	Element face ID.
Fluxseries	INTYPF(I)	+	Index of the flux gradient XY Series.

Card Type	**NB2**		
Description	Gradient flux boundary conditions, prescribed flux rate for transport simulation.		
Required	NO		
Format	NB2 elemid faceid fluxseries		
Sample	NB2 353 3 9		
Field	*Variable*	*Value*	*Description*
Elemid	NPNBT(I)	+	Element number.
Faceid	IDNT(I)	+	Element face ID.
Fluxseries	INTYPT(I)	+	Index of the contaminant mass flux XY Series.

Variable Boundary Conditions (RS)

Variable boundary conditions are typically applied to the top faces of the uppermost layer of elements in the mesh (i.e., at the soil-air interface), and are used to simulate evaporation, seepage due to precipitation and seepage exit faces. Another powerful use of the variable boundary condition is in combination with either Dirichlet or flux boundary conditions to simulate boundaries with known head or fluid flux rates but unknown concentrations. Variable boundary conditions are called “variable” not because they can vary with time, but because they correspond to either a Dirichlet or a flux boundary condition depending on the potential evaporation, the conductivity of the media, the availability of water such as rainfall and the level of the groundwater.

During the precipitation period, variable boundary conditions are defined as

$$h = h_p(x_b, y_b, z_b, t) \quad \text{on}\, B_v, \tag{60}$$

or

$$-\mathbf{n} \cdot \mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) = q_p(x_b, y_b, z_b, t) \quad \text{on}\, B_v, \tag{61}$$

During the nonprecipitation period, variable boundary conditions are defined as

$$h = h_p(x_b, y_b, z_b, t) \quad \text{on}\, B_v, \tag{62}$$

or

$$h = h_m(x_b, y_b, z_b, t) \quad \text{on}\, B_v, \tag{63}$$

or

$$-\mathbf{n} \cdot \mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) = q_e(x_b, y_b, z_b, t) \quad \text{on}\, B_v, \tag{64}$$

where

h = pressure head

h_p = ponding depth

(x_b,y_b,z_b) = spatial coordinate on the boundary

B_v = variable boundary

n = outward unit vector normal to the boundary

K = hydraulic conductivity tensor

ρ_o = reference (clean) fluid density

ρ = solution density

z = elevation

q_p = precipitation

h_m = minimum pressure head

q_e = maximum evaporation rate

Only one of Equations (60)-(64) is used at any given time on the variable boundary. During the precipitation period, rainfall infiltrates to the subsurface at a rate equal to the prescribed precipitation flux, q_p, according to Equation (61). If the precipitation flux exceeds the infiltration capacity of the soil or if the groundwater table rises to the surface of the mesh, the pressure head is not allowed to rise above the ponding depth, h_p. If it does, the boundary condition switches to the Dirichlet condition of Equation (60), i.e., the pressure head is set equal to the ponding depth.

During the nonprecipitation period, the boundary condition corresponds to one of Equations (62)-(64). Once again, the pressure head is not allowed to rise above the ponding depth (Equation (62)). If the pressure head is below the ponding depth, the boundary condition is switched to the flux condition of Equation 64 and the flux is set equal to the specified evaporation flux, q_e. If the pressure head drops below the minimum pressure head, h_m, the boundary condition is switched to the Dirichlet condition of Equation (63), and the pressure head is set equal to the minimum pressure.

The reader might question how a Dirichlet boundary condition could ever be switched to a flux boundary condition because a Dirichlet boundary condition, by definition, will continually add or remove water from the system as needed to maintain the prescribed head. However, in the case of the variable boundary condition, Dirichlet conditions are valid at either the “low water” or “high water” conditions when nodes are either only adding or only removing water, respectively. Thus if a variable boundary which is currently set to a Dirichlet “low water” condition begins to remove water from the system, it is then switched to a flux boundary and the water is removed at the prescribed flux rate. The same is true but in reverse at the other extreme where Dirichlet nodes are only adding water at the “high water” condition. When the boundary begins to add water, the boundary type is changed to flux and the water is added at the prescribed fluid flux rate.

During periods when the variable boundary condition is set to a flux boundary condition, the specified flux value is taken to be directed vertically (i.e. in either the positive or negative z-direction) in order to simulate the primary direction of rainfall and evaporation. To calculate the volumetric flux entering the system, FEMWATER takes the specified flux rate and multiplies this value by the area of the element face projected onto a horizontal xy-plane (perpendicular to the flux direction). For element faces that deviate from the horizontal by some degree, this calculation will ensure that the flux is not artificially exaggerated. The variable boundary condition differs from the specified flux boundary condition in this matter. The specified flux boundary condition always assumes that the specified flux is parallel to the element face normal and thus the normal element face area is used for the volumetric flux calculation. This feature of the variable boundary condition allows FEMWATER to more accurately simulate real world conditions of rainfall infiltration and evaporation on a topographically varying surface.

Yet another use of the variable boundary condition is in the modeling of seepage exit faces and groundwater drains. The variable boundary condition can be applied to element faces where groundwater seepage is possible but unknown in quantity or exact location. A reasonable evaporation rate should be prescribed for conditions where the faces are unsaturated but if the water table rises and the element face becomes saturated, the boundary will act as a seepage face, removing groundwater from the system. In this fashion, potential seepage locations can be identified. This ability also allows FEMWATER to model the conditions found with agricultural drains and “gaining” streams that are fed by groundwater seepage but do not provide flow back to the groundwater system.

For transport simulations, during both the precipitation and nonprecipitation periods, the variable boundary condition is either a flux boundary condition with a given total flux, or a zero gradient flux. Mathematically, this can be expressed as:

$$\mathbf{n} \cdot \left(\mathbf{V}C - \theta \mathbf{D} \cdot \nabla C\right) = \mathbf{n} \cdot \mathbf{V} C_v \left(x_b, y_b, z_b, t\right) \quad \text{if } \mathbf{n} \cdot \mathbf{V} \le 0 \tag{65}$$

$$\mathbf{n} \cdot \left(- \theta \mathbf{D} \cdot \nabla C\right) = 0 \quad \text{if } \mathbf{n} \cdot \mathbf{V} > 0 \tag{66}$$

where

C = concentration

V = Darcy velocity tensor

θ = moisture content

D = dispersion coefficient tensor

C_v = concentration at the variable boundary

If the flow direction through the boundary is into the region, the flux condition of Equation (65) governs and the concentration in the incoming fluid is equal to the prescribed concentration C_v. If the flow direction through the boundary is out of the region, the gradient flux condition of Equation (66) governs, meaning that the concentration of the outgoing fluid is whatever it was before it left the region.

The behavior of the variable boundary condition in transport simulations provides a powerful tool in modeling conditions where fluid head or flux conditions are known but concentration flux is unknown. As mentioned in both the Dirichlet and flux boundary condition sections, it is possible to assign either Dirichlet or flux conditions at a boundary for the fluid, leaving the concentration unassigned for these types. Then, by assigning a variable type boundary condition for the concentration, the user is able to specify the concentration of the fluid when flow is entering the system but have FEMWATER calculate the concentration of the fluid when flow is leaving the system.

Variable boundary conditions are defined by specifying the minimum pressure head, h_m, the ponding depth, h_p, the flux rates q_p and q_e, and the concentration, C_v. The minimum pressure head and the ponding depth are specified on the RS3 card. Only one RS3 card is entered in the model file thus the minimum pressure head, h_m, and the ponding depth, h_p are global values for the entire simulation. The specified minimum pressure head and ponding depth are used for all element faces where a variable boundary condition is specified.

The flux rates (L/T) for precipitation and evaporation, q_p and q_e, are both specified on the RS1 card as an index to a single XY Series. The values in the series that are positive are treated as precipitation flux, and the values that are negative are interpreted as evaporation flux. Thus, cycles of rainfall/evaporation can be designated on one curve by varying the curve in the XY Series from positive to negative and vice versa.

The concentration (M/L^3) is entered on the RS2 card as an index to an XY series. The ponding depth (L) and the minimum pressure head (L) are both specified on the RS3 card.

Card Type	**RS1**		
Description	Variable boundary condition, rainfall/evaporation/seepage flux rate for flow simulation.		
Required	NO		
Format	RS1 elemid faceid fluxseries		
Sample	RS1 8749 4 12		
Field	*Variable*	*Value*	*Description*
Elemid	NPVBF(I)	+	Element number.
Faceid	IDRF(I)	+	Element face index.
Fluxseries	IVTYPF(I)	+	Index number of rainfall/evaporation time series for the element face.

Card Type	**RS2**		
Description	Variable boundary condition, rainfall/seepage concentration for transport simulation.		
Required	NO		
Format	RS2 elemid faceid concseries		
Sample	RS2 8749 4 13		
Field	*Variable*	*Value*	*Description*
Elemid	NPVBT(I)	+	Element number.
Faceid	IDRT(I)	+	Element face index.
Concseries	IVTYPT(I)	+	Index number of concentration time series for the element face.

Card Type	**RS3**		
Description	Variable boundary condition, ponding depth and minimum pressure head.		
Required	NO		
Format	RS3 hcon hmin		
Sample	RS3 0.35 -8.0		
Field	*Variable*	*Value*	*Description*
1	HCON	+	The ponding depth.
2	HMIN	±	The minimum pressure head.

7 Initial Conditions

Introduction

Whenever a FEMWATER analysis is performed, a set of initial conditions must be defined. Initial conditions define the initial status of the pressure head and concentration. Which initial conditions are required for a particular problem depends on the type of simulation that is being performed. The rules, options, and formats for defining initial conditions for FEMWATER are described in this chapter.

Types of Initial Conditions

Three types of initial conditions are possible for a FEMWATER simulation: cold starts, hot starts, and flow solutions. Cold starts are used to establish a set of initial values at the beginning of a steady-state or transient simulation. Hot starts are used to continue a previous run of FEMWATER without having to start over from a cold start. Flow solutions are used to define the flow field that is necessary when performing a transport-only simulation (as opposed to coupled flow and transport).

Cold Starts

FEMWATER uses an iterative method to find a solution to flow and transport problems. An iterative approach involves starting with an initial estimate of heads and concentrations and utilizing an iterative solver to gradually modify the initial values until they converge to a set of values that satisfy the underlying governing equations. Using an initial estimate of values is called a cold start.

Steady state versus transient

Cold starts must be specified regardless of whether the simulation is steady-state or transient. However, the approach taken to define the cold start may vary depending on whether the simulation is steady state or transient. If an initial condition is chosen with values that are closer to the final solution, the simulation will converge with fewer iterations (resulting in a much shorter execution time). Because of the non-linear nature of unsaturated zone calculations, the solution to the flow equation is often non-unique. Various sets of initial condition values chosen for a simulation can result in different converged solutions for the same set of boundary conditions, especially if the initial conditions used in each simulation vary significantly from each other. Special care should be taken to ensure that reasonable estimates of groundwater conditions are used as initial conditions for FEMWATER simulations.

With a transient simulation, the specification of the initial condition becomes more critical. In this case, the initial condition should represent the actual state of the aquifer at the beginning of the simulation period. In order for the simulation to be accurate, the initial condition must be compatible with the specified boundary conditions and stresses. For example, suppose that the transient effect of a new injection/extraction well system on an aquifer is to be simulated. If an incompatible initial condition is used, the response of the aquifer in the early part of the simulation will be due in part to the wells and in part to an adjustment of the aquifer that is required to make the heads and concentrations compatible with the other stresses and boundary conditions. Therefore, it is often best to first perform a steady state simulation with the boundary conditions that will not change during the transient simulation, and then use the solution to this simulation as the initial condition for the transient simulation. For this example, this would involve first running a steady-state simulation with all of the boundary conditions except for the injection and extraction wells. The solution to this simulation would then be used as the initial condition for the transient simulation, and the response of the aquifer would be due solely to the added stresses from the injection and extraction wells.

In the case of concentration, a steady-state simulation is often not applicable since transport seldom achieves steady-state conditions. In many cases, the initial condition will correspond to an existing plume that has been characterized via three-dimensional interpolation from measurements at scattered locations.

Required values

If a flow-only simulation is performed, a set of pressure heads is required for the cold start initial condition. If a transport-only simulation is performed, a set of concentrations is required (in addition to the flow solution as explained below). If a coupled flow and transport simulation is being performed, a set of heads and a set of concentrations are both required.

The set of pressure head values required for a cold start simulation can be an estimate based on a level pool or an interpolated set of pressure heads based on measured values at discrete locations. However, another option is to use the result of a previous FEMWATER simulation as the initial condition pressure heads. This technique is often called “hot starting” a simulation. FEMWATER has a hot start feature that is slightly different and only applies to transient runs which is described in the next section. If a simulation is approaching convergence but the maximum number of iterations specified on the IP1 card is reached before convergence is achieved, FEMWATER will write out the unconverged pressure head file and terminate. This file can then be used as the initial condition file for the next simulation, effectively starting the simulation where it stopped in the previous run. This technique is useful in choosing initial conditions for a plan condition and a converged solution for the base condition has already been computed in a previous simulation.

Another useful application of this technique is called “spinning up” a model. As mentioned in the previous section, if the cold start initial condition differs greatly from the final solution, the stresses due to the boundary conditions may cause FEMWATER to oscillate and have difficulty converging. One solution to this problem is to select a better cold start. Another solution is to start with a set of boundary conditions that are more compatible with the specified cold start and compute a solution. The boundary conditions are then modified slightly and the solution from the first simulation is used as the initial condition for the second simulation. This process is repeated and the boundary conditions are gradually changed until they correspond to the desired values.

Convergence

The values chosen for a cold start can have a significant impact on convergence. If the selected cold start differs greatly from the converged solution, i.e., is highly incompatible with the model stresses, FEMWATER may have difficulty converging. The quickest and most robust solution is achieved when cold start values are as close to the converged solution as possible.

Hot Starts

Since FEMWATER is a nonlinear, finite element model, solutions can require extensive computation time even for moderately sized meshes, particularly when a transient simulation involving numerous time-steps is being performed. During a modeling study, it is not uncommon to encounter situations where the solution progresses well to a certain point in time and then diverges or “blows up.” This situation can often be solved by decreasing the time-step size near the point in time where the solution degraded. In such cases, it is often useful to be able to restart the simulation at some point just before the problem occurred rather than starting over from the beginning. In order to do this, the “hot start” type of initial condition is required.

Hot start time

A hot start consists of a set of files representing a transient solution of pressure head, moisture content (optional), velocity (optional), and concentration and a hot start time. FEMWATER reads the transient solution file(s) and locates the set(s) of values at the time-step matching the hot start time. The appropriate values are then loaded into FEMWATER as initial conditions and the simulation is restarted. Moisture content and velocity files are not required by FEMWATER to hot start a simulation. However, if a stopped simulation wherein moisture content or velocity files were output is hot started and the user specifies the velocity and/or moisture content file(s) as input to the simulation, the new solutions from the hot started simulation will be appended to the velocity and moisture content files from the previous simulation.

When the initial or original simulation is performed, time-variant boundary conditions are typically defined on a time scale whose beginning corresponds to the start time of the simulation. When a hot start is performed, it is not necessary to redefine time-variant boundary conditions so that they are defined from a point corresponding to the beginning of the new start time. FEMWATER uses the specified hot start time and automatically interpolates the time-variant boundary conditions to the correct beginning time.

Required values

The solution files necessary for a hot start depend on the type of simulation. If a flow-only simulation is being performed, pressure head is required. If a transport-only simulation is being performed, concentration is required (As in all transport-only simulations, a flow solution for the entire simulation time is required but this is not specified as a hot start input parameter). If a coupled flow and transport simulation is being performed, pressure head and concentration are required. The velocity and nodal moisture content files are used for post-processing and are only required as input if the user wishes to have the new simulation solutions appended to the previous simulation solution files as described above. If the user desires to have the flow and/or concentration values from a hot start simulation appended to the simulation results of a previous simulation, the same file name for the initial condition and output files should be specified in the super file. See the section titled “Super File Input” at the end of this chapter for the super file input specifications.

File format

The files used for hot starts correspond to the data set solution files output by FEMWATER for post-processing. These files consist of solutions organized by time-step and are sequential lists of scalar or vector values, with one value listed per node. The nodal values are used for post-processing. The output control options which control data set output are described in the section “Save interval (OC3).” The formats of the data set files are described in Appendix C.

Flow Solutions

A third type of initial condition is required when a transport-only simulation is being performed. A transport-only simulation uses a previously computed flow solution (steady-state or transient) to define the three-dimensional flow field required to properly model the contaminant migration. The flow solution consists of two options: pressure head or velocity and pressure head written to data set files. When the first option is used, the needed velocity and moisture content for transport simulation are computed from the pressure head. When the second option is used, the needed moisture content is computed from the pressure head and velocity is read directly from the velocity file. Thus, when running the flow simulation, care should be taken to ensure that the proper options are selected for solution output (see “Save options (OC4)”).

The flow solution for a transport-only simulation is used in combination with either a cold start or a hot start. With a cold start, a set of initial concentration values is provided for concentration in addition to the steady-state or transient flow solution. With a hot start, a transient concentration solution and a hot start time are provided in addition to the flow solution.

Summary of Initial Condition Requirements

It can sometimes be confusing to determine which combination of initial conditions is required for a particular situation. The sets of initial conditions required for all possible scenarios are shown in Table 6. The X’s in the lower right corner of the table indicate which sets of values are required. For example, for a cold start with a coupled flow and transport simulation, pressure head and concentration must be specified.

Table 6
Initial Condition Requirements

	Initial Condition			
Simulation Type	**Press. Head**	**Moisture Cont.**	**Velocity**	**Concentration**
Cold Start				
Flow	X			
Transport	X			X
Coupled	X			X
Hot Start				
Flow	X	X[1]	X[1]	
Transport	X		X[1]	X
Coupled	X	X[1]	X[1]	X

[1] Solution files are optional

Initial conditions are specified in a two-step process. First of all, a set of cards must be entered in the model file specifying the basic control parameters for initial conditions such as the hot start time and the format of the initial condition files (text or binary). Next, the initial conditions are assembled in a set of data set files. The data set files should be located in the same directory as the other input files. A set of cards are entered in the super file that specify the names of the data set files containing the initial conditions.

Model File Input

A set of input cards is required in the model file to designate the control parameters for the initial conditions. The cards are summarized in Figure 8. Each of the cards is explained in more detail in the following sections.

```
.
ICS istart                      /* Cold start vs. hot start */
ICT hstime                      /* Hot start time */
ICC icon conval                 /* Use constant concentration */
ICH ihead hconst                /* Compute press from const head */
ICF icfile ivfile iffu          /* Initial cond. file format */.
.
```

Figure 8. The initial condition cards in the model file

Start type (ICS)

The ICS card is used to specify whether the simulation is to be started with a cold start or a hot start.

Card Type	**ICS**		
Description	Initial condition start type.		
Required	YES		
Format	ICS istart		
Sample	ICS 1		
Field	*Variable*	*Value*	*Description*
1	ISTART	0 1	Cold start. Hot start.

Hot start time (ICT)

If a hot start is specified on the ICS card, the hot start time must be specified on the ICT card. This time is used to interpolate the time variant boundary conditions to a proper starting point and to locate the proper set of initial conditions from the multiple time-steps defined in the data set files of pressure head, moisture content, etc.

Card Type	**ICT**		
Description	Initial condition hot start time.		
Required	YES		
Format	ICT hstime		
Sample	ICT 234.0		
Field	*Variable*	*Value*	*Description*
1	HSTIME	±	Hot start time.

Constant or variable concentration (ICC)

When defining a single set of concentration values for a cold start initial condition, it is often useful to use a constant value of concentration everywhere in the problem domain. For example, in many cases, an initial condition of zero concentration everywhere in the problem domain is appropriate. The ICC card can be used to easily define a constant concentration for the entire mesh. This alleviates the need to create a data set file with the same value repeated for each node in the mesh. If a constant value is not appropriate, the ICC card specifies that the initial condition varies spatially and the values are defined by a data set file.

Card Type	**ICC**		
Description	Constant or variable concentration initial condition.		
Required	NO		
Format	ICC icon conval		
Sample	ICC 0 0.0		
Field	*Variable*	*Value*	*Description*
1	ICON	0 1	Constant concentration initial conditions. Spatially variable concentration initial conditions.
2	CONVAL	±	If ICON=0, the concentration value. Otherwise, omit.

Constant or variable head (ICH)

The ICH is similar to the ICC card in that it can be used as a quick and easy way to define a cold start initial condition for pressure head. The ICH card can be used to designate that the pressure head varies spatially and that the values will be read from a data set file. Alternately, the card can be used to specify a constant value of total head. The pressure head initial condition is then defined in FEMWATER by subtracting the elevation of each node from the specified total head to define the pressure head for the node.

Card Type	**ICH**		
Description	Constant or variable head initial condition.		
Required	NO		
Format	ICH ihead hconst		
Sample	ICH 0 100.0		
Field	*Variable*	*Value*	*Description*
1	IHEAD	0 1	Constant total head initial conditions. Spatially variable pressure head initial conditions.
2	HCONST	±	If IHEAD=0, the total head value. Otherwise, omit.

Initial condition file format (ICF)

With the exception of the constant value approach for cold starts, the initial conditions for cold starts, hot starts, and flow solutions are defined in data set files. Data set files can be in either text or binary format. The ICF card is used to designate the format of the files so that they can be read properly by FEMWATER. A format flag is specified for the cold start or hot start files and another format flag is specified for the flow solution files used for a transport-only simulation (if necessary). In addition, the card includes a flag indicating whether the flow solution is steady state or transient.

Card Type	**ICF**		
Description	Initial condition file format.		
Required	YES		
Format	ICF icfile ivfile iffu		
Sample	ICF 1 1 1		
Field	*Variable*	*Value*	*Description*
1	ICFILE	0 1	The initial condition files are in text format. The initial condition files are in binary format.
2	IVFILE	0 1	The flow files are in text format. The flow files are in binary format.
3	IFFU	0 1	Flow files contain a steady-state solution. Flow files contain a transient solution.

Super File Input

As mentioned in the previous paragraph, initial conditions are typically defined with a set of data set files. The data set files should be located in the same directory as the other input files. The names of the data set files to be used for initial conditions must be specified in the FEMWATER super file described in Chapter 2. The initial condition cards in the super file are shown in Figure 9. Each card consists of a header, which identifies the type of file, and the filename. Only the cards corresponding to files needed for a particular simulation should be included.

```
.
ICHD filename                /* Pressure head init. cond. */
ICMC filename                /* Moisture content (nodal) init. cond. */
ICVL filename                /* Velocity init. cond.*/
ICCN filename                /* Concentration init. cond. */
FLVL filename                /* Velocity flow file (optional)*/
FLPH filename                /* Pressure head file (required)*/
.
```

Figure 9. The initial condition cards in the super file

The super file is also used to designate the names of the solution files. During a hot start, it is possible to use the same names for the hot start files and the solution files. If the same names are used, FEMWATER will append the solution to the end of the hot start files rather than overwriting the files or creating new files.

8 Sample Applications

The purpose of this section is to provide typical examples of FEMWATER applications after which one might pattern new applications. Five examples are provided that demonstrate different capabilities of FEMWATER. Each discusses boundary and initial condition specification. The input files associated with these examples are distributed with the FEMWATER executable and can be found in the "examples" directory.

Problem 1: Steady-State Wellhead Protection

The initial support for developing FEMWATER within the GMS was provided by the AERL to advance the state of the art and provide practice in doing wellhead protection studies. As a result, of the technologies accepted by the U.S. Environmental Protection Agency (EPA) to do wellhead protection studies, FEMWATER and GMS provide the most rigorous modeling tools. Several of these studies have been completed within the Department of Defense and many more are in the planning phase.

The provided example illustrates how to use FEMWATER to conduct a steady-flow simulation for a wellhead protection problem. The left and right sides of the model are bounded by freshwater head boundary conditions (Figure 10). The front and back of the model are groundwater divides represented by no-flux boundary conditions. The bottom of the model is an impervious stratum that does not allow significant amounts of water to leave the model. This is commonly located at aquicludes or deep aquitards. The pressure head is assumed to be hydrostatic throughout the model with water table initial conditions assumed to be a constant 12.0 m. The following boundary conditions are given: constant head of 12.0 m on the left side of the model, 13.0 m on the right side of the model, and no flux on the front and back of the model. Seventeen pumping wells are located in the top layer of coarse sand near the middle of the region. The pumping rate of the wells is a constant 0.25 m^3/hr. One of the wells could be increased to determine the effect on the other wells. The head tolerance is 0.001 m. The pointwise iterative solver is used to solve the matrix equations. The computational mesh consists of 4,000 elements and 3,507 nodes.

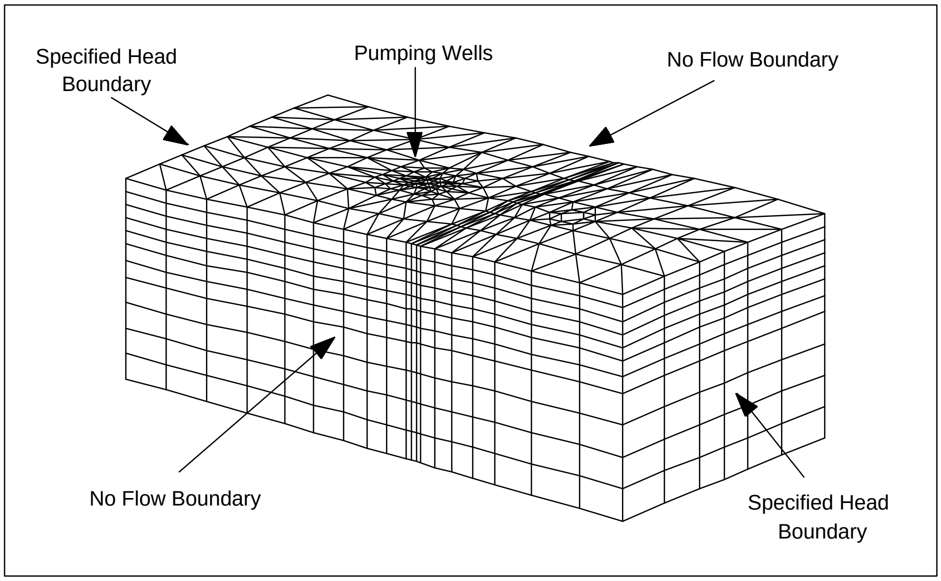


Figure 10. Wellhead protection application showing numerical model mesh and assigned boundary conditions

This problem took approximately 1.1 minutes for the steady-state solution to converge on a 200-MHz Pentium class personal computer with 32 MB of memory.

Problem 2: Transient Confined Disposal

Confined Disposal Facilities (CDF's) are essentially settling basins that accept dredged materials from navigation channels. The dredged material typically enters the CDF's as aqueous slurries for the purposes of settling out the solids and returning the clean water to the sea. This is a common activity for the U.S. Army Corps of Engineers. Regulatory agencies are increasingly requiring the use of 3-D groundwater models to determine the path of groundwater from the CDF to surrounding areas. Since dredged material sometimes contains compounds that should not enter drinking water supplies, models of CDF's can determine if the contamination of drinking water wells was caused by a particular CDF. In the interest of prevention, CDF's can be sited and designed to ensure that they will not contaminate local wells.

The provided example is presented to illustrate the procedure for conducting a transient flow simulation at a CDF. The left and right sides of the model are bounded by freshwater head boundary conditions (Figure 11). The front and back of the model are bounded by groundwater divides represented by no-flux boundary conditions. The bottom of the model is an impervious stratum. A pond (CDF) filled with a conservative chemical constituent is located on the right side of the model. Pressure is assumed to be hydrostatic throughout the model with water table initial conditions assumed to be a constant 12.0 m. The following boundary conditions are given: constant head of 12.0 m on the left side of the model, variable head with slowly rising water table assumed on the right side of the model, constant head of 13.0 m with concentration of 10 mg/ℓ assumed in the pond, and no flux imposed on the front and back of the model. A well with multiple screen openings is located near the center of the region. The pumping rate of the well is varied slowly with a slightly increasing rate. It starts with 0.05 m^3/hr and increased to 0.1 m^3/hr at time = 10.0 hr, where it stays constant at 0.1 m^3/hr. The head tolerance is 0.001 m and the pointwise iterative solver is used. A total of 200 hr with variable time-step is simulated.

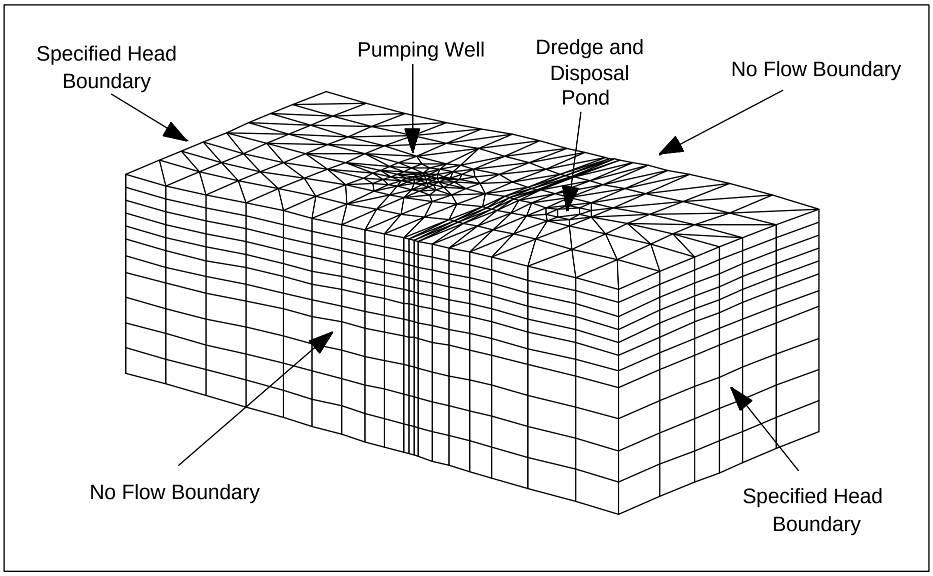


Figure 11. Confined disposal facility application showing numerical model mesh and assigned boundary conditions

This problem took approximately 9.1 minutes for the transient-flow solution to converge on a 200-MHz Pentium class personal computer with 32 MB of memory.

Problem 3: Transient Groundwater Remediation

This example illustrates a means for modeling a simple remediation situation using a cutoff wall to capture flows from a chemical-laden pond. The left and right sides of the model are bounded by freshwater head boundary conditions (Figure 12). The front and back of the model are bounded by groundwater divides represented by no-flux boundary conditions. The bottom of the model is an impervious stratum. A cutoff wall consisting of impervious media is installed in the middle of the model and a chemical lagoon is located on the right side. Pressure head is assumed hydrostatic throughout the region with water table initial conditions of 12.0 m throughout. The boundary conditions are as follows: constant head of 12.0 m on the left side, variable head with slowly rising head on the right side, constant head of 13.0 m with a concentration of 10 mg/ℓ in the ponding lagoon, and no-flux conditions on the front, back, and bottom. The pumping wells are located in the coarse sand layer near the center of the region. The pumping rate of the well slowly increases. The pumping starts with 0.05 m^3/hr and increases to 0.1 m^3/hr at time = 10.0 hr, where it stays constant at 0.1 m^3/hr. The pointwise iterative solver is used. A total of 500 hr with variable time-step is simulated.

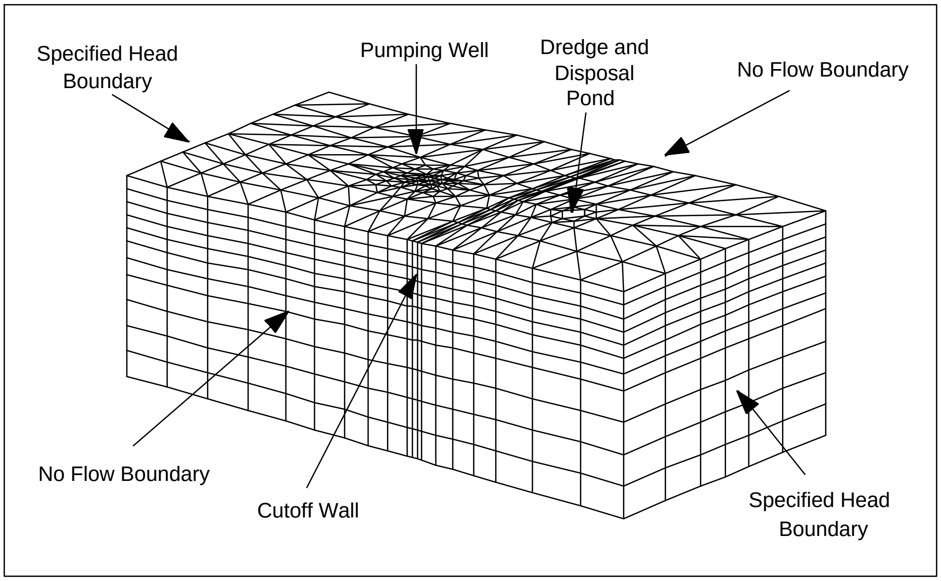


Figure 12. Groundwater remediation application showing numerical model mesh and assigned boundary conditions

This simulation took approximately 14.2 minutes for the flow and transport solution to converge on a 200-MHz Pentium class personal computer with 32 MB of memory.

Problem 4: Transient Non-point Chemical Spill

This problem presents an example of how to model a non-point chemical spill under transient conditions. It is presumed that a chemical was spilled over the entire surface of the region and as rainfall infiltrates, the spilled chemical dissolves and contaminates the rainfall that recharges the groundwater. The left and right sides are bounded by freshwater constant head boundary conditions (Figure 13). The front and back of the region are bounded by groundwater divides represented by no-flux boundary conditions. The bottom of the model is bounded by an impervious stratum. The pressure is assumed to be hydrostatic. The initial water surface elevation is a constant 11.0 m. The following boundary conditions are given: a constant head of 11.0 m on the left side of the region, a variable head with a slowly rising water table on the right side of the region, rainfall imposed as a variable boundary condition on the ground surface, a contaminated infiltrating fluid with a concentration of 10.0 mg/ℓ , and no flux imposed on the front and back of the region. A set of wells are located near the center of the region. The pumping rate of the well is varied with a slowly increasing rate. It starts with 0.05 m^3/hr and increases to 0.1 m^3/hr at time = 100 hr, and it stays constant at 0.1 m^3/hr. The head tolerance is 0.001 m and the pointwise iterative solver is used. A total of 150 hr with variable time-steps is simulated.

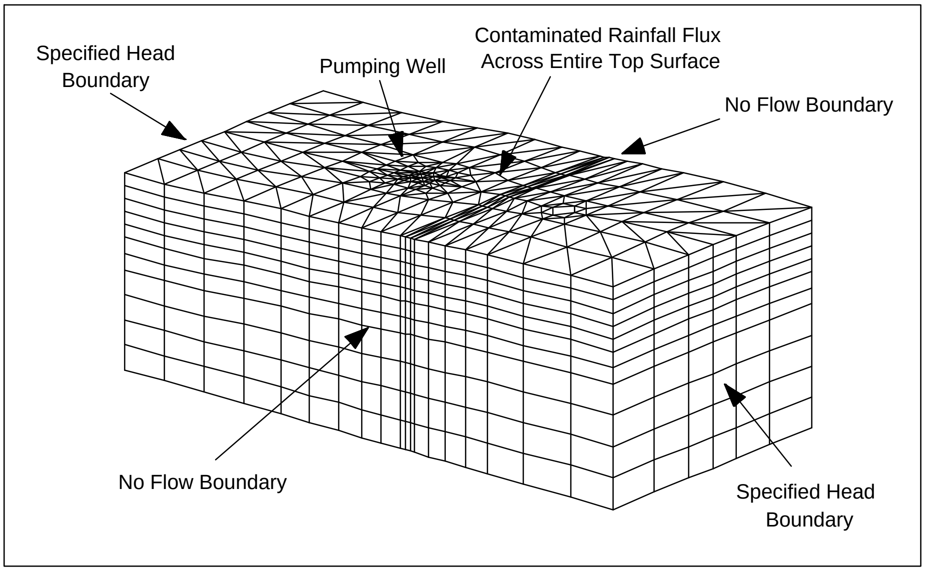


Figure 13. Non-point chemical spill application showing numerical model mesh and assigned boundary conditions

This problem took approximately 17.8 minutes for the flow and transport solution to converge on a 200-MHz Pentium class personal computer with 32 MB of memory.

Problem 5: Transient Salinity Intrusion

The salinity intrusion example demonstrates how FEMWATER can be used to study a variety of coastal aquifer problems. It is currently being used by the Corps of Engineers to determine the impacts of deepened navigation channels on salinity intrusion in coastal aquifers and how increased groundwater pumping will affect drinking water supplies.

In order to correctly simulate salinity movement in coastal aquifers, it is necessary to have coupled flow and density-driven transport. The example problem is a schematic example of a coastal salinity intrusion problem. The left side of the region is bounded by seawater, and the right side is bounded by fresh water (Figure 14). The front and back of the region are bounded by groundwater divides and the bottom by an impervious stratum. A pumping well is located near the center of the region. The boundary conditions are specified as a 20.0-kg/ ℓ tidal boundary on the left side and fresh water with a rising water table on the right side. The pumping rate of the well is varied with a slowly increasing rate. It starts with 0.5 m^3/hr and increases to 2.0 m^3/hr at time = 100.0 hr where it stays constant at 2.0 m^3/hr. A total of 10.0 hr with variable time-steps is simulated. The computational mesh consists of 5,664 elements and 3,484 nodes.

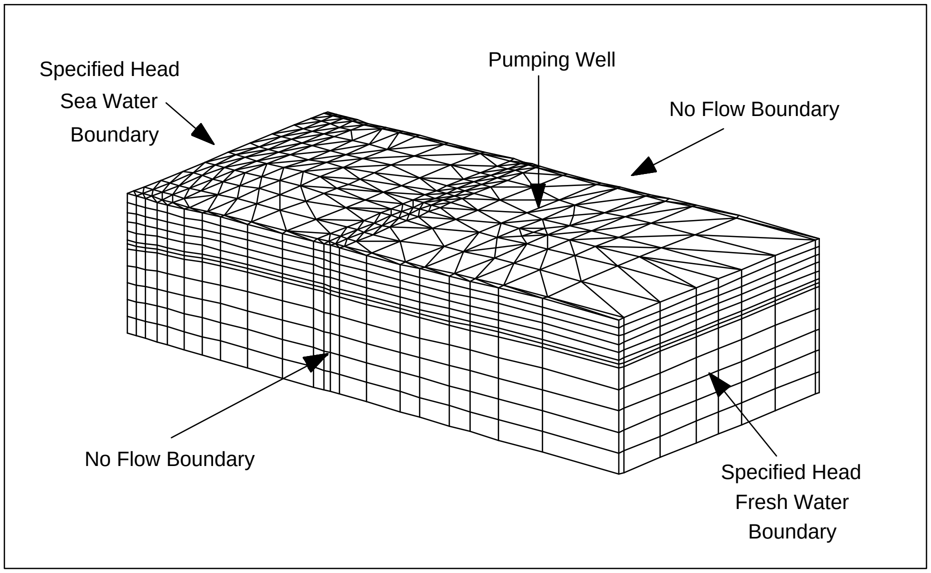


Figure 14. Transient salinity Intrusion application showing numerical model mesh and assigned boundary conditions

This problem took approximately 45.7 minutes for the transient solution to converge on a 200-MHz Pentium class personal computer with 32 MB of memory.

References

Carsel, R. F., and Parrish, R. S. (1988). "Developing joint probability distributions of soil-water retention characteristics," *Water Resources Research* 24(5),755-769.

Clough, R. W. (1971). "Analysis of structural vibrations and dynamic response." *Recent advances in matrix methods of structural analysis and design: Proceedings, United States-Japan Seminar on Matrix Methods of Structural Analysis and Design*. R. H. Gallagher, Y. Yamada, and J. T. Oden, ed., University of Alabama Press, Huntsville, AL.

Engineering Computer Graphics Laboratory (1996). *The Department of Defense Groundwater Modeling System Reference Manual*. Brigham Young University, Provo, UT.

Freeze, R. A. (1972a). "Role of subsurface flow in generating surface runoff: 1. Base flow contribution to channel flow," *Water Resources Research* 8, 609-623.

Freeze, R. A. (1972b). "Role of subsurface flow in generating surface runoff: 2. Upstream source areas," *Water Resources Research* 8, 609-623.

Frind, E. O. (1982). "Simulation of long-term transient density-dependent transport in groundwater," *Advances in Water Resources* 5(2), 73-88.

Galeati, G., Gambolati, G., and Neuman, S. P. (1992). "Coupled and partially coupled Eulerian-Langrangian model of freshwater-seawater mixing," *Water Resources Research* 28(1), 149-165.

Istok, J. D. (1989). *Groundwater modeling by the finite element method*. American Geophysical Union, Washington, DC.

Long, R. R. (1961). *Mechanics of solids and fluids*. Prentice-Hall, Englewood Cliffs, NJ.

Morse, P. M., and Feshbach, H. (1978). *Methods of theoretical physics*.

McGraw-Hill, New York.

Nguyen, V. V., Gray, W. G., Pinder, G. F., Botha, J. F., and Crefar, D. A. (1982). “A theoretical investigation on the transport of chemicals in reactive porous media,” *Water Resources Research* 18(4), 1149-1156.

Owczarek, J. A. (1964). *Fundamentals of gas dynamics*. International Textbook, Scranton, PA.

van Genuchten, M. T. (1980). “A closed-form equation for predicting the hydraulic conductivity of unsaturated soils,” *Soil Sci. Soc. J.* 44, 892-898.

Wang, J. D., and Conner, J. J. (1975). “Mathematical modeling of near-coastal circulation,” Report No. MITSG 75-13, Massachusetts Institute of Technology, Cambridge, MA, 1975.

Yeh, G. T. (1985). “Comparison of successive iteration and direct methods to finite element equations of aquifer contaminant transport,” *Water Resource Res.*21(3), 272-280.

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 media,” ORNL-6368, Oak Ridge National Laboratory, Oak Ridge, TN.

Yeh, G. T. (1990). “3DLEWASTE: A hybrid Lagrangian-Eulerian finite element model of waste transport through saturated-unsaturated media,” PSU Technical Report, Department of Civil Engineering, The Pennsylvania State University, University Park, PA.

Yeh, G. T. (1991). “Class notes: CE597D: Computational subsurface hydrology Part I,” Fall semester 1991, The Pennsylvania State University, University Park, PA.

Yeh, G. T. (1992). “Class notes: CE597C: Computational subsurface hydrology Part II,” Spring semester 1992, The Pennsylvania State University, University Park, PA.

Yeh, G. T., and Tripathi, V. S. (1987). “Strategies in modeling transport of reactive multi-chemical components.” *Groundwater and the environment: Proceedings, International Groundwater Conference*. Faculty of Engineering and Faculty of Physical and Applied Science, Universitti Kebangsaan Malaysis, Kuala Lampur, Malaysia, E26-E36.

Yeh, G. T., and Ward, D. S. (1981). “FEMWASTE: A finite element model of transport through saturated-unsaturated porous media,” ORNL-5601, Oak Ridge National Laboratory, Oak Ridge, TN.

Yeh, G. T., and Ward, D. S. (1980). “FEMWASTE: A finite element model of water flow through saturated-unsaturated porous media,” ORNL-5567, Oak Ridge National Laboratory, Oak Ridge, TN.

Appendix A Mathematical Formulation

Governing Equations for Flow

From the notion for continuity of fluid, continuity of solid, consolidation of the media, and the equation of state (Yeh 1992)[1] are obtained the starting equation for this derivation:

$$\nabla \cdot \left[\frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z \right) \right] - \nabla \cdot \left(\rho n S \mathbf{V_s} \right) + \rho^* q = \frac{\partial \left(n S \rho \right)}{\partial t} \qquad \text{(A1)}$$

where

ρ = fluid density (M/L^3)

k = intrinsic permeability tensor of the media (L^2)

μ = dynamic viscosity of the fluid (M/L/T)

p = fluid pressure [(ML/T^2)/L^2]

g = acceleration of gravity (L/T^2)

z = potential head (L)

n = porosity (L^3/L^3)

S = degree of saturation (dimensionless)

$\mathbf{V_s}$ = velocity of the deformable material due to consolidation (L/T)

[1] References cited in this Appendix are located at the end of the main text.

ρ* = density of the injected fluid (M/L^3)

q = internal source/sink [(L^3/T)/L^3]

t = time (T)

Expanding the right hand of Equation (A1):

$$\frac{\partial\left(nS\rho\right)}{\partial t} = Sn\frac{\partial \rho}{\partial t} + \rho S\frac{\partial n}{\partial t} + n\rho\frac{\partial S}{\partial t} \qquad \text{(A2)}$$

Expanding Equation (A2) by the chain rule:

$$\frac{\partial(nS\rho)}{\partial t} = Sn\frac{\partial \rho}{\partial p}\frac{\partial p}{\partial t} + Sn\frac{\partial \rho}{\partial C}\frac{\partial C}{\partial t} + \rho S\frac{\partial n}{\partial t} + n\rho\frac{\partial S}{\partial t} \qquad \text{(A3)}$$

where C is chemical concentration (M/L^3). Rearranging Equation (A3) gives:

$$\frac{\partial(nS\rho)}{\partial t} = Sn\frac{\partial \rho}{\partial p}\frac{\partial p}{\partial t} + \rho S\frac{\partial n}{\partial t} + Sn\frac{\partial \rho}{\partial C}\frac{\partial C}{\partial t} + n\rho\frac{\partial S}{\partial t} \qquad \text{(A4)}$$

where the first and second terms represent the storativity term, the third term is the density-concentration coupling term, and the fourth term is the unsaturated term. Substituting Equation (A4) into Equation (A1):

$$\begin{aligned} &\nabla \cdot \left[\frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z\right)\right] + \rho^* q = \\ &\quad Sn\frac{\partial \rho}{\partial p}\frac{\partial p}{\partial t} + Sn\frac{\partial \rho}{\partial C}\frac{\partial C}{\partial t} + n\rho\frac{\partial S}{\partial t} \\ &\quad + \rho S\frac{\partial n}{\partial t} + S\rho\nabla \cdot \left(n\mathbf{V}_s\right) + n\mathbf{V}_s \cdot \nabla(S\rho) \end{aligned} \qquad \text{(A5)}$$

Making the approximation by neglecting the second-order term:

$$n\mathbf{V}_s \cdot \nabla(S\rho) - - - - \rightarrow 0 \qquad \text{(A6)}$$

gives

$$\begin{aligned}
&\nabla \cdot \left[\frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z \right) \right] + \rho^* q = \\
&\quad Sn \frac{\partial \rho}{\partial p} \frac{\partial p}{\partial t} + Sn \frac{\partial \rho}{\partial C} \frac{\partial C}{\partial t} + n\rho \frac{\partial S}{\partial t} \\
&\quad + \rho S \frac{\partial n}{\partial t} + S\rho \nabla \cdot \left(n \mathbf{V}_s \right)
\end{aligned} \tag{A7}$$

Defining the compressibility of the fluid as:

$$\beta = \frac{1}{\rho} \frac{\partial \rho}{\partial p} \tag{A8}$$

where β is the compressibility of the fluid (LT^2/M). Also defining the moisture content as:

$$\theta = Sn \tag{A9}$$

where θ is the moisture content (dimensionless). Substitute Equations (A8) and (A9) into Equation (A7) and rewrite it to obtain:

$$\begin{aligned}
&\nabla \cdot \left[\frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z \right) \right] + \rho^* q = \\
&\quad \theta \beta \rho \frac{\partial p}{\partial t} + \theta \frac{\partial \rho}{\partial C} \frac{\partial C}{\partial t} + n\rho \frac{\partial S}{\partial t} \\
&\quad + \rho S \left[\frac{\partial n}{\partial t} + \nabla \cdot \left(n \mathbf{V}_s \right) \right]
\end{aligned} \tag{A10}$$

Remembering the continuity statement of incompressible solids, a compressible skeleton is defined as (Yeh 1992):

$$\frac{\partial (1-n)}{\partial t} + \nabla \cdot \left[(1-n) \mathbf{V}_s \right] = 0 \tag{A11}$$

Rearranging Equation (A11) in the following form:

$$\frac{\partial n}{\partial t} + \nabla \cdot \left(n \mathbf{V}_s \right) = \nabla \cdot \mathbf{V}_s \tag{A12}$$

Substituting Equation (A12) into Equation (A10) gives:

$$\nabla \cdot \left[\frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z \right) \right] + \rho^{*} q = \rho\theta\beta \frac{\partial p}{\partial t} + \theta \frac{\partial \rho}{\partial C} \frac{\partial C}{\partial t} + n\rho \frac{\partial S}{\partial t} + \rho S \nabla \cdot \mathbf{V}_s \qquad \text{(A13)}$$

Recalling that the flux of solid velocity is the divergence of $\mathbf{V}_s$ (Yeh 1992):

$$\nabla \cdot \mathbf{V}_s = \alpha \frac{\partial p}{\partial t} \qquad \text{(A14)}$$

where α is the coefficient of consolidation of the media (LT^2/M). Substituting Equation (A14) into Equation (A13) and rewriting:

$$\nabla \cdot \left[\frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z \right) \right] + \rho^{*} q = \rho(\theta\beta + S\alpha) \frac{\partial p}{\partial t} + \theta \frac{\partial \rho}{\partial C} \frac{\partial C}{\partial t} + n\rho \frac{\partial S}{\partial t} \qquad \text{(A15)}$$

Substituting Equation (A9) into Equation (15) gives

$$\nabla \cdot \left[\frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z \right) \right] + \rho^{*} q = \rho\left(\theta\beta + \frac{\theta}{n}\alpha \right) \frac{\partial p}{\partial t} + \theta \frac{\partial \rho}{\partial C} \frac{\partial C}{\partial t} + n\rho \frac{\partial S}{\partial t} \qquad \text{(A16)}$$

Experimental evidence has shown that the degree of saturation is a function of pressure as:

$$S = S(p) \qquad \text{(A17)}$$

Substitution of Equation (A17) into Equation (A16) gives:

$$\nabla\cdot\left[\frac{\rho\mathbf{k}}{\mu}\cdot\left(\nabla p+\rho g\nabla z\right)\right]+\rho^{*}q=$$
$$\rho\left(\theta\beta+\frac{\theta}{n}\alpha\right)\frac{\partial p}{\partial t}+\theta\frac{\partial\rho}{\partial C}\frac{\partial C}{\partial t}+\rho n\frac{dS}{dp}\frac{\partial p}{\partial t} \qquad \text{(A18)}$$

Next, the reference pressure head is defined as:

$$h=\frac{p}{\rho_o g} \qquad \text{(A19)}$$

where h is the reference pressure head (L) and ρ_o is the reference water density (M/L^3).

Substituting Equation (A19) into Equation (A18) gives:

$$\nabla\cdot\left[\frac{\rho\mathbf{k}}{\mu}\cdot\left(\rho_o g\nabla h+\rho g\nabla z\right)\right]+\rho^{*}q=$$
$$\rho\left(\theta\beta+\frac{\theta}{n}\alpha\right)\rho_o g\frac{\partial h}{\partial t}+\theta\frac{\partial\rho}{\partial C}\frac{\partial C}{\partial t}+\rho n\frac{dS}{dh}\frac{\partial h}{\partial t} \qquad \text{(A20)}$$

Dividing Equation (A20) by ρ_o and rearranging yields:

$$\nabla\cdot\left[\frac{\rho g\mathbf{k}}{\mu}\cdot\left(\nabla h+\frac{\rho}{\rho_o}\nabla z\right)\right]+\frac{\rho^{*}}{\rho_o}q=$$
$$\frac{\rho}{\rho_o}\left(\theta g\rho_o\beta+\frac{\theta}{n}g\rho_o\alpha\right)\frac{\partial h}{\partial t}+\frac{\theta}{\rho_o}\frac{\partial\rho}{\partial C}\frac{\partial C}{\partial t}+\frac{\rho}{\rho_o}n\frac{dS}{dh}\frac{\partial h}{\partial t} \qquad \text{(A21)}$$

Defining the modified compressibilities of the media and water as

$$\alpha'=\alpha\rho_o g \qquad \text{(A22)}$$

$$\beta'=\beta\rho_o g \qquad \text{(A23)}$$

where α' is the modified compressibility of the media (1/L) and β' is the modified compressibility of the water (1/L). Substituting Equations (A22) and (A23) into Equation (A21) and rearranging:

$$\nabla \cdot \left[\frac{\rho g \mathbf{k}}{\mu} \cdot \left(\nabla h + \frac{\rho}{\rho_o} \nabla z \right) \right] + \frac{\rho^*}{\rho_o} q = \frac{\rho}{\rho_o} \left(\alpha' \frac{\theta}{n} + \beta' \theta + n \frac{dS}{dh} \right) \frac{\partial h}{\partial t} + \frac{\theta}{\rho_o} \frac{\partial \rho}{\partial C} \frac{\partial C}{\partial t} \tag{A24}$$

Defining the storage coefficient as:

$$F = \alpha' \frac{\theta}{n} + \beta' \theta + n \frac{dS}{dh} \tag{A25}$$

where F is the storage coefficient. Substituting Equation (A25) into Equation (A24) and following Frind (1982) by neglecting the second term on the right side of Equation (A24),

$$\nabla \cdot \left[\frac{\rho g \mathbf{k}}{\mu} \cdot \left(\nabla h + \frac{\rho}{\rho_o} \nabla z \right) \right] + \frac{\rho^*}{\rho_o} q = \frac{\rho}{\rho_o} F \frac{\partial h}{\partial t} \tag{A26}$$

Defining the relation:

$$\mathbf{K} = \frac{\rho g \mathbf{k}}{\mu} \tag{A27}$$

where **K** is the hydraulic conductivity tensor. Substituting Equation (A27) into Equation (A26) and rearranging gives the density-dependent flow equation:

$$\frac{\rho}{\rho_o} F \frac{\partial h}{\partial t} = \nabla \cdot \left[\mathbf{K} \cdot \left(\nabla h + \frac{\rho}{\rho_o} \nabla z \right) \right] + \frac{\rho^*}{\rho_o} q \tag{A28}$$

From Darcy’s law:

$$\mathbf{V} = -\frac{1}{\rho} \frac{\rho \mathbf{k}}{\mu} \cdot \left(\nabla p + \rho g \nabla z \right) \tag{A29}$$

where **V** is the Darcy flux (L/T). Recalling Equation (A19) and substituting into Equation (A29) gives:

$$\mathbf{V} = -\frac{1}{\rho} \frac{\rho \mathbf{k}}{\mu} \cdot \left(\rho_o g \nabla h + \rho g \nabla z \right) \tag{A30}$$

Rearranging Equation (A30):

$$\mathbf{V} = -\frac{\rho g \mathbf{k}}{\mu} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) \tag{A31}$$

and substituting Equation (A27) into Equation (A31), we get the Darcy flux equation for density-dependent flow in its final form:

$$\mathbf{V} = -\mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \nabla h + \nabla z \right) \tag{A32}$$

The density and dynamic viscosity are functions of chemical concentration and are assumed to take the following form:

$$\frac{\rho}{\rho_o} = a_1 + a_2 C + a_3 C^2 + a_4 C^3 \tag{A33}$$

and

$$\frac{\mu}{\mu_o} = a_5 + a_6 C + a_7 C^2 + a_8 C^3 \tag{A34}$$

where C is the chemical concentration (M/L^3) and a_1, a_2, ..., a_7, a_8 are the parameters (L^3/M) that are used to describe the concentration dependence of water density and dynamic viscosity.

In the specific case of seawater intrusion, the constitutive relation between fluid density and concentration takes the following form:

$$\rho = \rho_o (1 + \varepsilon c) \tag{A35}$$

where c is the dimensionless chemical concentration (the actual one divided by the maximum one) and ε is the dimensionless density reference ratio defined as:

$$\varepsilon = \frac{\rho_{max}}{\rho_o} - 1 \tag{A36}$$

where ρ_{max} is the maximum density of the fluid (M/L^3) and ρ_o is the reference (freshwater) density (M/L^3). It should be noted that Equation (A36) implicitly assumes that the fluid is incompressible and under isothermal conditions (Galeati, Gambolati, and Neumann, 1992).

The initial conditions for the flow equations are stated as:

$$h = h_i(x, y, z) \quad \text{in} \quad R \tag{A37}$$

where R is the region of interest and h_i is the prescribed initial condition for hydraulic head. The specification of boundary conditions is probably the most critical and complex chore in flow modeling. As explained by Yeh (1987), the boundary conditions of the region of interest can be examined from a dynamic, physical, or mathematical point of view. From a dynamic standpoint, a boundary segment can be considered either as impermeable or flow-through. On the other hand, from a physical point of view, such a segment could be classified as a soil-soil interface, soil-air interface, or soil-water interface. Lastly, from a mathematical point of view, the boundary segment can be classified as one of four types of boundary conditions, namely as (a) Dirichlet, (b) gradient flux, (c) flux, or (d) variable boundary conditions. In addition, a good numerical model must be able to handle these boundary conditions when they vary on the boundary and are either abruptly or gradually time-dependent.

The Dirichlet boundary condition is usually applied to soil-water interfaces, such as streams, artificial impoundments, and coastal lines, and involves prescribing the functional value on the boundary. The gradient flux boundary condition, on the other hand, involves prescribing the gradient of the function on the boundary and does not occur very often in real-world problems. This condition, however, can be encountered at the base of the media where natural drainage occurs. The third type of boundary condition, the flux boundary condition, involves prescribing the total normal flux due to the gradient on the boundary. Usually surface water bodies with known infiltration rates through the bottom layers of their sediments or liners into the subsurface media are administered this boundary condition. If a soil-air interface exists in the region of interest, a variable boundary condition is employed. In such a case, either Dirichlet or flux boundary conditions dominate, mainly depending on the potential evaporation, the conductivity of the media, and the availability of water such as rainfall (Yeh 1987a).

From this discussion, four types of boundary conditions can be specified for the flow equations depending on the physical location of the boundaries:

a. Dirichlet boundary conditions:

$$h = h_d(x_b, y_b, z_b, t) \quad \text{on } B_d, \tag{A38}$$

b. Gradient flux boundary conditions:

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\cdot\nabla h\right)=q_n\left(x_b,y_b,z_b,t\right)\quad \text{on}\quad B_n, \tag{A39}$$

c. Flux boundary conditions:

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h+\nabla z\right)=q_c(x_b,y_b,z_b,t)\quad \text{on}\, B_c, \tag{A40}$$

d. Variable boundary conditions - during precipitation period:

$$h=h_p\left(x_b,y_b,z_b,t\right)\quad \text{on}\quad B_v, \tag{A41}$$

or

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h+\nabla z\right)=q_p\left(x_b,y_b,z_b,t\right)\quad \text{on}\quad B_v, \tag{A42}$$

e. Variable boundary conditions - during nonprecipitation period:

$$h=h_p\left(x_b,y_b,z_b,t\right)\quad \text{on}\quad B_v, \tag{A43}$$

or

$$h=h_m\left(x_b,y_b,z_b,t\right)\quad \text{on}\quad B_v, \tag{A44}$$

or

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h+\nabla z\right)=q_e\left(x_b,y_b,z_b,t\right)\quad \text{on}\quad B_v, \tag{A45}$$

f. River boundary conditions:

$$-\mathbf{n}\cdot\mathbf{K}\cdot\left(\frac{\rho_o}{\rho}\nabla h+\nabla z\right)=-\frac{K_R}{b_R}\left(h_R-h\right)\quad \text{on}\quad B_r\,, \tag{A46}$$

where

$\mathbf{n}$ = outward unit vector normal to the boundary

(x_b, y_b, z_b) = spatial coordinate on the boundary

h_d = Dirichlet functional value

q_h = Gradient flux value

q_c = Flux value

B_d = Dirichlet boundary

B_n = Gradient flux boundary

B_c = Flux boundary

B_v = variable boundary

h_p = ponding depth on the variable boundary

q_p = throughfall of precipitation on the variable boundary

h_m = minimum pressure on the variable boundary

q_e = evaporation rate (potential evaporation) on the variable boundary

K_R = hydraulic conductivity of the river bottom sediment layer

b_R = thickness of the river bottom sediment layer

h_R = depth of the river bottom measured from the river surface.

Note that only one of Equations (A41)-(A45) is utilized at any point on the variable boundary at any time.

Governing Equations for Transport

The governing equations for material transport through groundwater systems are derived based on the laws of continuity of mass and flux. The major processes that are included are advection, dispersion/diffusion, decay, adsorption, biodegradation through both liquid and solid phases, the compressibility of media, as well as source(s)/sink(s). Let C be the dissolved concentration and S be the adsorbed concentration. The governing equation of the spatial-temporal distribution of dissolved concentrations can be obtained by applying this principle of mass balance in integral form as follows:

$$\begin{aligned}\frac{D}{Dt}\int_v \left(\theta C + \rho_b S\right) dv = -\int_\Gamma n \cdot \left(\theta\, C \mathbf{V_{fs}}\right) d\Gamma \\ -\int_\Gamma \mathbf{n}\cdot \mathbf{J}\, d\Gamma - \int_v \left(\theta K_w C + \rho_b K_s S\right) dv \\ -\int_v \lambda\left(\theta C + \rho_b S\right) dv + \int_v m\, dv\end{aligned} \qquad \text{(A47)}$$

where

v = material volume containing constant amount of media (L^3)

C = dissolved concentration (M/L^3)

S = adsorbed concentration (M/M)

ρ_b = bulk density of the medium (M/L^3)

Γ = surface enclosing the material volume v (L^2)

n = outward unit vector normal to the surface G

$\mathbf{V}_{fs}$ = fluid velocity relative to the solid (L/T)

J = surface flux with respect to fluid velocity $\boldsymbol{V_{fs}}$ [$(M/T)/L^2$]

K_w = first-order biodegradation rate constant through dissolved phase (1/T)

K_s = first-order biodegradation rate constant through adsorbed phase (1/T)

λ = decay constant (1/T)

m = external source/sink rate per medium volume [$(M/L^3)/T$]

By the Reynolds transport theorem (Owczarek 1964), Equation (A47) can be written as

$$\begin{aligned}\int_v \frac{\partial\left(\theta\, C + \rho_b S\right)}{\partial\, t} dv + \int_\Gamma \mathbf{n}\cdot\left[\left(\theta\, C + \rho_b S\right)\mathbf{V_s}\right] d\Gamma + \int_\Gamma \mathbf{n}\cdot\left(\theta\, C\mathbf{V_{fs}}\right) d\Gamma = \\ -\int_\Gamma \mathbf{n}\cdot\mathbf{J}\, d\Gamma - \int_v \left(\theta K_w C + \rho_b K_s S\right) dv - \int_v \lambda\left(\theta\, C + \rho_b S\right) dv + \int_v m\, dv\end{aligned} \qquad \text{(A48)}$$

where $\mathbf{V}_s$ is the velocity of the solid. The fluid velocity relative to the solid $\mathbf{V}_{fs}$ and Darcy velocity **V** are related to each other by

$$\mathbf{V} = \theta\, \mathbf{V}_{\mathbf{fs}} \tag{A49}$$

Applying the Gaussian divergence theorem to Equation (A48) and using the fact that *v* is arbitrary, one can obtain the following continuity in differential form:

$$\begin{aligned}\frac{\partial\left(\theta C + \rho_s S\right)}{\partial t} + \nabla \cdot \left[\mathbf{V}_{\mathbf{s}}\left(\theta C + \rho_b S\right)\right] + \nabla \cdot \left(\mathbf{VC}\right) = -\nabla \cdot \mathbf{J} \\ -\left(\theta K_w C + \rho_b K_s S\right) - \lambda\left(\theta C + \rho_b S\right) + m\end{aligned} \tag{A50}$$

The second term of Equation (A50) can be expressed as

$$\nabla \cdot \left[\mathbf{V}_{\mathbf{s}}\left(\theta C + \rho_b S\right)\right] = \nabla\left(\theta C + \rho_b S\right) \cdot \mathbf{V}_{\mathbf{s}} + \left(\theta C + \rho_b S\right)\nabla \cdot \mathbf{V}_{\mathbf{s}} \tag{A51}$$

The first term on the right side of Equation (A51) is the product of two small vectors and will be neglected. If all the displacement of the medium is assumed to be vertical (e.g., vertical consolidation), the solid velocity becomes

$$\nabla \cdot \mathbf{V}_{\mathbf{s}} = \alpha \frac{\partial p}{\partial t} \tag{A52}$$

The surface flux **J** has been postulated to be proportional to the gradient of C (Nguyen et al. 1982) as

$$\mathbf{J} = -\theta \mathbf{D} \cdot \nabla C \tag{A53}$$

$$\theta \mathbf{D} = a_T \left|\mathbf{V}\right| \delta + \left(a_L - a_T\right) \mathbf{VV} / \left|\mathbf{V}\right| + a_m \theta \tau \delta \tag{A54}$$

where

a_T = transverse diffusivity (L)

δ = Kronecker delta tensor

|**V**| = magnitude of the Darcy velocity ***V*** (L/T)

a_L = longitudinal diffusivity (L)

a_m = molecular diffusion coefficient (L^2/T)

τ = tortuosity

Substitution of Equations (A51) through (A54) into Equation (A50) yields

$$\begin{aligned} &\frac{\partial\left(\theta C+\rho_b S\right)}{\partial t}+\nabla\cdot(\mathbf{V}C)-\nabla\cdot(\theta\mathbf{D}\cdot\nabla C)= \\ &\quad -\left(\alpha\frac{\partial p}{\partial t}+\lambda\right)\left(\theta C+\rho_b S\right)-\left(\theta K_w C+\rho_b K_s S\right)+m \end{aligned} \tag{A55}$$

Equation (A55) 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 consolidation and radioactive decay, the fifth term is the degradation rate due to microbial transformation through aqueous and adsorbed phases, and the last term is a source/sink term corresponding to artificial injection and or withdrawal.

Equation (A55) is written in conservative form. Using the advective form is sometimes more appropriate, especially if the finite element method is used to simulate the chemical transport equation. More importantly, an advective form of the transport equations allows one to use the mixed Lagrangian-Eulerian approach, which can better solve advection-dominant transport problems (Yeh and Tripathi 1987). Therefore, an advective form of the transport equation is derived by expanding the advection term and using the continuity equation for water flow. The water flow equation can be rewritten in the following form:

$$\begin{aligned} &\frac{\rho}{\rho_o}F\frac{\partial h}{\partial t}=-\nabla\cdot\left(\frac{\rho}{\rho_o}\mathbf{V}\right)+\frac{\rho^*}{\rho_o}q= \\ &\quad -\frac{\rho}{\rho_o}\nabla\cdot\mathbf{V}-\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)+\frac{\rho^*}{\rho_o}q \end{aligned} \tag{A56}$$

which is conservation of fluid mass. Substituting Equation (A56) into (A55) and performing the necessary manipulation gives

$$\theta\frac{\partial C}{\partial t}+\rho_b\frac{\partial S}{\partial t}+\mathbf{V}\cdot\nabla C-\nabla\cdot\left(\theta\mathbf{D}\cdot\nabla C\right)=$$

$$-\left(\alpha'\frac{\partial h}{\partial t}+\lambda\right)\left(\theta C+\rho_b S\right)-\left(\theta K_w C+\rho_b K_s S\right)+ \tag{A57}$$

$$m-\frac{\rho^*}{\rho}qC+\left(F\frac{\partial h}{\partial t}+\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)-\frac{\partial\theta}{\partial t}\right)C$$

Equation (A57) involves two unknowns C and S, so constitutive relationships must be posed. For the present model, the following empirical relationships are used:

$$S=K_dC \qquad \text{for linear isotherm} \tag{A58}$$

$$S=\frac{s_{max}KC}{1+KC} \qquad \text{for Langmuir isotherm} \tag{A59}$$

$$S=KC^n \qquad \text{for Freundlich isotherm} \tag{A60}$$

where

K_d = distribution coefficient (L^3/M)

S_{max} = maximum concentration of the medium in the Langmuir nonlinear isotherm

K = coefficient in the Langmuir or Freundlich nonlinear isotherm

n = power index in the Freundlich nonlinear isotherm.

In order to use the Lagrangian-Eulerian approach, Equation (A57) is rewritten as:

a. For Linear Isotherm model:

$$\left(\theta+\rho_b K_d\right)\frac{D_{v_d}C}{Dt}=\nabla\cdot\left(\theta\mathbf{D}\cdot\nabla C\right)-\left(\alpha'\frac{\partial h}{\partial t}+\lambda\right)\left(\theta C+\rho_b S\right)-$$
$$\left(\theta K_w C+\rho_b K_s S\right)+m-\frac{\rho^*}{\rho}qC+ \tag{A61}$$
$$\left(F\frac{\partial h}{\partial t}+\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)-\frac{\partial\theta}{\partial t}\right)C$$

$$\mathbf{V}_d^*=\frac{\mathbf{V}}{\theta+\rho_b K_d} \tag{A62}$$

b. For Langmuir and Freundlich models:

$$\theta\frac{D_{v_f}C}{Dt}+\rho_b\frac{dS}{dC}\frac{\partial C}{\partial t}=\nabla\cdot\left(\theta\mathbf{D}\cdot\nabla C\right)-$$
$$\left(\alpha'\frac{\partial h}{\partial t}+\lambda\right)\left(\theta C+\rho_b S\right)-\left(\theta K_w C+\rho_b K_s S\right)+ \tag{A63}$$
$$m-\frac{\rho^*}{\rho}qC+\left(F\frac{\partial h}{\partial t}+\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)-\frac{\partial\theta}{\partial t}\right)C$$

$$\mathbf{V}_f^*=\frac{\mathbf{V}}{\theta} \tag{A64}$$

where

$$\frac{D_{v_d}(\ )}{Dt}\text{ and }\frac{D_{v_f}(\ )}{Dt}$$

denote the material derivatives of ( ) with respect to time in reference to the retarded velocity $\mathbf{V}_d$ and fluid velocity $\mathbf{V}_\mathbf{f}$, respectively.

To complete the description of the hydrological transport as given by Equations (A61) or (A63), initial and boundary conditions must be specified in accordance with dynamic and physical considerations. It will be assumed that initially the dissolved concentrations are known throughout the region of interest, that is,

$$C = C_i\left(x, y, z\right) \quad \text{in} \quad R \tag{A65}$$

where C_i is the initial concentration and *R* is the region of interest. Initial concentrations for the dissolved concentrations may be obtained from field measurements or by solving the steady-state version of Equation (A61) or (A63) with time-invariant boundary conditions.

The specification of boundary conditions is a difficult and intricate task in transport modeling. From the dynamic point of view, a boundary segment may be classified as either flow-through or impervious. From a physical point of view, it is a soil-air interface, soil-soil interface, or soil-water interface. From the mathematical point of view, it may be treated as a Dirichlet boundary on which the total analytical concentration is prescribed, a gradient flux boundary on which the flux due to the gradient of total analytical concentration is known, or a flux boundary on which the total flux is given. An even more difficult mathematical boundary is the variable boundary conditions on which the boundary conditions are not known a priori but are themselves the solution to be sought. In other words, on the mathematically variable boundary, either gradient flux or flux 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; Yeh and Ward 1981).

Whatever point of view is chosen, all boundary conditions eventually must be transformed into mathematical equations for quantitative simulations. Thus, we will specify the boundary conditions from the mathematical point of view in concert with dynamic and physical considerations. The boundary conditions imposed on any segment of the boundary are taken to be either Dirichlet, gradient flux, flux, or variable. Thus, the global boundary may be split into four parts, B_d, B_n, B_c, and B_v, denoting Dirichlet, gradient flux, flux, and variable boundaries, respectively. The conditions imposed on the first three types of boundaries are given as:

a. Prescribed concentration (Dirichlet) boundary conditions:

$$C = C_d(x_b, y_b, z_b, t) \quad \text{on} \quad B_d \tag{A66}$$

b. Gradient flux boundary conditions:

$$\mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C\right) = q_b(x_b, y_b, z_b, t) \quad \text{on} \quad B_n \tag{A67}$$

c. Flux boundary conditions:

$$\mathbf{n}\cdot\left(\mathbf{V}\mathrm{C}-\theta\mathbf{D}\cdot\nabla \mathrm{C}\right)=\mathrm{q}_{\mathrm{c}}(\mathrm{x}_{\mathrm{b}},\mathrm{y}_{\mathrm{b}},\mathrm{z}_{\mathrm{b}},\mathrm{t}) \quad \text{on} \quad \mathrm{B}_{\mathrm{c}} \tag{A68}$$

where

C_d = concentration on the Dirichlet boundary B_d

(x_b,y_b,z_b) = spatial coordinate on the boundary

n = outward unit vector normal to the boundary

q_n = gradient flux through the gradient flux boundary B_n

q_c = total flux through the flux boundary B_c

The conditions imposed on the variable-type boundary, which is normally the soil-air interface or soil-water interface, are either a zero gradient flux or a given total flux. The former is specified when the water flow is directed out of the region from the faraway boundary, whereas the latter is specified when the water flow is directed into the region. This type of variable condition would normally occur at flow-through boundaries. Written mathematically, the variable boundary condition is given by

$$\begin{aligned}&\mathbf{n}\cdot\left(\mathbf{V}\mathrm{C}-\theta\mathbf{D}\cdot\nabla \mathrm{C}\right)=\mathbf{n}\cdot\mathbf{V}\mathrm{C}_{\mathrm{v}}(\mathrm{x}_{\mathrm{b}},\mathrm{y}_{\mathrm{b}},\mathrm{z}_{\mathrm{b}},\mathrm{t}) \text{ on } \mathrm{B}_{\mathrm{v}} \quad \text{if } \mathbf{n}\cdot\mathbf{V}\le 0\\&\mathbf{n}\cdot\left(-\theta\mathbf{D}\cdot\nabla \mathrm{C}\right)=0 \text{ on } \mathrm{B}_{\mathrm{v}} \qquad \text{if } \mathbf{n}\cdot\mathbf{V}>0\end{aligned} \tag{A69}$$

where C_v is the specified concentration of water through the variable boundary and B_v is the variable boundary.

Appendix B Numerical Formulation

Introduction

The initial-boundary value problem described by the governing equations of the flow and transport modules of FEMWATER along with the boundary conditions cannot, in general, be solved analytically using 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 simpler systems of algebraic equations, there are only two numerical methods that are most common and that can be employed to solve the most basic form of the governing equations. These are the finite difference and finite element methods.

The basic difference between the finite difference and finite element 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 produces solutions only 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:

a. Anisotropy and heterogeneity of aquifers are easily taken care of.

b. Formulation of special formulae to incorporate irregular boundaries is unnecessary.

c. Computer storage and computational time can sometimes be saved because often fewer nodal points are needed to portray the region of interest to the same level of accuracy.

d. Irregular grids for handling different levels of spatial discretization in different sections of the region of interest can be incorporated.

e. The integral formulation used in this method permits the flux types of the boundary conditions to come about naturally (Yeh 1987).

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 (1987).

The flow module of FEMWATER includes three options for solving the finite element equations. In other words, there are three iteration methods for solving the linearized matrix equations: successive point iteration, polynomial preconditioned conjugate gradient, and incomplete Cholesky preconditioned conjugate gradient. Direct elimination methods are not used in this report because they are impractical in dealing with large three-dimensional problems. Because the Newton-Raphson method will yield a nonsymmetric matrix, the Picard method is used to linearize the matrix equation.

To handle a large variety of possible problems, the flow module for FEMWATER contains 12 optional numerical schemes. Specifically, the mixture of schemes includes the combinations of (a) the three options for solving the resulting matrix equation as mentioned in the previous paragraph, (b) two options (lumping and consistent) for handling the mass matrix resulting from the storage term, and (c) two options (time-weighted difference and middifference) for approximating the time derivatives. The theoretical background for (b) and (c) may also be found in any respectable matrix computation book and in Yeh (1991).

The transport module for FEMWATER also includes these 16 options, plus more. While the Galerkin finite element method is used in the flow module, an option of two conventional finite element methods (Galerkin and upstream weighting) or the alternative option of a hybrid Lagrangian-Eulerian finite element method is provided in the transport 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).

Numerical Approximation of the Flow Equations

Spatial discretization with the Galerkin finite element method

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

$$h \approx \hat{h} = \sum_{j=1}^{N} h_j(t) N_j(x, y, z) \tag{B1}$$

where

h_j = amplitude of h at nodal point j

N_j = base function at nodal point j

N = total number of nodes

After defining a residual and forcing the weighted residual to zero, the flow equation, Equation (A28), is approximated as:

$$\begin{aligned} &\sum_{j=1}^{N}\left[\int_R N_i \frac{\rho}{\rho_o} F N_j dR\right]\frac{dh_j}{dt} + \sum_{j=1}^{N}\left[\int_R (\nabla N_i)\cdot \mathbf{K}\cdot(\nabla N_j) dR\right] h_j \\ &\quad = \int_R N_i \frac{\rho^*}{\rho_o} q dR - \int_R (\nabla N_i)\cdot \mathbf{K}\cdot\left(\frac{\rho}{\rho_o}\nabla z\right) dR \\ &\quad + \int_B \mathbf{n}\cdot\mathbf{K}\cdot\left(\nabla h + \frac{\rho}{\rho_o}\nabla z\right) N_i dB \qquad i = 1, \ldots, N \end{aligned} \tag{B2}$$

In matrix form, Equation (B2) is written as:

$$[M]\left\{\frac{dh}{dt}\right\} + [S]\{h\} = \{Q\} + \{G\} + \{B\} \tag{B3}$$

where

{dh/dt} = column vectors containing the values of dh/dt at all nodes

{h} = column vectors containing the values of h at all nodes

[M] = mass matrix resulting from the storage term

[S] = stiff matrix resulting from the action of conductivity

{Q} = load vectors from the internal source/sink

{G} = load vectors from the gravity force

{B} = load vectors from the boundary conditions

Furthermore, the mass matrix, [M], and stiff matrix, [S], are described as:

$$M_{ij} = \sum_{e \in M_e} \int_{R_e} N_\alpha^e \frac{\rho}{\rho_o} F N_\beta^e dR \tag{B4}$$

and

$$S_{ij} = \sum_{e \in M_e} \int_{R_e} \left(\nabla N_\alpha^e\right) \cdot \mathbf{K} \cdot \left(\nabla N_\beta^e\right) dR \tag{B5}$$

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

In addition, the three load vectors, {Q}, {G}, and {B}, are described as:

$$Q_i = \sum_{e \in M_e} \int_{R_e} N_\alpha^e \frac{\rho}{\rho_o} q dR \tag{B6}$$

$$G_i = -\sum_{e \in M_e} \int_{R_e} \left(\nabla N_\alpha^e\right) \cdot \mathbf{K} \cdot \frac{\rho}{\rho_o} \nabla z dR \tag{B7}$$

and

$$B_i = -\sum_{e \in N_{se}} \int_{B_e} N_\alpha^e \mathbf{n} \cdot \left[-K \cdot \left\{ \nabla h + \frac{\rho}{\rho_o} \nabla z \right\} \right] dB \tag{B8}$$

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

In most finite element work, the Darcy velocity components given in Equation (A32) are calculated numerically by taking the derivatives of the simulated h as

$$\mathbf{V} = -\mathbf{K} \cdot \left(\frac{\rho_o}{\rho} \sum_{j=1}^{N} \left(\nabla N_j \right) h_j + \nabla z \right) \tag{B9}$$

This formulation results in a velocity field that 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 Equation (A32); thus one obtains

$$[T]\{V_x\} = \{D_x\} \tag{B10}$$

$$[T]\{V_y\} = \{D_y\} \tag{B11}$$

$$[T]\{V_z\} = \{D_z\} \tag{B12}$$

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

$$T_{ij} = \sum_{e \in M_e} \int_{R_e} N_\alpha^e N_\beta^e dR \tag{B13}$$

$$D_{xi} = -\sum_{e \in M_e} \int_{R_e} N_\alpha^e \mathbf{i} \cdot \mathbf{K} \cdot \left\{ \frac{\rho_o}{\rho} \nabla h + \nabla z \right\} dB \tag{B14}$$

$$D_{yi} = -\sum_{e \in M_e} \int_{R_e} N_\alpha^e \mathbf{j} \cdot \mathbf{K} \cdot \left\{ \frac{\rho_o}{\rho} \nabla h + \nabla z \right\} dB \tag{B15}$$

$$D_{zi} = -\sum_{e \in M_e} \int_{R_e} N_\alpha^e \mathbf{k} \cdot \mathbf{K} \cdot \left\{ \frac{\rho_o}{\rho} \nabla h + \nabla z \right\} dB \tag{B16}$$

where V_x, V_y, and V_z are the Darcy velocity components along the x-, y-, and z-directions, respectively; and **i**, **j**, and **k** are the unit vectors along the x-, y-, and z-coordinates, respectively.

The reduction of the partial differential equation (Equation (A28)) to the set of ordinary differential equations (Equation (B3)) simplifies the evaluation of integrals on the right side of Equations (B4) through (B8) for every element for boundary surface e. The major tasks that remain to be done are the specification of base and weighting functions and the performance of integration to yield the element matrices. Linear hexahedral elements are employed in this documentation.

Base and weighting functions

The construction of base functions for hexahedral elements is best accomplished using the local coordinates (ξ,η,ζ). In the local coordinates, the original hexahedral element is mapped into a cubic whose corners are located at ξ = ±1, η = ±1, and ζ = ±1 as shown in Figure B1.

For trilinear hexahedral elements, the eight base functions are obtained by taking the tensor product of the three base functions of the linear line elements as

$$N_i(\xi,\eta,\zeta) = \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\zeta\zeta_i), \qquad i = 1, 2, \ldots, 8 \tag{B17}$$

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.

Numerical integration

To complete the reduction of the partial differential equation (Equation (B3)) to the ordinary differential equation (Equation (B3)), one has to evaluate the integrals on the right sides of Equations (B4)-(B8) for every element to yield the element mass matrix [M^e] and the stiff element matrix [S^e] as well as the element gravity column vector {G^e}, the source/sink column vector {Q^e}, and the boundary column vector {B^e} as

$$M^e_{\alpha\beta} = \int_{R_e} N^e_\alpha \frac{\rho}{\rho_o} F N^e_\beta dR \tag{B18}$$

$$S^e_{\alpha\beta} = \int_{R_e} \left(\nabla N^e_\alpha\right) \cdot \mathbf{K} \cdot \left(\nabla N^e_\beta\right) dR \tag{B19}$$

$$Q^e_\alpha = \int_{R_e} N^e_\alpha \frac{\rho^*}{\rho_o} q dR \tag{B20}$$

$$G^e_\alpha = -\int_{R_e} \left(\nabla N^e_\alpha\right) \cdot \mathbf{K} \cdot \frac{\rho}{\rho_o} \nabla z dR \tag{B21}$$

and

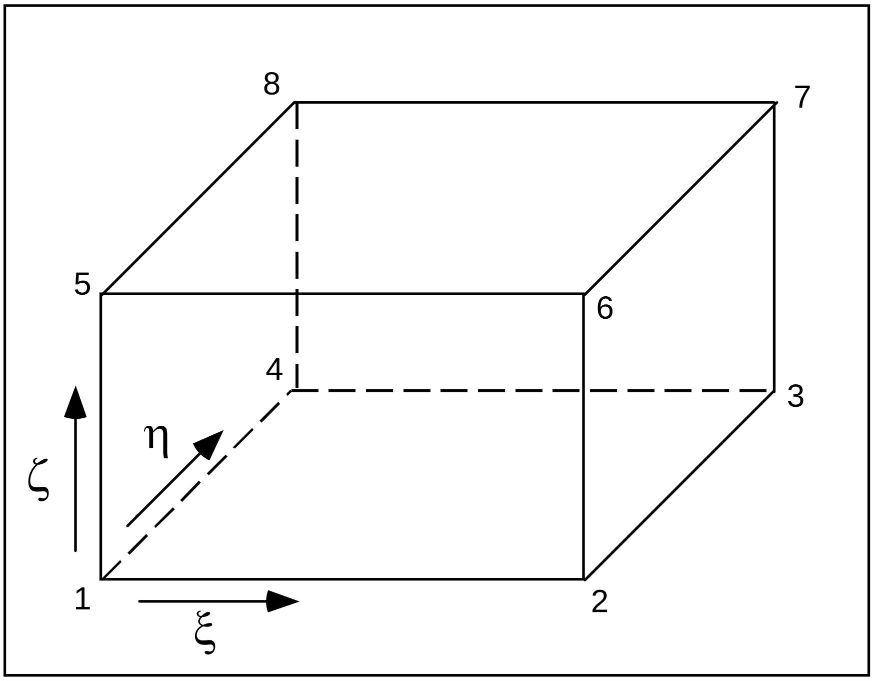


Figure B1. A hexahedral element in local coordinates

$$B_\alpha^e = -\int_{B_e} N_\alpha^e \mathbf{n} \cdot \left[-\mathbf{K} \cdot \left\{ \nabla h + \frac{\rho}{\rho_o} \nabla z \right\} \right] dB \qquad \text{(B22)}$$

Since Equations (B18)-(B22) are written in global coordinates and the base functions are defined in local coordinates, a transformation between the global and local coordinates is needed. The required transformation is obtained via the base functions as

$$x = \sum_{j=1}^{8} x_j N_j(\xi, \eta, \zeta) \qquad \text{(B23)}$$

$$y = \sum_{j=1}^{8} y_j N_j(\xi, \eta, \zeta) \qquad \text{(B24)}$$

$$z = \sum_{j=1}^{8} z_j N_j(\xi, \eta, \zeta), \qquad \text{(B25)}$$

Because the coordinate transformation uses the base functions, the element is termed an “isoparametric” element.

Using the transformation in Equations (B23)-(B25), differentiation of the base function with respect to the global coordinate can be changed to differentiation with respect to the local coordinate by

$$\begin{Bmatrix} \dfrac{\partial N_i}{\partial x} \\ \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \dfrac{\partial N_i}{\partial \xi} \\ \dfrac{\partial N_i}{\partial \eta} \\ \dfrac{\partial N_i}{\partial \zeta} \end{Bmatrix} \tag{B26}$$

$$[J] = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} & \dfrac{\partial z}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial z}{\partial \eta} \\ \dfrac{\partial x}{\partial \zeta} & \dfrac{\partial y}{\partial \zeta} & \dfrac{\partial z}{\partial \zeta} \end{bmatrix} \qquad [J]^{-1} = \text{Inverse of } [J]$$

where [J] is the Jacobian of the transformation and J = |[J]|. In the mean time, a differential volume written in the local coordinate becomes

$$\int_e dR = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} J d\xi d\eta d\zeta \tag{B27}$$

With Equations (B26) and (B27), all the integrals in Equations (B18)-(B21) can be reduced to the following form

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta) J d\xi d\eta d\zeta \tag{B28}$$

the integration of which can easily be carried out with a 2 × 2 × 2 = 8 point Gaussian quadrature. The surface integration of Equation (B22) in three-dimensional space 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 (Figure B2) can be reduced to the area integration. Let I represent the surface integral:

$$I = \int_S F(x, y, z) dS \tag{B29}$$

where the surface S is given by the following equation:

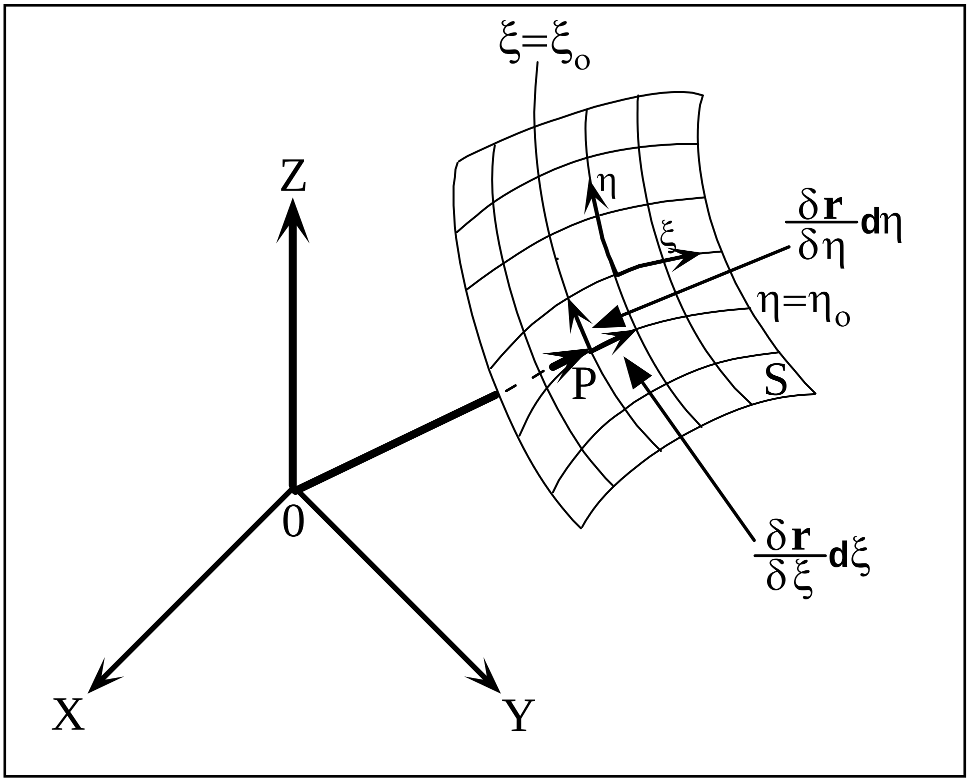


Figure B2. A surface area and its imbedded local coordinates

$$z = f(x, y) \tag{B30}$$

Let P be any point on the surface S with coordinates (x,y,z) or (ξ,η) (Figure B2). Then the vector **r** from O to P is given by

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{B31}$$

The tangent vectors to the coordinate curves $\xi = \xi_o$ and $\eta = \eta_o$ on the surface S are $\partial\mathbf{r}/\partial\eta$ and $\partial\mathbf{r}/\partial\xi$, respectively. The area dS is given by:

$$dS = \left| \frac{\partial \mathbf{r}}{\partial \xi} \times \frac{\partial \mathbf{r}}{\partial \eta} \right| d\xi d\eta \tag{B32}$$

where × represents vector multiplication. But

$$\frac{\partial \mathbf{r}}{\partial \xi} \times \frac{\partial \mathbf{r}}{\partial \eta} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{vmatrix} \tag{B33}$$

so that

$$dS = \sqrt{J_x^2 + J_y^2 + J_z^2 d\xi d\eta} \tag{B34}$$

where

$$J_z = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}, \quad J_y = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{vmatrix}, \quad J_x = \begin{vmatrix} \frac{\partial z}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial z}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}. \tag{B35}$$

Substituting Equation (B34) into Equation (B29) gives

$$\int_S F(x, y, z) dS = \int_{-1}^{1} \int_{-1}^{1} \phi(\xi, \eta) \sqrt{J_x^2 + J_y^2 + J_z^2 d\xi d\eta} \tag{B36}$$

where

$$\phi(\xi, \eta) = F\big(x(\xi, \eta), y(\xi, \eta), z(\xi, \eta)\big) \tag{B37}$$

The surface integral in Equation (B36) can easily be computed by Gaussian quadrature.

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). Mass lumping typically gives better stability but less accuracy than no lumping. However, lumping can give more accurate and stable solutions if it is used in conjunction with the central or backward-difference time marching (Yeh and Ward 1980). Therefore, options are provided for the lumping of the matrix [M]. More explicitly, [M] will be lumped according to:

$$M_{ij} = \sum_{e \in M_e} \left(\sum_{\beta=1}^{8} \int_{R_e} N_\alpha^e \frac{\rho}{\rho_o} F N_\beta^e \, dR \right) \quad \text{if } j = i, \tag{B38}$$

and

$$M_{ij} = 0 \quad \text{if } j \neq i \tag{B39}$$

Finite difference approximation in time

Next, a matrix equation is derived by integrating Equation (B3). For the time integration of Equation (B3), 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 of finite element approximations over the finite difference approximations 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:

$$\begin{aligned} &\frac{[M]}{\Delta t}\left(\{h\}_{t+\Delta t} - \{h\}_t\right) + \omega[S]\{h\}_{t+\Delta t} + (1-\omega)[S]\{h\}_t \\ &\quad = \{Q\} + \{G\} \end{aligned} \tag{B40}$$

where [M], [S], {Q}, and {G} are evaluated at (t + wΔt). In the Crank-Nicolson centered-in-time approach, w = 0.5; in the backward-difference (implicit difference), w = 1.0; and in the forward-difference (explicit scheme), w = 0.0. The central-Nicolson algorithm has a truncation error of O(Δt^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(Δt). 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, Δt. In this middifference method, the recurrence formula is written as:

$$\left(\frac{2}{\Delta t}[M]+[S]\right)\{h\}_{t+\Delta t/2}-\frac{2}{\Delta t}[M]\{h\}_t=\{Q\}+\{G\} \qquad \text{(B41)}$$

and

$$\{h\}_{t+\Delta t}=2\{h\}_{t+\Delta t/2}-\{h\}_t\ , \qquad \text{(B42)}$$

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

Equations (B40) and (B41) can be written as a matrix equation

$$[T]\{h\}=\{Y\}\ , \qquad \text{(B43)}$$

where [T] is the matrix, {h} is the unknown vector to be found and represents the values of discretized pressure field at a new time, and {Y} is the load vector. Take, for example, Equation (B40) with w = 1.0. [T] and {Y} represent the following:

$$[T]=\frac{[M]}{\Delta t}+[S]\, , \qquad \text{(B44)}$$

and

$$\{Y\}=\frac{[M]}{\Delta t}\{h\}_t+\{Q\}+\{G\} \qquad \text{(B45)}$$

where {h} is the vector of the discretized pressure field at a previous time.

Numerical implementation of boundary conditions

The following steps are the incorporation of boundary conditions into matrix equations by the finite element method. For the flux boundary condition given by Equation (A40), simply substitute Equation (A40) into Equation (B22) to yield a boundary-element column vector $\{B_c^e\}$ for a flux segment:

$$\{B_c^e\}=\{q_c^e\} \qquad \text{(B46)}$$

where $\{q_c^e\}$ is the flux boundary vector given by

$$q^e_{c\alpha} = -\int_{B_e} N^e_\alpha \frac{\rho}{\rho_o} q_c dB\,, \qquad \alpha = 1,\ldots,4 \tag{B47}$$

The flux boundary vector represents the normal fluxes through the two nodal points of the segment B_e on B_c. For the gradient flux boundary condition given by Equation (A39), substitute Equation (A39) into Equation (B22) to yield a boundary-element column vector $\{B_n^e\}$ for a gradient flux segment:

$$\left\{B_n^e\right\} = \left\{q_n^e\right\} \tag{B48}$$

where $\{q_{na}^e\}$ is the gradient flux boundary flux vector given by:

$$\left\{q^e_{n\alpha}\right\} = \int_{B_e} \left(N^e_\alpha n \cdot K \cdot \frac{\rho}{\rho_o} \nabla z - N^e_\alpha q_n \right) dB; \quad \alpha = 1,\ldots,4 \tag{B49}$$

which is independent of pressure head.

The implementation of variable-type boundary conditions is more involved. During the iteration of boundary conditions on the variable boundary, one of Equations (A41) through (A45) is used at a node. If either Equation (A42) or (A45) is used, substitute it into Equation (B22) to yield a boundary element column vector $\{B_v^e\}$ for a variable boundary segment:

$$\left\{B^e_v\right\} = \left\{q^e_v\right\} \tag{B50}$$

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

$$\begin{aligned} q^e_{v\alpha} &= -\int_{B_e} N^e_\alpha \frac{\rho}{\rho_o} q_p dB, \quad \text{or} \\ q^e_{v\alpha} &= -\int_{B_e} N^e_\alpha \frac{\rho}{\rho_o} q_e dB; \quad \alpha = 1,\ldots,4 \end{aligned} \tag{B51}$$

Assembling over all gradient flux, flux, and variable boundary segments yields the global boundary column vector {B} as:

$$\left\{B\right\} = \left\{q\right\} \tag{B52}$$

in which

$$\{q\} = \sum_{e\in N_{ne}}\{q_n^e\} + \sum_{e\in N_{ce}}\{q_c^e\} + \sum_{e\in N_{ve}}\{q_v^e\} \qquad \text{(B53)}$$

where N_{ne}, N_{ce}, and N_{ve} are the number of gradient flux boundary segments, flux boundary segments, and variable boundary segments with flux conditions imposed on them, respectively. The boundary flux {B} given by Equations (B52) and (B53) should be added to the right side of Equation (B43).

For the river boundary condition given by Equation (A46), simply substitute Equation (A46) into Equation (B22) to yield a boundary-element column vector $\{B_R^e\}$ and boundary matrix $[B_R^e]$ for a river boundary segment:

$$\{B_R^e\} = \{q_r^e\} \quad \text{and} \quad [B_R^e] = [b_r^e] \qquad \text{(B54)}$$

where $\{q_r^e\}$ and $\{b_r^e\}$ are the contribution of the flux boundary to the right and left sides of the matrix equation, respectively. They are given by

$$q_{r\alpha}^e = \int_{B_e} N_\alpha^e \frac{\rho}{\rho_o}\frac{K_R}{b_R} h_R dB, \quad \alpha = 1,\ldots,4 \ \text{ and}$$

$$b_{r\alpha\beta}^e = \int_{B_e} N_\alpha^e \frac{\rho}{\rho_o}\frac{K_R}{b_R} N_\beta^e dB, \quad \alpha = 1,\ldots,4, \quad \beta = 1,\ldots,4 \qquad \text{(B55)}$$

At nodes where Dirichlet boundary conditions are applied, an identity equation is generated for each node and included in the matrices of Equation (B43). The Dirichlet nodes include the nodes on the Dirichlet boundary and the nodes on the variable boundary to which either Equation (A41), (A43), or (A44) is applied.

After time discretization of Equation (B3) and incorporation of boundary conditions, the following matrix equation is obtained

$$[C]\{h\} = \{R\} \qquad \text{(B56)}$$

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

Solution of the matrix equations

Equation (B56) is in general a banded sparse matrix equation. It may be solved numerically by either direct or iteration methods. In direct methods, a single solution operation sequence is performed. This would result in an exact solution except for round-off error. In this method, one is concerned with the efficiency and magnitude of round-off error associated with the sequence of operations. On the other hand, in an iterative method, one attempts to find a 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 advantages of direct methods are as follows: (a) the efficient computation when the bandwidth of the matrix [C] is small, and (b) the fact that no convergence problem is encountered when the matrix equation is linear or small convergence problems when the mass equation is nonlinear. The 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 advantages of iterative methods are the efficiencies in terms of CPU storage and CPU time when large problems are encountered. Their disadvantages are the requirements 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 direction solution method is not practical. Only iterative methods are implemented in FEMWATER. Four iteration methods are used in solving the linearized matrix equation: (a) block iteration, (b) successive point iteration, (c) polynomial preconditioned conjugate gradient method, and (d) incomplete Cholesky preconditioned conjugate gradient method.

The matrix equation, Equation (B56), 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: (a) the Picard method and (b) the Newton-Raphson method. The Newton-Raphson method has a second order of convergent rate and is very robust. However, the Newton-Raphson 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 give a convergent solution for the nonsymmetric 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, the coefficient matrix [C] is then computed and the linearized matrix equation solved using linear algebra. The new estimate is now obtained by the weighted average of the new solution and the previous estimate:

$$\left\{h^{k+1}\right\} = \omega\left\{h\right\} + \left(1-\omega\right)\left\{h^{k}\right\}, \tag{B57}$$

where

$\{h^{k+1}\}$ = new estimate

$\{h^{k}\}$ = previous estimate

$\{h\}$ = new solution

ω = iteration parameter

The procedure is repeated until the new solution {h} is within a tolerance error. If ω is greater than or equal to 0 but is less than 1, the iteration is underrelaxation. If $\omega = 1$, the method is the exact relaxation. If ω is greater than 1 but less than or equal to 2, the iteration is termed overrelaxation. The underrelaxation should be used to overcome cases of non-convergence or slow convergence rates due to fluctuations rather than due to "divergent" computations. Overrelaxation should be used to speed up the convergence rate when it decreases monotonically.

In summary, there are sixteen optional numerical schemes here to deal with a wide range of problems. These are the combinations of (a) two ways of treating the mass matrix (lumping and no-lumping); (b) two ways of approximating the time derivatives (time-weighting and middifference), and (c) four ways of solving the linearized matrix equation.

Transport Equation

Spatial discretization with the weighted residual finite element method

Let C_j be approximated by a finite element interpolation as

$$C \approx \hat{C} = \sum_{j=1}^{N} C_j(t) N_j(x, y, z) \tag{B58}$$

Substituting Equation (B58) into Equations (A57), (A61), and (A63), and forcing a weighted residual to zero gives the following ordinary differential equations:

a. For the conventional finite element approach:

$$
\begin{aligned}
&\sum_{j=1}^{N}\left[\int_R N_i\left(\theta+\rho_b\frac{dS}{dC}\right)N_j dR\right]\frac{dC_j}{dt}+\sum_{j=1}^{N}\left[\int_R N_i\mathbf{V}\cdot\nabla N_j dR\right]C_j+\\
&\sum_{j=1}^{N}\left[\int_R \nabla N_i\cdot\theta\mathbf{D}\cdot\nabla N_j dR\right]C_j+\sum_{j=1}^{N}\left[\int_R N_i\left(\theta K_w+\rho_b K_s\frac{dS}{dC}\right)N_j dR\right]C_j+\\
&\sum_{j=1}^{N}\left[\int_R N_i(\alpha'\frac{\partial h}{\partial t}+\lambda)\left(\theta+\rho_b\frac{dS}{dC}\right)N_j dR\right]C_j+\\
&\sum_{j=1}^{N}\left\{\int_R N_i\left[\frac{\rho^*}{\rho}q-(\frac{\alpha'\theta}{n}+\beta'\theta)\frac{\partial h}{\partial t}-\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)\right]N_j dR\right\}C_j=\\
&\int_R N_i\left[-(\alpha'\frac{\partial h}{\partial t}+\lambda+K_s)\rho_b\left(S-\frac{dS}{dC}C\right)\right]dR+\\
&\int_R N_i qC_{in}dR+\int_B N_i\,\mathbf{n}\cdot\theta\mathbf{D}\cdot\nabla C dB \qquad i=1,\ldots,N
\end{aligned}
\tag{B59}
$$

b. For the Lagrangian-Eulerian approach with a linear isotherm:

$$
\begin{aligned}
&\sum_{j=1}^{N}\left[\int_R N_i\left(\theta+\rho_b\frac{dS}{dC}\right)N_j dR\right]\frac{D_{V_d}C_j}{dt}+\sum_{j=1}^{N}\left[\int_R \nabla N_i\cdot\theta\mathbf{D}\cdot\nabla N_j dR\right]C_j+\\
&\sum_{j=1}^{N}\left[\int_R N_i\left(\theta K_w+\rho_b K_s\frac{dS}{dC}\right)N_j dR\right]C_j+\\
&\sum_{j=1}^{N}\left[\int_R N_i\left(\lambda+\alpha'\frac{\partial h}{\partial t}\right)\left(\theta+\rho_b\frac{dS}{dC}\right)N_j dR\right]C_j+\\
&\sum_{j=1}^{N}\left\{\int_R N_i\left[\frac{\rho^*}{\rho}q-(\frac{\alpha'\theta}{n}+\beta'\theta)\frac{\partial h}{\partial t}-\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)\right]N_j dR\right\}C_j=\\
&\int_R N_i\left[-(\alpha'\frac{\partial h}{\partial t}+\lambda+K_s)\rho_b\left(S-\frac{dS}{dC}C\right)\right]dR+\\
&\int_R N_i qC_{in}dR+\int_B N_i\,\mathbf{n}\cdot\theta\mathbf{D}\cdot\nabla C dB \qquad i=1,\ldots,N
\end{aligned}
\tag{B60}
$$

c. For the Lagrangian-Eulerian approach with a nonlinear isotherm:

$$
\begin{aligned}
&\sum_{j=1}^{N}\left(\int_R N_i\theta N_j dR\right)\frac{D_{V_f}C_j}{dt} + \sum_{j=1}^{N}\left(\int_R N_i\,\rho_b\frac{dS}{dC}N_j dR\right)\frac{dC_j}{dt} + \\
&\sum_{j=1}^{N}\left(\int_R \nabla N_i\cdot\theta\mathbf{D}\cdot\nabla N_j dR\right)C_j + \sum_{j=1}^{N}\left[\int_R N_i\left(\theta K_w + \rho_b K_s\frac{dS}{dC}\right)N_j dR\right]C_j + \\
&\sum_{j=1}^{N}\left[\int_R N_i\left(\alpha\frac{\partial h}{\partial t}+\lambda\right)\left(\theta+\rho_b\frac{dS}{dC}\right)N_j dR\right]C_j + \\
&\sum_{j=1}^{N}\left\{\int_R N_i\left[\frac{\rho^*}{\rho}q-\left(\frac{\alpha\theta}{n}+\beta\theta\right)\frac{\partial h}{\partial t}-\frac{\rho_o}{\rho}\mathbf{V}\cdot\nabla\left(\frac{\rho}{\rho_o}\right)\right]N_j dR\right\}C_j = \\
&\int_R N_i\left[-\left(\alpha\frac{\partial h}{\partial t}+\lambda+K_s\right)\rho_b\left(S-\frac{dS}{dC}C\right)\right]dR + \\
&\int_R N_i q C_{in} dR + \int_B N_i\,\mathbf{n}\cdot\theta\mathbf{D}\cdot\nabla C dB \qquad i=1,\ldots,N
\end{aligned}
\tag{B61}
$$

Equations (B59)-(B61) are written in matrix form as:

a. For the conventional finite element approach:

$$
[M]\left\{\frac{dC}{dt}\right\}+\left([A]+[D]+[K]\right)\{C\} = \{Q\}+\{B\} \tag{B62}
$$

b. For the Lagrangian-Eulerian approach with the linear isotherm:

$$
[M]\left\{\frac{D_{V_d}C}{dt}\right\}+\left([D]+[K]\right)\{C\} = \{Q\}+\{B\} \tag{B63}
$$

c. For the Lagrangian-Eulerian approach with nonlinear isotherms:

$$
\left([M_1]\left\{\frac{D_{V_f}C}{dt}\right\}+[M_2]\left\{\frac{dC}{dt}\right\}\right)+\left([D]+[K]\right)\{C\} = \{Q\}+\{B\} \tag{B64}
$$

where

{C} = vector whose components are the concentrations at all nodes

{dC/dt} = derivative of {C} with respect to time

[M], [M_1] = mass matrices associated with the material derivative term

[M$_2$] = mass matrix associated with the partial derivative term

[D] = stiff matrix associated with the dispersion term

[A] = stiff matrix associated with the advection term

[K] = stiff matrix associated with all the first-order terms

{Q} = load vector associated with all zero-order derivative terms

{B} = load vector associated with boundary conditions

These matrices and vectors are given as:

$$M_{ij} = \sum_{e \in M} \int_{R_e} N_\alpha^e \left(\theta + \rho_b \frac{dS}{dC} \right) N_\beta^e dR \tag{B65}$$

$$M_{1ij} = \sum_{e \in M} \int_{R_e} N_\alpha^e \theta N_\beta^e dR \tag{B66}$$

$$M_{2ij} = \sum_{e \in M} \int_{R_e} N_\alpha^e \left(\rho_b \frac{dS}{dC} \right) N_\beta^e dR \tag{B67}$$

$$A_{ij} = \sum_{e \in M} \int_{R_e} N_\alpha^e \mathbf{V} \cdot \nabla N_\beta^e dR \tag{B68}$$

$$D_{ij} = \sum_{e \in M} \int_{R_e} \nabla N_\alpha^e \cdot \theta \mathbf{D} \cdot \nabla N_\beta^e dR \tag{B69}$$

$$K_{ij} = \sum_{e \in M} \int_{R_e} N_\alpha^e \begin{bmatrix} \left(\theta K_w + \rho_b K_s \frac{dS}{dC} \right) + \\ (\alpha \frac{\partial h}{\partial t} + \lambda) \left(\theta + \rho_b \frac{dS}{dC} \right) \\ + \frac{\rho *}{\rho} q - (\alpha \frac{\theta}{n} + \beta \theta) \frac{\partial h}{\partial t} \\ - \frac{\rho_o}{\rho} \mathbf{V} \cdot \nabla \left(\frac{\rho}{\rho_o} \right) \end{bmatrix} N_\beta^e \, dR \tag{B70}$$

$$Q_i = \sum_{e \in M} \int_{R_e} N_\alpha^e \left[-(\lambda + \alpha \frac{\partial h}{\partial t} + K_s)\rho_b \left(S - \frac{dS}{dC} C \right) + qC_{in} \right] dR \qquad \text{(B71)}$$

$$B_i = -\sum_{e \in M} \int_{B_e} N_\alpha^e \mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C \right) dB \qquad \text{(B72)}$$

where C_{in} is the concentration of the source.

Base and weighting functions

For the flow case, the weighting functions are taken as the same set as the base functions. However, in the transport formulation using an Eulerian finite element approach, sometimes it is advantageous to use weighting functions which are one or two orders higher than the base functions: (N+1) or (N+2) upstream weighting. This section will present only the N+1 upstream weighting functions. Recently, the N+2 weighting functions have been the subject of several investigations. The success of the N+2 weighting is still under investigation. They will not be included here. First define, for the line element, the following N+1 upstream weighting functions

$$F_1(\xi, \alpha) = N_1(\xi) - \alpha \frac{3}{4}(1+\xi)(1-\xi) \qquad \text{(B73)}$$

$$F_2(\xi, \alpha) = N_2(\xi) + \alpha \frac{3}{4}(1+\xi)(1-\xi) \qquad \text{(B74)}$$

where α is the weighting factor along the line from node 1 to node 2 (Figure B3).

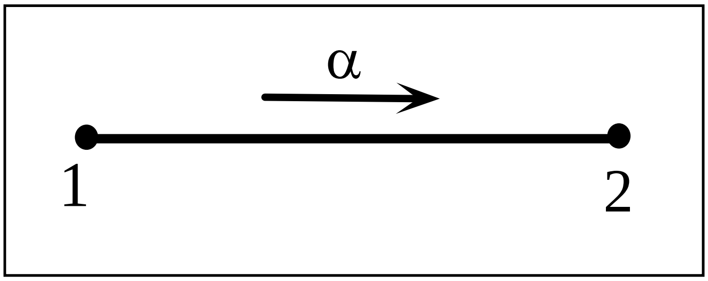


Figure B3. Weighting factor along a line element

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

$$W_1 = F_1(\xi, \alpha_1) F_1(\eta, \beta_1) F_1(\zeta, \gamma_1) \tag{B75}$$

$$W_2 = F_2(\xi, \alpha_1) F_1(\eta, \beta_2) F_1(\zeta, \gamma_2) \tag{B76}$$

$$W_3 = F_2(\xi, \alpha_2) F_2(\eta, \beta_2) F_1(\zeta, \gamma_4) \tag{B77}$$

$$W_4 = F_1(\xi, \alpha_2) F_2(\eta, \beta_1) F_1(\zeta, \gamma_3) \tag{B78}$$

$$W_5 = F_1(\xi, \alpha_3) F_1(\eta, \beta_3) F_2(\zeta, \gamma_1) \tag{B79}$$

$$W_6 = F_2(\xi, \alpha_3) F_1(\eta, \beta_4) F_2(\zeta, \gamma_2) \tag{B80}$$

$$W_7 = F_2(\xi, \alpha_4) F_2(\eta, \beta_4) F_2(\zeta, \gamma_4) \tag{B81}$$

$$W_8 = F_1(\xi, \alpha_4) F_2(\eta, \beta_3) F_2(\zeta, \gamma_3) \tag{B82}$$

where α's, β's, and γ's are the weighting factors along the side given in Figure B4.

Numerical integration

To reduce the partial differential equations, Equations (A57), (A61), and (A63) to ordinary differential equations, Equations (B62)-(B64), one has to evaluate the integrals on the right sides of Equations (B65)-(B72) for every element to yield the element mass matrices $[M^e]$, $[M_1^e]$, and $[M_2^e]$ and the stiff element matrices $[A^e]$, $[D^e]$, and $[K^e]$ as well as the source/sink column vector $\{Q^e\}$ and the boundary column vector $\{B^e\}$ as:

$$M^e_{\alpha\beta} = \int_{R_e} N^e_\alpha \left(\theta + \rho_b \frac{dS}{dC} \right) N^e_\beta dR \tag{B83}$$

$$M^e_{1\alpha\beta} = \int_{R_e} N^e_\alpha \theta N^e_\beta dR \tag{B84}$$

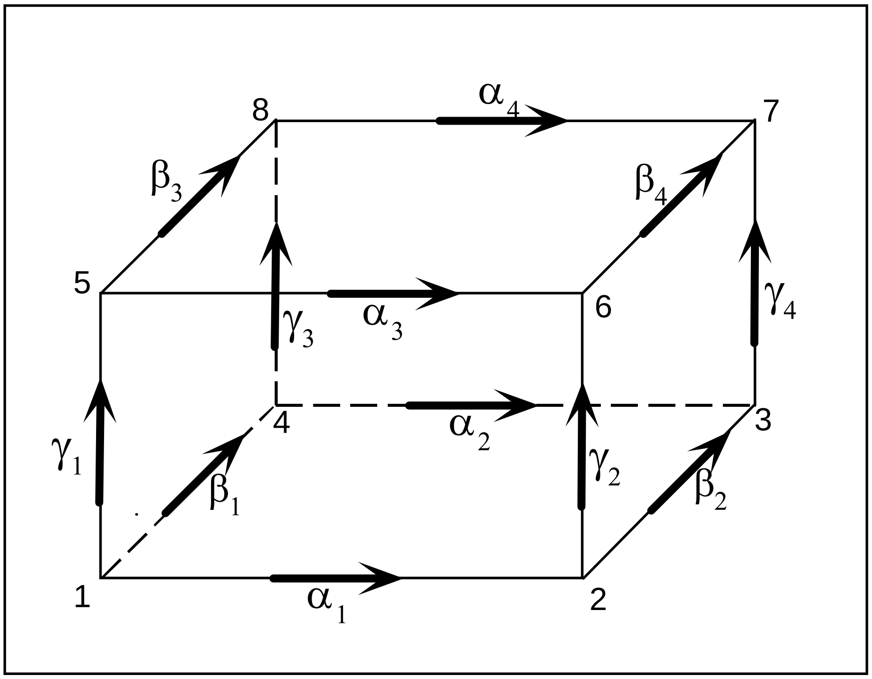


Figure B4. Upstream weighting factors along 12 sides of a hexahedral element

$$M^e_{2\alpha\beta} = \int_{R_e} N^e_\alpha \left(\rho_b \frac{dS}{dC} \right) N^e_\beta dR \tag{B85}$$

$$A^e_{\alpha\beta} = \int_{R_e} N^e_\alpha \mathbf{V} \cdot \nabla N^e_\beta dR \tag{B86}$$

$$D^e_{\alpha\beta} = \int_{R_e} \nabla N^e_\alpha \cdot \theta \mathbf{D} \cdot \nabla N^e_\beta dR \tag{B87}$$

$$K^e_{\alpha\beta} = \int_{R_e} N^e_\alpha \begin{bmatrix} \left(\theta K_w + \rho_b K_s \frac{dS}{dC} \right) \\ + (\alpha \frac{\partial h}{\partial t} + \lambda). \left(\theta + \rho_b \frac{dS}{dC} \right) \\ + \frac{\rho^*}{\rho} q - (\alpha \frac{\theta}{n} + \beta\theta) \frac{\partial h}{\partial t} \\ - \frac{\rho_o}{\rho} \mathbf{V} \cdot \nabla \left(\frac{\rho}{\rho_o} \right) \end{bmatrix} N^e_\beta dR \tag{B88}$$

$$Q_\alpha^e = \int_{R_e} N_\alpha^e \left[-(\lambda + \alpha \frac{\partial h}{\partial t} + K_s)\rho_b \left(S - \frac{dS}{dC} C \right) + qC_{in} \right] dR \tag{B89}$$

$$B_\alpha^e = -\int_{B_e} N_\alpha^e \mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C \right) dB \tag{B90}$$

Following the procedures presented in the section “Numerical integration,” Equations (B83)-(B90) are first transformed in terms of local coordinates. Then the resulting equations are integrated with Gaussian quadrature. The transformation between the global and local coordinates is given by Equations (B23)-(B25) resulting in isoparametric elements. The surface integration from the boundary conditions also follows that presented in “Numerical integration.”

Mass lumping option

As with the solution of the flow equations, a consistent mass matrix or mass lumping option can be used when the Eulerian approach is used. Although a consistent mass matrix option can be used when the hybrid Lagrangian-Eulerian approach is taken, a mass lumping scheme is more appropriate and easier to implement.

Finite difference approximation in time

When the Eulerian approach is taken in approximating the governing equations, either time-weighted difference or middifference can be used as in the section “Finite difference approximation in time.” However, when the Lagrangian-Eulerian approach is taken, the time integration is different from the flow problem. Although there is still a choice of time-weighted difference or middifference, the time-weighted difference scheme is preferred. In the following, the time integration for the Lagrangian-Eulerian approach is demonstrated. As in the time integration of the flow equations, the boundary load vector will be ignored in the time integration of the transport equations in this section. This load vector will be discussed in the next section.

In the Lagrangian-Eulerian approach, Equations (B63) and (B64) are integrated along the characteristic lines. First Equation (B63) will be integrated for the linear isotherm case. Then Equation (B64) will be integrated for nonlinear isotherms. The time-weighted integration of Equation (B63) yields

$$\begin{gathered} \frac{[M]}{\Delta\tau}\left(\left\{C^{n+1}\right\} - \left\{C^*\right\}\right) + \omega\left([D] + [K]\right)\left\{C^{n+1}\right\} + \\ (1-\omega)\left([D] + [K]\right)\left\{C^*\right\} = \{Q\} \end{gathered} \tag{B91}$$

where Δτ is the time-step size (the determination of Δτ will be explained later), $\{C^{n+1}\}$ is the concentration vector containing the concentration at all nodes at the new time n+1, and $\{C^*\}$ is the Lagrangian concentration vector. The Lagrangian concentration $\{C^*\}$ is computed by the backward method of characteristics as follows.

$$\mathbf{x}_\mathbf{i}^* = \mathbf{x}_\mathbf{i} - \int_t^{t+\Delta t} \mathbf{V}_\mathbf{d} dt \qquad C_i^* = \sum_j C_j(t) N_j(\mathbf{x}_\mathbf{i}^*) \tag{B92}$$

where $\mathbf{x}_\mathbf{i}$* (the Lagrangian point) is the location of a fictitious particle originating at time t and arriving at the node $\mathbf{x}_\mathbf{i}$ at time t + Δt. $C_i(t)$ is the concentration value at node j at time t and $N_j(\mathbf{x}_\mathbf{i}*)$ is the interpolation function associated with node j evaluated at the Lagrangian point $\mathbf{x}_\mathbf{i}$*. If $\mathbf{x}_\mathbf{i}$* is located in the interior of the region of interest, Δτ in Equation (B91) is defined as

$$\Delta\tau = \Delta t \tag{B93}$$

If Δ**x*** is located outside the region of interest, a $\Delta\tau(\mathbf{x}_\mathbf{i}*)$ as a function of $\mathbf{x}_\mathbf{i}$* must be found such that

$$\mathbf{x}_\mathbf{i}^* = \mathbf{x}_\mathbf{i} - \int_t^{t+\Delta\tau(\mathbf{x}_\mathbf{i}^*)} \mathbf{V}_\mathbf{d} dt \tag{B94}$$

will be located on the boundary. Thus, it can be seen that Δτ is less than or equal to Δt.

For the nonlinear isotherm case, Equation (B64) can be integrated to yield

$$\left(\frac{[M_1]}{\Delta\tau} + \frac{[M_2]}{\Delta t}\right)\{C^{n+1}\} + \omega([D]+[K])\{C^{n+1}\} = \left(\frac{[M_1]}{\Delta\tau}\{C^*\} + \frac{[M_2]}{\Delta t}\{C^n\}\right) - (1-\omega)([D]+[K])\{C^*\} + \{Q\} \tag{B95}$$

The computation of Δt and the Lagrangian concentrations C^* in Equation (B95) follows Equations (B92)-(B94) but with $\mathbf{V}_\mathbf{d}$ replaced by $\mathbf{V}_\mathbf{f}$.

Numerical implementation of boundary conditions

To incorporate the boundary conditions, the right side of Equation (B90) must be evaluated for every boundary face B_e to yield the load vector $\{B^e\}$:

$$B_\alpha^e = -\int_{B_e} N_\alpha^e \mathbf{n} \cdot \left(-\theta \mathbf{D} \cdot \nabla C\right) dB, \qquad \alpha = 1, \ldots, 4 \tag{B96}$$

For the gradient flux boundary condition given by Equation (A67), simply substitute Equation (A67) into Equation (B90) to yield a boundary-element column vector $\{B_n^e\}$ for a gradient flux face:

$$\left\{B_n^e\right\} = \left\{q_n^e\right\} \tag{B97}$$

where $\{q_n^e\}$ is the gradient flux boundary flux vector given by

$$q_{n\alpha}^e = -\int_{B_e} N_\alpha^e q_N dB, \qquad \alpha = 1, \ldots, 4 \tag{B98}$$

This gradient flux boundary flux vector represents the normal fluxes through the two nodal points of the face B_e on B_n.

For the flux boundary condition given by Equation (A68), Equation (B90) may be rewritten in the following form:

$$B_\alpha^e = -\int_{B_e} N_\alpha^e \mathbf{n} \cdot \left(\mathbf{V}C - \theta \mathbf{D} \cdot \nabla C\right) dB + \int_{B_e} N_\alpha^e \mathbf{n} \cdot \mathbf{V} C dB, \qquad \alpha = 1, \ldots, 4 \tag{B99}$$

The concentration on the boundary segment B_e can be approximated by

$$C = \sum_{\beta=1}^{4} C_\beta N_\beta^e \tag{B100}$$

Substituting Equations (A68) and (B100) into Equation (B99) gives a boundary-element column vector $\{B_c^e\}$ for a flux face:

$$\left\{B_c^e\right\} = \left\{q_c^e\right\} + \left[V_c^e\right]\left\{C\right\} \tag{B101}$$

in which the flux boundary vector $\{q_c^e\}$ and the flux boundary matrix $[\boldsymbol{V}_c^e]$ from the normal velocity component are given by

$$\left\{q_{c\alpha}^{e}\right\} = -\int_{B_e} N_{\alpha}^{e} q_c dB, \quad \alpha = 1,\ldots,4$$
$$V_{c\alpha\beta}^{e} = \int_{b_e} N_{\alpha}^{e} \mathbf{n} \cdot \mathbf{V} N_{\beta}^{e} dB, \quad \alpha = 1,\ldots,4 \quad \text{and} \quad \beta = 1,\ldots,4 \tag{B102}$$

Faces on which the variable boundary conditions are imposed are the flow-through boundaries on which the flow direction is not known a priori. When the flow is directed into the region, flux boundary conditions will be used. The boundary element column vector $\{B_v^e\}$ for a variable boundary segment can be obtained similarly to $\{B_c^e\}$:

$$\left\{B_v^e\right\} = \left\{q_v^e\right\} + \left[V_v^e\right]\left\{C\right\} \tag{B103}$$

in which the variable boundary flux vector $\{q_v^e\}$ and the variable boundary matrix $[V_V^e]$ from the normal velocity component are given by:

$$\left\{q_{v\alpha}^{e}\right\} = -\int_{B_e} N_{\alpha}^{e} \left(\mathbf{n} \cdot \mathbf{V}\right) C_{in} dB, \quad \alpha = 1,\ldots,4$$
$$V_{v\alpha\beta}^{e} = \int_{b_e} N_{\alpha}^{e} \mathbf{n} \cdot \mathbf{V} N_{\beta}^{e} dB, \quad \alpha = 1,\ldots,4 \quad \text{and} \quad \beta = 1,\ldots,4 \tag{B104}$$

where C_{in} is the total dissolved concentration of the incoming fluid. When the flow is directed out from the region, both $\{q_v^e\}$ and $[V_V^e]$ are set equal to 0.

Assembling over all gradient flux, flux, and variable boundary segments, the global boundary column vector $\{B\}$ is obtained as:

$$\left\{B\right\} = \left\{q\right\} + \left[V\right]\left\{C\right\} \tag{B105}$$

in which

$$\left\{q\right\} = \sum_{e \in N_{ne}} \left\{q_n^e\right\} + \sum_{e \in N_{ce}} \left\{q_c^e\right\} + \sum_{e \in N_{ve}} \left\{q_v^e\right\}$$
$$\left[V\right] = \sum_{e \in N_{ce}} \left\{V_c^e\right\} + \sum_{e \in N_{ve}} \left\{V_v^e\right\} \tag{B106}$$

where N_{ne}, N_{ce}, and N_{ve} are the number of gradient flux, flux, and variable boundary segments, respectively.

At nodes where Dirichlet boundary conditions are applied, an identity equation is generated for each node and included in the matrices of Equation

(B91) for the case of linear isotherms or Equation (B95) for the case of nonlinear isotherms. The detailed method of applying this type of boundary condition can be found in Wang and Connor (1975).

Boundary conditions need to be implemented in the computation of the Lagrangian concentrations $\{C^*\}$. Gradient flux boundary conditions are normally applied to the boundary when flow is directed out from the region of interest. On the gradient flux boundary, the backtracking would locate x_i* in the interior of the domain; hence the Lagrangian concentration at the ith gradient flux boundary node is simply computed via interpolation. On the Dirichlet boundary nodes, the Lagrangian concentration is simply set to the specified value.

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, the concentration of incoming fluid is specified. An intermediate concentration C^{**} is calculated according to

$$C_i^{**} = \frac{\int\limits_{B_v^-} N_i V_n C_{in} dB}{\int\limits_{B_v^-} N_i V_n dB} , \qquad \text{(B107)}$$

where C_i** is the concentration due to the boundary source at the boundary node i, V_n is the normal vertically integrated Darcy’s velocity, and C_{in} is the concentration of the incoming fluid.

Flux boundary conditions are normally applied to the boundary where flow is directed into the region, where the material flux of incoming fluid is specified. The intermediate concentration is thus calculated according to

$$C_i^{**} = \frac{\int\limits_{B_c} N_i q_c dB}{\int\limits_{B_c} N_i V_n dB} , \qquad \text{(B108)}$$

where C_i** is the concentration due to flux fluxes at the boundary node i, V_n is the normal Darcy’s velocity, and q_c is the flux of the incoming fluid.

The Lagrangian concentration is obtained by using the value C_i** and C_i^n (the concentration at previous time-step) as follows:

$$C_i^* = \frac{\sum_{j=1}^{N} \int\limits_B N_i \theta N_j C_j^{**} dB + \sum_{j=1}^{N} \int\limits_B N_i \rho_b K_d N_j C_j^n dB}{\int\limits_B N_i \left(\theta + \rho_b K_d\right) dB} \quad \text{linear isotherm (B109)}$$

$$C_i^* = C_i^{**} \quad \text{for the nonlinear isotherm} \tag{B110}$$

Solution of the matrix equations

Because the Lagrangian-Eulerian approach results in a symmetric positive definite matrix, the system of algebraic equations can be solved by any of the four options: block iteration, successive point iteration, polynomial preconditioned conjugate gradient, and incomplete Cholesky preconditioned conjugate gradient methods. For the Eulerian approach, however, the block iteration and successive point iteration methods are the preferred choice for solving the matrix equation. When the advection transport is dominant, the two basic iteration methods with underrelaxation are very effective in reducing the number of iterations required for a convergent solution (Yeh 1985).

Appendix C Output File Formats

Introduction

The output from FEMWATER consists of a printed output file and a number of solution files. The printed output file is a text file listing a summary of the input, iteration and convergence data, and solution summaries. The solution files are used for post-processing or as initial conditions for subsequent runs. The content and format of the solution files are described in this appendix.

There are five solution files that can be output from FEMWATER: pressure head, nodal boundary fluid flux, nodal moisture content, velocity, and nodal concentration. All of the files can be output in either binary or text format. The options for selecting which files are to be output and which format is used are specified on the OC4 card in the model file (see section, “Save options (OC4),” in main text).

Data Set Files

The various output files are all written using the standard Department of Defense Groundwater Modeling System (GMS) data set file format. The data set file formats are two of the standard file formats used by the GMS. Multiple data sets, including both scalar and vector, can be included in one file. However, the data set files used by FEMWATER for input and output are assumed to contain one data set per file.

With the scalar data sets, one scalar value is listed per node, in sequential order based on the node ID’s. If the data set is transient, a complete set of scalar values is listed for each time-step. With vector data sets, the x-, y-, and z-components of the vector (velocity in this case) are listed on a node-by-node basis.

Data set files can be saved in either text or binary format. The binary format results in smaller files and can result in large reductions in the disk space required, particularly when doing transient analyses on large meshes.

Text Data Set File Format

A summary of the text version of the scalar data set file format is shown in Figure C1.

```
DATASET                              /* File type identifier */
OBJTYPE  type                        /* Type of objects */
BEGSCL                               /* Beginning of scalar data set */
ND numdata                           /* Number of data values */
NC numcells                          /* Number of cells or elements */
NAME "name"                          /* Data set name */
TS istat time                        /* Time-step of the following data */
stat_1
stat_2
.
stat_numcells
val_1
val_2
.
val_numdata
/* Repeat TS card for each time-step */
ENDDS                                /* End of data set */
BEGVEC                               /* Beginning of vector data set */
ND numdata                           /* Number of data values */
NC numcells                          /* Number of cells or elements */
NAME "name"                          /* Data set name */
TS istat time                        /* Time-step of the following data */
stat_1
stat_2
.
stat_numcells
v_x1 v_y1 v_z1
v_x2 v_y2 v_z2
.
 v_xnumdata v_ynumdata v_znumdata
/* Repeat TS card for each time-step */
ENDDS                                /* End of data set */
/* Repeat BEGSCL and BEGVEC sequences for each data set. */
```

Figure C1. Text data set file format

The first line of the file is a card without any fields that serves as a file type identifier.

Card Type	**DATASET**
Description	File type identifier.
Required	YES

The next line is an identifier that tells GMS which of the objects the data set is associated with. For FEMWATER, the object type should always be three-dimensional (3-D) mesh.

Card Type	**OBJTYPE**		
Description	Identifies the type of objects that the data sets in the file are associated with.		
Required	YES		
Format	OBJTYPE type		
Sample	OBJTYPE tin		
Field	*Variable*	*Value*	*Description*
1	type	tin mesh2d grid2d scat2d **mesh3d** grid3d scat3d	TINs. 2-D meshes. 2-D grids. 2-D scatter points. **3-D meshes.** 3-D grids. 3-D scatter points.

To begin a data set, either a BEGSCL or BEGVEC card is required, depending on the type of data set.

Card Type	**BEGSCL**
Description	Marks the beginning of a set of cards defining a scalar data set.
Required	YES

Card Type	**BEGVEC**
Description	Marks the beginning of a set of cards defining a vector data set.
Required	YES

The pair of lines are the ND and NC cards. These cards are used to specify the number of nodes and elements in the mesh.

Card Type	**ND**		
Description	The number of data values that will be listed per time-step. This number should correspond to the number of nodes for a 3-D mesh.		
Required	YES		
Format	ND numdata		
Sample	ND 10098		
Field	*Variable*	*Value*	*Description*
1	numdata	+	The number of items. At each time-step, numdata values are listed.

Card Type	**NC**		
Description	This number should correspond to the number of elements for a 3-D mesh.		
Required	YES		
Format	NC numcells		
Sample	NC 3982		
Field	*Variable*	*Value*	*Description*
1	numcells	+	The number of elements.

The next line is for the name of the data set.

Card Type	**NAME**		
Description	The name of the data set.		
Required	YES		
Format	NAME “name”		
Sample	NAME “Total head”		
Field	*Variable*	*Value*	*Description*

1	“name”	str	The name of the data set, in double quotes.

A data set can contain multiple solutions, each solution representing a complete set of nodal values at a particular time-step. The TS card is used to list a time value and the corresponding set of scalar or vector values. If the solution is steady state, only one TS card is used and the time value is set to 0.0. If the solution is transient, the TS card is repeated once for each time-step.

Card Type	**TS (SCALAR)**		
Description	Defines a set of scalar values associated with a time-step. Should be repeated for each time-step.		
Required	YES		
Format	TS istat time stat1 stat2 . statnumcells val1 val2 . valnumdata		
Sample	TS 0 12.5 34.5 74.3 58.4 72.9		
Field	*Variable*	*Value*	*Description*
1	istat	0 1	Use status flags from previous time-step. For first time-step, this value indicates that all cells are active. Status flags will be listed.
2	time	+	The time-step value. This number is ignored if there is only one time-step.
	stat	0 1	Inactive. Active. One status flag should be repeated per line for each cell or element. These flags are included only when ISTAT = 1.
	val	±	The scalar values, one per line.

Card Type	**TS (VECTOR)**		
Description	Defines a set of vector values associated with a time-step. Should be repeated for each time-step.		
Required	YES		
Format	TS istat time stat1 stat2 . statnumcells vx1 vy1 vz1 vx2 vy2 vz2 . vxn vyn vzn		
Sample	TS 0 12.5 34.5 74.4 634.4 74.3 643.4 636.3 58.4 745.4 346.3 72.9 734.3 345.3		
Field	*Variable*	*Value*	*Description*
1	istat	0 1	Use status flags from previous time-step. For first time-step, this value indicates that all cells are active. Status flags will be listed.
2	time	+	The time-step value. This number is ignored if there is only one time-step.
	stat	0 1	Inactive. Active. One status flag should be repeated per line for each cell or element. These flags are included only when ISTAT = 1.
	vx vy vz	±	The vector values, one set per line.

Each data set should be terminated with an ENDDS card.

Card Type	**ENDDS**
Description	Signals the end of a set of cards defining a data set.
Required	YES

Binary Data Set File Format

Data set files can also be written in binary format. The binary format is patterned after the text format in that information is written to the file in groups or “cards.” There are a few additional cards in the binary version, but most of the cards are identical to cards in the text version except that the card identifiers are not written with character strings. Rather, they are written with integer codes.

The cards in the binary data set file are as follows:

Card Type	**VERSION**
Card ID	3000
Description	File type identifier.
Required	Yes

Card Type	**OBJTYPE**			
Card ID	100			
Description	Identifies the type of objects that the data sets in the file are associated with.			
Required	YES.			
Field	*Variable*	*Size*	*Value*	*Description*
1	id	4 byte int	1	TINs.
			2	Boreholes.
			3	2-D meshes.
			4	2-D grids.
			5	2-D scatter points.
			6	**3-D meshes.**
			7	3-D grids.
			8	3-D scatter points.

Card Type	**SFLT**			
Card ID	110			
Description	Identifies the number of bytes that will be used in the remainder of the file for each floating point value (4, 8, or 16).			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	sflt	4 byte int	+	4, 8, or 16

Card Type	**SFLG**			
Card ID	120			
Description	Identifies the number of bytes that will be used in the remainder of the file for status flags (1, 2, or 4).			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	sflg	4 byte int	+	1, 2, or 4

Card Type	**BEGSCL**
Card ID	130
Description	Marks the beginning of a set of cards defining a scalar data set.
Required	YES

Card Type	**BEGVEC**
Card ID	140
Description	Marks the beginning of a set of cards defining a vector data set.
Required	YES

Card Type	**NUMDATA**			
Card ID	170			
Description	The number of data values that will be listed per time-step. This number should correspond to the number of nodes.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	numdata	4 byte int	+	The number of items. At each time-step, numdata are listed.

Card Type	**NUMCELLS**			
Card ID	180			
Description	This number should correspond to the number of elements.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	numcells	4 byte int	+	The number of elements or cells.

Card Type	**NAME**			
Card ID	190			
Description	The name of the data set.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	name	80 bytes	str	The name of the data set. Use one character per byte. Mark the end of the string with the \0 character.

Card Type	**TS (SCALAR)**			
Card ID	200			
Description	Defines a set of scalar or vector values associated with a time-step. Should be repeated for each time-step.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	istat	SFLG int	0 1	Use status flags from previous time-step. For the first time-step, this value indicates that all cells are active. Status flags will be listed.
2	time	SFLT real	+	The time-step value. This number is ignored if there is only one time-step.
	stat	SFLG int	0 1	Inactive. Active. One status flag should be repeated for each cell or element. These flags are included only when istat = 1.
	val	SFLT real		The scalar values. Repeat numdata times.

Card Type	**TS (VECTOR)**			
Card ID	200			
Description	Defines a set of scalar or vector values associated with a time-step. Should be repeated for each time-step.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	istat	SFLG int	0 1	Use status flags from previous time-step. For the first time-step, this value indicates that all cells are active. Status flags will be listed.
2	time	SFLT real	+	The time-step value. This number is ignored if there is only one time-step.
	stat	SFLG int	0 1	Inactive. Active. One status flag should be repeated for each cell or element. These flags are included only when istat = 1.
	vx vy vz	SFLT real		The vector values. Repeat numdata times.

Card Type	**ENDDS**
Card ID	210
Description	Signals the end of a set of cards defining a data set.
Required	YES