PNNL-SA-11720
March, 1998

RT3D

(version 1.0beta)

A Modular Computer Code for Simulating Reactive Multi-species Transport in 3-Dimensional Groundwater Aquifers

By T. P. Clement

Draft report (version date, February 17, 1998)
(Please do not quote without author's permission)

Prepared for
The U.S. Department of Energy
Under Contract DE-AC06-76RLO 1830

Pacific Northwest National Laboratory
Richland, Washington 99352.
Operated for the U.S. Department of Energy
By Battelle Memorial Institute.

TABLE OF CONTENTS

Acknowledgments

Funding Support

Funding support for RT3D code development efforts was, in part, provided by the office of Technology Development, within the Department of Energy's Office of Environmental Management, under the "Plume Focus Area" and the "Subsurface Contamination Focus Area." The RT3D code was originally developed for the Remediation Technology Development Forum (RTDF) with the finacial support from a DOE project that provided microbial characterization research and groundwater-modeling support to the bioremediation of chlorinated solvents consortium of the RTDF.

Technical Support

Dr. T. Prabhakar Clement, principal investigator of the project at the Battelle Pacific Northwest National Laboratory, developed the design concepts for the RT3D computer code. Dr. Clement coded this version of RT3D (v1.0) from MT3D_DoD_1.5 (a public domain version of MT3D developed by Dr. Chunmiao Zheng at the University of Alabama). Several researchers within PNNL and other organizations provided technical support during various phases of this project: Dr. Yunwei Sun, Department of Chemical Engineering at Washington State University, provided debugging and coding support for developing a prototype version and a pre-processor, MT2RT; Dr. Brian S. Hooker, Pacific Northwest National Laboratory, provided support for developing reaction expressions for hydrocarbon and chlorinated solvent degradation models; Christian D. Johnson, Pacific Northwest National Laboratory provided documentation and code testing support; Dr. James N. Petersen, Department of Chemical Engineering at Washington State University, provided debugging support for developing a prototype version and MT2RT; Dr. Chunmiao Zheng, Department of Geology, University of Alabama, provided support for solving MT3D (DoD_1.5) interfacing problems. In addition, Dr. Clement also benefited from the helpful suggestions from several other researchers— suggestions of Dr.

Ashok Chilakapati (at PNNL), Dr. Rodney Skeen (at PNNL), and Dr. Norman Jones (at BYU) are particularly acknowledged.

NOTICE:

For Details Contact:

T. Prabhakar Clement, Ph.D., P.E.
Senior Development Engineer
Mail stop P7-41, Pacific Northwest National Laboratory
Richland, WA – 99352.
tp.clement@pnl.gov

For recent updates:

http://etd.pnl.gov:2080/bioprocess/rt3d.htm
http://www.ecgl.byu.edu/gms/gms.html

INTRODUCTION

RT3D is a reactive transport code that solves multi-species contaminant transport in three-dimensional saturated groundwater systems. RT3D is a generalized multi-species version of the U.S. EPA code, MT3D (Zheng, 1990). The current version RT3D uses the advection and dispersion solvers from the DOD_1.5 (1997) version of MT3D. As with MT3D, RT3D also requires the USGS groundwater flow code MODFLOW for computing spatial and temporal variations in groundwater head distribution.

The RT3D code was originally developed to support the contaminant transport modeling efforts at a natural attenuation demonstration site. As a research tool, RT3D has been used for modeling several laboratory and pilot-scale bioremediation experiments. The performance of the code has been previously tested against various other numerical and analytical solutions (Clement et al., 1998; Sun and Clement, 1998). The code is also currently being used to model several sets of field-scale natural attenuation data.

The RT3D code is unique in that it includes an implicit reaction solver that makes the code sufficiently flexible for simulating various types of chemical and microbial reaction kinetics. RT3D v1.0 supports seven pre-programmed reaction modules that can be used for simulating different types of reactive contaminants including BTEX and chlorinated solvents such as PCE and TCE. In addition, other reactive transport systems can also be simulated by using a user-defined reaction module.

This report describes the mathematical details of the RT3D computer code and its input/output data structure. It is assumed that the user is familiar with the basics of groundwater flow and contaminant transport mechanics. In addition, RT3D users are expected to have some experience in using the MODFLOW and MT3D computer codes and must be familiar with their input/output data structure.

MATHEMATICAL MODEL

Governing Equations

The general macroscopic equations describing the fate and transport of aqueous- and solid-phase species, respectively, in multi-dimensional saturated porous media is written as:

$$\frac{\partial C_k}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial C_k}{\partial x_j} \right) - \frac{\partial}{\partial x_i}\left(v_i C_k \right) + \frac{q_s}{\phi} C_{s_k} + r_c - r_a + r_d, \text{ where, } k = 1,2,..m \tag{1}$$

$$\frac{d\tilde{C}_{im}}{dt} = \tilde{r}_c + r_a - r_d, \text{ where, } im = 1,2,...,(n-m) \tag{2}$$

where n is the total number of species, m is the total number of aqueous-phase (mobile) species (thus, n minus m is the total number of solid-phase or immobile species), C_k is the aqueous-phase concentration of the kth species [ML^{-3}], $\tilde{C}_{im}$ is the solid-phase concentration of the *im*th species [MM^{-1}], D_{ij} is the hydrodynamic dispersion coefficient [L^2T^{-1}], v is the pore velocity [LT^{-1}], q_s is the volumetric flux of water per unit volume of aquifer representing sources and sinks [T^{-1}], C_s is the concentration of source/sink [ML^{-3}], r_c is the reaction rate that describes the mass of the species removed or produced per unit volume per unit time [ML^3T^{-1}], $\tilde{r}_c$ is the reaction rate at the solid phase [$MM^{-1}T^{-1}$], and r_a and r_d, respectively, are attachment (or adsorption) and detachment (or desorption) rates that describe the kinetic exchange of the transported species between aqueous and solid phases [$ML^{-3}T^{-1}$].

Saturated groundwater flow velocities are calculated from the hydraulic-head values that are computed by solving a three-dimensional groundwater flow model. The flow equations used are (Zheng, 1990):

$$S_s \frac{\partial h}{\partial t} = \frac{\partial}{\partial x_i}\left(K_{ii} \frac{\partial h}{\partial x_i} \right) + q_s \tag{3}$$

$$v_i = -\frac{K_{ii}}{\phi}\frac{\partial h}{\partial x_i} \tag{4}$$

where h is the hydraulic head [L], S_s is the specific storage coefficient [L^{-1}], and K_{ii} are the principal components of the hydraulic conductivity tensor [LT^{-1}] (non-principal components are assumed to be zero), and ϕ is the soil porosity. Solution of the flow model, with appropriate boundary and initial conditions, is accomplished using the USGS flow code MODFLOW (McDonald and Harbaugh, 1988).

Numerical Solution Procedure

RT3D code was developed to solve the multi-species reactive transport equations (1) and (2). The code utilizes a reaction Operator-Split (OS) numerical strategy to solve any number of coupled transport equations [of the form (1) and (2)]. Previously, Walter et al. (1994) have successfully used a similar OS approach to solve multi-component transport with geochemical reactions. Clement et al. (1996a) used the OS strategy to solve a biologically reactive flow problem in a radial system. Valocchi and Malmstead (1992) and Kaluarachchi and Morshed (1995) have noted that splitting the reaction terms using the standard OS strategy may have numerical limitations. They recommended an improved alternating OS strategy that may yield more accurate numerical results. However, Barry et al. (1995) states that the improvement provided by the alternating OS may not be applicable for multi-component nonlinear problems. They also demonstrated the efficiency of the standard OS approach by solving a two species reactive transport problem. In this work, we use a standard OS strategy, similar to the one used by Zheng (1990), to develop a general numerical solution scheme for solving the coupled partial/ ordinary differential equations (1) and (2).

Employing the OS strategy, first the mobile species transport equation (1) is divided into four distinct equations: the advection equation,

$$\frac{\partial C}{\partial t} = -\frac{\partial(v_i C)}{\partial x_i} \tag{5}$$

dispersion equation,

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial C}{\partial x_j}\right) \tag{6}$$

source/sink-mixing equation,

$$\frac{\partial C}{\partial t} = \frac{q_s}{\phi}C_s \tag{7}$$

and, reaction equation,

$$\frac{dC}{dt} = r \tag{8}$$

where, the term r represents all possible reaction terms that appear in a typical mobile-species transport equation. Note in a typical immobile species equation [of the form (2)], the advection, dispersion, and source-sink mixing terms are zero and only the reaction term exists.

The USEPA code MT3D uses a similar operator-split approach to solve a mathematical model of the form equation (1) that describes single-species transport with first-order reaction (Zheng, 1990). MT3D is coded into four different packages, each of which is used to sequentially solve the advection equation (5), dispersion equation (6), source-sink mixing equation (7), and single-species reaction equation (8). The advection package has the method of characteristics, modified method of characteristics, hybrid method of characteristics, and upstream finite-difference solution schemes (Zheng, 1990). The dispersion and source-sink mixing packages use explicit finite-difference approximations. RT3D utilizes all the original transport routines available in MT3D for solving the advection-dispersion problem. The routines are invoked by RT3D multiple times to compute the transport of multiple mobile species. The original MT3D-reaction solver (an explicit solver) was replaced by a new reaction module that has an improved implicit reaction solver. Appropriate code modifications were also implemented to input multiple initial-and-boundary conditions and the multi-species reaction information.

The logical steps involved in the numerical solution procedure are illustrated in Figure 1. As shown in the figure, the use of the operator-split strategy helps solve the complex coupled reactive transport system in a modular fashion. The solution algorithm initially solves the advection, dispersion and source-sink mixing steps for all mobile components for a transport time step Δt. The length of transport step is restricted by the constraints posed from the advection, dispersion, and source-sink mixing solvers (Zheng, 1990). After solving the transport, the coupled reaction equations are solved implicitly by using multiple reaction-time steps. Computation of the required reaction-time step sizes to precisely integrate the differential equations is automated within the differential equation solver. Use of this modular operator-split approach for solving the reactive transport problem facilitates representation of different contaminant transport systems through a set of pre-programmed reaction packages. Further, other user-defined reaction kinetics may also be easily incorporated into the simulator as a run-time library.

Numerical Solution Procedure...

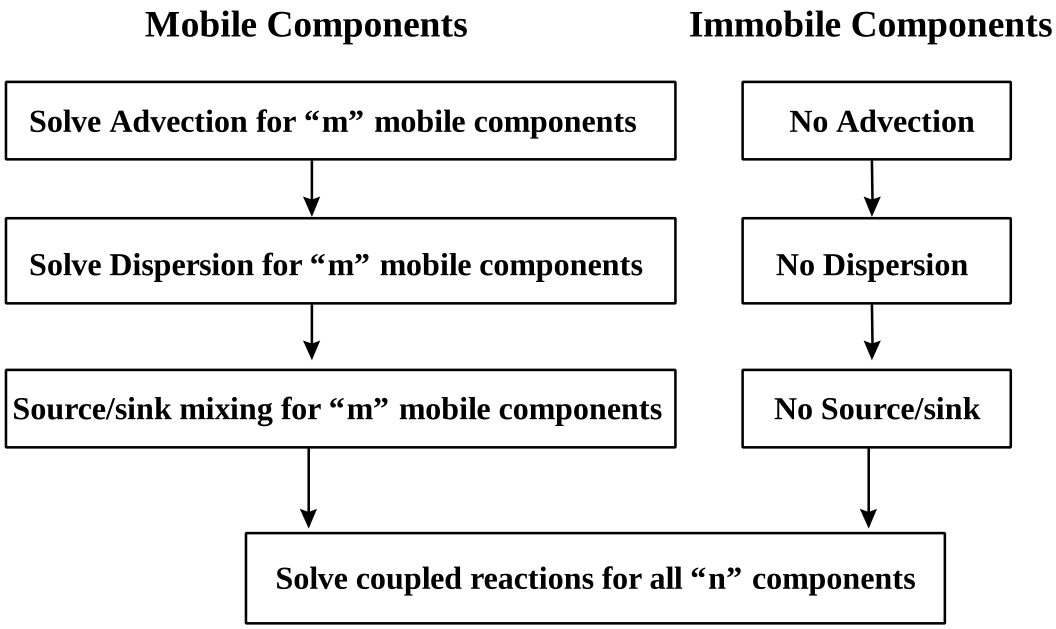

Figure 1. Block Diagram Illustrating the Numerical Solution Scheme

INPUT/ OUTPUT DATA STRUCTURE

Since the RT3D code evolved from the MT3D code, the input data structure of RT3D is very similar to the MT3D input data structure. The following sections discuss only the differences in the RT3D and MT3D input data files. It is assumed that the user is familiar with the MT3D input data structure given in Zheng (1990). Example RT3D data sets are given in a later section.

Differences between RT3D and MT3D Input Data

Basic Transport Package (BTN) file

In a *MT3D-BTN* file, multi-species initial and boundary are defined. Two new variables, "ncomp" and "mcomp", are used to define the type and number of species in the transport problem. The value of ncomp represents the total number of species and the value of mcomp represents the number of mobile species. The difference between the two values (ncomp - mcomp) represents the number of immobile species in the problem. It is important to note that all immobile species should be defined at the end. For example, if one wants to simulate the reactive transport of dissolved hydrocarbon, sorbed hydrocarbon, dissolved oxygen, and attached bacteria, assign species numbers 1 and 2 for dissolved hydrocarbon and oxygen, and assign the numbers 3 and 4 for sorbed-hydrocarbon and bacteria. In this example ncomp should be set to 4 and mcomp should be set to 2.

To implement appropriate multi-species changes to a *MT3D-BTN* file, the record A3 (described in page 6-10 of MT3D manual) needs to be modified. The differences between *MT3D-* and *RT3D-BTN* files are given below:

CODE	***RECORD***	***INPUTS***
MT3D	A3	NLAY,NROW,NCOL,NPER (Format 4I10)

RT3D	A3	NLAY,NROW,NCOL,NPER,NCOMP,MCOMP (Format 6I10)

The second change involves assignment of initial conditions. In a *RT3D-BTN* file, “ncomp” number of starting concentrations should be specified to define the initial condition of all the species. To implement this, the record A13 (described in page 6-13 of MT3D manual) must be modified:

CODE	***RECORD***	***INPUTS***
MT3D	A13	SCONC Format: RARRAY
RT3D	A13	SCONC(1) SCONC(2) SCONC(3)…. So on… until SCONC(ncomp) Format for each array: RARRAY

All mobile species starting concentration arrays should be described first, and the immobile species starting concentration arrays should be described later.

Source-Sink Mixing (SSM) File

In RT3D source sink mixing (SSM) package, multi-species concentration information should be defined by inputting the CRCH array “mcomp” number of times, one species at a time. The same procedure must be repeated for the evapotranspiration concentration array.

CODE	***RECORD***	***INPUTS***
MT3D	D4	CRCH Format: RARRAY
RT3D	D4	CRCH(1) Format: RARRAY

		CRCH(2) Format: RARRAY CRCH(3) Format: RARRAY.. Similarly, repeat up to CRCH(mcomp), for “mcomp” number of times, to define all mobile species concentrations that enter through recharge.
MT3D	D6	CEVT Format: RARRAY
RT3D	D6	CEVT(1) Format: RARRAY CEVT(2) Format: RARRAY CEVT(3) Format: RARRAY.. Similarly, repeat up to CEVT(mcomp), for “mcomp” number of times, to define all mobile species concentrations that escape via evopotranspiration.

To specify point source sinks, such as wells, rivers and drains, concentrations of all (mcomp number of) species must be specified next to the source type information. To implement this change, the records D4, D6, and D8 need to be modified.

MT3D	D8	KSS, ISS, JSS, CSS, ITYPE Format: I10, I10, I10, F10.0, I10
RT3D	D8	KSS, ISS, JSS, CSS(1), ITYPE, CSS(1), CSS(2), CSS(3), CSS(4)…. CSS(mcomp) to define mcomp number of mobile species concentrations. Note,

CSS(1) is repeated twice to ensure backward compatibility with MT3D.

Format: I10, I10, I10, F10.0, I10, free format

In this version of RT3D, a new point source/sink boundary condition, designated as the constant-concentration condition, can be assigned by setting ITYPE = -1. This boundary condition should be used if it is necessary to assign constant concentrations only to a selected number of species. When using ITYPE = -1, assign concentration values to a desired set of constant concentration species, and assign dummy negative values for all other species. The negative value is a flag used by RT3D to skip assigning the constant-concentration condition. Note the present version of GMS_2.1 (February 98) does not support this new boundary condition.

RT3D Reaction (RCT) File

The input data structure of *RT3D-RCT* file is completely different from the *MT3D-RCT* data structure. This is because, the MT3D code uses an explicit solution scheme to solve a first-order decay equation, whereas, RT3D uses an implicit solution scheme to solve a number of coupled reaction equations. The data structure of *RT3D-RCT* file is described below:

E1: Record: ISOTHM, IREACT, NCRXNDATA, NVRXNDATA, ISOLVER

Format: (5I10)

ISOTHM = 1, Linear adsorption isotherm is simulated
= 2, Freundlich adsorption isotherm is simulated
= 3, Langmuir adsorption isotherm is simulated
= 0, no sorption is simulated

IREACT = if positive, then it corresponds tot that reaction model #
= 0, no reaction is simulated (tracer transport)

NCRXNDATA = number of constant reaction parameter values

NVRXNDATA = number of variable reaction parameter arrays

ISOLVER = Use 1 or 0 to select the implicit solver or no solver options. The implicit solver is the only option currently available for solving kinetic reaction equations. However, to use the instantaneous reaction models #1 and #2, the no solver option should be selected.

Note NCRXNDATA is used to define the number of reaction parameters, which are spatially constant; NVRXNDATA is used to define spatially variable reaction parameters.

E2: ARRAY: RHOB(NLAY) (one value for each layer)
READER: RARRAY
RHOB is the bulk density of the porous medium [ML^{-3}].

Units for bulk density and adsorption constants should be self-consistent to compute a dimensionless retardation parameter.

(Enter E3, if ISOTHM > 0)

E3: ARRAY: SP1(NLAY) (one value for each layer)
(Then, repeat SP1 arrays for each mobile component)
READER: RARRAY
SP1 is the first sorption constant.

For linear sorption, SP1 is the distribution coefficient K_d [L^3m^{-1}]. Other details are described in the MT3D manual.

(Enter E4, if ISOTHM > 0)

E4: ARRAY: SP2(NLAY) (one value for each layer)
(Then, repeat SP1 arrays for each mobile component)
READER: RARRAY

SP2 is the second sorption constant. Other details are described in the MT3D manual.

(Enter E5, if ISOLVER = 1)

E5: ARRAYS: atol(ncomp), rtol(ncomp)

Format: Free format, two entries per line; total of “ncomp” number of entries.

The parameters atol and rtol are used by the differential equations solver to control convergence error while solving the reaction model. The differential equation solver will attempt to control a vector e = e(i), which defines estimated local error (or convergence level) in the variable y(i) according to the inequality: max-norm of [e(i)/ewt(i)] is less than or equal to 1.0. Here, ewt = (ewt(i)) is a vector of positive error weights and is computed using the formula: ewt(i) = rtol(i)*abs(y(i)) + atol(i). The following rule of thumb to set atol and rtol values. If m is the number of significant digits required in the solution component y(i), set RTOL(i) = $10^{-(m+1)}$ and set ATOL(i) to a small value at which the absolute value of y(i) is essentially insignificant. Note the values of rtol and atol should always be non-negative.

(Enter E6, if IREACT > 0)

Input all constant reaction parameter values (NCRXNDATA entries); the variable that stores these parameters is: RXNARRAY1(NCRXNDATA)

E6: ARRAY: RXNARRAY1(NCRXNDATA)

Format : Free format, one entry per line with a total of NCRXNDATA number of entries. RXNARRAY1(NCRXNDATA) is the one-dimensional array that stores all constant (spatially invariable) reaction parameters. The type of parameters stored in this array will depend on the type of reaction model used.

Input variable reaction parameters (NVRXNDATA number of arrays); the variable that stores these parameters is: RXNARRAY2(NCOL, NROW, NLAY, NVRXNDATA)

E7: ARRAY: RXNARRAY2(NCOL,NROW) (one array for each layer)
(Then, repeat RXNARRAY2 for NRXNDATA number of times)

READER: DPRARRAY (a new array reader which is exactly similar to the reader RARRAY, but with double precision).

Note, RXNARRAY2(NCOL,NROW,NLAY,NVRXNDATA) is a four-dimensional array that stores all spatially variable reaction kinetic parameters.

RT3D Super File

The RT3D code uses a super file, typically has an extension “*.rts”, to define all input data file names, specie names, and reaction constant names. The file format and an example are given below:

File Format

```
RT3DSUP /* File type identifier */
BTN “filename.btn” /* The Basic Transport package file */
ADV “filename.adv” /* The Advection package file */
DSP “filename.dsp” /* The Dispersion package file */
SSM “filename.ssm” /* The Source/Sink Mixing package file */
RCT “filename.rct” /* The Chemical Reaction package file */
FLO “filename.hff” /* The MODFLOW head and flow file */
CHK “flag” /* Flag for output of heads & flows (T Or F) */
OUT “filename.out”/* The standard ouput file */
CON “filename.con” /* Binary concentration output files for GMS post processing */
Note: filename001.con, filename002.con, etc will be automatically
Created.  UCN file users should use rt3d001.ucn, rt3d002.ucn.. files.
DSS “filename.dss” /* Concentration data set super file for GMS */
MAS “filename.mas” /* The standard mass balance file */
SPC “new_species_name1” specie_ID, specie_type
SPC “new_species_name2” specie_ID, specie_type
```

(Repeat the SPC card for all ncomp number of species. Integer specie_ID is a numerical identifier for species; start with 1 and number up to ncomp. Integer specie_type can either be 1 or 0; 1 means mobile specie and 0 is immobile specie. Note all the mobile species should be named and numbered first before including any immobile species).

Example Data File

```
RT3DSUP
BTN “test.btn”
FLO “test.hff”
ADV “test.adv”
DSP “test.dsp”
SSM “test.ssm”
RCT “test.rct”
CHK Y
OUT “test.out”
CON “test.con”
DSS “test.dss”
MAS “test.mas”
SPC “HC” 1 1
SPC “Oxy” 2 1
SPC “HC_adsorb” 3 1
SPC “Bacteria” 4 0
```

Note in this example, HC_adsorb (adsorbed-phase hydrocarbon) and bacteria (attached-phase bacteria) are immobile species and hence are defined as the last two species.

Details of RT3D Output Files

Similar to MT3D, RT3D will output a standard output file and mass-budget files. RT3D will also write the following sets of concentration output files:

Filenamexxx.ucn: unformatted concentration files; where “xxx” is the component (or specie) number. For example, if “test” is the name of the file then, test001.ucn, test002.ucn.. etc. will be the output concentration files)

Filenamexxx.con: data stored in this file is identical to the data in the ucn file; however, these files are written in a binary format that can be directly imported in to GMS 2.1 for post processing purposes.

RT3Dxxx.mas: mass balance summary files; Caveat: the mass balance summary for immobile species are not complete; therefore, ignore the massbalance result reported for all immobile species.

Except for the GMS-specific Filenamexxx.con files, the format and structure of all other output files are identical to those used in MT3D.

Details of Reaction Models available in RT3D 1.0

The RT3D model always requires a reaction module to define the problem-specific reactions (i.e. how the contaminants react with each other and with the subsurface). In RT3D (v1.0), seven preprogrammed reaction modules and a user-define reaction module are available. The following sections describe the details of all the reaction modules.

Model #1: Instantaneous Aerobic Decay of BTEX

Purpose: To simulate aerobic BTEX degradation using a 2-component, instantaneous reaction model (the reactions simulated are similar to those simulated by BIOPLUME-II (Rifai et al., 1987); also see Clement et al. (1998) for comparison between RT3D and bioplume codes.

Reaction algorithm: This module simulates the instantaneous degradation of fuel hydrocarbons under aerobic conditions:

$$C_6H_6 + 7.5O_2 \rightarrow 6CO_2 + 3H_2O \qquad (9)$$

The following algorithm dictates the instantaneous removal of either contaminant (H) or oxygen (O):

$$H(t+1) = H(t) - O(t)/F \text{ and } O(t+1) = 0\text{, when } H(t) > O(t)/F$$

$$O(t+1) = O(t) - H(t) \bullet F \text{ and } H(t+1) = 0\text{, when } O(t) > H(t) \bullet F$$

where, t refers to a particular time step and F is the stoichiometric ratio. Given this algorithm, either the hydrocarbon or the oxygen concentration in a given grid cell will be reduced to zero at each time step, depending on which component is stoichiometrically limiting in the prior time step.

Details of the Reaction Module:

Total number of components (ncomp) = 2

Mobile component (mcomp) = 2

The component names are: BTEX and Oxygen

NCRXNDATA = 1; NVRXNDATA = 0

Constant#1: Stoichiometric ratio F (oxygen to BTEX) = 3.14.

Note that by changing the value of F, instantaneous reaction between any other two chemicals may be simulated.

Model #2: Instantaneous Degradation of BTEX using Multiple Electron Acceptors

Purpose: To simulate instantaneous biodegradation of BTEX via five different pathways: aerobic respiration, denitrification, iron reduction, sulfate reduction, and methanogenesis.

Reaction Algorithm: The reaction algorithm used is similar to the one used in Model #1, except five different instantaneous degradation pathways are considered here (Rifai et al., 1995; Wiedemeier et al., 1996).

Details of the Reaction Module:
Total number of components (ncomp) = 6
Mobile component (mcomp) = 6
The component names are: BTEX, Oxygen, Nitrate, Fe2+, Sulfate, Methane
NCRXNDATA = 5 or 7; NVRXNDATA = 0 or 2
Constant #1: Max value of Fe2+
Constant #2: Max value of CH4
Other five constants are fixed values (the values assume mg/L basis):
F1: Stochiometric ratio of oxygen (consumed) to BTEX = 3.14
F2: Stochiometric ratio of nitrate (consumed) to BTEX = 4.9
F3: Stochiometric ratio of iron^{2+} (produced) to BTEX = 21.8
F4: Stochiometric ratio of sulfate (consumed) to BTEX = 4.7
F5: Stochiometric ratio of methane (produced) to BTEX = 0.78

Model #3: Kinetic-limited degradation of BTEX using multiple electron acceptors

Purpose: To simulate kinetic-limited biodegradation of BTEX via five different pathways: aerobic respiration, denitrification, iron reduction, sulfate reduction, and methanogenesis (Clement et al. 1997).

Reaction Algorithm: The kinetic model considered here describes rate-limited degradation of hydrocarbon through five distinct degradation pathways. The model describes sequential degradation of fuel hydrocarbons under aerobic, denitrifying, iron-reducing, sulfate-reducing, and methanogenic conditions. The kinetics of hydrocarbon decay are assumed to be first order with respect to hydrocarbon concentration. A Monod-type term is used to account for the presence (or the absence) of various electron acceptors, and an inhibition model is used to describe sequential utilization of various electron acceptors. The following kinetic framework is used to represent degradation of hydrocarbon through different electron acceptor pathways:

$$r_{HC,O_2} = -k_{O_2}[HC]\frac{[O_2]}{K_{O_2}+[O_2]} \tag{10}$$

$$r_{HC,NO_3} = -k_{NO_3}[HC]\frac{[NO_3]}{K_{NO_3}+[NO_3]}\frac{K_{i,O_2}}{K_{i,O_2}+[O_2]} \tag{11}$$

$$r_{HC,Fe^{2+}} = -k_{Fe^{3+}}[HC]\frac{[Fe^{3+}]}{K_{Fe^{3+}}+[Fe^{3+}]}\frac{K_{i,O_2}}{K_{i,O_2}+[O_2]}\frac{K_{i,NO_3}}{K_{i,NO_3}+[NO_3]} \tag{12}$$

$$r_{HC,SO_4} = -k_{SO_4}[HC]\frac{[SO_4]}{K_{SO_4}+[SO_4]}\frac{K_{i,O_2}}{K_{i,O_2}+[O_2]}\frac{K_{i,NO_3}}{K_{i,NO_3}+[NO_3]}\frac{K_{i,Fe^{3+}}}{K_{i,Fe^{3+}}+[Fe^{3+}]} \tag{13}$$

$$r_{HC,CH_4} = -k_{CH_4}[HC]\frac{[CO_2]}{K_{CH_4}+[CO_2]}\frac{K_{i,O_2}}{K_{i,O_2}+[O_2]}\frac{K_{i,NO_3}}{K_{i,NO_3}+[NO_3]}\frac{K_{i,Fe^{3+}}}{K_{i,Fe^{3+}}+[Fe^{3+}]}\frac{K_{i,SO_4}}{K_{i,SO_4}+[SO_4]} \tag{14}$$

where $r_{HC,O2}$ is the rate at which hydrocarbon is destroyed by utilizing oxygen, $r_{HC,NO3}$ is the destruction rate utilizing nitrate, $r_{HC,Fe2+}$ is the destruction rate utilizing Fe^{3+} (or producing Fe^{3+}), $r_{HC,SO4}$ is the destruction rate utilizing sulfate, $r_{HC,CH4}$ is the destruction rate via methane production, $[O_2]$ is oxygen concentration $[ML^{-3}]$, k_{O2} is the first-order rate constant $[T^{-1}]$, Ko_2 is the Monod half saturation constant $[ML^{-3}]$ (note by pre-setting the half-saturation constants to a small value we can simulate zero-order dependency with respect electron donor, and hence a first-order degradation model with respect to hydrocarbon), and Ki,o_2 is the oxygen inhibition constant [ML^{-3}]; similar nomenclature was used for all other reactions.

Since the concentrations of Fe^{3+} and CO_2 are not readily measurable under normal field conditions, these terms were replaced with the "assimilative capacity" for iron reduction and methanogenesis, defined as:

$$[\mathrm{Fe}^{3+}] = [\mathrm{Fe}^{2+}{}_{\mathrm{max}}] - [\mathrm{Fe}^{2+}] \tag{15}$$

$$[\mathrm{MC}] = [\mathrm{CO}_2] = [\mathrm{CH}_{4,\mathrm{max}}] - [\mathrm{CH}_4] \tag{16}$$

where [$Fe^{2+}{}_{max}$] and [$CH_{4,max}$] are the maximum measured aquifer levels of these species that represent the aquifer's maximum capacity for iron reduction and methanogenesis (Wiedeimeir et al., 1996). Note, the concentration of CO_2 used here is the CO_2 evolved while the hydrocarbon is destroyed via methanogenesis (it can also be thought of as the Methanogenic Capacity (MC) of the aquifer). Using these relations, both these destruction processes are related back to field-measurable Fe^{2+} and CH_4 concentration levels.

The total rate of hydrocarbon destruction is given as:

$$\frac{d[\mathrm{HC}]}{dt} = r_{\mathrm{HC,O_2}} + r_{\mathrm{HC,NO_3}} + r_{\mathrm{HC,Fe^{2+}}} + r_{\mathrm{HC,SO_4}} + r_{\mathrm{HC,CH_4}} \tag{17}$$

Rates of electron acceptor utilization are given as the corresponding rate of hydrocarbon destruction multiplied by the appropriate yield coefficients (Y):

$$\frac{d[\mathrm{O}_2]}{dt} = Y_{\mathrm{O_2/HC}}\, r_{\mathrm{HC,O_2}} \tag{18}$$

$$\frac{d[\mathrm{NO}_3]}{dt} = Y_{\mathrm{NO_3/HC}}\, r_{\mathrm{HC,NO_3}} \tag{19}$$

$$\frac{d[Fe^{2+}]}{dt} = -Y_{Fe^{2+}/HC}\ r_{HC,Fe^{2+}} \qquad (20)$$

$$\frac{d[SO_4]}{dt} = Y_{SO_4/HC}\ r_{HC,SO_4} \qquad (21)$$

$$\frac{d[CH_4]}{dt} = -Y_{CH_4/HC}\ r_{HC,CH_4} \qquad (22)$$

The default value set for all the half-saturation constants is 0.5 mg/L, and for all the inhibition constants is 0.01 mg/L. The yield value for $Y_{02/HC}$ is set to 3.14, $Y_{NO3/HC}$ is set to 4.9, $Y_{Fe2+/HC}$ is set to 21.8, $Y_{SO4/HC}$ is set to 4.7 , and $Y_{CH4/HC}$ is set to 0.78. Equations (17) to (22) are coded into the reaction module #3.

Details of the Reaction Module:

Total number of components (ncomp) = 6

Mobile component (mcomp) = 6

The component names are: BTEX, Oxygen, Nitrate, Iron (Fe2+), Sulfate, Methane

NCRXNDATA = 21 or 0; NVRXNDATA = 0 or 21.

Constant #1: Max value of Fe2+ observed in the field (Max_Fe2+)

Constant #2: Max value of Methane observed in the field (Max_Methane)

Constant #3 Hydrocarbon decay rate via aerobic process (khco)

Constant #4 Hydrocarbon decay rate via denitrification (khcn)

Constant #5 Hydrocarbon decay rate via iron reduction (khcFe)

Constant #6 Hydrocarbon decay rate via sulfate reduction (khcs)

Constant #7 Hydrocarbon decay rate via methanogenesis (khcme)

Constant #8 Half saturation constant for oxygen (K_{O2} = 0.5 mg/L)

Constant #9 Half saturation constant for nitrate (K_{NO3} = 0.5 mg/L)

Constant #10 Half saturation constant for Fe^{3+} (K_{Fe3+} = 0.5 mg/L)

Constant #11 Half saturation constant for sulfate (K_{SO4} = 0.5 mg/L)

Constant #12 Half saturation constant for methane (K_{CH4} = 0.5 mg/L)

Constant #13 Inhibition coefficient for oxygen reaction (Ki_{02} = 0.01 mg/L)

Constant #14 Inhibition coefficient for nitrate reaction (Ki_{NO3} = 0.01 mg/L)

Constant #15 Inhibition coefficient for Fe^{3+} reaction ($Ki_{Fe3+} = 0.01$ mg/L)

Constant #16 Inhibition coefficient for sulfate reaction ($Ki_{SO4} = 0.01$ mg/L)

Constant #17 Oxygen yield ($Y_{02/HC} = 3.14$)

Constant #18 Nitrate yield ($Y_{NO3/HC} = 4.9$)

Constant #19 Iron (Fe^{2+}) production yield ($Y_{Fe2+/HC} = 21.8$)

Constant #20 Sulfate yield ($Y_{SO4/HC} = 4.7$)

Constant #21 Methane production yield ($Y_{CH4/HC} = 0.78$)

All the kinetics constants are reported in mg/L basis.

Model #4: Rate-limited Sorption Reactions

Purpose: To simulate mass-transfer-limited sorption reactions. A detailed example for using this reaction model is discussed in Clement et al. (1998).

Reaction Algorithm: When sorption is assumed to be rate-limited, it is necessary to track contaminant concentrations in both mobile (groundwater) and immobile (soil) phases. Following Haggerty and Gorelick's (1994) approach, the fate and transport of a sorbing solute at aqueous and soil phases can be predicted using the following transport equations:

$$\frac{\partial C}{\partial t} + \frac{\rho}{\phi}\frac{\partial \tilde{C}}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial C}{\partial x_j}\right) - \frac{\partial}{\partial x_i}\left(v_i C\right) + \frac{q_s}{\phi}C_s \qquad (23)$$

$$\frac{\rho}{\phi}\frac{\partial \tilde{C}}{\partial t} = \xi\left(C - \frac{\tilde{C}}{\lambda}\right) \qquad (24)$$

where C is the concentration of contaminant in the mobile-phase [ML^{-3}], $\tilde{C}$ is the concentration of the contaminant in the immobile phase (mass of the contaminants per unit mass of porous media, [MM^{-1}]), ρ is the bulk density of the soil matrix, ϕ is the soil porosity, ξ is the first-order, mass-transfer rate parameter [T^{-1}], and λ is the linear partitioning coefficient [L^3M^{-1}].

The mass-transfer model discussed above was set-up as a RT3D (one mobile species and one immobile species) reaction package. After employing reaction-operator splitting, the reaction package for the problem reduces to:

$$\frac{dC}{dt} = -\xi\left(C - \frac{\tilde{C}}{\lambda}\right) \tag{25}$$

$$\frac{d\tilde{C}}{dt} = \frac{\phi\xi}{\rho}\left(C - \frac{\tilde{C}}{\lambda}\right) \tag{26}$$

These two differential equations are coded into the rate-limited reaction module.

Details of the Reaction Module:

Total number of components (ncomp) = 2

Mobile component (mcomp) = 1

The component names are: Aqueous_conc, Solid_conc

NCRXNDATA = 2 or 0; NVRXNDATA = 0 or 2

Constant#1 = Mass transfer coefficient, ξ

Constant#2 = Linear partitioning coefficient, λ (same as K_d)

Model #5: Double Monod Model

Purpose: To simulate the reaction between an electron donor and an electron acceptor mediated by actively growing bacteria cells living in both aqueous and soil phases. Application of different types of double-monod models are discusssed by: Molz et al. (1986); Rifai and Bedient (1990); Tayler and Jaffe (1990); Clement et al. (1996a); Clement et al. (1996b); Reddy et al. (1997); Clement et al. (1997b); and Clement et al. (1998).

Reaction Algorithm: Assuming an equilibrium model for sorption and a Monod kinetic model for biological reactions (Rifai and Bedient 1990; Clement et al. 1996a), the fate and transport of an electron donor (hydrocarbon, for example) in a multi-dimensional saturated porous media can be written as:

$$R_D \frac{\partial [D]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [D]}{\partial x_j} \right) - \frac{\partial}{\partial x_i}\left(v_i [D]\right) + \frac{q_s}{\phi}[D_s] - \mu_m \left([X] + \frac{\rho \tilde{X}}{\phi} \right) \left(\frac{[D]}{K_D + [D]} \right) \left(\frac{[A]}{K_A + [A]} \right) \tag{27}$$

where [D] is the electron donor concentration in the aqueous phase [ML^{-3}], [D_s] is the donor concentration in the sources/sinks [ML^{-3}], [X] is the aqueous phase bacterial cell concentration [ML^{-3}], $\tilde{X}$ is the solid-phase cell concentration (mass of bacterial cells per unit mass of porous media [MM^{-1}]), [A] is the electron acceptor concentration in the aqueous phase [ML^{-3}], R_H is the retardation coefficient of the hydrocarbon, K_D is the half saturation coefficient for the electron donor [ML^{-3}], K_A is the half saturation coefficient for the electron acceptor [ML^{-3}], and μ_m is the contaminant utilization rate [T^{-1}]. The model assumes that the degradation reactions occur only in the aqueous phase, which is usually a conservative assumption.

The fate and transport of the electron donor (oxygen, for example) can be modeled using the equation:

$$R_A \frac{\partial [A]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [A]}{\partial x_j} \right) - \frac{\partial}{\partial x_i}\left(v_i [A]\right) + \frac{q_s}{\phi}[A_s] - Y_{A/D}\, \mu_m \left([X] + \frac{\rho \tilde{X}}{\phi} \right) \left(\frac{[D]}{K_D + [D]} \right) \left(\frac{[A]}{K_A + [A]} \right) \tag{28}$$

where $Y_{A/D}$ is the stoichiometric yield coefficient, and R_A is the retardation coefficient of the electron acceptor.

The fate and transport of bacteria in the aqueous phase can be described using the equation:

$$\frac{\partial[X]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial[X]}{\partial t}\right) - \frac{\partial}{\partial x_i}\left(v_i[X]\right) + \frac{q_s}{\phi}[X_s] - K_{att}[X] + \frac{K_{det}\rho\tilde{X}}{\phi} + Y_{X/D}\,\mu_m[X]\left(\frac{[D]}{K_D+[D]}\right)\left(\frac{[A]}{K_A+[A]}\right) - K_e[X] \tag{29}$$

where K_{att} is the bacterial attachment coefficient $[T^{-1}]$, K_{det} is the bacterial detachment coefficient $[T^{-1}]$, and K_e is the endogenous cell death or decay coefficient $[T^{-1}]$.

The growth of attached-phase bacteria can be described using an ordinary differential equation of the form:

$$\frac{d\tilde{X}}{dt} = \frac{K_{att}\,\phi[X]}{\rho} - K_{det}\tilde{X} + Y_{X/D}\,\mu_m\tilde{X}\left(\frac{[D]}{K_D+[D]}\right)\left(\frac{[A]}{K_A+[A]}\right) - K_e\tilde{X} \tag{30}$$

The conceptual model for representing soil bacteria, implicitly assumed in the above formulation is similar to the macroscopic model described by Baveye and Valocchi (1989); no specific microscopic biomass structure is assumed, and diffusional limitations across biofilm are also neglected. Equations (29) and (30) also assume first-order kinetic expressions for representing the exchange of bacteria cells between aqueous and solid phases (Taylor and Jaffe, 1990; Hornberger et al. 1992; Peyton et al. 1995). More recently, Reddy and Ford (1996) compared equilibrium and kinetic expressions for modeling the microbial exchange process and concluded that kinetic models provide more general description. Permeability and porosity changes caused by bacterial growth are ignored in this formulation. However, if required, macroscopic models for biomass-affected porous-media properties, described in Clement et al. (1996b), may be integrated within this modeling approach.

After reaction-operator splitting, the reaction terms are assembled together and represented by a set of coupled, non-linear differential equations of the form:

$$\frac{d[D]}{dt} = -\frac{\mu_m}{R_D}\left([X] + \frac{\rho\tilde{X}}{\phi}\right)\left(\frac{[D]}{K_D+[D]}\right)\left(\frac{[A]}{K_{[A]}+[A]}\right) \tag{31}$$

$$\frac{d[A]}{dt} = -\frac{Y_{A/D}\,\mu_m}{R_A}\left([X] + \frac{\rho\tilde{X}}{\phi}\right)\left(\frac{[D]}{K_D + [D]}\right)\left(\frac{[A]}{K_A + [A]}\right) \tag{32}$$

$$\frac{d[X]}{dt} = -Y_{X/D}\,\mu_m\left([X] + \frac{\rho\tilde{X}}{\phi}\right)\left(\frac{[D]}{K_D + [D]}\right)\left(\frac{[A]}{K_A + [A]}\right) + \frac{K_{det}\rho\tilde{X}}{\phi} - K_{att}[X] - K_e[X] \tag{33}$$

$$\frac{d\tilde{X}}{dt} = \frac{K_{att}\phi[X]}{\rho} - K_{det}\tilde{X} + Y_{X/D}\,\mu_m\tilde{X}\left(\frac{[D]}{K_D + [D]}\right)\left(\frac{[A]}{K_A + [A]}\right) - K_e\tilde{X} \tag{34}$$

These four equations are coded into the double-monod model reaction module.

Details of the Reaction Module:

Total number of components (ncomp) = 4

Mobile component (mcomp) = 3

The component names are: ED (Electron Donor), EA (Electron Acceptor), AB (Aqueous phase mobile Bacteria), SB (Soil-phase immobile Bacteria).

NCRXNDATA = 8 or 0; NVRXNDATA = 0 or 8

Constant#1 = Specific utilization rate

Constant#2 = Monod half saturation constant for electron donor

Constant#3 = Monod half saturation constant for electron acceptor

Constant#4 = Biomass produced per unit amount of electron donor utilized, $Y_{X/D}$

Constant#5 = Electron acceptor used per unit amount of electron donor utilized, $Y_{A/D}$

Constant#6 = First-order, bacterial death or decay rate, K_{decay}

Constant#7 = First-order, bacterial attachment rate, K_{att}

Constant#8 = First-order, bacterial detachment rate, K_{det}

This module describes a general double Monod model. By setting appropriate yield and kinetic constants, users can model any type of biological systems. Kinetic constants for an aerobic system are given in Clement et al. (1998), and for an anaerbic denitrifying system are given in Clement et al. (1997b). Also see Taylor and Jaffe (1990); Hornberger et al. (1992); Zysset et al. (1994); and Reddy and Ford (1996).

Model #6: Sequential Decay Reactions

Purpose: To simulate the transport couple by sequential degradation chain reactions (up to four components). The example considered here focuses on modeling dechlorination of PCE and its daughter products under anaerobic conditions. However, this module can be used to describe any 4-component sequential decay chain reactions. For more details, please refer to a PCE example problem discussed in Clement et al. (1998). Also see Wiedemeier et al. (1997).

Reaction Algorithm: The kinetic reaction equations solved by this model are:

$$\frac{d[A]}{dt} = -K_A[A] \tag{35}$$

$$\frac{d[B]}{dt} = Y_{B/A}K_A[A] - K_B[B] \tag{36}$$

$$\frac{d[C]}{dt} = Y_{C/B}K_B[B] - K_C[C] \tag{37}$$

$$\frac{d[D]}{dt} = Y_{D/C}K_C[C] - K_D[D] \tag{38}$$

Even though this module provides a general description for any 4-component sequential decay model, all the default values given here are specific to PCE dechlorination reactions.

Details of the Reaction Module:

Total number of components (ncomp) = 4

Mobile component (mcomp) = 4

The component names are: PCE, TCE, DCE, and VC

NCRXNDATA = 7 or 0; NVRXNDATA = 0 or 7

Constant#1 = PCE first-order degradation rate (K_A)

Constant#2 = TCE first-order degradation rate (K_B)

Constant#3 = DCE first-order degradation rate (K_C)

Constant#4 = VC first-order degradation rate (K_D)

Constant#5 = Yield, $Y_{B/A}$ ($Y_{tce\_pce} = 0.792$)

Constant#6 = Yield, $Y_{C/B}$ ($Y_{dce\_tce} = 0.738$)

Constant#7 = Yield, $Y_{D/C}$ ($Y_{vc\_dce} = 0.644$)

The yield values are in mg/L basis; so to be consistent, the user must use mg/L unit for all concentrations.

Model #7: Aerobic/ anaerobic Model for PCE/TCE Degradation

Purpose: To simulate degradation of PCE/TCE and their degradation products via both aerobic and anaerobic pathways.

Reaction Algorithm: The conceptual model used for modeling all chlorinated solvents degradation reactions, mediated by aerobic and anaerobic dechlorination processes, is described in the figure below:

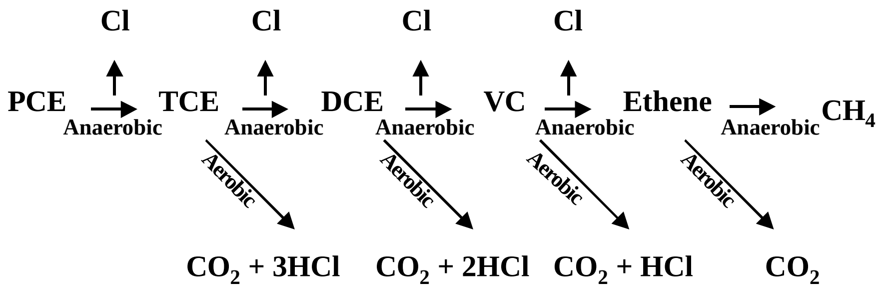

Figure 2. Conceptual Biochemical Model

Assuming first-order biodegradation kinetics, transport and transformation of PCE, TCE, DCE, VC, ETH, and Cl can be simulated by solving the following set of partial differential equations:

$$R_P \frac{\partial [PCE]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [PCE]}{\partial x_j} \right) - \frac{\partial \left(v_i [PCE] \right)}{\partial x_i} + \frac{q_s}{\phi}[PCE]_s - K_P[PCE] \tag{39}$$

$$R_T \frac{\partial [TCE]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [TCE]}{\partial x_j} \right) - \frac{\partial \left(v_i [TCE] \right)}{\partial x_i} + \frac{q_s}{\phi}[TCE]_s + Y_{T/P} K_P[PCE] - K_{T1}[TCE] - K_{T2}[TCE] \tag{40}$$

$$R_D \frac{\partial [DCE]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [DCE]}{\partial x_j} \right) - \frac{\partial \left(v_i [DCE] \right)}{\partial x_i} + \frac{q_s}{\phi}[DCE]_s + Y_{D/T} K_T[TCE] - K_{D1}[DCE] - K_{D2}[DCE] \tag{41}$$

$$R_V \frac{\partial [VC]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [VC]}{\partial x_j} \right) - \frac{\partial \left(v_i [VC] \right)}{\partial x_i} + \frac{q_s}{\phi}[VC]_s + Y_{V/D} K_{D1}[DCE] - K_{V1}[VC] - K_{V2}[VC] \tag{42}$$

$$R_E \frac{\partial [ETH]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [ETH]}{\partial x_j} \right) - \frac{\partial \left(v_i [ETH] \right)}{\partial x_i} + \frac{q_s}{\phi}[ETH]_s + Y_{E/V} K_{V1}[VC] - K_{E1}[ETH] - K_{E2}[ETH] \tag{43}$$

$$\begin{aligned} R_C \frac{\partial [Cl]}{\partial t} = {} & \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial [Cl]}{\partial x_j} \right) - \frac{\partial \left(v_i [Cl] \right)}{\partial x_i} + \frac{q_s}{\phi}[Cl]_s + Y1_{C/P} K_{P1}[PCE] + Y1_{C/T} K_{T1}[TCE] \\ & + Y1_{C/D} K_{D1}[DCE] + Y1_{C/V} K_{V1}[VC] + Y2_{C/T} K_{T2}[TCE] + Y2_{C/D} K_{D2}[DCE] \\ & + Y2_{C/V} K_{V2}[VC] \end{aligned} \tag{44}$$

where [PCE], [TCE], [DCE], [VC], [ETH], and [Cl] represent contaminant concentrations of various species [mg/L]; K_P, K_{T1}, K_{D1}, and K_{V1}, and K_{E1} are first-order anaerobic degradation rates [day^{-1}]; K_{T2}, K_{D2}, and K_{V2}, and K_{E2} are first-order aerobic degradation rates [day^{-1}]; R_P, R_T, R_D, R_V, R_E, and R_C are retardation factors; $Y_{T/P}$, $Y_{D/T}$, $Y_{V/D}$, and $Y_{E/V}$ are chlorinated compound yields under anaerobic reductive dechlorination conditions– their values are: 0.79, 0.74, 0.64 and 0.45, respectively; $Y1_{C/P}$, $Y1_{C/T}$, $Y1_{C/D}$, and $Y1_{C/V}$ are yield values for chloride under anaerobic conditions– their values are: 0.21, 0.27, 0.37, and 0.57, respectively; and $Y2_{C/T}$, $Y2_{C/D}$, and $Y2_{C/V}$ are yield values for chloride under aerobic conditions– their values are: 0.81, 0.74, and 0.57, respectively. The yield values are estimated from the reaction stochiometery and molecular weights; for example, anaerobic degradation of one mole of PCE would yield one mole of TCE, therefore $Y_{T/P}$ = molecular weight of TCE/molecular weight of PCE (131.4/165.8 = 0.79).

Note the reaction models presented above assume that the biological degradation reactions only occur in the aqueous phase, which is a conservative assumption.

Using the reaction operator-split strategy, the biological reaction kinetics included in the transport equations (39) to (44) are separated and assembled together as a set of ordinary differential equations:

$$\frac{d[PCE]}{dt} = \frac{K_P[TCE]}{R_P} \tag{45}$$

$$\frac{d[TCE]}{dt} = \frac{Y_{T/P}K_P[PCE] - K_{T1}[TCE] - K_{T2}[TCE]}{R_D} \tag{46}$$

$$\frac{d[DCE]}{dt} = \frac{Y_{D/T}K_T[TCE] - K_{D1}[DCE] - K_{D2}[DCE]}{R_D} \tag{47}$$

$$\frac{d[VC]}{dt} = \frac{Y_{V/D}K_{D1}[DCE] - K_{V1}[VC] - K_{V2}[VC]}{R_V} \tag{48}$$

$$\frac{d[ETH]}{dt} = \frac{Y_{E/V}K_{V1}[VC] - K_{E1}[ETH] - K_{E2}[ETH]}{R_E} \tag{49}$$

$$\begin{aligned}\frac{d[Cl]}{dt} = \{&Y1_{C/P}K_{P1}[PCE] + Y1_{C/T}K_{T1}[TCE] + Y1_{C/D}K_{D1}[DCE] + Y1_{C/V}K_{V1}[VC] \\ &+ Y2_{C/T}K_{T2}[TCE] + Y2_{C/D}K_{D2}[DCE] + Y2_{C/V}K_{V2}[VC]\}/R_C\end{aligned} \tag{50}$$

These five equations are coded into the reaction module #7.

Details of the Reaction Module:

The component names are: PCE, TCE, DCE, VC, ETH, and Cl

NCRXNDATA = 9 or 0; NVRXNDATA = 0 or 9

Constant#1 = Anaerobic decay rate for PCE, K_P

Constant#2 = Anaerobic decay rate for TCE, K_{T1}

Constant#3 = Aerobic decay rate for TCE, K_{T2}

Constant#4 = Anaerobic decay rate for DCE, K_{D1}

Constant#5 = Aerobic decay rate for DCE, K_{D2}

Constant#6 = Anaerobic decay rate for VC, K_{V1}

Constant#7 = Aerobic decay rate for VC, K_{V2}

Constant#8 = Anaerobic decay rate for ETH, K_{E1}

Constant#9 = Aerobic removal rate for ETH, K_{E2}

All the yield values are fixed internally; to be consistent, user must use mg/L unit for all specie concentrations.

Model #8: Under construction

Will be included in the next version.

Model #9: Under construction

Will be included in the next version.

Model #10: User-Defined Reaction Module

Purpose: To simulate reactive transport based on user-defined reaction kinetics. This is the most versatile option available in RT3D. Using this option, one can describe and solve any type of kinetic-limited reactive transport problem. The reaction information is input via a Fortran-90 subroutine, which should be compiled as a dynamic link library (DLL) using either the Microsoft FORTRAN Power station 4.0 or the Digital DVF Fortran compiler.

Details of the Reaction Module:

Total number of components (ncomp) are defined by the user

Mobile component (mcomp) are defined by the user

NCRXNDATA should be defined by the user

NVRXNDATA should be defined by the user.

The user-defined reactions are specified in a subroutine called rxns(x, y, z…) and stored in a file (say rxns.f). Note a user-defined reaction subroutine must be named as rxns(), and it should be compiled as a DLL “rxns.DLL”. See example #2, and also refer to Clement and Jones (1998) for further details about using the user-reaction option.

MODFLOW to be used with RT3D

The present version of RT3D uses the MODFLOW that is compatible with GMS (available from the GMS web page at: www.ecgl.byu.edu/gms/gms.html). If the user is not using GMS as your preprocessor, he/she should assemble all MODFLOW input-file names into a super file to run the GMS version of MODFLOW. Here is an example for the *GMS-MODFLOW* super file:

```
MODSUP
IJK -y +x -z
LIST   26 “flow.out”
BAS1    1 “flow.bas”
BCF3   11 “flow.bcf”
OUT1   10 “flow.oc”
HEAD  -30 “flow.hed”
PCG2   12 “flow.pcg”
WEL1   13 “flow.wel”
MT3D  -29 “flow.hff”
```

If your MODFLOW input files were created by a pre-processor other than GMS then create a super file, similar to the one described above, using an ASCII text editor and include appropriate MODFLOW input-file names. Save the data in a file say “flow.mfs”. From a DOS window run the modflow executable downloaded from the GMS web site. The code will prompt for a superfile name. Type the filename (flow.mfs) and press enter to run MODFLOW.

EXAMPLE PROBLEMS

Two example problems are discussed below to demonstrate the use of the RT3D code. The data files used in these test examples can be downloaded from the web site:

http://etd.pnl.gov:2080/bioprocess/rt3d.html. GMS users should follow the instructions in the tutorial document: RT3D Tutorials for GMS Users by Clement and Jones (1998).

Example 1: Using Reaction Model #1

Problem Statement

The problem we will be solving in this example is shown in the following figure:

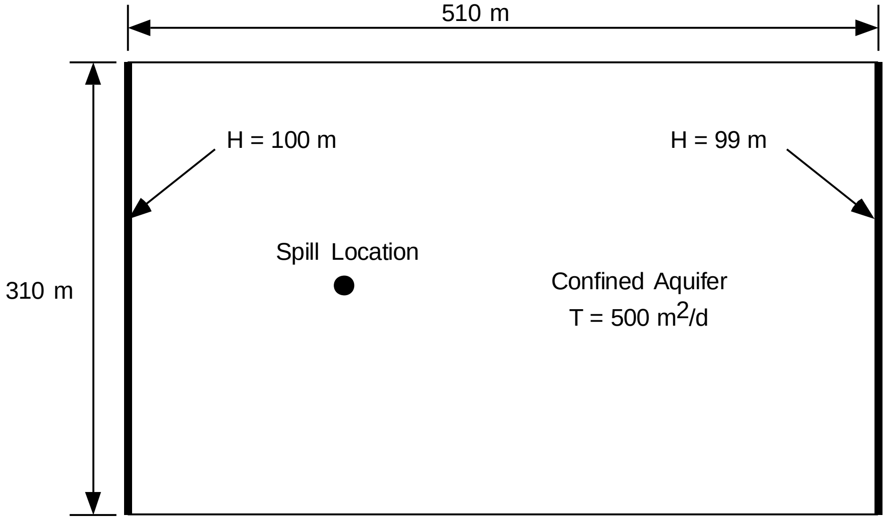

Figure 3. Example Problem to be Modeled using RT3D

The site is a 510 m X 310 m section of a confined aquifer (of 10 m thick) with a flow gradient from left to right (1/500m). The hydraulic conductivity of the aquifer is 50 m/day. An underground storage tank is leaking fuel hydrocarbon contaminants at 2 m^3/day at a cell centered at 155m x 155m (location is shown in the figure). Concentration of hydrocarbons is 1000 mg/L and the spill may be assumed to be devoid of oxygen. Details of the aquifer hydrology and geometry are given below. The aquifer is assumed have a uniform porosity of 0.3, longitudinal dispersivity of 10 m, and the ratio of longitudinal to transverse dispersivity of 0.3. Initial levels of hydrocarbon and oxygen in the aquifer are

0.0 and 9.0 mg/L, respectively. We will simulate a continuous spill event and compute the resulting hydrocarbon and oxygen contours after one year using the model #1 (instantaneous, aerobic degradation model). The RT3D input files for this example problem are listed below:

Input Data Set

RT3D-super file (test1.rts)

```
RT3DSUP
BTN “test1.btn”
FLO “flow.hff”
ADV “test1.adv”
DSP “test1.dsp”
SSM “test1.ssm”
RCT “test1.rct”
CHK Y
OUT “test1.out”
CON “test1.con”
MAS “test1.mas”
SPC “BTEX” 1 1
SPC “Oxygen” 2 1
```

RT3D-BTN file (test1.btn)

```
         1        31        51         1         2         2
  hr   m  kg
 T T T T F F F F F F
0
         0 10.000000
         0 10.000000
         0 10.000000
         0 10.000000
         0 0.3000000
       100         1              (30I3)         0
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
         0       0.0
         0 9.0000000
 -999.0000
         1         1         1         1         T
         0
         0
         T
 730.00000         1 1.0000000
       0.0      1000
```

RT3D-DSP file (test1.dsp)

```
0 10.000000
0 0.3000000
0 1.0000000
0       0.0
```

RT3D-SSM file (test1.ssm)

```
T F F F F F
        63
         1
         1        16        16 1000.0000          2 1000.0000       0.0
```

RT3D-RCT file (test1.rct)

```
0         1         1         0         0
         0 1600000.0
3.080000000000000e+000
```

RT3D Output Results

After creating all necessary files, run the RT3D code. The code will ask for the input superfile name, type “test1.rts”.

If you have a demo version of GMS 2.1, you can directly import test1001.con and test1002.con files to plot the contour profiles of hydrocarbon and oxygen. If you do not have GMS, use the test1001.ucn and test1001.ucn (which are output in the same format as the MT3D ucn file) in your favorite graphic software. The predicted hydrocarbon and oxygen contours after a 2-year simulation period are shown in the following figures.

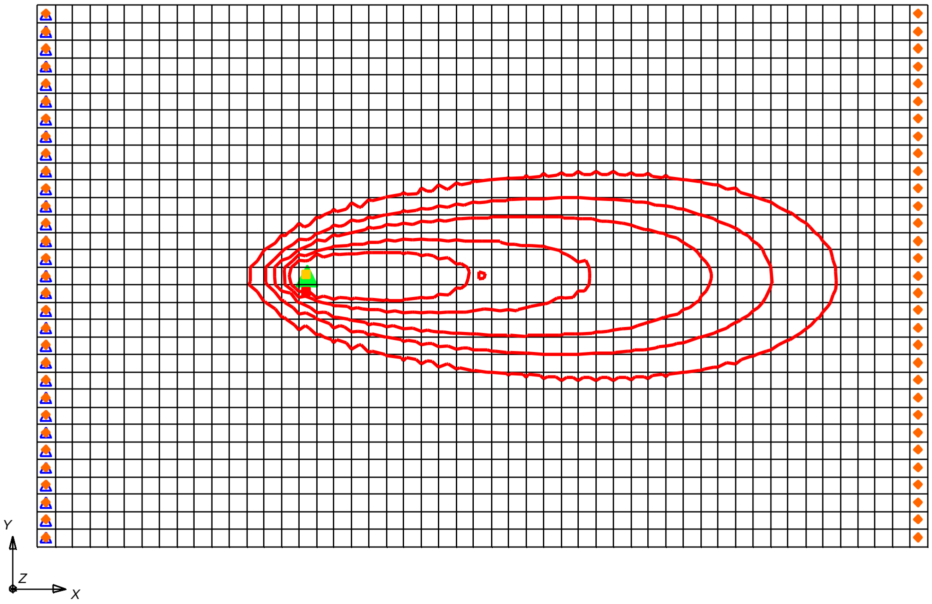

Figure 4. Hydrocarbon Contours (contour levels are: 30, 20, 10, 5, and 1 mg/L)

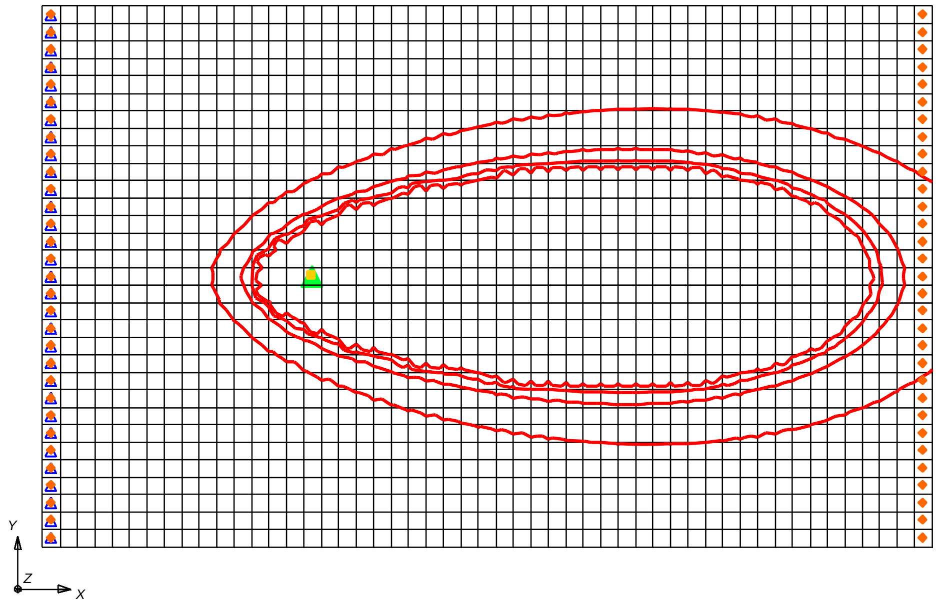

Figure 5. Oxygen Contours (contour levels are: 1, 2, 4, and 8 mg/L)

Example 2: User-defined Reaction Module (Model # 10)

The objective of this section is to describe the steps involved in developing a user-defined module for simulating a new reactive transport system. This is the most powerful option provide in the *RT3D* modeling environment. Under this option, a user has the choice to define any type of kinetic-reactions. Once developed, the user can distribute his/her reaction model, to other interested modelers, for applying the model at different sites.

User-defined reaction packages can be created using one of following two options: the dynamically linked library (DLL) option or the linked subroutine option. With the DLL option, a Fortran subroutine for the reaction package is compiled as a a stand-alone DLL (using either Microsoft Fortran Powerstation or Digital Visual Fortran). The DLL is then copied to the same directory as the *RT3D* executable and is automatically launched by *RT3D* when *RT3D* is executed. The *RT3D* executable does not need to be recompiled. Because of the portability of the developed reaction package, the DLL option is the more convenient of the two options. However, the DLL option is only available on the Windows 95/NT platform. It is not available on Unix.

With the linked subroutine option, the code for the new reaction subroutine is compiled and linked with the rest of the *RT3D* source code (or the *RT3D* library made for a specific computer platform). In other words, the *RT3D* executable must be recompiled each time the reaction package is modified. This is the only option available on Unix. Most of this section describes the pc-based DLL approach for creating user-defined reaction packages. However, special instructions for Unix users are listed in the end of this section.

Required Background

As a reaction module developer, you are considered as an advanced user of the *RT3D* code. The required background is as follows:

- Should have a basic understanding of the functionality of *RT3D* code and understand how different types of components (mobile and immobile) are mathematically described within *RT3D*-modeling framework.

- Should be familiar with the input data structure of *MODFLOW* and *RT3D* codes, and must be able to create the input files with little effort.

- Should be familiar with the data structure of *RT3D* input files such as BTN, SSM, and RCT files.

- Should have a basic understanding of the FORTRAN language, and have access to *Microsoft FORTRAN Powerstation* or *DEC Visual FORTRAN* compiler (or a UNIX system based FORTRAN 90 compiler).

- Should have some background/understanding of biochemical reaction kinetics. Note, use of inappropriate kinetic expressions and/or kinetic constants may lead to unpredictable code behavior.

- Should be familiar with contaminant transport equations and coupled nonlinear differential equations.

Problem Definition

The reaction model considered here predicts the transport and biodegradation of PCE and its degradation products, TCE, DCE, and VC. Assuming first-order sequential biodegradation kinetics, the transformation of PCE and its decay products, along with its transport, can be predicted using the following set of partial differential equations:

$$\frac{\partial[\mathrm{PCE}]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial[\mathrm{PCE}]}{\partial x_j}\right) - \frac{\partial\left(v_i[\mathrm{PCE}]\right)}{\partial x_i} + \frac{q_s}{\phi}[\mathrm{PCE}]_s - K_{Pce}[\mathrm{PCE}] \tag{51}$$

$$\frac{\partial[\mathrm{TCE}]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial[\mathrm{TCE}]}{\partial x_j}\right) - \frac{\partial\left(v_i[\mathrm{TCE}]\right)}{\partial x_i} + \frac{q_s}{\phi}[\mathrm{TCE}]_s + Y_{Tce/Pce}K_{Pce}[\mathrm{PCE}] - K_{Tce}[\mathrm{TCE}] \tag{52}$$

$$\frac{\partial[\mathrm{DCE}]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial[\mathrm{DCE}]}{\partial x_j}\right) - \frac{\partial\left(v_i[\mathrm{DCE}]\right)}{\partial x_i} + \frac{q_s}{\phi}[\mathrm{DCE}]_s + Y_{Dce/Tce}K_{Tce}[\mathrm{TCE}] - K_{Dce}[\mathrm{DCE}] \tag{53}$$

$$\frac{\partial[\mathrm{VC}]}{\partial t} = \frac{\partial}{\partial x_i}\left(D_{ij}\frac{\partial[\mathrm{VC}]}{\partial x_j}\right) - \frac{\partial\left(v_i[\mathrm{VC}]\right)}{\partial x_i} + \frac{q_s}{\phi}[\mathrm{VC}]_s + Y_{Vc/Dce}K_{Dce}[\mathrm{DCE}] - K_{Vc}[\mathrm{VC}] \tag{54}$$

where [PCE], [TCE], [DCE], and [VC] are the concentrations of the respective contaminants in mg/L; K_{PCE}, K_{TCE}, K_{DCE}, and K_{VC} are first-order degradation rates, R_P, R_T, R_D, R_V are retardation coefficients; and $Y_{TCE/PCE}$, $Y_{DCE/TCE}$, and $Y_{VC/DCE}$ are yield coefficients whose values can be computed from stoichiometric relations as 0.79, 0.74, and 0.64, respectively [for example, based the chemical reaction stochiometery, one mole of PCE will yield one mole of TCE (or 165.8 grams of PCE will yield 131.36 grams of TCE), therefore the yield value for $Y_{TCE/PCE} = 131.36/165.8 = 0.79$]. The kinetic model formulation presented assumes that the biological degradation reactions occur only in the liquid phase (a more conservative assumption).

Using the reaction operator-split strategy, the biological reaction kinetics can be separated from the transport equations and assembled together into a set differential equations:

$$\frac{d[PCE]}{dt} = - K_{Pce}[PCE] \tag{55}$$

$$\frac{d[TCE]}{dt} = Y_{Tce/Pce} K_{Pce}[PCE] - K_{Tce}[TCE] \tag{56}$$

$$\frac{d[DCE]}{dt} = Y_{Dce/Tce} K_{Tce}[TCE] - K_{Dce}[DCE] \tag{57}$$

$$\frac{d[VC]}{dt} = Y_{Vc/Dce} K_{Dce}[DCE] - K_{Vc}[VC] \tag{58}$$

The above set of coupled differential equations describes the kinetics of PCE degradation and its daughter products.

Developing a New Reaction Module

Three different methods can be used for coding a new user-defined reaction package. Each method treats the reaction parameter information (values of K_{tce}, K_{pce}, etc.) in a different fashion.

1) In the first method, all reaction parameter values are explicitly assigned within the reaction module prior to compilation. This is not an efficient method because it requires recompilation of the reaction routine whenever a reaction parameter value is modified. However, this method is recommended for testing a new reaction module with the BATCHRXN utility.

2) In the second method, all the reaction parameters are assumed be constant (spatially invariable) but are assigned or modified externally, as input data, via the *.RCT file (to run RT3D) or via the “batch.in” file (to run BATCHRXN).

3) In the third method, some or all of the reaction parameters are treated as spatially variable (i.e. a different value may be assigned to each cell). The parameter values should be externally assigned/modified as input data, via the *.RCT file. This option should be

used with caution because it may require significant computer resources, both execution time and memory.

The complete listing of the FORTRAN subroutine (using method-#1) that describes the PCE degradation reactions is given below:

```
SUBROUTINE rxns(ncomp,nvrxndata,j,i,k,y,dydt,
     &        poros,rhob,reta,rc,nlay,nrow,ncol,vrc)
c ***** Block 1: Comments block *******
c23456789012345678901234567890123456789012345678901234567890123456789012
c ncomp - Total number of components
c nvrxndata - Total number of variable reaction parameters to be input via RCT file
c J, I, K - node location (used if reaction parameters are spatially variable)
c y - Concentration value of all component at the node [array variable y(ncomp)]
c dydt - Computed RHS of your differential equation [array variable dydt(ncomp)]
c poros – porosity of the node
c reta -  Retardation factor [array variable reta(mcomp)]
c rhob -  bulk density of the node
c rc - Stores spatially constant reaction parameters (up to 100 values)
c nlay, nrow, ncol - Grid size (used only for dimensioning purposes)
c vrc - Array variable that stores spatially variable reaction parameters
c ***** End of Block 1 *******

c *** Block 2: Please do not modify this standard interface block ***
      !MS$ATTRIBUTES DLLEXPORT :: rxns
      IMPLICIT NONE
      INTEGER ncol,nrow,nlay
      INTEGER ncomp,nvrxndata,j,i,k
      INTEGER First_time
      DATA First_time/1/
      DOUBLE PRECISION y,dydt,poros,rhob,reta
      DOUBLE PRECISION rc,vrc
      DIMENSION y(ncomp),dydt(ncomp),rc(50)
      DIMENSION vrc(ncol,nrow,nlay,nvrxndata),reta(50)
C ****** End of block 2 *******
C *** Block 3: Declare your problem-specific new variables here ***
C     INTEGER
      DOUBLE PRECISION pce,tce,dce,vc,kpce,ktce,kdce,kvc
      DOUBLE PRECISION ytcepce,ydcetce,yvcdce
C ***** End of Block 3  ******
C *** Block 4: Initilize reaction parameters here, if required ***
      IF (First_time .EQ. 1) THEN
         kpce = 0.005  !PCE first-order degradation rate
         ktce = 0.003  !TCE first-order degradation rate
         kdce = 0.002  !DCE first-order degradation rate
         kvc = 0.001  !VC first-order degradation rate
        ytcepce = 131.36/165.8
        ydcetce = 96.9/131.36
        yvcdce = 62.45/96.9
         First_time = 0 !reset First_time to skip this block later
      END IF
C ***** End of Block 4  ******
C  *** Block 5: Definition of other variable names ***
       pce = y(1)
       tce = y(2)
       dce = y(3)
       vc = y(4)
C ***** End of Block 5  ******
c  *** Block 6: Definition of Differential Equations ***
       dydt(1) = -kpce*pce/reta(1)
       dydt(2) = (-ktce*tce + kpce*pce*ytcepce)/reta(2)
       dydt(3) = (-kdce*dce + ktce*tce*ydcetce)/reta(3)
       dydt(4) = (-kvc*vc + kdce*dce*yvcdce)/reta(4)
C ***** End of Block 6  ******
      RETURN
      END
```

Note that the reaction subroutine listed above is organized into the following six distinct blocks of FORTRAN code:

1) The comment block: this block contains a definition for all the variables that are passed into the rxns routine from the *RT3D* main program.

2) The interfacing block: this block contains FORTRAN statements that define the type of passed variables. This block should not be altered under any circumstances.

3) The local variable definition block: this block is used define all the new local variables that are used exclusively in the rxns subroutine.

4) The reaction parameter initialization block; this block is used to specify values for all the reaction parameters (such as K_{TCE}, K_{PCE}, etc.). Note that this initialization block is executed only when the subroutine is called for the first time.

5) Variables naming block; this block is typically used to transfer certain passed variables into meaningful names (this is required for code clarity purposes only).

6) The differential equations block; this block is used to describe the PCE degradation reaction equations in FORTRAN language.

Compiling a Reaction Module

The FORTRAN subroutine presented above can be copied from the pdf viewer and pasted into an ASCII editor. Save the file as rxns_1.f. Now the reaction module is ready for compilation. To create a dynamic link library (RXNS.DLL file), open a DOS window and go to the subdirectory where RXNS_1.F was saved and type the following command (this assumes that you have previously installed the *Microsoft FORTRAN Power Station or DVF* compiler): `fl32 /LD rxns_1.f`

At this stage, the compiler will display all the syntax errors. If no changes have been made to the code, there shouldn’t be any errors. If there are errors, fix them and recompile the code. After a successful compilation step, the compiler will create a DLL file (named as

RXNS_1.DLL). Use the following *DOS* command (or use file manager) to rename the DLL as RXNS.DLL: `RENAME RXNS_1.DLL RXNS.DLL`

Testing and Debugging a Reaction Module

After creating a new reaction package, it is always a good practice to test its response in a batch mode (i.e. how the reactions would behave in a batch reactor). For this purpose, we have developed a utility called BATCHRXN. This debugging tool can read the reaction information in a RXNS.DLL file and simulate its behavior in a batch mode.

To use BATCHRXN, it is recommended that the user define all the kinetic reaction parameters within the rxns subroutine (i.e. using method-#1). This is because the BATCHRXN code will not read information from RT3D input files, including the RCT file. Therefore, a new reaction module should always be developed in a format similar to method-#1 (or #2) and later, if required, the kinetic parameters can be moved into a RCT file to run RT3D simulations.

Running BATHCHRXN

Whenever BATCHRXN is executed, the code searches for a RXNS.DLL file in the local subdirectory. Thus, you should copy the RXNS.DLL file to the RT3D-directory where the BATCHRXN.EXE and RT3D.EXE files are stored by default. While developing a new reaction module, always keep the source code (rxns**.f) in a local project directory, and rename and copy the DLL as RXNS.DLL and move it to the *RT3D* directory.

After you have copied the RXNS.DLL file into the *RT3D* directory, open a *DOS* window and run the BATCHRXN code from your project subdirectory. The code will query for several input data. The queries and the appropriate responses for this example problem are given below:

```
Input-> ncomp, no_of_timesteps, delt
Format-> INTEGER, INTERGER, REAL
4 100 10.
Input-> Initial Values of y(i)
ncomp lines - One REAL entry per line
100.    <Initial PCE concentration>
```

```
0.0     <Initial TCE concentration>
0.0     <Initial DCE concentration>
0.0     <Initial VC concentration>
Do you want to change default atol & rtol: y/n
Default values are:- atol=1.0e-10 & rtol=1.0e-9
n
Input-> ncrxndata (number of reaction parameters)
Format-> INTEGER
0
```

You can also store these responses in a data file (say, batch.in) and use the command "`batchrxn < batch.in`" to automatically answer all the BATCHRXN queries. Here is an example batch.in file:

```
4 1000 1.
100.
0.
0.
0.
y
1.0e-11 1.0e-10
1.0e-11 1.0e-10
1.0e-11 1.0e-10
1.0e-11 1.0e-10
0
```

Note in the second case atol and rtol values are not initialized as default values, instead are set explicitly by the user. For guidelines on setting atol and rtol values, please see "Inputs for RT3D-RCT files" section, given above.

Viewing BATCHRXN Results

After reading the input, BATCHRXN will execute and create an output file called batchrxn.out. The output file will contain a set of white-space separated ASCII data in column format (written using the Format statement 21E15.5). For our example problem, five columns are output: Time, specie-1 (PCE), specie-2 (TCE), specie-3 (DCE), and specie-4 (VC). Use any spreadsheet software such as *Excel* to plot and analyze this output data. The BATCHRXN output results for the example problem are shown in the figure below:

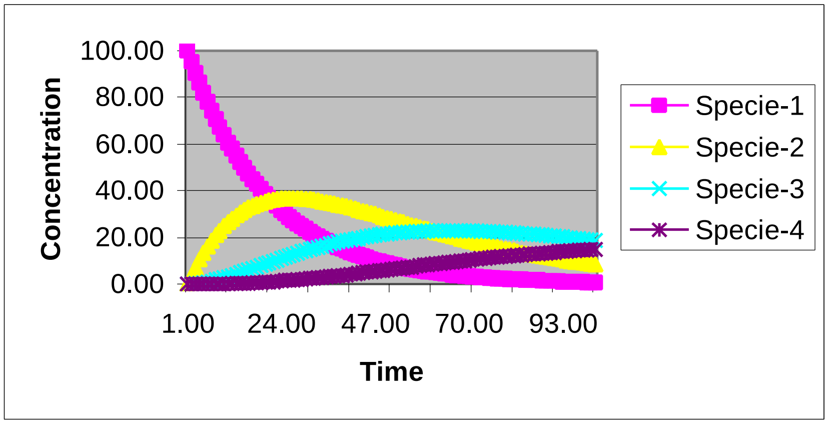

Figure 6. BATCHRXN Ouput

The results of a BATCHRXN run should be used to: a) debug errors in reaction expressions; b) check whether the reaction expressions and parameter values are reasonable (microcosm or batch reactor data may be used to verify this); c) select appropriate values for numerical parameters atol (absolute tolerance) and rtol (relative tolerance).

Once a reaction package is tested using BATCHRXN, the DLL can be used with *RT3D* to run a full-fledged *RT3D* simulation. However, before running *RT3D* simulations, it is always useful to perform tracer transport simulations using MT3D or (using *RT3D* in single-species tracer mode) and verify the results.

The flow and transport model we will be using in tutorial exercise is identical to the one described in exercise #1. A complete listing of all the *RT3D*-input files for this example problem is listed below:

Input Data Set

RT3D Super file (test2.rts)

```
RT3DSUP
BTN “test2.btn”
FLO “flow.hff”
ADV “test2.adv”
DSP “test2.dsp”
SSM “test2.ssm”
RCT “test2.rct”
CHK Y
```

```
OUT “test2.out”
CON “test2.con”
DSS “test2.dss”
MAS “test2.mas”
SPC “PCE” 1 1
SPC “TCE” 2 1
SPC “DCE” 3 1
SPC “VC” 4 1
```

RT3D-BTN file (test2.btn)

```
         1        31        51         1         4         4
  hr   m  kg
 T T T T F F F F F F
 0
         0 10.000000
         0 10.000000
         0 10.000000
         0 10.000000
         0 0.3000000
         0         1
         0       0.0
         0       0.0
         0       0.0
         0       0.0
      999.0000
         1         1         1         1         T
         0
         0
         T
 1095.0000         1 1.0000000
       0.0      1000
```

RT3D-ADV file (test2.adv)

```
1 1.0000000     75000
         1 0.5000000
 0.0000100         0         0        16         2        32 1.0000000
```

RT3D-DSP file (test2.dsp)

```
0 10.000000
         0 0.3000000
         0 1.0000000
         0       0.0
```

RT3D-SSM file (test2.ssm)

```
T F F F F F
        63
         1
         1        16        16 1000.0000         2 1000.0  0.0  0.0  0.0
```

RT3D-RCT file (test2.rct)

```
0        10         4         0         1
         0 1600000.0
1.000000000000000e-010 1.000000000000000e-009
1.000000000000000e-010 1.000000000000000e-009
1.000000000000000e-010 1.000000000000000e-009
1.000000000000000e-010 1.000000000000000e-009
5.000000000000000e-003
```

```
3.000000000000000e-003
2.000000000000000e-003
1.000000000000000e-003
```

Note the user should be aware of the order in which the reaction constants should to be entered in the RCT package. This order would depend on the way in which the reaction package was originally coded. In this example, the order should be: k_{pce}, k_{tce}, k_{dce}, and k_{vc}. Therefore, the first reaction constant entry will be transferred to k_{pce}, second one to k_{tce}, and so on. Use of a pre-processor, such as GMS, would help keep track of the order of reaction parameters, and would make the data input process more user-friendly and flexible (Clement and Jones, 1998).

RT3D Output Results

An example of PCE and its decay products transport problem, similar to the one presented above, was solved and the results are furnished in Clement et al. (1998). Here we only present the result for the first (PCE) and the last (VC) components after a 3-year simulation period.

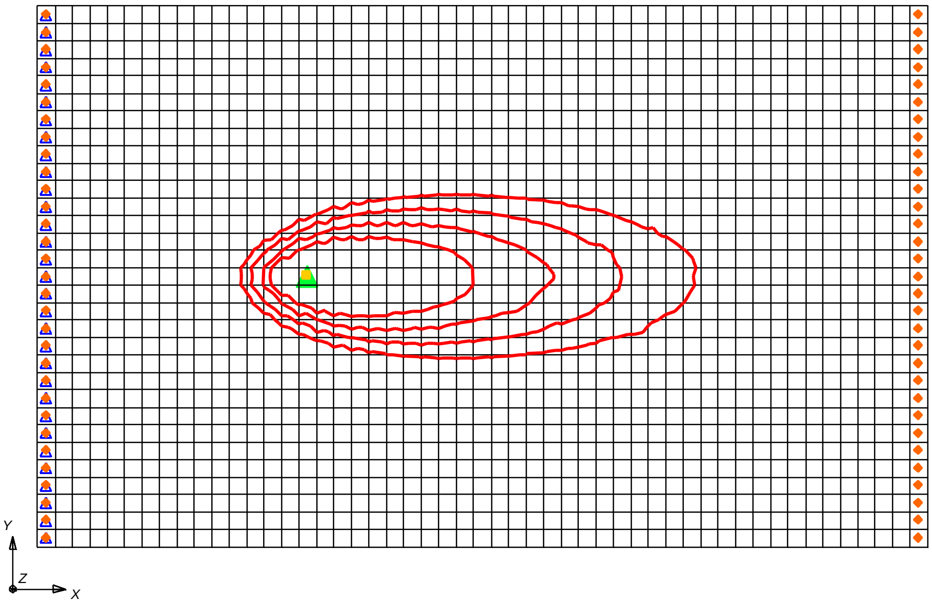

Figure 7. PCE contours after 3 years (contour levels are: 8, 4, 2, and 1 mg/L)

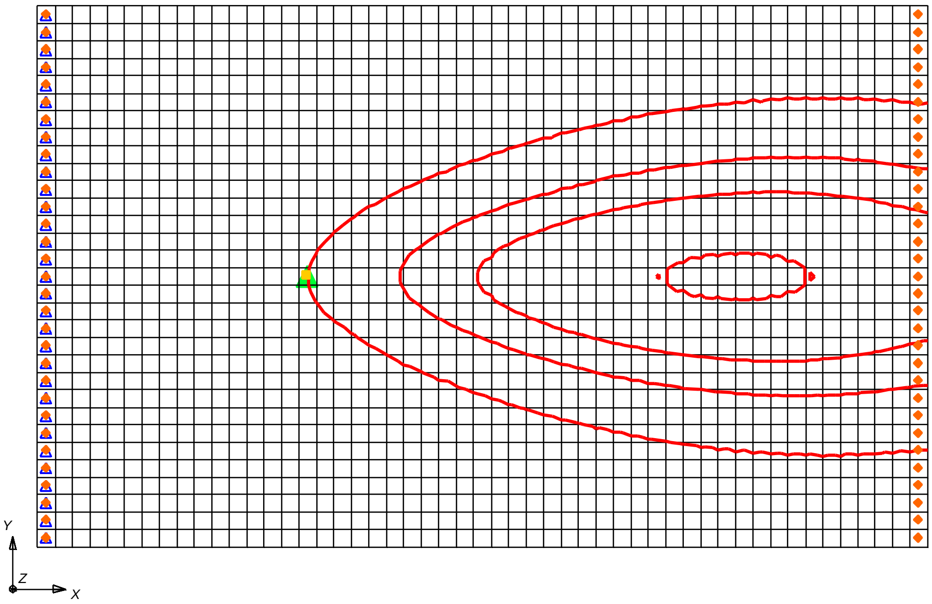

Figure 8. VC contours after 3 years (contour levels are: 2, 1, 0.5, and 0.1 mg/L)

Generalized User-Defined Reaction Module

In an earlier section, we mentioned that there are three different methods available for coding the reaction subroutine. Method-#1, which is the preferred method for testing and debugging a new reaction package, was discussed in detail in previous sections. Now, let's assume that we have sufficiently tested the reaction package and ready to distribute the package to other users who might be interested in using the reaction package, with some other site-specific reaction parameter values, without having to recompile the code. To accomplish this, we have to re-code the reaction package based on either method-#2 or method-#3. We will utilize method-#3 (which is the most advanced method) here for demonstration purposes. As described earlier, method-#3 provides an option to let some or all of the reaction parameters be defined as spatially variable parameters. This method also allows the user to set the values of reaction parameters via the *.RCT input file.

In this test problem, we have a total of seven reaction parameters. For illustration purposes, we will assume that five of them are constant parameters: K_{pce}, K_{tce}, Y_{tce_pce},

Y_{dce_tce}, and Y_{vc_dce} (set NCRXNDATA = 5), and two of them are spatially variable parameters: K_{dce} and K_{vc} (set NVRXNDATA = 2). The following FORTRAN listing for the reaction module (using method-#3) incorporates these changes:

```
      SUBROUTINE rxns(ncomp,nvrxndata,j,i,k,y,dydt,
     $         poros,rhob,reta,rc,nlay,nrow,ncol,vrc)
c ***** Block 1: Comments block *******
c23456789012345678901234567890123456789012345678901234567890123456789012
c ncomp - Total number of components
c nvrxndata - Total number of variable reaction parameters to be input via RCT file
c J, I, K - node location (used if reaction parameters are spatially variable)
c y - Concentration value of all component at the node [array variable y(ncomp)]
c dydt - Computed RHS of your differential equation [array variable dydt(ncomp)]
c poros – porosity of the node
c reta -  Retardation factor [array variable dpreta(mcomp)]
c rhob -  Bulk density of the node
c rc - Stores spatially constant reaction parameters (up to 100 values)
c nlay, nrow, ncol - Grid size (used only for dimensioning purposes)
c vrc - Array variable that stores spatially variable reaction parameters
c ***** End of Block 1 *******

c *** Block 2: Please do not modify this standard interface block ***
      !MS$ATTRIBUTES DLLEXPORT :: rxns
      IMPLICIT NONE
      INTEGER ncol,nrow,nlay
      INTEGER ncomp,nvrxndata,j,i,k
      INTEGER First_time
      DATA First_time/1/
      DOUBLE PRECISION y,dydt,poros,rhob,reta
            DOUBLE PRECISION rc,vrc
            DIMENSION y(ncomp),dydt(ncomp),rc(50)
      DIMENSION vrc(ncol,nrow,nlay,nvrxndata),reta(50)
C ****** End of block 2 *******
C *** Block 3: Declare your problem-specific new variables here ***
C     INTEGER
            DOUBLE PRECISION pce,tce,dce,vc,kpce,ktce,kdce,kvc
      DOUBLE PRECISION ytcepce,ydcetce,yvcdce
C ***** End of Block 3  ******
C *** Block 4: INITIALIZE YOUR CONSTANTS HERE, IF DESIRED ***
      IF (First_time .EQ. 1) THEN
         kpce = rc(1)  !PCE first-order degradation rate (constant)
         ktce = rc(2)  !TCE first-order degradation rate (constant)
c        kdce is a spatially variable parameter, it is defined in block-5
c        kvc is a spatially variable parameter, it is defined in block-5
               ytcepce = rc(3)
               ydcetce = rc(4)
               yvcdce = rc(5)
         First_time = 0 !reset First_time to skip this block later
      END IF
C ***** End of Block 4  ******
C  *** Block 5: Definition of other variable names ***
       pce = y(1)
       tce = y(2)
       dce = y(3)
       vc = y(4)
       kdce = vrc(j,i,k,1)!Spatially variable DCE decay rate
       kvc = vrc(j,i,k,2)  ! Spatially variable vc decay rate
C ***** End of Block 5  ******
c  ***  Block 6: Definition of Differential Equations ***
      dydt(1) = -kpce*pce/reta(1)
      dydt(2) = (-ktce*tce + kpce*pce*ytcepce)/reta(2)
      dydt(3) = (-kdce*dce + ktce*tce*ydcetce)/reta(3)
      dydt(4) = (-kvc*vc + kdce*dce*yvcdce)/reta(4)
C ***** End of Block 6  ******
      RETURN
      END
```

This subroutine is identical to the previous version except that in blocks 4 and 5, the rc() and vrc() arrays are used to initialize reaction parameters. The values of the parameters will be defined in the *RT3D* input file *.RCT. The RCT file (test3.rct) that should be used to run this reaction module, is given below (other *RT3D* input files in “test3” data set must be identical to those used in the previous data set “test2”):

```
0        10         5         2         1
         0 1600000.0
1.000000000000000e-010 1.000000000000000e-009
1.000000000000000e-010 1.000000000000000e-009
1.000000000000000e-010 1.000000000000000e-009
1.000000000000000e-010 1.000000000000000e-009
5.000000000000000e-003 <this entry is a constant value for kpce>
3.000000000000000e-003 <this entry is a constant value for ktce>
7.923000000000000e-001 <this entry is a constant value for ytce_pce>
7.377000000000000e-001 <this entry is a constant value for ydce_tce>
6.445000000000000e-001 <this entry is a constant value for yvc_dce>
         0 0.0020000   <this entry is an array entry for kdce>
         0 0.0010000   <this entry is an array entry for kvc>
```

It is again emphasized that the user should be aware of the order in which the reaction constants must be entered in the RCT package. It is also important to note that in the RCT file, all constant reaction parameters should be defined first before defining the (spatially) variable reaction parameters. Short notes (which are not a part of the input data) are included in the above data set, just to emphasize this point. The results of this example should be identical to the previous example results.

Special Instructions for UNIX users

Since the DLL option requires *Microsoft* or *Digital’s visual FORTRAN* compiler specific commands, this method is incompatible with *Unix* platforms. Therefore, *Unix* users should always comment the first line in the block-2 of rxns.f file that begin with the command “!MS$ATTRIBUTES”. In addition, whenever changes are made to the RXNS.F code, the RT3D executable should to be recreated, using a FORTRAN-90 compiler, by linking RXNS.OBJ with the RT3D.LIB (to use *RT3D*) or with BATCHRXN.LIB (to use BATCHRXN).

REFERENCES

Baveye, P. and Valocchi, V.A., 1989, An evaluation of mathematical models of the transport of biologically reacting solutes in saturated soils and aquifers, *Water Resources Research,* 25, 1413-1421.

Bear, J., Hydraulics of groundwater, 1979, McGraw-Hill Book Co., Inc., New York, N.Y.

Clement, T.P., Hooker, B.S., and Skeen, R.S., 1996a, Numerical modeling of biologically reactive transport near a nutrient injection well, *ASCE Journal of Environmental Engineering Division*, 122(9), 833-839.

Clement, T.P., Hooker, B.S., and Skeen, R.S., 1996b, Macroscopic models for predicting changes in saturated porous media properties caused by microbial growth, *Ground Water*, 34(5), 934-942.

Clement, T.P., Sun, Y., Hooker, B.S., Petersen, J.N., 1997a, Modeling Natural Attenuation of Contaminants in Saturated Groundwater, Proceedings of the Fourth International In situ and On site Bioremediation conference, New Orleans, *Bioremediation,* 4 (1), 37-42.

Clement, T.P, Peyton, B.M., Skeen, R.S., Jennings, D.A., Petersen, J.N., 1997b, Microbial growth and transport in porous media under denitrification conditions: Experiment and simulations, *Journal of Contaminant Hydrology,* 24, 269-285.

Clement, T.P., Sun, Y., Hooker, B.S., Petersen, J.N., 1998, Modeling multi-species reactive transport in groundwater aquifers, *Groundwater Monitoring & Remediation Journal*, In press, will appear in the Spring 1998 issue.

Clement, T.P., and N.L. Jones, 1998, RT3D Tutorials for GMS users, Pacific Northwest National Laboratory Report, PNNL-11805.

Haggerty, R., and Gorelick, S.M., 1994, Design of multiple contaminant remediation: Sensitivity to rate-limited mass transfer, *Water Resources Research*, 30(2), 435-446.

Hornberger, G.M., Mills, A.L., and Herman, J.S., 1992, Bacterial transport in porous media: Evaluation of a model using laboratory observations, *Water Resources Research*, 28(3), 915-938.

Kaluarachchi, J., and J., Morshed, 1995, Critical assessment of the operator-splitting technique in solving the advection-dispersion-reaction equation: 1. First-order reaction, *Advances in Water Resources*, 18 (2), 89-100.

McDonald, M.D., and Harbaugh, A.W., A modular three-dimensional finite-difference flow model, *Techniques in Water Resources Investigations of the U.S. Geological Survey*, Book 6., 586 pp.

Molz, F.J., Widdowson, M.A., and Benefield, L.D., 1986, Simulation of microbial growth dynamics coupled to nutrient oxygen transport in porous media, *Water Resources Research*, 22(8), 1207-1216.

Peyton, B.M., Skeen, R.S., Hooker, B.S., Lundman R.W., and Cunningham, A.B., 1995, Evaluation of bacterial detachment rates in porous media, *Applied Biochemistry and Biotechnology*, 51, 785-797.

Reddy, H.L., and Ford, R.M., 1996, Analysis of biodegration and bacterial transport: Comparison of models with kinetic and equilibrium bacterial adsorption, *Journal of Contaminant Hydrology*, 22, 271-287.

Rifai, S.H., Bedient, P.B., Borden, R.C., and Haasbeek, J.F., 1987, BIOPLUME II - Computer model of two-dimensional contaminant transport under the influence of oxygen limited biodegradation in ground water, National Center for Ground Water Research, Rice University.

Rifai, S.H., and P.B. Bedient, 1990, Comparison of biodegradation kinetics with an instantaneous reaction model for groundwater, Water Resources Research, 26(4), 637-645.

Rifai, S.H., Newell, C.J., Miller, R., Taffinder, S., and Rounsaville, M., 1995, Simulation of natural attenuation with multiple electron acceptors, *Bioremediation*, 3(1), 53-58.

Sun, Y. and T.P. Clement, 1998, Verification of a reactive transport code (RT3D) against 1-, 2-, and 3-dimensional analytical solutions, submitted, *Ground Water*.

Taylor, S.W., and P. R., Jaffe, 1990, Substrate and biomass transport in a porous medium, *Water Resources Research*, 26, 2153-2159.

Valocchi, A.J. and M. Malmstead, 1992, Accuracy of operator-splitting for advection-dispersion-reaction problems, *Water Resources Research*, 28(5), 1471-1476.

Walter, A.L., Frind, E.O., Blowes, D.W., Ptacek, C.J., and Molson, J.W., 1994, Modeling of multicomponent reactive transport in groundwater 1. Model development and evaluation, *Water Resources Research*, 30(11), 3137-3148.

Wiedemeier, T.H., Wilson, J.T., Kampbell, D.H., Miller, R.N., and Hansen, J., 1995, Technical protocol for implementing intrinsic remediation with long-term monitoring for natural attenuation of fuel contamination dissolved in groundwater, Volume 1 & 2, Air Force Center for Environmental Excellence, Technology Transfer Division, Brooks AFB, San Antonio, Texas.

Wiedemeier, T.H., Swanson, M.A., Wilson, J.T., Kampbell, D.H., Miller, R.N., and Hansen, J., 1996, Approximation of biodegradation rate constants for monoaromatic hydrocarbons (BTEX) in groundwater, Groundwater Monitoring & Remediation, XVI (3), 186-194.

Wiedemeier, T.H., M.A. Swanson, D.E. Moutoux, E.K. Gordon, J.T.Wilson, B.H. Wilson, D.H. Kampbel, J. Hansen, and P. Haas, 1997, Technical protocol for evaluating natural attenuation of chlorinated solvents in groundwater, Air Force Center for Environmental Excellence, Technology Transfer Division, Brooks AFB, San Antonio, Texas.

Zheng, C., MT3D, 1990, A modular three-dimensional transport model for simulation of advection, dispersion and chemical reactions of contaminants in groundwater systems, U.S.E.P.A Report.

Zysset, A., Stauffer, F., and Dracos, T., 1994, Modeling of reactive groundwater transport governed by biodegradation, *Water Resources Research*, 30(8), 2423-2434.