



What's New in *RT3D* version 2.5

January 2002 (with minor update - February 2003)

CONTACT INFORMATION	1
LICENSE AND COPYRIGHT INFORMATION.....	2
DISCLAIMER OF WARRANTY	2
INTRODUCTION TO RT3D VERSION 2.5.....	2
RT3DV2.5 SOURCE/SINK (SSM) PACKAGE	3
SSM FILE STRUCTURE	4
<i>Comment on Using Artificial Injection Wells to Model Sources</i>	4
RT3DV2.5 REACTION (RCT) PACKAGE	5
RCT FILE STRUCTURE	5
RT3DV2.5 USER-DEFINED REACTION MODULE	8
EXAMPLE RXNS.F FILE	8
EVALUATION OF JACOBIAN MATRICES	10
<i>An Example of Setting up a Jacobian Matrix</i>	10
<i>Example jacrxns.f File</i>	13
<i>Example Dummy jacrxns.f File</i>	15
DLL VERSUS NON-DLL USER-DEFINED REACTION MODULE	15
<i>How to Implement the DLL Option</i>	16
USING MODFLOW 2000	16
ACKNOWLEDGMENTS	16
BRIEF HISTORY OF RT3D	17
REFERENCES AND RT3D-RELATED JOURNAL PUBLICATIONS	19

CONTACT INFORMATION

There are several resources for RT3D users. The official RT3D web site provides information on features, manuals, downloads, and utilities. Several of the developers of groundwater modelling packages that have RT3D interfaces (e.g., GMS, Visual Modflow, Groundwater Vistas) have support forums accessible through their respective web sites. If those resources do not provide the answers you seek, contact information is provided below.

Direct any general questions on RT3D use, functionality, documentation, etc. to Christian Johnson. Complex questions, "wish list" requests, and RT3D project related inquiries should be directed to Dr. Clement. Contact Christian Johnson for discussion of how Battelle may assist your project with customized reaction modules, accelerated *in situ* remediation design, and/or natural attenuation evaluations.

Official RT3D Web Site: <http://bioprocess.pnl.gov/rt3d.htm>

RT3D Author

Dr. Prabhakar Clement
Auburn University
e-mail: clement@eng.auburn.edu

RT3D Solution Developer

Mr. Christian Johnson
Battelle, Pacific Northwest Division
e-mail: rt3d@pnl.gov

LICENSE AND COPYRIGHT INFORMATION

Like any other literary work, computer programs are protected by copyright. Unauthorized reproduction or distribution of this computer code, or any portion of it, is not permitted. Users can, however, modify the code and use it for solving a specific problem. Users are not permitted to resell or redistribute the RT3D code or a modified version thereof (or any portion of RT3D) directly or via computer bulletin boards, web pages, or other publicly accessible archives. If you have modified the code for any research application and would like to share the code with other researchers, please contact the developers first and let them know the details of your work. In all your publications, please cite the RT3D manual (PNNL-11720) and the 1998 Groundwater Monitoring and Remediation journal paper by Clement et al (references are listed in the RT3Dv25_Update document and at http://bioprocess.pnl.gov/rt3d_pubs.htm). Also, consider citing other published RT3D papers that are appropriate. If you have any questions or comments, please contact the developers (see above).

DISCLAIMER OF WARRANTY

The RT3D computer code is provided without any warranty. We make no warranties, expressed or implied, that the RT3D code is free of errors or whether it will meet your need for solving a particular problem. You use the code at your own risk. The developers disclaim all liability for direct or consequential damage resulting from your use of the code.

INTRODUCTION TO RT3D VERSION 2.5

RT3D (Reactive Transport in 3-Dimensions) is a finite-difference numerical code for simulation of three-dimensional, multi-species reactive transport in groundwater aquifers. RT3D comes with 7 built-in reaction kinetics modules, but also has the flexibility to utilize reaction mechanisms of any complexity that are defined by the user. RT3D calculates the reactive transport using the flow solution from a finite difference groundwater flow package such as MODFLOW. RT3D may be applied to a wide variety of reactive flow and transport scenarios, including accelerated *in situ* bioremediation system design, natural attenuation evaluation, and *in situ* chemical oxidation.

RT3D version 2.5 (RT3Dv2.5) has been updated (compared to version 1.0) in several aspects. The base code has been updated to incorporate changes introduced into version 3.5 of the MT3DMS code (see Zheng and Wang [1999] for details on these changes, including a third-order total-variation-diminishing advection solver and an iterative generalized conjugate gradient solver). The RT3Dv2.5 code now uses standard Fortran 90, which should allow seamless use across multiple platforms. RT3Dv2.5 has new source/sink options of constant concentration and decaying sources. Four new reaction solver options have been added, including Runge-Kutta solvers and solvers using an explicit Jacobian matrix for stiff problems.

The input data structures of the BTN, ADV, and DSP files (basic, advection, and dispersion packages, respectively) are identical to those described for MT3DMS version 3.5 and the reader is referred to the MT3DMS manual [Zheng and Wang, 1999] for those packages. The input data for the source/sink package (SSM file) is identical to the MT3DMS source/sink package with the exception that the RT3Dv2.5 SSM file supports an additional boundary condition designated as the "decaying constant source" boundary condition. The input data structure for the reaction package (RCT file) is similar to the data structure used in RT3D v. 1.0, but the version 2.5 reaction package provides four new solver options and the sorption parameters can now be specified on a cell-by-cell basis (in addition to the layer-by-layer basis).

This document discusses the changes made to the source/sink and reaction package input file structure to incorporate the new features of RT3Dv2.5 and notes how the SSM and RCT files differ from the MT3DMS counterparts. Also discussed are some changes in the way the user-defined reaction module is compiled.

RT3DV2.5 SOURCE/SINK (SSM) PACKAGE

RT3Dv2.5 supports two new source/sink types for describing time-varying constant concentration source cells. The first new type (ITYPE = -1) is a constant concentration source type similar to setting a constant-concentration boundary condition (i.e., by setting ICBUND = -1 in the Basic Transport Package). The SSM constant concentration differs from setting the boundary condition in that 1) the concentration may be different in each stress period and 2) for a multi-species simulation, concentrations need not be held constant for all species. The second new source type is a decaying source. The decaying source starts at a given initial concentration and decays at a first-order rate according to Equation 1.

$$C_i = C_{i,0} \cdot \exp(-\lambda_i \cdot t) \quad (1)$$

where C_i is the concentration of species i at a given time t , $C_{i,0}$ is the initial concentration of species i in the source, and λ_i is the source decay constant for species i . The decaying source condition is defined by setting ITYPE = -2. As an example, this feature could be used to model a waste pit with a depleting NAPL source.

RT3Dv2.5 requires that the point source/sinks have NCOMP number of concentration values (which includes both mobile and immobile species) for CSS in Record D8. Similarly, the recharge and evapotranspiration arrays of records D4 and D6 each require NCOMP arrays. RT3D version 1.0 only required MCOMP values (which includes only mobile species). This change was required because of the new point source/sink types, which are also applicable to immobile species. Note that the source/sink concentrations of immobile species are used only when ITYPE = -1 or -2. For all other ITYPE values the source/sink concentrations of immobile species are ignored and may be "dummy" values. RT3D version 1.0 SSM input files must be updated to the version 2.5 format to run with RT3Dv2.5.

SSM File Structure

The data structure required for the SSM input file is identical to MT3DMS, with the exceptions that 1) RT3Dv2.5 does not support a mass-loading source (ITYPE = 15), 2) RT3Dv2.5 has a decaying source (ITYPE = -2, which MT3DMS does not support), and 3) the SSM input file structure for RT3Dv2.5 includes record D8a to support the decaying source.

RT3D2.5 D8 KSS, ISS, JSS, CSS(1), ITYPE, CSS(1), CSS(2), CSS(3), CSS(4), ..., CSS(ncomp) to define NCOMP number of species concentrations. Note that, in a change from RT3D version 1.0, a value for all species must be specified, not just for mobile species. RT3Dv1.0 SSM input files must be updated to the version 2.5 format (if immobile species are included in the model) to run simulations with RT3Dv2.5. When ITYPE = -1 or -2, positive values of CSS(i) are used as constant/decaying source concentrations and negative values are indicators that those species should not have their concentrations held constant.
Format: Free Format

The following record (D8a) is required only if the value of ITYPE = -2.

RT3D2.5 D8a Source_decay(1), Source_decay(2), Source_decay(3), ..., Source_decay(ncomp); used to define "ncomp" number of source decay constants. The values should be positive if the source is decreasing with time and negative if it is increasing with time. Specify decay value as zero for the species that were assigned negative concentration values in the previous record. Also, zero can be used for non-decaying species.
Format: Free Format.

Note that constant (or non-decaying) concentration boundary conditions can be simulated by either using the ITYPE = -1 option or the ITYPE = -2 option and setting the corresponding source decay constants to zero.

COMMENT ON USING ARTIFICIAL INJECTION WELLS TO MODEL SOURCES

Injection wells are often used to simulate contaminant release (mass discharge) from a source into an aquifer. This is a reasonable approximation for single species transport if the amount of water injected into the aquifer does not alter the flow field. However, for multi-species simulations, additional care should be exercised to maintain the injection rate at a very low level. At discharge points (i.e., at the hypothetical injection wells) the concentrations of species other than the primary contaminant are usually assumed to be zero and an injection flow rate that is too high may introduce artificial dilution of these species. For example, if a TCE source is created using an injection well, then the concentration of TCE biotransformation byproducts (DCE and VC) will be assumed to be zero in the injected water. Maintaining the injection flow rate at a high value would lead to artificial local dilution of the DCE and VC plumes near the injection well. Maintaining low flow rates will help minimize this dilution effect.

RT3DV2.5 REACTION (RCT) PACKAGE

RCT File Structure

The revised reaction package input data structure for RT3Dv2.5 is described below. RT3Dv2.5 is fully backward compatible with earlier versions of RCT file formats. Note that the RT3D code does not support the dual-porosity and NAPL options of MT3DMS. However, if required, dual porosity and NAPL dissolution kinetics can be defined via a user-defined reaction module (see the dualpore.pdf and napl.pdf files, respectively).

E1	Record:	ISOTHM, IREACT, NCRXNDATA, NVRXNDATA, ISOLVER, <u>IRCTOP</u>
	Format:	(6I10)
	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	= Reaction module number = 0, no reaction is simulated (i.e., tracer transport) = 1, Two-Species Instantaneous Reactions (BIOPLUME-II type reactions) = 2, {module reserved for future implementation} = 3, Six Species, First-Order, Rate-Limited, BTEX Degradation using Sequential Electron Acceptors = 4, Rate-Limited Sorption = 5, Double Monod Model = 6, Sequential First-Order Decay (up to 4 species, e.g., PCE/TCE/DCE/VC) = 7, Aerobic/Anaerobic PCE/TCE Dechlorination = 8 {module reserved for future implementation} = 9 {module reserved for future implementation} = 10, User-Defined Kinetics
	NCRXNDATA	= number of constant reaction parameter values
	NVRXNDATA	= number of variable reaction parameter arrays
	ISOLVER	= 0, for the instantaneous reaction modules 1 and 2. = 1, Automatic switching Gear-stiff/non-stiff solver. For stiff systems, this option will automatically compute the Jacobian matrix using finite-difference approximations. = 2, Automatic switching Gear-stiff/ non-stiff solver. For stiff systems, this option will require an external routine to compute analytical Jacobian. Need to provide an external subroutine "jacrxns.f" that specifies the Jacobian matrix for the differential reaction equations. = 3, Fehlberg fourth-fifth order Runge-Kutta method

= 4, Stiff solver based on a semi-implicit extrapolation method.
This option requires an external routine "jacrxns.f" to compute the analytical Jacobian matrix for the differential reaction equations.

= 5, Non-stiff Runge-Kutta solver

IRCTOP = This variable was included to maintain compatibility with the previous version. Set the value to 0 or 1 (preferably 1) to read sorption parameters in a layer-by-layer mode (default mode). Set the value to 2 to read 3D arrays in a cell-by-cell mode.

Note: NCRXNDATA defines the number of reaction parameters that are constant, and NVRXNDATA defines the number of reaction parameters that are spatially variable.

(Enter E2a if IRCTOP = 1)

E2a: ARRAY: RHOB(NLAY) (one value for each layer)
 READER: RARRAY

RHOB is the bulk density of the porous medium [ML⁻³].

Units used for defining bulk density and adsorption coefficient values should be consistent to yield a dimensionless retardation parameter.

(Enter E2b if IRCTOP = 2)

E2b: ARRAY: RHOB(NCOL,NROW) (one array for each layer)
 READER: RARRAY

RHOB is the bulk density of the porous medium [ML⁻³].

Units used for defining bulk density and adsorption coefficient values should be consistent to yield a dimensionless retardation parameter.

(Enter E3 data, if ISOTHM > 0)

(Enter in E3a format if IRCTOP = 1)

E3a: ARRAY: SP1(NLAY) (one value for each layer)
 (Then repeat SP1 array entries for each mobile component)

 READER: RARRAY

 SP1 is the first sorption constant.

(Enter in E3b format if IRCTOP = 2)

E3b: ARRAY: SP1(NCOL,NROW) (one array 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 [L³M⁻¹]. Other details are described in the MT3DMS manual.

(Enter E4 data, if ISOTHM > 0)

(Enter in E4a format if IRCTOP = 1)

E4a: ARRAY: SP2(NLAY) (one value for each layer)
 (Then repeat SP2 array entries for each mobile component)

 READER: RARRAY

(Enter in E4b format if IRCTOP = 2)

E4b: ARRAY: SP2(NCOL,NROW) (one array for each layer)
 (Then, repeat SP2 arrays for each mobile component)

 READER: RARRAY

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

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

 Format: Free format, two entries per line; total number of entries should be equal to "ncomp".

If ISOLVER = 1 or 2, then the atol and rtol values specified for each species are used by the solvers. If ISOLVER = 3, then only the first pair of values, atol(1) and rtol(1), is used by the solver. If ISOLVER = 4 or 5, then only the first absolute error tolerance value, atol(1), is used by the solvers. However, to maintain a general structure for the input data, atol and rtol must be specified for each species, regardless of the choice of reaction solver.

For ISOLVER = 1 or ISOLVER = 2 (i.e., the Gear-based solvers), relative (rtol) and absolute (atol) error tolerances are needed for the solver to determine how accurate the solution for the species concentration, $y(i)$, must be. The differential equation solver will attempt to control the species concentration local error, $e(i)$, such that, for each species, the local error has (approximately) a smaller magnitude than the error weight, $ewt(i)$:

$$|e(i)| \leq ewt(i)$$

where,

$$ewt(i) = rtol(i) \cdot |y(i)| + atol(i)$$

Here $ewt(i)$ is a vector of weights which must always be positive, and the values of $rtol$ and $atol$ should all be non-negative.

Setting $atol(i) = 0.0$ results in a pure relative error test on that component (species). Setting $rtol(i) = 0.0$ results in a pure absolute error test on that component. A mixed test with non-zero $rtol(i)$ and $atol(i)$ corresponds roughly to a relative error test when the solution component, $y(i)$, is much bigger than $atol$ and to an absolute error test when the solution component is smaller than the threshold $atol$.

For practical problems, the following rules of thumb may be used to set $atol$ and $rtol$ values. The $atol(i)$ should be set to a tolerable absolute error level selected by the user (around 10^{-6} to 10^{-9}). Defining m as the number of significant digits required in the solution for species $y(i)$, set $rtol(i) = 10^{-(m+1)}$. Caution: Actual (global) errors may exceed the local tolerances, so choose $atol(i)$ and $rtol(i)$ conservatively.

For ISOLVER = 4 or ISOLVER = 5, one global tolerance value, known as the error residue "eps", specified as $atol(1)$, is used to control convergence. For further details about error residue

refer to Numerical Recipes in Fortran 77 [Press et al., 1992, p. 711]. For practical problems, set $\text{eps}=\text{atol}(1)$ to a small value between 10^{-6} to 10^{-9} .

(Enter E6, if IREACT > 0)

Input all the constant reaction parameter values (number of entries must be equal to NCRXNDATA); the values will be stored in a variable RC(NCRXNDATA) which will be used in the user-defined reaction module and in other pre-defined reaction modules.

E6: ARRAY: RC(NCRXNDATA)
 Format : Free format, one entry per line, with a total of NCRXNDATA number of entries. RC(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 reaction model used in the simulation.

Input the variable reaction parameters (number of arrays must be equal to NVRXNDATA); the parameters will be stored in a variable VRC(NCOL, NROW, NLAY, NVRXNDATA) which will be used in the user-defined reaction module and in other pre-defined reaction modules.

E7: ARRAY: VRC(NCOL,NROW) (one array for each layer)
 (Then, repeat VRC for NVRXNDATA number of times)
 READER: DPRARRAY (a new array reader which is exactly similar to the reader RARRAY, but with double precision).

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

RT3DV2.5 USER-DEFINED REACTION MODULE

In RT3D version 1.0, user-defined reactions could be described in the rxns.f file. In RT3Dv2.5, an additional file named jacrxns.f is also required to complement the rxns.f file. In the jacrxns.f file, the user is expected to provide the necessary information for computing an analytical Jacobian matrix for the reaction equations. Both rxns.f and jacrxns.f subroutines will be dummy procedures when one of the pre-programmed reaction modules is used. When the user-defined reaction module is used, jacrxns.f will be a dummy procedure for reaction solvers 1, 3, and 5.

For RT3Dv2.5, the user now has the option of compiling the user-defined reaction module into the main executable (default for RT3Dv2.5) or into dynamic link libraries (DLLs). The latter option is only applicable to Microsoft® Windows® operating systems.

This section describes the rxns.f and jacrxns.f files (with examples of each) and the compilation options.

Example rxns.f File

The format for the rxns.f file is unchanged from RT3D version 1.0. An example is provided here for reference only.


```

c Reaction package for Example-2 (method-1)
c Please refer to the RT3Dv1 user manual for further details
c
      SUBROUTINE Rxns(ncomp,nvrndata,j,i,k,y,dydt,
        &      poros,rhob,reta,rc,nlay,nrow,ncol,vrc)
C*Block 1:*****
c List of calling arguments
c ncomp - Total number of components
c nvrndata - 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 [ignore dummy reta values of immobile species]
c rhob - bulk density of the node
c rc - Stores spatially constant reaction parameters (can dimension upto 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:*****
c*      *Please do not modify this standard interface block*
      IMPLICIT NONE
      INTEGER ncol,nrow,nlay
      INTEGER ncomp,nvrndata,j,i,k
      INTEGER, SAVE :: First_time=1
      DOUBLE PRECISION y,dydt,poros,rhob,reta
      DOUBLE PRECISION rc,vrc
      DIMENSION y(ncomp),dydt(ncomp),rc(100)
      DIMENSION vrc(ncol,nrow,nlay,nvrndata),reta(ncomp)
C*End of block 2*****

C*Block 3:*****
c      *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:*****
c      *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:*****
c      *Assign or compute values for new variables, if required*
      pce = y(1)
      tce = y(2)
      dce = y(3)
      vc = y(4)
C*End of block 5*****

C*Block 6:*****
c      *Differential Reaction 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

```

Evaluation of Jacobian Matrices

If solver options 2 or 4 are invoked, the user should provide the necessary information for computing an analytical Jacobian matrix for the differential reaction equations. The analytical Jacobian matrix is of size (NCOMP × NCOMP) and the elements are the partial derivatives of the system of differential equations. For a system of n number of differential equations, F_n , the Jacobian matrix is defined as in Equation 2.

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix} \quad (2)$$

where $F_i = f(x_1, x_2, \dots, x_n)$ is the differential equation for species i .

AN EXAMPLE OF SETTING UP A JACOBIAN MATRIX

This section illustrates the method for determining the analytical Jacobian matrix for a simple reaction system. The test problem models first order degradation of PCE and its daughter products under anaerobic conditions. For more details, please refer to the Example 3 discussed in Clement et al. [1998]. The transport and transformation of the sequential decay chain $\text{PCE} \rightarrow \text{TCE} \rightarrow \text{DCE} \rightarrow \text{VC}$ can be determined by solving the set of partial differential equations in Equations 3 to 6.

$$R_p \frac{\partial [PCE]}{\partial t} = \frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial [PCE]}{\partial x_j} \right) - \frac{\partial (v_i [PCE])}{\partial x_i} + \frac{q_s}{\phi} [PCE]_s - K_{pce} [PCE] \quad (3)$$

$$R_T \frac{\partial [TCE]}{\partial t} = \frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial [TCE]}{\partial x_j} \right) - \frac{\partial (v_i [TCE])}{\partial x_i} + \frac{q_s}{\phi} [TCE]_s + Y_{Tce/Pce} K_{pce} [PCE] - K_{Tce} [TCE] \quad (4)$$

$$R_D \frac{\partial [DCE]}{\partial t} = \frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial [DCE]}{\partial x_j} \right) - \frac{\partial (v_i [DCE])}{\partial x_i} + \frac{q_s}{\phi} [DCE]_s + Y_{Dce/Tce} K_{Tce} [TCE] - K_{Dce} [DCE] \quad (5)$$

$$R_V \frac{\partial [VC]}{\partial t} = \frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial [VC]}{\partial x_j} \right) - \frac{\partial (v_i [VC])}{\partial x_i} + \frac{q_s}{\phi} [VC]_s + Y_{Vc/Dce} K_{Dce} [DCE] - K_{Vc} [VC] \quad (6)$$

where, R_n is the retardation factor (unitless), $[n]$ is the aqueous-phase concentration of the n^{th} specie (ML^{-3}), D_{ij} is the hydrodynamic dispersion coefficient (L^2T^{-1}), v_i is the pore water velocity (LT^{-1}), q_s is the volumetric flux of water per unit volume of aquifer that represents sources and sinks (T^{-1}), ϕ is the aquifer porosity (unitless), $[n]_s$ is the concentration of the n^{th} specie in the sources/sinks (ML^{-3}), $Y_{d/p}$ represents the stoichiometric yield (i.e., the amount of daughter specie "d" produced by degrading a unit mass of parent specie "p") (MM^{-1}), and K_n represents the first-order degradation rate constant of the n^{th} specie (T^{-1}). Note that all reactions are assumed to occur only in the aqueous phase, which is a conservative assumption. Using the operator split strategy [Clement et al., 1998], the reaction kinetics can be separated from the transport equations and written as a set of ordinary differential equations given by Equations 7 to 10.

$$F_1 = \frac{d[PCE]}{dt} = - \frac{K_{pce} [PCE]}{R_p} \quad (7)$$

$$F_2 = \frac{d[TCE]}{dt} = \frac{Y_{Tce/Pce} K_{pce} [PCE] - K_{Tce} [TCE]}{R_T} \quad (8)$$

$$F_3 = \frac{d[DCE]}{dt} = \frac{Y_{Dce/Tce} K_{Tce} [TCE] - K_{Dce} [DCE]}{R_D} \quad (9)$$

$$F_4 = \frac{d[VC]}{dt} = \frac{Y_{Vc/Dce} K_{Dce} [DCE] - K_{Vc} [VC]}{R_V} \quad (10)$$

Elements of the first row of the Jacobian matrix, represented as pd(1,1), pd(1,2), etc. in the code, are the partial derivatives of the first reaction equation with respect to each of the four species. These partial derivatives are shown in Equations 11 to 14.

$$pd(1,1) = \frac{\partial F_1}{\partial x_1} = \frac{\partial}{\partial [PCE]} \left(\frac{d[PCE]}{dt} \right) = - \frac{K_{pce}}{R_p} \quad (11)$$

$$pd(1,2) = \frac{\partial F_1}{\partial x_2} = \frac{\partial}{\partial [TCE]} \left(\frac{d[PCE]}{dt} \right) = 0 \quad (12)$$

$$pd(1,3) = \frac{\partial F_1}{\partial x_3} = \frac{\partial}{\partial [DCE]} \left(\frac{d[PCE]}{dt} \right) = 0 \quad (13)$$

$$pd(1,4) = \frac{\partial F_1}{\partial x_4} = \frac{\partial}{\partial [VC]} \left(\frac{d[PCE]}{dt} \right) = 0 \quad (14)$$

Similarly, elements of the second, third and fourth rows of the Jacobian matrix are the partial derivatives of the second, third, and fourth reaction equations, respectively, with respect to the four species (Equations 15 to 26).

$$pd(2,1) = \frac{\partial F_2}{\partial x_1} = \frac{\partial}{\partial [PCE]} \left(\frac{d[TCE]}{dt} \right) = \frac{Y_{Tce/Pce} K_{Pce}}{R_T} \quad (15)$$

$$pd(2,2) = \frac{\partial F_2}{\partial x_2} = \frac{\partial}{\partial [TCE]} \left(\frac{d[TCE]}{dt} \right) = -\frac{K_{Tce}}{R_T} \quad (16)$$

$$pd(2,3) = \frac{\partial F_2}{\partial x_3} = \frac{\partial}{\partial [TCE]} \left(\frac{d[TCE]}{dt} \right) = 0 \quad (17)$$

$$pd(2,4) = \frac{\partial F_2}{\partial x_4} = \frac{\partial}{\partial [TCE]} \left(\frac{d[TCE]}{dt} \right) = 0 \quad (18)$$

$$pd(3,1) = \frac{\partial F_3}{\partial x_1} = \frac{\partial}{\partial [PCE]} \left(\frac{d[DCE]}{dt} \right) = 0 \quad (19)$$

$$pd(3,2) = \frac{\partial F_3}{\partial x_2} = \frac{\partial}{\partial [TCE]} \left(\frac{d[DCE]}{dt} \right) = \frac{Y_{Dce/Tce} K_{Tce}}{R_D} \quad (20)$$

$$pd(3,3) = \frac{\partial F_3}{\partial x_3} = \frac{\partial}{\partial [DCE]} \left(\frac{d[DCE]}{dt} \right) = -\frac{K_{Dce}}{R_D} \quad (21)$$

$$pd(3,4) = \frac{\partial F_3}{\partial x_4} = \frac{\partial}{\partial [VC]} \left(\frac{d[DCE]}{dt} \right) = 0 \quad (22)$$

$$pd(4,1) = \frac{\partial F_4}{\partial x_1} = \frac{\partial}{\partial [PCE]} \left(\frac{d[VC]}{dt} \right) = 0 \quad (23)$$

$$pd(4,2) = \frac{\partial F_4}{\partial x_2} = \frac{\partial}{\partial [TCE]} \left(\frac{d[VC]}{dt} \right) = 0 \quad (24)$$

$$pd(4,3) = \frac{\partial F_4}{\partial x_3} = \frac{\partial}{\partial [DCE]} \left(\frac{d[VC]}{dt} \right) = \frac{Y_{Vc/Dce} K_{Dce}}{R_V} \quad (25)$$

$$pd(4,4) = \frac{\partial F_4}{\partial x_4} = \frac{\partial}{\partial [VC]} \left(\frac{d[VC]}{dt} \right) = -\frac{K_{Vc}}{R_V} \quad (26)$$

The Jacobian matrix for the system of Equations 7 to 10 is then formulated as shown in Equation 27.

$$J = \begin{bmatrix} \left(-\frac{K_{Pce}}{R_p} \right) & 0 & 0 & 0 \\ \left(\frac{Y_{Tce/Pce} K_{Pce}}{R_T} \right) & \left(-\frac{K_{Tce}}{R_T} \right) & 0 & 0 \\ 0 & \left(\frac{Y_{Dce/Tce} K_{Tce}}{R_D} \right) & \left(-\frac{K_{Dce}}{R_D} \right) & 0 \\ 0 & 0 & \left(\frac{Y_{Vc/Dce} K_{Dce}}{R_V} \right) & \left(-\frac{K_{Vc}}{R_V} \right) \end{bmatrix} \quad (27)$$

EXAMPLE JACRXNS.F FILE

A jacrxns.f file that uses the Jacobian matrix for the example above (Equation 27) is listed below. The Jacobian matrix equations are coded in Block 6 of the subroutine.

```
c Evaluation of Jacobian for the PCE degradation problem
c Details of this routine are discussed in the v. 2.5 update manual
c
      SUBROUTINE jrxns(ncomp,nvrndata,j,i,k,ml,mu,y,pd,
&                  poros,rhob,reta,rc,nlay,nrow,ncol,vrc)

C*Block 1:*****
c List of calling arguments
c ncomp - Total number of components
c nvrndata - 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 ml,mu - are for banded jacobian (not used)
c y - Concentration of all components at the node [array variable y(ncomp)]
c pd - jacobian matrix [(ncomp x ncomp) array]
c poros - porosity of the node
c reta - Retardation factor [ignore dummy reta values of immobile species]
c rhob - bulk density of the node
c rc - Stores spatially constant reaction parameters (up to 100 parameters)
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:*****
c*   *Please do not modify this standard interface block*
      IMPLICIT NONE
      INTEGER ncol,nrow,nlay
      INTEGER ncomp,nvrndata,j,i,k,ml,mu
      INTEGER, SAVE :: First_time=1
      DOUBLE PRECISION y,pd,poros,rhob,reta,rc,vrc
      DIMENSION y(ncomp),pd(ncomp,ncomp),rc(100)
      DIMENSION vrc(ncol,nrow,nlay,nvrndata),reta(ncomp)
C*End of block 2*****

C*Block 3:*****
c   *Declare your problem-specific new variables here*
      DOUBLE PRECISION pce,tce,dce,vc,kpce,ktce,kdce,kvc
      DOUBLE PRECISION ytcepce,ydcetce,yvcdce
C*End of block 3*****

C*Block 4:*****
c   *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:*****
c   *Assign or compute values for new variables, if required*
      pce = y(1)
      tce = y(2)
      dce = y(3)
      vc = y(4)
C*End of block 5*****

C*Block 6:*****
c   *Definition of full (n x n) analytical jacobian matrix*
      pd(1,1) = -kpce/reta(1)
      pd(1,2) = 0.0d0
      pd(1,3) = 0.0d0
      pd(1,4) = 0.0d0
      pd(2,1) = kpce*ytcepce/reta(2)
      pd(2,2) = -ktce/reta(2)
      pd(2,3) = 0.0d0
      pd(2,4) = 0.0d0
      pd(3,1) = 0.0d0
      pd(3,2) = ktce*ydcetce/reta(3)
      pd(3,3) = -kdce/reta(3)
      pd(3,4) = 0.0d0
      pd(4,1) = 0.0d0
      pd(4,2) = 0.0d0
      pd(4,3) = kdce*yvcdce/reta(4)
      pd(4,4) = -kvc/reta(4)
C*End of block 6*****
      RETURN
      END

```

EXAMPLE DUMMY JACRXNS.F FILE

An analytical Jacobian matrix will be used by RT3Dv2.5 only when solver options 2 or 4 are selected. However, a jacrxns.f subroutine is required regardless of the solver option. RT3Dv2.5 comes with a dummy jacrxns.f file, but for completeness, the source for such a dummy file is shown below.

```

c      Dummy jacrxns.f file
c
c      SUBROUTINE jacrxns (ncomp, nvrndata, j, i, k, ml, mu, y, pd,
c      &      poros, rhob, reta, rc, nlay, nrow, ncol, vrc)
C*Block 1:*****
c List of calling arguments
c ncomp - Total number of components
c nvrndata - 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 ml, mu - are for banded jacobian (not used)
c y - Concentration value of all component at the node [array variable
y(ncomp)]
c pd - jacobian matrix [(ncomp x ncomp) array]
c poros - porosity of the node
c reta - Retardation factor [ignore dummy reta values of immobile species]
c rhob - bulk density of the node
c rc - Stores spatially constant reaction parameters (can dimension upto 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:*****
c*      *Please do not modify this standard interface block*
      IMPLICIT NONE
      INTEGER ncol, nrow, nlay
      INTEGER ncomp, nvrndata, j, i, k, ml, mu
      INTEGER, SAVE :: First_time=1
      DOUBLE PRECISION y, pd, poros, rhob, reta
      DOUBLE PRECISION rc, vrc
      DIMENSION y(ncomp), pd(ncomp, ncomp), rc(100)
      DIMENSION vrc(ncol, nrow, nlay, nvrndata), reta(ncomp)
C*End of block 2*****
      write(*,*) "Use solver option#1 or 3"
      write(*,*) "Jacobian routine is not available for this module"
      STOP
      RETURN
      END

```

DLL Versus Non-DLL User-Defined Reaction Module

Numerous types of biological/geochemical reactions occur in subsurface environments and it is impossible to develop a code that could model all these reactions. Therefore, RT3D provides a versatile user-defined reaction option that allows the user to simulate reactive transport based on his/her own reaction kinetics. Using this option, any type of kinetically limited reactive transport problem can be formulated and solved. As discussed above, user-defined reactions in the RT3Dv2.5 environment are incorporated into the two Fortran subroutines rxns and jacrxns. At the compilation stage, these two routines are either compiled together with the rest of the RT3D code to build a single executable (which is the standard option for RT3Dv2.5) or (on

Microsoft® Windows® operating systems) they can be compiled into separate dynamic link libraries (DLLs). Fortran "make" files are provided with RT3Dv2.5 for building a non-DLL version of RT3D using the (Compaq) Visual Fortran compiler and a Fortran 90 compiler on a UNIX system.

HOW TO IMPLEMENT THE DLL OPTION

To create a version of RT3Dv2.5 that uses DLL files (which was the standard in version 1.0), the following modifications to the source code must be made. These modifications assume that the compiler is (Compaq) Visual Fortran. Note that the compiler directive may take the form of "!DEC\$" or "!MS\$", the latter of which is for Microsoft Fortran Powerstation. Visual Fortran appears to accept the "!MS\$" declaration, although this form may not be supported in future versions.

- Uncomment line# 53 of rt3dv25.f (USE DFLIB)
- Uncomment line# 54 of rt3dv25.f (USE DFPORT)
- Uncomment line# 1428 of rteqnv25.f (!DEC\$ ATTRIBUTES DLLIMPORT :: rxns)
- Uncomment line# 1445 of rteqnv25.f (!DEC\$ ATTRIBUTES DLLIMPORT :: jrxns)
- Uncomment line# 24 of rxns.f (!DEC\$ ATTRIBUTES DLLEXPORT :: rxns)
- Uncomment line# 26 of jacrxns.f (!DEC\$ ATTRIBUTES DLLEXPORT :: jrxns)

In the DLL version, the subroutines rxns.f and jacrxns.f should be compiled first as standalone DLLs (e.g., the Visual Fortran command is: `df -dll rxns.f`). The RT3Dv2.5 executable can then be compiled and linked against the rxns.lib and jacrxns.lib files that were created along with the DLLs. The DLL files (rxns.dll and jacrxns.dll) should be copied/moved to the same directory as the RT3Dv2.5 executable or to a location in the file search path. If the DLL file is not present in the same directory as the RT3Dv2.5 executable, Windows will search for the file in the directories in the search path. Confusion may arise if Windows finds a different DLL file (of the same name) than the one you thought was going to be used.

USING MODFLOW 2000

The distribution of RT3Dv2.5 comes with MODFLOW-96 for generating a flow solution. If the user wishes to employ MODFLOW 2000, two things must be done. First, compile MODFLOW 2000 and RT3Dv2.5 using the same compiler; the executable for MODFLOW 2000 distributed by the U.S. Geological Survey (<http://water.usgs.gov/software/modflow-2000.html>) is compiled with Lahey Fortran while the RT3Dv2.5 executable is compiled with Visual Fortran, resulting in incompatibilities. Second, in the LMT6 input file, you must specify the "Standard" header (or use a blank file) because RT3Dv2.5 does not support the "Extended" header; see U.S. Geological Survey Open-File Report 01-82 for more details.

ACKNOWLEDGMENTS

The RT3D software has significant applications in areas ranging from risk assessment of groundwater pollution, management of large-scale groundwater plumes, design of natural attenuation systems, and design of active bioremediation systems. Over the past nine years,

several researchers from various organizations have participated in the RT3D development project, and the key personnel and their contributions are acknowledged below:

- Dr. Prabhakar Clement (Director and principal developer, 1994 - present)
- Dr. Brian Hooker (Project management and technical support, 1994 - 1996)
- Dr. Rodney Skeen (Technical support, 1994 - 1999)
- Dr. Chunmiao Zheng (MT3D related technical support, 1996 - present)
- Dr. James Petersen (WSU project management and technical support, 1995 - 1997)
- Dr. Norman Jones (GMS related technical support, 1996 - present)
- Mr. Christian Johnson (Technical support, documentation, application of RT3D to waste sites, publicity, 1996 - present)
- Dr. Yunwei Sun (Debugging, development support, validation, and MT2RT support, 1996 - 1998)
- Mr. Mike Truex (Project management, funding, and application of RT3D to waste sites, 1999 - present)
- Dr. Tirtha Gautam (Code management and development support, 2001 - present)

BRIEF HISTORY OF RT3D

In 1993, the U.S. Department of Energy (U.S. DOE) funded a field project for researchers at the Pacific Northwest National Laboratory (PNNL) to demonstrate pilot-scale bioremediation of carbon tetrachloride at the U.S. DOE's Hanford site in southeastern Washington State. The RT3D code development activities were initially funded from February 1994 to September 1995 as a part of this bioremediation demonstration. Dr. Clement originally developed a one-dimensional finite-difference code (IDCART) for simulating bioreactive transport in a denitrifying soil column reactor. In winter 1995, the one-dimensional model was applied to simulate Hanford bioremediation scenarios, including carbon tetrachloride removal rates in a bench-scale soil column reactor. Later, the PNNL research team (including Drs. Clement, Hooker, Skeen, and Peyton) collaborated with Prof. Jim Petersen's group at Washington State University and used the 1-D code to model nutrient feeding strategies. The primary objective of the collaborative work was to optimize the nutrient feeding strategy to obtain better distribution of microbial growth at locations away from the nutrient injection point, thereby minimizing bioclogging at the injection well. Details of this preliminary modeling work and the mathematical algorithms were published in Clement et al. [1995], Clement et al. [1996a], Clement et al. [1996b], Franzen et al. [1997] and Clement et al. [1997].

In the summer of 1995, Dr. Clement modified the numerical algorithms and implemented the simple aerobic reaction modules of his 1-D code within the U.S. Environmental Protection Agency's (U.S. EPA) MT3D modelling framework. This resulted in the first prototype version of the RT3D software (Reactive Transport in 3-Dimensions). Simulation results based on the prototype code were presented to several prospective clients, including Parson Engineering Science. Based on their review comments, a detailed proposal was developed to standardize RT3D and to benchmark its capabilities at a national bioremediation test facility located at Dover Air Force Base, Delaware. The test facility was run by a joint team consisting of industry, research organizations, regulatory agencies, and consulting companies (e.g., U.S. DOE, Dow Chemical Company, DuPont, U.S. Geological Survey, U.S. EPA, a university consortium,

Dames and Moore, Geosync), collectively known as the Remediation Technologies Development Forum (RTDF). Dr. Brian Hooker championed the project development efforts and Dr. Clement led the modelling efforts related to the RT3D design tool development task. The RTDF work was funded through a DOE research grant over a three-year period (from October 1995 - September 1998). In 1996, Dr. Yunwei Sun was hired as a post-doctoral fellow at Washington State University (in Prof. Jim Petersen's group) to support the debugging efforts and to help benchmark the code against various analytical solutions. The second prototype version of RT3D, which supported simple recharge and well packages, was finalized in the Fall of 1996. This code ran with MT2RT, a data pre-processor developed by Dr. Sun, which facilitated the use of the RT3D within a beta version of the U.S. Department of Defense Groundwater Modelling System (GMS 1.0) software. Dr. Sun also lead the code validation efforts by testing RT3D results against various analytical solutions, including a new solution that he developed as part of this project effort. Further, Mr. Christian Johnson at PNNL tested the prototype versions of RT3D and MT2RT codes and developed a series of test examples. The PNNL-WSU team jointly conducted a training course in November 1996 to disseminate the technology to various RTDF members.

In winter 1997, a subcontract was signed between Battelle and the University of Alabama to revise the prototype version to include all MODFLOW source/sink packages (i.e., river, recharge, drain, etc.) and to standardize the code with a new version of MT3D. Dr. Clement and Dr. Zheng jointly worked at the University of Alabama and developed a multi-species version of the MT3D_DoD_1.5 code. Dr. Clement later ported all reaction modules into this multi-species code. The first stand-alone, beta versions of RT3D (v 1.0) and the batch reaction utility BATCHRXN were developed in May 1997 and the details of these codes were documented in Clement [1997].

In 1998, the RT3D software was transferred to several industrial partners as a public-domain code, which facilitated integration of RT3D with commercial graphical user interfaces. Dr. Clement and Mr. Johnson managed key collaborative research efforts between PNNL and Brigham Young University's Environmental Modeling Research Laboratory (EMRL), the developers of GMS, to standardize the GMS interface for RT3D. A detailed user document was developed to facilitate the use RT3D within the GMS 2.1 environment [Clement and Jones, 1998]. More recently, Dr. Clement received a grant from the Australian Research Council, with some additional industrial support from Battelle, to continue the development work related to the RT3D software. As a part of this project, a new version of RT3D (v. 2.5) was released in November 2001. RT3D (v.2.5) is now a standard Fortran-90 code and should be compilable using any Fortran-90 complier.

REFERENCES AND RT3D-RELATED JOURNAL PUBLICATIONS

- Clement, T.P. 2001 "A Generalized Analytical Method for Solving Multi-Species Transport Equations Coupled with a First-Order Reaction Network." *Water Resources Research*, 37(1):157-163.
- Clement, T.P., C.D. Johnson, Y. Sun, G.M. Klecka, and C. Bartlett. 2000. "Natural Attenuation of Chlorinated Solvent Compounds: Model Development and Field-Scale Application." *Journal of Contaminant Hydrology*, 42(2-4):113-140.
- Clement, T.P. and N.L. Jones. 1998. *RT3D Tutorials for GMS Users*. Pacific Northwest National Laboratory, Richland, Washington, USA. PNNL-11805. Found online at: <http://bioprocess.pnl.gov/rt3d.htm>.
- Clement, T.P., Y. Sun., B.S. Hooker, and J.N. Petersen. 1998. "Modeling Multi-species Reactive Transport in Groundwater Aquifers." *Groundwater Monitoring & Remediation*, 18(2):79-92.
- Clement, T.P., B.M. Peyton, T.R. Ginn, and R.S. Skeen. 1999. "Modeling Bacterial Transport and Accumulation Processes in Saturated Porous Media: A Review," in *Advances in Nuclear Science and Technology*, J. Lewins and M. Becker, eds. Kluwer Academic/Plenum Publishers, New York. pp. 59-78.
- Clement, T.P., B.M. Peyton, R.S. Skeen, B.S. Hooker, J.M. Petersen, and D. Jennings. 1997. "Microbial Growth and Transport in Porous Media Under Denitrification Conditions: Experiment and Simulations Results." *Journal of Contaminant Hydrology*, 24:269-285.
- Clement, T.P. 1997. *RT3D - A Modular Computer Code for Simulating Reactive Multi-Species Transport in 3-Dimensional Groundwater Aquifers*. Pacific Northwest National Laboratory, Richland, WA, USA. PNNL-11720. Found online at: <http://bioprocess.pnl.gov/rt3d.htm>.
- Clement, T.P., B.M. Peyton, R.S. Skeen, B.S. Hooker, J.M. Petersen, and D. Jennings. 1997. "Microbial Growth and Transport in Porous Media Under Denitrification Conditions: Experiment and Simulations Results." *Journal of Contaminant Hydrology*, 24:269-285.
- Clement, T.P., B.S. Hooker, and R.S. Skeen. 1996b. "Macroscopic Models for Predicting Changes in Saturated Porous Media Properties Cause by Microbial Growth." *Ground Water*, 34(5):934-942.
- Clement, T.P., B.S. Hooker, and R.S. Skeen. 1996a. "Numerical Modeling of Biologically Reactive Transport From a Nutrient Injection Well." *ASCE Journal of Environmental Engineering*, 122(9):833-839.
- Clement, T.P., B.S. Hooker, and R.S. Skeen. 1995 "Modeling Biologically Reactive Transport in Porous Media," in Proceedings of the International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analyses, Portland, Oregon, April-May 1995, Vol. 1., pp. 192-201.

- Franzen, M.F.L., J.M. Petersen, T.P. Clement, B.S. Hooker, and R.S. Skeen. 1997. "Pulsing as a Strategy To Achieve Large Biologically Active Zones During In Situ Carbon Tetrachloride Remediation." *Computational Geosciences*, 1(3-4):217-288.
- Lu, G., T.P. Clement, C. Zheng, and T.H. Wiedemeier. 1999. "Natural Attenuation of BTEX Compounds: Model Development and Field-Scale Application." *Ground Water*, 37(5):707-717.
- Peyton, B.M., T.P. Clement, and J.P. Connolly. 2000. "Modeling of Natural Remediation: Contaminant Fate and Transport," Chapter 5 in *Natural Remediation of Environmental Contaminants*, M. Swindoll, R.G. Stahl, and S.J. Eells, eds. Society for Environmental and Toxicology and Chemistry, Pensacola, Florida. pp. 79-120.
- Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. 1992. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. Cambridge University Press, New York.
- Sun, Y., J.N. Petersen, T.P. Clement, and R.S. Skeen. 1999. "Development of Analytical Solutions for Multi-Species Transport with Serial and Parallel Reactions." *Water Resources Research*, 35(1):185-190.
- Sun, Y., J.N. Petersen, J. Bear, T.P. Clement, and B.S. Hooker. 1999. "Modeling Microbial Transport and Biodegradation in a Dual-Porosity System." *Transport in Porous Media*, 35(1):49-65.
- Sun, Y. and T. P. Clement. 1999. "A Generalized Decomposition Method for Solving Coupled Multi-Species Reactive Transport Problems." *Transport in Porous Media*, 37(3):327-346.
- Sun, Y., J.N. Petersen, T.P. Clement, and B.S. Hooker. 1998. "Effect of Reaction Kinetics on Predicted Concentration Profiles During Subsurface Bioremediation." *Journal of Contaminant Hydrology*, 31:359-372.
- Zheng, C. and P. Wang. 1999. *MT3DMS: A Modular Three-Dimensional Multispecies Transport Model for Simulation of Advection, Dispersion, and Chemical Reactions of Contaminants in Groundwater Systems; Documentation and User's Guide*. U.S. Army Corps of Engineers, U.S. Army Engineer Research and Development Center, Vicksburg, Mississippi. SERDP-99-1.