GMS 7.0 TUTORIALS

RT3D – User-Defined Reactions

The objective of this chapter is to describe the steps involved in developing a user defined reaction module for simulating reactions that can not be simulated using a pre-defined RT3D-reaction package. Once developed, users can distribute the new reaction package to other interested modelers for applying the reaction model at different sites (i.e., it can be treated like a pre-defined reaction package). Note that RT3D (version 2 beta) has an option to support a new reaction solver that uses Jacobian matrices. However, this option will not be discussed in this example. Interested users should refer to the latest RT3D documentation.

1.1 Contents

1.2 Outline

This is what you will do:

1. Import a MODFLOW model.
2. Define conditions.
3. Run MODFLOW.
4. Run RT3D.
5. Modify the model.
6. Run RT3D.

1.3 Required Modules/Interfaces

You will need the following components enabled to complete this tutorial:

- Grid
- MODFLOW
- RT3D

You can see if these components are enabled by selecting the *File* | *Register* command.

2 Steps Involved in Creating a User-Defined Reaction

User-defined reaction packages can be created using one of two approaches: the dynamically linked library (DLL) option or the linked subroutine option. With the DLL option, the subroutine for the reaction package is compiled as a stand-alone DLL. The DLL is then copied to the same directory as the RT3D executable and the DLL is

automatically launched by RT3D when RT3D is executed. The DLL can be recompiled or a new DLL can be used simply by copying the new DLL to the same directory as the RT3D executable and naming it appropriately. The RT3D executable does not need to be recompiled.

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.

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* platform. It is not available on *Unix*.

This tutorial describes the DLL approach for creating user-defined reaction packages. Special instructions for the *Unix* platform are contained in Section 17.

3 Required Background

As a reaction module developer, you are considered 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 GMS and be able to create MODFLOW and RT3D 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 A Digital FORTRAN compiler (or a UNIX system based FORTRAN 90 compiler).
- Should have some background/understanding of biochemical reaction kinetics. Note 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.

4 Reaction Module Description

The flow and transport model we will be developing in this tutorial exercise is identical to the one described in the tutorial entitled: *RT3D – Sequential Anaerobic Degradation*.

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:

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

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

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

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

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 of differential equations:

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

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

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

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

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

5 Coding a New Reaction Package

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 since 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 of the reaction parameter values are spatially constant but are assigned/modified externally, as input data, via the *.RCT file (to run RT3D) or 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!

We will first use method #1 in this tutorial; later, we will modify the code to demonstrate method #3.

5.1 Editing the Source Code of Reaction Module

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 *******
c234567890123456789012345678901234567890123456789012345678901234567890123456789012
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

2. This block contains a definition for all the variables that are passed into the rxns routine from the RT3D main program.

3. The interfacing block

4. This block contains FORTRAN statements that define the type of passed variables. This block should not be altered under any circumstances.

5. The local variable definition block

6. This block is used to define all the new local variables that are used exclusively in the rxns subroutine.

7. The reaction parameter initialization block

8. 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.

9. Variables naming block

10. This block is typically used to transfer certain passed variables into meaningful names (this is required for code clarity purposes only).

11. The differential equations block

12. This block is used to describe the PCE degradation reaction equations in FORTRAN language.

The FORTRAN subroutine presented above is provided with the GMS files in the following location:

```
tutfiles\RT3D\userdef\rxns_1.f
```

You may wish to open and view this file using your favorite ASCII editor (such as *Notepad* or *WordPad*).

5.2 Compiling the Source Code

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.F was saved and type the following command (this assumes that you have previously installed the *Digital FORTRAN* 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). Change the name of the DLL to RXNS.DLL

If you do not have access to a FORTRAN compiler but wish to continue with the tutorial, you can find a pre-compiled copy of rxns_1.dll in the tutfiles\RT3D\userdef directory.

6 Testing and Debugging a New Reaction Package

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.

In addition to testing the new reaction module using BATHCHRXN, it is also a good practice to test the transport problem using RT3D in single-species tracer mode. Once a reaction package is tested using BATCHRXN, the DLL can be used with RT3D to run a full-fledged RT3D simulation.

6.1 Running BATCHRXN

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 GMS directory (this is 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 into the GMS directory.

After you have copied the RXNS.DLL file into the GMS 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 in this case*
0.0     *Initial TCE concentration in this case*
0.0     *Initial DCE concentration in this case*
0.0     *Initial VC concentration in this case*
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.

6.2 Viewing the 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 Figure 1.

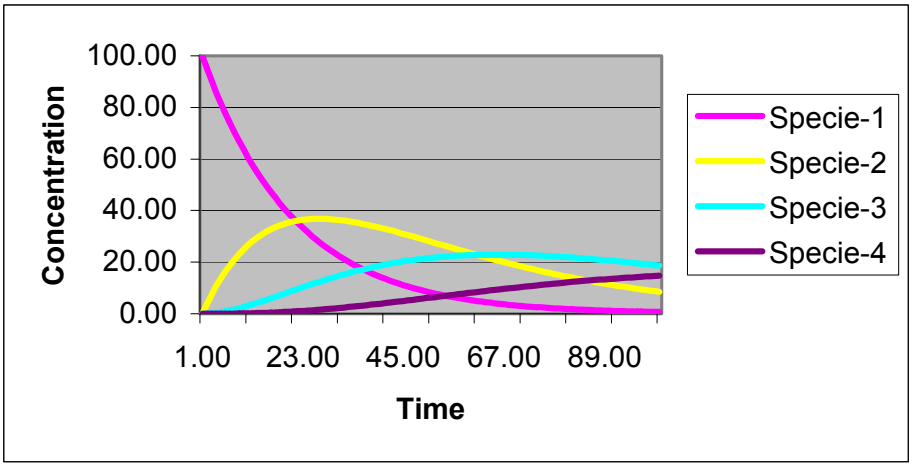

Figure 1. Sample Output from the BATCHRXN Code for the Sample Problem.

The results of a BATCHRXN run should be used to:

1. Debug errors in reaction expressions.
2. Check whether the reaction expressions and parameter values are reasonable (if available, microcosm or batch reactor data may be used to verify this).
3. Select appropriate values for numerical parameters atol (absolute tolerance) and rtol (relative tolerance).

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 a variable y(i), according to the inequality: max-norm of [e(i)/ewt(i)] is less than or equal to 1.0, where 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).

We recommend 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: since transport is not solved while executing batchrxn, all the retardation factors in the array reta() are automatically set to 1.0 with the DLL. However, if the same DLL is

used with RT3D, appropriate species retardation factors will be transferred through the reta() array.

7 Setting up the RT3D Simulation

At this stage, we are ready to run a RT3D simulation using the newly developed reaction package. To do this, will define the transport model and enter the input data using the GMS interface. Once the model is built, we will save it to a set of RT3D input files and launch RT3D. When the user-defined reaction package option is selected, RT3D automatically launches the RXNS.DLL subroutine to solve the reaction terms.

8 Getting Started

If you have not yet done so, launch GMS. If you have already been using GMS, you may wish to select the *New* command from the *File* menu to ensure the program settings are restored to the default state.

9 Importing the MODFLOW Model

The first part of the simulation is to import the MODFLOW flow model. A steady state flow model has been previously computed and is supplied with the tutorial files.

1. In the *Open* dialog, locate and open the file entitled **tutfiles\RT3D\flowmod\flowmod.gpr**.

At this point, you should see a grid appear.

10 Building the Transport Model

Now that the flow model is established, the next step is to perform the RT3D simulation. For this part of the simulation, we will select the user-defined reaction option, define and initialize appropriate species and reaction parameters, and assign concentrations to the well.

10.1 Initializing the Model

To initialize the RT3D data:

1. In the *Project Explorer* expand the *3D Grid Data* folder  .
2. Right-click on the 3D grid  and select the *New MT3D* command from the pop-up menu.

10.2 The BTN Package

The first step is to modify the data in the Basic Transport Package. First, we will select the user-defined reaction module option and name appropriate species and initialize their concentrations.

1. In the *Model* section, select the *RT3D* option.
2. Select the *Packages* button.
3. Turn on the following packages:
   - *Advection Package*
   - *Dispersion Package*
   - *Source/Sink Mixing Package*
   - *Chemical Reaction Package*
4. For the reaction type, select the *User-defined Reaction* option.
5. Select the *OK* button to exit the *Packages* dialog.

Defining the Species

For a pre-defined reaction package, the appropriate list of species would be automatically loaded by GMS. However, for user-defined reactions, we must explicitly define the species.

1. Select the *Define Species* button.
2. If a name file appropriate to the user-defined reaction package is available, we can click on the *Import* button to initialize all species names in a single step (see Section 16). Since it is not available (we are using this new module for the first time), we have to define the species manually. To begin this process, click on the *New* button.
3. Change the name of the species to **PCE**.
4. Leave the mobile option checked (this is the default).
5. Click on *New* 3 more times and give the species the following names:
   - TCE
   - DCE
   - VC
6. Select the *OK* button.

Notice that the *Starting Concentration* section of the dialog now lists all the four species associated with our new reaction module. By default all of the starting concentrations are zero. Since the aquifer is assumed to be initially clean, we will simply accept the default values for starting concentrations.

Porosity

Next, we will define the porosity as 0.3. Since this is the default supplied by GMS, no changes need to be made.

Setting up the Stress Periods

Next, we will define the stress periods. Since the injection rate and the boundary conditions do not change, we will use a single stress period with a length of 730 days (two years).

1. Select the *Stress Periods* button.
2. Select the *Initialize* button. If warned that this will overwrite all current boundary condition data, click *Yes* to continue.
3. Enter the values as shown in the following figure.

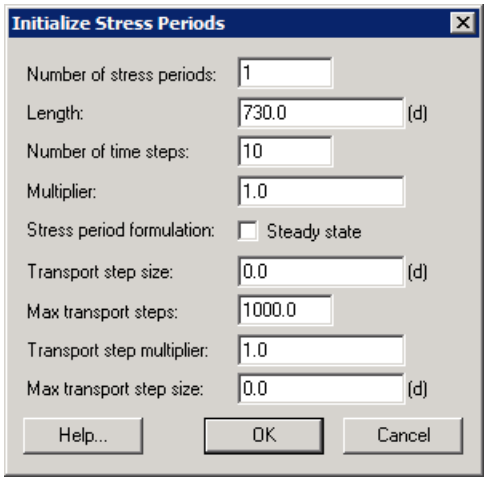

Figure 2. Initialize Stress Periods dialog.

4. Click *OK* to exit the *Initialize Stress Periods* dialog.
5. Select the *OK* button to exit the *Stress Periods* dialog.

Output Options

Finally, we will define the output options. One binary solution file is created by RT3D for each of the species. By default, RT3D saves a solution at each transport step for each species. Since this results in large files containing more solutions than we need for the simple post-processing we intend to do, we will specify that a solution be saved every 73 days (every time step).

1. Select the *Output Control* button.
2. Select the *Print or save at specified times* option.
3. Select the *Times* button.
4. Select the *Initialize Values* button.
5. Enter **73.0** for the *Initial time step size*.
6. Enter **73.0** for the *Maximum time step size*.
7. Enter **730.0** for the *Maximum simulation time*.
8. Select the *OK* button to exit the *Initialize Time Steps* dialog.
9. Select the *OK* button to exit the *Variable Time Steps* dialog.
10. Select the *OK* button to exit the *Output Control* dialog.

This completes the input for the Basic Transport package.

11. Select the *OK* button to exit the *Basic Transport Package* dialog.

10.3 The Advection Package

The next step is to initialize the data for the Advection package.

1. Select the *MT3D* | *Advection Package* command.
2. Select the *Standard finite-difference method* option.
3. Select the *OK* button to exit the dialog.

10.4 The Dispersion Package

Next, we will enter the data for the Dispersion package. The aquifer has a longitudinal dispersivity of 10.0 m and a transverse (horizontal) dispersivity of 3.0 m.

1. Select the *MT3D* | *Dispersion Package* command.
2. Enter a value of **0.3** for the *TRPT* value.
3. Select the *Longitudinal Dispersivity* button.
4. Select the *Constant* → *Grid* button.
5. Enter a value of **10.0** and select *OK*.
6. Select the *OK* button to exit the *Longitudinal Dispersivity* dialog.

7. Select the *OK* button to exit the *Dispersion Package* dialog.

10.5 The Source/Sink Mixing Package

Next, we will initialize the Source/Sink Mixing package and define the concentration at the spill location. We will define the concentrations at the wells by selecting the cells and assigning the concentration directly with the *Point Source/Sink* command. We will assign a concentration of 1000 mg/L for PCE and leave TCE, DCE, and VC concentrations at the default value of 0.0.

1. Select the *Select Cell* tool  .
2. Select the well at the spill location.
3. Select the *MT3D* | *Point Sources/Sinks* command.
4. Verify that *PCE (mobile)* is selected in the top left corner of the dialog.
5. Turn on the toggle in the *Well* section.
6. In the *Well* section, enter a value of **1000.0** for concentration. Leave the concentrations for the other species at zero.
7. Select the *OK* button to exit the *Point Source/Sink* dialog.

10.6 The Chemical Reaction Package

Next, we will review the data for the Chemical Reaction package.

1. Select the *MT3D* | *Chemical Reaction Package* command.

Note that the reaction parameter box is blank. We will leave it blank in this case because in the DLL we coded all of the reaction constants directly within the reaction subroutine. We will demonstrate how to define them as user-defined variables later in this tutorial.

2. Select the *OK* button to accept the defaults.

11 Run MODFLOW

Before running RT3D, we will regenerate the MODFLOW solution.

1. Select the *File* | *Save As* command.
2. In the *Save As* dialog, locate and open the directory entitled **tutfiles\RT3D\userdef**.
3. Enter **"rtmod1"** for the file name.

4. Select the *Save* button to save the files.

To run MODFLOW:

5. Select the *MODFLOW* | *Run MODFLOW* command.
6. When the simulation is finished, close the window.

12 Running RT3D

At this point, we are ready to save the model and run RT3D.

To run RT3D:

1. Select the *MT3D* | *Run RT3D* command.
2. Select *Yes* at the prompt to save your changes.
3. When the simulation is finished, select the *Close* button.

GMS should read in the solution automatically.

13 Viewing the Results

To verify your results, compare the concentration values with the output from the tutorial entitled: *RT3D – Sequential Anaerobic Degradation*.

14 Developing a Reusable Reaction Package

Defining a new reaction model via a reaction package is a powerful option supported by the RT3D code. Once tested and developed, the reaction package (or DLL) can be used in other projects that have different conceptual flow and transport models. DLLs can also be distributed via web pages to other users who might be interested in modeling similar type of reactive transport systems. This section outlines the steps involved in developing a reusable version of reaction package.

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 are ready to distribute the package to other users who might be interested in using it 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 (i.e., in the *Chemical Reactions Package* dialog in GMS).

14.1 Editing the Source Code

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} (note NCRXNDATA = 5), and two of them are spatially variable parameters: K_{dce} and K_{vc} (note 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 *******
c234567890123456789012345678901234567890123456789012345678901234567890123456789012
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
```

Note that 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. In the section below, we will demonstrate how the reaction parameter values can be defined efficiently using the GMS interface.

14.2 Compiling the Source Code

Now follow the steps detailed in Section 5 to compile and create a new rxns.dll file. The FORTRAN source code is provided in the following file:

```
tutfiles\RT3D\userdef\rxns_2.f
```

Once the source is compiled, rename the DLL as rxns.dll and copy it to the GMS subdirectory where RT3D.EXE is saved. If you wish to preserve the previous version of the rxns.dll file, you will need to rename the old version before copying the new version to the GMS directory.

Once again, if you do not have access to a FORTRAN compiler but wish to continue with the tutorial, you can find a pre-compiled copy of rxns_2.dll in the tutfiles\RT3D\userdef directory.

15 Modifying the Transport Model

We are now ready to use the new module for reactive transport simulations. If you have exited GMS, or if you have made any changes to the RT3D simulation defined above, you will need to re-import both the MODFLOW and RT3D simulation.

15.1 The Chemical Reaction Package

Next, we will define and initialize the chemical reaction parameters.

1. Select the *MT3D* | *Chemical Reaction Package* command.

Note that the reaction parameter box is blank. In this section, we will define and initialize the reaction parameters.

2. Click on the *Define Parameters* button.

3. Once again, if a name file appropriate to the user-defined reaction package is available, we can click on the *Import* button and initialize all reaction parameter names in a single step (see Section 16). Since it is not available in this case, we will define them manually.

4. Use the *New* button to create all the parameters as shown in the following figure. Note that the dialog automatically sorts the parameters such that those that vary spatially are listed at the top.

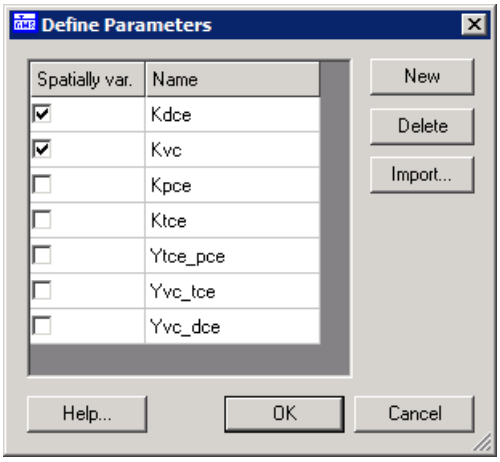

Figure 3. Define Parameters dialog.

5. Select the *OK* button.

Now notice in the *Define Parameters* section all the reaction parameters are listed. Next, we will assign values for all these parameters.

6. In the *Reaction Parameters* section, click on K_{pce} and set the constant value to **0.005**.

7. Click on K_{tce} and set the constant value to **0.003**

8. Click on Y_{tce_pce} and set the constant value to **0.792**

9. Click on Y_{dce_tce} and set the constant value to **0.738**

10. Click on Y_{vc_dce} and set the constant value to **0.644**

11. Now, for K_{dce} *(variable)* select the *Edit* button.

12. Click the *Constant* → *Grid* button.

13. Enter a value of **0.002** and select *OK*

14. Select the *OK* button to exit the *Reaction Parameters Array* dialog.

15. For K_{vc} *(variable)* select the *Edit* button.

16. Select the *Constant* → *Grid* button.

17. Enter a value of **0.001** and select *OK*.

18. Select the *OK* button to exit the *Reaction Parameters Array* dialog.

19. Select the *OK* button to exit the *RT3D Chemical Reaction Package* dialog.

We have defined five spatially constant reaction parameters and two spatially variable reaction parameters. Therefore, in the *.RCT package file, NCRXNDATA will be set to 5 and NVRXNDATA will be set to 2. During execution, the RT3D code stores and transfers all the constant reaction parameters through a one-dimensional reaction constant array rc(). The parameter values are transferred in the order in which they are originally entered; i.e. K_{pce} is stored in rc(1), K_{tce} is stored in rc(2) and so on. Similarly, all the variable reaction parameters are stored and transferred through a four-dimensional variable reaction constant array vrc().

Note that although two of the reaction parameters can vary spatially, for the sake of keeping this tutorial simple, we have assigned a single value to all cells in both cases. The reaction parameters can be edited on a cell by cell basis using the *Array Editor* dialog in the *Chemical Reaction Package* dialog or by selecting a set of cells and selecting the *Cell Properties* command in the *MT3D* menu.

15.2 Select MODFLOW Solution

Since the MODFLOW solution will not have the same name as the RT3D simulation, we need to specify the MODFLOW solution used by RT3D.

1. Select the *MT3D* | *Run Options* command.

2. Select the *Single run with selected MODFLOW solution* option, choose the *rtmod1 (MODFLOW)* solution, and select *OK*.

15.3 Running RT3D

At this point, we are ready to save the model to run RT3D.

1. Select the *File* | *Save As* command.

2. Check to ensure that the directory is still set to **tutfiles\RT3D\userdef**.

3. Enter "**rtmod2**" for the file name.

4. Select the *Save* button to save the files.

To run RT3D:

5. Select the *MT3D* | *Run RT3D* command.

6. When the simulation is finished, close the window and return to GMS.

GMS should read the solution automatically. The results should be similar to those computed earlier.

16 Creating a Name File for GMS import

When a new reaction package is created, the package developer should create a name file. The information regarding species names, species types, species order, reaction parameter names, reaction parameter types, and reaction parameter sequence are stored in a name file. All of the information needed for a name file is included in the RT3D super file created by GMS. Thus, the simplest way to create the name file is to set up the first simulation using the steps defined above and then open the RT3D super file for the simulation (*.rts) in a text editor. Delete all the cards except the first card, and the cards beginning with SPC and RXNPARAM. The edited file should look like this:

```
RT3DSUP
SPC "PCE" 1 1
SPC "TCE" 2 1
SPC "DCE" 3 1
SPC "VC" 4 1
RXNPARAM "kdce" 3 1
RXNPARAM "kvc" 4 1
RXNPARAM "kpce" 1 0
RXNPARAM "ktce" 2 0
RXNPARAM "Ytce_pce" 5 0
RXNPARAM "Ydc_tce" 6 0
RXNPARAM "Yvc_dce" 7 0
```

Save this information in a new file, say “specie.nam”. This file can be later used for directly importing species and reaction parameter information. Whenever a user-defined reaction package (as a DLL or source code) is distributed to another GMS user, a specie name file, appropriate to the reaction package, should be included with it.

16.1 Importing Species Names

The specie name file can be used to set-up a new RT3D transport problem using a (previously developed) user-defined reaction package. To import the specie names into GMS, select *Basic Transport Package*, click on the *Define Species* option, click on the *Import* button, and read the “specie.nam” file. These steps will import appropriate specie names in the correct order.

16.2 Importing Reaction Parameter Names

To import reaction parameter names, select the *Chemical Reaction Package* command, click on the *Define Parameters* option, click on the *Import* button, and read the “specie.nam” file again. This will import appropriate reaction parameter names in the correct order.

17 Special Instructions for UNIX Users

Since development of a DLL requires *Microsoft* or *Digital's visual FORTRAN* compiler specific commands, this option 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 a command "!MS$ATTRIBUTES". Also whenever any 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).

This concludes the tutorial.