

# **Interactive Groundwater Modeling**

## ***Guide To Visual Fortran-Based DLL Source Code***

By

Huasheng Liao

Visiting Professor  
Department of Civil and Environmental Engineering  
Michigan State University

February, 2002

# Table of Contents

1. INTRODUCTION .....	4
2. TWO DIMENSIONAL MODEL SOURCE CODE .....	5
2.1 Flow .....	5
2.1.1 Governing Equation .....	5
2.1.2 Grid Layout .....	7
2.1.3 Scheme to Discretize Equation .....	10
A) Traditional Control Volume Technique .....	10
a) Approximation to Diffusion Term .....	10
b) Approximation to Time Derivative Term .....	12
c) Approximation to Source/Sink Term .....	12
d) Coefficient Matrix Assembling .....	13
e) Matrix Solver .....	13
B) Rotational Control Volume Technique .....	13
a) Approximation to Diffusion Term .....	16
b) Approximation to Time Derivative Term .....	19
c) Approximation to Source/Sink Term .....	20
d) Coefficient Matrix Assembling .....	20
e) Matrix Solver .....	21
2.1.4 Special Treatments .....	22
2.1.5 Matrix Solver .....	23
2.1.6 Numerical Solution Procedure Flow Chart .....	24
2.2 Transport .....	25
2.2.1 Governing Equation .....	25
2.2.2 Grid Layout .....	27
2.2.3 Scheme to Discretize Equation .....	28
A) Full Implicit Finite Difference Scheme .....	28
a) Approximation of Advection Term .....	28
b) Traditional Control Volume Technique to Approximate Diffusion Term ..	29
c) Rotational Control Volume Technique to Approximate Diffusion Term .....	31
d) Approximation of Time-Derivative term .....	33
e) Approximation of Source/Sink Term .....	34
f) Approximation of Decay Term .....	34
g) Coefficient Matrix Assembling and Solution Technique .....	35
B) Mixed Eulerian-Lagrangian Methods .....	36
a) Particle Tracking Technique .....	36
b) Determination of $C_p^{n*}$ .....	40
c) Approximation of Diffusion Term .....	43
d) Approximation of Time-Derivative Term .....	45
e) Approximation of Source/Sink Term .....	45
f) Approximation of Decay Term .....	45
g) Coefficient Matrix Assembling and Solution Technique .....	46

- C) Full Lagrangian Method— Random Walk ..... 47
  - a) Determination of Advective Displacement..... 47
  - b) Determination of Dispersive Random Displacement ..... 48
  - c) Final Displacement ..... 48
  - d) Evaluation of Concentration..... 48
- 2.2.4 Special Treatments..... 50
- 2.2.5 Matrix Solver ..... 51
- 2.2.6 Numerical Solution Procedure Flow Chart..... 52
- 2.3 Monte Carlo Simulation..... 53
  - 2.3.1 Random Field Generator..... 53
  - 2.3.2 Governing Equations ..... 56
  - 2.3.3 Calculation of Statistical Distributions ..... 57
    - A) Point-Based Statistics ..... 57
    - B) Field-Based Statistics..... 60
  - 2.3.4 Special Treatments..... 61
  - 2.2.5 Numerical Solution Procedure Flow Chart..... 62
- 2.4 One-Layer-Based Profile Model..... 63
  - 2.4.1 Governing Equation..... 63
  - 2.4.2 Discretizing Conceptual Features of Profile Model..... 67
    - A) Approximation to Point-Related Conceptual Feature..... 67
    - B) Approximation to Poly-line-Related Conceptual Feature..... 69
    - C) Determination of Computational Domain ..... 72
      - a) Upper Boundary..... 72
      - b) Lower Boundary..... 72
      - c) Computational Domain..... 72
      - d) Inactive Cell..... 72
      - e) Boundary Conditions ..... 72
  - 2.4.3 Special Treatments..... 75
  - 2.4.4 Numerical Solution Procedure Flow Chart..... 76
- 3. THREE DIMENSIONAL MODEL SOURCE CODE ..... 77
  - 3.1 Flow ..... 77
    - 3.1.1 Governing Equation..... 77
    - 3.1.2 Grid Layout..... 79
    - 3.1.3 Scheme to Discrete Equation..... 85
      - A) Traditional Control Volume Technique..... 85
        - a) Approximation to Diffusion Term..... 85
        - b) Approximation to Time Derivative Term..... 87
        - c) Approximation to Source/Sink Term..... 87
        - d) Matrix Coefficients Assembling..... 90
        - e) Matrix Solver ..... 90
      - B) Rotational Control Volume Technique..... 90
        - a) Approximation to Diffusion Term..... 93
        - b) Approximation to Time Derivative Term..... 99
        - c) Approximation to Source/Sink Term..... 100
        - d) Coefficient Matrix Assembling ..... 100
        - e) Matrix Solver ..... 101

3.1.4	Special Treatments.....	102
3.1.5	Numerical Solution Procedure Flow Chart.....	106
3.2	<i>Transport</i> .....	107
3.2.1	Governing Equation.....	107
3.2.2	Grid Layout.....	109
3.2.3	Scheme to Discretize Equation.....	110
A)	Full Implicit Finite Difference Scheme.....	110
a)	Approximation of Advection Term.....	110
b)	Traditional Control Volume Technique to Approximate Diffusion Term.....	112
c)	Rotational Control Volume Technique to Approximate Diffusion Term.....	114
d)	Approximation of Time-Derivative Term.....	120
e)	Approximation of Source/Sink Term.....	121
f)	Approximation of Decay Term.....	121
g)	Coefficient Matrix Assembling and Solution Technique.....	122
B)	Mixed Eulerian-Lagrangian Methods.....	124
a)	Particle Tracking.....	124
b)	Determination of $C_p^{n*}$ .....	130
c)	Approximation of Diffusion Term.....	133
d)	Approximation of Time-Derivative Term.....	139
e)	Approximation of Source/Sink Term.....	139
f)	Approximation of Decay Term.....	140
g)	Coefficient Matrix Assembling and Solution Technique.....	140
3.2.4	Special Treatments.....	143
3.2.5	Matrix Solver.....	144
3.2.6	Numerical Solution Procedure Flow Chart.....	145
3.3	<i>Chemical Reaction in Three Dimensional Model</i> .....	146
3.3.1	General Technique for Dealing with Chemical Reaction.....	146
3.3.2	Chemical Reaction Models.....	148
A)	Model 0.....	148
a)	Rate-Limited Sorption Reactions.....	148
b)	Dual Domain Model.....	148
c)	NAPL Dissolution Process.....	149
B)	Model 1: Instantaneous Aerobic Decay of BTEX.....	150
C)	Model 2: Instantaneous Degradation of BTEX using Multiple Electron Acceptors.....	150
D)	Model 3: Kinetic-Limited Degradation of BTEX Using Multiple Electron Acceptors.....	152
E)	Model 4: Rate-Limited Sorption Reactions.....	154
F)	Model 5: Double Monod Model.....	154
G)	Model 6: Sequential Decay Reactions.....	156
H)	Model 7: Aerobic-Anaerobic Model for PCE-TCE Degradation.....	157
3.3.3	Linearized Approach to Solving Chemical Reaction Equations.....	160
3.4	<i>Monte Carlo Simulation in Three Dimensional Model</i> .....	161
3.4.1	Random Field Generator.....	161
3.4.2	Value at Monitoring Well with Screen Thickness.....	162
3.4.3	Numerical Solution Procedure Flow Chart.....	164

# Interactive Groundwater Modeling

----- Guide to Visual Fortran-based DLL Source Code

**Huasheng Liao**  
**Visiting Professor**  
**Department of Civil and Environmental Engineering**  
**Michigan State University**

## 1. Introduction

From the programming point of view, IGW software includes two parts – Visual Fortran-based DLL and Visual Basic-based user interface coding. In this guide, an overall VF-based DLL coding structure and a couple of new numerical methods employing in this software for solving 2-D or 3-D flow and transport equations will be described in detail. This guide is organized in two sections -- guide to 2-D model source code and guide to 3-D model source code.

The purpose of this guide is to describe the mathematical concepts used in designing the flow and transport models in IGW and some special treatments. This is a guide only and it must be said that any guide such as the present can only deal with major aspects. Many details must remain unmentioned. Of these, the majority may not require discussion, but in some areas question marks may arise which will remain unanswered by the guide. It is hoped that such cases will be few and isolated. The code, it should be stressed, has been written with the inexperienced user constantly in mind, however the reader is still assumed to be aware of the basic numerical methods such as standard finite difference, control volume, MMOC, etc..

## 2. Two Dimensional Model Source Code

### 2.1 Flow

#### 2.1.1 Governing Equation

The partial differential equation describing flow in porous medial is usually written as

$$S_s \frac{\partial h}{\partial t} = \frac{\partial}{\partial X_i} (K_{ij} \frac{\partial h}{\partial X_j}) + q_s \quad (1)$$

Where

- $S_s$  storage coefficient of the porous materials
- $h$  hydraulic head
- $K_{ij}$  hydraulic conductivity tensor
- $X_i$  Cartesian coordinate.
- $q_s$  source/sink

With knowing head  $h$ , the seepage or linear pore velocity can be defined as

$$v_i = -\frac{K_{ij}}{n} \frac{\partial h}{\partial X_j} \quad (2)$$

Where  $n$  is the porosity of the porous medium.

The hydraulic conductivity tensor should have nine components in 3-D cases or four components in 2-D cases. Most of existing flow models such as MODFLOW involve only three of them in 3-D case or two in 2-D case, that is, assuming that the principal components of the hydraulic conductivity tensor,  $K_{ii}$  are aligned with the  $X_i$  coordinate axes so that those non-principal component (cross terms),  $K_{ij}$  ( $i \neq j$ ), become zero. In the real world, there are many cases of anisotropy aquifer in which  $K_{ij}$  may not be assumed to be zero any more. When  $K_{ij}$  is considered as part of the equation, it is very easy to give a negative coefficient in the derived discretized-matrix based on the Traditional Control Volume (TCV) technique. In order to avoid appearance of any 'negative coefficient' in the derived discretized-matrix, we have proposed a new technique – Rotation Control Volume(RCV) technique in our 2-D and 3-D flow and transport models.

Taking an integration of Eq(1) along the depth direction, gives us a depth-averaged governing equation of Eq(1) (usually referring to as 2-D equation) as follow,

$$S \frac{\partial h}{\partial t} = \frac{\partial}{\partial X_i} (T_{ij} \frac{\partial h}{\partial X_j}) + Q_s \quad , \quad i = 1, 2; j = 1, 2 \quad (3)$$

Where

- $S$  depth-averaged storage coefficient of the porous materials  
 $h$  depth-averaged hydraulic head  
 $T_{ij}$  transmissivity tensor (4 components)  
 $X_i$  Cartesian coordinate.  
 $Q_s$  source/sink, it could be kind of head dependent sources such as river or drain.

Equation (3) is used in IGW 2-D model. In IGW, only the first principal hydraulic conductivity,  $K'_{xx}$ , is given and the second one,  $K'_{yy}$ , is obtained from the given anisotropy ratio. When  $K'_{ii}$  are not aligned with the  $X_i$  coordinate axes, an anisotropy orientation angle,  $\theta$ , will be given in order to determine the rest of two components in the tensor  $K_{ij}$ . There is a formula to convert  $K'_{ii}$  to  $K_{ij}$  for 2-D case,

$$\begin{aligned}
 K_{xx} &= \frac{K'_{xx} + K'_{yy}}{2} + \cos 2\theta \frac{K'_{xx} - K'_{yy}}{2} \\
 K_{yy} &= \frac{K'_{xx} + K'_{yy}}{2} - \cos 2\theta \frac{K'_{xx} - K'_{yy}}{2} \\
 K_{xy} &= K_{yx} = \sin 2\theta \frac{K'_{xx} - K'_{yy}}{2}
 \end{aligned} \tag{4}$$

It is seen from Eq(4) that  $K_{xx}$  and  $K_{yy}$  are always positive, however,  $K_{xy}$  or  $K_{yx}$  may be negative. Conversion of  $T'_{ij}$  to  $T_{ij}$  has the same express as Eq(4).

### 2.1.2 Grid Layout

In IGW, parameters are assigned to a block or a cell. Placing a representative node in each cell forms the grid layout using in our spatial discretizing. Figure 1 shows a typical grid layout of current use.

A typical node-cell and its neighborhood is shown in Figure 2. The figure is self-explanatory, and those grid related geometrical quantities used in the scheme are illustrated in the figure too. The following is a list of the variables using in the code and their counterparts in this figure,

Table 1

	Notation in Figure 1	Variable in Code
X coordinate	$X_{ij}$	Xmesh(I,J)
Y coordinate	$Y_{ij}$	Ymesh(I,J)
X-Grid Spacing	$\Delta X_i$	HX(I)
Y-Grid Spacing	$\Delta Y_j$	HY(J)
X-CV area	$\Delta X_s$	DXS(I)
Y-CV area	$\Delta Y_s$	DYS(J)

It is noted that VF code is written based on non-uniform grid spacing, although a uniform grid spacing has been used in IGW interface. The quantities at the cell-face, such as  $k_{ij}^e$  in Figure 2, must be evaluated in terms of nodal values before proceeding the calculation. There are many methods available to handle it, such as those of linear interpolation, harmonic averaged, etc. Harmonic mean was adopted in IGW 2-D model. X-velocity  $u_x$  is located on the East and West cell-faces and Y-velocity  $u_y$  is located on the North and South cell-faces. This arrangement will make use of cell-face's  $K_{ij}$  which is the real one involving in the calculation, and central difference scheme to approximate the head gradient between two known nodal heads. To obtain the velocities at nodes, an interpolation method or simple arithmetic mean has to be used. Arithmetic mean has been used in IGW 2-D model due to its uniform grid system. Subroutine CALUVXJQT has the details. A linear interpolation should be used when the grid system was not uniform.

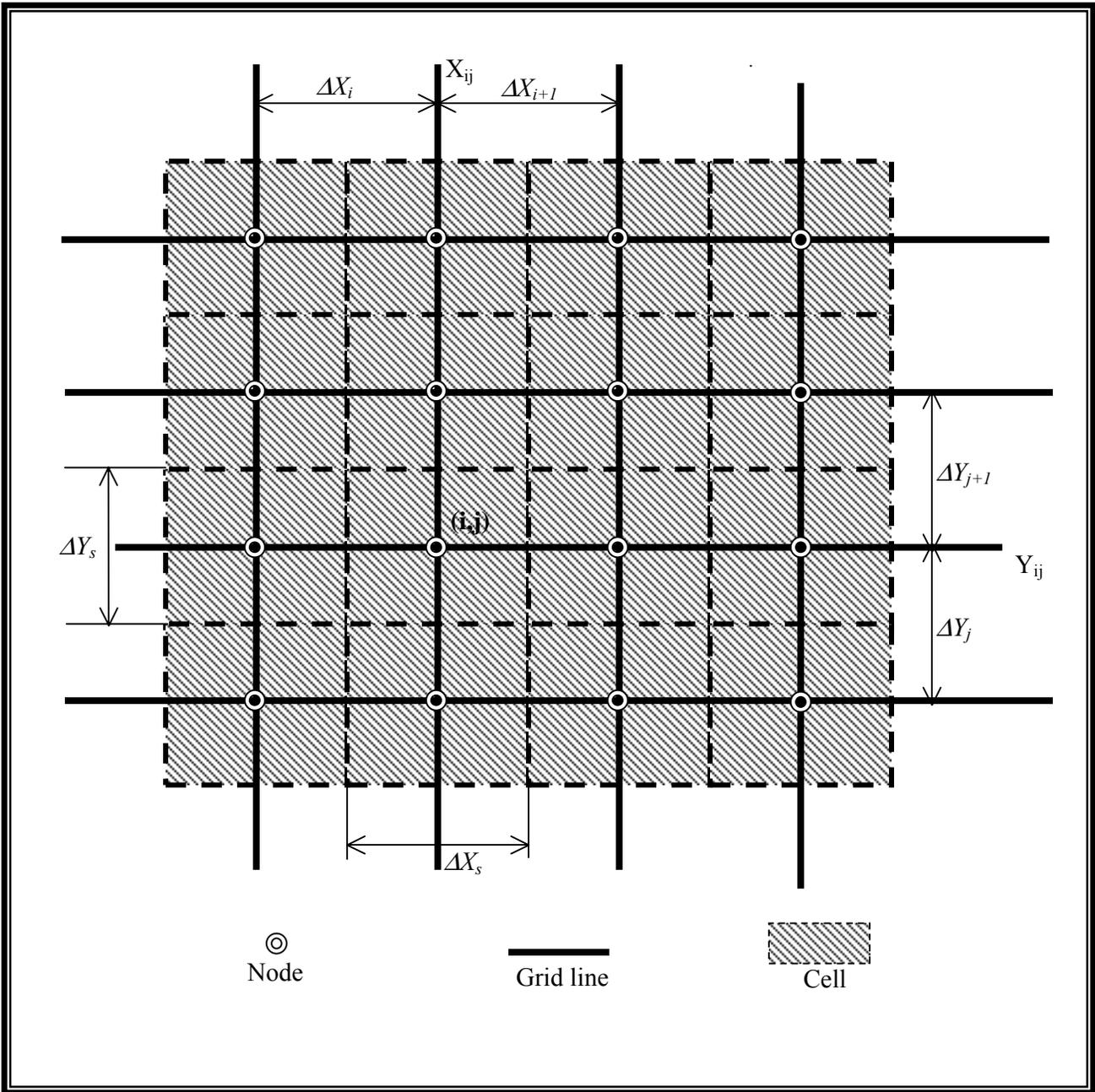


Figure 1. Grid Layout Using in IGW 2-D Model

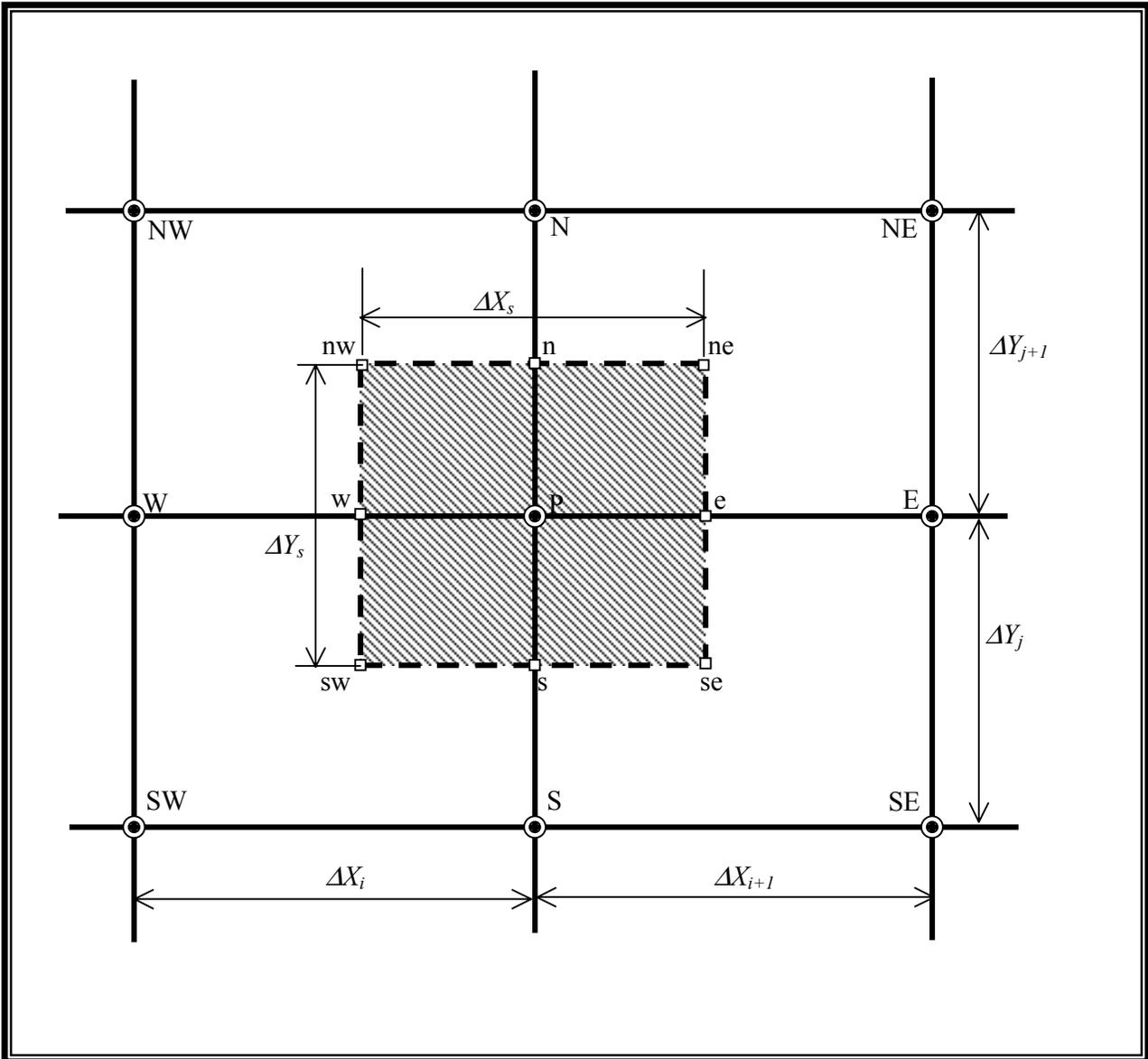


Figure 2. Typical Cell and Notation

### 2.1.3 Scheme to Discretize Equation

#### A) Traditional Control Volume Technique

As mentioned above, despite of the fact that  $T_{ij}$  easily causes a ‘negative coefficient’ in the discretized-matrix based on the traditional Control Volume (CV) technique, traditional control volume technique will still be employed in our IGW 2-D model for purpose of comparison.

#### a) Approximation to Diffusion Term

Referring to Figure 2 and applying control volume technique, diffusion terms of Eq(3) may be written as,

$$Diff = \Delta X_s \Delta Y_s \frac{J_e - J_w}{\Delta X_s} + \Delta X_s \Delta Y_s \frac{J_n - J_s}{\Delta Y_s} \quad (5)$$

Where  $J_e$ ,  $J_w$ ,  $J_n$  and  $J_s$  are the fluxes through east, west, north and south cell-faces:

$$\begin{cases} J_e = T_{xx}^e \frac{h_E - h_P}{\Delta X} + T_{xy}^e \frac{h_{ne} - h_{se}}{\Delta Y_s} \\ J_w = T_{xx}^w \frac{h_P - h_W}{\Delta X} + T_{xy}^w \frac{h_{nw} - h_{sw}}{\Delta Y_s} \\ J_n = T_{yy}^n \frac{h_N - h_P}{\Delta Y} + T_{yx}^n \frac{h_{ne} - h_{nw}}{\Delta X_s} \\ J_s = T_{yy}^s \frac{h_P - h_S}{\Delta Y} + T_{yx}^s \frac{h_{se} - h_{sw}}{\Delta X_s} \end{cases} \quad (6)$$

All symbols in Eq(6) were denoted in Figure 2.

Note that non-nodal quantities in Eq(6), such as  $h_{ne}$ ,  $h_{se}$ ,  $h_{nw}$ , and  $h_{sw}$ , must be evaluated in terms of nodal values. A simple four points averaged scheme has been used in IGW 2-D model. After re-arranging Eq(6), Eq(5) becomes

$$Diff = a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} - a_P h_P \quad (7)$$

Where  $a_i$  are called the coefficients of the discretized-matrix, they have the forms as below

$$\left\{ \begin{array}{l}
 a_E = \frac{\Delta Y_s T_{xx}^e}{\Delta X} + \frac{T_{yx}^n - T_{yx}^s}{4} \\
 a_W = \frac{\Delta Y_s T_{xx}^w}{\Delta X} - \frac{T_{yx}^n - T_{yx}^s}{4} \\
 a_N = \frac{\Delta X_s T_{yy}^n}{\Delta Y} + \frac{T_{xy}^e - T_{xy}^w}{4} \\
 a_S = \frac{\Delta X_s T_{yy}^s}{\Delta Y} - \frac{T_{xy}^e - T_{xy}^w}{4} \\
 a_{NE} = \frac{T_{xy}^e + T_{yx}^n}{4} \\
 a_{NW} = -\frac{T_{xy}^w + T_{yx}^n}{4} \\
 a_{SE} = -\frac{T_{xy}^e + T_{yx}^s}{4} \\
 a_{SW} = \frac{T_{xy}^w + T_{yx}^s}{4} \\
 a_P = a_E + a_W + a_N + a_S + a_{NE} + a_{NW} + a_{SE} + a_{SW}
 \end{array} \right. \quad (8)$$

The process of calculation of  $a_i$  was implemented in Subroutine COEFFLOW in the VF source code. A derived type variable CST2 was used to store these coefficients:

Table 2

Notation in Eq(8)	Variable in Code
$a_E$	CST2 (I,J)%SE
$a_W$	CST2 (I,J)%SW
$a_N$	CST2 (I,J)%SN
$a_S$	CST2 (I,J)%SS
$a_{NE}$	CST2 (I,J)%SNE
$a_{NW}$	CST2 (I,J)%SNW
$a_{SE}$	CST2 (I,J)%SSE
$a_{SW}$	CST2 (I,J)%SSW
$a_P$	CST2 (I,J)%SP

From Eq(8), it is very obvious that  $a_{NE}$ ,  $a_{NW}$ ,  $a_{SE}$  and  $a_{SW}$  may easily turn into negative ones, for example,  $a_{NW} < 0$  and  $a_{SE} < 0$  when  $T_{xy} > 0$ ; or  $a_{NE} < 0$  and  $a_{SW} < 0$  when  $T_{xy} < 0$ .

**b) Approximation to Time Derivative Term**

For transient flow, a backward finite difference scheme is used to approximate the time-derivative term in Eq(3). It can be read as

$$S \frac{\partial h}{\partial t} = S \frac{h^{n+1} - h^n}{\Delta t} \Delta X_s \Delta Y_s = a_p^t h^{n+1} - S_f^t \tag{9}$$

Where  $h^{n+1}$  is the head at new time level  $n+1$ ,  $h^n$  is the head at old time level  $n$  and

$$\begin{cases} a_p^t = \Delta X_s \Delta Y_s \frac{S}{\Delta t} \\ S_f^t = \Delta X_s \Delta Y_s \frac{S}{\Delta t} h^n \end{cases} \tag{10}$$

**c) Approximation to Source/Sink Term**

Source/sink term may include head independent one such as well or recharge and head dependent one such as river or drain. Head independent source/sink terms can be directly added to Right Hand Side (RHS) vector of the derived-matrix, however, head dependent source/sink terms must be divided into two parts: one goes to RHS, another one goes to the main diagonal entry after being linearized . In general, Q can be expressed as

$$Q = a_p^Q h_p + S_f^Q \tag{11}$$

Table 3 is a list of source/sink available in IGW 2-D model and their corresponding  $a_p^Q$  and  $S_f^Q$ .

Tabel 3

Type of Source/Sink	$a_p^Q$	$S_f^Q$	Marks
Well	0	$Q_{well}$	
Recharge	0	$q \Delta X_s \Delta Y_s$	
River	$L_{river} \Delta X_s \Delta Y_s$	$L_{river} \Delta X_s \Delta Y_s h_{river}$	$h > R_{bed}$
	0	$L_{river} \Delta X_s \Delta Y_s (h_{river} - R_{bed})$	$h < R_{bed}$
Drain	$L_{drain} \Delta X_s \Delta Y_s$	$L_{drain} \Delta X_s \Delta Y_s D_{bed}$	$h > D_{bed}$

Where

- $Q_{well}$  well's flow rate ( $L^3/T$ );
- $q$  recharge rate;
- $L_{river}$  river leakance;
- $L_{drain}$  drain leakance ;
- $h_{river}$  river stage;
- $R_{bed}$  bottom elevation of river;

$D_{bed}$  bottom elevation of drain

There may be different kinds of source/sink applying at the same nodal point, therefore,  $a_p^Q$  and  $S_f^Q$  in Eq(11) actually are summation of the contributions from all kinds of source/sink. Implementation of this process was done in Subroutine ADDQS1 (head independent source/sink) and Subroutine ADDQS2 (head dependent source/sink) in the source code.

#### d) Coefficient Matrix Assembling

From Eq(7), Eq(9) and Eq(11), readily gives a set of linear equations as follow

$$(a_p + a_p^t + a_p^Q)h_p = a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} + S_f^Q + S_f^t \quad (12)$$

It is seen that there are 9 diagonal entries in the derived-coefficient matrix for 2-D case by using traditional control volume technique.

#### e) Matrix Solver

Solution to Eq(12) can be obtained by using SOR technique or other efficient matrix solvers. SOR was adopted in IGW 2-D model. Its iterative equation can be expressed as follow

$$h_p^{k+1} = h_p^k + \frac{\alpha}{(a_p + a_p^t + a_p^Q)} \{ a_E h_E^k + a_W h_W^{k+1} + a_N h_N^k + a_S h_S^{k+1} + a_{NE} h_{NE}^k + a_{NW} h_{NW}^k + a_{SE} h_{SE}^{k+1} + a_{SW} h_{SW}^{k+1} + S_f^Q + S_f^t - (a_p + a_p^t + a_p^Q)h_p^k \} \quad (13)$$

Where  $k$  is index of iteration number and  $\alpha$  is relaxation factor.

The final matrix assembling and iterative processes are carried out in Subroutine SORHEADT in the source code.

### B) Rotational Control Volume Technique

Negative coefficient in the derived discretized matrix based on the traditional control volume technique may lead to a fake numerical solution to the physical equation such as negative concentration in plume transporting. In our IGW flow and transport 2-D and 3-D models, a new technique, rotational control volume, has been proposed to keep the coefficients of the derived discretized matrix always being positive. The fundamental idea behind this technique is that rotating the  $\mathbf{XY}$  coordinate system by the anisotropy orientation angle  $\theta$  to adapt the preferential flow direction forms a local  $\mathbf{X}_L \mathbf{Y}_L$  coordinate system, then constructs a control volume and its numerical equation in this local system. This sounds like directly rotating a traditional control volume by  $\theta$ . It must be pointed out that the size of rotational control volume is not the same as that of traditional control volume

again after doing a  $\theta \neq 0$  rotation. With  $\theta = 0$ , rotational control volume goes back to traditional control volume. A typical rotational control volume is shown in Figure 3. Always keeping the local  $\mathbf{X}_L \mathbf{Y}_L$  coordinate system in mind will help understand the following derivation.

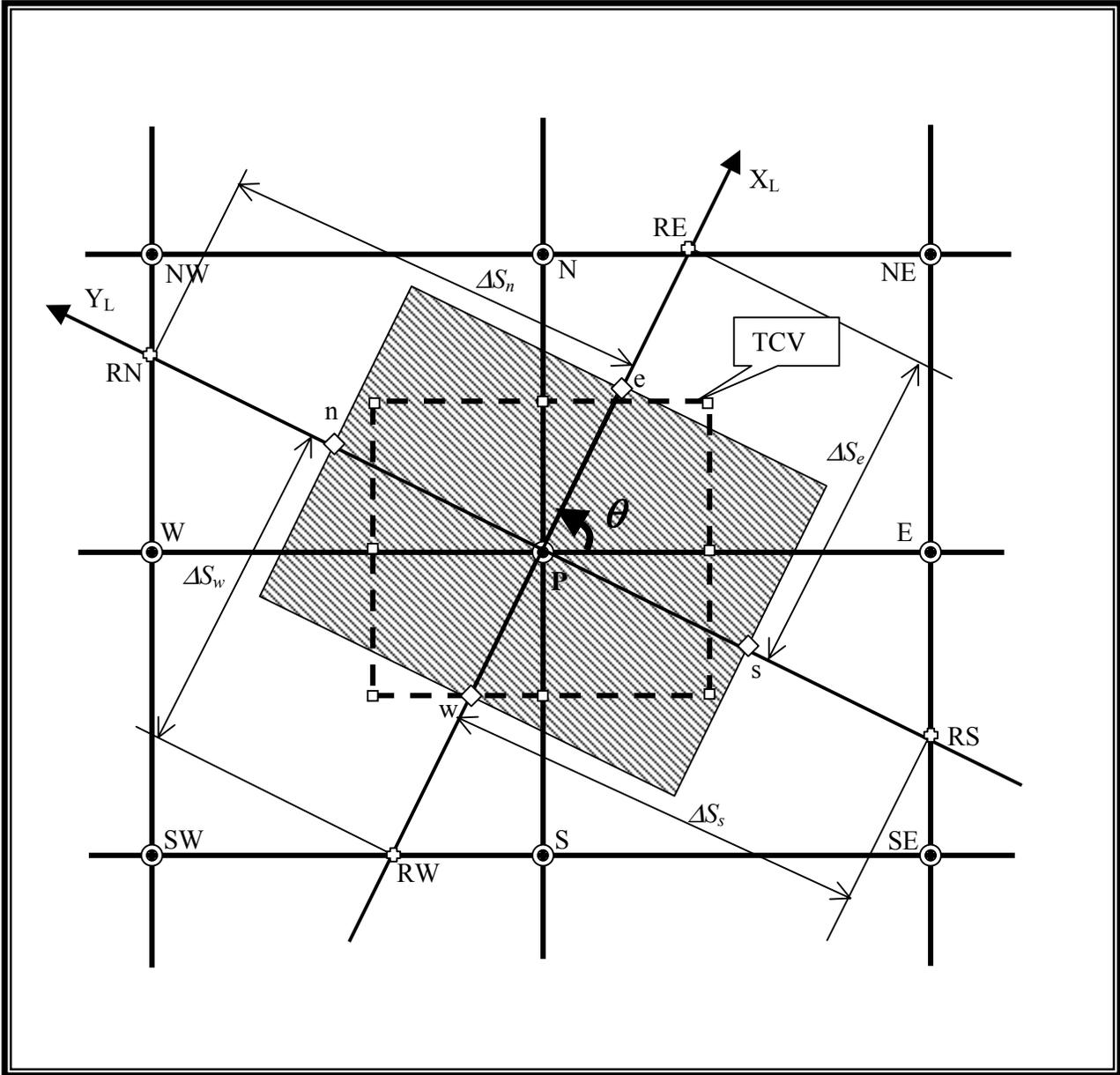


Figure 3. Rotational Control Volume

### a) Approximation to Diffusion Term

Referring to Figure 3 and applying CVT in the local coordinate system, diffusion terms of Eq(3) may be written as,

$$Diff = \frac{\Delta S_n + \Delta S_s}{2} (J_e - J_w) + \frac{\Delta S_e + \Delta S_w}{2} (J_n - J_s) \quad (14)$$

where  $J_e$ ,  $J_w$ ,  $J_n$  and  $J_s$  are the fluxes through east, west, north and south cell-faces:

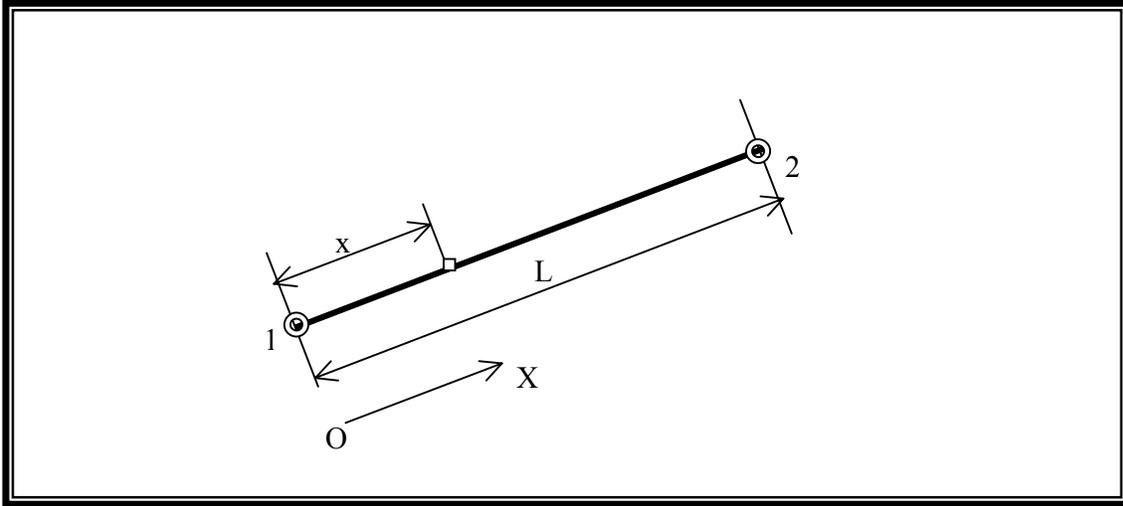
$$\begin{cases} J_e = T_{xx}^{xe} \frac{h_{RE} - h_P}{\Delta S_e} \\ J_w = T_{xx}^{xw} \frac{h_P - h_{RW}}{\Delta S_w} \\ J_n = T_{yy}^{yn} \frac{h_{RN} - h_P}{\Delta S_n} \\ J_s = T_{yy}^{ys} \frac{h_P - h_{RS}}{\Delta S_s} \end{cases} \quad (15)$$

All symbols in Eq(14) and Eq(15) were denoted in Figure 3.  $T'_{ii}$  are the principal components of transmissivity tensor which can be calculated through the known  $K'_{ii}$ , given as input parameters by VB interface.

Note that non-nodal quantities in Eq(15), such as  $h_{RE}$ ,  $h_{RW}$ ,  $h_{RN}$ , and  $h_{RS}$ , must be evaluated in terms of nodal values. A simple linear interpolation scheme has been used in IGW 2-D model which has a general form as

$$\phi(x) = \frac{L-x}{L} \phi_1 + \frac{x}{L} \phi_2 = \alpha(x) \phi_1 + \beta(x) \phi_2 \quad (16)$$

Where  $\phi_1$  and  $\phi_2$  are nodal quantities.  $\alpha(x)$  and  $\beta(x)$  in Eq. (16) are also called line element shape functions in FEM. Symbols in Eq(16) can be found in Figure 4.



**Figure 4. Shape function of line element**

In 2-D case, RE, RW, RN and RS are intersection locations of each principal direction line and the rectangular shape made by node P's eight neighboring nodes (E, SE, S, SW, W, NW, N, NE). They obviously vary with the given  $\theta$  so that those non-nodal quantities  $h_{RE}$ ,  $h_{RW}$ ,  $h_{RN}$ ,  $h_{RS}$ , have no fixed nodes to be related. For example,  $h_{RE}$  may locate between node N and node NE when  $\theta \leq \pi/2$ , then  $h_{RE}$  can be expressed as a function of  $h_E$  and  $h_{NE}$ ; however,  $h_{RE}$  may locate between node NW and node N when  $\pi/2 \leq \theta \leq 3\pi/4$ , then  $h_{RE}$  can be expressed as a function of  $h_E$  and  $h_{NW}$ . In general, each of  $h_{RE}$ ,  $h_{RW}$ ,  $h_{RN}$ ,  $h_{RS}$ , has eight possibilities to be related to two of the eight nodal heads by mean of linear interpolation. They can be expressed as follows

$$\left\{ \begin{array}{l} h_{RE} = \alpha_E^E h_E + \alpha_W^E h_W + \alpha_N^E h_N + \alpha_S^E h_S \\ \quad + \alpha_{NE}^E h_{NE} + \alpha_{SE}^E h_{SE} + \alpha_{SW}^E h_{SW} + \alpha_{NW}^E h_{NW} \\ h_{RW} = \alpha_E^W h_E + \alpha_W^W h_W + \alpha_N^W h_N + \alpha_S^W h_S \\ \quad + \alpha_{NE}^W h_{NE} + \alpha_{SE}^W h_{SE} + \alpha_{SW}^W h_{SW} + \alpha_{NW}^W h_{NW} \\ h_{RN} = \alpha_E^N h_E + \alpha_W^N h_W + \alpha_N^N h_N + \alpha_S^N h_S \\ \quad + \alpha_{NE}^E h_{NE} + \alpha_{SE}^N h_{SE} + \alpha_{SW}^N h_{SW} + \alpha_{NW}^N h_{NW} \\ h_{RS} = \alpha_E^S h_E + \alpha_W^S h_W + \alpha_N^S h_N + \alpha_S^S h_S \\ \quad + \alpha_{NE}^S h_{NE} + \alpha_{SE}^S h_{SE} + \alpha_{SW}^S h_{SW} + \alpha_{NW}^S h_{NW} \end{array} \right. \quad (17)$$

Where  $\alpha_i^j$  ( $j=E, W, N, S$ ;  $I=E, W, N, S, NE, SE, SW, NW$ ) are shape functions, which feature that:

- (1) there are six of them to be zero for a given index j
- (2)  $\alpha_i^j$  also has the following properties:

$$\begin{cases} \sum_i \alpha_i = 1 \\ \alpha_i^j > 0 \end{cases} \quad (18)$$

Table 4 shows a list of the shape functions and their corresponding variables using in the source code.

Table 4

Notation in Eq(17)	Variable in code
$\alpha_E$	WPLT(4)
$\alpha_W$	WPLT(8)
$\alpha_N$	WPLT(6)
$\alpha_S$	WPLT(2)
$\alpha_{NE}$	WPLT(5)
$\alpha_{SE}$	WPLT(3)
$\alpha_{SW}$	WPLT(1)
$\alpha_{NW}$	WPLT(7)

Substituting Eq(17) and Eq(15) into Eq(14), the following equation is obtained

$$Diff = a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} - a_P h_P \quad (19)$$

Where

$$\left\{ \begin{aligned}
a_E &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T_{xx}^{ie} \sum_j \alpha_E^j, & j = E, W, N, S \\
a_W &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T_{xx}^{iw} \sum_j \alpha_W^j, & j = E, W, N, S \\
a_N &= \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T_{yy}^{in} \sum_j \alpha_N^j, & j = E, W, N, S \\
a_S &= \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T_{yy}^{is} \sum_j \alpha_S^j, & j = E, W, N, S \\
a_{NE} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T_{xx}^{ie} \alpha_{NE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T_{xx}^{iw} \alpha_{NE}^W \\
&\quad + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T_{yy}^{in} \alpha_{NE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T_{yy}^{is} \alpha_{NE}^S \\
a_{NW} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T_{xx}^{ie} \alpha_{NW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T_{xx}^{iw} \alpha_{NW}^W \\
&\quad + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T_{yy}^{in} \alpha_{NW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T_{yy}^{is} \alpha_{NW}^S \\
a_{SE} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T_{xx}^{ie} \alpha_{SE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T_{xx}^{iw} \alpha_{SE}^W \\
&\quad + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T_{yy}^{in} \alpha_{SE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T_{yy}^{is} \alpha_{SE}^S \\
a_{SW} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} T_{xx}^{ie} \alpha_{SW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} T_{xx}^{iw} \alpha_{SW}^W \\
&\quad + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} T_{yy}^{in} \alpha_{SW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} T_{yy}^{is} \alpha_{SW}^S \\
a_P &= a_E + a_W + a_N + a_S + a_{NE} + a_{NW} + a_{SE} + a_{SW}
\end{aligned} \right. \quad (20)$$

Implementation of this process was done in Subroutine NEWCOEFFLOW. The same derived type variable CST2 as mentioned in TCV was used to store these coefficients.

From Eq(20), it is very obvious that every quantity in this equation is positive. Therefore, no negative coefficients happen again in the derived-matrix.

### b) Approximation to Time Derivative Term

For transient flow, a backward finite difference scheme is used to approximate the time-derivative term in Eq(3). It can be read as

$$S \frac{\partial h}{\partial t} = S \frac{h^{n+1} - h^n}{\Delta t} \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} = a_P^t h^{n+1} - S_f^t \quad (21)$$

Where

$$\begin{cases} a_p^t = \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{S}{\Delta t} \\ S_f^t = \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{S}{\Delta t} h^n \end{cases} \quad (22)$$

### c) Approximation to Source/Sink Term

Source/sink term discretizing is almost the same as that in traditional control volume technique except the control volume's area. In general,  $Q$  can be expressed as

$$Q = a_p^Q h_p + S_f^Q \quad (23)$$

Table 5 is a list of source/sink available in IGW 2-D model and their corresponding  $a_p^Q$  and  $S_f^Q$  when applying rotational control volume technique.

Table 5

Type of Source/Sink	$a_p^Q$	$S_f^Q$	
Well	0	$Q_{well}$	
Recharge	0	$q \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$	
River	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river}$	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river} h_{river}$	$h > R_{bed}$
	0	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river} (h_{river} - R_{bed})$	$h < R_{bed}$
Drain	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{drain}$	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{drain} D_{bed}$	$h > D_{bed}$

All symbols have been explained above. Implementation of this process was done in Subroutine ADDQS1 and Subroutine ADDQS2 .

### d) Coefficient Matrix Assembling

From Eq(19), Eq(21) and Eq(23), gives a set of linear equations as follow

$$\begin{aligned} (a_p + a_p^t + a_p^Q) h_p = & a_E h_E + a_W h_W + a_N h_N + a_S h_S \\ & + a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} + S_f^Q + S_f^t \end{aligned} \quad (24)$$

It is seen that there are still 9 diagonal entries in the derived-coefficient matrix by using rotational control volume technique, however, these entries in the matrix are totally different from those obtaining from traditional control volume technique when the anisotropy orientation angle  $\theta$  is not equal to zero.

### e) Matrix Solver

Solution to Eq(24) can be obtained by using SOR technique or other efficient matrix solvers. SOR was adopted in IGW 2-D model. Its iterative equation can be expressed as follow

$$h_L^{k+1} = h_P^k + \frac{\alpha}{(a_P + a_P^t + a_P^Q)} \left\{ a_E h_E^k + a_W h_W^{k+1} + a_N h_N^k + a_S h_S^{k+1} \right. \\ \left. + a_{NE} h_{NE}^k + a_{NW} h_{NW}^k + a_{SE} h_{SE}^{k+1} + a_{SW} h_{SW}^{k+1} \right. \\ \left. + S_f^Q + S_f^t - (a_P + a_P^t + a_P^Q) h_P^k \right\} \quad (25)$$

Where  $k$  is index of iteration number and  $\alpha$  is relaxation factor.

The final matrix assembling and iterative processes are carried out in Subroutine SORHEADT.

### 2.1.4 Special Treatments

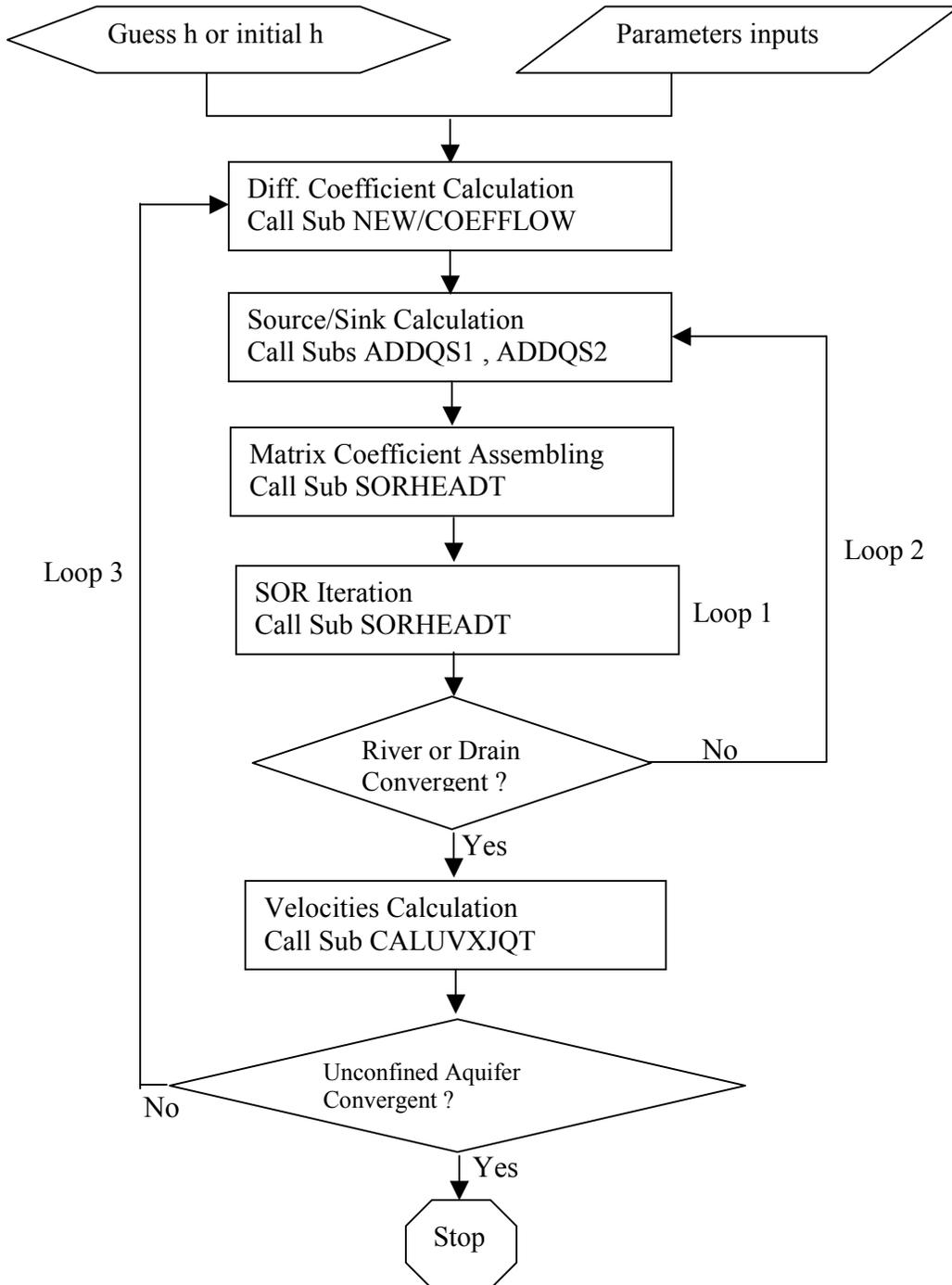
- a) To avoid doing derivation of every kind of FD equation on boundary, computational domain with dimension of 1 to NI and 1 to NJ has been expanded to that with dimension of 0 to NI+1 and 0 to NJ+1 and every parameter at this expanded node is assigned to be zero. This implies that the computational domain is confined in one no flux boundary box for all cases. No flow boundary condition also can be taken care automatically by simply letting  $K_{ii}=0$ .
- b) A boundary indicator variable IBOUND() is allocated to identify cell of first kind of boundary condition, IBOUND()=-1, cell of second kind of boundary condition or no flow boundary condition, IBOUND()=0 (inactive cell) and active cell, IBOUND()=1.
- c) When anisotropy orientation angle  $\theta$  exists,  $\theta$  will be assumed to be zero at any nodes adjacent to any inactive cell in current version. The reason why we are doing so is that there will be inconsistency for an inactive cell. For example, from Figure 2, assuming that west cell is inactive leads to no flux through west cell-face, that means that coefficient  $a_w$  should be equal to zero. However, as mentioned above, the condition  $a_w=0$  is achieved by letting  $K_{ii}=0$ . From Eq(8), contribution to  $a_w$  includes two parts – one from  $K_{ii}$ , one from  $K_{ij}$  when  $\theta \neq 0$ . Therefore,  $a_w$  will not be zero due to the contribution from  $K_{ij}$ , and the west cell is not an inactive cell any more numerically.
- d) A non-linear iteration technique is applied to the case having head dependent source/sink. This is also part of ‘inner loop’.
- e) Water table iteration for unconfined aquifer is conducted in VB for the sake of visualization of each iteration. A dry-wet trick is also used to deal with unconfined cases. This part was coded by using VB.

### 2.1.5 Matrix Solver

SOR iterative technique was employed in IGW 2-D model. In addition to SOR, Subroutine SORHEADT also give a final coefficient matrix and a right-hand-side vector which could be solved by other mean of advanced methods. Main diagonal elements are stored in  $S00(I,J)+CST2(I,J)\%SP$ , other 8 set of diagonal elements are stored in those variables listed Table 2, and RHS vector is stored in variable SUM23(I,J). Therefore, slightly modification to Subroutine SORHEADT can made the current matrix fit to your preferred matrix solvers.

### 2.1.6 Numerical Solution Procedure Flow Chart

The sequence of operations for IGW 2-D flow model is illustrated in the following flow diagram:



## 2.2 Transport

### 2.2.1 Governing Equation

The partial differential equation describing 2-D solute transport in porous medial is usually written as

$$\frac{\partial(nBC)}{\partial t} + \frac{\partial(nBu_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (nBD_{ij} \frac{\partial C}{\partial X_j}) - \rho_b \frac{\partial(BC^*)}{\partial t} - \lambda nBC + q_s C_s \quad (26)$$

Where

- $C$  solute concentration
- $B$  aquifer thickness
- $n$  porosity of the porous medium
- $u_i$  seepage or averaged pore velocity in the direction  $X_i$ ;
- $D_{ij}$  dispersion coefficient tensor
- $X_i$  Cartesian coordinate
- $C^*$  concentration of species adsorbed on the solid
- $\rho_b$  bulk density of the solid
- $\lambda$  decay coefficient
- $q_s$  volume flow rate per unit volume of the source or sink
- $C_s$  solute concentration in the source or sink fluid

Considering equilibrium transport and assuming that the adsorption isotherm can be described with a linear and reversible equation, one can write

$$C^* = K_d C \quad (27)$$

Where  $K_d$  is called the distribution coefficient. Now, by incorporating Eq(27) into Eq(26), we obtain

$$\frac{\partial(nBR_d C)}{\partial t} + \frac{\partial(nBu_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (nBD_{ij} \frac{\partial C}{\partial X_j}) - \lambda nBC + q_s C_s \quad (28)$$

or

$$\frac{\partial C}{\partial t} + \frac{u_i}{R_d} \frac{\partial C}{\partial X_i} = \frac{1}{nBR_d} \frac{\partial}{\partial X_i} (nBD_{ij} \frac{\partial C}{\partial X_j}) - \left( \frac{q_s + \lambda nB}{nBR_d} + \frac{R_d - 1}{BR_d} \frac{\partial B}{\partial t} \right) C + \frac{q_s C_s}{nBR_d} \quad (29)$$

Where

$$R_d = 1 + \frac{\rho_b K_d}{n} \quad (30)$$

The parameter  $R_d$  is called the retardation factor.

Note that there is an extra term arising from the unconfined case,  $\frac{R_d - 1}{BR_d} \frac{\partial B}{\partial t} C$ , in

Eq(29) with comparing to other general forms shown up in many text books. Eq(29) has been used in IGW 2-D transport model. The hydrodynamic dispersion tensor for isotropic porous media is defined in the following component forms:

$$\begin{cases} D_{xx} = \alpha_L \frac{u_x u_x}{\sqrt{u_x^2 + u_y^2}} + D^* \\ D_{yy} = \alpha_T \frac{u_y u_y}{\sqrt{u_x^2 + u_y^2}} + D^* \\ D_{xy} = D_{yx} = (\alpha_L - \alpha_T) \frac{u_x u_y}{\sqrt{u_x^2 + u_y^2}} \end{cases} \quad (31)$$

Where

- $\alpha_L$  the longitudinal dispersivity;
- $\alpha_T$  the transverse dispersivity;
- $D^*$  the effective molecular diffusion coefficient.

It is seen from Eq(31) that  $D_{xx}$  and  $D_{yy}$  are always positive, however,  $D_{xy}$  or  $D_{yx}$  may be negative.

### 2.2.2 Grid Layout

This section is the same as FLOW model and may be skipped.

In IGW, parameters are assigned to a block or a cell. Placing a representative node in each cell forms the grid layout using in our spatial discretizing. Figure 1 shows a typical grid layout of current use.

A typical node-cell and its neighborhood is shown in Figure 2. The figure is self-explanatory, and those grid related geometrical quantities used in the scheme are illustrated in the figure too. The following is a list of the variables using in the code and their counterparts in this figure,

Table 6

	Notation in Figure1	Variable in Code
X coordinate	$X_{ij}$	Xmesh(I,J)
Y coordinate	$Y_{ij}$	Ymesh(I,J)
X-Grid Spacing	$\Delta X_i$	HX(I)
Y-Grid Spacing	$\Delta Y_j$	HY(J)
X-CV area	$\Delta X_s$	DXS(I)
Y-CV area	$\Delta Y_s$	DYS(J)

It is noted that VF code is written based on non-uniform grid spacing, although a uniform grid spacing has been used in IGW interface. The quantities at the cell-face, such as dispersion coefficient  $D_{ij}^e$ , must be evaluated in terms of nodal values before proceeding the calculation. There are many methods available to handle it, such as those of linear interpolation, harmonic averaged, etc. Harmonic mean was adopted in IGW 2-D model.

### 2.2.3 Scheme to Discretize Equation

There are four methods being introduced to IGW 2-D transport model: method of characteristics (MOC), modified method of characteristics (MMOC), full implicit finite difference method (FD) and random walk. The difference among the first three methods is only the way to approximate the advection term, the second term on the left-hand side of Eq(26), which describes the transport of miscible contaminants at the same velocity as the groundwater. Both MOC and MMOC invoke the Particle Tracking Technique to approximate the advection term. FD scheme is only applied at well cells in which the Particle Tracking is not available due to the fact that there is no unique characteristic curve at well node. Random walk method is a pure Lagrangian approach to simulate solute transport.

#### A) Full Implicit Finite Difference Scheme

##### a) Approximation of Advection Term

The target equation to be discretized by FD method is Eq(28). Applying the finite difference algorithm, the advection term can be approximated by the concentration values at the cell-faces as below

$$\begin{aligned} ADV &= \Delta X_s \Delta Y_s \frac{\partial(nBu_i C)}{\partial X_i} = \Delta X_s \Delta Y_s \frac{\partial(nBu_x C)}{\partial X} + \Delta X_s \Delta Y_s \frac{\partial(nBu_y C)}{\partial Y} \\ &= \Delta Y_s (F_e - F_w) + \Delta X_s (F_n - F_s) \end{aligned} \quad (32)$$

Where

$$\begin{cases} F_e = C_e q_e \\ F_w = C_w q_w \\ F_n = C_n q_n \\ F_s = C_s q_s \end{cases} \quad (33)$$

$q_e$ ,  $q_w$ ,  $q_n$  and  $q_s$  are fluxes through the four cell-faces.  $C_e$ ,  $C_w$ ,  $C_n$  and  $C_s$  are the concentration values at the four cell-faces.

Again, non-nodal quantities  $C_e$ ,  $C_w$ ,  $C_n$  and  $C_s$  must be evaluated in term of nodal values. How to determine the interface concentration,  $C_e$ ,  $C_w$ ,  $C_n$  and  $C_s$ , is what distinguishes one solution technique from another. The simple upwind scheme has been used in IGW 2-D transport model. For the upwind scheme, the cell-face concentration between two neighboring nodes in a particular direction ( $X_i$ ) is set equal to the concentration at the upstream node along the same direction

$$\begin{cases} C_e = \begin{cases} C_E, & \text{if } q_e > 0 \\ C_P, & \text{if } q_e < 0 \end{cases} \\ C_w = \begin{cases} C_W, & \text{if } -q_w > 0 \\ C_P, & \text{if } -q_w < 0 \end{cases} \\ C_n = \begin{cases} C_N, & \text{if } q_n > 0 \\ C_P, & \text{if } q_n < 0 \end{cases} \\ C_s = \begin{cases} C_S, & \text{if } -q_s > 0 \\ C_P, & \text{if } -q_s < 0 \end{cases} \end{cases} \quad (34)$$

Note that we have a convention for the flux direction: “+” for flux entering the cell, “-” for flux leaving the cell.

Substituting Eq(33) and (34) into Eq(32), gives

$$ADV = a_E^{ADV} C_E + a_W^{ADV} C_W + a_N^{ADV} C_N + a_S^{ADV} C_S - a_P^{ADV} C_P \quad (35)$$

Where

$$\begin{cases} a_E^{ADV} = \Delta Y_s \max[q_e, 0] \\ a_W^{ADV} = \Delta Y_s \max[-q_w, 0] \\ a_N^{ADV} = \Delta X_s \max[q_n, 0] \\ a_S^{ADV} = \Delta X_s \max[-q_s, 0] \\ a_P^{ADV} = \Delta Y_s (\max[-q_e, 0] + \max[q_w, 0]) + \Delta X_s (\max[-q_n, 0] + \max[q_s, 0]) \end{cases} \quad (36)$$

$q_e$ ,  $q_w$ ,  $q_n$  and  $q_s$  were stored in the array FLUX(Nwell, 4) which are calculated in Subroutine LHSWELL. Implementation of this process was done by Subroutine FDCOEF.

### b) Traditional Control Volume Technique to Approximate Diffusion Term

Following the similar derivation procedure in flow model and applying control volume technique, from Figure 2, diffusion term of Eq(28) may be written as,

$$Diff = \Delta X_s \Delta Y_s \frac{J_e - J_w}{\Delta X_s} + \Delta X_s \Delta Y_s \frac{J_n - J_s}{\Delta Y_s} \quad (37)$$

Where  $J_e$ ,  $J_w$ ,  $J_n$  and  $J_s$  are the solute fluxes through east, west, north and south cell-faces:

$$\left\{ \begin{array}{l} J_e = nBD_{xx}^e \frac{C_E - C_P}{\Delta X} + nBD_{xy}^e \frac{C_{ne} - C_{se}}{\Delta Y_s} \\ J_w = nBD_{xx}^w \frac{C_P - C_W}{\Delta X} + nBD_{xy}^w \frac{C_{nw} - C_{sw}}{\Delta Y_s} \\ J_n = nBD_{yy}^n \frac{C_N - C_P}{\Delta Y} + nBD_{yx}^n \frac{C_{ne} - C_{nw}}{\Delta X_s} \\ J_s = nBD_{yy}^s \frac{C_P - C_S}{\Delta Y} + nBD_{yx}^s \frac{C_{se} - C_{sw}}{\Delta X_s} \end{array} \right. \quad (38)$$

All symbols in Eq(37) and Eq(38) have been denoted in Figure 2.

Note that non-nodal quantities in Eq(38) ( $C_{ne}$ ,  $C_{se}$ ,  $C_{nw}$ , and  $C_{sw}$ ) must be evaluated in terms of nodal values. A simple four points averaged scheme has been used in IGW 2-D model. After re-arranging Eq(38), Eq(37) becomes

$$\begin{aligned} Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S \\ & + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} - a_P^{Diff} C_P \end{aligned} \quad (39)$$

Where

$$\left\{ \begin{array}{l} a_E^{Diff} = nB \frac{\Delta Y_s D_{xx}^e}{\Delta X} + nB \frac{D_{yx}^n - D_{yx}^s}{4} \\ a_W^{Diff} = nB \frac{\Delta Y_s D_{xx}^w}{\Delta X} - nB \frac{D_{yx}^n - D_{yx}^s}{4} \\ a_N^{Diff} = nB \frac{\Delta X_s D_{yy}^n}{\Delta Y} + nB \frac{D_{xy}^e - D_{xy}^w}{4} \\ a_S^{Diff} = nB \frac{\Delta X_s D_{yy}^s}{\Delta Y} - nB \frac{D_{xy}^e - D_{xy}^w}{4} \\ a_{NE} = nB \frac{D_{xy}^e + D_{yx}^n}{4} \\ a_{NW} = -nB \frac{D_{xy}^w + D_{yx}^n}{4} \\ a_{SE} = -nB \frac{D_{xy}^e + D_{yx}^s}{4} \\ a_{SW} = nB \frac{D_{xy}^w + D_{yx}^s}{4} \\ a_P^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW} \end{array} \right. \quad (40)$$

This process was implemented in Subroutine OLDcoefTRSP. A derived type variable CST1 was used to store these coefficients, which has the following match table:

Table 7

Notation in Eq(40)	Variable in VF Code
$a_E^{Diff}$	CST1(I,J)%SE
$a_W^{Diff}$	CST1(I,J)%SW
$a_N^{Diff}$	CST1(I,J)%SN
$a_S^{Diff}$	CST1(I,J)%SSE
$a_{NE}$	CST1(I,J)%SNE
$a_{NW}$	CST1(I,J)%SNW
$a_{SE}$	CST1(I,J)%SSE
$a_{SW}$	CST1(I,J)%SSW
$a_P^{Diff}$	CST1(I,J)%SP

From Eq(40), it is very obvious that  $a_{NE}$ ,  $a_{NW}$ ,  $a_{SE}$  and  $a_{SW}$  may easily turn into negative ones, for example,  $a_{NW} < 0$  and  $a_{SE} < 0$  when  $D_{xy} > 0$ ; or  $a_{NE} < 0$  and  $a_{SW} < 0$  when  $D_{xy} < 0$ .

### c) Rotational Control Volume Technique to Approximate Diffusion Term

Following the similar derivation procedure in flow model and applying control volume technique in local coordinate system  $\mathbf{X}_L \mathbf{Y}_L$ , from figure 3, diffusion terms of Eq(28) may be written as,

$$Diff = \frac{\Delta S_n + \Delta S_s}{2} (J_e - J_w) + \frac{\Delta S_e + \Delta S_w}{2} (J_n - J_s) \quad (41)$$

Where  $J_e$ ,  $J_w$ ,  $J_n$  and  $J_s$  are the solute fluxes through east, west, north and south cell-faces:

$$\begin{cases} J_e = nBD_{xx}^{e} \frac{C_{RE} - C_P}{\Delta S_e} \\ J_w = nBD_{xx}^{w} \frac{C_P - C_{RW}}{\Delta S_w} \\ J_n = nBD_{yy}^{n} \frac{C_{RN} - C_P}{\Delta S_n} \\ J_s = nBD_{yy}^{s} \frac{C_P - C_{RS}}{\Delta S_s} \end{cases} \quad (42)$$

All symbols in Eq(41) and Eq(42) were denoted in Figure 3.  $D'_{ii}$  are the principal components of the dispersion coefficient tensor which can be easily determined by the following forms

$$\begin{cases} D'_{xx} = \alpha_L \sqrt{u_x^2 + u_y^2} \\ D'_{yy} = \alpha_T \sqrt{u_x^2 + u_y^2} \end{cases} \quad (43)$$

Note that non-nodal quantities in Eq(42) must be evaluated in terms of nodal values. Adopting the same linear interpolation scheme described in flow model,  $C_{RE}$ ,  $C_{RW}$ ,  $C_{RN}$ ,  $C_{RS}$  in Eq(42) can be expressed as follows

$$\begin{cases} C_{RE} = \alpha_E^E C_E + \alpha_W^E C_W + \alpha_N^E C_N + \alpha_S^E C_S \\ \quad + \alpha_{NE}^E C_{NE} + \alpha_{SE}^E C_{SE} + \alpha_{SW}^E C_{SW} + \alpha_{NW}^E C_{NW} \\ C_{RW} = \alpha_E^W C_E + \alpha_W^W C_W + \alpha_N^W C_N + \alpha_S^W C_S \\ \quad + \alpha_{NE}^W C_{NE} + \alpha_{SE}^W C_{SE} + \alpha_{SW}^W C_{SW} + \alpha_{NW}^W C_{NW} \\ C_{RN} = \alpha_E^N C_E + \alpha_W^N C_W + \alpha_N^N C_N + \alpha_S^N C_S \\ \quad + \alpha_{NE}^E C_{NE} + \alpha_{SE}^N C_{SE} + \alpha_{SW}^N C_{SW} + \alpha_{NW}^N C_{NW} \\ C_{RS} = \alpha_E^S C_E + \alpha_W^S C_W + \alpha_N^S C_N + \alpha_S^S C_S \\ \quad + \alpha_{NE}^S C_{NE} + \alpha_{SE}^S C_{SE} + \alpha_{SW}^S C_{SW} + \alpha_{NW}^S C_{NW} \end{cases} \quad (44)$$

Substituting Eq(42) and Eq(44) into Eq(41), the following equation is obtained

$$\begin{aligned} Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S \\ & + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} - a_P^{Diff} C_p \end{aligned} \quad (45)$$

Where

$$\left\{ \begin{aligned}
a_E^{Diff} &= nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{ie} \sum_j \alpha_E^j, & j = E, W, N, S \\
a_W^{Diff} &= nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{iw} \sum_j \alpha_W^j, & j = E, W, N, S \\
a_N^{Diff} &= nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{in} \sum_j \alpha_N^j, & j = E, W, N, S \\
a_S^{Diff} &= nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{is} \sum_j \alpha_S^j, & j = E, W, N, S \\
a_{NE} &= nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{ie} \alpha_{NE}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{iw} \alpha_{NE}^W \\
&\quad + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{in} \alpha_{NE}^N + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_s} D_{yy}^{is} \alpha_{NE}^S \\
a_{NW} &= nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{ie} \alpha_{NW}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{iw} \alpha_{NW}^W \\
&\quad + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{in} \alpha_{NW}^N + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_s} D_{yy}^{is} \alpha_{NW}^S \\
a_{SE} &= nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{ie} \alpha_{SE}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{iw} \alpha_{SE}^W \\
&\quad + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{in} \alpha_{SE}^N + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_s} D_{yy}^{is} \alpha_{SE}^S \\
a_{SW} &= nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{ie} \alpha_{SW}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{iw} \alpha_{SW}^W \\
&\quad + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{in} \alpha_{SW}^N + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_s} D_{yy}^{is} \alpha_{SW}^S \\
a_P^{Diff} &= a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW}
\end{aligned} \right. \quad (46)$$

Implementation of this process was done in Subroutine NEWCOEFTRSP. The same derived type variable CST1 was used to store these coefficients.

From Eq(46), it is very obvious that every coefficient in this equation is positive. Therefore, no negative coefficients would be appear in the discretized matrix again.

#### d) Approximation of Time-Derivative term

A backward finite difference scheme is used to approximate the time-derivative term

$$\Delta X_s \Delta Y_s nBR_d \frac{\partial C}{\partial t} = nBR_d \Delta X_s \Delta Y_s \frac{C_P^{n+1} - C_P^n}{\Delta t} = a_p^t C_P^{n+1} - S_f^t \quad (47)$$

Where  $C^{n+1}$  is the concentration at new time level  $n+1$ ,  $C^n$  is the concentration at old time level  $n$  and

$$\begin{cases} a_p^t = \Delta X_s \Delta Y_s \frac{nBR_d}{\Delta t} \\ S_f^t = \Delta X_s \Delta Y_s \frac{nBR_d}{\Delta t} C_P^n \end{cases} \quad (48)$$

For rotational control volume technique,  $\Delta X_s \Delta Y_s$  will be replaced with  $\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$  in Eq(47) and Eq(48).

### e) Approximation of Source/Sink Term

Source/sink term of the governing equation,  $q_s C_s$ , represents solute mass entering the model domain through source ( $q_s > 0$ ) or leaving the model domain through sink ( $q_s < 0$ ). For sources, it is necessary to specify the concentration of source water. For sinks, the concentration of sink water is generally equal to the concentration of groundwater in the aquifer at the sink location and can not be specified. Evapotranspiration may be assumed to take only pure water away from the aquifer so that the concentration of evapotranspiration flux is zero. In general,  $q_s C_s$  can be expressed as

$$q_s C_s = a_p^Q C_P + S_f^Q \quad (49)$$

Where

$$\begin{cases} a_p^Q = \text{Max}[-q_s, 0] \\ S_f^Q = \text{Max}[q_s, 0] C_s \end{cases} \quad (50)$$

$q_s$  is sum of the contributions from all kinds of source/sink which is stored in array QT(I,J) in source code. Calculation of  $q_s$  is performed by Subroutine QTOTAL.

### f) Approximation of Decay Term

From Eq(28), decay term,  $-\lambda nBC$ , can be easily approximated as

$$\text{Decay} = a_p^D C_P \quad (51)$$

Where  $a_p^D = \lambda nB$

Note that  $\lambda$  is assumed to be evaluated at node and input through VB interface.

### g) Coefficient Matrix Assembling and Solution Technique

From Eq(35), Eq(39) or Eq(45), Eq(47), Eq(49) and Eq(51), gives a set of linear equations as follow

$$\begin{aligned}
 (a_p^{ADV} + a_p^{Diff} + a_p^t + a_p^Q + a_p^D)C_p &= (a_E^{ADV} + a_E^{Diff})C_E + (a_W^{ADV} + a_W^{Diff})C_W \\
 &+ (a_N^{ADV} + a_N^{Diff})C_N + (a_S^{ADV} + a_S^{Diff})C_S \\
 &+ a_{NE}C_{NE} + a_{NW}C_{NW} + a_{SE}C_{SE} + a_{SW}C_{SW} \\
 &+ S_f^Q + S_f^t
 \end{aligned} \tag{52}$$

SOR was adopted in IGW 2-D model to solve Eq(52). Its iterative equation can be expressed as follow

$$\begin{aligned}
 C_p^{k+1} = C_p^k + \frac{\alpha}{a_p} &(a_E C_E^k + a_W C_W^{k+1} + a_N C_N^k + a_S C_S^{k+1} \\
 &+ a_{NE} C_{NE}^k + a_{NW} C_{NW}^k + a_{SE} C_{SE}^{k+1} + a_{SW} C_{SW}^{k+1} \\
 &+ S_f^Q + S_f^t - a_p C_p^k)
 \end{aligned} \tag{53}$$

Where

$$a_p = a_p^{ADV} + a_p^{Diff} + a_p^t + a_p^Q + a_p^D$$

$$a_E = a_E^{ADV} + a_E^{Diff}$$

$$a_W = a_W^{ADV} + a_W^{Diff}$$

$$a_N = a_N^{ADV} + a_N^{Diff}$$

$$a_S = a_S^{ADV} + a_S^{Diff}$$

$k$  index of iteration number

$\alpha$  relaxation factor.

The final matrix assembling process is carried out in Subroutine SORCBAR.

## B) Mixed Eulerian-Lagrangian Methods

Eq(29) is an Eulerian expression in which the partial derivative,  $\frac{\partial C}{\partial t}$ , represents the rate of change in solute concentration at a fixed point in space. It can be expressed in the Lagrangian form as

$$\frac{DC}{Dt} = \frac{1}{nBR_d} \frac{\partial}{\partial X_i} (nBD_{ij} \frac{\partial C}{\partial X_j}) - (\frac{q_s + \lambda nB}{nBR_d} + \frac{R_d - 1}{BR_d} \frac{\partial B}{\partial t}) C + \frac{q_s C_s}{nBR_d} \quad (54)$$

where the substantial derivative,  $\frac{DC}{Dt} = \frac{\partial C}{\partial t} + \bar{u}_i \frac{\partial C}{\partial X_i}$ , indicates the rate of change in solute concentration along the path line of a contaminant particle (or a characteristic curve of the velocity field).  $\bar{u}_i = \frac{u_i}{R_d}$  represents the retarded velocity of a contaminant particle.

By introducing the finite-difference algorithm to the substantial derivative, Eq(54) can be approximated as

$$\frac{DC}{Dt} = \frac{C_P^{n+1} - C_P^{n*}}{\Delta t} = \text{RHS} \quad (55a)$$

or

$$C_P^{n+1} = C_P^{n*} + \Delta t \times \text{RHS} \quad (55b)$$

Where

- $C_P^{n+1}$  is the average solute concentration for cell **P** at the new time level ( $n+1$ );
- $C_P^{n*}$  is the average solute concentration for cell **P** at the new time level ( $n+1$ ) due to advection alone, also referred to as the intermediate time level  $n^*$ .
- RHS finite-difference approximation to the terms on the right-hand side of Eq(54).

Depending on the use of different Lagrangian techniques to approximate the advection term, the mixed Euler-Lagrangian methods may be loosely classified as the forward-tracking MOC, the backward-tracking MMOC and a combination of these two. MOC and MMOC were used in IGW 2-D transport model. Both the method of characteristics and the modified method of characteristics involve the use of a particle tracking technique.

### a) Particle Tracking Technique

With the velocity field known, a numerical tracking scheme can be used to move particles from one position to another to approximate the advection of contaminant front.

Traditionally, the first-order Euler algorithm has been used for particle tracking:

$$\begin{cases} X^{n+1} = \frac{\Delta t}{R_d} u_x(X^n, Y^n) \\ X^{n+1} = \frac{\Delta t}{R_d} u_y(X^n, Y^n) \end{cases} \quad (56)$$

Where  $X^{n+1}, Y^{n+1}$  are the particle coordinates at the new time level ( $n+1$ );  $X^n, Y^n$  are the particle coordinates at the old time level  $n$ ;  $u_x, u_y$  are the velocities evaluated at  $(X^n, Y^n)$  which may not be always situated at a node. A uniform time step size,  $\Delta t$ , is used for all particles during the particle tracking. For particles located in areas of relatively uniform velocity, the first order Euler algorithm may have sufficient accuracy. However, for particles located in areas of strongly converging or diverging flows, the first order algorithm may not be sufficiently accurate, unless time step size is very small. In these case a higher order algorithm such as the fourth-order Runge-Kutta method may be used. The basic idea of the fourth-order Runge Kutta method is to evaluate the velocity four times for each tracking step: once at the initial point, twice at two trial midpoints, and once at trial end point. A weighted velocity based on values evaluated at these four points is used to move the particle to the new position. This process may be expressed as follows:

$$\begin{cases} X^{n+1} = X^n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\ Y^{n+1} = Y^n + \frac{l_1 + 2l_2 + 2l_3 + l_4}{6} \end{cases} \quad (57)$$

Where

$$k_1 = \Delta t u_x(X^n, Y^n, t^n)$$

$$l_1 = \Delta t u_y(X^n, Y^n, t^n)$$

$$k_2 = \Delta t u_x\left(X^n + \frac{k_1}{2}, Y^n + \frac{l_1}{2}, t^n + \frac{\Delta t}{2}\right)$$

$$l_2 = \Delta t u_y\left(X^n + \frac{k_1}{2}, Y^n + \frac{l_1}{2}, t^n + \frac{\Delta t}{2}\right)$$

$$k_3 = \Delta t u_x\left(X^n + \frac{k_2}{2}, Y^n + \frac{l_2}{2}, t^n + \frac{\Delta t}{2}\right)$$

$$l_3 = \Delta t u_y\left(X^n + \frac{k_2}{2}, Y^n + \frac{l_2}{2}, t^n + \frac{\Delta t}{2}\right)$$

$$k_4 = \Delta t u_x(X^n + k_3, Y^n + l_3, t^n + \Delta t)$$

$$l_4 = \Delta t u_y(X^n + k_3, Y^n + l_3, t^n + \Delta t)$$

In IGW 2-D model, first-order Euler method was used in MOC and fourth-order Runge-Kutta method was used in MMOC.

From above analysis, one can obviously see that the evaluation of velocity at an arbitrary point or any non-nodal quantity is required in either MOC or MMOC. The velocity interpolation scheme used in IGW 2-D transport model is simple bilinear interpolation having the general form as below (see Figure 5)

$$u(\xi, \eta) = u_1\phi_1(\xi, \eta) + u_2\phi_2(\xi, \eta) + u_3\phi_3(\xi, \eta) + u_4\phi_4(\xi, \eta) \quad (58)$$

Where  $u_i$  ( $i=1,2,3,4$ ) are nodal quantities;  $\phi_i$  ( $i=1,2,3,4$ ) are shape functions which can be expressed as

$$\begin{cases} \phi_1(\xi, \eta) = \frac{(1-\xi)(1-\eta)}{4} \\ \phi_2(\xi, \eta) = \frac{(1+\xi)(1-\eta)}{4} \\ \phi_3(\xi, \eta) = \frac{(1+\xi)(1+\eta)}{4} \\ \phi_4(\xi, \eta) = \frac{(1-\xi)(1+\eta)}{4} \end{cases} \quad (59)$$

$$\xi = \frac{X - X_C}{a}$$

$$\eta = \frac{Y - Y_C}{b}$$

Other symbols in Eq(59) can be found in Figure 5. The bilinear interpolation was performed in Subroutine PTLINEAR.

First-order Euler method was implemented in Subroutine TRACKFB\_RW . Fourth-order Runge-Kutta was implemented in Subroutine FORWARDTK0.

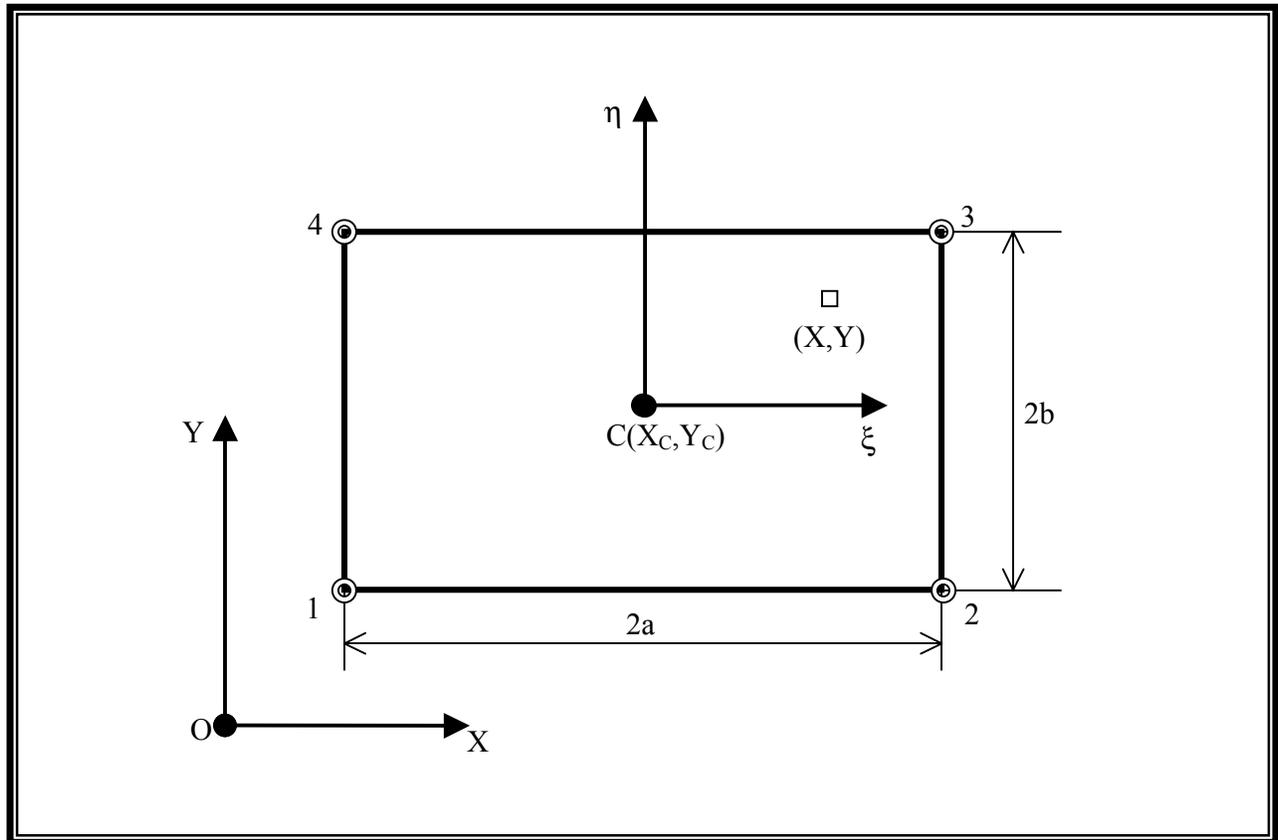


Figure 5. Bilinear Interpolation Scheme

### b) Determination of $C_p^{n*}$

The MOC uses a conventional particle-tracking technique for solving the advection term. At the beginning of the simulation, a set of moving particles is distributed in the flow field either randomly or with fixed pattern. A concentration and a position in the Cartesian coordinate system are associated with each of these particles. Particles are tracked forward through the flow field. At the end of each time increment, the concentration at cell  $\mathbf{P}$  due to advection alone over time increment,  $C_p^{n*}$ , is evaluated from the concentrations of moving particles which are located within that cell (see Figure 6). If a simple arithmetical averaged algorithm is used, this concentration can be expressed by the following equation:

$$C_p^{n*} = \frac{\sum_{i=1}^{NP_m} C_i^n}{NP_m}, \quad NP_m > 0 \quad (60)$$

where

- $NP_m$  number of particles within cell  $\mathbf{P}$  (particles marked by empty circles in Figure 6)
- $C_i^n$  concentration of the  $i^{th}$  particle at the old time level  $n$ , which is assumed to be constant in IGW.

Eq(60) is also employed in Random Walk method.

The MMOC was originally developed to approximate the advection term accurately without sacrificing a great deal of computational efficiency. Unlike the MOC, which tracks a large number of moving particles forward in time and keeps track of the concentration and position of each particle, the MMOC places one fictitious particle at each nodal point of the fixed grid at each new time level  $n+1$ . The particle is tracked backward to find its position at the old time level  $n$ . The concentration associated with that position is used to approximate the  $C_p^{n*}$  term, that is

$$C_p^{n*} = C^n(X_p, Y_p) \quad (61)$$

where  $(X_p, Y_p)$  is coordinate of the position which a particle starting from nodal point  $\mathbf{P}$  reaches when it is tracked backward along the reverse path line over the time increment  $\Delta t$  (see Figure 7). The concentration at position  $\mathbf{p}$  at the old time level ( $n$ ),  $C^n(X_p, Y_p)$ , generally is obtained by interpolating from concentrations at its neighboring nodal points. In IGW 2-D model, the bilinear interpolation scheme as described in Eq(58) was used.

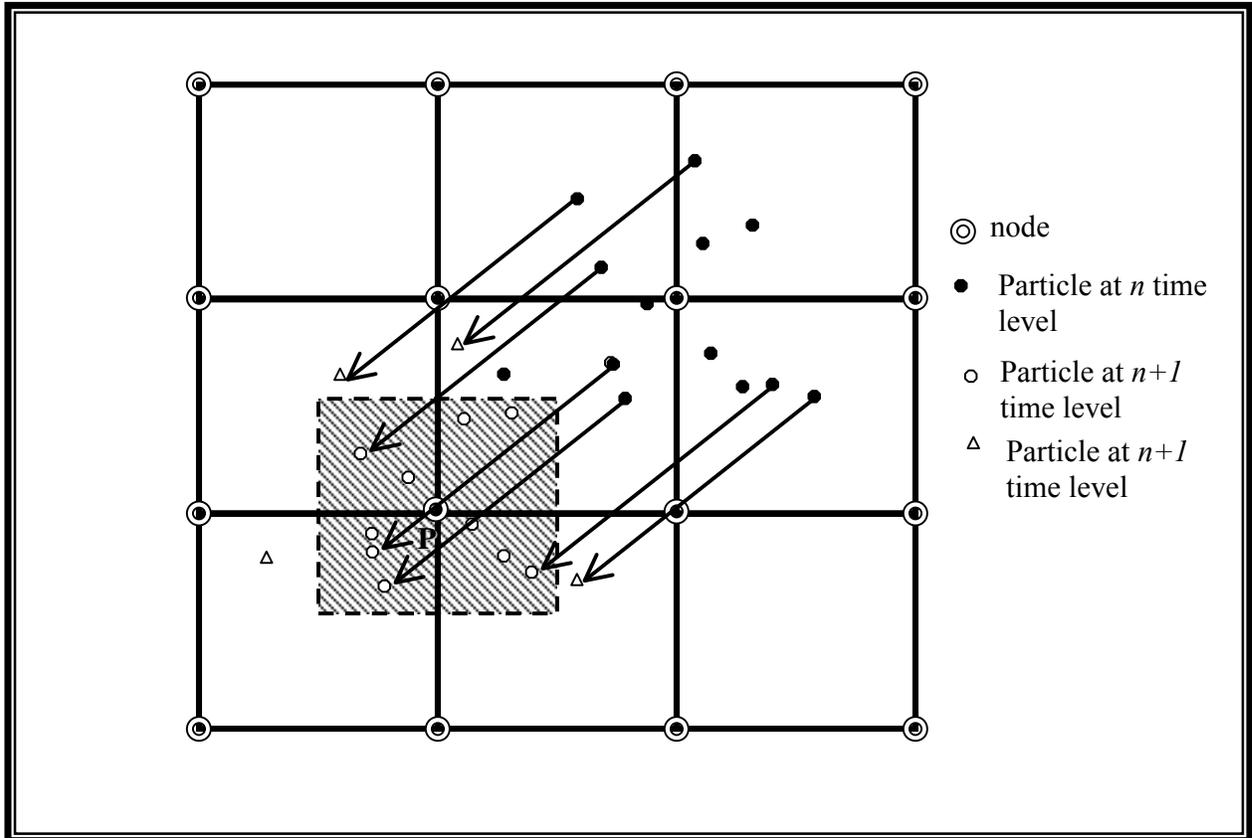


Figure 6. Illustration of the MOC

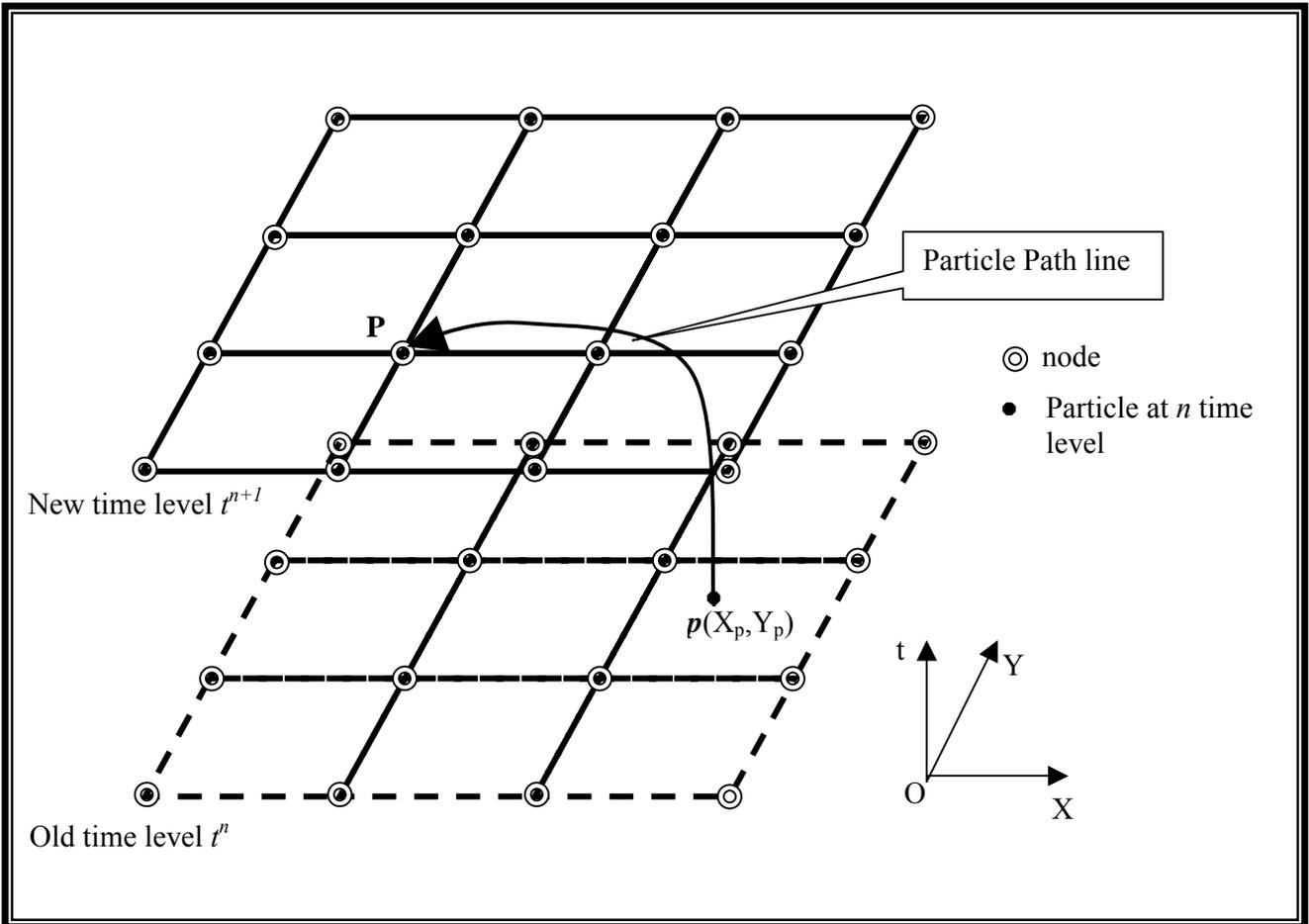


Figure 7. Illustration of the MMOC

### c) Approximation of Diffusion Term

Unlike the FD scheme, the target equation of Mixed Eulerian-Lagrangian Methods is Eq(29) in which the diffusion term has the similar form as that of Eq(28).

Referring to the course of derivation in FD scheme, diffusion term in Eq(29) may be approximated as

For traditional control volume technique:

$$\begin{aligned} Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S \\ & + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} - a_P^{Diff} C_P \end{aligned} \quad (62)$$

Where

$$\left\{ \begin{aligned} a_E^{Diff} &= \left( nB \frac{\Delta Y_s D_{xx}^e}{\Delta X} + nB \frac{D_{yx}^n - D_{yx}^s}{4} \right) \frac{1}{nBR_d} \\ a_W^{Diff} &= \left( nB \frac{\Delta Y_s D_{xx}^w}{\Delta X} - nB \frac{D_{yx}^n - D_{yx}^s}{4} \right) \frac{1}{nBR_d} \\ a_N^{Diff} &= \left( nB \frac{\Delta X_s D_{yy}^n}{\Delta Y} + nB \frac{D_{xy}^e - D_{xy}^w}{4} \right) \frac{1}{nBR_d} \\ a_S^{Diff} &= \left( nB \frac{\Delta X_s D_{yy}^s}{\Delta Y} - nB \frac{D_{xy}^e - D_{xy}^w}{4} \right) \frac{1}{nBR_d} \\ a_{NE} &= nB \frac{D_{xy}^e + D_{yx}^n}{4} \frac{1}{nBR_d} \\ a_{NW} &= -nB \frac{D_{xy}^w + D_{yx}^n}{4} \frac{1}{nBR_d} \\ a_{SE} &= -nB \frac{D_{xy}^e + D_{yx}^s}{4} \frac{1}{nBR_d} \\ a_{SW} &= nB \frac{D_{xy}^w + D_{yx}^s}{4} \frac{1}{nBR_d} \\ a_P^{Diff} &= a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW} \end{aligned} \right. \quad (63)$$

For rotational control volume technique:

$$\begin{aligned} Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S \\ & + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} - a_P^{Diff} C_P \end{aligned} \quad (64)$$

Where

$$\left\{ \begin{aligned} a_E^{Diff} &= \frac{1}{nBR_d} nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{1e} \sum_j \alpha_E^j, & j = E, W, N, S \\ a_W^{Diff} &= \frac{1}{nBR_d} nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{1w} \sum_j \alpha_W^j, & j = E, W, N, S \\ a_N^{Diff} &= \frac{1}{nBR_d} nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{1n} \sum_j \alpha_N^j, & j = E, W, N, S \\ a_S^{Diff} &= \frac{1}{nBR_d} nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{1s} \sum_j \alpha_S^j, & j = E, W, N, S \\ a_{NE} &= \frac{1}{nBR_d} \left( nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{1e} \alpha_{NE}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{1w} \alpha_{NE}^W \right. \\ &\quad \left. + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{1n} \alpha_{NE}^N + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{1s} \alpha_{NE}^S \right) \\ a_{NW} &= \frac{1}{nBR_d} \left( nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{1e} \alpha_{NW}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{1w} \alpha_{NW}^W \right. \\ &\quad \left. + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{1n} \alpha_{NW}^N + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{1s} \alpha_{NW}^S \right) \\ a_{SE} &= \frac{1}{nBR_d} \left( nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{1e} \alpha_{SE}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{1w} \alpha_{SE}^W \right. \\ &\quad \left. + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{1n} \alpha_{SE}^N + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{1s} \alpha_{SE}^S \right) \\ a_{SW} &= \frac{1}{nBR_d} \left( nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} D_{xx}^{1e} \alpha_{SW}^E + nB \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} D_{xx}^{1w} \alpha_{SW}^W \right. \\ &\quad \left. + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} D_{yy}^{1n} \alpha_{SW}^N + nB \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} D_{yy}^{1s} \alpha_{SW}^S \right) \\ a_P^{Diff} &= a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW} \end{aligned} \right. \quad (65)$$

#### d) Approximation of Time-Derivative Term

The substantial derivative in Eq(54) can be written as

$$\Delta X_s \Delta Y_s \frac{DC}{Dt} = \Delta X_s \Delta Y_s \frac{C_P^{n+1} - C_P^{n*}}{\Delta t} = a_p^t C_P^{n+1} - S_f^t \quad (66)$$

Where

$$\begin{cases} a_p^t = \Delta X_s \Delta Y_s \frac{1}{\Delta t} \\ S_f^t = \Delta X_s \Delta Y_s \frac{1}{\Delta t} C_P^{n*} \end{cases} \quad (67)$$

For rotational control volume technique,  $\Delta X_s \Delta Y_s$  will be replaced with  $\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$  in Eq(66) and Eq(67).

#### e) Approximation of Source/Sink Term

Similar approximation forms as in FD scheme can be obtained as below

$$q_s C_s = a_p^Q C_P + S_f^Q \quad (68)$$

Where

$$\begin{cases} a_p^Q = \frac{1}{nBR_d} \text{Max}[-q_s, 0] \\ S_f^Q = \frac{1}{nBR_d} \text{Max}[q_s, 0] C_s \end{cases} \quad (69)$$

#### f) Approximation of Decay Term

From Eq(54), instead having only one decay term as in FD scheme, there are two more decay terms,  $\frac{q_s}{nBR_d} C$ ,  $\frac{R_d - 1}{BR_d} \frac{\partial B}{\partial t} C$ , need to be included in these methods in addition to the real one decay term. They can be easily approximated as

$$\text{Decay} = a_p^D C_P \quad (70)$$

$$\text{Where } a_p^D = \frac{q_s + \lambda n B}{n B R_d} + \frac{R_d - 1}{B R_d} \frac{\partial B}{\partial t}$$

### g) Coefficient Matrix Assembling and Solution Technique

After obtaining  $C_p^{n*}$  by MOC or MMOC, from Eq(62) or Eq(64), Eq(66), Eq(68) and Eq(70), readily gives a set of linear equations for solving the final concentration at new time level as follow

$$\begin{aligned} (a_p^{ADV} + a_p^{Diff} + a_p^t + a_p^Q + a_p^D)C_p &= (a_E^{ADV} + a_E^{Diff})C_E + (a_W^{ADV} + a_W^{Diff})C_W \\ &+ (a_N^{ADV} + a_N^{Diff})C_N + (a_S^{ADV} + a_S^{Diff})C_S \\ &+ a_{NE}C_{NE} + a_{NW}C_{NW} + a_{SE}C_{SE} + a_{SW}C_{SW} \\ &+ S_f^Q + S_f^t \end{aligned} \quad (71)$$

The same routine SOR technique was applied to solve Eq(71) in IGW 2-D model. Its iterative equation can be expressed as follow

$$\begin{aligned} C_p^{k+1} = C_p^k + \frac{\alpha}{a_p} &(a_E C_E^k + a_W C_W^{k+1} + a_N C_N^k + a_S C_S^{k+1} \\ &+ a_{NE} C_{NE}^k + a_{NW} C_{NW}^k + a_{SE} C_{SE}^{k+1} + a_{SW} C_{SW}^{k+1} \\ &+ S_f^Q + S_f^t - a_p C_p^k) \end{aligned} \quad (72)$$

Where

$$\begin{aligned} a_p &= a_p^{ADV} + a_p^{Diff} + a_p^t + a_p^Q + a_p^D \\ a_E &= a_E^{ADV} + a_E^{Diff} \\ a_W &= a_W^{ADV} + a_W^{Diff} \\ a_N &= a_N^{ADV} + a_N^{Diff} \\ a_S &= a_S^{ADV} + a_S^{Diff} \end{aligned}$$

$k$  index of iteration number  
 $\alpha$  relaxation factor.

The final matrix assembling process is carried out in Subroutine SORCBAR.

### C) Full Lagrangian Method— Random Walk

Lagrangian methods treat the transport of solute mass by a large number of moving particles, and avoid solving the advection-dispersion equation directly. The random walk method is a typical example of the Lagrangian methods.

Particle tracking involved in MOC or MMOC was used to approximate/simulate advection effects only. This technique can not take care of the physical diffusion. The effect of dispersion can be modeled by the random walk method. In the random walk method the effect of dispersion is incorporated by adding a random displacement to the particle location after each advection movement. Each particle is assigned a mass and is moved according to the equation (see Figure 8):

$$\begin{cases} X_P^{n+1} = X_P^n + \Delta X_{ADV} + \Delta X_{Diff} \\ Y_P^{n+1} = Y_P^n + \Delta Y_{ADV} + \Delta Y_{Diff} \end{cases} \quad (73)$$

where  $X_P^n$ ,  $X_P^{n+1}$ ,  $Y_P^n$  and  $Y_P^{n+1}$  represent the particle coordinates after successive transport steps,  $\Delta X_{ADV}$  and  $\Delta Y_{ADV}$  are the advective displacement in one transport step; and  $\Delta X_{Diff}$  and  $\Delta Y_{Diff}$  are the random displacement associated with dispersion.

#### a) Determination of Advective Displacement

The advective displacement in Eq(73) can be calculated by the particle tracking technique mentioned above. Simple first-order Euler method was used in IGW 2-D to do such works. In addition to the normal advective displacement, to solve the advection-dispersion transport equation correctly using the random walk method, the velocity term used to move particles must contain two components, the seepage velocity plus a velocity correction terms, the spatial derivative of the dispersion coefficients, which means that an extra advection-like displacement arising from velocity correction terms must be added.

In 2-D case, the required correction in velocity terms are as

$$\begin{cases} \bar{u}_x = \frac{1}{R_d}(u_x + \Delta u_x) = \frac{1}{R_d}\left(u_x + \frac{\partial D_{xx}}{\partial x} + \frac{\partial D_{xy}}{\partial y}\right) \\ \bar{u}_y = \frac{1}{R_d}(u_y + \Delta u_y) = \frac{1}{R_d}\left(u_y + \frac{\partial D_{yx}}{\partial x} + \frac{\partial D_{yy}}{\partial y}\right) \end{cases} \quad (74)$$

Hence, the modified advective displacement becomes

$$\begin{cases} \Delta X_{ADV} = \frac{1}{R_d}\left(u_x + \frac{\partial D_{xx}}{\partial x} + \frac{\partial D_{xy}}{\partial y}\right)\Delta t \\ \Delta Y_{ADV} = \frac{1}{R_d}\left(u_y + \frac{\partial D_{yx}}{\partial x} + \frac{\partial D_{yy}}{\partial y}\right)\Delta t \end{cases} \quad (75)$$

Note that Eq(75) is obtained by assuming that the first-order Euler method was used only.

Other methods would lead to different forms.

### b) Determination of Dispersive Random Displacement

For 2-D transport problem, we must consider random displacements due to both longitudinal and transverse dispersion. We can express these random dispersive displacements in the longitudinal and transverse directions as follows

$$\begin{cases} Z_L = N_L(0, \sigma_L^2) + N_D(0, \sigma_D^2) = \sigma_L N_L(0,1) + \sigma_D N_D(0,1) \\ Z_T = N_T(0, \sigma_T^2) + N_D(0, \sigma_D^2) = \sigma_T N_T(0,1) + \sigma_D N_D(0,1) \end{cases} \quad (76)$$

where  $Z_L$  and  $Z_T$  are the random displacements in the longitudinal and transverse directions, respectively;  $\sigma_L = \sqrt{2\alpha_L U \Delta t}$ ;  $\sigma_T = \sqrt{2\alpha_T U \Delta t}$ ;  $U = \sqrt{u_x^2 + u_y^2}$ ;  $\sigma_D = \sqrt{2D^* \Delta t}$ ;  $N_L(0,1)$ ,  $N_T(0,1)$  and  $N_D(0,1)$  are normally distributed random number with zero mean and unit standard deviations.

### c) Final Displacement

Referring to Figure 8, from Eq(75) and Eq(76), Eq(73) becomes

$$\begin{cases} X_P^{n+1} = X_P^n + \Delta X_{ADV} + Z_L \frac{\Delta X_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} + Z_T \frac{\Delta Y_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \\ Y_P^{n+1} = Y_P^n + \Delta Y_{ADV} + Z_L \frac{\Delta Y_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} - Z_T \frac{\Delta X_{ADV}}{\sqrt{\Delta X_{ADV}^2 + \Delta Y_{ADV}^2}} \end{cases} \quad (77)$$

Eq(77) is the equation using to track particles in our IGW 2-D random walk method.

### d) Evaluation of Concentration

In IGW 2-D model, there are two options to visualize plume transport: moving particles and concentration color map or contour lines. The options need a conversion between particles distribution and concentration distribution. The conversion in two ways was coded by VB.

For the converting particles distribution to concentration distribution, the solute concentrations are evaluated at nodal points and can be expressed as the same form as Eq(60). The random walk method was coded in Subroutine TRACKFB\_RW.

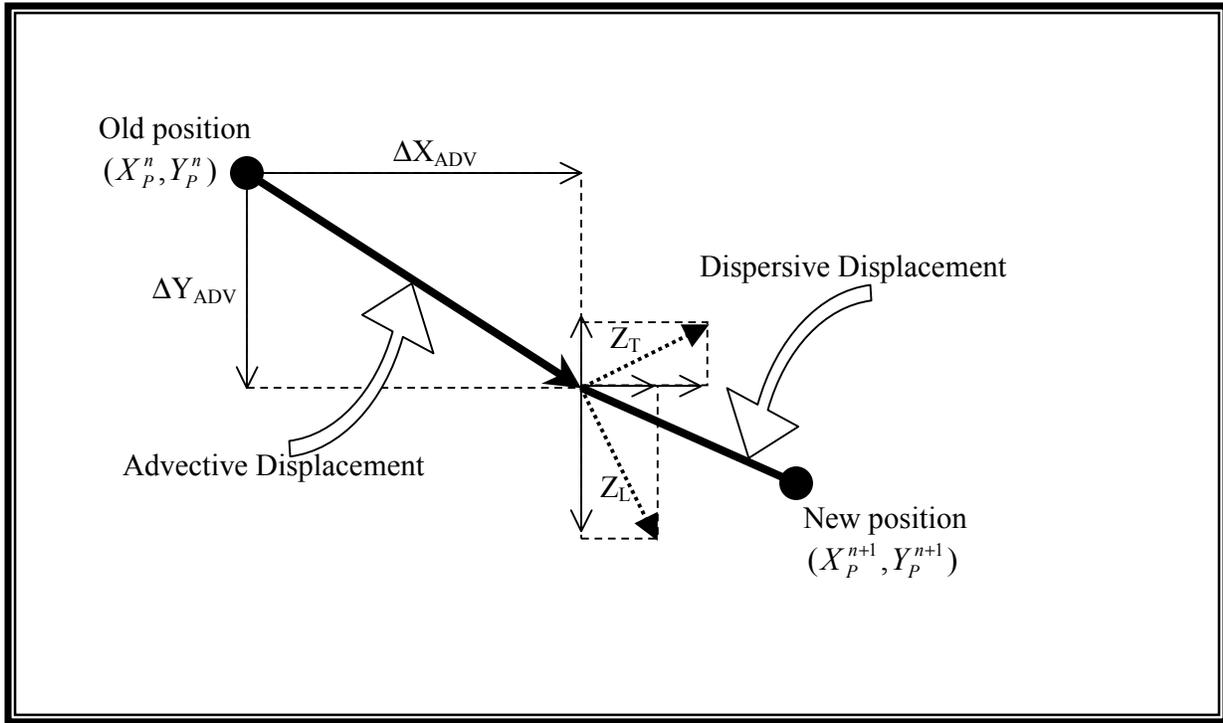


Figure 8 Illustration of particle movement by Random Walk

### 2.2.4 Special Treatments

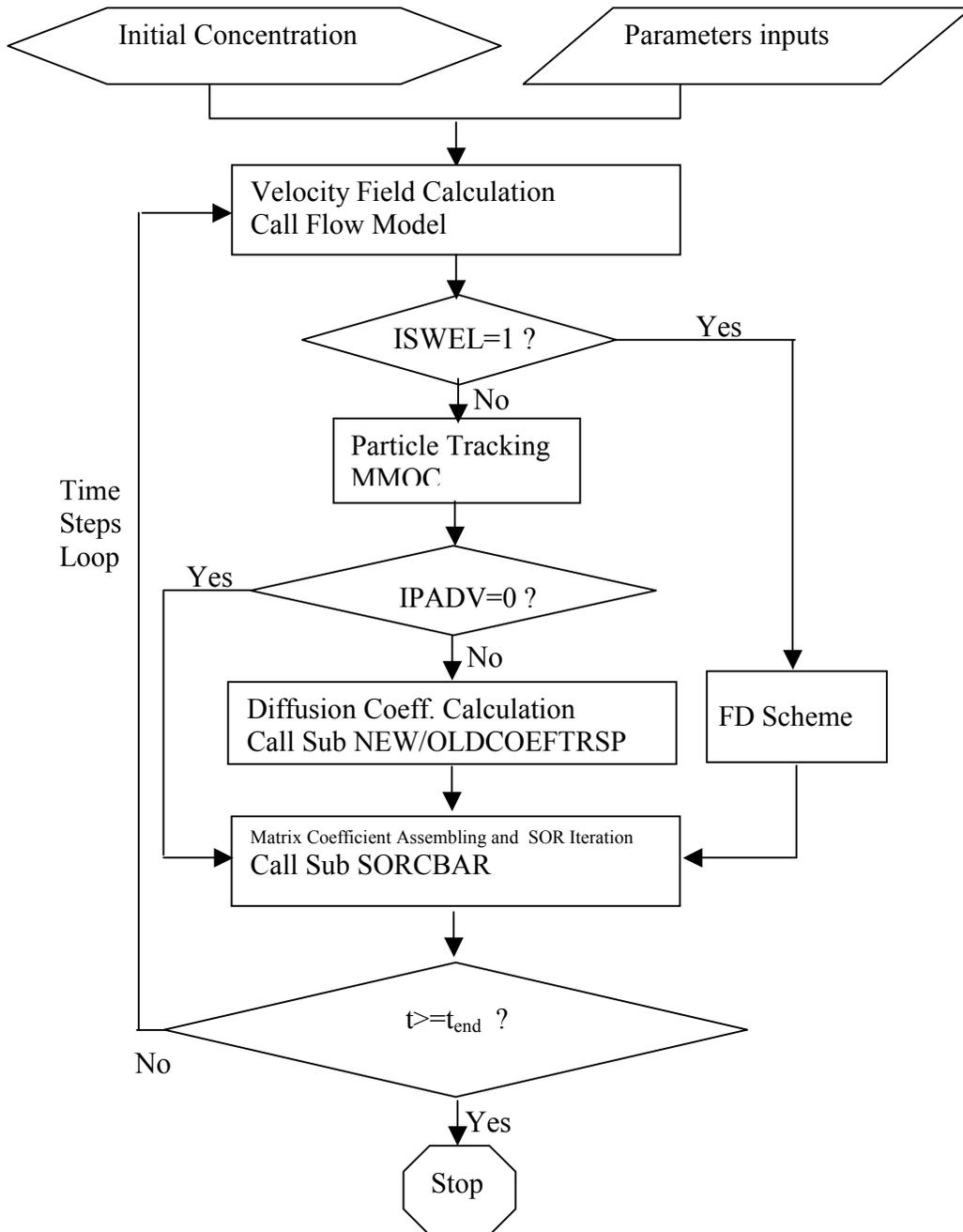
- a) As doing in IGW 2-D flow model, computational domain in IGW 2-D transport model, with dimension of 1 to NI and 1 to NJ, is also expanded to that with dimension of 0 to NI+1 and 0 to NJ+1 and every parameter at these expanded nodes is assigned to be zero.
- b) A concentration boundary condition indicator variable ICB() is allocated to identify cell of the continuous plume source, ICB=-1, or cell of instantaneous plume source.
- c) The cross terms,  $D_{xy}$  and  $D_{yx}$ , are not usually equal zero in non-uniform flow field unless the longitudinal dispersivity is equal to the transverse dispersivity or both are equal to zero (pure advection). In this case, rotational control volume technique has to be used in order to obtain a non-negative concentration distribution. The difference in implementation of rotational control volume technique between the flow and transport models is how to determine the rotation angle  $\theta$ . In flow model,  $\theta$  is assumed to be equal to the anisotropy orientation angle. In transport model,  $\theta$  will be assumed to be aligned with velocity vector.
- d) A switch, IPADV was set to skip the burden calculation of diffusion coefficients when there is no dispersivities in both longitudinal and transverse directions.
- e) In IGW transport model, velocity at well node is assumed to be zero.
- f) FD scheme is always applied at well cell in IGW.
- g) No decay and source/sink terms are considered in random walk method.
- h) No solute concentration in the source/sink has been considered in current IGW 2-D version, although it has already been coded in the Visual Fortran code.

### 2.2.5 Matrix Solver

SOR iterative technique was also employed in IGW 2-D transport model. In addition to SOR technique, Subroutine SORCBAR also give a final matrix which could be solved by other mean of advanced methods. Main diagonal elements are stored in  $S00(I,J)+CST1(I,J)\%SP$ , other 8 set of diagonal elements are stored in those variables listed Table 7, and RHS vector is stored in variable SUM23(I,J). Therefore, slightly modification to Subroutine SORCBAR can made the current matrix fit to your preferred matrix solvers.

2.2.6 Numerical Solution Procedure Flow Chart

The sequence of operations for IGW 2-D transport model is illustrated in the following flow diagram:



## 2.3 Monte Carlo Simulation

Monte Carlo Simulation is used in evaluating the impacts of model input uncertainty on calculated results. Monte Carlo method is by far the most commonly used method for analysis of uncertainty associated with complex numerical models. The primary advantages of the Monte Carlo method are conceptual simplicity, general applicability, and the ability to quantify fully the uncertainty in model output. The Monte Carlo Simulation in IGW includes three parts – generating random fields of model input parameters, solving head and solute transport equations based on the generated parameters, recursively calculating output parameters' statistical distributions such as PDF, CDF.

### 2.3.1 Random Field Generator

The heart of the Monte Carlo method is the generation of multiple realizations (or samples) of input parameters that are considered to be random variable. Each random variable is assumed to follow a certain probabilistic model characterized by its spectral density function or covariance function. IGW 2-D model uses the Fast Fourier Transform (FFT) technique as its random field generator.

Assuming that  $h(t_1, t_2)$  is a zero mean 2-D stochastic process with a stationary covariance function  $R(\tau_1, \tau_2) = E[h(t_1 + \tau_1, t_2 + \tau_2)h^*(t_1, t_2)]$ , where  $h^*(t_1, t_2)$  is the conjugate function of  $h(t_1, t_2)$ . The spectrum of the stochastic process,  $S(f_1, f_2)$ , can be expressed as the FFT of the covariance function:

$$\begin{cases} S(f_1, f_2) = \iint R(\tau_1, \tau_2) e^{2\pi i(f_1 \tau_1 + f_2 \tau_2)} d\tau_1 d\tau_2 \\ R(\tau_1, \tau_2) = \iint S(f_1, f_2) e^{-2\pi i(f_1 \tau_1 + f_2 \tau_2)} df_1 df_2 \end{cases} \quad (78)$$

Their corresponding discrete FFT are

$$S_{n_1, n_2} = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} R_{k_1, k_2} e^{2\pi i \left( \frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2} \right)} \quad (79)$$

$$R_{k_1, k_2} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} S_{n_1, n_2} e^{-2\pi i (k_1 \Delta_1 f_{n_1} + k_2 \Delta_2 f_{n_2})} \quad (80)$$

Where

- $N_1, N_2$  2-D domain dimensions
- $f$  the frequency
- $\tau$  the time or spatial quantity
- $\Delta$  the grid spacing

If  $h_k$  is a stochastic process and its discrete FFT has the form

$$h_{k_1, k_2} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} H_{n_1, n_2} e^{-2\pi i(k_1 \Delta_1 f_{n_1} + k_2 \Delta_2 f_{n_2})} \quad (81)$$

then  $H_{n_1, n_2}$  are random.

Assuming,

$$\begin{cases} E[H_{n_1, n_2}] = 0 \\ E[H_{n_1, n_2} H_{m_1, m_2}^*] = 0 \quad (n_i \neq m_i) \end{cases} \quad (82)$$

where  $H^*$  means conjugate function of  $H$ , from Eq(81) and using Eq(82), one obtains

$$E[h_{k_1, k_2}] = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} E[H_{n_1, n_2}] e^{-2\pi i(k_1 \Delta_1 f_{n_1} + k_2 \Delta_2 f_{n_2})} = 0 \quad (83)$$

$$\begin{aligned} R_{j_1, j_2} &= E[h_{k_1+j_1, k_2+j_2} h_{k_1, k_2}^*] \\ &= \frac{1}{N_1^2 N_2^2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} E[H_{n_1, n_2} H_{n_1, n_2}^*] e^{2\pi i(j_1 \Delta_1 f_{n_1} + j_2 \Delta_2 f_{n_2})} \end{aligned} \quad (84)$$

Comparing Eq(84) with Eq(80) leads to the following equation

$$E[H_{n_1, n_2} H_{n_1, n_2}^*] = N_1 N_2 S_{n_1, n_2} \quad (85)$$

It is seen that generating of a zero mean stochastic process  $h_k$  with a specific covariance structure becomes generating of another stochastic process  $H_{n_1, n_2}$  in frequency domain under the conditions:

$$\begin{cases} E[H_{n_1, n_2}] = 0 \\ E[H_{n_1, n_2} H_{m_1, m_2}^*] = 0 \quad , (n_i \neq m_i) \\ E[H_{n_1, n_2} H_{n_1, n_2}^*] = N_1 N_2 S_{n_1, n_2} \end{cases} \quad (86)$$

If

$$H_{n_1, n_2} = \sqrt{N_1 N_2 S_{n_1, n_2}} e^{i\theta_{n_1, n_2}} \quad (87)$$

and  $\theta_{n_1, n_2}$  are independent random variables, uniformly distributed over  $[0, 2\pi]$ , then the Eq(86) is met. For the process to be real,  $H_{N_1-n_1, N_2-n_2} = H_{n_1, n_2}^*$ , which implies that

$$\theta_{N_1-n_1, N_2-n_2} = \theta_{n_1, n_2} \cdot$$

The major steps to generate a realization of zero mean stochastic process can be outlined as,

1)  $H_{0,0} = 0 + 0i$

2)  $H_{\frac{N_1}{2}, n_2} = 0 + 0i$ , for all  $n_2$ ; and  $H_{n_1, \frac{N_2}{2}} = 0 + 0i$  for all  $n_1$

3)  $H_{n_1, n_2} = \sqrt{N_1 N_2 S_{n_1 n_2}} e^{i\theta_{n_1, n_2}}$  for  $1 \leq n_1 \leq N_1/2 - 1$  and  $N_1/2 + 1 \leq n_1 \leq N_1 - 1$ , and  $1 \leq n_2 \leq N_2/2 - 1$ , where  $\theta_{n_1, n_2}$  are independent random variables, uniformly distributed over  $[0, 2\pi]$

4)  $H_{n_1, n_2} = H_{N_1 - n_1, N_2 - n_2}^*$ , for  $1 \leq n_1 \leq N_1/2 - 1$  and  $N_1/2 + 1 \leq n_1 \leq N_1 - 1$ , and  $N_2 + 1 \leq n_2 \leq N_2 - 1$

5)  $H_{n_1, n_2} = \sqrt{N_1 N_2 S_{n_1 n_2}} e^{i\theta_{n_1, n_2}}$  for  $n_2=0$  and  $1 \leq n_1 \leq N_1/2 - 1$ , and for  $n_1=0$  and  $1 \leq n_2 \leq N_2/2 - 1$

6)  $H_{n_1, n_2} = H_{N_1 - n_1, n_2}^*$ , for  $n_2=0$  and  $N_1/2 + 1 \leq n_1 \leq N_1 - 1$

7)  $H_{n_1, n_2} = H_{n_1, N_2 - n_2}^*$ , for  $n_1=0$  and  $N_2/2 + 1 \leq n_2 \leq N_2 - 1$

then  $h_{k_1, k_2}$  can be generated by using the inverse FFT

The polygon-based random field generator was coded in Subroutine RANDOM\_FIELD.

### 2.3.2 Governing Equations

The flow and transport equations to be solved in IGW 2-D Monte Carlo Simulation are Eq(3) and Eq(26). All those methods mentioned above can be applied to IGW 2-D Monte Carlo Simulation except the random walk method.

### 2.3.3 Calculation of Statistical Distributions

The point-based output statistics available in IGW 2-D model are: probability distribution function or probability density function (PDF), cumulative density function CDF, the mean or first moment, the standard deviation or second moment, the skewness or third moment and the kurtosis or fourth moment. The mean's alternative estimators, such as median and the mode, and the average deviation can be also calculated in IGW 2-D model. The field-based output statistics include the means, variances and 25 pair of auto or cross co-variances. All the field-based statistics are updated recursively as increasing of number of realization (samples).

In current IGW 2-D model, the output parameters involving the statistical calculation include hydraulic conductivity, head, concentration and the flux through a polyline.

#### A) Point-Based Statistics

Let a set of values  $x_1, x_2, \dots, x_i, \dots, x_N$ , represent the values of one of the output parameters at a specific point (nodal or non-nodal point) with respect to different realizations, we have

#### PDF Histogram or Bar-graph $p(x)$ :

This process can be implemented by following the steps:

- $X_{\max} = \text{MAX}(x_1, x_2, \dots, x_i, \dots, x_N)$ ;
- $X_{\min} = \text{MIN}(x_1, x_2, \dots, x_i, \dots, x_N)$ ;
- By giving number of intervals,  $M$ , obtains  $\Delta x = \frac{X_{\max} - X_{\min}}{M}$ ;
- summing up the total number of data which value is between each interval  $[x_j, x_{j+1}]$  ( $j=1, 2, \dots, M$ ),  $n_j$ , (note that  $\sum_{j=1}^M n_j = N$ );
- calculating PDF:  $p(x_j) = \frac{n_j}{N}$  (88)

#### CDF Histogram or Bar-graph:

A CDF can be derived from a PDF by the following formula

$$c(x) = \int_{X_{\min}}^x p(x) dx$$

or

$$c(x_j) \approx \Delta x \sum_{i=1}^j p(x_i) \quad (89)$$

#### Mean:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i \quad (90)$$

**Standard Deviation:**

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{X})^2} \quad (91)$$

**Skewness:**

$$Skew = \frac{1}{N} \sum_{i=1}^N \frac{(x_i - \bar{X})^3}{\sigma} \quad (92)$$

**Kurtosis:**

$$Kurt = \left\{ \frac{1}{N} \sum_{i=1}^N \frac{(x_i - \bar{X})^4}{\sigma} \right\} - 3 \quad (93)$$

**Average Deviation:**

$$ADev = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{X}| \quad (94)$$

**Median:**

The median of a probability distribution function  $p(x)$  is the value  $X_{med}$  for which larger and smaller values of  $x$  are equally probable:

$$\int_{-\infty}^{X_{med}} p(x) dx = 0.5 = \int_{X_{med}}^{\infty} p(x) dx \quad (95)$$

This is an implicit equation for  $X_{med}$ . It can be solved iteratively by given  $p(x)$ .

**Mode or Maximum like-hood:**

The mode of probability distribution function  $p(x)$  is the value  $X_{mod}$  of  $x$  where it takes on a maximum value.

$$p(x) \Big|_{X_{mod}} = p_{MAX} \quad (96)$$

This is also an implicit equation for  $X_{mod}$ . It can be solved simply by finding the maximum value of  $p(x_j)$ , then, gives directly,  $X_{mod} = x_j$ . This simple method was used in our IGW 2-D Monte Carlo Simulation.

Note that in point-based statistics calculation all the values of realizations sampling at one point (it may be nodal or non-nodal point),  $x_i$ , are stored in an array first, then those statistics mentioned above are computed based on this series data. It means that the mean  $\bar{X}$

used to calculate other high order moments is more accuracy than that obtaining from recursive method. The statistics calculation has been packaged in Subroutine CALPDFCDF.

**B) Field-Based Statistics**

Assume that  $f(x,y)$  is a random field, by invoking the statistical theory, leads

**Recursive Mean:**

$$\bar{F}^{(k)}(x,y) = \frac{\bar{F}^{(k-1)}(x,y)(k-1) + f(x,y)}{k}, \quad k = 1, 2, \dots, N_k \quad (97)$$

where  $k$  is index of realizations. The total number of realization,  $N_k$ , may be limit or infinite in IGW 2-D Monte Carlo Simulation.

**Recursive Variance:**

$$\sigma^{(k)}(x,y) = \frac{\sigma^{(k-1)}(x,y)(k-1) + [f(x,y) - \bar{F}^{(k)}(x,y)]^2}{k}, \quad k = 1, 2, \dots, N_k \quad (98)$$

**Recursive Covariance:**

Given two random fields  $f(x,y)$  and  $g(x,y)$ , the covariance between point  $P_f(x,y)$  and  $P_g(x_0,y_0)$  can be expressed

$$\begin{aligned} \overline{f'(x,y)g'(x_0,y_0)}^{(k)} &= \frac{\overline{f'(x,y)g'(x_0,y_0)}^{(k-1)}(k-1)}{k} \\ &+ \frac{[f(x,y) - \bar{F}^{(k)}(x,y)][g(x_0,y_0) - \bar{G}^{(k)}(x_0,y_0)]}{k} \\ k &= 1, 2, \dots, N_k \end{aligned} \quad (99)$$

Note that if  $P_f$  or  $P_g$  is not nodal point, then Eq(58) has to be used to obtain  $f(x,y)$ ,  $\bar{F}(x,y)$  or  $g(x_0,y_0)$ ,  $\bar{G}(x_0,y_0)$ .

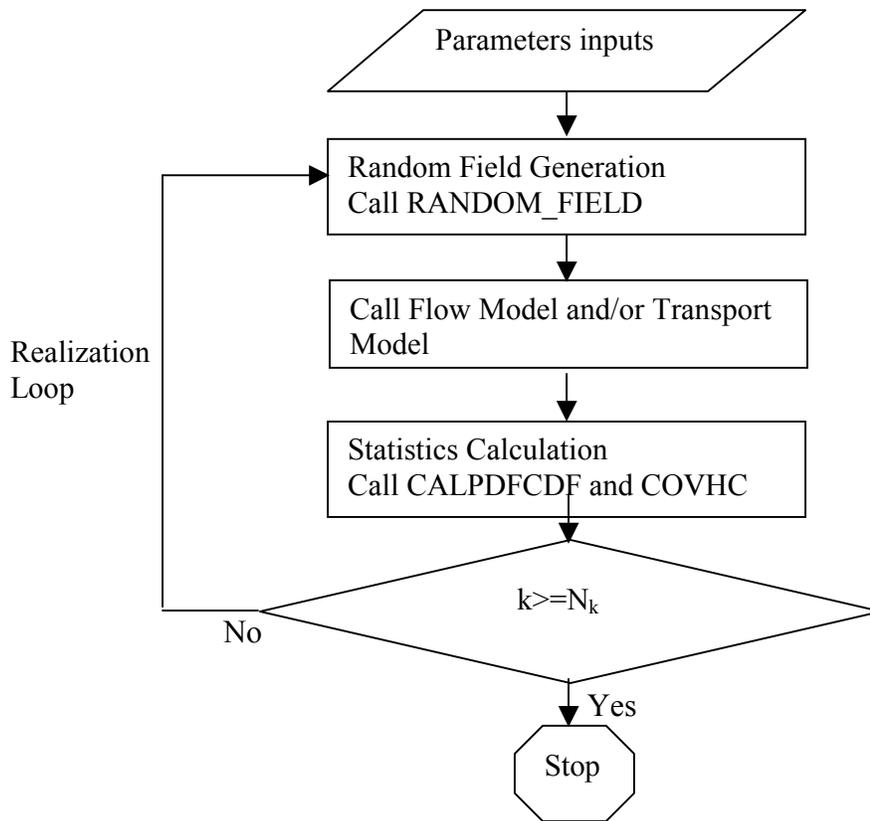
$\overline{f'(x,y)g'(x_0,y_0)}^{(k)}$  was stored in array COV() in the source code. It is also pointed out that recursive mean dose not represent the real mean that can only be obtained after knowing the whole values of all the considered realizations. To obtain a real mean, one has to store the whole values of all the considered realizations first, then calculate it. In IGW 2-D Monte Carlo Simulation, recursive mean is still adopted to deal with the field-based statistics calculation in order to save the huge memory, although somewhat not-good-enough statistics results occur by doing so (such as  $\sigma$  or those covariances). Field-based statistics calculation was carried out in Subroutine COVHC.

### 2.3.4 Special Treatments

- a) An array COV, which contains all 25 pair of co-variances, is allocated as a dynamic array. Storing data to COV or fetching data from COV has to follow a certain rule that is used in order to save memory and reduce the numbers of calling dummy arguments interacting between VB and VF.
- b) In current IGW 2-D version, only the hydraulic conductivity is considered as random in our Monte Carlo Simulation. Other model input parameters are still assumed to be deterministic ones.

**2.2.5 Numerical Solution Procedure Flow Chart**

The sequence of operations for IGW 2-D Monte Carlo Simulation is illustrated in the following flow diagram (The whole operations were organized by using Visual Basic Language) :



## 2.4 One-Layer-Based Profile Model

Unlike most current profile models, one-layer-based profile model in IGW is not a simple vertical 2-D model but has something new. Basing on the plane 2-D model, it uses a slice with certain thickness to approximate a profile model (see Figure.9). Further more, in our profile model, the summation of fluxes entering or leaving the slice has been taken into account as the approximated contributions from the third dimension. This new technique make our profile model giving much more 3-D-liked results. The assumptions made in this model are:

- a) Information of a profile model, such as its location, hydraulic features and boundary condition, comes from a plane 2-D model;
- b) Use rule of ‘point to line’ to define profile model’s boundary conditions which means that node-based information from plane 2-D model without or incapable of providing Z coordinate was assigned to all the nodes located on the vertical line going through this node. For example, from Figure 9, if the information of node B is constant head node, then all the nodes discretized from line BB’ in vertical model (see Figure 10) will be assumed to be constant head.
- c) No transient effectiveness has been considered yet in current version.

### 2.4.1 Governing Equation

Considering a coordinate system **OLTZ** shown in Figure 10, Eq(1) can be written as the following form

$$\begin{aligned}
 S_s \frac{\partial h}{\partial t} = & \frac{\partial}{\partial X_L} (K_{LL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_L} (K_{LZ} \frac{\partial h}{\partial X_Z}) + \frac{\partial}{\partial X_L} (K_{LT} \frac{\partial h}{\partial X_T}) \\
 & + \frac{\partial}{\partial X_Z} (K_{ZL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_Z} (K_{ZZ} \frac{\partial h}{\partial X_Z}) + \frac{\partial}{\partial X_Z} (K_{ZT} \frac{\partial h}{\partial X_T}) \\
 & + \frac{\partial}{\partial X_T} (K_{TL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_T} (K_{TZ} \frac{\partial h}{\partial X_Z}) + \frac{\partial}{\partial X_T} (K_{TT} \frac{\partial h}{\partial X_T}) \\
 & + q_s
 \end{aligned} \tag{100}$$

Taking integration of Eq(100) for the whole slice along the transverse direction , that is, from

$$X_T = -\frac{b}{2} \text{ to } X_T = \frac{b}{2}, \text{ gives}$$

$$\begin{aligned}
 S \frac{\partial h}{\partial t} = & \frac{\partial}{\partial X_L} (T_{LL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_L} (T_{LZ} \frac{\partial h}{\partial X_Z}) \\
 & + \frac{\partial}{\partial X_Z} (T_{ZL} \frac{\partial h}{\partial X_L}) + \frac{\partial}{\partial X_Z} (T_{ZZ} \frac{\partial h}{\partial X_Z}) \\
 & + Q_{Tr} + Q_s
 \end{aligned} \tag{101}$$

where

$$S = S_s b$$

$$T_{ij} = K_{ij} b$$

$$Q_{Tr} = \int_{-b/2}^{b/2} \left[ \frac{\partial}{\partial X_T} \left( K_{TT} \frac{\partial h}{\partial X_T} \right) \right] dX_T$$

$Q_s$  source/sink term after averaging over transverse direction within  $[-b/2, b/2]$ .

It is seen that  $Q_{Tr}$  actually represents the net transverse direction flux entering and leaving the slice which is calculated from the plane 2-D model. Note that Eq(101) implies the following assumptions

$$\begin{cases} \frac{\partial}{\partial X_L} \left( K_{LT} \frac{\partial h}{\partial X_T} \right) = 0 \\ \frac{\partial}{\partial X_T} \left( K_{TL} \frac{\partial h}{\partial X_L} \right) = 0 \\ \frac{\partial}{\partial X_Z} \left( K_{ZT} \frac{\partial h}{\partial X_T} \right) = 0 \\ \frac{\partial}{\partial X_T} \left( K_{TZ} \frac{\partial h}{\partial X_Z} \right) = 0 \end{cases} \quad (102)$$

Eq(101) is the governing equation of our profile model and can be solved by the previous methods mentioned in IGW 2-D flow model. Calculation of  $Q_{Tr}$  was implemented in Subroutine FLUXFORVMD, solving Eq(101) can be achieved by calling Subroutine TSHEAD.

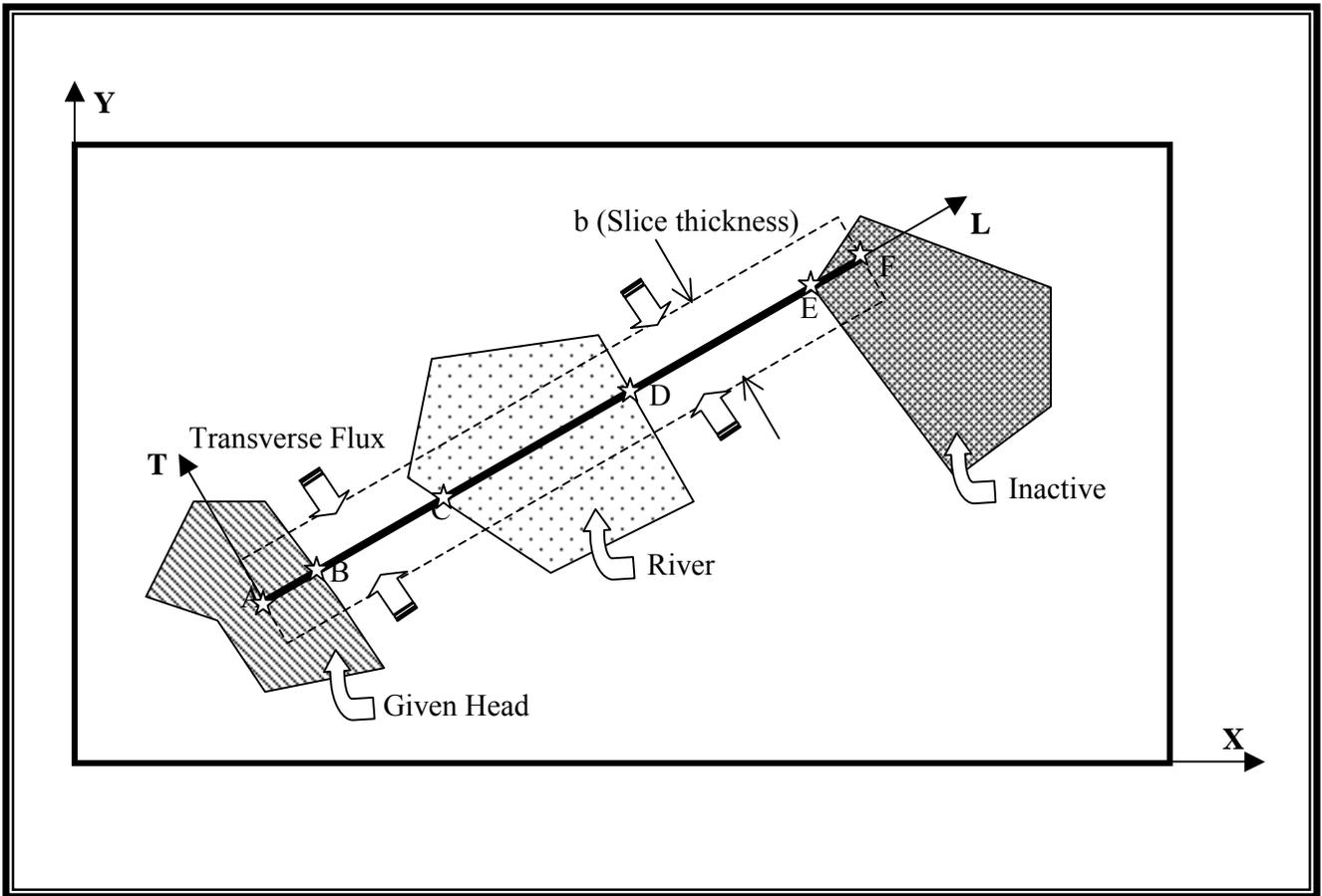


Figure 9 Illustration of a Profile Location and Features on Plane 2-D Model

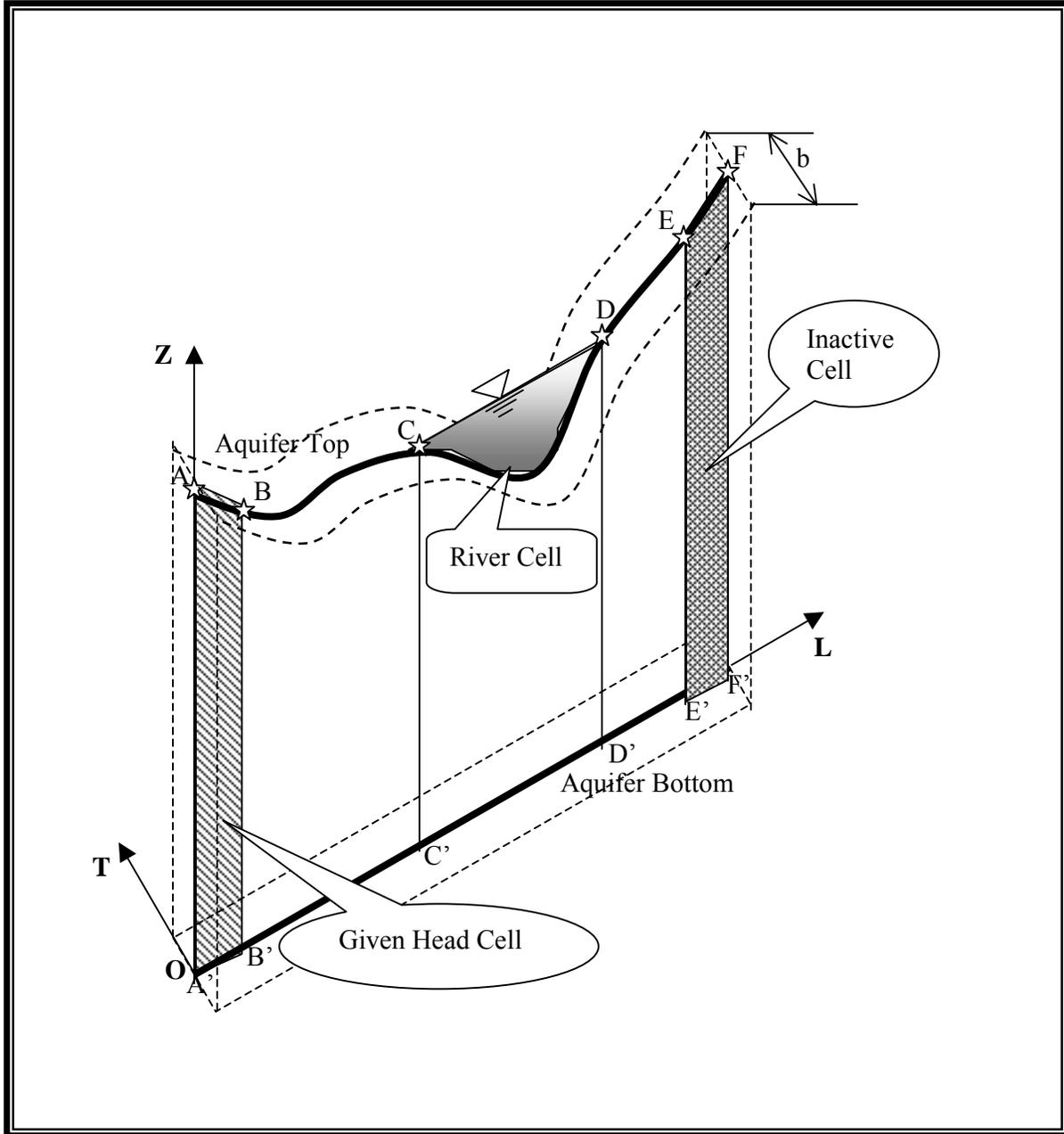


Figure 10 Illustration of a Slice in Vertical Plane

## 2.4.2 Discretizing Conceptual Features of Profile Model

When a profile location created in a plane 2-D model crosses the conceptual features of plane 2-D model, such as aquifer top or bottom, river bed, river stage, inactive cell and so on (see Figure 9 and Figure 10), these features must be appended to the profile model. These conceptual features is then transferred or mapped to numerical parameters evaluated at nodal points. In IGW profile model, the conceptual features can be classified as point-related feature such as inactive cell and poly-line-related feature such as river bed.

### A) Approximation to Point-Related Conceptual Feature

Figure 11 illustrates an example of mapping a point-related conceptual feature to the nodes. There are two steps to be followed in this mapping process:

- a) Searching for the nearest node (B or index of (i,j)) to the sampling node (A) according to the given coordinates of  $A(X_L^A, X_Z^A)$  and  $\Delta X_L, \Delta X_Z$ ;  
In IGW profile model, the following formula is used to locate the index(i,j) of the nearest node

$$\begin{cases} i = \text{int}\left(\frac{X_L - X_L^O}{\Delta X_L}\right) + 1 \\ j = \text{int}\left(\frac{X_Z - X_Z^O}{\Delta X_Z}\right) + 1 \end{cases} \quad (103)$$

where  $(X_L^O, X_Z^O)$  is original coordinates of computational domain in profile model.

- b) Following the ‘point to line’ rule, assign the parameter value,  $p_A$  at node A to that at node B and those at nodes located on line AA’, that is

$$\begin{cases} p_B = p_A \\ p_{il} = p_A, l = j, j-1, \dots, j_{A'} \end{cases} \quad (104)$$

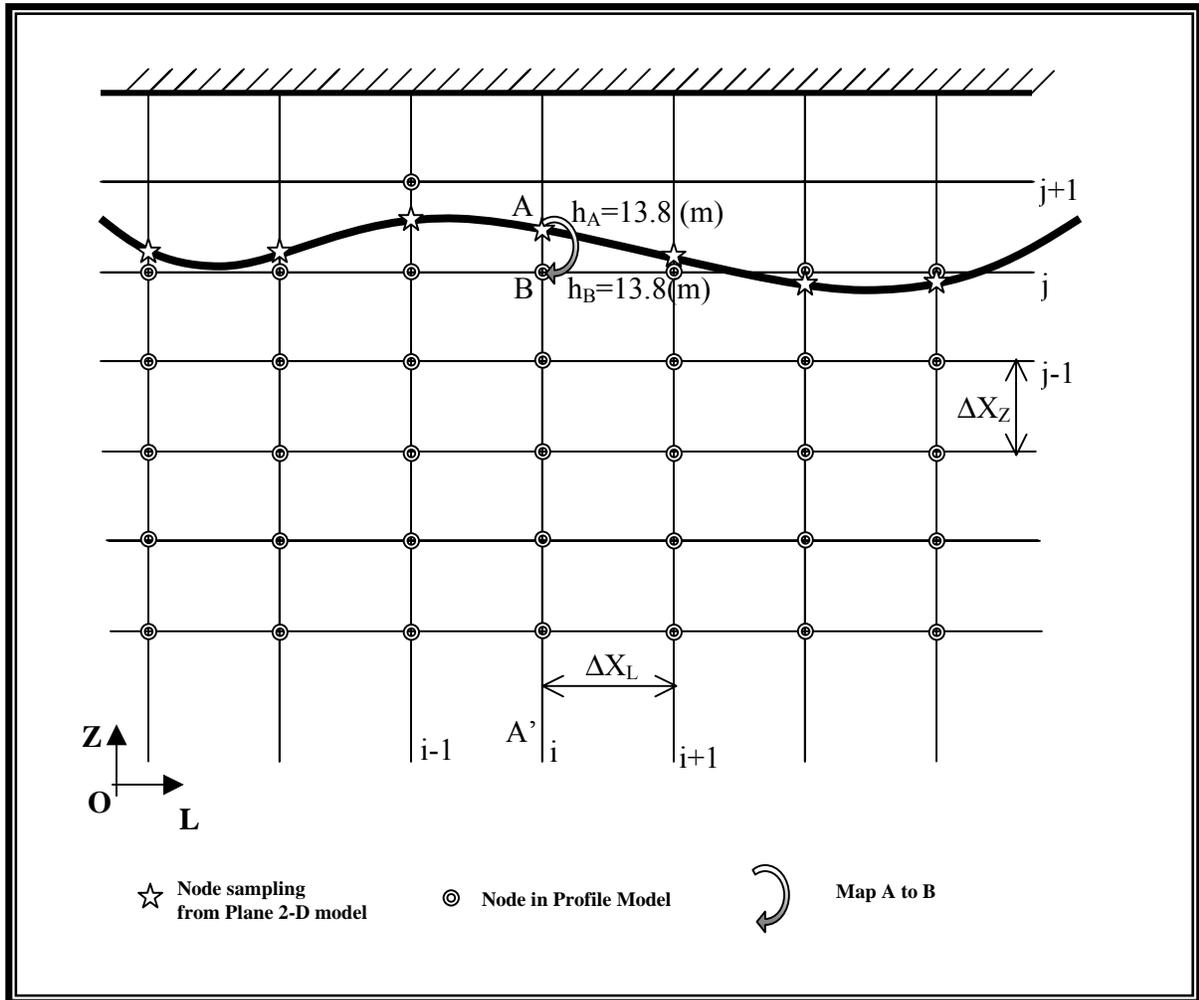


Figure 11 Mapping A Point-related Conceptual Feature to Nodes

## B) Approximation to Poly-line-Related Conceptual Feature

Poly-line-related conceptual features that are presently available in IGW profile model include aquifer's top, aquifer's bottom, water table, river stage, river bed, river leakance, drain elevation, drain leakance, wells with screen thickness and constant head with given distribution along **Z** direction (this feature is available in VF source code only). The basic idea of mapping a Poly-line-related conceptual feature is shown in Figure 12. The whole process can be implemented by following the steps as below,

- a) With  $\Delta X_L$  providing, the considering polyline is divided into  $N_p$  segments for which their two end points are located on vertical grid lines (**Z** direction), this work is automatically done by VB interface when drawing a profile location in a plane 2-D model;
- b) For each segment AB, as shown in Figure 12, maps its two end-points to the nearest nodes A' or  $(i_A, j_A)$  and B' or  $(i_B, j_B)$ ,

$$\begin{cases} i_{A'} = \text{int}\left(\frac{X_L^A - X_L^O}{\Delta X_L}\right) + 1 \\ j_{A'} = \text{int}\left(\frac{X_Z^A - X_Z^O}{\Delta X_Z}\right) + 1 \\ i_{B'} = \text{int}\left(\frac{X_L^B - X_L^O}{\Delta X_L}\right) + 1 \\ j_{B'} = \text{int}\left(\frac{X_Z^B - X_Z^O}{\Delta X_Z}\right) + 1 \end{cases} \quad (105)$$

- c) Assigns values of end-points A and B to nodal points A' and B':

$$\begin{cases} p(i_{A'}, j_{A'}) = p_A \\ p(i_{B'}, j_{B'}) = p_B \end{cases} \quad (106)$$

- d) As shown in Figure 12, there may be the time when the segment AB crosses multiple horizontal grid lines (**L**, Longitudinal direction). Such case will lead

$$ABS(j_{A'} - j_{B'}) \geq 2 \quad (107)$$

If Eq(107) is true, those nodes between the two points of A' and B' must be mapped as the same conceptual feature as the two end-points and their parameter values are calculated by using a linear interpolation,

$$p(i, j) = \left(1 - \frac{s}{l}\right)p_A + \frac{s}{l}p_B, \quad \begin{cases} i = i_{A'}, i_{B'} \\ j = j_{A'} + 1, \dots, i_{B'} - 1 \end{cases} \quad (108)$$

where  $s$  and  $l$  are denoted in Figure 12.

Note that parameter value  $p(i, j)$  can be one of those poly-line-related conceptual features mentioned above.

Steps to approximate to a well with screen thickness ( $Z_{sT}$  and  $Z_{sB}$ ) are little bit different from those described above. After obtaining  $J_{A'}$  and  $J_{B'}$  corresponding to  $Z_{sT}$  and  $Z_{sB}$ , Well's pumping or injection flow rate  $Q_{well}$  was uniformly distributed to the nodes within  $[J_{A'}, J_{B'}]$  according to the following equation:

$$Q_j = \frac{Q_{well}}{ABS(j_{B'} - j_{A'}) + 1}, \quad j = j_{A'}, \dots, j_{B'} \quad (109)$$

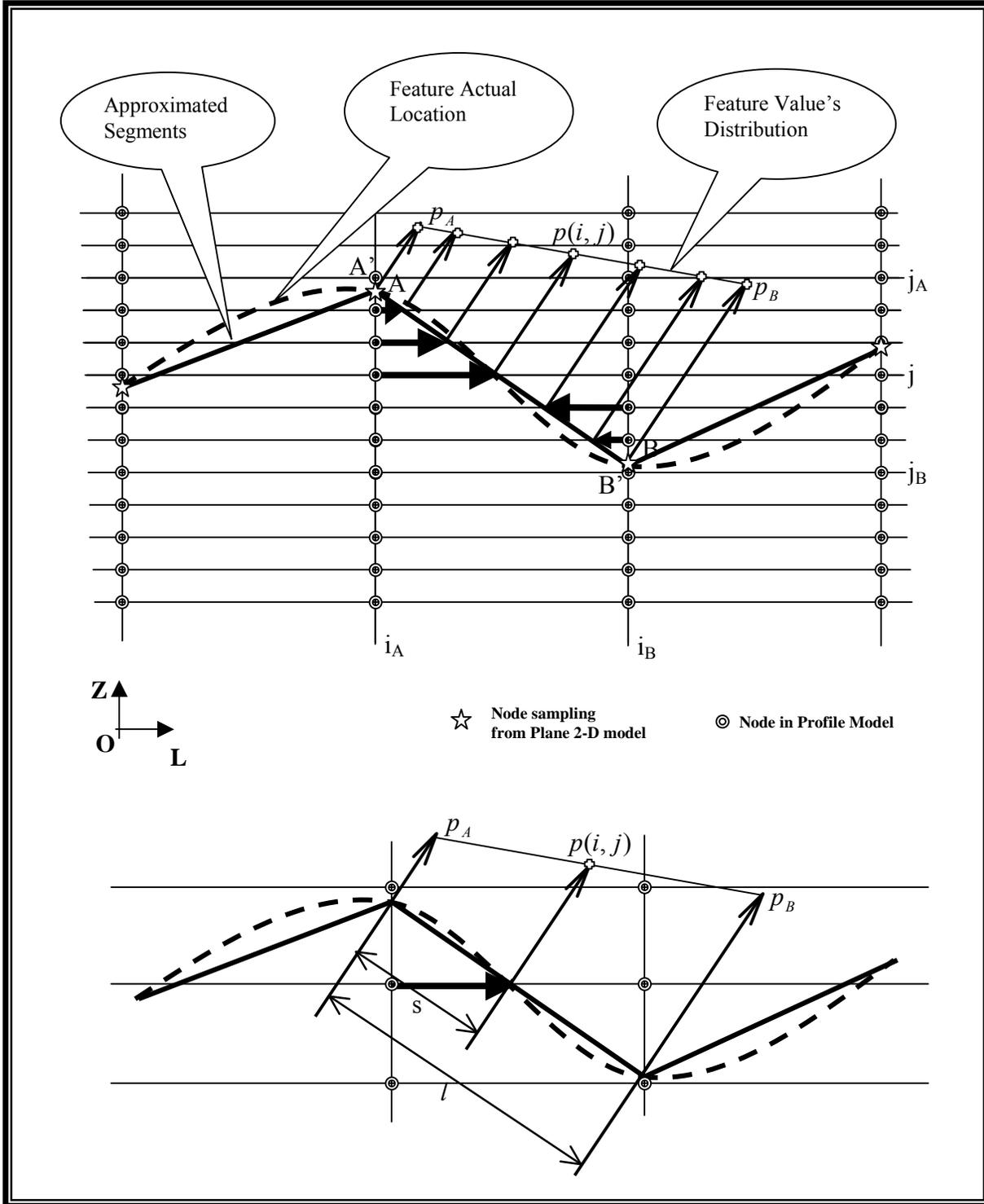


Figure 12 Illustration of Poly-line Related Conceptual Feature Mapping

### C) Determination of Computational Domain

Referring to Figure 13, one can see that the computational domain in profile model depends on a combination of aquifer top (T-T), aquifer bottom (B-B), bed elevation of river or drain (E-E) and water table (W-W). The procedure to determine the computational domain is listed as below

#### a) Upper Boundary

Upper boundary should be determined by T-T, E-E and W-W. The first thing need to do in IGW profile model is to determine the intersection points among T-T, E-E and W-W. From Figure 13, we have:

$$\{P_{WT1}, P_{WT2}\} = \overline{WW} \cap \overline{TT} \quad (110)$$

$$\{P_{WE1}, P_{WE2}, P_{WE3}, P_{WE4}\} = \overline{WW} \cap \overline{EE} \quad (111)$$

With these intersection points providing, taking the figure 13 as an example, the upper boundary is formed by T- $P_{WT1}$ - $P_{WE1}$ - $P_{WE2}$ - $P_{WE3}$ - $P_{WE4}$ - $P_{WT2}$ -T and the position of most top of this boundary,  $Z_{\max}$ , can be further determined. (in Figure 13,  $Z_{\max} = P_{WT1}$ )

#### b) Lower Boundary

Lower boundary usually comes from the aquifer bottom in our one-layer-based profile model. There are some special cases that a river may go through the whole aquifer, then the same method using to configure upper boundary has to be used. The current IGW profile model always uses the aquifer bottom as the computational lower boundary, that is B-B as shown in Figure 13.

The position of most lower of this boundary,  $Z_{\min}$ , can be further determined.

#### c) Computational Domain

A rectangular shape made by most-left boundary which is always a vertical line, most-right boundary which is also a vertical line, line  $Z=Z_{\max}$  and line  $Z=Z_{\min}$  can be finally obtained with  $Z_{\max}$  and  $Z_{\min}$  known. A computational grid system can be configured out in the domain (see Figure 13).

#### d) Inactive Cell

Regions out of the upper and lower boundaries in the computational domain are considered as inactive cells. Figure 13 is self-explanatory.

#### e) Boundary Conditions

In IGW profile model, boundary condition on most-left or most-right boundary is always assumed to be either constant head or no-flow. Lower boundary is assumed to be no-flow boundary condition. However, boundary condition on upper boundary becomes a little bit complicated due to the fact that there are always multiple conceptual features exist simultaneously on the upper boundary. For example, in

figure 13, upper boundary can be divided into the following seven parts, and each part implies different conceptual feature:

$$\left\{ \begin{array}{ll} T - P_{WT1} : & \text{Aquifer Top (confined)} \\ P_{WT1} - P_{WE1} : & \text{Water Table or Head Given (unconfined)} \\ P_{WE1} - P_{WE2} : & \text{River Bed (river cell)} \\ P_{WE2} - P_{WE3} : & \text{Water Table or Head Given (unconfined)} \\ P_{WE3} - P_{WE4} : & \text{River Bed (river cell)} \\ P_{WE4} - P_{WT2} : & \text{Water Table or Head Given (unconfined)} \\ P_{WT2} - T : & \text{Aquifer Top (confined)} \end{array} \right. \quad (112)$$

Hence, the boundary conditions can be described as follows

$$\left\{ \begin{array}{ll} T - P_{WT1} : & \text{no - flow} \\ P_{WT1} - P_{WE1} : & \text{constant head} \\ P_{WE1} - P_{WE2} : & \text{river cell} \\ P_{WE2} - P_{WE3} : & \text{constant head} \\ P_{WE3} - P_{WE4} : & \text{river cell} \\ P_{WE4} - P_{WT2} : & \text{constant} \\ P_{WT2} - T : & \text{no - flow} \end{array} \right. \quad (113)$$

All the features shown in Eq(113) are polyline –related conceptual features. The approximation method described in section **B**) should be used to map these feature to nodal points .

The mapping works mentioned in above sections **A**), **B**) and **C**) were grouped together by Subroutine NEWCELLCNCPT.

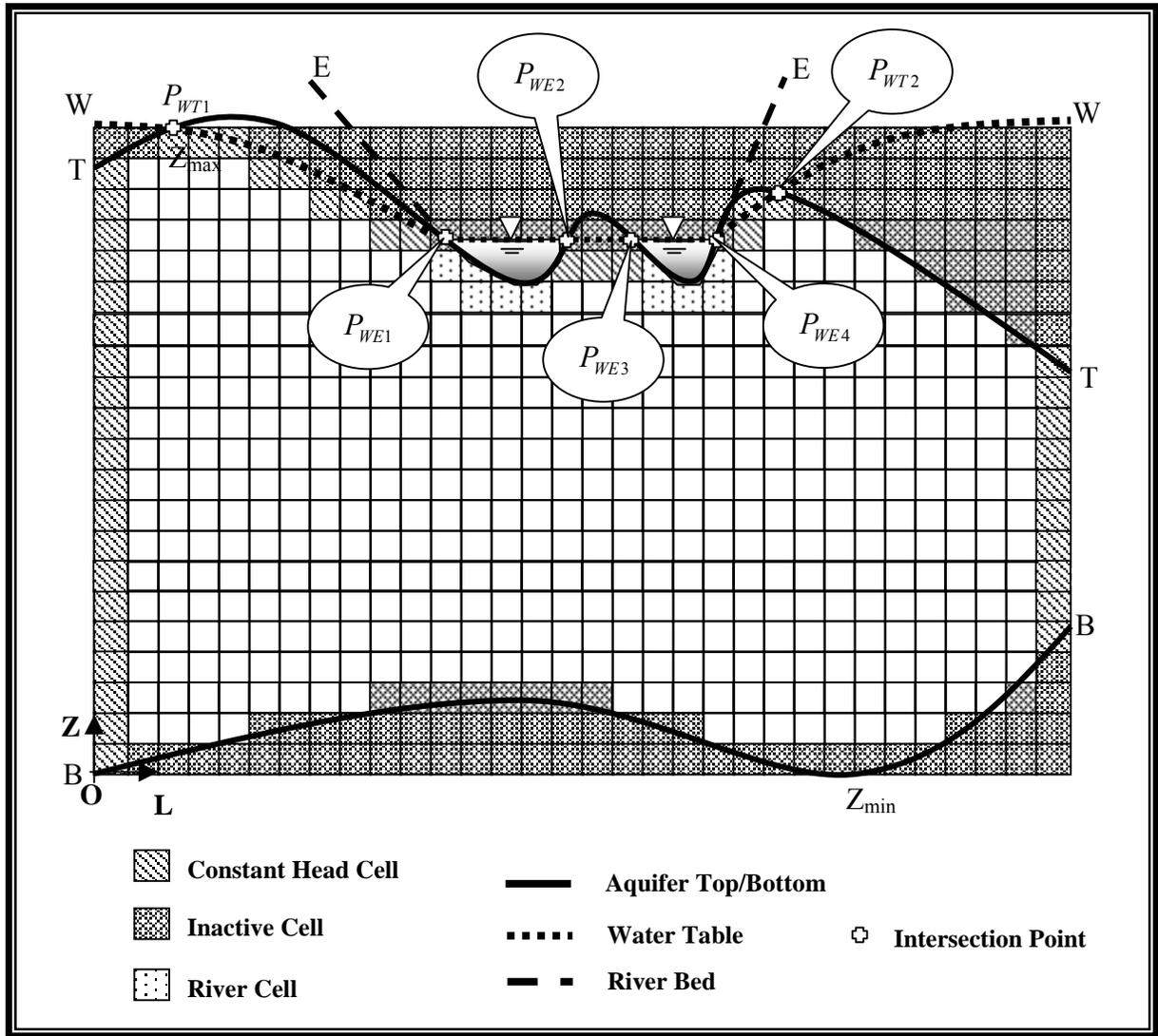


Figure 13 Illustration of Determining Computational Domain

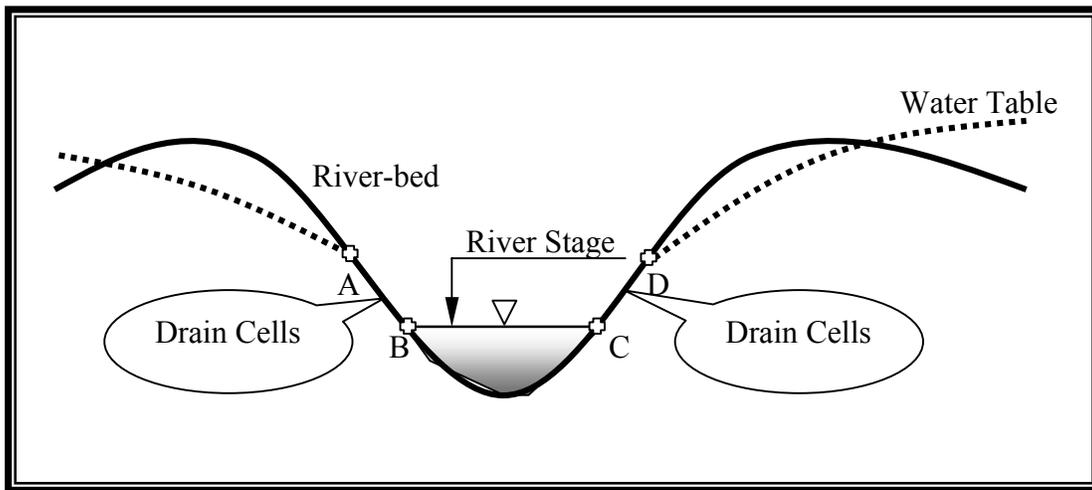
### 2.4.3 Special Treatments

- a) In order to have an appropriate convergent solution to the profile model, ratio of grid spacing in longitudinal and vertical direction,  $\frac{\Delta X_L}{\Delta X_Z}$ , was specified according to the anisotropic ratio as below

$$\frac{\Delta X_L}{\Delta X_Z} = \sqrt{R_{anisf}} \tag{114}$$

where  $R_{anisf}$  is the anisotropic ratio:  $R_{anisf} = \frac{K'_{xx}}{K'_{zz}}$ .

- b) River cells were assigned to the cells just adjacent to the river-bed cells (see Figure 13)
- c) If there is any case of that river-bed is larger than river stage as shown in Figure 14, then part of the river cells (AB and CD in figure 14) were turned into to drain cells ;

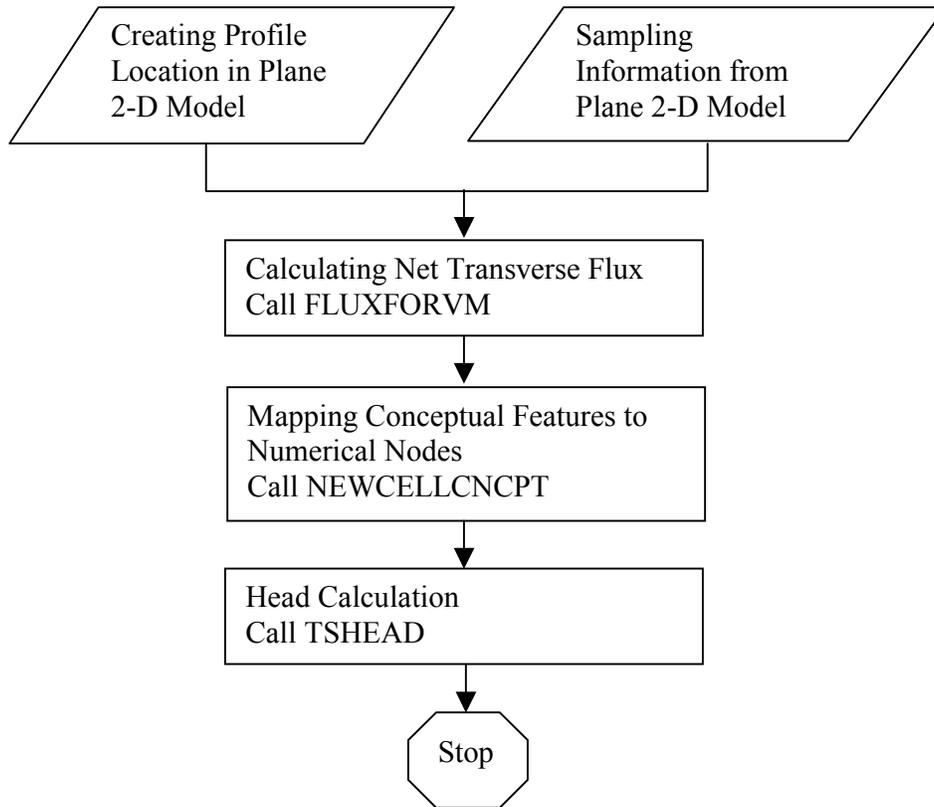


**Figure 14 River Cell ⇒ Drain Cell**

- d) For transient case, no transient effectiveness was considered in the current profile model except the transient water table obtained from the plane 2-D model. In other word, steady-state calculation is always performed in IGW profile model no matter what the case is transient or not.

**2.4.4 Numerical Solution Procedure Flow Chart**

The sequence of operations for Profile Model is illustrated in the following flow diagram:



### 3. Three Dimensional Model Source Code

#### 3.1 Flow

##### 3.1.1 Governing Equation

The partial differential equation describing flow in porous medial is Eq(1), and can be re-written as

$$S_s \frac{\partial h}{\partial t} = \frac{\partial}{\partial X_i} \left( K_{ij} \frac{\partial h}{\partial X_j} \right) + q_s \quad (115)$$

Where

- $S_s$  storage coefficient of the porous materials
- $h$  hydraulic head
- $K_{ij}$  hydraulic conductivity tensor
- $X_i$  Cartesian coordinate.
- $q_s$  source/sink

With knowing head  $h$ , the seepage or linear pore velocity can be defined as

$$v_i = - \frac{K_{ij}}{n} \frac{\partial h}{\partial X_j} \quad (116)$$

Where  $n$  is the porosity of the porous medium.

As mentioned in 2-D models, the hydraulic conductivity tensor,  $K_{ij}$ , should have nine components in 3-D cases and those non-principal components (cross terms) very easily cause the ‘negative coefficient’ in the derived discretized matrix. The Rotation Control Volume Technique is needed to avoid such situation. In IGW 3-D, only the first principal hydraulic conductivity,  $K'_{xx}$ , is given. The second and third ones,  $K'_{yy}$ ,  $K'_{zz}$  are obtained from the given anisotropy ratio. Again, as described in 2-D models, when  $K'_{ii}$  are not aligned with the  $X_i$  coordinate axes, anisotropy orientation angles,  $\theta$  and  $\beta$  (see Figure 15) will be given in order to determine the rest of six components in the tensor  $K_{ij}$ . Being a second-rank tensor, the transformation of  $K_{ij}$  from one coordinate system (say,  $X_i$ ) into  $K'_{ij}$  in another coordinate system (say,  $X'_i$ ), is given by

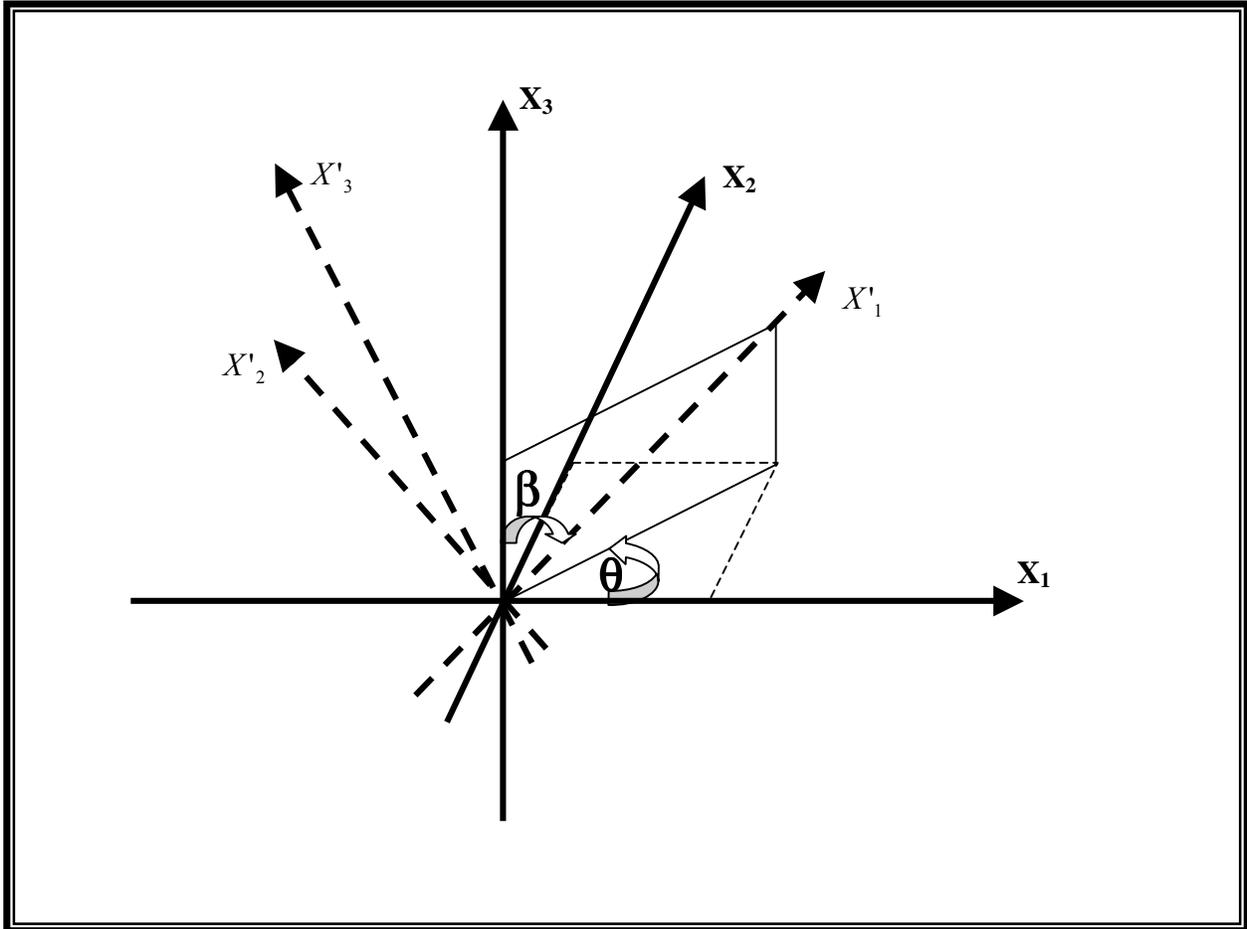
$$K_{pq} = K'_{ij} \alpha_{pi} \alpha_{qj}, \quad i, j, p, q = 1, 2, 3 \quad (117)$$

where  $\alpha_{mn}$  is the direction cosine between the axes  $X_m$  and  $X'_n$ , which can be calculated by the given anisotropy orientation angles,  $\theta$  and  $\beta$ .

If the  $X'_i$  coincide with the principal directions of hydraulic conductivity, we have:

$$K_{ij} = K'_{11} \alpha_{i1} \alpha_{j1} + K'_{22} \alpha_{i2} \alpha_{j2} + K'_{33} \alpha_{i3} \alpha_{j3}, \quad i, j = 1, 2, 3 \quad (118)$$

Eq(118) was used in IGW 3-D model to convert the input  $K'_{ii}$  from VB interface to  $K_{ij}$  in which there may be non-zero cross terms.



**Figure 15 Two Orientation Angles  $\theta$  and  $\beta$  Needed to Define the Geometric Anisotropy of Hydraulic Conductivity Structure in 3D**

### 3.1.2 Grid Layout

As described in 2-D models, in IGW, parameters are assigned to a block or a cell. Placing a representative node in the center of each cell forms the grid layout using in our spatial discretizing. Figure 16 shows a typical node-cell and its neighborhood of current use. The figure is self-explanatory, and those grid related geometrical quantities used in the scheme are illustrated in the related figure 17 to figure 19. The following table is a list of the variables using in the code and their counterparts in these figure,

Table 9

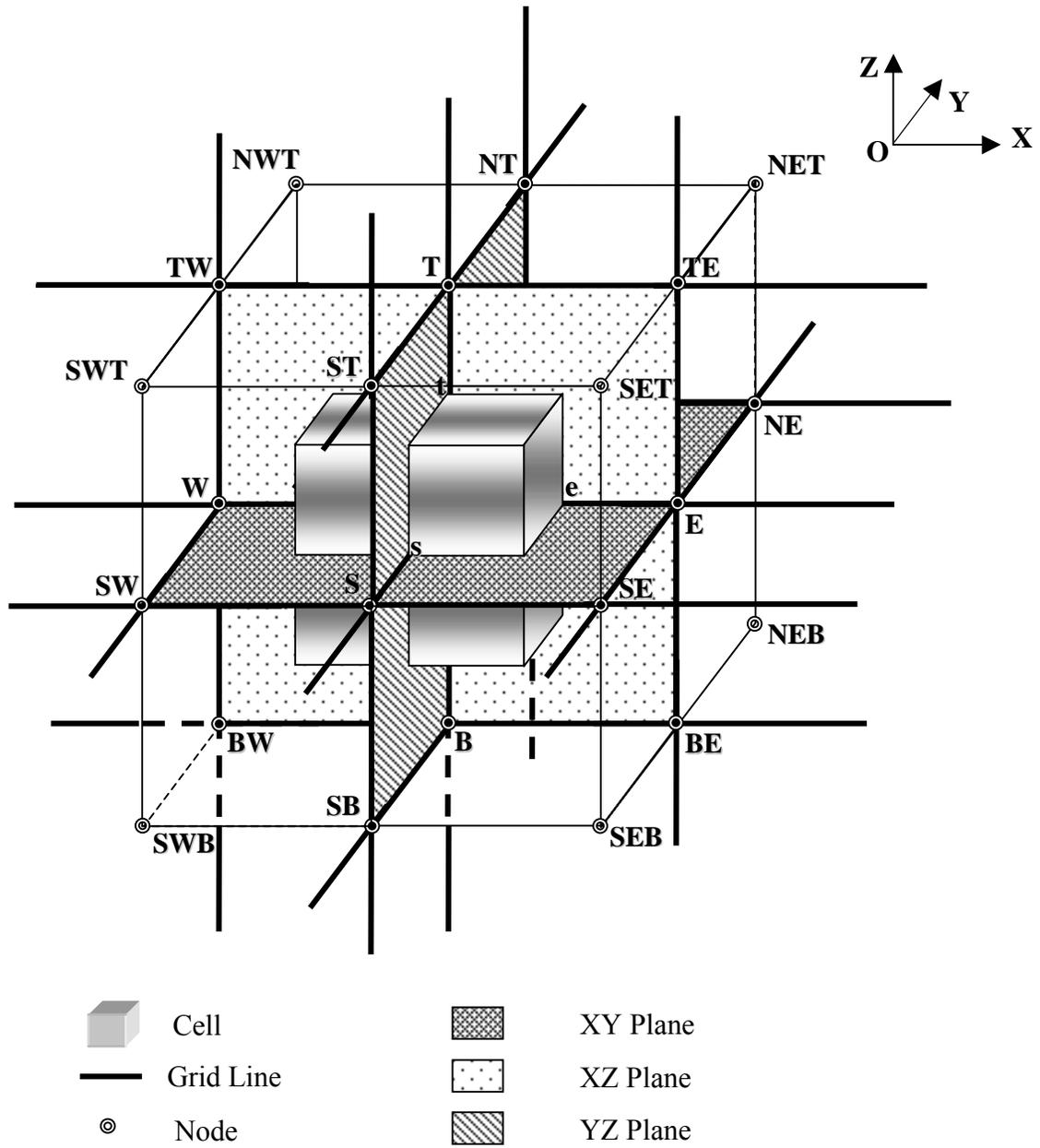
	Notation in Figure 17 to 19	Variable in Code
X coordinate	$X_{ijk}$	X(I,J,K)
Y coordinate	$Y_{ijk}$	Y(I,J,K)
Z coordinate	$Z_{ijk}$	Z(I,J,K)
X-Grid Spacing	$\Delta X_i$	HX(I)
Y-Grid Spacing	$\Delta Y_j$	HY(J)
Z-Grid Spacing	$\Delta Z_{ijk}$	Calculated from aquifer's thickness
X-CV area	$\Delta X_s$	DXS(I)
Y-CV area	$\Delta Y_s$	DYS(J)
Z-CV area	$\Delta Z_s$	Calculated from aquifer's thickness

It is noted that Visual Fortran code is written based on non-uniform grid spacing, although a uniform grid spacing has been used in IGW interface. The quantities at the cell-face, such as  $K_{ij}^e$ , must be evaluated in terms of nodal values before proceeding the calculation. There are many methods available to handle it, such as those of linear interpolation, harmonic averaged, etc. Harmonic mean was adopted in IGW 3-D model. X-velocity  $u_x$  is located on the East and West cell-faces, Y-velocity  $u_y$  is located on the North and South cell-faces and Z-velocity  $u_z$  is located on the Top and Bottom cell-faces. This arrangement will make use of cell-face  $K_{ij}$  which is the real one involving in the calculation, and central difference scheme to approximate the head gradient between two known nodal heads. To obtain the velocities at nodes, an interpolation method or simple arithmetic mean has to be used. Arithmetic mean has been used in IGW 3-D model. Subroutine NEWUVXJQT3D has the details. A linear interpolation should be used to evaluate the nodal  $u_z$  if the elevation of aquifer top or bottom is not uniform.

To adapt non-uniform aquifer thickness, non-constant Z-direction grid spacing is used in IGW 3-D model. Unlike the grid spacing in other two directions having the following feature:

$$\begin{cases} \Delta X_{i,j,k} = \Delta X_i, & j, k = 1, 2, \dots \\ \Delta Y_{i,j,k} = \Delta Y_j, & i, k = 1, 2, \dots \end{cases} \quad (119)$$

$\Delta Z_{i,j,k}$  is no longer constant for the same index k but dependent with indexes i and j in 3-D models. This will result in an irregular grid system for the case of non-uniform aquifer top or bottom elevation. An example of such case is shown in Figure 20. In the following course of derivation, for the simplicity purpose,  $\Delta Z_{i,j,k}$  is still assumed to be constant in those equations but actually non-constant in the source code.



The invisible nodes in this figure include **P, N, NW, NB, NWB**

**Figure 16 Grid Layout of 3-D Model**

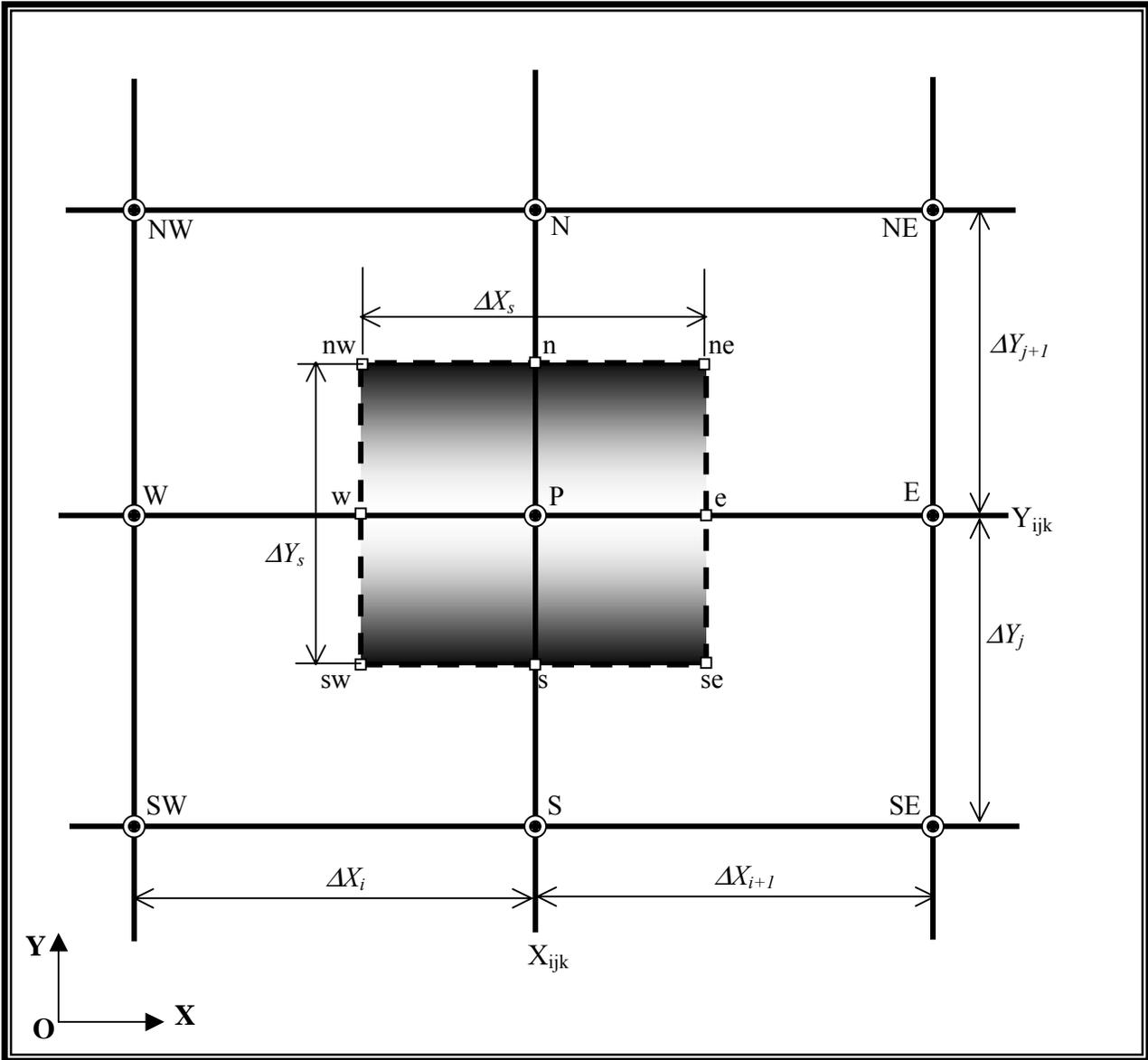


Figure 17. Typical Cell and Notation in XY-Plane

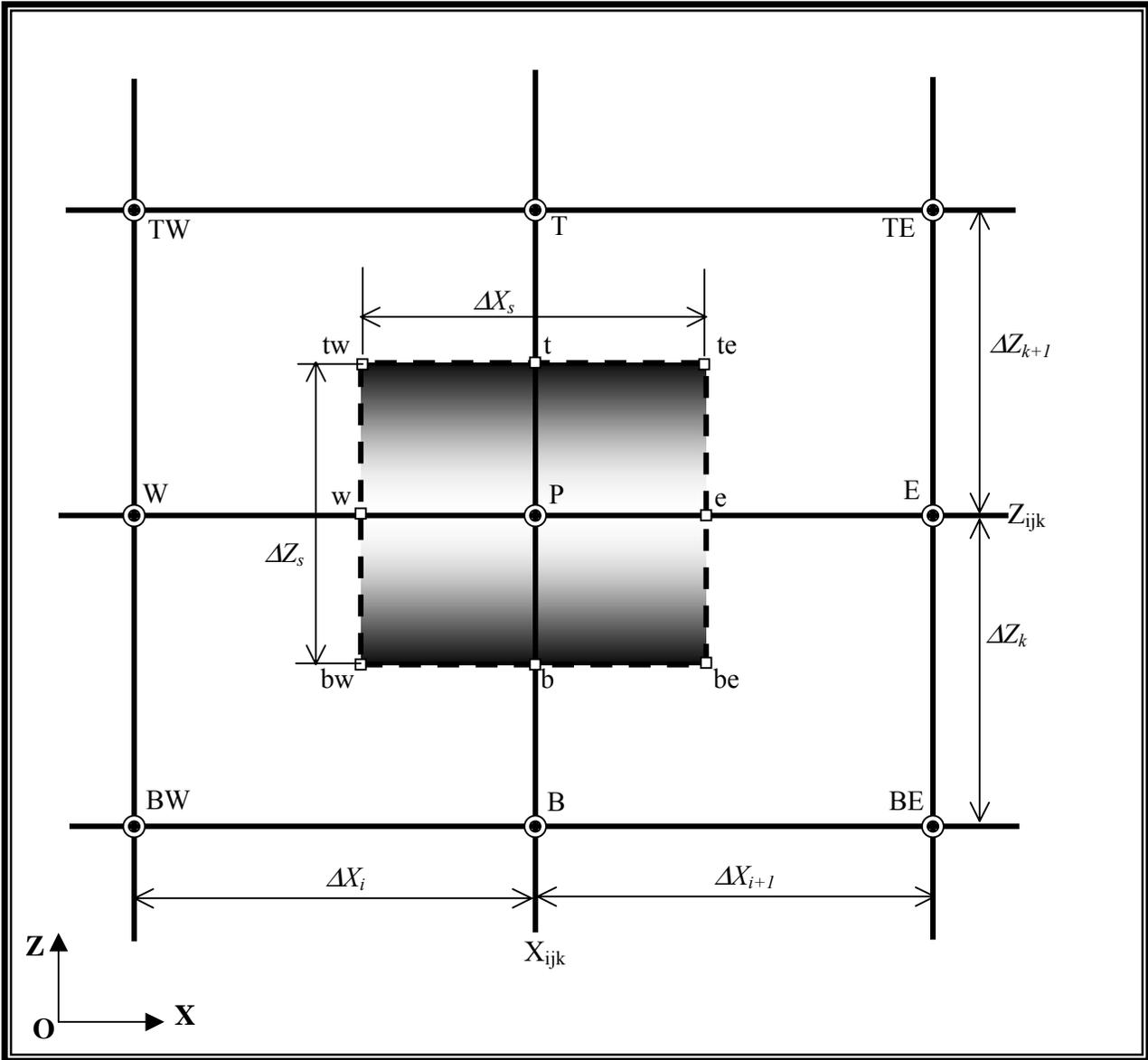


Figure 18. Typical Cell and Notation in XZ-Plane

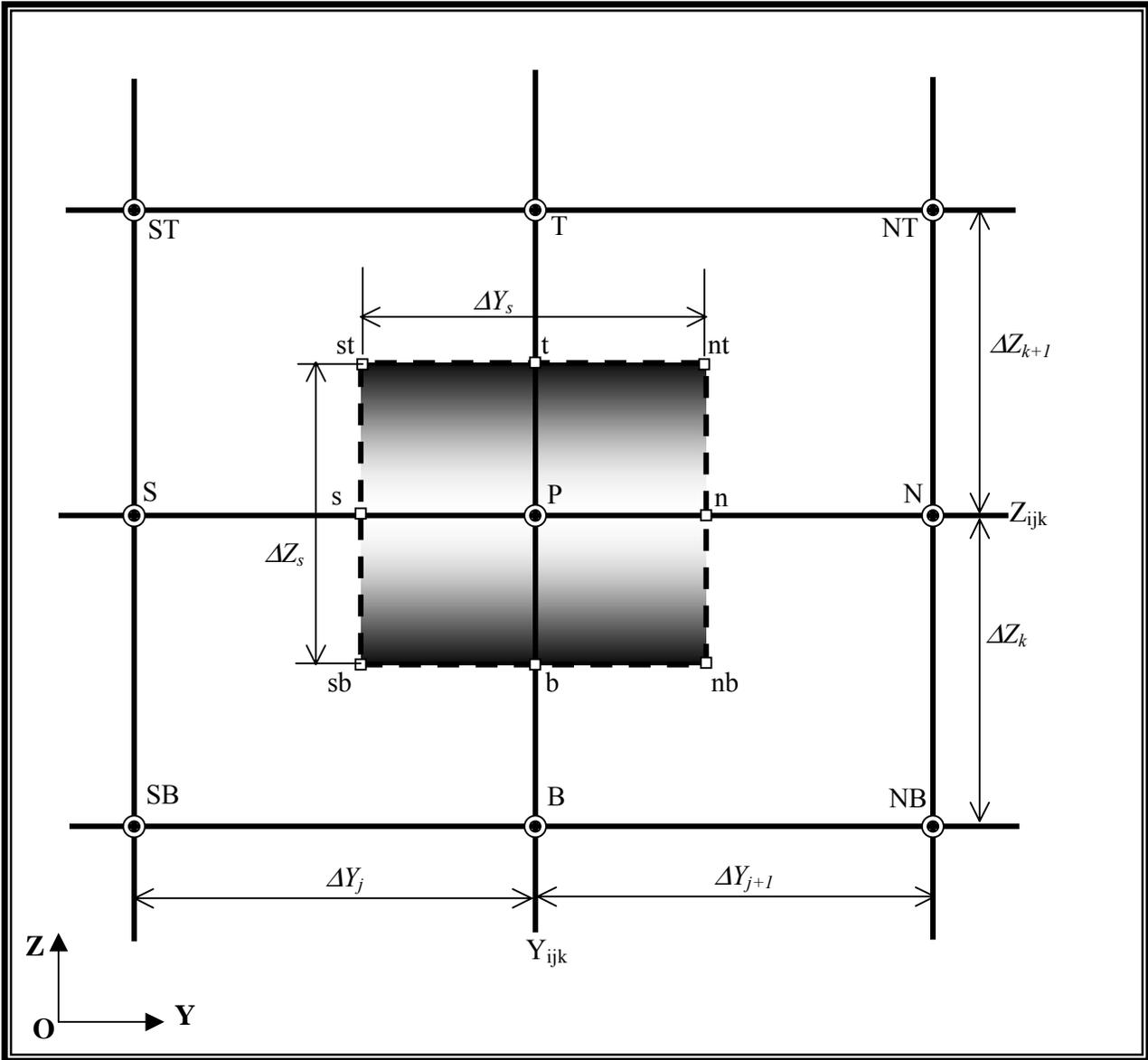
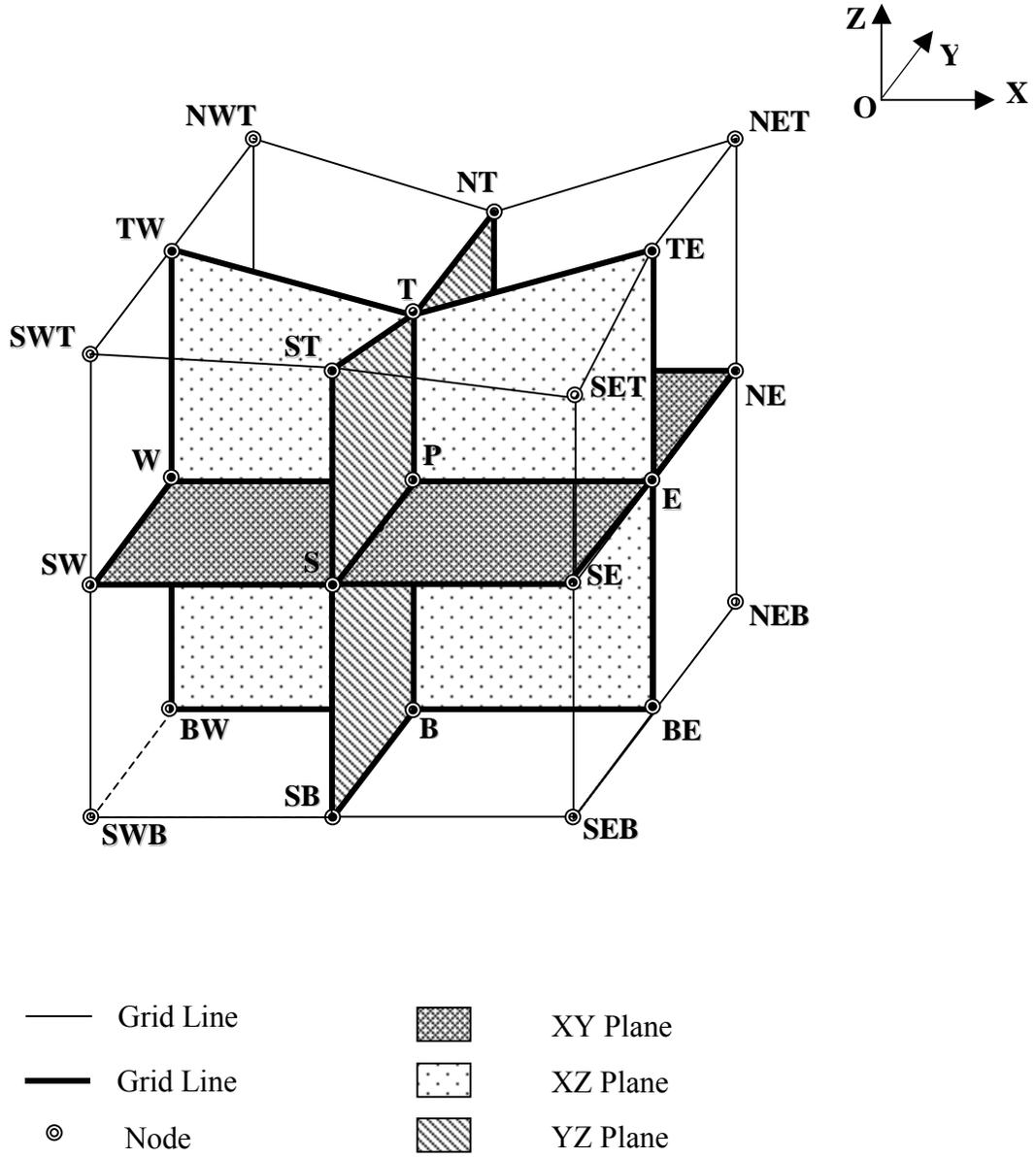


Figure 19. Typical Cell and Notation in YZ-Plane



The invisible nodes in this figure include **N, NW, NB, NWB**

**Figure 20 Irregular Grid Layout of 3-D Model**

### 3.1.3 Scheme to Discrete Equation

#### A) Traditional Control Volume Technique

As mentioned above, despite of the fact that  $K_{ij}$  easily causes a ‘negative coefficient’ in the derived discretized matrix based on the traditional control volume technique, traditional control volume technique will still be employed in our IGW 3-D model for purpose of comparison.

#### a) Approximation to Diffusion Term

Referring to Figure 16 to Figure 19 and applying control volume technique, diffusion terms of Eq(115) may be written as,

$$Diff = \Delta X_s \Delta Y_s \Delta Z_s \left( \frac{J_e - J_w}{\Delta X_s} + \frac{J_n - J_s}{\Delta Y_s} + \frac{J_t - J_b}{\Delta Z_s} \right) \quad (120)$$

Where  $J_e$ ,  $J_w$ ,  $J_n$ ,  $J_s$ ,  $J_t$  and  $J_b$  are the fluxes through east, west, north, south, top and bottom cell-faces respectively. They have the following forms:

$$\left\{ \begin{array}{l} J_e = K_{xx}^e \frac{h_E - h_P}{\Delta X} + K_{xy}^e \frac{h_{ne} - h_{se}}{\Delta Y_s} + K_{xz}^e \frac{h_{te} - h_{be}}{\Delta Z_s} \\ J_w = K_{xx}^w \frac{h_P - h_W}{\Delta X} + K_{xy}^w \frac{h_{nw} - h_{sw}}{\Delta Y_s} + K_{xz}^w \frac{h_{tw} - h_{bw}}{\Delta Z_s} \\ J_n = K_{yy}^n \frac{h_N - h_P}{\Delta Y} + K_{yx}^n \frac{h_{ne} - h_{nw}}{\Delta X_s} + K_{yz}^n \frac{h_{nt} - h_{nb}}{\Delta Z_s} \\ J_s = K_{yy}^s \frac{h_P - h_S}{\Delta Y} + K_{yx}^s \frac{h_{se} - h_{sw}}{\Delta X_s} + K_{yz}^s \frac{h_{st} - h_{sb}}{\Delta Z_s} \\ J_t = K_{zz}^t \frac{h_T - h_P}{\Delta Z} + K_{zx}^t \frac{h_{te} - h_{tw}}{\Delta X_s} + K_{zy}^t \frac{h_{nt} - h_{st}}{\Delta Y_s} \\ J_b = K_{zz}^b \frac{h_P - h_B}{\Delta Z} + K_{zx}^b \frac{h_{be} - h_{bw}}{\Delta X_s} + K_{zy}^b \frac{h_{nb} - h_{sb}}{\Delta Y_s} \end{array} \right. \quad (121)$$

All symbols in Eq(121) were denoted in Figure 16 to Figure 19.

Note that non-nodal quantities in Eq(121) must be evaluated in terms of nodal values. The same simple four points averaged scheme as used in 2-D model has been used in IGW 3-D model. After re-arranging Eq(121), Eq(120) becomes

$$\begin{aligned} Diff = & a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_T h_T + a_B h_B \\ & + a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} \\ & + a_{NT} h_{NT} + a_{NB} h_{NB} + a_{ST} h_{ST} + a_{SB} h_{SB} \\ & + a_{TW} h_{TW} + a_{BW} h_{BW} + a_{TE} h_{TE} + a_{BE} h_{BE} \\ & - a_P h_P \end{aligned} \quad (122)$$

Where

$$\left\{ \begin{aligned}
 a_E &= \frac{\Delta Y_s \Delta Z_s K_{xx}^e}{\Delta X} + \frac{K_{yx}^n - K_{yx}^s}{4} \Delta Z_s + \frac{K_{zx}^t - K_{zx}^b}{4} \Delta Y_s \\
 a_W &= \frac{\Delta Y_s \Delta Z_s K_{xx}^w}{\Delta X} - \frac{K_{yx}^n - K_{yx}^s}{4} \Delta Z_s - \frac{K_{zx}^t - K_{zx}^b}{4} \Delta Y_s \\
 a_N &= \frac{\Delta X_s \Delta Z_s K_{yy}^n}{\Delta Y} + \frac{K_{xy}^e - K_{xy}^w}{4} \Delta Z_s + \frac{K_{zy}^t - K_{zy}^b}{4} \Delta X_s \\
 a_S &= \frac{\Delta X_s \Delta Z_s K_{yy}^s}{\Delta Y} - \frac{K_{xy}^e - K_{xy}^w}{4} \Delta Z_s - \frac{K_{zy}^t - K_{zy}^b}{4} \Delta X_s \\
 a_T &= \frac{\Delta X_s \Delta Y_s K_{zz}^t}{\Delta Z} + \frac{K_{xz}^e - K_{xz}^w}{4} \Delta Y_s + \frac{K_{yz}^n - K_{yz}^s}{4} \Delta X_s \\
 a_B &= \frac{\Delta X_s \Delta Y_s K_{zz}^b}{\Delta Z} - \frac{K_{xz}^e - K_{xz}^w}{4} \Delta Y_s - \frac{K_{yz}^n - K_{yz}^s}{4} \Delta X_s \\
 a_{NE} &= \frac{K_{xy}^e + K_{yx}^n}{4} \Delta Z_s, & a_{NW} &= -\frac{K_{xy}^w + K_{yx}^n}{4} \Delta Z_s \\
 a_{SE} &= -\frac{K_{xy}^e + K_{yx}^s}{4} \Delta Z_s, & a_{SW} &= \frac{K_{xy}^w + K_{yx}^s}{4} \Delta Z_s \\
 a_{TE} &= \frac{K_{xz}^e + K_{zx}^t}{4} \Delta Y_s, & a_{BE} &= -\frac{K_{xz}^e + K_{zx}^b}{4} \Delta Y_s \\
 a_{TW} &= -\frac{K_{xz}^w + K_{zx}^t}{4} \Delta Y_s, & a_{BW} &= \frac{K_{xz}^w + K_{zx}^b}{4} \Delta Y_s \\
 a_{NT} &= \frac{K_{yz}^n + K_{zy}^t}{4} \Delta X_s, & a_{NB} &= -\frac{K_{yz}^n + K_{zy}^b}{4} \Delta X_s \\
 a_{ST} &= -\frac{K_{yz}^s + K_{zy}^t}{4} \Delta X_s, & a_{SB} &= \frac{K_{yz}^s + K_{zy}^b}{4} \Delta X_s \\
 a_P &= a_E + a_W + a_N + a_S + a_T + a_S + a_{NE} + a_{NW} + a_{SE} + a_{SW} \\
 &\quad + a_{TE} + a_{BE} + a_{TW} + a_{BW} + a_{NT} + a_{NB} + a_{ST} + a_{SB}
 \end{aligned} \right. \tag{123}$$

This process was implemented in Subroutine OLDCOEFFLOW3D. A derived type variable CST2 was used to store these coefficients:

Table 10

Notation in Eq(123)	Variable in Code
$a_E$	CST2 (I,J,K)%SE
$a_W$	CST2 (I,J,K)%SW
$a_N$	CST2 (I,J,K)%SN
$a_S$	CST2 (I,J,K)%SS
$a_T$	CST2 (I,J,K)%ST

$a_B$	CST2 (I,J,K)%SB
$a_{NE}$	CST2 (I,J,K)%SNE
$a_{NW}$	CST2 (I,J,K)%SNW
$a_{SE}$	CST2 (I,J,K)%SSE
$a_{SW}$	CST2 (I,J,K)%SSW
$a_{TE}$	CST2 (I,J,K)%STE
$a_{BE}$	CST2 (I,J,K)%SBE
$a_{TW}$	CST2 (I,J,K)%STW
$a_{BW}$	CST2 (I,J,K)%SBW
$a_{NT}$	CST2 (I,J,K)%SNT
$a_{NB}$	CST2 (I,J,K)%SNB
$a_{ST}$	CST2 (I,J,K)%SST
$a_{SB}$	CST2 (I,J,K)%SSB
$a_P$	CST2 (I,J,K)%SP

From Eq(123), it is very obvious that  $a_{NE}$ ,  $a_{NW}$ ,  $a_{SE}$ ,  $a_{SW}$ ,  $a_{TE}$ ,  $a_{BE}$ ,  $a_{TW}$ ,  $a_{BW}$ ,  $a_{NT}$ ,  $a_{NB}$ ,  $a_{ST}$ , and  $a_{SB}$ , may easily turn into negative ones, for example,  $a_{NW} < 0$  and  $a_{SE} < 0$  when  $K_{xy} > 0$ ; or  $a_{NE} < 0$  and  $a_{SW} < 0$  when  $K_{xy} < 0$ .

### b) Approximation to Time Derivative Term

For transient flow, a backward finite difference scheme is used to approximate the time-derivative term

$$S \frac{\partial h}{\partial t} = S \frac{h^{n+1} - h^n}{\Delta t} \Delta X_s \Delta Y_s \Delta Z_s = a_p^t h^{n+1} - S_f^t \quad (124)$$

Where  $h^{n+1}$  is the head at new time level  $n+1$ ,  $h^n$  is the head at old time level  $n$  and

$$\begin{cases} a_p^t = \Delta X_s \Delta Y_s \Delta Z_s \frac{S}{\Delta t} \\ S_f^t = \Delta X_s \Delta Y_s \Delta Z_s \frac{S}{\Delta t} h^n \end{cases} \quad (125)$$

### c) Approximation to Source/Sink Term

Source/sink term may include head independent one such as well or recharge and head dependent one such as river or drain. General head and evapotranspiration were included in IGW 3-D model. The evapotranspiration simulation reflects the effects of plant transpiration and direct evaporation in removing water from the saturated ground water regime. The approach is based on the following assumptions: (1) when the water table is at or above a specified elevation,  $h_s$ , termed the “bottom elevation of surface land” in this guide, evapotranspiration loss from water table occurs at a maximum rate specified by the user; (2)

when the depth of the water table below the  $h_s$ , exceeds a specified interval, termed the “extinction depth” or “cutoff depth” in this guide, evapotranspiration loss from water table ceases; and (3) between these limits, evapotranspiration loss from water table varies linearly with water table elevation (see Figure 21).

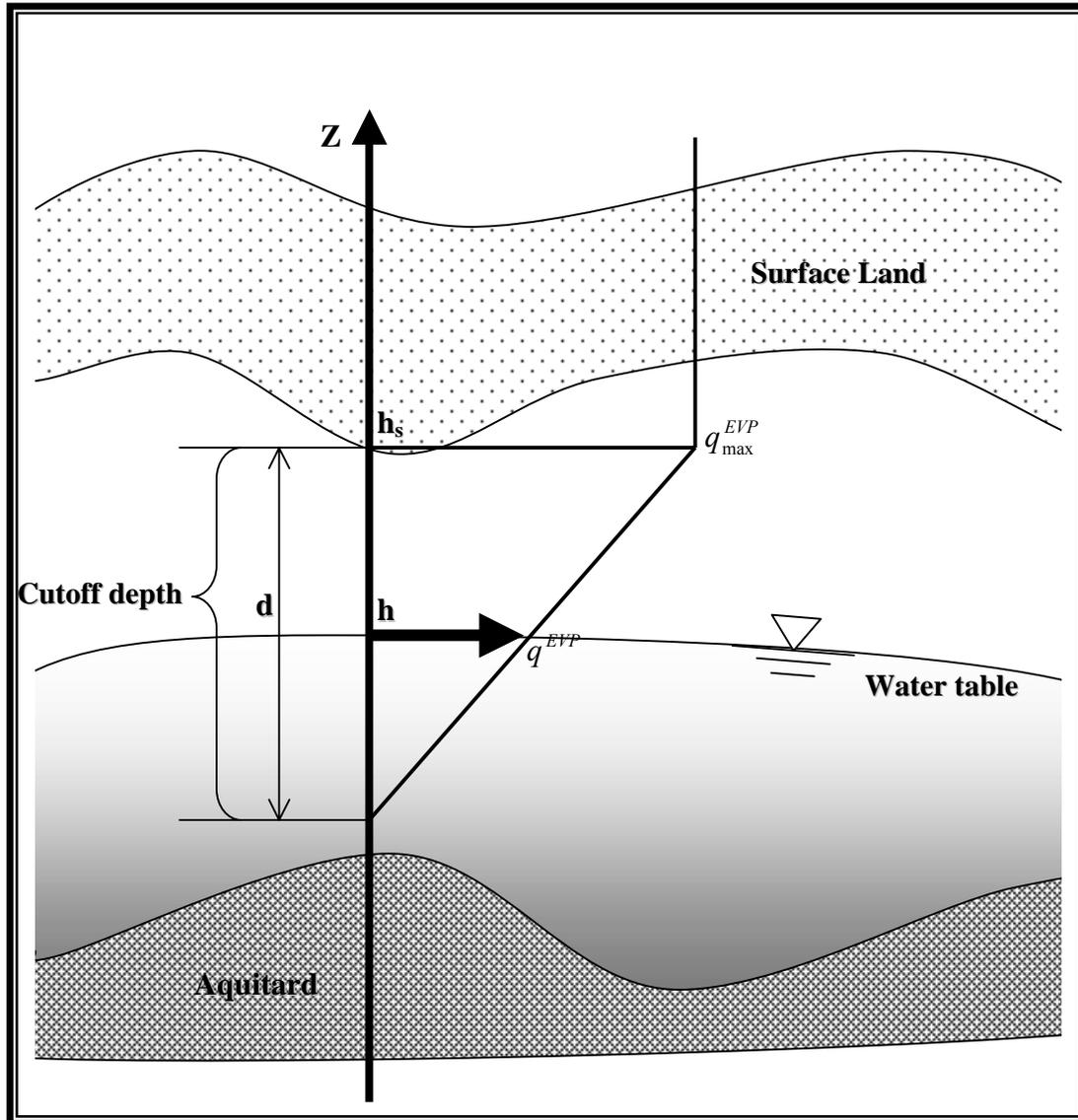


Figure 21 Illustration of Evapotranspiration Distribution

Head independent source/sink terms can be directly added to Right Hand Side (RHS) vector, however, head dependent source/sink terms must be divided into two parts: one goes to RHS vector, another one goes to the main diagonal entry after linearized . In general,  $Q$  can be expressed as

$$Q = a_p^Q h_p + S_f^Q \tag{126}$$

Table 11 is a list of source/sink available in IGW 3-D model and their corresponding  $a_p^Q$  and  $S_f^Q$ .

Tabel 11

Type of Source/Sink	$a_p^Q$	$S_f^Q$	Marks
Well	0	$Q_{well}$	
Recharge	0	$q\Delta X_s \Delta Y_s$	
River	$L_{river} \Delta X_s \Delta Y_s$	$L_{river} \Delta X_s \Delta Y_s h_{river}$	$h > R_{bed}$
	0	$L_{river} \Delta X_s \Delta Y_s (h_{river} - R_{bed})$	$h < R_{bed}$
Drain	$L_{drain} \Delta X_s \Delta Y_s$	$L_{drain} \Delta X_s \Delta Y_s D_{bed}$	$h > D_{bed}$
General Head	$L_G \Delta X_s \Delta Y_s$	$L_G \Delta X_s \Delta Y_s h_{source}$	
Evapotranspiration	0	$q_{max}^{EVP} \Delta X_s \Delta Y_s$	$h > h_s$
	$\frac{q_{max}^{EVP} \Delta X_s \Delta Y_s}{d}$	$\frac{q_{max}^{EVP} \Delta X_s \Delta Y_s}{d} (h_s - d)$	$(h_s - d) < h < h_s$
	0	0	$h < (h_s - d)$

Where

- $Q_{well}$  well's flow rate ( $L^3/T$ );
- $q$  recharge rate;
- $L_{river}$  river leakance ;
- $L_{drain}$  drain leakance ;
- $L_G$  general head leakance ;
- $h_{river}$  river stage;
- $h_{source}$  source head;
- $R_{bed}$  bottom elevation of river;
- $D_{bed}$  bottom elevation of drain
- $q_{max}^{EVP}$  the maximum value of evapotranspiration loss
- $h_s$  elevation at which the maximum value of evapotranspiration loss occurs
- $d$  the cutoff or extinction depth

There may be different kinds of source/sink applying at the same nodal point, therefore,  $a_p^Q$  and  $S_f^Q$  in Eq(126) actually are summation of the contributions from all kinds of source/sink. Implementation of this process was done in Subroutine ADDQS13D (head independent source/sink) and Subroutine ADDQS23D(head dependent source/sink).

#### d) Matrix Coefficients Assembling

From Eq(122), Eq(124) and Eq(126), readily gives a set of linear equations as follow

$$\begin{aligned}
 (a_p + a_p^t + a_p^Q)h_p &= a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_T h_T + a_B h_B \\
 &+ a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} \\
 &+ a_{NT} h_{NT} + a_{NB} h_{NB} + a_{ST} h_{ST} + a_{SB} h_{SB} \\
 &+ a_{TW} h_{TW} + a_{BW} h_{BW} + a_{TE} h_{TE} + a_{BE} h_{BE} \\
 &+ S_f^Q + S_f^t
 \end{aligned} \tag{127}$$

It is seen that there are 19 diagonal entries in the derived-coefficient matrix when applying traditional control volume technique.

#### e) Matrix Solver

Solution to Eq(127) can be obtained by using SOR technique or other efficient matrix solvers. SOR was adopted in IGW 3-D model. Its iterative equation can be expressed as follow

$$\begin{aligned}
 h_p^{k+1} = h_p^k + \frac{\alpha}{(a_p + a_p^t + a_p^Q)} \{ &a_E h_E^k + a_W h_W^{k+1} + a_N h_N^k + a_S h_S^{k+1} + a_T h_T^k + a_B h_B^{k+1} \\
 &+ a_{NE} h_{NE}^k + a_{NW} h_{NW}^k + a_{SE} h_{SE}^{k+1} + a_{SW} h_{SW}^{k+1} \\
 &+ a_{NT} h_{NT}^k + a_{NB} h_{NB}^k + a_{ST} h_{ST}^k + a_{SB} h_{SB}^{k+1} \\
 &+ a_{TW} h_{TW}^k + a_{BW} h_{BW}^{k+1} + a_{TE} h_{TE}^k + a_{BE} h_{BE}^{k+1} \\
 &+ S_f^Q + S_f^t - (a_p + a_p^t + a_p^Q)h_p^k \}
 \end{aligned} \tag{128}$$

Where  $k$  is index of iteration number and  $\alpha$  is relaxation factor.

The final matrix assembling and iterative processes are carried out in Subroutine SORHEADT3D.

### B) Rotational Control Volume Technique

As described in 2-D model, the proposed rotational control volume technique can give surety that the coefficients in the derived discretized matrix are always positive. A typical rotational control volume is shown in Figure 22. Always keeping the local  $X_L Y_L Z_L$  coordinate system (see Figure 22) in mind will help understand the following derivation.

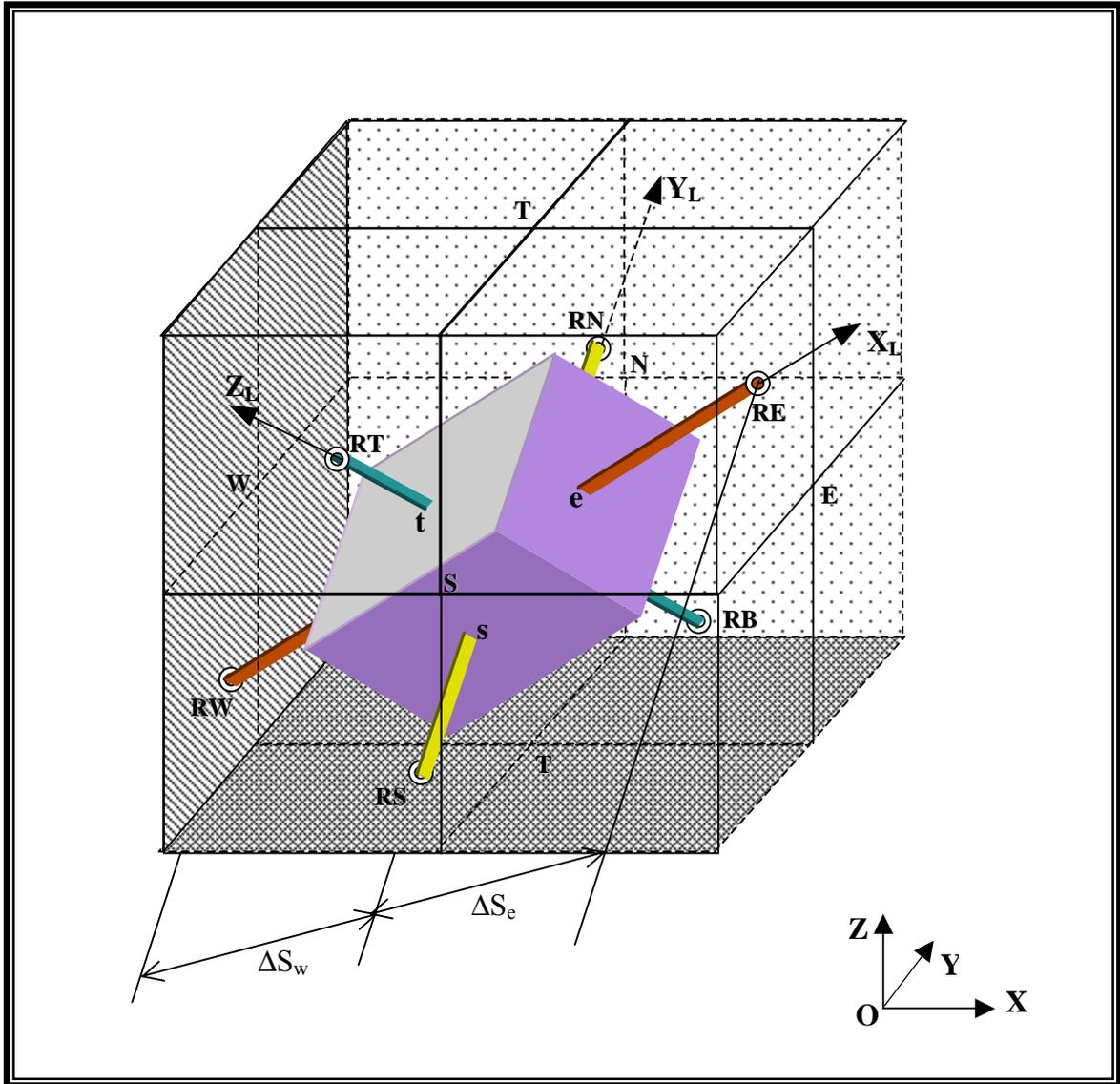


Figure 22 A 3-D Rotation Control Volume

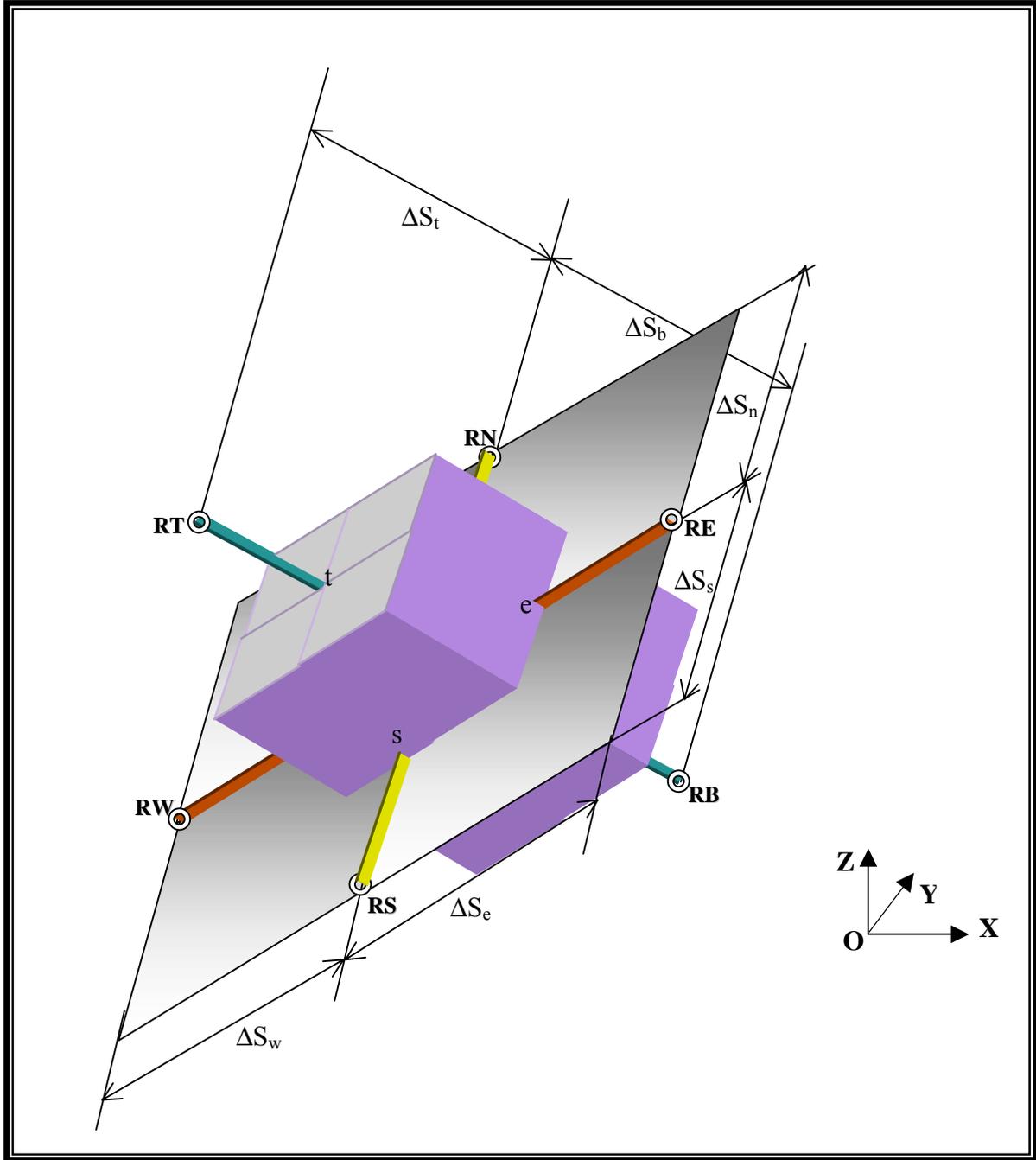


Figure 23 Geometric Notation in A 3-D Rotation Control Volume

**a) Approximation to Diffusion Term**

Referring to Figure 22 and Figure 23 and applying control volume technique in the local coordinate system, diffusion terms of Eq(115) may be written as,

$$\begin{aligned}
 Diff = & \frac{\Delta S_n + \Delta S_s}{2} \frac{\Delta S_t + \Delta S_b}{2} (J_e - J_w) \\
 & + \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_t + \Delta S_b}{2} (J_n - J_s) \\
 & + \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} (J_t - J_b)
 \end{aligned} \tag{129}$$

where  $J_e$ ,  $J_w$ ,  $J_n$ ,  $J_s$ ,  $J_t$ , and  $J_b$  are the fluxes through east, west, north, south, top and bottom cell-faces:

$$\left\{ \begin{aligned}
 J_e &= K_{xx}^{1e} \frac{h_{RE} - h_P}{\Delta S_e} \\
 J_w &= K_{xx}^{1w} \frac{h_P - h_{RW}}{\Delta S_w} \\
 J_n &= K_{yy}^{1n} \frac{h_{RN} - h_P}{\Delta S_n} \\
 J_s &= K_{yy}^{1s} \frac{h_P - h_{RS}}{\Delta S_s} \\
 J_t &= K_{zz}^{1t} \frac{h_{RT} - h_P}{\Delta S_t} \\
 J_b &= K_{zz}^{1b} \frac{h_P - h_{RB}}{\Delta S_b}
 \end{aligned} \right. \tag{130}$$

All symbols in Eq(129) and Eq(130) were denoted in Figure 22 and Figure 23.  $K_{ii}^{1i}$  are the principal components of hydraulic conductivity tensor which were given as input parameters by VB interface.

Note that non-nodal quantities in Eq(130) must be evaluated in terms of nodal values. Unlike in 2-D model, the interpolation scheme using to evaluate those non-nodal quantities must be bilinear or higher order scheme. Bilinear interpolation scheme described in Eq(58) has been used in IGW 3-D model.

In 3-D case, RE, RW, RN, RS, RT and RB are intersection locations of each principal direction line and the arbitrary polyhedron shaped by node P's 26 neighboring nodes, E, SE, S, SW, W, NW, N, NE, TE, STE, ST, SWT, TW, NWT, NT, NET, BE, SEB, SB, SWB, BW, NWB, NB, NEB, T and B (see Figure 20 and 21). They obviously vary with the given  $\theta$  and  $\beta$  so that those non-nodal quantities  $h_{RE}$ ,  $h_{RW}$ ,  $h_{RN}$ ,  $h_{RS}$ ,  $h_{RT}$  and  $h_{RB}$ , have no fixed nodal

points to be related (see Figure 22). In general, each of  $h_{RE}$ ,  $h_{RW}$ ,  $h_{RN}$ ,  $h_{RS}$ ,  $h_{RT}$  and  $h_{RB}$ , can be expressed as follows

$$\begin{cases} h_{RE} = \sum \alpha_i^E h_i \\ h_{RW} = \sum \alpha_i^W h_i \\ h_{RN} = \sum \alpha_i^N h_i \\ h_{RS} = \sum \alpha_i^S h_i \\ h_{RT} = \sum \alpha_i^T h_i \\ h_{RB} = \sum \alpha_i^B h_i \end{cases} \quad (131)$$

Where  $\alpha_i^j$  (j=E, W, N, S, T, B; i=E, SE, S, SW, W, NW, N, NE, TE, STE, ST, SWT, TW, NWT, NT, NET, BE, SEB, SB, SWB, BW, NWB, NB, NEB, T, B) are shape functions which also has the following properties:

$$\begin{cases} \sum_i \alpha_i = 1 \\ \alpha_i^j > 0 \end{cases} \quad (132)$$

The shape functions were stored in the corresponding variable named WPLT(27) in the source code .

Substituting Eq(131) and Eq(130) into Eq(129), the following equation is obtained

$$\begin{aligned} Diff = & a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_T h_T + a_B h_B \\ & + a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} \\ & + a_{NT} h_{NT} + a_{NB} h_{NB} + a_{ST} h_{ST} + a_{SB} h_{SB} \\ & + a_{TW} h_{TW} + a_{BW} h_{BW} + a_{TE} h_{TE} + a_{BE} h_{BE} \\ & + a_{NET} h_{NET} + a_{SET} h_{SET} + a_{SWT} h_{SWT} + a_{NWT} h_{NWT} \\ & + a_{NEB} h_{NEB} + a_{SEB} h_{SEB} + a_{SWB} h_{SWB} + a_{NWB} h_{NWB} \\ & - a_p h_p \end{aligned} \quad (133)$$

Where

$$\left\{ \begin{aligned}
a_{NE} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{'e}_{xx} \alpha_{NE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{'w}_{xx} \alpha_{NE}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{'n}_{yy} \alpha_{NE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{'s}_{yy} \alpha_{NE}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{'t}_{zz} \alpha_{NE}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{'b}_{zz} \alpha_{NE}^B \\
a_{NW} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{'e}_{xx} \alpha_{NW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{'w}_{xx} \alpha_{NW}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{'n}_{yy} \alpha_{NW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{'s}_{yy} \alpha_{NW}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{'t}_{zz} \alpha_{NW}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{'b}_{zz} \alpha_{NW}^B \\
a_{SE} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{'e}_{xx} \alpha_{SE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{'w}_{xx} \alpha_{SE}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{'n}_{yy} \alpha_{SE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{'s}_{yy} \alpha_{SE}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{'t}_{zz} \alpha_{SE}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{'b}_{zz} \alpha_{SE}^B \\
a_{SW} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{'e}_{xx} \alpha_{SW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{'w}_{xx} \alpha_{SW}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{'n}_{yy} \alpha_{SW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{'s}_{yy} \alpha_{SW}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{'t}_{zz} \alpha_{SW}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{'b}_{zz} \alpha_{SW}^B
\end{aligned} \right. \quad (134a)$$

$$\left\{ \begin{aligned}
a_{NT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{NT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{NT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{NT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{NT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{NT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{NT}^B \\
a_{NB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{NB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{NB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{NB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{NB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{NB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{NB}^B \\
a_{ST} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{ST}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{ST}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{ST}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{ST}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{ST}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{ST}^B \\
a_{SB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{SB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{SB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{SB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{SB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{SB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{SB}^B
\end{aligned} \right. \quad (134b)$$

$$\left\{ \begin{aligned}
a_{NET} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1e} \alpha_{NET}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1w} \alpha_{NET}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1n} \alpha_{NET}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1s} \alpha_{NET}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1t} \alpha_{NET}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1b} \alpha_{NET}^B \\
a_{NWT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1e} \alpha_{NWT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1w} \alpha_{NWT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1n} \alpha_{NWT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1s} \alpha_{NWT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1t} \alpha_{NWT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1b} \alpha_{NWT}^B \\
a_{SET} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1e} \alpha_{SET}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1w} \alpha_{SET}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1n} \alpha_{SET}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1s} \alpha_{SET}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1t} \alpha_{SET}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1b} \alpha_{SET}^B \\
a_{SWT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1e} \alpha_{SWT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K_{xx}^{1w} \alpha_{SWT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1n} \alpha_{SWT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K_{yy}^{1s} \alpha_{SWT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1t} \alpha_{SWT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K_{zz}^{1b} \alpha_{SWT}^B
\end{aligned} \right. \quad (134c)$$

$$\left\{ \begin{aligned}
a_{NEB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{NEB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{NEB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{NEB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{NEB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{NEB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{NEB}^B \\
a_{NWB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{NWB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{NWB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{NWB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{NWB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{NWB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{NWB}^B \\
a_{SEB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{SEB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{SEB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{SEB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{SEB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{SEB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{SEB}^B \\
a_{SWB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K^{1e}_{xx} \alpha_{SWB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K^{1w}_{xx} \alpha_{SWB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K^{1n}_{yy} \alpha_{SWB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K^{1s}_{yy} \alpha_{SWB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K^{1t}_{zz} \alpha_{SWB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K^{1b}_{zz} \alpha_{SWB}^B
\end{aligned} \right. \quad (134d)$$

$$\left\{ \begin{array}{l}
a_E = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} K'_{xx}{}^e \sum_j \alpha_E^j, \quad j = E, W, N, S, T, B \\
a_W = \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} K'_{xx}{}^w \sum_j \alpha_W^j, \quad j = E, W, N, S, T, B \\
a_N = \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} K'_{yy}{}^n \sum_j \alpha_N^j, \quad j = E, W, N, S, T, B \\
a_S = \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} K'_{yy}{}^s \sum_j \alpha_S^j, \quad j = E, W, N, S, T, B \\
a_T = \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} K'_{zz}{}^t \sum_j \alpha_T^j, \quad j = E, W, N, S, T, B \\
a_B = \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} K'_{zz}{}^b \sum_j \alpha_B^j, \quad j = E, W, N, S, T, B \\
a_P = a_E + a_W + a_N + a_S + a_T + a_B \\
\quad + a_{NE} + a_{NW} + a_{SE} + a_{SW} \\
\quad + a_{NT} + a_{NB} + a_{ST} + a_{SB} \\
\quad + a_{TW} + a_{BW} + a_{TE} + a_{BE} \\
\quad + a_{NET} + a_{SET} + a_{SWT} + a_{NWT} \\
\quad + a_{NEB} + a_{SEB} + a_{SWB} + a_{NWB}
\end{array} \right. \quad (134e)$$

Implementation of calculating coefficients was done in Subroutine NEWCOEFFLOW3D. The same derived type variable CST2 was used to store these coefficients.

From Eq(134), it is very obvious that every quantity in this equation is positive. Therefore, no negative coefficients happen again.

### b) Approximation to Time Derivative Term

For transient flow, a backward finite difference scheme is used to approximate the time-derivative term

$$S \frac{\partial h}{\partial t} = S \frac{h^{n+1} - h^n}{\Delta t} \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{\Delta S_t + \Delta S_b}{2} = a_P^t h^{n+1} - S_f^t \quad (135)$$

Where

$$\left\{ \begin{array}{l}
a_P^t = \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{\Delta S_t + \Delta S_b}{2} \frac{S}{\Delta t} \\
S_f^t = \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{\Delta S_t + \Delta S_b}{2} \frac{S}{\Delta t} h^n
\end{array} \right. \quad (136)$$

### c) Approximation to Source/Sink Term

Source/sink term discretizing is almost the same as that of traditional control volume except the control volume's volume. In general,  $Q$  can be expressed as

$$Q = a_p^Q h_p + S_f^Q \quad (137)$$

Table 12 is a list of source/sink available in IGW 3-D model and their corresponding  $a_p^Q$  and  $S_f^Q$ .

Table 12

Type of Source/Sink	$a_p^Q$	$S_f^Q$	Marks
Well	0	$Q_{well}$	
Recharge	0	$q \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$	
River	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river}$	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river} h_{river}$	$h > R_{bed}$
	0	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{river} (h_{river} - R_{bed})$	$h < R_{bed}$
Drain	$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} L_{drain}$	$L_{drain} \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} D_{bed}$	$h > D_{bed}$
General Head	$L_G \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$	$L_G \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} h_{source}$	
Evapo-transpiration	0	$q_{max}^{EVP} \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$	$h > h_s$
	$\frac{q_{max}^{EVP}}{d} \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$	$\frac{q_{max}^{EVP}}{d} (h_s - d) \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2}$	$(h_s - d) < h < h_s$
	0	0	$h < (h_s - d)$

All symbols have been explained above. Implementation of this process was done in Subroutine ADDQS13D and Subroutine ADDQS23D.

### d) Coefficient Matrix Assembling

From Eq(133), Eq(135) and Eq(137), gives a set of linear equations as follow

$$\begin{aligned}
(a_p + a_p^t + a_p^O)h_p &= a_E h_E + a_W h_W + a_N h_N + a_S h_S + a_T h_T + a_B h_B \\
&+ a_{NE} h_{NE} + a_{NW} h_{NW} + a_{SE} h_{SE} + a_{SW} h_{SW} \\
&+ a_{NT} h_{NT} + a_{NB} h_{NB} + a_{ST} h_{ST} + a_{SB} h_{SB} \\
&+ a_{TW} h_{TW} + a_{BW} h_{BW} + a_{TE} h_{TE} + a_{BE} h_{BE} \\
&+ a_{NET} h_{NET} + a_{SET} h_{SET} + a_{SWT} h_{SWT} + a_{NWT} h_{NWT} \\
&+ a_{NEB} h_{NEB} + a_{SEB} h_{SEB} + a_{SWB} h_{SWB} + a_{NWB} h_{NWB} \\
&+ S_f^O + S_f^t
\end{aligned} \tag{138}$$

It is seen that instead having of 19 diagonal entries in the derived-coefficient matrix by using traditional control volume technique, there are 27 diagonal entries in the derived-coefficient matrix when applying rotational control volume technique.

### e) Matrix Solver

Solution to Eq(138) can be obtained by using SOR technique or other efficient matrix solvers. SOR was adopted in IGW 3-D model. Its iterative equation can be expressed as follow

$$\begin{aligned}
h_p^{k+1} &= h_p^k + \frac{\alpha}{(a_p + a_p^t + a_p^O)} \{ a_E h_E^k + a_W h_W^{k+1} + a_N h_N^k + a_S h_S^{k+1} + a_T h_T^k + a_B h_B^{k+1} \\
&+ a_{NE} h_{NE}^k + a_{NW} h_{NW}^k + a_{SE} h_{SE}^{k+1} + a_{SW} h_{SW}^{k+1} \\
&+ a_{NT} h_{NT}^k + a_{NB} h_{NB}^k + a_{ST} h_{ST}^k + a_{SB} h_{SB}^{k+1} \\
&+ a_{TW} h_{TW}^k + a_{BW} h_{BW}^{k+1} + a_{TE} h_{TE}^k + a_{BE} h_{BE}^{k+1} \\
&+ a_{NET} h_{NET}^k + a_{SET} h_{SET}^k + a_{SWT} h_{SWT}^k + a_{NWT} h_{NWT}^k \\
&+ a_{NEB} h_{NEB}^k + a_{SEB} h_{SEB}^{k+1} + a_{SWB} h_{SWB}^{k+1} + a_{NWB} h_{NWB}^k \\
&+ S_f^O + S_f^t - (a_p + a_p^t + a_p^O)h_p^k \}
\end{aligned} \tag{139}$$

Where  $k$  is index of iteration number and  $\alpha$  is relaxation factor.

The final matrix assembling and iterative processes are carried out in Subroutine SORHEADT3D.

### 3.1.4 Special Treatments

- (1) To avoid doing derivation of every kind of finite difference equation on different boundary conditions, computational domain with dimension of 1 to NI, 1 to NJ and 1 to NK has been expanded to that with dimension of 0 to NI+1, 0 to NJ+1 and 0 to NK+1, and every parameter at these expanded nodes is assigned to be zero. This implies that the computational domain is confined in one no flux boundary cube for all cases. No flow boundary condition also can be taken care automatically by simply letting  $K_{ii}=0$ .
- (2) A boundary indicator variable IBOUND() is allocated to identify cell of first kind of boundary condition, IBOUND()=-1, cell of second kind of boundary condition or no flow boundary condition, IBOUND()=0 (inactive cell) and active cell, IBOUND()=1.
- (3) When anisotropy orientation angle  $\theta$  and  $\beta$  exist,  $\theta$  and  $\beta$  will be assumed to be zero at any nodes adjacent to any inactive cell. The reason why we are doing so has been described in 2-D model.
- (4) A non-linear iteration technique is applied to the case having head dependent source/sink. This is also part of 'inner loop'.
- (5) Water table iteration for unconfined aquifer is conducted in VB for the sake of visualization of each iteration step. A dry-wet trick is also used to deal with unconfined cases. This part was coded by using VB.
- (6) There may be the time when a numerical dry occurs in rotational control volume method, that is,  $a_p$  in Eq(123) or Eq(134) is equal to zero in a given active cell due to the arbitrary anisotropy angles. It happens when all intersection points locate at the inactive layers (cells). Figure 24a shows an example of this case.
- (7) Determination of the intersection points on top or bottom becomes a little bit complicated in our 3D model due to the fact that top or bottom is usually an arbitrary surface in most cases (see Figure 24b). The method to obtain intersection points on top or bottom in IGW 3D is that:
  - (i) divides the top or bottom into eight parts in which there are three nodes such that each part forms a plane including three points, for example, plane T-TE-STE in Figure 24b;
  - (ii) establishes each plane's equation using the three points coordinates and the straight line equation which represents one of the principle directions with the known anisotropy angles;
  - (iii) solves point of intersection of a line and a plane.

There are may be other efficient methods to do so. If any, just go to and modify the Subroutine SURF\_AND\_LINE.

- (8) As mentioned in (1), there is no need to derive FD equations for every kind of boundary condition after computational domain is expanded. However, it must bring

some requirement for the geometrical quantities at inactive cells for rotational control volume technique to avoid resulting in any unreasonable computational grid system. These requirements, at least, include that aquifer top and bottom elevations and initial head in an inactive cell which is most near to an active cell should be provided appropriately.

- (9) There may be the time when the vertical grid size  $\Delta Z$  is very small with comparison to the other two grid sizes ( $\Delta X$  and  $\Delta Y$ ), that is,  $\frac{\Delta Z}{\Delta X} \ll \varepsilon$  or  $\frac{\Delta Z}{\Delta Y} \ll \varepsilon$ . This case will lead to an abnormal derived matrix. It would be very hard or impossible to gain a convergent solution by using SOR technique. This issue is still leaving unsolved in current version 3-D model. Advance matrix solver should be used in such case.

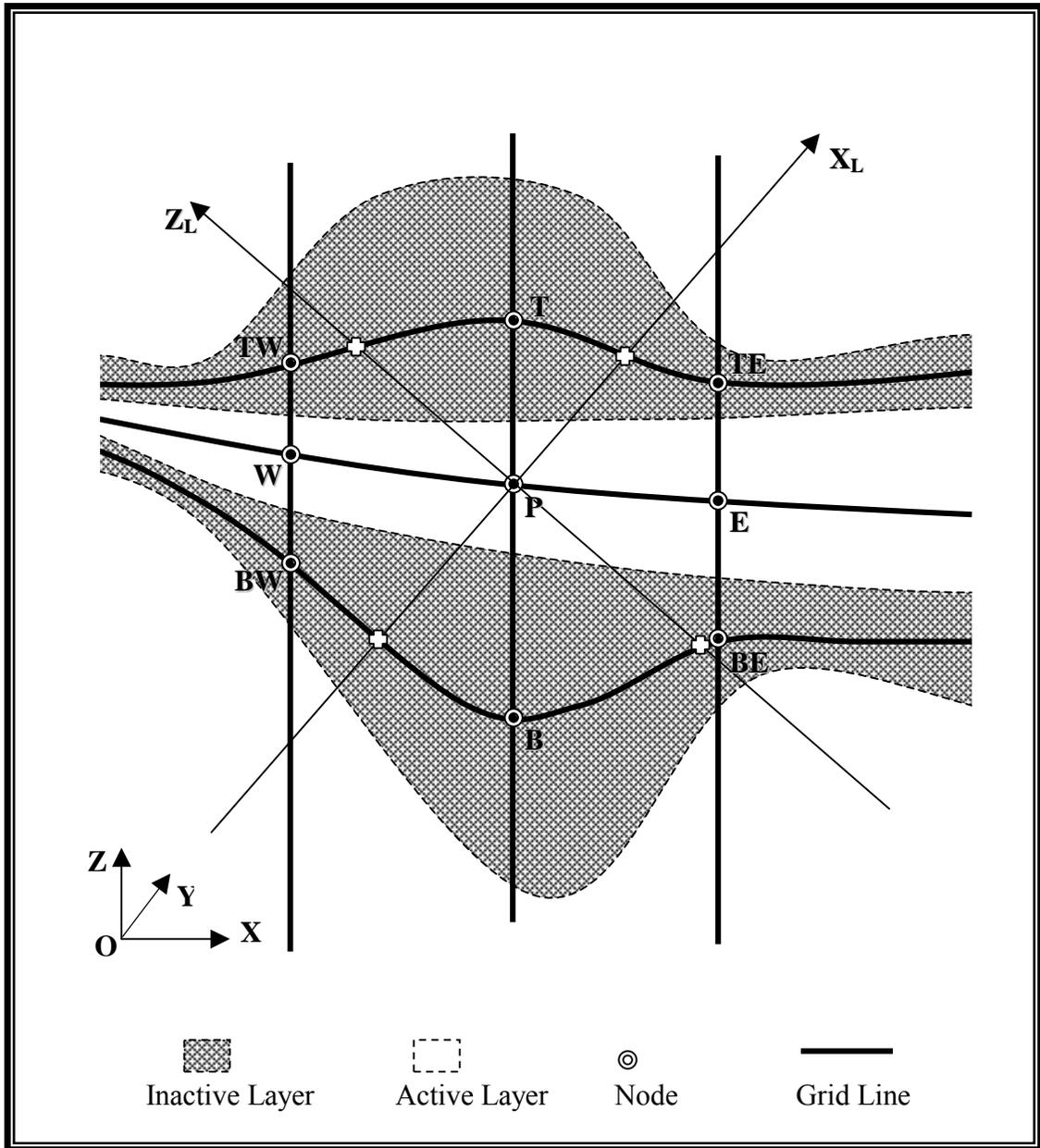
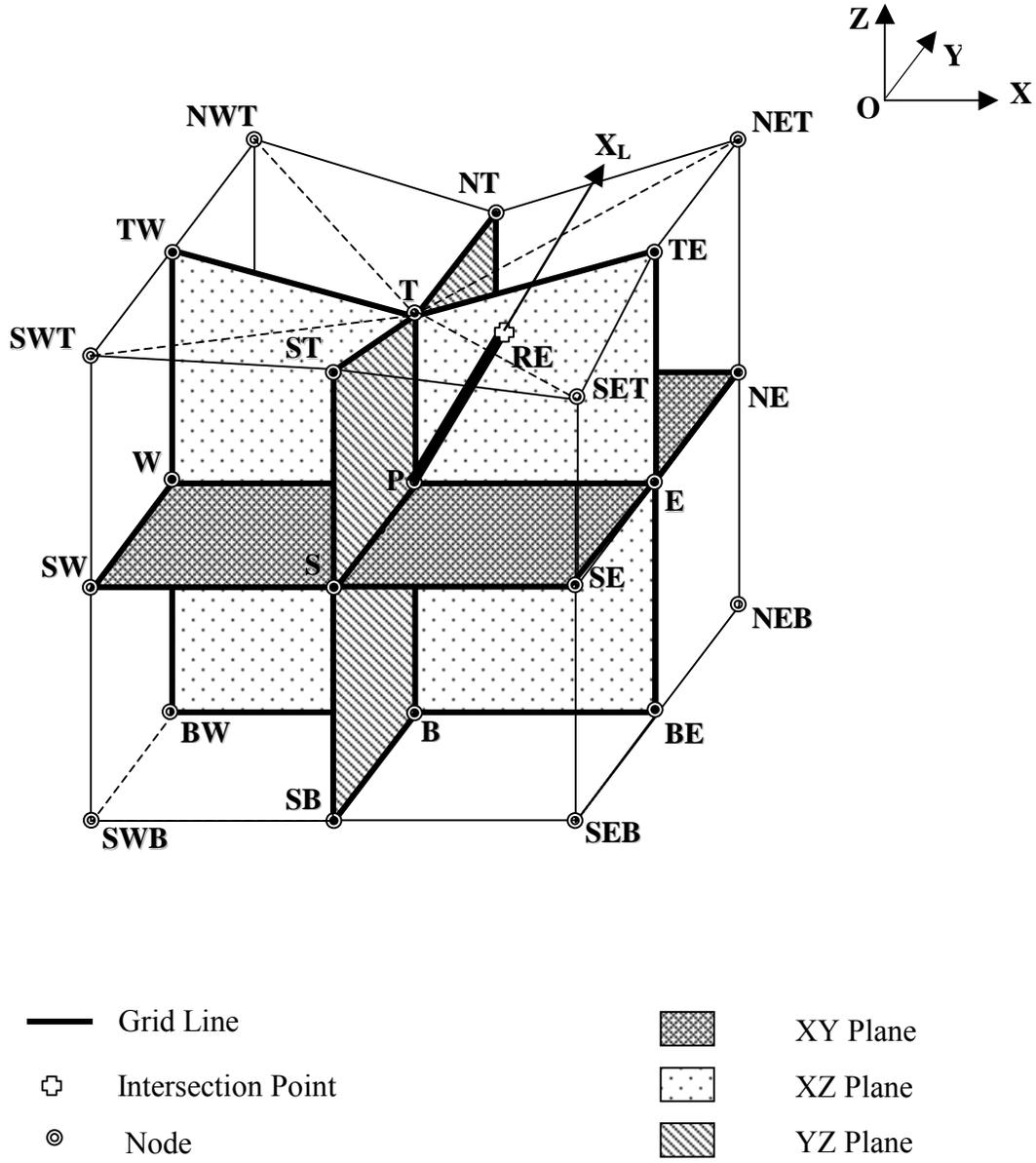


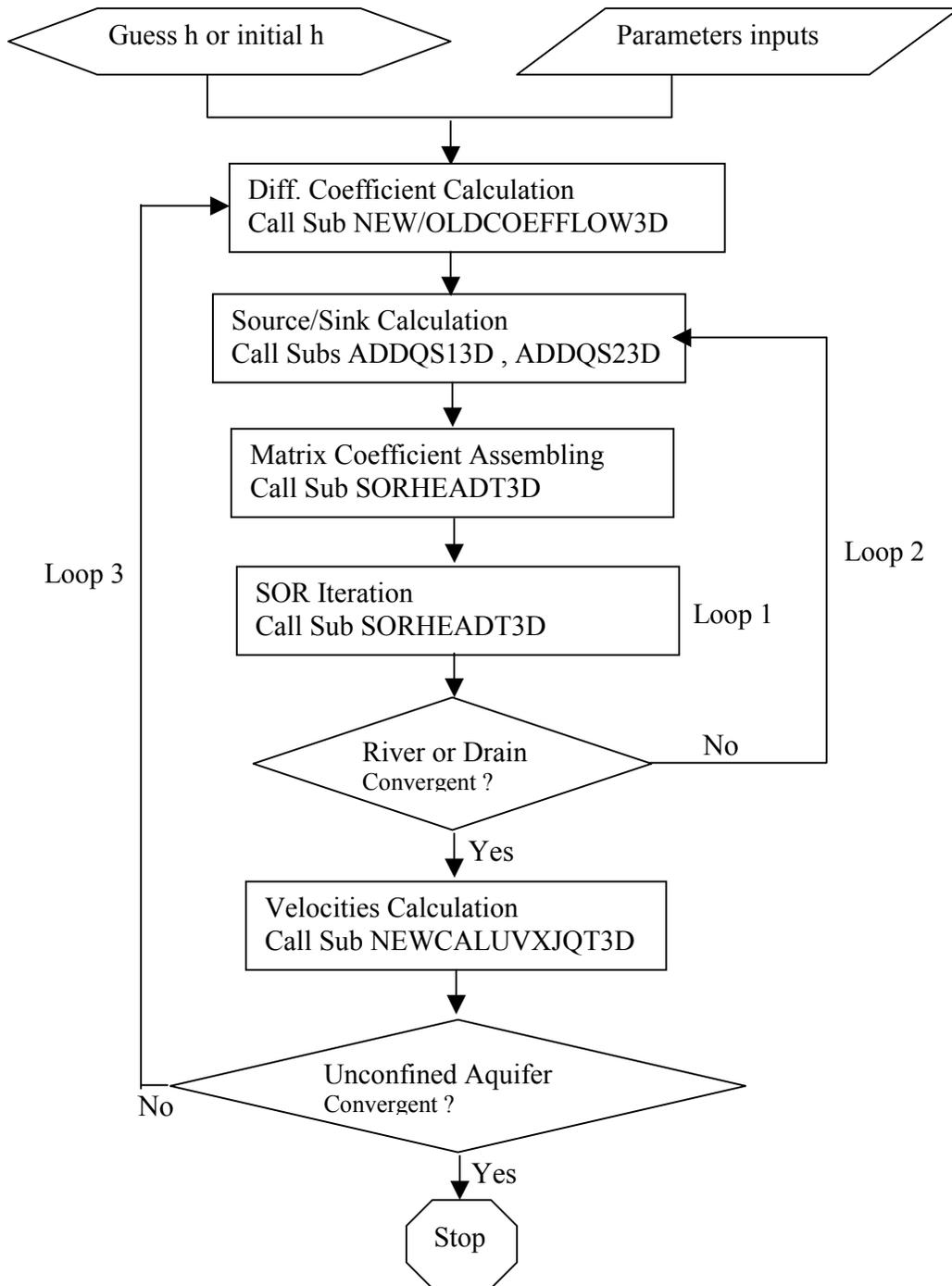
Figure 24a Illustration of Numerical Dry Cell



**Figure 24b Intersection Point between the First Principal Direction and Top Face in an Irregular 3-D Grid Layout**

### 3.1.5 Numerical Solution Procedure Flow Chart

The sequence of operations for IGW 3-D flow model is illustrated in the following flow diagram:



## 3.2 Transport

### 3.2.1 Governing Equation

The partial differential equation describing 3-D solute transport in porous medial is usually written as

$$\frac{\partial(nC)}{\partial t} + \frac{\partial(nu_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C}{\partial X_j}) - \rho_b \frac{\partial C^*}{\partial t} - \lambda nC + q_s C_s \quad (140)$$

Where

$C$  solute concentration

$n$  porosity of the porous medium

$u_i$  seepage or averaged pore velocity in the direction  $X_i$ ;

$D_{ij}$  dispersion coefficient tensor

$X_i$  Cartesian coordinate

$C^*$  concentration of species adsorbed on the solid

$\rho_b$  bulk density of the solid

$\lambda$  decay coefficient

$q_s$  volume flow rate per unit volume of the source or sink

$C_s$  solute concentration in the source or sink fluid

Considering equilibrium transport and assuming that the adsorption isotherm can be described with a linear and reversible equation, one can write

$$C^* = K_d C \quad (141)$$

Where  $K_d$  is called the distribution coefficient. Now, by incorporating Eq(141) into Eq(140), we obtain

$$\frac{\partial(nR_d C)}{\partial t} + \frac{\partial(nu_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C}{\partial X_j}) - \lambda nC + q_s C_s \quad (142)$$

or

$$\frac{\partial C}{\partial t} + \frac{u_i}{R_d} \frac{\partial C}{\partial X_i} = \frac{1}{nR_d} \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C}{\partial X_j}) - \frac{q_s + \lambda n}{nR_d} C + \frac{q_s C_s}{nR_d} \quad (143)$$

Where

$$R_d = 1 + \frac{\rho_b K_d}{n} \quad (144)$$

The parameter  $R_d$  is called the retardation factor.

The hydrodynamic dispersion tensor used in IGW 3-D transport model is defined in the following component forms:

$$\left\{ \begin{array}{l} D_{xx} = \alpha_L \frac{u_X^2}{U} + \alpha_{TH} \frac{u_Y^2}{U} + \alpha_{TV} \frac{u_Z^2}{U} + D^* \\ D_{yy} = \alpha_L \frac{u_Y^2}{U} + \alpha_{TH} \frac{u_X^2}{U} + \alpha_{TV} \frac{u_Z^2}{U} + D^* \\ D_{zz} = \alpha_L \frac{u_Z^2}{U} + \alpha_{TV} \frac{u_X^2}{U} + \alpha_{TV} \frac{u_Y^2}{U} + D^* \\ D_{xy} = D_{yx} = (\alpha_L - \alpha_{TH}) \frac{u_X u_Y}{U} \\ D_{xz} = D_{zx} = (\alpha_L - \alpha_{TV}) \frac{u_X u_Z}{U} \\ D_{yz} = D_{zy} = (\alpha_L - \alpha_{TV}) \frac{u_Y u_Z}{U} \end{array} \right. \quad (145)$$

where

$\alpha_L$  is the longitudinal dispersivity;

$\alpha_{TH}$  is the horizontal transverse dispersivity ;

$\alpha_{TV}$  is the vertical transverse dispersivity ;

$D^*$  is the effective molecular diffusion coefficient;

$u_X, u_Y, u_Z$  are components of the velocity vector along the  $X, Y$  and  $Z$  axes;

$U = \sqrt{u_X^2 + u_Y^2 + u_Z^2}$  is the magnitude of the velocity vector.

Strictly speaking, the dispersion tensor defined by two or three independent dispersivities for isotropic media as in equation (145) is not valid for anisotropic porous media, which require five independent dispersivities . Unfortunately, it is generally not feasible to obtain all five dispersivities in the field. As a result, the usual practice in transport modeling is to assume that the isotropic dispersion coefficient is also applicable to anisotropic porous media.

It is seen from Eq(145) that  $D_{xx}$  ,  $D_{yy}$  and  $D_{zz}$  are always positive, however,  $D_{xy}$  or  $D_{yx}$  ,  $D_{xz}$  or  $D_{zx}$  ,  $D_{yz}$  or  $D_{zy}$  may easily turn to be negative.

### 3.2.2 Grid Layout

This section is the same as FLOW model and may be skipped.

As described in 2-D models, in IGW, parameters are assigned to a block or a cell. Placing a representative node in the center of each cell forms the grid layout using in our spatial discretizing. Figure 16 shows a typical node-cell and its neighborhood of current use. The figure is self-explanatory, and those grid related geometrical quantities used in the scheme are illustrated in the related figure 17 to figure 19. The following table is a list of the variables using in the code and their counterparts in these figure,

Table 9

	Notation in Figure 17 to 19	Variable in Code
X coordinate	$X_{ijk}$	X(I,J,K)
Y coordinate	$Y_{ijk}$	Y(I,J,K)
Z coordinate	$Z_{ijk}$	Z(I,J,K)
X-Grid Spacing	$\Delta X_i$	HX(I)
Y-Grid Spacing	$\Delta Y_j$	HY(J)
Z-Grid Spacing	$\Delta Z_{ijk}$	Calculated from aquifer's thickness
X-CV area	$\Delta X_s$	DXS(I)
Y-CV area	$\Delta Y_s$	DYS(J)
Z-CV area	$\Delta Z_s$	Calculated from aquifer's thickness

It is noted that Visual Fortran code is written based on non-uniform grid spacing, although a uniform grid spacing has been used in IGW interface. The quantities at the cell-face, such as  $D_{ij}^e$ , must be evaluated in terms of nodal values before proceeding the calculation. There are many methods available to handle it, such as those of linear interpolation, harmonic averaged, etc. Harmonic mean was adopted in IGW 3-D model. X-velocity  $u_x$  is located on the East and West cell-faces, Y-velocity  $u_y$  is located on the North and South cell-faces and Z-velocity  $u_z$  is located on the Top and Bottom cell-faces. This arrangement will make use of cell-face  $K_{ij}$  which is the real one involving in the calculation, and central difference scheme to approximate the head gradient between two known nodal heads. To obtain the velocities at nodes, an interpolation method or simple arithmetic mean has to be used. Arithmetic mean has been used in IGW 3-D model. Subroutine NEWUVXJQT3D has the details. A linear interpolation should be used to evaluate the nodal  $u_z$  if the elevation of aquifer top or bottom is not uniform.

### 3.2.3 Scheme to Discretize Equation

There are three methods being introduced to IGW 3-D transport model: method of characteristics (MOC), modified method of characteristics (MMOC), full implicit finite difference method (FD). No random walk method is available in the current IGW3-D transport model. The difference among these three methods is only the way to approximate the advection term, the second term on the left-hand side of Eq(140), which describes the transport of miscible contaminants at the same velocity as the groundwater. Both MOC and MMOC invoke the Particle Tracking Technique to approximate the advection term. FD scheme is only applied at well cells in which the Particle Tracking is not available due to the fact that there is no unique characteristic curve at well node.

#### A) Full Implicit Finite Difference Scheme

##### a) Approximation of Advection Term

The target equation to be discretized by FD method is Eq(142). Applying the finite difference algorithm, the advection term can be approximated by the concentration values at the cell-faces as below

$$\begin{aligned} ADV &= \Delta X_s \Delta Y_s \Delta Z_s \frac{\partial(nu_i C)}{\partial X_i} = \Delta X_s \Delta Y_s \Delta Z_s \left[ \frac{\partial(nu_x C)}{\partial X} + \frac{\partial(nBu_y C)}{\partial Y} + \frac{\partial(nu_z C)}{\partial Z} \right] \\ &= \Delta Y_s \Delta Z_s (F_e - F_w) + \Delta X_s \Delta Z_s (F_n - F_s) + \Delta X_s \Delta Y_s (F_t - F_b) \end{aligned} \quad (146)$$

Where

$$\begin{cases} F_e = C_e q_e \\ F_w = C_w q_w \\ F_n = C_n q_n \\ F_s = C_s q_s \\ F_t = C_t q_t \\ F_b = C_b q_b \end{cases} \quad (147)$$

$q_e, q_w, q_n, q_s, q_t$  and  $q_b$  are fluxes through the six cell-faces.  $C_e, C_w, C_n, C_s, C_t$  and  $C_b$  are the concentration values at the six cell-faces.

Again, non-nodal quantities  $C_e, C_w, C_n, C_s, C_t$  and  $C_b$  must be evaluated in term of nodal values. How to determine the interface concentrations,  $C_e, C_w, C_n, C_s, C_t$  and  $C_b$  is what distinguishes one solution technique from another. The simple upwind scheme has been used in IGW 3-D transport model. For the upwind scheme, the cell-face concentration between two neighboring nodes in a particular direction ( $X_i$ ) is set equal to the concentration at the upstream node along the same direction

$$\left\{ \begin{array}{l}
 C_e = \begin{cases} C_E, & \text{if } q_e > 0 \\
 C_P, & \text{if } q_e < 0 \end{cases} \\
 C_w = \begin{cases} C_W, & \text{if } -q_w > 0 \\
 C_P, & \text{if } -q_w < 0 \end{cases} \\
 C_n = \begin{cases} C_N, & \text{if } q_n > 0 \\
 C_P, & \text{if } q_n < 0 \end{cases} \\
 C_s = \begin{cases} C_S, & \text{if } -q_s > 0 \\
 C_P, & \text{if } -q_s < 0 \end{cases} \\
 C_t = \begin{cases} C_T, & \text{if } q_t > 0 \\
 C_P, & \text{if } q_t < 0 \end{cases} \\
 C_b = \begin{cases} C_B, & \text{if } -q_b > 0 \\
 C_P, & \text{if } -q_b < 0 \end{cases}
 \end{array} \right. \quad (148)$$

Note that we have a convention for the flux direction: “+” for flux entering the cell, “-” for flux leaving the cell.

Substituting Eq(147) and (148) into Eq(146), gives

$$\begin{aligned}
 ADV &= a_E^{ADV} C_E + a_W^{ADV} C_W \\
 &\quad + a_N^{ADV} C_N + a_S^{ADV} C_S \\
 &\quad + a_T^{ADV} C_T + a_B^{ADV} C_B \\
 &\quad - a_P^{ADV} C_P
 \end{aligned} \quad (149)$$

Where

$$\left\{ \begin{array}{l}
 a_E^{ADV} = \Delta Y_s \Delta Z_s \max[q_e, 0] \\
 a_W^{ADV} = \Delta Y_s \Delta Z_s \max[-q_w, 0] \\
 a_N^{ADV} = \Delta X_s \Delta Z_s \max[q_n, 0] \\
 a_S^{ADV} = \Delta X_s \Delta Z_s \max[-q_s, 0] \\
 a_T^{ADV} = \Delta X_s \Delta Y_s \max[q_t, 0] \\
 a_B^{ADV} = \Delta X_s \Delta Y_s \max[-q_b, 0] \\
 a_P^{ADV} = \Delta Y_s \Delta Z_s (\max[-q_e, 0] + \max[q_w, 0]) \\
 \quad + \Delta X_s \Delta Z_s (\max[-q_n, 0] + \max[q_s, 0]) \\
 \quad + \Delta X_s \Delta Y_s (\max[-q_t, 0] + \max[q_b, 0])
 \end{array} \right. \quad (150)$$

$q_e, q_w, q_n, q_s, q_t$  and  $q_b$  are stored in array FLUX(Nwell, 6) which are calculated in Subroutine LHSWELL3D. Implementation of this process was done by Subroutine FDCOE3D.

## b) Traditional Control Volume Technique to Approximate Diffusion Term

Following the similar derivation procedure in flow model and applying control volume technique, from Figure 16 to Figure 19, diffusion terms of Eq(142) may be written as,

$$Diff = \Delta X_s \Delta Y_s \Delta Z_s \left( \frac{J_e - J_w}{\Delta X_s} + \frac{J_n - J_s}{\Delta Y_s} + \frac{J_t - J_b}{\Delta Z_s} \right) \quad (151)$$

Where  $J_e, J_w, J_n, J_s, J_t$  and  $J_b$  are the fluxes through east, west, north, south, top and bottom cell-faces respectively. They have the following forms:

$$\left\{ \begin{array}{l} J_e = nD_{xx}^e \frac{C_E - C_P}{\Delta X} + nD_{xy}^e \frac{C_{ne} - C_{se}}{\Delta Y_s} + nD_{xz}^e \frac{C_{te} - C_{be}}{\Delta Z_s} \\ J_w = nD_{xx}^w \frac{C_P - C_W}{\Delta X} + nD_{xy}^w \frac{C_{nw} - C_{sw}}{\Delta Y_s} + nD_{yz}^w \frac{C_{tw} - C_{bw}}{\Delta Z_s} \\ J_n = nD_{yy}^n \frac{C_N - C_P}{\Delta Y} + nD_{yx}^n \frac{C_{ne} - C_{nw}}{\Delta X_s} + nD_{yz}^n \frac{C_{nt} - C_{nb}}{\Delta Z_s} \\ J_s = nD_{yy}^s \frac{C_P - C_S}{\Delta Y} + nD_{yx}^s \frac{C_{se} - C_{sw}}{\Delta X_s} + nD_{yz}^s \frac{C_{st} - C_{sb}}{\Delta Z_s} \\ J_t = nD_{zz}^t \frac{C_T - C_P}{\Delta Z} + nD_{zx}^t \frac{C_{te} - C_{tw}}{\Delta X_s} + nD_{zy}^t \frac{C_{nt} - C_{st}}{\Delta Y_s} \\ J_b = nD_{zz}^b \frac{C_P - C_B}{\Delta Z} + nD_{zx}^b \frac{C_{be} - C_{bw}}{\Delta X_s} + nD_{zy}^b \frac{C_{nb} - C_{sb}}{\Delta Y_s} \end{array} \right. \quad (152)$$

All symbols in Eq(152) were denoted in Figure 16 to Figure 19.

Note that non-nodal quantities in Eq(152) must be evaluated in terms of nodal values. The same simple four points averaged scheme as used in 2-D model has been used in IGW 3-D model. After re-arranging Eq(152), Eq(151) becomes

$$\begin{aligned} Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S + a_T^{Diff} C_T + a_B^{Diff} C_B \\ & + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} \\ & + a_{NT} C_{NT} + a_{NB} C_{NB} + a_{ST} C_{ST} + a_{SB} C_{SB} \\ & + a_{TW} C_{TW} + a_{BW} C_{BW} + a_{TE} C_{TE} + a_{BE} C_{BE} \\ & - a_P^{Diff} C_P \end{aligned} \quad (153)$$

Where

$$\left\{ \begin{aligned}
 a_E^{Diff} &= \frac{\Delta Y_s \Delta Z_s n D_{xx}^e}{\Delta X} + \frac{D_{yx}^n - D_{yx}^s}{4} n \Delta Z_s + \frac{D_{zx}^t - D_{zx}^b}{4} n \Delta Y_s \\
 a_W^{Diff} &= \frac{\Delta Y_s \Delta Z_s n D_{xx}^w}{\Delta X} - \frac{D_{yx}^n - D_{yx}^s}{4} n \Delta Z_s - \frac{D_{zx}^t - D_{zx}^b}{4} n \Delta Y_s \\
 a_N^{Diff} &= \frac{\Delta X_s \Delta Z_s n D_{yy}^n}{\Delta Y} + \frac{D_{xy}^e - D_{xy}^w}{4} n \Delta Z_s + \frac{D_{zy}^t - D_{zy}^b}{4} n \Delta X_s \\
 a_S^{Diff} &= \frac{\Delta X_s \Delta Z_s n D_{yy}^s}{\Delta Y} - \frac{D_{xy}^e - D_{xy}^w}{4} n \Delta Z_s - \frac{D_{zy}^t - D_{zy}^b}{4} n \Delta X_s \\
 a_T^{Diff} &= \frac{\Delta X_s \Delta Y_s n D_{zz}^t}{\Delta Z} + \frac{D_{xz}^e - D_{xz}^w}{4} n \Delta Y_s + \frac{D_{yz}^n - D_{yz}^s}{4} n \Delta X_s \\
 a_B^{Diff} &= \frac{\Delta X_s \Delta Y_s n D_{zz}^b}{\Delta Z} - \frac{D_{xz}^e - D_{xz}^w}{4} n \Delta Y_s - \frac{D_{yz}^n - D_{yz}^s}{4} n \Delta X_s \\
 a_{NE} &= \frac{D_{xy}^e + D_{yx}^n}{4} n \Delta Z_s, & a_{NW} &= -\frac{D_{xy}^w + D_{yx}^n}{4} n \Delta Z_s \\
 a_{SE} &= -\frac{D_{xy}^e + D_{yx}^s}{4} n \Delta Z_s, & a_{SW} &= \frac{D_{xy}^w + D_{yx}^s}{4} n \Delta Z_s \\
 a_{TE} &= \frac{D_{xz}^e + D_{zx}^t}{4} n \Delta Y_s, & a_{BE} &= -\frac{D_{xz}^e + D_{zx}^b}{4} n \Delta Y_s \\
 a_{TW} &= -\frac{D_{xz}^w + D_{zx}^t}{4} n \Delta Y_s, & a_{BW} &= \frac{D_{xz}^w + D_{zx}^b}{4} n \Delta Y_s \\
 a_{NT} &= \frac{D_{yz}^n + D_{zy}^t}{4} n \Delta X_s, & a_{NB} &= -\frac{D_{yz}^n + D_{zy}^b}{4} n \Delta X_s \\
 a_{ST} &= -\frac{D_{yz}^s + D_{zy}^t}{4} n \Delta X_s, & a_{SB} &= \frac{D_{yz}^s + D_{zy}^b}{4} n \Delta X_s \\
 a_P^{Diff} &= a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_T^{Diff} + a_B^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW} \\
 &\quad + a_{TE} + a_{BE} + a_{TW} + a_{BW} + a_{NT} + a_{NB} + a_{ST} + a_{SB}
 \end{aligned} \right. \tag{154}$$

This process was implemented in Subroutine OLDCOEFTRSP3D. A derived type variable CST1 was used to store these coefficients:

Table 2

Notation in Eq(154)	Variable in Code
$a_E^{Diff}$	CST2 (I,J,K)%SE
$a_W^{Diff}$	CST2 (I,J,K)%SW
$a_N^{Diff}$	CST2 (I,J,K)%SN

$a_S^{Diff}$	CST2 (I,J,K)%SS
$a_T^{Diff}$	CST2 (I,J,K)%ST
$a_B^{Diff}$	CST2 (I,J,K)%SB
$a_{NE}$	CST2 (I,J,K)%SNE
$a_{NW}$	CST2 (I,J,K)%SNW
$a_{SE}$	CST2 (I,J,K)%SSE
$a_{SW}$	CST2 (I,J,K)%SSW
$a_{TE}$	CST2 (I,J,K)%STE
$a_{BE}$	CST2 (I,J,K)%SBE
$a_{TW}$	CST2 (I,J,K)%STW
$a_{BW}$	CST2 (I,J,K)%SBW
$a_{NT}$	CST2 (I,J,K)%SNT
$a_{NB}$	CST2 (I,J,K)%SNB
$a_{ST}$	CST2 (I,J,K)%SST
$a_{SB}$	CST2 (I,J,K)%SSB
$a_P^{Diff}$	CST2 (I,J,K)%SP

From Eq(154), it is very obvious that  $a_{NE}$ ,  $a_{NW}$ ,  $a_{SE}$ ,  $a_{SW}$ ,  $a_{TE}$ ,  $a_{BE}$ ,  $a_{TW}$ ,  $a_{BW}$ ,  $a_{NT}$ ,  $a_{NB}$ ,  $a_{ST}$ , and  $a_{SB}$ , may easily turn into negative ones, for example,  $a_{NW} < 0$  and  $a_{SE} < 0$  when  $D_{xy} > 0$ ; or  $a_{NE} < 0$  and  $a_{SW} < 0$  when  $D_{xy} < 0$ .

### c) Rotational Control Volume Technique to Approximate Diffusion Term

Following the similar derivation procedure in flow model and applying control volume technique in local coordinate system, from figure 22 and Figure 23, diffusion terms of Eq(142) may be written as,

$$\begin{aligned}
 Diff = & \frac{\Delta S_n + \Delta S_s}{2} \frac{\Delta S_t + \Delta S_b}{2} (J_e - J_w) \\
 & + \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_t + \Delta S_b}{2} (J_n - J_s) \\
 & + \frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} (J_t - J_b)
 \end{aligned} \tag{155}$$

where  $J_e$ ,  $J_w$ ,  $J_n$ ,  $J_s$ ,  $J_t$ , and  $J_b$  the solute fluxes through east, west, north, south, top and bottom cell-faces. They can be expressed as :

$$\left\{ \begin{array}{l} J_e = nD'_{xx} \frac{C_{RE} - C_P}{\Delta S_e} \\ J_w = nD'_{xx} \frac{C_P - C_{RW}}{\Delta S_w} \\ J_n = nD'_{yy} \frac{C_{RN} - C_P}{\Delta S_n} \\ J_s = nD'_{yy} \frac{C_P - C_{RS}}{\Delta S_s} \\ J_t = nD'_{zz} \frac{C_{RT} - C_P}{\Delta S_t} \\ J_b = nD'_{zz} \frac{C_P - C_{RB}}{\Delta S_b} \end{array} \right. \quad (156)$$

All symbols in Eq(155) and Eq(156) were denoted in Figure 22 and Figure 23.  $D'_{ii}$  are the principal components of hydrodynamic dispersion tensor which can be easily given by the following forms

$$\left\{ \begin{array}{l} D'_{xx} = \alpha_L \sqrt{u_x^2 + u_y^2 + u_z^2} \\ D'_{yy} = \alpha_{TH} \sqrt{u_x^2 + u_y^2 + u_z^2} \\ D'_{zz} = \alpha_{TV} \sqrt{u_x^2 + u_y^2 + u_z^2} \end{array} \right. \quad (157)$$

Note that non-nodal quantities in Eq(156) must be evaluated in terms of nodal values. Adopting the same linear interpolation scheme described in flow model,  $C_{RE}$ ,  $C_{RW}$ ,  $C_{RN}$ ,  $C_{RS}$ ,  $C_{RT}$  and  $C_{RB}$  in Eq(156) can be expressed as follows

$$\left\{ \begin{array}{l} C_{RE} = \sum \alpha_i^E h_i \\ C_{RW} = \sum \alpha_i^W h_i \\ C_{RN} = \sum \alpha_i^N h_i \\ C_{RS} = \sum \alpha_i^S h_i \\ C_{RT} = \sum \alpha_i^T h_i \\ C_{RB} = \sum \alpha_i^B h_i \end{array} \right. \quad (158)$$

Substituting Eq(156) and Eq(158) into Eq(155), the following equation is obtained

$$\begin{aligned}
Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S + a_T^{Diff} C_T + a_B^{Diff} C_B \\
& + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} \\
& + a_{NT} C_{NT} + a_{NB} C_{NB} + a_{ST} C_{ST} + a_{SB} C_{SB} \\
& + a_{TW} C_{TW} + a_{BW} C_{BW} + a_{TE} C_{TE} + a_{BE} C_{BE} \\
& + a_{NET} C_{NET} + a_{SET} C_{SET} + a_{SWT} C_{SWT} + a_{NWT} C_{NWT} \\
& + a_{NEB} C_{NEB} + a_{SEB} C_{SEB} + a_{SWB} C_{SWB} + a_{NWB} C_{NWB} \\
& - a_P^{Diff} C_P
\end{aligned} \tag{159}$$

Where

$$\left\{ \begin{aligned}
a_{NE} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{e'} \alpha_{NE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{w'} \alpha_{NE}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{n'} \alpha_{NE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{s'} \alpha_{NE}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{t'} \alpha_{NE}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{b'} \alpha_{NE}^B \\
a_{NW} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{e'} \alpha_{NW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{w'} \alpha_{NW}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{n'} \alpha_{NW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{s'} \alpha_{NW}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{t'} \alpha_{NW}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{b'} \alpha_{NW}^B \\
a_{SE} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{e'} \alpha_{SE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{w'} \alpha_{SE}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{n'} \alpha_{SE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{s'} \alpha_{SE}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{t'} \alpha_{SE}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{b'} \alpha_{SE}^B \\
a_{SW} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{e'} \alpha_{SW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{w'} \alpha_{SW}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{n'} \alpha_{SW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{s'} \alpha_{SW}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{t'} \alpha_{SW}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{b'} \alpha_{SW}^B
\end{aligned} \right. \tag{160a}$$

$$\left\{ \begin{aligned}
a_{NT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NT}^B \\
a_{NB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NB}^B \\
a_{ST} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{ST}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{ST}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{ST}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{ST}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{ST}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{ST}^B \\
a_{SB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{SB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{SB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{SB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{SB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{SB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{SB}^B
\end{aligned} \right. \quad (160b)$$

$$\left\{ \begin{aligned}
a_{NET} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NET}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NET}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NET}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NET}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NET}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NET}^B \\
a_{NWT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NWT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{NWT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NWT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{NWT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NWT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{NWT}^B \\
a_{SET} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{SET}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{SET}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{SET}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{SET}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{SET}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{SET}^B \\
a_{SWT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{SWT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD'_{xx} \alpha_{SWT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{SWT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD'_{yy} \alpha_{SWT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{SWT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD'_{zz} \alpha_{SWT}^B
\end{aligned} \right. \quad (160c)$$

$$\left\{ \begin{aligned}
a_{NEB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{ie} \alpha_{NEB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{iw} \alpha_{NEB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{in} \alpha_{NEB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{is} \alpha_{NEB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{it} \alpha_{NEB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{ib} \alpha_{NEB}^B \\
a_{NWB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{ie} \alpha_{NWB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{iw} \alpha_{NWB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{in} \alpha_{NWB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{is} \alpha_{NWB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{it} \alpha_{NWB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{ib} \alpha_{NWB}^B \\
a_{SEB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{ie} \alpha_{SEB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{iw} \alpha_{SEB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{in} \alpha_{SEB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{is} \alpha_{SEB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{it} \alpha_{SEB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{ib} \alpha_{SEB}^B \\
a_{SWB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{ie} \alpha_{SWB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{iw} \alpha_{SWB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{in} \alpha_{SWB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{is} \alpha_{SWB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{it} \alpha_{SWB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{ib} \alpha_{SWB}^B
\end{aligned} \right. \quad (160d)$$

$$\left\{ \begin{array}{l}
a_E^{Diff} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{te} \sum_j \alpha_E^j, \quad j = E, W, N, S, T, B \\
a_W^{Diff} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2} nD_{xx}^{tw} \sum_j \alpha_W^j, \quad j = E, W, N, S, T, B \\
a_N^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{tn} \sum_j \alpha_N^j, \quad j = E, W, N, S, T, B \\
a_S^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2} nD_{yy}^{ts} \sum_j \alpha_S^j, \quad j = E, W, N, S, T, B \\
a_T^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{tt} \sum_j \alpha_T^j, \quad j = E, W, N, S, T, B \\
a_B^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2} nD_{zz}^{tb} \sum_j \alpha_B^j, \quad j = E, W, N, S, T, B \\
a_P^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_T^{Diff} + a_B^{Diff} \\
\quad + a_{NE} + a_{NW} + a_{SE} + a_{SW} \\
\quad + a_{NT} + a_{NB} + a_{ST} + a_{SB} \\
\quad + a_{TW} + a_{BW} + a_{TE} + a_{BE} \\
\quad + a_{NET} + a_{SET} + a_{SWT} + a_{NWT} \\
\quad + a_{NEB} + a_{SEB} + a_{SWB} + a_{NWB}
\end{array} \right. \quad (160e)$$

Implementation of this process was done in Subroutine NEWCOEFTRSP3D. The same derived type variable CST1 was used to store these coefficients.

From Eq(160), it is very obvious that every coefficient in these equations is positive. Therefore, no negative coefficients would be appear in the discretized matrix again.

#### d) Approximation of Time-Derivative Term

A backward finite difference scheme is used to approximate the time-derivative term

$$nR_d \Delta X_s \Delta Y_s \Delta Z_s \frac{\partial C}{\partial t} = nR_d \Delta X_s \Delta Y_s \Delta Z_s \frac{C_P^{n+1} - C_P^n}{\Delta t} = a_P^t C_P^{n+1} - S_f^t \quad (161)$$

Where  $C^{n+1}$  is the concentration at new time level  $n+1$ ,  $C^n$  is the concentration at old time level  $n$  and

$$\left\{ \begin{array}{l}
a_P^t = \Delta X_s \Delta Y_s \Delta Z_s \frac{R_d}{\Delta t} \\
S_f^t = \Delta X_s \Delta Y_s \Delta Z_s \frac{R_d}{\Delta t} C_P^n
\end{array} \right. \quad (162)$$

For rotational control volume technique,  $\Delta X_s \Delta Y_s \Delta Z_s$  will be replaced with

$$\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{\Delta S_t + \Delta S_b}{2} \text{ in Eq(161) and Eq(162).}$$

### e) Approximation of Source/Sink Term

Source/sink term of the governing equation,  $q_s C_s$ , represents solute mass entering the model domain through source ( $q_s > 0$ ) or leaving the model domain through sink ( $q_s < 0$ ). For sources, it is necessary to specify the concentration of source water. For sinks, the concentration of sink water is generally equal to the concentration of groundwater in the aquifer at the sink location and can not be specified. Evapotranspiration may be assumed to take only pure water away from the aquifer so that the concentration of evapotranspiration flux is zero. In general,  $q_s C_s$  can be expressed as

$$q_s C_s = a_p^O C_p + S_f^O \quad (163)$$

Where

$$\left\{ \begin{array}{l} a_p^O = \text{Max}[-q_s^{Well}, 0] + \text{Max}[-q_s^{Recharge}, 0] \\ \quad + \text{Max}[-q_s^{River}, 0] + \text{Max}[-q_s^{Ghead}, 0] \\ \quad + \text{Max}[-q_s^{Drain}, 0] \\ S_f^O = \text{Max}[q_s^{Well}, 0] C_s^{Well} + \text{Max}[q_s^{Recharge}, 0] C_s^{Recharge} \\ \quad + \text{Max}[q_s^{River}, 0] C_s^{River} + \text{Max}[q_s^{Ghead}, 0] C_s^{Ghead} \end{array} \right. \quad (164)$$

$q_s^{Well}$ ,  $q_s^{Recharge}$ ,  $q_s^{River}$ ,  $q_s^{Drain}$  and  $q_s^{Ghead}$  are fluxes contributed from source/sink of wells, recharge, river, drain and general head which are stored in derived types (one dimension) Wells%Q, RECHS%Q, Rivers%Q, Drain%Q and HDEPENDS %Q respectively in source code. Calculation of every kind of  $q_s$  is performed by Subroutine QTOTAL3D.

### f) Approximation of Decay Term

From Eq(142), decay term,  $-\lambda n C$ , can be easily approximated as

$$\text{Decay} = a_p^D C_p \quad (165)$$

Where  $a_p^D = \lambda n$

Note that  $\lambda$  is assumed to be evaluated at node.

### g) Coefficient Matrix Assembling and Solution Technique

From Eq(149), Eq(153) or Eq(159), Eq(161), Eq(163) and Eq(165), gives a set of linear equations as follows

Traditional control volume technique:

$$\begin{aligned}
 (a_p^{ADV} + a_p^{Diff} + a_p^t + a_p^Q + a_p^D)C_p &= (a_E^{ADV} + a_E^{Diff})C_E + (a_W^{ADV} + a_W^{Diff})C_W \\
 &+ (a_N^{ADV} + a_N^{Diff})C_N + (a_S^{ADV} + a_S^{Diff})C_S \\
 &+ (a_T^{ADV} + a_T^{Diff})C_T + (a_B^{ADV} + a_B^{Diff})C_B \\
 &+ a_{NE}C_{NE} + a_{NW}C_{NW} + a_{SE}C_{SE} + a_{SW}C_{SW} \\
 &+ a_{NT}C_{NT} + a_{NB}C_{NB} + a_{ST}C_{ST} + a_{SB}C_{SB} \\
 &+ a_{TW}C_{TW} + a_{BW}C_{BW} + a_{TE}C_{TE} + a_{BE}C_{BE} \\
 &+ S_f^Q + S_f^t
 \end{aligned} \tag{166a}$$

Rotational control volume technique:

$$\begin{aligned}
 (a_p^{ADV} + a_p^{Diff} + a_p^t + a_p^Q + a_p^D)C_p &= (a_E^{ADV} + a_E^{Diff})C_E + (a_W^{ADV} + a_W^{Diff})C_W \\
 &+ (a_N^{ADV} + a_N^{Diff})C_N + (a_S^{ADV} + a_S^{Diff})C_S \\
 &+ (a_T^{ADV} + a_T^{Diff})C_T + (a_B^{ADV} + a_B^{Diff})C_B \\
 &+ a_{NE}C_{NE} + a_{NW}C_{NW} + a_{SE}C_{SE} + a_{SW}C_{SW} \\
 &+ a_{NT}C_{NT} + a_{NB}C_{NB} + a_{ST}C_{ST} + a_{SB}C_{SB} \\
 &+ a_{TW}C_{TW} + a_{BW}C_{BW} + a_{TE}C_{TE} + a_{BE}C_{BE} \\
 &+ a_{NET}C_{NET} + a_{SET}C_{SET} + a_{SWT}C_{SWT} + a_{NWT}C_{NWT} \\
 &+ a_{NEB}C_{NEB} + a_{SEB}C_{SEB} + a_{SWB}C_{SWB} + a_{NWB}C_{NWB} \\
 &+ S_f^Q + S_f^t
 \end{aligned} \tag{166b}$$

SOR technique was adopted in IGW 3-D model to solve Eq(166). Its iterative equation can be expressed as follow

Traditional control volume:

$$\begin{aligned}
 C_P^{k+1} = C_P^k + \frac{\alpha}{(a_P + a_P^t + a_P^Q)} & \{a_E C_E^k + a_W C_W^{k+1} + a_N C_N^k + a_S C_S^{k+1} + a_T C_T^k + a_B C_B^{k+1} \\
 & + a_{NE} C_{NE}^k + a_{NW} C_{NW}^k + a_{SE} C_{SE}^{k+1} + a_{SW} C_{SW}^{k+1} \\
 & + a_{NT} C_{NT}^k + a_{NB} C_{NB}^k + a_{ST} C_{ST}^k + a_{SB} C_{SB}^{k+1} \\
 & + a_{TW} C_{TW}^k + a_{BW} C_{BW}^{k+1} + a_{TE} C_{TE}^k + a_{BE} C_{BE}^{k+1} \\
 & + S_f^Q + S_f^t - (a_P + a_P^t + a_P^Q) C_P^k \} \quad (167a)
 \end{aligned}$$

Rotational control volume:

$$\begin{aligned}
 C_P^{k+1} = C_P^k + \frac{\alpha}{(a_P + a_P^t + a_P^Q)} & \{a_E C_E^k + a_W C_W^{k+1} + a_N C_N^k + a_S C_S^{k+1} + a_T C_T^k + a_B C_B^{k+1} \\
 & + a_{NE} C_{NE}^k + a_{NW} C_{NW}^k + a_{SE} C_{SE}^{k+1} + a_{SW} C_{SW}^{k+1} \\
 & + a_{NT} C_{NT}^k + a_{NB} C_{NB}^k + a_{ST} C_{ST}^k + a_{SB} C_{SB}^{k+1} \\
 & + a_{TW} C_{TW}^k + a_{BW} C_{BW}^{k+1} + a_{TE} C_{TE}^k + a_{BE} C_{BE}^{k+1} \\
 & + a_{NET} C_{NET}^k + a_{SET} C_{SET}^k + a_{SWT} C_{SWT}^k + a_{NWT} C_{NWT}^k \\
 & + a_{NEB} C_{NEB}^k + a_{SEB} C_{SEB}^{k+1} + a_{SWB} C_{SWB}^{k+1} + a_{NWB} C_{NWB}^k \\
 & + S_f^Q + S_f^t - (a_P + a_P^t + a_P^Q) C_P^k \} \quad (167b)
 \end{aligned}$$

Where

$$\begin{aligned}
 a_P &= a_P^{ADV} + a_P^{Diff} + a_P^t + a_P^Q + a_P^D \\
 a_E &= a_E^{ADV} + a_E^{Diff} \\
 a_W &= a_W^{ADV} + a_W^{Diff} \\
 a_N &= a_N^{ADV} + a_N^{Diff} \\
 a_S &= a_S^{ADV} + a_S^{Diff} \\
 a_T &= a_T^{ADV} + a_T^{Diff} \\
 a_B &= a_B^{ADV} + a_B^{Diff}
 \end{aligned}$$

$k$  index of iteration number

$\alpha$  relaxation factor.

The final matrix assembling process is carried out in Subroutine SORCBAR3D.

## B) Mixed Eulerian-Lagrangian Methods

Eq(143) is an Eulerian expression in which the partial derivative,  $\frac{\partial C}{\partial t}$ , represents the rate of change in solute concentration at a fixed point in space. It can be expressed in the Lagrangian form as

$$\frac{DC}{Dt} = \frac{1}{nR_d} \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C}{\partial X_j}) - \frac{q_s + \lambda n}{nR_d} C + \frac{q_s C_s}{nR_d} \quad (168)$$

where the substantial derivative,  $\frac{DC}{Dt} = \frac{\partial C}{\partial t} + \bar{u}_i \frac{\partial C}{\partial X_i}$ , indicates the rate of change in solute concentration along the path line of a contaminant particle (or a characteristic curve of the velocity field).  $\bar{u}_i = \frac{u_i}{R_d}$  represents the retarded velocity of a contaminant particle.

By introducing the finite-difference algorithm to the substantial derivative, Eq(168) can be approximated as

$$\frac{DC}{Dt} = \frac{C_p^{n+1} - C_p^{n*}}{\Delta t} = RHS \quad (169)$$

or

$$C_p^{n+1} = C_p^{n*} + \Delta t \times RHS \quad (170)$$

Where

- $C_p^{n+1}$  is the average solute concentration for cell  $P$  at the new time level  $(n+1)$ ;
- $C_p^{n*}$  is the average solute concentration for cell  $P$  at the new time level  $(n+1)$  due to advection alone, also referred to as the intermediate time level  $n^*$ .
- $RHS$  finite-difference approximation to the terms on the right-hand side of Eq(168).

Depending on the use of different Lagrangian techniques to approximate the advection term, the mixed Euler-Lagrangian methods may be loosely classified as the forward-tracking MOC, the backward-tracking MMOC and a combination of these two. MOC and MMOC were used in IGW 3-D transport model. Both the method of characteristics and the modified method of characteristics involve the use of a particle tracking technique.

### a) Particle Tracking

With the velocity field known, a numerical tracking scheme can be used to move particles from one position to another to approximate the advection of contaminant front. Traditionally, the first-order Euler algorithm has been used for particle tracking:

$$\begin{cases} X^{n+1} = \frac{\Delta t}{R_d} u_x(X^n, Y^n, Z^n) \\ Y^{n+1} = \frac{\Delta t}{R_d} u_y(X^n, Y^n, Z^n) \\ Z^{n+1} = \frac{\Delta t}{R_d} u_z(X^n, Y^n, Z^n) \end{cases} \quad (171)$$

Where  $X^{n+1}$ ,  $Y^{n+1}$  and  $Z^{n+1}$  are the particle coordinates at the new time level ( $n+1$ );  $X^n$ ,  $Y^n$  and  $Z^n$  are the particle coordinates at the old time level  $n$ ;  $u_x$ ,  $u_y$  and  $u_z$  are the velocities evaluated at  $(X^n, Y^n, Z^n)$ . A uniform time step size,  $\Delta t$ , is used for all particles during the particle tracking. For particles located in areas of relatively uniform velocity, the first order Euler algorithm may have sufficient accuracy. However, for particles located in areas of strongly converging or diverging flows, the first order algorithm may not be sufficiently accurate, unless time step size is very small. In these case a higher order algorithm such as the fourth-order Runge-Kutta method may be used. The basic idea of the fourth-order Runge Kutta method is to evaluate the velocity four times for each tracking step: once at the initial point, twice at two trial midpoints, and once at trial end point. A weight velocity based on values evaluated at these four points is used to move the particle to the new position. This process may be expressed as follows:

$$\begin{cases} X^{n+1} = X^n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\ Y^{n+1} = Y^n + \frac{l_1 + 2l_2 + 2l_3 + l_4}{6} \\ Z^{n+1} = Z^n + \frac{m_1 + 2m_2 + 2m_3 + m_4}{6} \end{cases} \quad (172)$$

Where

$$\begin{aligned}
k_1 &= \Delta t u_x(X^n, Y^n, Z^n, t^n) \\
l_1 &= \Delta t u_y(X^n, Y^n, Z^n, t^n) \\
m_1 &= \Delta t u_z(X^n, Y^n, Z^n, t^n) \\
k_2 &= \Delta t u_x\left(X^n + \frac{k_1}{2}, Y^n + \frac{l_1}{2}, Z^n + \frac{m_1}{2}, t^n + \frac{\Delta t}{2}\right) \\
l_2 &= \Delta t u_y\left(X^n + \frac{k_1}{2}, Y^n + \frac{l_1}{2}, Z^n + \frac{m_1}{2}, t^n + \frac{\Delta t}{2}\right) \\
m_2 &= \Delta t u_z\left(X^n + \frac{k_1}{2}, Y^n + \frac{l_1}{2}, Z^n + \frac{m_1}{2}, t^n + \frac{\Delta t}{2}\right) \\
k_3 &= \Delta t u_x\left(X^n + \frac{k_2}{2}, Y^n + \frac{l_2}{2}, Z^n + \frac{m_2}{2}, t^n + \frac{\Delta t}{2}\right) \\
l_3 &= \Delta t u_y\left(X^n + \frac{k_2}{2}, Y^n + \frac{l_2}{2}, Z^n + \frac{m_2}{2}, t^n + \frac{\Delta t}{2}\right) \\
m_3 &= \Delta t u_z\left(X^n + \frac{k_2}{2}, Y^n + \frac{l_2}{2}, Z^n + \frac{m_2}{2}, t^n + \frac{\Delta t}{2}\right) \\
k_4 &= \Delta t u_x(X^n + k_3, Y^n + l_3, Z^n + m_3, t^n + \Delta t) \\
l_4 &= \Delta t u_y(X^n + k_3, Y^n + l_3, Z^n + m_3, t^n + \Delta t) \\
m_4 &= \Delta t u_z(X^n + k_3, Y^n + l_3, Z^n + m_3, t^n + \Delta t)
\end{aligned}$$

In IGW 3-D model, first-order Euler method and fourth-order Runge-Kutta both are available.

From above analysis, one can obviously see that the evaluation of velocity at an arbitrary point or non-nodal quantity is required in either MOC or MMOC. The velocity interpolation schemes used in IGW 3-D transport model are simple trilinear and inverse distant interpolations.

Trilinear scheme has a general form as below (see Figure 24)

$$\begin{aligned}
u(\xi, \eta) &= u_1\phi_1(\xi, \eta, \zeta) + u_2\phi_2(\xi, \eta, \zeta) + u_3\phi_3(\xi, \eta, \zeta) + u_4\phi_4(\xi, \eta, \zeta) \\
&\quad + u_5\phi_5(\xi, \eta, \zeta) + u_6\phi_6(\xi, \eta, \zeta) + u_7\phi_7(\xi, \eta, \zeta) + u_8\phi_8(\xi, \eta, \zeta)
\end{aligned} \tag{173}$$

Where  $u_i$  ( $i=1,2,\dots,8$ ) are nodal quantities;  $\phi_i$  ( $i=1,2,3,\dots,8$ ) are shape functions which can be expressed as

$$\left\{ \begin{array}{l} \phi_1(\xi, \eta, \zeta) = \frac{(1-\xi)(1-\eta)(1-\zeta)}{8} \\ \phi_2(\xi, \eta, \zeta) = \frac{(1+\xi)(1-\eta)(1-\zeta)}{8} \\ \phi_3(\xi, \eta, \zeta) = \frac{(1+\xi)(1-\eta)(1+\zeta)}{8} \\ \phi_4(\xi, \eta, \zeta) = \frac{(1-\xi)(1-\eta)(1+\zeta)}{8} \\ \phi_5(\xi, \eta, \zeta) = \frac{(1-\xi)(1+\eta)(1-\zeta)}{4} \\ \phi_6(\xi, \eta, \zeta) = \frac{(1+\xi)(1+\eta)(1-\zeta)}{4} \\ \phi_7(\xi, \eta, \zeta) = \frac{(1+\xi)(1+\eta)(1+\zeta)}{4} \\ \phi_8(\xi, \eta, \zeta) = \frac{(1-\xi)(1+\eta)(1+\zeta)}{4} \end{array} \right. \quad (174)$$

$$\xi = \frac{X - X_C}{a}$$

$$\eta = \frac{Y - Y_C}{b}$$

$$\zeta = \frac{Z - Z_C}{c}$$

The inverse distant interpolation scheme has a general form as below

$$u(x, y, z) = u_1 w_1 + u_2 w_2 + u_3 w_3 + u_4 w_4 + u_5 w_5 + u_6 w_6 + u_7 w_7 + u_8 w_8 \quad (175)$$

Where  $u_i$  ( $i=1,2,\dots,8$ ) are nodal quantities;  $w_i$  ( $i=1,2,3,\dots,8$ ) are weighting factors which can be expressed as

$$w_i = \frac{S_i^{-pw}}{\sum_{j=1}^8 S_j^{-pw}} \quad (176)$$

where  $S_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2}$ ,  $pw$  is power which will be given by VB interface.

Other symbols in Eq(174) can be found in Figure 24. The interpolation was performed in Subroutine PTLINEAR3D.

First-order Euler method was implemented in Subroutine FORWARDTK0. Fourth-order Runge-Kutta was implemented in Subroutine FORWARDTK1.

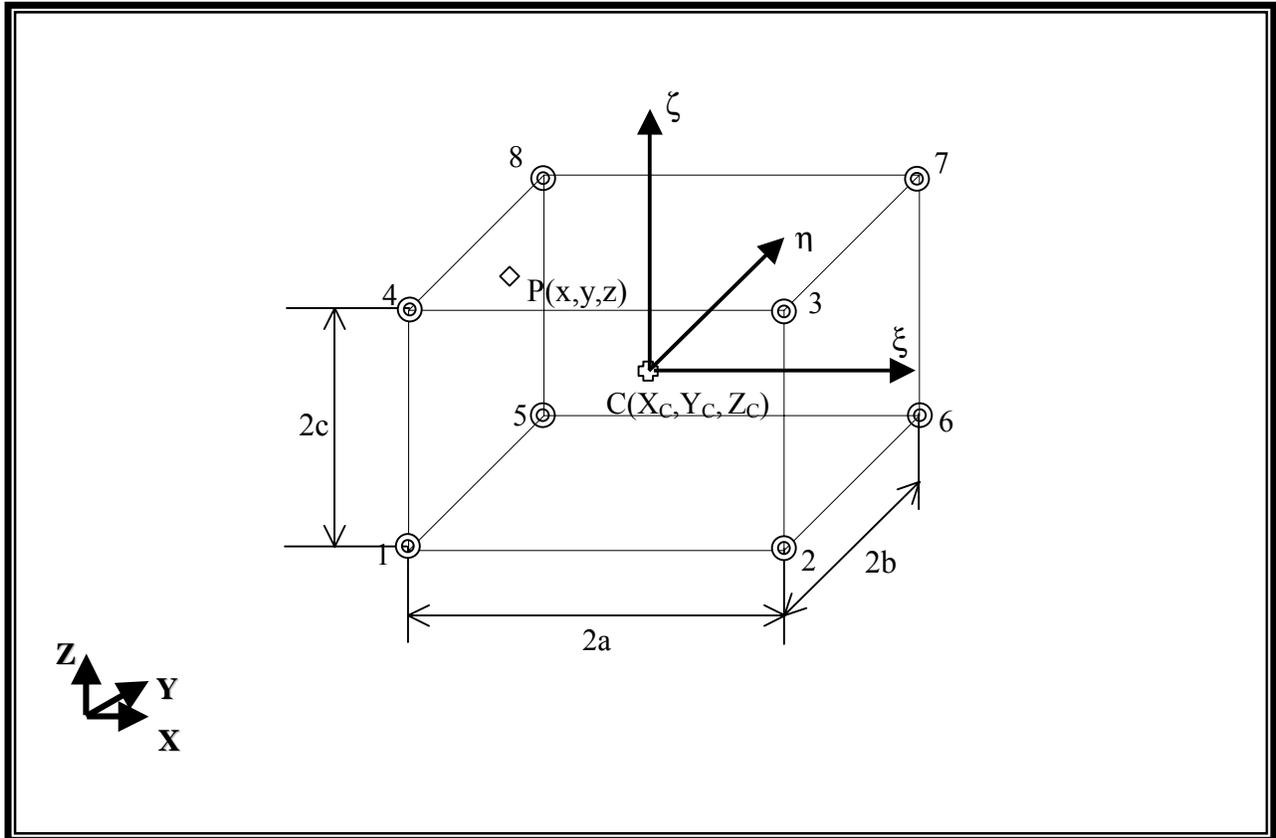


Figure 24. Trilinear Interpolation Scheme

**b) Determination of  $C_p^{n*}$** 

The MOC uses a conventional particle-tracking technique for solving the advection term. At the beginning of the simulation, a set of moving particles is distributed in the flow field either randomly or with fixed pattern. A concentration and a position in the Cartesian coordinate system are associated with each of these particles. Particles are tracked forward through the flow field. At the end of each time increment, the concentration at cell **P** due to advection alone over time increment,  $C_p^{n*}$ , is evaluated from the concentrations of moving particles which are located within that cell (see Figure 25). If a simple arithmetical averaged algorithm is used, this concentration can be expressed by the following equation:

$$C_p^{n*} = \frac{\sum_{i=1}^{NP_m} C_i^n}{NP_m}, \quad NP_m > 0 \quad (177)$$

where

- $NP_m$  number of particles within cell **P**
- $C_i^n$  concentration of the  $i^{th}$  particle at the old time level n, which is assumed to be constant in IGW.

Eq(177) could also be employed in Random Walk method which is not available in current IGW 3-D model yet.

The MMOC was originally developed to approximate the advection term accurately without sacrificing a great deal of computational efficiency. Unlike the MOC, which tracks a large number of moving particles forward in time and keeps track of the concentration and position of each particle, the MMOC places one fictitious particle at each nodal point of the fixed grid at each new time level  $n+1$ . The particle is tracked backward to find its position at the old time level n. The concentration associated with that position is used to approximate the  $C_p^{n*}$  term, that is

$$C_p^{n*} = C^n(X_p, Y_p, Z_p) \quad (178)$$

where  $(X_p, Y_p, Z_p)$  is coordinate of the position which a particle starting from nodal point **P** reaches when it is tracked backward along the reverse path line over the time increment  $\Delta t$  (see Figure 26). The concentration at position p at the old time level (n),  $C^n(X_p, Y_p, Z_p)$ , generally is obtained by interpolating from concentrations at its neighboring nodal points. In IGW 3-D model, the trilinear and inverse distant interpolation schemes as described in Eq(173) and Eq(175) were used.

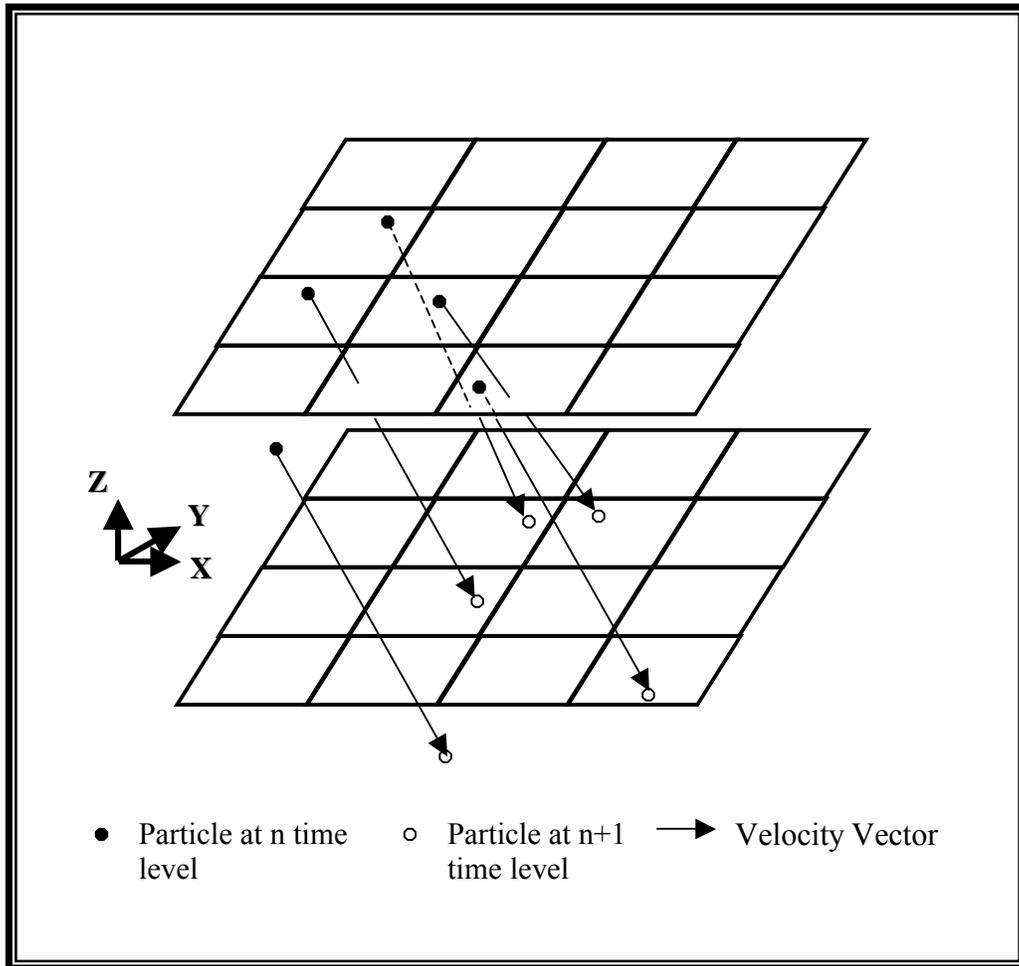


Figure 25. Illustration of the MOC in 3-D Case

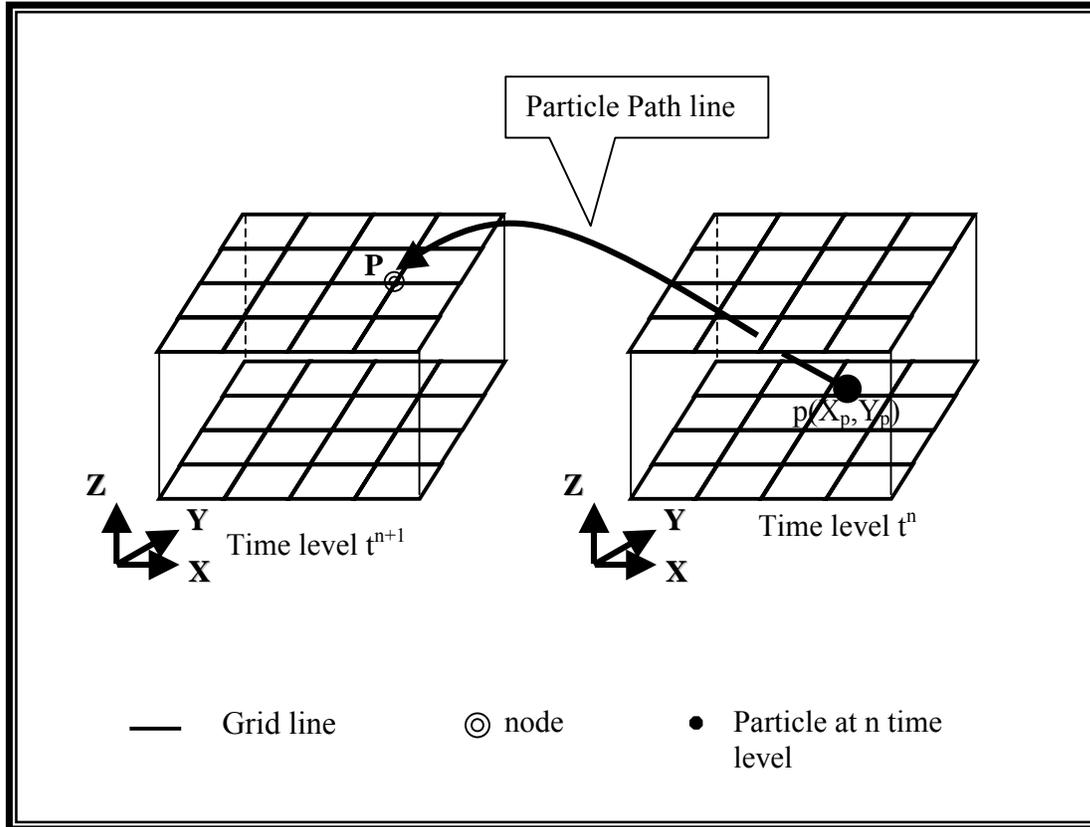


Figure 26 Illustration of the MMOC

### c) Approximation of Diffusion Term

Unlike the FD scheme, the target equation of Mixed Eulerian-Lagrangian Methods is Equation(143) in which the diffusion term has the similar form as that of Eq(142).

Referring to the course of derivation in FD scheme , diffusion term in Eq(143) may be approximated as

For traditional control volume technique:

$$\begin{aligned}
 Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S + a_T^{Diff} C_T + a_B^{Diff} C_B \\
 & + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} \\
 & + a_{NT} C_{NT} + a_{NB} C_{NB} + a_{ST} C_{ST} + a_{SB} C_{SB} \\
 & + a_{TW} C_{TW} + a_{BW} C_{BW} + a_{TE} C_{TE} + a_{BE} C_{BE} \\
 & - a_P^{Diff} C_P
 \end{aligned} \tag{179}$$

Where

$$\left\{ \begin{array}{l}
a_E^{Diff} = \left( \frac{\Delta Y_s \Delta Z_s n D_{xx}^e}{\Delta X} + \frac{D_{yx}^n - D_{yx}^s}{4} n \Delta Z_s + \frac{D_{zx}^t - D_{zx}^b}{4} n \Delta Y_s \right) \frac{1}{n R_d} \\
a_W^{Diff} = \left( \frac{\Delta Y_s \Delta Z_s n D_{xx}^w}{\Delta X} - \frac{D_{yx}^n - D_{yx}^s}{4} n \Delta Z_s - \frac{D_{zx}^t - D_{zx}^b}{4} n \Delta Y_s \right) \frac{1}{n R_d} \\
a_N^{Diff} = \left( \frac{\Delta X_s \Delta Z_s n D_{yy}^n}{\Delta Y} + \frac{D_{xy}^e - D_{xy}^w}{4} n \Delta Z_s + \frac{D_{zy}^t - D_{zy}^b}{4} n \Delta X_s \right) \frac{1}{n R_d} \\
a_S^{Diff} = \left( \frac{\Delta X_s \Delta Z_s n D_{yy}^s}{\Delta Y} - \frac{D_{xy}^e - D_{xy}^w}{4} n \Delta Z_s - \frac{D_{zy}^t - D_{zy}^b}{4} n \Delta X_s \right) \frac{1}{n R_d} \\
a_T^{Diff} = \left( \frac{\Delta X_s \Delta Y_s n D_{zz}^t}{\Delta Z} + \frac{D_{xz}^e - D_{xz}^w}{4} n \Delta Y_s + \frac{D_{yz}^n - D_{yz}^s}{4} n \Delta X_s \right) \frac{1}{n R_d} \\
a_B^{Diff} = \left( \frac{\Delta X_s \Delta Y_s n D_{zz}^b}{\Delta Z} - \frac{D_{xz}^e - D_{xz}^w}{4} n \Delta Y_s - \frac{D_{yz}^n - D_{yz}^s}{4} n \Delta X_s \right) \frac{1}{n R_d} \\
a_{NE} = \frac{1}{n R_d} \frac{D_{xy}^e + D_{yx}^n}{4} n \Delta Z_s, \quad a_{NW} = -\frac{1}{n R_d} \frac{D_{xy}^w + D_{yx}^n}{4} n \Delta Z_s \\
a_{SE} = -\frac{1}{n R_d} \frac{D_{xy}^e + D_{yx}^s}{4} n \Delta Z_s, \quad a_{SW} = \frac{1}{n R_d} \frac{D_{xy}^w + D_{yx}^s}{4} n \Delta Z_s \\
a_{TE} = \frac{1}{n R_d} \frac{D_{xz}^e + D_{zx}^t}{4} n \Delta Y_s, \quad a_{BE} = -\frac{1}{n R_d} \frac{D_{xz}^e + D_{zx}^b}{4} n \Delta Y_s \\
a_{TW} = -\frac{1}{n R_d} \frac{D_{xz}^w + D_{zx}^t}{4} n \Delta Y_s, \quad a_{BW} = \frac{1}{n R_d} \frac{D_{xz}^w + D_{zx}^b}{4} n \Delta Y_s \\
a_{NT} = \frac{1}{n R_d} \frac{D_{yz}^n + D_{zy}^t}{4} n \Delta X_s, \quad a_{NB} = -\frac{1}{n R_d} \frac{D_{yz}^n + D_{zy}^b}{4} n \Delta X_s \\
a_{ST} = -\frac{1}{n R_d} \frac{D_{yz}^s + D_{zy}^t}{4} n \Delta X_s, \quad a_{SB} = \frac{1}{n R_d} \frac{D_{yz}^s + D_{zy}^b}{4} n \Delta X_s \\
a_P^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_T^{Diff} + a_B^{Diff} + a_{NE} + a_{NW} + a_{SE} + a_{SW} \\
+ a_{TE} + a_{BE} + a_{TW} + a_{BW} + a_{NT} + a_{NB} + a_{ST} + a_{SB}
\end{array} \right. \quad (180)$$

For rotational control volume technique:

$$\begin{aligned}
Diff = & a_E^{Diff} C_E + a_W^{Diff} C_W + a_N^{Diff} C_N + a_S^{Diff} C_S + a_T^{Diff} C_T + a_B^{Diff} C_B \\
& + a_{NE} C_{NE} + a_{NW} C_{NW} + a_{SE} C_{SE} + a_{SW} C_{SW} \\
& + a_{NT} C_{NT} + a_{NB} C_{NB} + a_{ST} C_{ST} + a_{SB} C_{SB} \\
& + a_{TW} C_{TW} + a_{BW} C_{BW} + a_{TE} C_{TE} + a_{BE} C_{BE} \\
& + a_{NET} C_{NET} + a_{SET} C_{SET} + a_{SWT} C_{SWT} + a_{NWT} C_{NWT} \\
& + a_{NEB} C_{NEB} + a_{SEB} C_{SEB} + a_{SWB} C_{SWB} + a_{NWB} C_{NWB} \\
& - a_P^{Diff} C_P
\end{aligned} \tag{181}$$

Where

$$\left\{ \begin{aligned}
a_{NE} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{NE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{NE}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{NE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{NE}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{NE}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{NE}^B \\
a_{NW} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{NW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{NW}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{NW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{NW}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{NW}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{NW}^B \\
a_{SE} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{SE}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{SE}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{SE}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{SE}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{SE}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{SE}^B \\
a_{SW} = & \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{SW}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{SW}^W \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{SW}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{SW}^S \\
& + \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{SW}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{SW}^B
\end{aligned} \right. \tag{182a}$$

$$\left. \begin{aligned}
a_{NT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NT}^B \\
a_{NB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NB}^B \\
a_{ST} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{ST}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{ST}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{ST}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{ST}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{ST}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{ST}^B \\
a_{SB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{SB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{SB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{SB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{SB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{SB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{SB}^B
\end{aligned} \right\} \quad (182b)$$

$$\left\{ \begin{aligned}
a_{NET} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NET}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NET}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NET}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NET}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NET}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NET}^B \\
a_{NWT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NWT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{NWT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NWT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{NWT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NWT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{NWT}^B \\
a_{SET} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{SET}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{SET}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{SET}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{SET}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{SET}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{SET}^B \\
a_{SWT} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{SWT}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \alpha_{SWT}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{SWT}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \alpha_{SWT}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{SWT}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \alpha_{SWT}^B
\end{aligned} \right. \quad (182c)$$

$$\left\{ \begin{aligned}
a_{NEB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{NEB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{NEB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{NEB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{NEB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{NEB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{NEB}^B \\
a_{NWB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{NWB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{NWB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{NWB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{NWB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{NWB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{NWB}^B \\
a_{SEB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{SEB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{SEB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{SEB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{SEB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{SEB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{SEB}^B \\
a_{SWB} &= \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{ie} \alpha_{SWB}^E + \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{xx}^{iw} \alpha_{SWB}^W \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{in} \alpha_{SWB}^N + \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD_{yy}^{is} \alpha_{SWB}^S \\
&+ \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{it} \alpha_{SWB}^T + \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD_{zz}^{ib} \alpha_{SWB}^B
\end{aligned} \right. \quad (182d)$$

$$\left\{ \begin{array}{l}
a_E^{Diff} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_e} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \sum_j \alpha_E^j, \quad j = E, W, N, S, T, B \\
a_W^{Diff} = \frac{\Delta S_n + \Delta S_s}{2\Delta S_w} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{xx} \sum_j \alpha_W^j, \quad j = E, W, N, S, T, B \\
a_N^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_n} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \sum_j \alpha_N^j, \quad j = E, W, N, S, T, B \\
a_S^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_s} \frac{\Delta S_t + \Delta S_b}{2nR_d} nD'_{yy} \sum_j \alpha_S^j, \quad j = E, W, N, S, T, B \\
a_T^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_t} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \sum_j \alpha_T^j, \quad j = E, W, N, S, T, B \\
a_B^{Diff} = \frac{\Delta S_e + \Delta S_w}{2\Delta S_b} \frac{\Delta S_n + \Delta S_s}{2nR_d} nD'_{zz} \sum_j \alpha_B^j, \quad j = E, W, N, S, T, B \\
a_P^{Diff} = a_E^{Diff} + a_W^{Diff} + a_N^{Diff} + a_S^{Diff} + a_T^{Diff} + a_B^{Diff} \\
\quad + a_{NE} + a_{NW} + a_{SE} + a_{SW} \\
\quad + a_{NT} + a_{NB} + a_{ST} + a_{SB} \\
\quad + a_{TW} + a_{BW} + a_{TE} + a_{BE} \\
\quad + a_{NET} + a_{SET} + a_{SWT} + a_{NWT} \\
\quad + a_{NEB} + a_{SEB} + a_{SWB} + a_{NWB}
\end{array} \right. \quad (182e)$$

#### d) Approximation of Time-Derivative Term

The substantial derivative in Eq(143) can be written as

$$\Delta X_s \Delta Y_s \Delta Z_s \frac{DC}{Dt} = \Delta X_s \Delta Y_s \Delta Z_s \frac{C_P^{n+1} - C_P^{n*}}{\Delta t} = a_P^t C_P^{n+1} - S_f^t \quad (183)$$

Where

$$\left\{ \begin{array}{l}
a_P^t = \Delta X_s \Delta Y_s \Delta Z_s \frac{1}{\Delta t} \\
S_f^t = \Delta X_s \Delta Y_s \Delta Z_s \frac{1}{\Delta t} C_P^{n*}
\end{array} \right. \quad (184)$$

For rotational control volume technique,  $\Delta X_s \Delta Y_s \Delta Z_s$  will be replaced with  $\frac{\Delta S_e + \Delta S_w}{2} \frac{\Delta S_n + \Delta S_s}{2} \frac{\Delta S_t + \Delta S_b}{2}$  in Eq(183) and Eq(184).

#### e) Approximation of Source/Sink Term

Similar approximation forms as in FD scheme can be obtained as below

$$q_s C_s = a_p^O C_p + S_f^O \quad (185)$$

Where

$$\left\{ \begin{array}{l} a_p^O = \frac{1}{nR_d} \text{Max}[-q_s^{\text{Well}}, 0] + \frac{1}{nR_d} \text{Max}[-q_s^{\text{Recharge}}, 0] \\ \quad + \frac{1}{nR_d} \text{Max}[-q_s^{\text{River}}, 0] + \frac{1}{nR_d} \text{Max}[-q_s^{\text{Ghead}}, 0] \\ \quad + \frac{1}{nR_d} \text{Max}[-q_s^{\text{Drain}}, 0] \\ S_f^O = \frac{1}{nR_d} \text{Max}[q_s^{\text{Well}}, 0] C_s^{\text{Well}} + \frac{1}{nR_d} \text{Max}[q_s^{\text{Recharge}}, 0] C_s^{\text{Recharge}} \\ \quad + \frac{1}{nR_d} \text{Max}[q_s^{\text{River}}, 0] C_s^{\text{River}} + \frac{1}{nR_d} \text{Max}[q_s^{\text{Ghead}}, 0] C_s^{\text{Ghead}} \end{array} \right. \quad (186)$$

Again,  $q_s^{\text{Well}}$ ,  $q_s^{\text{Recharge}}$ ,  $q_s^{\text{River}}$ ,  $q_s^{\text{Drain}}$  and  $q_s^{\text{Ghead}}$  are fluxes contributed from source/sink of wells, recharge, river, drain and general head which are stored in derived types (one dimension) Wells%Q, RECHS%Q, Rivers%Q, Drain%Q and HDEPENDS %Q respectively in source code. Calculation of every kind of  $q_s$  is performed by Subroutine QTOTAL3D.

#### f) Approximation of Decay Term

From Eq(143), instead having only one decay term as in FD scheme, there are one more decay term,  $\frac{q_s}{nR_d} C$ , need to be included in these methods in addition to the real one decay term. They can be easily approximated as

$$\text{Decay} = a_p^D C_p \quad (187)$$

$$\text{Where } a_p^D = \frac{q_s + \lambda n}{nR_d}$$

#### g) Coefficient Matrix Assembling and Solution Technique

After obtaining  $C^{n*}_p$  by MOC or MMOC, from Eq(179) or Eq(181), Eq(183), Eq(185) and Eq(187), readily gives a set of linear equations for solving the final concentration at new time level as follow

Traditional control volume technique:

$$\begin{aligned}
(a_P^{ADV} + a_P^{diff} + a_P^t + a_P^Q + a_P^D)C_P = & (a_E^{ADV} + a_E^{Diff})C_E + (a_W^{ADV} + a_W^{Diff})C_W \\
& + (a_N^{ADV} + a_N^{Diff})C_N + (a_S^{ADV} + a_S^{Diff})C_S \\
& + (a_T^{ADV} + a_T^{Diff})C_T + (a_B^{ADV} + a_B^{Diff})C_B \\
& + a_{NE}C_{NE} + a_{NW}C_{NW} + a_{SE}C_{SE} + a_{SW}C_{SW} \\
& + a_{NT}C_{NT} + a_{NB}C_{NB} + a_{ST}C_{ST} + a_{SB}C_{SB} \\
& + a_{TW}C_{TW} + a_{BW}C_{BW} + a_{TE}C_{TE} + a_{BE}C_{BE} \\
& + S_f^Q + S_f^t
\end{aligned} \tag{188a}$$

Rotation control volume technique:

$$\begin{aligned}
(a_P^{ADV} + a_P^{diff} + a_P^t + a_P^Q + a_P^D)C_P = & (a_E^{ADV} + a_E^{Diff})C_E + (a_W^{ADV} + a_W^{Diff})C_W \\
& + (a_N^{ADV} + a_N^{Diff})C_N + (a_S^{ADV} + a_S^{Diff})C_S \\
& + (a_T^{ADV} + a_T^{Diff})C_T + (a_B^{ADV} + a_B^{Diff})C_B \\
& + a_{NE}C_{NE} + a_{NW}C_{NW} + a_{SE}C_{SE} + a_{SW}C_{SW} \\
& + a_{NT}C_{NT} + a_{NB}C_{NB} + a_{ST}C_{ST} + a_{SB}C_{SB} \\
& + a_{TW}C_{TW} + a_{BW}C_{BW} + a_{TE}C_{TE} + a_{BE}C_{BE} \\
& + a_{NET}C_{NET} + a_{SET}C_{SET} + a_{SWT}C_{SWT} + a_{NWT}C_{NWT} \\
& + a_{NEB}C_{NEB} + a_{SEB}C_{SEB} + a_{SWB}C_{SWB} + a_{NWB}C_{NWB} \\
& + S_f^Q + S_f^t
\end{aligned} \tag{188b}$$

The same routine SOR technique was applied to solve Eq(188) in IGW 3-D model. Its iterative equation can be expressed as follow

Traditional control volume:

$$\begin{aligned}
C_P^{k+1} = C_P^k + \frac{\alpha}{(a_P + a_P^t + a_P^Q)} \{ & a_E C_E^k + a_W C_W^{k+1} + a_N C_N^k + a_S C_S^{k+1} + a_T C_T^k + a_B C_B^{k+1} \\
& + a_{NE} C_{NE}^k + a_{NW} C_{NW}^k + a_{SE} C_{SE}^{k+1} + a_{SW} C_{SW}^{k+1} \\
& + a_{NT} C_{NT}^k + a_{NB} C_{NB}^k + a_{ST} C_{ST}^k + a_{SB} C_{SB}^{k+1} \\
& + a_{TW} C_{TW}^k + a_{BW} C_{BW}^{k+1} + a_{TE} C_{TE}^k + a_{BE} C_{BE}^{k+1} \\
& + S_f^Q + S_f^t - (a_P + a_P^t + a_P^Q)C_P^k \}
\end{aligned} \tag{189a}$$

Rotation control volume:

$$\begin{aligned}
 C_P^{k+1} = C_P^k + \frac{\alpha}{(a_P + a_P^t + a_P^Q)} & \{ a_E C_E^k + a_W C_W^{k+1} + a_N C_N^k + a_S C_S^{k+1} + a_T C_T^k + a_B C_B^{k+1} \\
 & + a_{NE} C_{NE}^k + a_{NW} C_{NW}^k + a_{SE} C_{SE}^{k+1} + a_{SW} C_{SW}^{k+1} \\
 & + a_{NT} C_{NT}^k + a_{NB} C_{NB}^k + a_{ST} C_{ST}^k + a_{SB} C_{SB}^{k+1} \\
 & + a_{TW} C_{TW}^k + a_{BW} C_{BW}^{k+1} + a_{TE} C_{TE}^k + a_{BE} C_{BE}^{k+1} \\
 & + a_{NET} C_{NET}^k + a_{SET} C_{SET}^k + a_{SWT} C_{SWT}^k + a_{NWT} C_{NWT}^k \\
 & + a_{NEB} C_{NEB}^k + a_{SEB} C_{SEB}^{k+1} + a_{SWB} C_{SWB}^{k+1} + a_{NWB} C_{NWB}^k \\
 & + S_f^Q + S_f^t - (a_P + a_P^t + a_P^Q) C_P^k \} \quad (189b)
 \end{aligned}$$

Where

$$\begin{aligned}
 a_P &= a_P^{ADV} + a_P^{Diff} + a_P^t + a_P^Q + a_P^D \\
 a_E &= a_E^{ADV} + a_E^{Diff} \\
 a_W &= a_W^{ADV} + a_W^{Diff} \\
 a_N &= a_N^{ADV} + a_N^{Diff} \\
 a_S &= a_S^{ADV} + a_S^{Diff} \\
 a_T &= a_T^{ADV} + a_T^{Diff} \\
 a_B &= a_B^{ADV} + a_B^{Diff}
 \end{aligned}$$

$k$  index of iteration number

$\alpha$  relaxation factor.

The final matrix assembling process is carried out in Subroutine SORCBAR3D.

### 3.2.4 Special Treatments

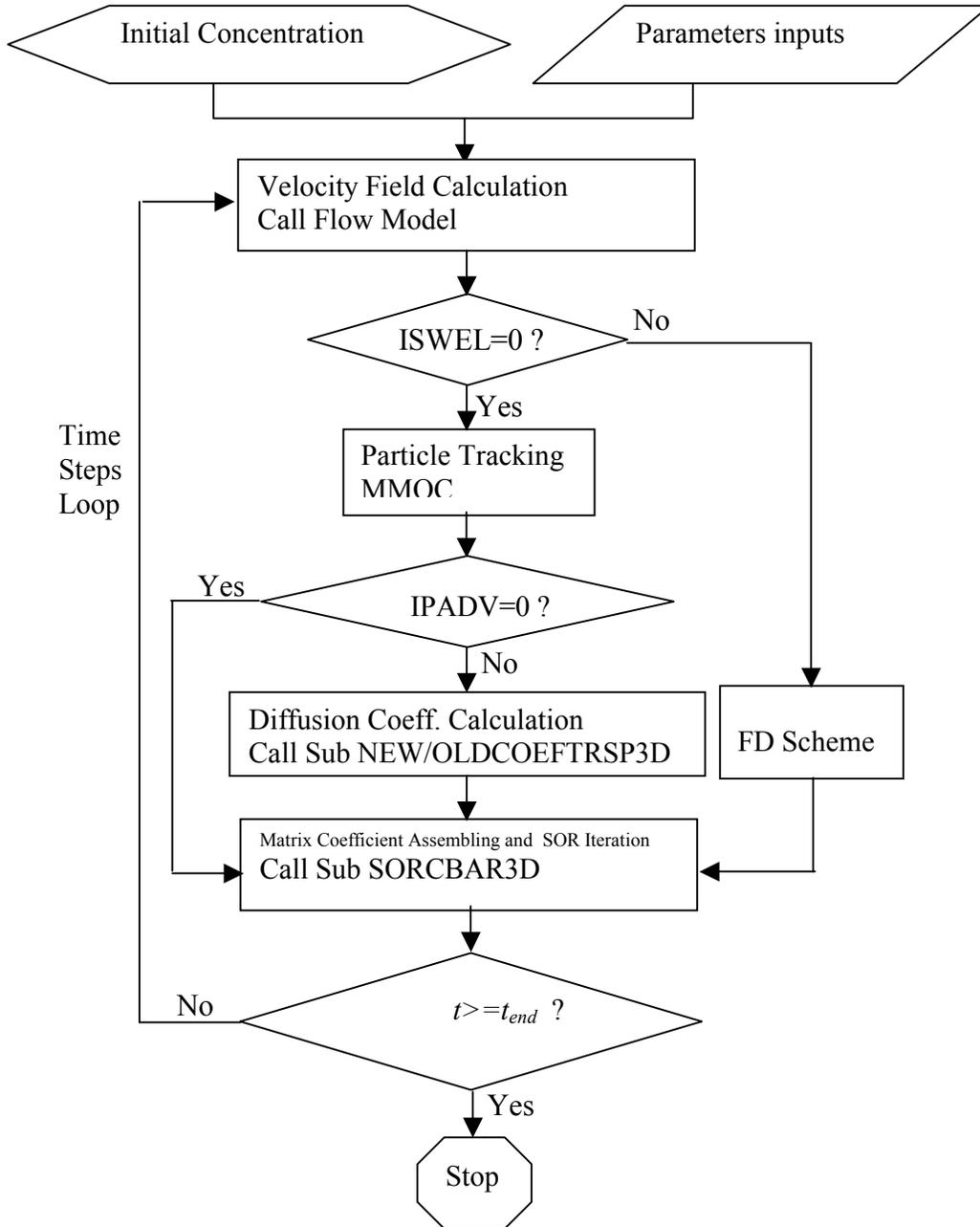
- (1) As doing in IGW 3-D flow model, computational domain in IGW 3-D transport model, with dimension of 1 to NI, 1 to NJ and 1 to NK, is also expanded to that with dimension of 0 to NI+1, 0 to NJ+1 and 0 to NK+1 and every parameter at these expanded nodes are assigned to be zero.
- (2) A concentration boundary condition indicator variable ICB() is allocated to identify cell of the continuous source, ICB=-1, or cell of instantaneous source.
- (3) The cross terms,  $D_{ij}$ , are not usually equal to zero in non-uniform flow field unless the longitudinal dispersivity is equal to both horizontal transverse and vertical transverse dispersivities or both are equal to zero (pure advection). In this case, rotational control volume technique has to be used in order to obtain a non-negative concentration distribution. The difference in implementation of rotational control volume technique between the flow and transport models is how to determine the rotation angles  $\theta$  and  $\beta$ . In flow model,  $\theta$  and  $\beta$  are assumed to be equal to the anisotropy orientation angles or aligned with geological layer's formation. In transport model, the first principal direction has been assumed to be aligned with velocity vector.
- (4) A switch, IPADV was set to skip the burden calculation of diffusion coefficient when there is no dispersivities in both longitudinal and transverse directions.
- (5) In IGW transport model, velocity at well node is assumed to be zero.
- (6) FD scheme is always applied at well cell in IGW.
- (7) When grid size is such large that multiple wells would be mapped to a same node, the Visual Fortran code still can automatically handle this situation by dividing those wells into two parts – normal well (saying that one well, one node) and additional wells which have been considered as another kind of source/sink applying simultaneously at the same node.
- (8) No random walk method in current 3-D version.

### 3.2.5 Matrix Solver

SOR iterative technique was employed in IGW 3-D transport model. In addition to SOR, Subroutine SORCBAR also give a final matrix which could be solved by other mean of advanced methods. Main diagonal elements are stored in  $S00(I,J,k)+CST1(I,J,k)\%SP$ , other diagonal elements are stored in those variables listed Table 4, and RHS vector is stored in variable SUM23(I,J,k). Therefore, slightly modification to Subroutine SORCBAR3D can made the current matrix fit to your preferred matrix solvers.

### 3.2.6 Numerical Solution Procedure Flow Chart

The sequence of operations for IGW 3-D transport model is illustrated in the following flow diagram:



### 3.3 Chemical Reaction in Three Dimensional Model

#### 3.3.1 General Technique for Dealing with Chemical Reaction

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

$$\frac{\partial(nC_k)}{\partial t} + \frac{\partial(nu_i C_k)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C_k}{\partial X_j}) - \lambda_k nC_k + q_s C_{s_k} + R_c, \quad k = 1, 2, \dots, m \quad (190)$$

$$\frac{d\tilde{C}_{im}}{dt} = \tilde{R}_c, \quad im = 1, 2, \dots, (n - m) \quad (191)$$

Where

- $n$  the total number of species
- $m$  the total number of aqueous-phase (mobile) species
- $C_k$  the aqueous-phase concentration of the  $k^{\text{th}}$  species
- $\tilde{C}_{im}$  the solid-phase concentration of the  $im^{\text{th}}$  species
- $R_c$  the rate of all reaction that occur in the aqueous-phase
- $\tilde{R}_c$  the rate of all reaction that occur in the soil-phase

other symbols can be found in above sections.

In IGW 3-D model, a reaction operator-split numerical strategy was introduced to solve any number of the coupled transport equations of the forms (190) and (191). The operator-split technique allows Eq(190) being divided into two distinct equations:

$$\frac{\partial(nC_k)}{\partial t} + \frac{\partial(nu_i C_k)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C_k}{\partial X_j}) - \lambda_k nC_k + q_s C_{s_k}, \quad k = 1, 2, \dots, m \quad (192)$$

$$\frac{\partial(nC_k)}{\partial t} = R_c, \quad k = 1, 2, \dots, m \quad (193)$$

Eq(192) and Eq(193) together with Eq(191) form chemical reaction equations to be solved in IGW 3-D model. The logical steps involved in the numerical solution procedure are outlined as below:

(1) Equation (192) which describes the intermediate concentration of mobile species,  $C_k^{n*}$ , is initially solved for a transport time step  $\Delta t$  by any methods mentioned above sections;

(2) Considering  $C_k^{n*}$  as the concentration at previous time step, equation (193) and (191) which form a set of ordinary differential equations respect with  $C_k$  and  $\tilde{C}_{im}$  then are complexly solved by the proposed Linearized Approach to solving chemical reaction equations.

### 3.3.2 Chemical Reaction Models

The express of rate term,  $R_c$  or  $\tilde{R}_c$ , in Eq(193) or Eq(191) depends on what kind of reaction model would be used. There are seven pre-defined reaction models available in IGW 3-D model. Among them, models describing NAPL dissolution process, dual-porosity system and rate-limited sorption reactions are grouped together as an add-on model due to the fact that they all have the same form of the reaction equations, which means that one or all of these three models can be added in any others to form a new combinations --- new derived models. A briefly description of each model is given below, however, the details of all the reaction models included in source code can be found in RT3D 1.0 Manual by P. J. Clement.

#### A) Model 0

Model 0 deals with only the concentration of the contaminant in immobile phase. This model includes three sub-models: NAPL dissolution process, dual-porosity system and rate-limited sorption reactions.

##### a) Rate-Limited Sorption Reactions

This model is used to simulate mass-transfer-limited sorption reaction. The fate and transport of sorbing solute in aqueous and soil phases can be predicted using the following equations:

$$\frac{\partial(nC)}{\partial t} + \rho \frac{\partial \tilde{C}}{\partial t} + \frac{\partial(nu_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C}{\partial X_j}) - \lambda nC + q_s C_s \quad (194)$$

$$\rho \frac{d\tilde{C}}{dt} = \xi \left( C - \frac{\tilde{C}}{\gamma} \right) \quad (195)$$

where  $\xi$  is the mass transfer rate coefficient,  $\gamma$  is the linear partitioning coefficient.

After reaction-operator splitting, the reaction model for the considered problem reduces to:

$$\begin{cases} \frac{dC}{dt} = -\xi \left( C - \frac{\tilde{C}}{\gamma} \right) \\ \frac{d\tilde{C}}{dt} = \frac{\xi}{\rho} \left( C - \frac{\tilde{C}}{\gamma} \right) \end{cases} \quad (196)$$

##### b) Dual Domain Model

This model is used to simulate contaminant transport in dual domain porous media. Dual domain model can also be used for modeling transport in fractured porous

formation if the continuum approximation is assumed to valid. The contaminant fate and transport in a dual porosity system can be predicted using the following equations:

$$\frac{\partial(n_m C)}{\partial t} + \frac{\partial(n_m u_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (n_m D_{ij} \frac{\partial C}{\partial X_j}) - \lambda n_m C + q_s C_s + \frac{\xi}{n_m} (\tilde{C} - C) \quad (197)$$

$$n_{im} \frac{d\tilde{C}}{dt} = -\xi (\tilde{C} - C) \quad (198)$$

where  $n_m$  is the mobile-phase porosity,  $n_{im}$  is the immobile phase porosity.

After reaction-operator splitting, the reaction model for the problem reduces to:

$$\begin{cases} \frac{dC}{dt} = \frac{\xi}{n_m} (\tilde{C} - C) \\ \frac{d\tilde{C}}{dt} = -\frac{\xi}{n_{im}} (\tilde{C} - C) \end{cases} \quad (199)$$

### c) NAPL Dissolution Process

This model is used to simulate the NAPL dissolution process that are coupled with biodegradation kinetics. The fate and transport of contaminants originating from a NAPL zone can be predicted using the following equations:

$$\frac{\partial(nC)}{\partial t} + \frac{\partial(nu_i C)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C}{\partial X_j}) - \lambda nC + q_s C_s + K_{La} (C^* - C) \quad (200)$$

$$n \frac{d\tilde{C}}{dt} = -K_{La} (C^* - C) \quad (201)$$

where  $C^*$  is equilibrium aqueous phase concentration,  $K_{La}$  is a lumped mass transfer rate.

After reaction-operator splitting, the reaction model for the problem reduces to:

$$\begin{cases} \frac{dC}{dt} = K_{La} (C^* - C) \\ \frac{d\tilde{C}}{dt} = -K_{La} (C^* - C) \end{cases} \quad (201)$$

From Eq(196), Eq(199) and Eq(201), one can see that they all have the same form, and can be grouped as the same kind of model. Model 0 has been coded in Subroutine MODEL0 in VF source code.

### B) Model 1: Instantaneous Aerobic Decay of BTEX

This model is used to simulate aerobic degradation of BTEX using an instantaneous reaction model. The transport equations solved in this model are:

$$\frac{\partial(nC^{HC})}{\partial t} + \frac{\partial(nu_i C^{HC})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{HC}}{\partial X_j}) - \lambda^{HC} nC^{HC} + q_s C_s^{HC} + R^{HC} \quad (202)$$

$$\frac{\partial(nC^{O_2})}{\partial t} + \frac{\partial(nu_i C^{O_2})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{O_2}}{\partial X_j}) - \lambda^{O_2} nC^{O_2} + q_s C_s^{O_2} + R^{O_2} \quad (203)$$

where

$C^{HC}$	the hydrocarbon concentration
$C^{O_2}$	the oxygen concentration
$R^{HC}$	the removal rate of hydrocarbon
$R^{O_2}$	the removal rate of oxygen

At each time step, an instantaneous reaction algorithm is used to model the removal rates. According to this algorithm, either hydrocarbon or oxygen (whichever is limiting) will be reduced to zero within a grid cell, after a reaction time step. The reaction equations can be written as:

$$\begin{cases} C^{HC}(t+1) = C^{HC}(t) - \frac{C^{O_2}(t)}{F}, & \text{when } C^{HC}(t) > \frac{C^{O_2}(t)}{F} \\ C^{O_2}(t+1) = 0 \end{cases} \quad (204a)$$

$$\begin{cases} C^{O_2}(t+1) = C^{O_2}(t) - F \times C^{HC}(t), & \text{when } C^{O_2}(t) > F \times C^{HC}(t) \\ C^{HC}(t+1) = 0 \end{cases} \quad (204b)$$

where  $t$  refers to a particular time step and  $F$  is the stoichiometric ratio.

Model 1 has been coded in Subroutine MODEL1.

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

This model is used to simulate instantaneous biodegradation of BTEX via five different degradation pathways: aerobic respiration, denitrification, iron reduction, sulfate reduction, and methanogenesis. The transport equations solved in this model are:

$$\frac{\partial(nC^{HC})}{\partial t} + \frac{\partial(nu_i C^{HC})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{HC}}{\partial X_j}) - \lambda^{HC} nC^{HC} + q_s C_s^{HC} + R^{HC} \quad (204)$$

$$\frac{\partial(nC^{O_2})}{\partial t} + \frac{\partial(nu_i C^{O_2})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{O_2}}{\partial X_j}) - \lambda^{O_2} nC^{O_2} + q_s C_s^{O_2} + R^{O_2} \quad (205)$$

$$\frac{\partial(nC^{NO_3})}{\partial t} + \frac{\partial(nu_i C^{NO_3})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{NO_3}}{\partial X_j}) - \lambda^{NO_3} nC^{NO_3} + q_s C_s^{NO_3} + R^{NO_3} \quad (206)$$

$$\frac{\partial(nC^{Fe^{2+}})}{\partial t} + \frac{\partial(nu_i C^{Fe^{2+}})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{Fe^{2+}}}{\partial X_j}) - \lambda^{Fe^{2+}} nC^{Fe^{2+}} + q_s C_s^{Fe^{2+}} + R^{Fe^{2+}} \quad (207)$$

$$\frac{\partial(nC^{SO_4})}{\partial t} + \frac{\partial(nu_i C^{SO_4})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{SO_4}}{\partial X_j}) - \lambda^{SO_4} nC^{SO_4} + q_s C_s^{SO_4} + R^{SO_4} \quad (208)$$

$$\frac{\partial(nC^{CH_4})}{\partial t} + \frac{\partial(nu_i C^{CH_4})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{CH_4}}{\partial X_j}) - \lambda^{CH_4} nC^{CH_4} + q_s C_s^{CH_4} + R^{CH_4} \quad (209)$$

All removal terms “R” are computed using an instantaneous reaction model similar to the aerobic instantaneous model. The following general instantaneous reaction algorithm is used to utilize different electron acceptors sequentially:

$$\begin{cases} C^D(t+1) = C^D(t) - \frac{C^A(t)}{F}, & \text{when } C^D(t) > \frac{C^A(t)}{F} \\ C^A(t+1) = 0 \end{cases} \quad (210a)$$

$$\begin{cases} C^A(t+1) = C^A(t) - F \times C^D(t), & \text{when } C^A(t) > F \times C^D(t) \\ C^D(t+1) = 0 \end{cases} \quad (210b)$$

where

- $t$  refers to a particular time step
- $F$  is the stoichiometric ratio
- $C^D$  is the electron donor concentration
- $C^A$  is the electron acceptor concentration

Model 2 has been coded in Subroutine MODEL2.

**D) Model 3: Kinetic-Limited Degradation of BTEX Using Multiple Electron Acceptors**

This model is used to simulate kinetic-limited biodegradation of BTEX via five different degradation pathways: aerobic respiration, denitrification, iron reduction, sulfate reduction, and methanogenesis. The transport equations solved in this model are:

$$\frac{\partial(nC^{HC})}{\partial t} + \frac{\partial(nu_i C^{HC})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{HC}}{\partial X_j}) - \lambda^{HC} nC^{HC} + q_s C_s^{HC} + R^{HC} \quad (211)$$

$$\frac{\partial(nC^{O_2})}{\partial t} + \frac{\partial(nu_i C^{O_2})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{O_2}}{\partial X_j}) - \lambda^{O_2} nC^{O_2} + q_s C_s^{O_2} + R^{O_2} \quad (212)$$

$$\frac{\partial(nC^{NO_3})}{\partial t} + \frac{\partial(nu_i C^{NO_3})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{NO_3}}{\partial X_j}) - \lambda^{NO_3} nC^{NO_3} + q_s C_s^{NO_3} + R^{NO_3} \quad (213)$$

$$\frac{\partial(nC^{Fe^{2+}})}{\partial t} + \frac{\partial(nu_i C^{Fe^{2+}})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{Fe^{2+}}}{\partial X_j}) - \lambda^{Fe^{2+}} nC^{Fe^{2+}} + q_s C_s^{Fe^{2+}} + R^{Fe^{2+}} \quad (214)$$

$$\frac{\partial(nC^{SO_4})}{\partial t} + \frac{\partial(nu_i C^{SO_4})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{SO_4}}{\partial X_j}) - \lambda^{SO_4} nC^{SO_4} + q_s C_s^{SO_4} + R^{SO_4} \quad (215)$$

$$\frac{\partial(nC^{CH_4})}{\partial t} + \frac{\partial(nu_i C^{CH_4})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{CH_4}}{\partial X_j}) - \lambda^{CH_4} nC^{CH_4} + q_s C_s^{CH_4} + R^{CH_4} \quad (216)$$

The total rate of hydrocarbon destruction via all decay processes is written as rates of electron acceptor utilization or product formation are given by rates of hydrocarbon destruction multiplied by an appropriate yield coefficient (Y):

$$\begin{cases} R^{O_2} = Y_{O_2/HC} R_{HC,O_2} \\ R^{NO_3} = Y_{NO_3/HC} R_{HC,NO_3} \\ R^{Fe^{2+}} = Y_{Fe^{2+}/HC} R_{HC,Fe^{2+}} \\ R^{SO_4} = Y_{SO_4/HC} R_{HC,SO_4} \\ R^{CH_4} = Y_{CH_4/HC} R_{HC,CH_4} \end{cases} \quad (217)$$

Where

$$R_{HC,O_2} = -k_{O_2} C^{HC} \frac{C^{O_2}}{K_{O_2} + C^{O_2}}$$

$$R_{HC,NO_3} = -k_{NO_3} C^{HC} \frac{C^{NO_3}}{K_{NO_3} + C^{NO_3}} \frac{K_{i,O_2}}{K_{i,O_2} + C^{O_2}}$$

$$R_{HC,Fe^{2+}} = -k_{Fe^{3+}} C^{HC} \frac{C^{Fe^{3+}}}{K_{Fe^{3+}} + C^{Fe^{3+}}} \frac{K_{i,O_2}}{K_{i,O_2} + C^{O_2}} \frac{K_{i,NO_3}}{K_{i,NO_3} + C^{NO_3}}$$

$$R_{HC,SO_4} = -k_{SO_4} C^{HC} \frac{C^{SO_4}}{K_{SO_4} + C^{SO_4}} \frac{K_{i,O_2}}{K_{i,O_2} + C^{O_2}} \frac{K_{i,NO_3}}{K_{i,NO_3} + C^{NO_3}} \frac{K_{i,Fe^{3+}}}{K_{i,Fe^{3+}} + C^{Fe^{3+}}}$$

$$R_{HC,CH_4} = -k_{CH_4} C^{HC} \frac{C^{CH_4}}{K_{CH_4} + C^{CH_4}} \frac{K_{i,O_2}}{K_{i,O_2} + C^{O_2}} \frac{K_{i,NO_3}}{K_{i,NO_3} + C^{NO_3}} \frac{K_{i,Fe^{3+}}}{K_{i,Fe^{3+}} + C^{Fe^{3+}}} \frac{K_{i,SO_4}}{K_{i,SO_4} + C^{SO_4}}$$

- $k_{O_2}$  hydrocarbon decay rate via aerobic process  
 $k_{NO_3}$  hydrocarbon decay rate via denitrification  
 $k_{Fe^{3+}}$  hydrocarbon decay rate via iron reduction  
 $k_{SO_4}$  hydrocarbon decay rate via sulfate reduction  
 $k_{CH_4}$  hydrocarbon decay rate via methanogenesis  
 $K_{O_2}$  half saturation constant for oxygen  
 $K_{NO_3}$  half saturation constant for nitrate  
 $K_{Fe^{3+}}$  half saturation constant for  $Fe^{3+}$   
 $K_{SO_4}$  half saturation constant for sulfate  
 $K_{CH_4}$  half saturation constant for methane  
 $K_{i,O_2}$  inhibition coefficient for oxygen reaction  
 $K_{i,NO_3}$  inhibition coefficient for nitrate reaction  
 $K_{i,Fe^{3+}}$  inhibition coefficient for  $Fe^{3+}$  reaction  
 $K_{i,SO_4}$  inhibition coefficient for sulfate reaction

Since the concentration of  $Fe^{3+}$  and  $CO_2$  cannot be measured under normal field conditions, these concentration terms were replaced with ‘assimilative capacity terms’ (for iron reduction and methanogenesis) defined as:

$$C^{Fe^{3+}} = C_{\max}^{Fe^{2+}} - C^{Fe^{2+}} \quad (218)$$

$$C^{MC} = C^{CO_2} = C^{CH_4, \max} - C^{CH_4} \quad (219)$$

where  $C_{\max}^{Fe^{2+}}$  and  $C^{CH_4, \max}$  are the maximum levels (or expected levels) of  $Fe^{2+}$  and  $CH_4$ , respectively, measured in the field, and represent the aquifer’s total capacity for iron reduction and methanogenesis.

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

$$\left\{ \begin{array}{l} \frac{dC^{O_2}}{dt} = R^{O_2} \\ \frac{dC^{NO_3}}{dt} = R^{NO_3} \\ \frac{dC^{Fe^{2+}}}{dt} = R^{Fe^{2+}} \\ \frac{dC^{SO_4}}{dt} = R^{SO_4} \\ \frac{dC^{CH_4}}{dt} = R^{CH_4} \end{array} \right. \quad (220)$$

It should be noted that the kinetic model described above assumes that degradation reactions occur only in the aqueous phase, which is a conservative assumption. Model 3 has been coded in Subroutine MODEL3.

#### E) Model 4: Rate-Limited Sorption Reactions

This model has been mentioned in MODEL0 and can be easily achieved by choosing the corresponding option for Rate-Limited Sorption Reactions model in MODEL0. Model 4 has been coded in Subroutine MODEL4.

#### F) Model 5: Double Monod Model

This model is used to simulate reaction between an electron donor and an electron acceptor mediated by actively growing bacteria cells living in both aqueous and soil phases. The fate and transport of an electron donor in a multi-dimensional saturated porous media can be written as:

$$\frac{\partial(nC^D)}{\partial t} + \frac{\partial(nu_i C^D)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^D}{\partial X_j}) - \lambda^D nC^D + q_s C_s^D + R^D \quad (221)$$

where

$$R^D = -\mu_m \left( C^X + \frac{\rho \tilde{C}^X}{n} \right) \left( \frac{C^D}{K_D + C^D} \right) \left( \frac{C^A}{K_A + C^A} \right)$$

- $C^D$  is the electron donor concentration in the aqueous phase
- $C^A$  is the electron acceptor concentration in the aqueous phase
- $C^X$  is the aqueous phase bacterial cell concentration
- $\tilde{C}^X$  is the soil-phase cell concentration
- $K_D$  is the half saturation coefficient for electron donor
- $K_A$  is the half saturation coefficient for electron acceptor

$\mu_m$  is the contaminant utilization rate

The fate and transport of an electron acceptor in a multi-dimensional saturated porous media can be written as:

$$\frac{\partial(nC^A)}{\partial t} + \frac{\partial(nu_i C^A)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^A}{\partial X_j}) - \lambda^A nC^A + q_s C_s^A + R^A \quad (222)$$

where

$$R^A = -Y_{A/D} \mu_m \left( C^X + \frac{\rho \tilde{C}^X}{n} \right) \left( \frac{C^D}{K_D + C^D} \right) \left( \frac{C^A}{K_A + C^A} \right)$$

$Y_{A/D}$  is the stoichiometric yield coefficient.

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

$$\frac{\partial(nC^X)}{\partial t} + \frac{\partial(nu_i C^X)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^X}{\partial X_j}) - \lambda^X nC^X + q_s C_s^X + R^X \quad (223)$$

where

$$R^X = -Y_{X/D} \mu_m \left( C^X + \frac{\rho \tilde{C}^X}{n} \right) \left( \frac{C^D}{K_D + C^D} \right) \left( \frac{C^A}{K_A + C^A} \right) + \frac{K_{det} \rho \tilde{C}^X}{n} - C^X (K_e + K_{att})$$

$K_{att}$  is the bacterial attachment coefficient

$K_{det}$  is the bacterial detachment coefficient

$K_e$  is the endogenous cell death or decay coefficient

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

$$\frac{d\tilde{C}^X}{dt} = \tilde{R}^X \quad (224)$$

where

$$\tilde{R}^X = Y_{X/D} \mu_m \tilde{C}^X \left( \frac{C^D}{K_D + C^D} \right) \left( \frac{C^A}{K_A + C^A} \right) + \frac{K_{att} n C^X}{\rho} - \tilde{C}^X (K_e + K_{det})$$

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

$$\begin{cases} \frac{dC^D}{dt} = R^D \\ \frac{dC^A}{dt} = R^A \\ \frac{dC^X}{dt} = R^X \\ \frac{d\tilde{C}^X}{dt} = \tilde{R}^X \end{cases} \quad (225)$$

This model describes a general double Monod model. By setting appropriate yield and kinetic constants, users can model any type of biological systems. Model 5 has been coded in Subroutine MODEL5.

### G) Model 6: Sequential Decay Reactions

This model is used to simulate reactive transport coupled by a series of sequential degradation reaction (four components only in this model). Assuming first decay kinetics, the transport and transformation of a sequential decay chain  $A \rightarrow B \rightarrow C \rightarrow D$  can be simulated by solving the following set of partial differential equations:

$$\frac{\partial(nC^A)}{\partial t} + \frac{\partial(nu_i C^A)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^A}{\partial X_j}) - \lambda^A nC^A + q_s C_s^A + R^A \quad (226)$$

$$\frac{\partial(nC^B)}{\partial t} + \frac{\partial(nu_i C^B)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^B}{\partial X_j}) - \lambda^B nC^B + q_s C_s^B + R^B \quad (227)$$

$$\frac{\partial(nC^C)}{\partial t} + \frac{\partial(nu_i C^C)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^C}{\partial X_j}) - \lambda^C nC^C + q_s C_s^C + R^C \quad (228)$$

$$\frac{\partial(nC^D)}{\partial t} + \frac{\partial(nu_i C^D)}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^D}{\partial X_j}) - \lambda^D nC^D + q_s C_s^D + R^D \quad (229)$$

where  $C^A$ ,  $C^B$ ,  $C^C$  and  $C^D$  represent specie concentrations and

$$R^A = -K_A C^A$$

$$R^B = -K_B C^B + Y_{B/A} K_A C^A$$

$$R^C = -K_C C^C + Y_{C/B} K_B C^B$$

$$R^D = -K_D C^D + Y_{D/C} K_C C^C$$

$$Y_{B/A} \quad \text{specie B yield}$$

$Y_{C/B}$	specie C yield
$Y_{D/C}$	specie D yield
$K_A$	specie A first-order degradation rate
$K_B$	specie B first-order degradation rate
$K_C$	specie C first-order degradation rate
$K_D$	specie D first-order degradation rate

Note that all decay reactions in this model are assumed to occur only in the aqueous phase, which is a conservative assumption.

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

$$\begin{cases} \frac{dC^A}{dt} = R^A \\ \frac{dC^B}{dt} = R^B \\ \frac{dC^C}{dt} = R^C \\ \frac{dC^D}{dt} = R^D \end{cases} \quad (230)$$

This model provides a general description for any 4-component sequential decay chain. Model 6 has been coded in Subroutine MODEL6.

#### H) Model 7: Aerobic-Anaerobic Model for PCE-TCE Degradation

This model is used to simulate degradation of PCE/TCE and their degradation products via both aerobic and anaerobic pathways. Assuming first decay kinetics, the transport and transformation of PCE, TCE, DCE, VC ETH and Cl can be simulated by solving the following set of partial differential equations:

$$\frac{\partial(nC^{PCE})}{\partial t} + \frac{\partial(nu_i C^{PCE})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{PCE}}{\partial X_j}) - \lambda^{PCE} nC^{PCE} + q_s C_s^{PCE} + R^{PCE} \quad (231)$$

$$\frac{\partial(nC^{TCE})}{\partial t} + \frac{\partial(nu_i C^{TCE})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{TCE}}{\partial X_j}) - \lambda^{TCE} nC^{TCE} + q_s C_s^{TCE} + R^{TCE} \quad (232)$$

$$\frac{\partial(nC^{DCE})}{\partial t} + \frac{\partial(nu_i C^{DCE})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{DCE}}{\partial X_j}) - \lambda^{DCE} nC^{DCE} + q_s C_s^{DCE} + R^{DCE} \quad (233)$$

$$\frac{\partial(nC^{VC})}{\partial t} + \frac{\partial(nu_i C^{VC})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{VC}}{\partial X_j}) - \lambda^{VC} nC^{VC} + q_s C_s^{VC} + R^{VC} \quad (234)$$

$$\frac{\partial(nC^{ETH})}{\partial t} + \frac{\partial(nu_i C^{ETH})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{ETH}}{\partial X_j}) - \lambda^{ETH} nC^{ETH} + q_s C_s^{ETH} + R^{ETH} \quad (235)$$

$$\frac{\partial(nC^{Cl})}{\partial t} + \frac{\partial(nu_i C^{Cl})}{\partial X_i} = \frac{\partial}{\partial X_i} (nD_{ij} \frac{\partial C^{Cl}}{\partial X_j}) - \lambda^{Cl} nC^{Cl} + q_s C_s^{Cl} + R^{Cl} \quad (236)$$

where

$$\begin{aligned} C^{PCE} & \text{contaminant concentration of PCE} \\ C^{TCE} & \text{contaminant concentration of TCE} \\ C^{DCE} & \text{contaminant concentration of DCE} \\ C^{VC} & \text{contaminant concentration of VC} \\ C^{ETH} & \text{contaminant concentration of ETH} \\ C^{Cl} & \text{contaminant concentration of Cl} \\ R^{PCE} & = -K_P C^{PCE} \\ R^{TCE} & = Y_{T/P} K_P C^{PCE} - (K_{T1} - K_{T2}) C^{TCE} \\ R^{DCE} & = Y_{D/T} K_{T1} C^{TCE} - (K_{D1} - K_{D2}) C^{DCE} \\ R^{VC} & = Y_{V/D} K_{D1} C^{DCE} - (K_{V1} - K_{V2}) C^{VC} \\ R^{ETH} & = Y_{E/V} K_{V1} C^{VC} - (K_{E1} - K_{E2}) C^{ETH} \\ R^{Cl} & = Y1_{C/P} K_P C^{PCE} + Y1_{C/T} K_{T1} C^{TCE} + Y1_{C/D} K_{D1} C^{DCE} + Y1_{C/V} K_{V1} C^{VC} \\ & \quad + Y2_{C/T} K_{T2} C^{TCE} + Y2_{C/D} K_{D2} C^{DCE} + Y2_{C/V} K_{V2} C^{VC} \end{aligned}$$

$K_P$ ,  $K_{T1}$ ,  $K_{D1}$ ,  $K_{V1}$  and  $K_{E1}$  are first-order anaerobic degradation rates;  $K_{T2}$ ,  $K_{D2}$ ,  $K_{V2}$  and  $K_{E2}$  are first-order aerobic degradation rates;  $Y_{T/P}$ ,  $Y_{D/T}$ ,  $Y_{V/D}$  and  $Y_{E/V}$  are chlorinated compound yields under anaerobic reductive dechlorination conditions;  $Y1_{C/P}$ ,  $Y1_{C/T}$ ,  $Y1_{C/D}$  and  $Y_{C/V}$  are yield values for chloride under anaerobic conditions; and  $Y2_{C/T}$ ,  $Y2_{C/D}$  and  $Y2_{C/V}$  are yield values for chloride under aerobic conditions.

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

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

$$\left\{ \begin{array}{l} \frac{dC^{PCE}}{dt} = R^{PCE} \\ \frac{dC^{TCE}}{dt} = R^{TCE} \\ \frac{dC^{DCE}}{dt} = R^{DCE} \\ \frac{dC^{VC}}{dt} = R^{VC} \\ \frac{dC^{ETH}}{dt} = R^{ETH} \\ \frac{dC^{CI}}{dt} = R^{CI} \end{array} \right. \quad (237)$$

Model 7 has been coded in Subroutine MODEL7.

### 3.3.3 Linearized Approach to Solving Chemical Reaction Equations

From above sections, the chemical reaction equations can be generally written as a set of ordinary differential equations:

$$\begin{cases} \frac{dC^1}{dt} = R^1(C^1, C^2, \dots, C^i, \dots, C^n) \\ \frac{dC^2}{dt} = R^2(C^1, C^2, \dots, C^i, \dots, C^n) \\ \vdots \\ \frac{dC^i}{dt} = R^i(C^1, C^2, \dots, C^i, \dots, C^n) \\ \vdots \\ \frac{dC^n}{dt} = R^n(C^1, C^2, \dots, C^i, \dots, C^n) \end{cases} \quad (238)$$

As mentioned in Model 3 and Model 5, Eq(238) may become non-linear differential equations. In IGW 3-model, a linearized approach to solve Eq(238) has been proposed. In this method, the rate terms,  $R^i$ , was initially linearized as:

$$R^i(C^1, C^2, \dots, C^i, \dots, C^n) = \sum_{j=1}^n A^j C^j + B^i \quad (239)$$

where  $A^j$  and  $B^i$  could be still functions of  $C^1, C^2, \dots, C^i, \dots, C^n$ , and those concentrations in  $A^j$  and  $B^i$  then are assumed to be known or equal to their corresponding values at previous time step. Therefore, Eq(238) becomes a set of linear ordinary differential equations after substituting Eq(239) into Eq(238):

$$\begin{cases} \frac{dC^1}{dt} = \sum_{j=1}^n A^j C^j + B^1 \\ \frac{dC^2}{dt} = \sum_{j=1}^n A^j C^j + B^2 \\ \vdots \\ \frac{dC^i}{dt} = \sum_{j=1}^n A^j C^j + B^i \\ \vdots \\ \frac{dC^n}{dt} = \sum_{j=1}^n A^j C^j + B^n \end{cases} \quad (240)$$

An implicit finite difference scheme can be always applied to Eq(240).

## 3.4 Monte Carlo Simulation in Three Dimensional Model

### 3.4.1 Random Field Generator

Sequential Gaussian Simulation adopted from GSLIB was immigrated into IGW 3-D Monte Carlo Simulation. SGS can handle not only unconditional simulations but also conditional simulations when there is data available.

In IGW 3-D model, a mutipl-volume based random field generator has been coded based on the GSLIB's Sequential Gaussian Simulation code. It was coded in Subroutine RANDOM\_FIELD3D in the Visual Fortran source code.

Note that GSLIB's Sequential Gaussian Simulation code implies that the final fields' values will be back-transformed to the original data values if a normal scores transform was performed.

More detail about this generator can be found in GSLIB's manual.

### 3.4.2 Value at Monitoring Well with Screen Thickness

Calculation of the point-based and field-based statistics in 3-D model is similar to that in 2-D model, but with a few subtle differences in the sampling values at monitoring wells. In 3-D model, monitoring well is considered to have a certain screen thickness. This leads that value at a monitoring well should be evaluated in term of averaging within the screen limit (AB in Figure 27).

In IGW 3-D model, trapezoidal rule was used to calculate the distribution area (formed by ABCD in Figure 27 ) within the screen limit  $S_{AB}$ , then the value was evaluated as below

$$V_{well} = \frac{\int_A^B V(Z)dZ}{L_{AB}} = \frac{S_{AB}}{L_{AB}} \quad (241)$$

A simply trilinear interpolation scheme as described in Eq(173) has been used to calculate those non-nodal values that are necessary for obtaining  $S_{AB}$  . The steps to obtain  $V_{well}$  can be listed as below:

- (1) dividing screen limit AB into M intervals according to the desirable integrating accuracy;
- (2) using Eq(173) to calculate values at the M+1 integrating points;
- (3) employing the trapezoidal rule to obtain  $S_{AB}$ ;
- (4) using Eq(241) to obtain the value  $V_{well}$  as that at monitoring well with screen thickness.

This process has been coded in Subroutine Z1Z2LINEAR3D in the source code.

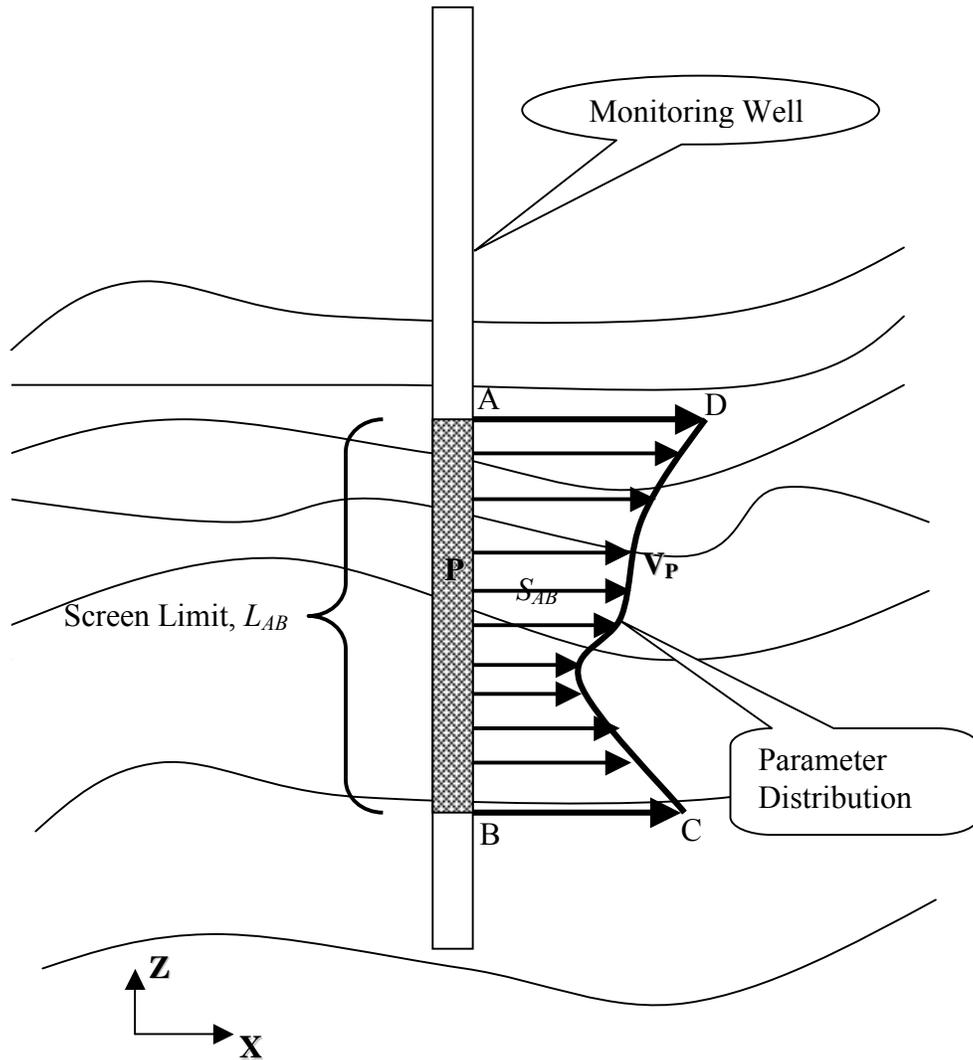


Figure 27 Monitoring Well in 3-D Model

**3.4.3 Numerical Solution Procedure Flow Chart**

The sequence of operations for IGW 3-D Monte Carlo Simulation is the same as that in 2-D model and illustrated in the following flow diagram (The whole operations will be organized by using Visual Basic Language) :

