

# **SVCHEM SVENVIRO**

**1D/2D/3D FINITE ELEMENT  
CONTAMINANT TRANSPORT MODELING**

## **Theory Manual**

**Written by:  
The Bentley Systems Team**

**Last Updated: Tuesday, June 23, 2020**

**Bentley Systems Incorporated.**

## **COPYRIGHT NOTICE**

**Copyright © 2020, Bentley Systems, Incorporated. All Rights Reserved.**

*Including software, file formats, and audiovisual displays; may only be used pursuant to applicable software license agreement; contains confidential and proprietary information of Bentley Systems, Incorporated and/or third parties which is protected by copyright and trade secret law and may not be provided or otherwise made available without proper authorization.*

## **TRADEMARK NOTICE**

*Bentley, "B" Bentley logo, SoilVision.com, SoilVision logo, and SOILVISION, SVSLOPE, SVOFFICE, SVOFFICE 5/GE, SVOFFICE 5/GT, SVOFFICE 5/WR, SVSOILS, SVFLUX, SVSOLID, SVCHEM, SVAIR, SVHEAT, SVSEISMIC and SVDESIGNER are either registered or unregistered trademarks or service marks of Bentley Systems, Incorporated. All other marks are the property of their respective owners.*

---

<b>1</b>	<b>INTRODUCTION</b> .....	<b>4</b>
<b>2</b>	<b>THEORY</b> .....	<b>5</b>
2.1	GOVERNING EQUATIONS .....	5
2.1.1	<i>Groundwater Seepage</i> .....	5
2.1.2	<i>Hydrodynamic Dispersion</i> .....	6
2.1.3	<i>Advection</i> .....	9
2.1.4	<i>Sorption</i> .....	10
2.1.5	<i>Decay</i> .....	11
2.1.6	<i>Density-Dependent</i> .....	12
2.2	GOVERNING PARTIAL DIFFERENTIAL EQUATION.....	12
2.3	GOVERNING EQUATION FOR GAS DIFFUSION.....	13
2.3.1	<i>Effective gas diffusion coefficient</i> .....	13
2.3.2	<i>Gas decay</i> .....	14
2.4	BOUNDARY CONDITIONS .....	15
2.4.1	<i>Dirichlet (first-type) boundary condition</i> .....	15
2.4.2	<i>Neumann (second-type) boundary condition</i> .....	15
2.4.3	<i>Cauchy (third-type) boundary condition</i> .....	15
2.5	FLUX SECTIONS .....	16
<b>3</b>	<b>NUMERICAL IMPLEMENTATION</b> .....	<b>17</b>
3.1	EQUATION ANALYZER .....	17
3.2	MESH GENERATION.....	17
3.3	MESH REFINEMENT .....	18
3.4	INTERPOLATION ORDER.....	18
3.5	TIME STEPPING .....	18
3.5.1	<i>Time Integration</i> .....	18
3.5.2	<i>Time step control</i> .....	19
<b>4</b>	<b>REFERENCES</b> .....	<b>21</b>

# 1 INTRODUCTION

Contaminant transport problems in geotechnical and geo-environmental engineering involve the solution of a partial differential equation referred to as a PDE. The PDE must be solved for all "finite elements" which when combined form a "continuum" (or the geometry of the problem). The theory of contaminant transport expressed in mathematical form embraces the physical behavior of the material (e.g., soil) and the conservative laws of physics (i.e., conservation of energy). The physical behavior of many materials, (particularly unsaturated soils), is nonlinear and as a consequence, the PDE becomes nonlinear in character. It is well known that the solution of nonlinear PDEs can present a challenge to the numerical modeler.

The purpose of the theory manual is to provide the user with details regarding the theoretical formulation of the PDE as well as the numerical method used in the solution. The intent of the theory manual is not to provide an exhaustive summary of all theories associated with contaminant flow. Rather, the intent is to clearly describe details of the theory used in the SVCHEM software.

The SoilVision finite element software packages utilize the FlexPDE generic finite element solver to solve the partial differential equation for contaminant transport. The FlexPDE solver algorithm has implemented cutting-edge numerical solution techniques that can accommodate linear and highly nonlinear PDEs. The solution technique utilizes automatic mathematically designed mesh generation as well as automatic mesh refinement. The application of these advanced numerical techniques is particularly valuable in solving highly nonlinear and complex problems. Most commonly it is the unsaturated soil portion of the soil continuum that brings in nonlinear soil behavior. The advanced FlexPDE solver makes it possible to obtain converged and accurate solutions for many problems that were previously unsolvable.

The primary attributes of the SoilVision solution process are as follows:

- Fully automatic mesh generation,
- Fully automatic mesh refinement based on any model variable,
- Fully implicit approach in the solver, which provides for a robust solution of difficult models with convergence issues,
- Contaminant mass-balance tracking,
- Axisymmetric and plan analysis formulations,
- 3, 6, or 9-noded triangles as elements for 2D analysis and 4, 10, or 20-noded tetrahedrons in 3D elements,
- Adaptive time-stepping with automatic generation and control of time steps,
- Newton-Raphson convergence iteration schemes, and
- Use matrix preconditioning in conjugate-gradient solutions. The default preconditioner is the diagonal-block inverse matrix.

## 2 THEORY

This chapter will present the governing partial differential equation and discuss the theory and formulation of each contaminant transport process including dispersion, advection, sorption, and decay.

### 2.1 GOVERNING EQUATIONS

The governing equations for each contaminant transport process are presented below.

#### 2.1.1 Groundwater Seepage

The starting point for each contaminant transport model involves the solution for the average linear seepage velocities ( $\bar{v}$ ). SVCHEM requires that you perform this analysis in SVFLUX or as a simplification provide the average linear seepage velocities as constants. SVFLUX can be instructed to generate an advection gradient file that contains the average linear seepage velocities.

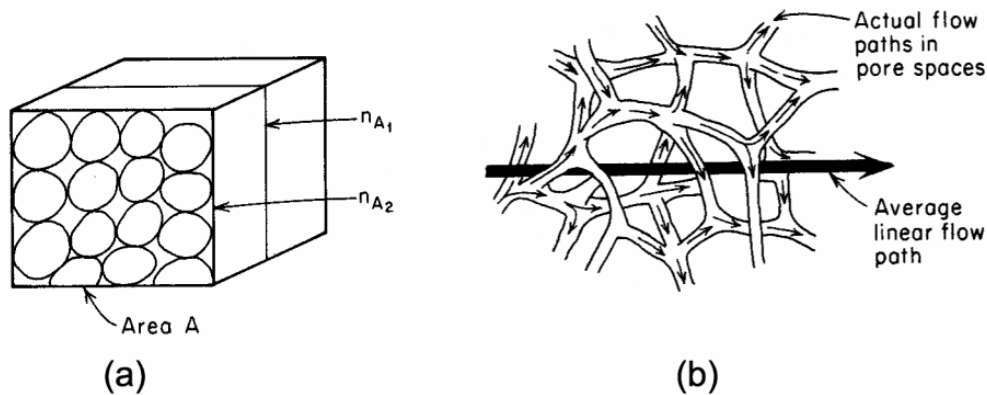


Figure 1 (a) areal porosity (b) average linear velocity (Freeze and Cherry, 1979)

In order to calculate the average linear velocity one must first understand volumetric porosity. Figure 1 (a) illustrates a column of soil. It can be seen that the total area ( $A_t$ ) of any given slice through this column of soil is made up of both area of voids ( $A_v$ ) and area of solids ( $A_s$ ). The areal porosity,  $n_A$ , is  $n_A = A_v / A_t$ . Various cross-sections within the given unit volume may exhibit differing areal porosities  $n_{A1}$ ,  $n_{A2}$ . The porosity,  $n$ , is an average of the various possible areal porosities,  $n_{A1}$  (Freeze and Cherry, 1979). At any one cross-section through the soil column, the specific discharge, or Darcian velocity, is defined by Equation [1] as,

$$v = \frac{Q}{A_t} \tag{1}$$

where:

- $v$  = Darcian velocity, or specific discharge,  $L T^{-1}$
- $Q$  = total flow,  $L^3 T^{-1}$ , and
- $A_t$  = cross-sectional area,  $L^2$ .

In this case the specific discharge is divided by the full cross-sectional area inclusive of voids and solids. In reality, water can only move through the area occupied by voids. Multiplying the total area in Equation [1] by the porosity, we obtain the seepage velocity for the area through which water actually flows,  $v_s = Q/n_{A1}A$ . For the various sections  $A_1, A_2, \dots$  illustrated in Figure 1, we can define  $v_{s1}, v_{s2}, \dots$ . If we specify as the average, it follows that,

$$\bar{v} = \frac{Q}{nA} \tag{2}$$

where:

- $\bar{v}$  = average linear velocity,  $L T^{-1}$ , and
- $n$  = porosity.

In unsaturated models, Equation [2] becomes,

$$\bar{v} = \frac{Q}{\theta A} \quad [3]$$

where:

- $\bar{v}$  = average linear velocity,  $L T^{-1}$ , and
- $\theta$  = volumetric water content.

It should be noted that  $\bar{v}$  is a macroscopic quantity and does not represent the actual microscopic velocity of the water traveling through each individual pore space. As illustrated in Figure 1 (b), the average linear flow path represents a much shorter distance than the non-linear pore spaces and as such the average linear velocity is generally slower than the actual flow velocity. The true, microscopic velocities that exist in the pore channels are seldom of interest. This fact is fortunate because the true microscopic velocities are largely indeterminate (Freeze and Cherry, 1979). SVCHEM is formulated to use the average linear velocities calculated in SVFLUX for all calculations involving groundwater seepage velocities.

### 2.1.2 Hydrodynamic Dispersion

Hydrodynamic dispersion is the combined effect of molecular diffusion and mechanical dispersion.

- **Solute concentration**

Solute concentration,  $C$ , is the mass per unit volume of solution in a porous media. For example in a groundwater application,  $C$  may be defined as the mass of solute per volume of water content. In the SVCHEM software, the unit of  $C$  is in  $g/m^3$  with the metric system, and it is in  $pcf$  with the Imperial system.

- **Molecular Diffusion**

When placed in water a solute will move from an area of high concentration to an area of lower concentration. This mixing due to concentration gradients is known as molecular diffusion. This process can be described according to Fick's first law, in one dimension, for systems where the concentration gradient changes with time as:

$$\frac{\partial(\theta c)}{\partial t} = \frac{\partial}{\partial x} \left( \theta D_d \frac{\partial c}{\partial x} \right) \quad [4]$$

where:

- $\frac{\partial(\theta c)}{\partial t}$  = change in concentration with change in time,  $M L^{-3} T^{-1}$ ,
- $\theta$  = volumetric water content,  $L^{-3}/L^{-3}$ ,
- $D_d$  = diffusion coefficient,  $L^2 T^{-1}$ , and
- $C$  = solute concentration in a liquid phase,  $M L^{-3}$ .

A negative  $D_d$  implies flow is from high concentration to low concentration. It is like Darcy's law,  $Q = -ki A$ , where the negative sign means that flow is from a high head potential to a low head potential (SVFLUX Theory Manual).

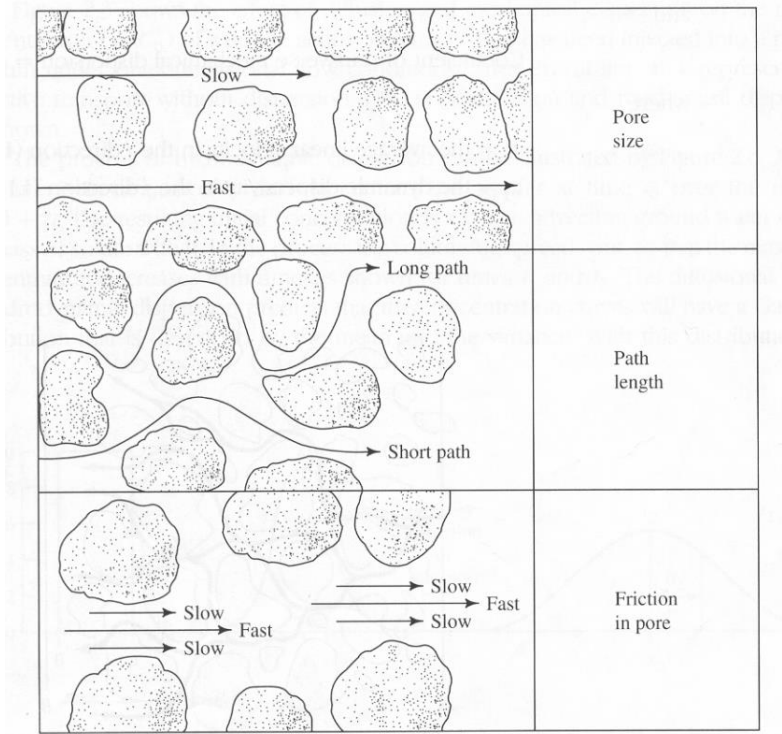
A factor that decreases the effects of diffusion is tortuosity ( $\omega$ ). The diffusing chemical must follow irregular flow paths through the soil causing the diffusion process to slow down. Tortuosity is a measure of the irregularity of the flow path of a molecule of water. Simply multiplying  $D_d$  by  $\omega$ , which is always less than 1.0, gives effective molecular diffusion,  $D^* = \omega D_d$ . To include molecular diffusion, SVCHEM requires  $D^*$  and assumes that you have scaled this number to account for tortuosity. Molecular diffusion is generally secondary and negligible, compared with the effects of mechanical dispersion, and only becomes important when groundwater velocity is very low (Zheng and Wang, 1999).

For the saturated zone if the volumetric water content,  $\theta$ , is constant, the variable of  $\theta$  in equation [4] can be removed such that

$$\frac{\partial(c)}{\partial t} = \frac{\partial}{\partial x} \left( D_d \frac{\partial c}{\partial x} \right) \quad [5]$$

- **Mechanical Dispersion**

Mechanical dispersion is the mixing that occurs from one of three processes. On a macroscopic scale, and in a domain including a sufficient volume that the effects of individual pores are averaged. There are three processes that contribute to mechanical dispersion (Bear, 1972): 1) as fluid moves through the pores, it will move faster in the center of the pores than along the edges. 2) Some of the fluid particles will travel along longer flow paths in the porous media than other particles to go the same linear distance. 3) Some pores are larger than others, which allow the fluid flowing through these pores to move faster (Fetter, 1993). Each process is illustrated in Figure 2.



**Figure 2 Processes contributing to mechanical dispersion (Fetter, 1993)**

Assuming that mechanical dispersion can be described by Fick’s law for diffusion and that the amount of mechanical dispersion is a function of the average linear velocity (Fetter, 1993), SVCHEM implements a coefficient of dispersion termed dispersivity,  $\alpha$ . Mixing that occurs along the direction of the flow path is called longitudinal dispersivity  $\alpha_L$ , while mixing that occurs in directions normal to the flow path is called transverse dispersivity  $\alpha_T$  (Fetter, 1993).

For isotropic conditions when the velocity vector is aligned with one of the coordinate axes, the use of  $\alpha_L$  and  $\alpha_T$  will suffice. This is not valid for anisotropic porous media, which requires five independent dispersivities (Bear, 1972). However, it is generally not feasible to obtain all five dispersivities in the field and as a practical alternative; the usual practice in transport modeling is to assume that the isotropic dispersion coefficient is also applicable to anisotropic porous media (Zheng and Wang, 1999). The notation used by SVCHEM is simply  $\alpha_L$ , and  $\alpha_T$ .

- **Hydrodynamic Dispersion**

Adding the effects of molecular diffusion and dispersivity results in hydrodynamic dispersion:

$$D = \alpha \bar{v} + D^* \tag{6}$$

where:

- $D$  = hydrodynamic dispersion,  $L^2 T$ ,
- $\alpha$  = dispersivity,  $L$ ,
- $\bar{v}$  = average linear velocity,  $L T^{-1}$ , and
- $D^*$  = effective diffusion,  $L^2 T^{-1}$ .

For two-dimensional flow the hydrodynamic dispersion,  $D$  in matrix form is (Bear, 1979):

$$\begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{7}$$

where:

$$\begin{aligned}
 D_{xx} &= \alpha_L \frac{v_x^2}{v} + \alpha_T \frac{v_y^2}{v} + D^* \\
 D_{yy} &= \alpha_T \frac{v_x^2}{v} + \alpha_L \frac{v_y^2}{v} + D^* \\
 D_{xy} &= D_{yx} = (\alpha_L - \alpha_T) \frac{|v_x v_y|}{v} \\
 v &= \sqrt{v_x^2 + v_y^2}
 \end{aligned}
 \tag{8}$$

For three-dimensional flow the hydrodynamic dispersion, D in matrix form is (Bear, 1979):

$$\begin{bmatrix}
 D_{xx} & D_{xy} & D_{xz} \\
 D_{yx} & D_{yy} & D_{yz} \\
 D_{zx} & D_{zy} & D_{zz}
 \end{bmatrix}
 \tag{9}$$

where:

$$\begin{aligned}
 D_{xx} &= \alpha_L \frac{v_x^2}{|v|} + \alpha_T \frac{v_y^2}{|v|} + \alpha_T \frac{v_z^2}{|v|} + D^* \\
 D_{yy} &= \alpha_L \frac{v_y^2}{|v|} + \alpha_T \frac{v_x^2}{|v|} + \alpha_T \frac{v_z^2}{|v|} + D^* \\
 D_{zz} &= \alpha_L \frac{v_z^2}{|v|} + \alpha_T \frac{v_x^2}{|v|} + \alpha_T \frac{v_y^2}{|v|} + D^* \\
 D_{xy} &= D_{yx} = (\alpha_L - \alpha_T) \frac{|v_x v_y|}{|v|} \\
 D_{xz} &= D_{zx} = (\alpha_L - \alpha_T) \frac{|v_x v_z|}{|v|} \\
 D_{yz} &= D_{zy} = (\alpha_L - \alpha_T) \frac{|v_y v_z|}{|v|} \\
 |v| &= \sqrt{v_x^2 + v_y^2 + v_z^2}
 \end{aligned}
 \tag{10}$$

where:

- $D_{xx} D_{yy} D_{zz}$  = principle components of the dispersion tensor,  $L^2 T^{-1}$ ,
- $D_{xy} D_{xz} D_{yx}$   
 $D_{yz} D_{zx} D_{zy}$  = cross-terms of the dispersion tensor,  $L^2 T^{-1}$ ,
- $\alpha_L$  = longitudinal dispersivity,  $L$ ,
- $\alpha_T$  = transverse dispersivity,  $L$ ,
- $v_x v_y v_z$  = components of the velocity vector along the  $x$ ,  $y$ , and  $z$  axes,  $L T^{-1}$ , and
- $D^*$  = effective molecular diffusion coefficient,  $L^2 T^{-1}$ .

The coefficient of diffusion,  $D^*$  is generally a function of volumetric water content. SVACHEM allows the option of specifying  $D^*$  as a function of volumetric water content. This function may be defined by a table of values. Linear interpolation is used to determine values between tabled points.



**2.1.3 Advection**

Advection is the process that describes the movement of dissolved chemical traveling at the same velocity as the groundwater. The one-dimensional unsaturated advective transport equation is:

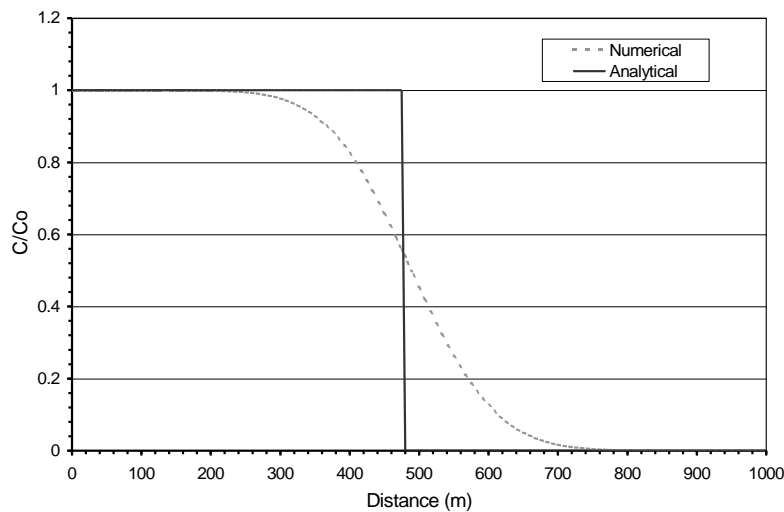
$$\frac{\partial c}{\partial t} = - \frac{\partial(qc)}{\partial x} = - \frac{\partial(\theta \bar{v}_x c)}{\partial x} \quad [ 11 ]$$

where:

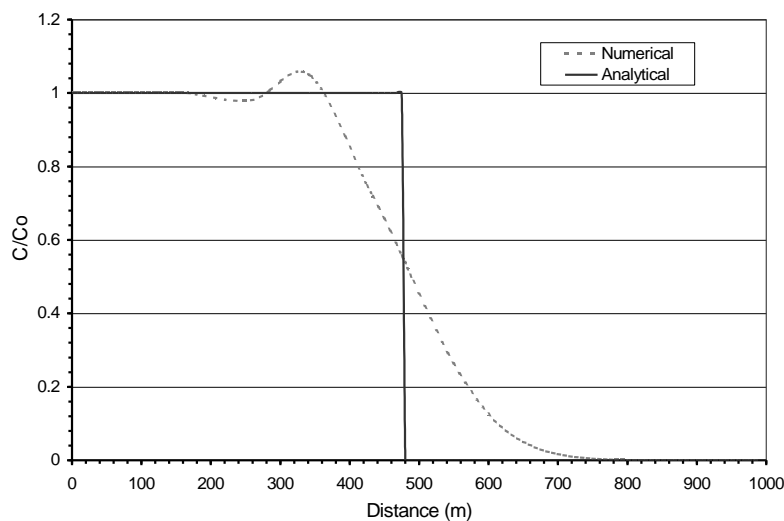
- $\theta$  = volumetric water content,
- $q$  = volumetric water flux,  $L T^{-1}$ , and
- $\bar{v}$  = average linear velocity,  $L T^{-1}$ .

The average linear velocity information required to solve advection is imported from a steady state or transient state SVFLUX analysis via a .GRD file. In both cases the .GRD file is written at the end of the model solution. In order for SVFLUX to output average linear velocity the user is required to supply a soil-water characteristic curve, for each soil in the seepage model. Without a soil-water characteristic curve SVFLUX does not have adequate information to calculate the average linear velocities and will instead export Darcian velocities. Average linear velocities are the Darcian velocities divided by the volumetric water content (see Groundwater Seepage section).

For advection-dominated models, the solution of the transport equation is often plagued to some degree by numerical dispersion and artificial oscillation (Zheng and Wang, 1999). These processes are discussed below:



**Figure 3 Numerical dispersion which appears as actual physical dispersion in a numerical model**



**Figure 4 Artificial oscillation in a numerical solution which causes the concentration to appear to exceed 1.0**

Numerical dispersion shown in Figure 3, has an effect similar to that of physical dispersion, but is caused by truncation errors. Artificial oscillation shown in Figure 4 is typical of some higher-order schemes designed to eliminate numerical dispersion which tends to become more severe as the concentration front becomes sharper (Zheng and Wang, 1999). SVCHEM uses an Eulerian method to solve the advection dispersion equation. For advection dominated models that exist in many field

conditions, an Eulerian method may be susceptible to excessive numerical dispersion or artificial oscillation. To overcome these problems, restrictively small grid spacing and time-steps may be required (Zheng and Wang, 1999). The SVCHEM solver uses both automated mesh refinement and automatic time step refinement to overcome these two numerical problems. For further information see the verification example in the SVCHEM Verification Manual.

#### 2.1.4 Sorption

Sorption is the process by which chemicals dissolve in the groundwater and cling to a solid surface thus decreasing the concentration of the solute. If the sorptive process is slow compared with the rate of fluid flow in the porous media, the solute may not come to equilibrium with the sorbed phase, and a kinetic sorption model will need to be used to describe the process (Fetter, 1993). It is generally assumed that equilibrium conditions exist between the aqueous-phase and solid-phase concentrations and that the sorption reaction is fast enough, relative to groundwater velocity, to be treated as instantaneous (Zheng and Wang, 1999). SVCHEM accepts this assumption and does not provide a kinetic sorption model. Rather, four sorption isotherm models are used including: Linear, Freundlich, Langmuir, and a User-Defined Model. A sorption isotherm is a relationship between the sorbed and dissolved concentrations of the solute at a constant temperature. Isotherms are usually incorporated into the model with a retardation factor. Each isotherm and its corresponding retardation factor are presented below.

- **Linear Sorption Isotherm**

The linear sorption isotherm models the relationship between the concentration of sorbed and dissolved solute as a straight line. As presented in Fetter (1993), the linear sorption isotherm is,

$$C^* = K_d C \quad [12]$$

where:

- $C^*$  = sorbed concentration in the solid particles,  $g/kg$ , i.e. the mass of solute (g) sorbed by the mass of solid (kg),
- $C$  = dissolved concentration in the solution,  $g/m^3$ , and
- $K_d$  = distribution coefficient (slope of linear isotherm),  $L^3 M^{-1}$ .

SVCHEM implements this isotherm with a retardation factor. The retardation factor as presented in Fetter (1993), is,

$$R = 1 + \frac{\rho_d}{\theta} K_d \quad [13]$$

where:

- $R$  = retardation factor,
- $\rho_d$  = bulk density,  $kg/m^3$ ,
- $\theta$  = volumetric water content,  $m^3/m^3$ , and
- $K_d$  = distribution coefficient (slope of linear isotherm),  $m^3/kg$ .

The main benefit to the linear isotherm is that mathematically, it provides a stable solution. The user should be aware that theoretically there is no upper limit on the amount of solute that could sorb to soil particles.

- **Freundlich Isotherm**

The nonlinear Freundlich isotherm is more general and is expressed as,

$$C^* = K C^N \quad [14]$$

where:

- $C^*$  = sorbed concentration in the solid particles,  $g/kg$ ,
- $C$  = dissolved concentration in solution,  $g/m^3$ ,
- $K$  = Freundlich constant,  $m^3/kg \times (m^3/g)^{N-1}$ , and
- $N$  = dimensionless constant.

The retardation factor as presented in Fetter (1993) is:

$$R = 1 + \frac{\rho_d K N C^{N-1}}{\theta} \quad [15]$$

where:

- $R$  = retardation factor,  
 $\rho_d$  = bulk density,  $kg/m^3$ ,  
 $K$  = Freundlich constant,  $m^3/kg \times (m^3/g)^N$ ,  
 $N$  = dimensionless constant, and  
 $\theta$  = volumetric water content,  $m^3/m^3$ .

If  $N$  is greater than 1, Equation [14] will lead to a spreading front, whereas if  $N$  is less than 1, the front will be self-sharpening. If  $N$  is equal to 1, the Freundlich isotherm becomes the linear isotherm (Fetter, 1993). It should also be noted that like the linear isotherm the Freundlich isotherm does not have an upper limit on the amount of solute that could be sorbed.

- **Langmuir Isotherm**

The non-linear Langmuir isotherm is expressed as,

$$C^* = \frac{\alpha\beta C}{1 + \alpha C} \quad [16]$$

where:

- $C^*$  = sorbed concentration in the solid particles,  $g/kg$ ,  
 $C$  = dissolved concentration in solution,  $g/m^3$ ,  
 $\alpha$  = Langmuir constant relating to the binding energy,  $m^3/g$ , and  
 $\beta$  = maximum amount of solute that can be sorbed by the solid,  $g/kg$ .

The retardation factor as presented in Fetter (1993), is,

$$R = 1 + \frac{\rho_d}{\theta} \left( \frac{\alpha\beta}{(1 + \alpha C)^2} \right) \quad [17]$$

where:

- $R$  = retardation factor,  
 $\rho_d$  = bulk density,  $kg/m^3$ ,  
 $\theta$  = volumetric water content,  $m^3/m^3$ ,  
 $\alpha$  = constant relating to the binding energy,  $m^3/g$ ,  
 $\beta$  = maximum amount of solute that can be sorbed by the solid,  $g/kg$ , and  
 $C$  = dissolved concentration in the solution,  $g/m^3$ .

The Langmuir isotherm is formulated to allow the user the ability to describe an upper limit on the amount of solute that can be sorbed.

- **User Defined Isotherm**

SVCHEM allows the user the ability to input their own curve to describe the sorption process. The user is required to supply a curve of concentration versus adsorption. The retardation factor for the user-defined isotherm is:

$$R = 1 + \frac{\rho_d}{\theta} \frac{\partial(C^*)}{\partial C} \quad [18]$$

where:

- $R$  = retardation factor,  
 $C^*$  = the mass of solute sorbed in solid,  $g/kg$ ,  
 $C$  = dissolved concentration in solution,  $g/m^3$ ,  
 $\rho_d$  = bulk density,  $kg/m^3$ , and  
 $Dc^*/dc$  = slope of the concentration versus adsorption curve,  $m^3/kg$ .

### 2.1.5 Decay

If radionuclides enter the groundwater system, the cation radionuclides are subjected to retardation on soil surfaces. In addition, they will undergo radioactive decay, which will reduce the concentration of radionuclides in both the dissolved and sorbed phases (Fetter, 1993). Certain biological reactions may cause half-life for the sorbed radionuclides to differ from the half-life of the dissolved radionuclides. To account for this phenomenon, SVCHEM allows the entry of two separate half-life coefficients, one for the sorbed phase and one for the dissolved phase.

$$-\lambda_1 \theta C - \lambda_2 \rho_d C^* \quad [19]$$

where:

- $\lambda_1$  = dissolved half-life,  $T^{-1}$ ,
- $\theta$  = volumetric water content, ,
- $C$  = dissolved concentration,  $L^3 M^{-1}$ ,
- $\lambda_2$  = sorbed half-life,  $T^{-1}$ ,
- $\rho_d$  = bulk density,  $M L^{-3}$ , and
- $c^*$  = sorbed concentration.

### 2.1.6 Density-Dependent

The dependence of the solute concentration on water density is approximated with the following linear expression (Boufadel et al, 1999):

$$\rho_{mw} = \rho_w (1 + \beta_c C) = \beta \rho_w \quad [20]$$

$$\beta = \frac{\rho_{mw}}{\rho_w} \quad [21]$$

where:

- $\rho_{mw}$  = density of mixture of water and solute,  $ML^{-3}$ ,
- $\rho_w$  = density of fresh water,  $ML^{-3}$ ,
- $\beta_c$  = empirical constant,  $L^3 M^{-1}$  and
- $C$  = concentration,  $ML^{-3}$ .

The  $\beta_c$  can be estimated with the expression below (Simpson 2003):

$$\beta_c = \frac{\rho_{max} - \rho_w}{\rho_w C_{max}} \quad [22]$$

where:

- $\rho_{max}$  = the maximum density of mixture of water and solute,  $ML^{-3}$ , and
- $C_{max}$  = maximum concentration at the maximum density of  $\rho_{max}$ ,  $ML^{-3}$ .

The values of  $\rho_{max}$  and  $C_{max}$  can be specified in SVCHEM model settings.

## 2.2 GOVERNING PARTIAL DIFFERENTIAL EQUATION

Combining all of the contaminant transport processes, including hydrodynamic dispersion, advection, sorption, decay, and density-dependent results in the following partial differential Equation:

$$\beta \frac{\partial}{\partial x} \left( \theta D_x \frac{\partial c}{\partial x} \right) - \theta \bar{v}_x \frac{\partial c}{\partial x} - \lambda_1 \theta c - \lambda_2 \rho_d c^* = \frac{\partial (\theta c + \rho_d c^*)}{\partial t} \quad [23]$$

$$\beta \frac{\partial}{\partial x} \left( \theta D_x \frac{\partial c}{\partial x} \right) - \theta \bar{v}_x \frac{\partial c}{\partial x} - \lambda_1 \theta c - \lambda_2 \rho_d c^* = \theta R \frac{\partial (c)}{\partial t} + c \frac{\partial (\theta)}{\partial t} \quad [24]$$

$$R = 1 + \frac{\rho_d}{\theta} \frac{\partial c^*}{\partial c} \quad [25]$$

where:

- $D_x$  = hydrodynamic dispersion in  $x$ ,  $L^2 T$ ,
- $\bar{v}_x$  = average linear velocity (intrinsic velocity) in  $x$ ,  $L T^{-1}$ ,
- $\lambda_1$  = dissolved half-life,  $T^{-1}$ ,
- $\lambda_2$  = sorbed half-life,  $T^{-1}$ ,
- $c$  = dissolved concentration,  $M L^{-3}$ ,

$c^*$	= sorbed concentration, $M L^{-3}$ , as given in equation [ 12 ], [ 14 ], and [ 16 ],
$\rho_d$	= bulk density, $M L^{-3}$ ,
$\theta$	= volumetric water content,
$R$	= retardation factor of sorption isotherm as given in equation [ 13 ], [ 15 ], and [ 17 ], and
$\beta$	= defined in equation [ 21 ].

The above equation is for one-dimensional transport. SVCHEM expands this equation to two and three-dimensions by simply adding the corresponding terms for hydrodynamic dispersion and advection in the  $y$  and  $z$  directions.

Please see the **SVFLUX theory manual** for the density-dependent-flow in seepage governing equation.

## 2.3 GOVERNING EQUATION FOR GAS DIFFUSION

The governing equation presented in above section is used for the mass transport of aqueous solute dissolved in water. This section presents the governing equation for the mass transport of gaseous solute dissolved in air of porous media such as an unsaturated soil, industrial wastes and mine tailings etc. Oxygen diffusion in a void space is one example of the gas transport in a porous media. Another example is the gas vapor diffusion in an unsaturated soil.

The gas transport through the porous media is mainly governed by molecular diffusion. There are many published research papers in the area of gas diffusion. The following theory is primarily based on the research results presented by Aachib et al (2004).

The gas flux in porous media is described by Fick's law:

$$q_g = -n_{eq} D^* \frac{\partial c_g}{\partial x} = -D_e \frac{\partial c_g}{\partial x} \quad [ 26 ]$$

where:

$q_g$	= mass flux of gas transport in $x$ , $M L^{-2} T^{-1}$ ,
$C_g$	= gaseous concentration, $ML^{-3}$ ,
$n_{eq}$	= equivalent porosity,
$D^*$	= bulk diffusion coefficient, $L^2 T^{-1}$ ,
$D_e$	= effective diffusion coefficient in $x$ , $L^2 T^{-1}$ , and
$x$	= coordinate, $L$ .

The equivalent porosity  $n_{eq}$  is used to consider the gas transport in dual phases of liquid and gas, which is defined as

$$n_{eq} = \theta_a + H\theta_w \quad [ 27 ]$$

where:

$\theta_a$	= volumetric air content (air-filled porosity),
$\theta_w$	= volumetric water content (water-filled porosity), and
$H$	= Henry's equilibrium constant which is defined by the ratio of gas concentration dissolved in water to the gas dissolved in air phase.

For oxygen, Henry's constant is approximated to 0.03 at 25 °C (Fredlund and Rahardjo, 1993).

In the case without considering the gas advection, the governing equation for gas diffusion can be described by Fick's second law, i.e.

$$D_e \frac{\partial^2 c_g}{\partial x^2} + D_e \frac{\partial^2 c_g}{\partial y^2} + D_e \frac{\partial^2 c_g}{\partial z^2} - \theta_{eq} k_r^* c_g = \frac{\partial (\theta_{eq} c_g)}{\partial t} \quad [ 28 ]$$

where:

$k_r^*$	= reaction rate coefficient of gas decay due to the chemical reaction, $T^{-1}$ .
---------	---

### 2.3.1 Effective gas diffusion coefficient

The effective gas diffusion coefficient,  $D_e$ , is related to the tortuosity through the gas and liquid phase. SVCHEM has implemented the following approaches to estimate the effective gas diffusion coefficient.

- M-Q model (Millington and Quick, 1961)  
In the assumption when the gaseous solute dissolved in liquid phase can be neglected, equation [ 29 ] can be utilized to calculate the effective diffusion coefficient:

$$D_e = D_a = \theta_a \beta_a D_{ao} \quad [ 29 ]$$

where:

- $D_a$  = effective diffusion coefficient in the gas phase,  $L^2 T^{-1}$ ,
- $\theta_a$  = volumetric air content,
- $\beta_a$  = gaseous-phased tortuosity, and
- $D_{ao}$  = gas free (unobstructed) diffusion coefficient in air,  $L^2 T^{-1}$ .

The gaseous-phased tortuosity is estimated by the M-Q approach (Millington and Quick, 1961):

$$\beta_a = \theta_a^{7/3} n^{-2} \quad [ 30 ]$$

where:

- $n$  = porosity of porous media.

- Modified C-R model (Collin and Rasmuson, 1998)

To include the gas solubility in a liquid phase, Collin and Rasmuson (1998) modified the C-R model in the following expression.

$$D_e = D_a + HD_w \quad [ 31 ]$$

$$D_w = \theta_w \beta_w D_{wo} \quad [ 32 ]$$

where:

- $D_w$  = effective diffusion coefficient component in the water phase,  $L^2 T^{-1}$ ,
- $D_a$  = is described by equation [ 29 ],
- $H$  = Henry's law constant,
- $\theta_w$  = volumetric water content,
- $\beta_w$  = aqueous phase tortuosity, and
- $D_{wo}$  = gas free (unobstructed) diffusion coefficient in water,  $L^2 T^{-1}$ .

For oxygen,  $D_{ao} = 1.8E-5 \text{ m}^2/\text{s}$  and,  $D_{wo} = 2.2E-9 \text{ m}^2/\text{s}$  (Fredlund and Rahardjo 1993).

Combining equations [ 29 ], [ 30 ], [ 31 ], and [ 32 ], Achib et al ( 2004) presented the gas diffusion coefficient with equation [ 33 ], which is implemented within the SVCHEM software:

$$D_e = \frac{1}{n^2} (D_{ao} \theta_a^{7/3} + HD_{wo} \theta_w^{7/3}) \quad [ 33 ]$$

$$P_a = 1.201 \theta_a^3 - 1.515 \theta_a^2 + 0.987 \theta_a + 3.119 \quad [ 34 ]$$

$$P_w = 1.201 \theta_w^3 - 1.515 \theta_w^2 + 0.987 \theta_w + 3.119 \quad [ 35 ]$$

where:

- $P_a, P_w$  = parameters related to tortuosity in the air and the water phase.

Achib et al (2004) suggested  $P_a$  and  $P_w$  can be approximated with a constant value in the range of 3.3 to 3.5.

### 2.3.2 Gas decay

The reaction rate coefficient of gas decay is represented by the half-time life, i.e.

$$k_r^* = \frac{\ln 2}{\lambda^*} \quad [ 36 ]$$

Or

$$k_r = \theta_{eq} k_r^* = \frac{\ln 2}{\lambda^*} \quad [ 37 ]$$

where:

$$\lambda^* = \text{half-time life of gas decay, } T.$$

## 2.4 BOUNDARY CONDITIONS

Three types of boundary condition can be applied for the mass transport.

### 2.4.1 Dirichlet (first-type) boundary condition

When the concentration at the boundary is known, the Dirichlet boundary condition can be applied, i.e.

$$c(x, y, z, t) = c_0(t) \quad [ 38 ]$$

where:

$$c_0(t) = \text{known concentration, } M/L^3.$$

#### NOTE:

In the SVCHEM software,  $C_0(t)$  can be specified as a constant value, expression, or time-dependent data grid for concentration.

### 2.4.2 Neumann (second-type) boundary condition

If the boundary is impermeable, i.e. the water flux is zero, the solute mass flux satisfies

$$\theta D_x \frac{\partial c}{\partial x} = 0 \quad [ 39 ]$$

#### NOTE:

In the SVCHEM software, Zero Flux can be used as this type of boundary condition.

### 2.4.3 Cauchy (third-type) boundary condition

The Cauchy boundary condition can be used in the boundary where the solute mass flux is required to be applied, i.e.

$$-\frac{\partial}{\partial x} \left( \theta D_x \frac{\partial c}{\partial x} \right) + qc = q_0(t) c_0(t) \quad [ 40 ]$$

where:

$$c_0(t) = \text{known concentration, } M/L^3, \text{ and}$$

$$q_0(t) = \text{known water flux, } L^3/T-L^2.$$

#### NOTE:

In the SVCHEM software, Flux Expression or Flux Data method can be used for this type of boundary condition.

The Flux boundary condition represents a generalized flux boundary condition derived from the divergence theorem. The expression may take on an explicit specification, involving only constants and coordinates, or it may be an implicit relationship involving values and derivatives of system variables. The Flux boundary condition is a generalization of the concept of a flux boundary condition. In diffusion equations, it is in fact the outward flux of the diffusing quantity (FlexPDE User's Manual, 2001).

FlexPDE uses integration by parts to reduce the order of second derivative terms in the system equations. Application of this technique over a two-dimensional computation cell produces an interior area integral term and a boundary line integral term. Forming the same integral in two adjacent computation cells produces the same boundary integral at their interface, except that the direction of integration is opposite in the two cells. If the integrals are added together to form the total integral, the shared boundary integrals cancel (FlexPDE User's Manual, 2001).

Applied to the term  $Dx(f)$ , where  $f$  is an expression containing further derivative terms, integration by parts yields Integral  $[Dx(f) dV] = \text{Integral } [f \times c \times dS]$ , where  $c$  denotes the  $x$ -component of the outward surface-normal unit vector and  $dS$  is the differential surface element (FlexPDE User's Manual, 2001). ( $y$ - and  $z$ - derivative terms are handled similarly, with  $c$  replaced by the appropriate unit-vector component).

Applied to the term  $D_{xx}(f)$ , where  $f$  denotes a scalar quantity, integration by parts yields  $\text{Integral}[D_{xx}(f) dV] = \text{Integral}[D_x(f) \times c \times dS]$ , where  $c$  denotes the  $x$ -component of the outward surface-normal unit vector and  $dS$  is the differential surface element FlexPDE (2001). ( $y$ - and  $z$ - derivative terms are handled similarly, with  $c$  replaced by the appropriate unit-vector component).

FlexPDE performs these integrations in 3 dimensions, including the volume and surface elements appropriate to the geometry. In 2D Cartesian geometry, the volume cell is extended one unit in the  $z$ -direction. This technique forms the basis of the treatment of exterior boundary conditions and interior material interface behavior in (FlexPDE User's Manual, 2001).

**NOTE:**

See the User's Manual for more a more detailed discussion on each specific system of boundary condition usage.

**NOTE:**

If statements will be of the form *IF expression THEN expression ELSE expression*. The *THEN ELSE* expression may contain nested *IF...THEN...ELSE* expressions. Each *ELSE* will bind to the nearest *IF*.

## 2.5 FLUX SECTIONS

The integration of mass flux across a given boundary accounts for both advective and dispersive components of mass flux as described in the following sections.

- **Advective Mass Flux**

The advective mass flux is calculated as the mass flux crossing the flux section carried in solution with the groundwater flow.

$$Q_A = v_i c A \quad [41]$$

where:

- $Q_A$  = advective flow,  $M T^{-1}$ ,
- $v_i$  = Darcian velocity, or specific discharge,  $L T^{-1}$ ,
- $c$  = concentration,  $M L^{-3}$ , and
- $A$  = cross-sectional area,  $L^2$ .

- **Dispersive Mass Flux**

The dispersive mass flux is a calculation of the mass flux across a flux section due to dispersive flux as defined by Fick's Law.

$$Q_D = DiA \quad [42]$$

where:

- $Q_D$  = dispersive flow,  $M T^{-1}$ ,
- $i$  = concentration gradient, and
- $A$  = cross-sectional area,  $L^2$ .

- **Total Mass Flux**

Total mass flux is simply the addition of the Mass Flux and the Dispersive Mass Flux.



### 3 NUMERICAL IMPLEMENTATION

The SVCHEM software makes use of a generic finite element solver called FlexPDE. FlexPDE has been designed and maintained by Bob Nelson of PDE Solutions Inc. over the course of the past 31 years and is the result of many years of rigorous testing and peer review. The most detailed source of documentation of the numerical schemes used in the FlexPDE software may be found in the User's Guide and Reference Manual provided with the software. The following sections provide a brief overview of some of the more significant numerical methods used in the software.

Primary advantages of the FlexPDE finite element solver are as follows:

- Fully automatic mesh generation,
- Fully automatic mesh refinement based on any model variable,
- Fully implicit approach in the solver, which provides for a robust solution of difficult models with convergence issues,
- 3, 6, or 9-noded triangles as elements for 2D analysis and 4, 10, or 20-noded tetrahedrons in 3D elements,
- Adaptive time-stepping with automatic generation and control of time steps,
- Newton-Raphson convergence iteration schemes, and
- Use matrix preconditioning in conjugate-gradient solutions. The default preconditioner is the diagonal-block inverse matrix.

#### 3.1 EQUATION ANALYZER

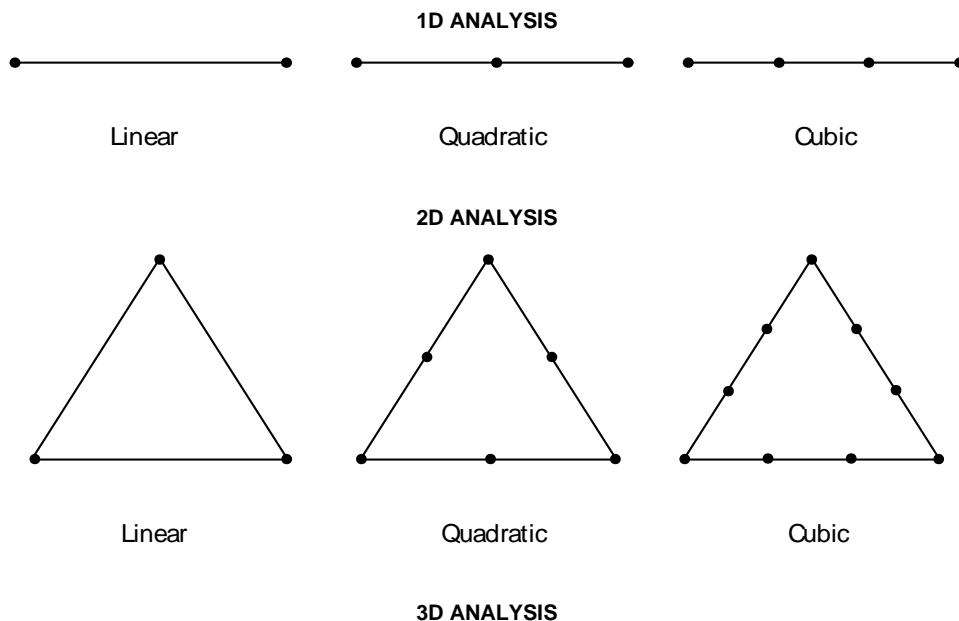
A symbolic *equation analyzer* expands defined parameters and relations, performs spatial differentiation, and symbolically applies integration by parts to reduce second order terms to create symbolic Galerkin equations. It then differentiates these equations to form the Jacobian coupling matrix.

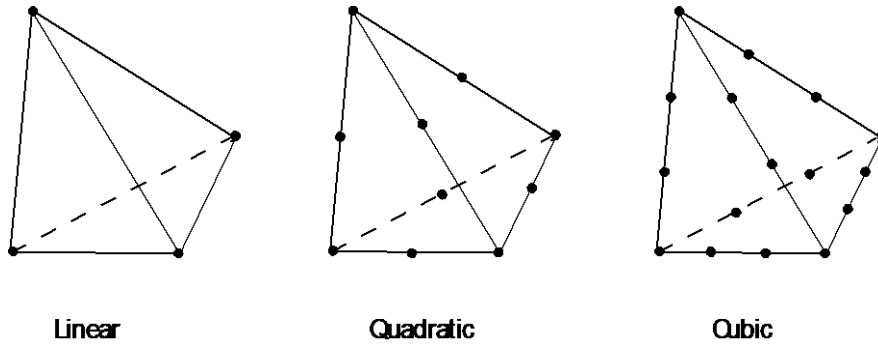
#### 3.2 MESH GENERATION

A *mesh generation* module constructs a triangular finite element mesh over an arbitrary two-dimensional model domain. In three-dimensional models, the 2D mesh is extruded into a tetrahedral mesh covering an arbitrary number of non-planar layers in the extrusion dimension. Bilinear interpolation is used in 3D models to approximate a curved surface.

FlexPDE uses an "advancing front" mesh generation method, triangular in 2D and tetrahedral in 3D (the specific algorithm is proprietary, but similar mesh generation methods are widely reported in the literature). The mesh generator allows spatially-varying node density, in order to concentrate cells in regions of structural detail.

Based on the order of the selected analysis the following types of elements can be generated.





### 3.3 MESH REFINEMENT

The FlexPDE finite element solver implements fully automatic mesh refinement. The mesh refinement may be triggered based on any model variable through a combination of the RESOLVE statement as well as the XERRLIM variable.

FlexPDE automatically adapts the computation mesh to the needs of the solution. Error estimates are formed in each cell, and cells with high error are split into two new cells, resulting in a new computation mesh and a repeat of the solution process. The reference error is user-selectable (ERRLIM). The exact algorithm is proprietary, but similar techniques are discussed in the open literature.

The error estimation process is based on the following observation. In the finite element method, each nodal value is computed by minimizing the residual of the partial differential equation, PDE, weighted by a node-specific weighting function and integrated over all cells surrounding the node. A centrally-weighted integral of the PDE residual over an individual cell provides an independent measure of the residual and provides the error estimate.

### 3.4 INTERPOLATION ORDER

Finite element basis is quadratic.

### 3.5 TIME STEPPING

Both explicit and implicit methods have been presented in literature as viable methods of solving for nodes while moving forward in time. Explicit methods use known data to "explicitly" define each nodal value at an advanced time. Such explicit methods usually suffer in accuracy and stability unless the time step is small. Implicit methods solve "implicitly" for self-consistent values at the advanced time (i.e., the simultaneous finite element spatial equations are solved at the advanced time). This improves the stability and accuracy for large time steps, at the cost of a simultaneous solution for all nodes in the mesh. Explicit methods can be more economical under the right circumstances.

The implicit generalized Gear (1971) method has been selected in SVCHEM as the most robust method for general application. The research literature supports this decision.

#### 3.5.1 Time Integration

FlexPDE uses a selectable-order Backward Difference Formulation to advance variables in time. The default is second order, but you can select linear or cubic with the TORDER selector. Linear backward difference (torder=1) is equivalent to Backward Euler integration. All the methods are Implicit. The analysis for default quadratic integration is as follows:

Assume that each variable can be approximated as a polynomial in time:

$$U(t) = p_0(t)u_0 + p_1(t)u_1 + p_2(t)u_2$$

where  $u_0$  and  $u_1$  are known values at times  $t_0$  and  $t_1$ ,  $u_2$  is the unknown value at time  $t_2$ , and  $t_0 < t_1 < t_2$ .

The coefficients  $p_0$ ,  $p_1$  and  $p_2$  are Lagrange Interpolation coefficients,

$$\begin{aligned}
 p_0(t) &= (t-t_1)(t-t_2) / ((t_1-t_0)(t_2-t_0)), \\
 p_1(t) &= -(t-t_0)(t-t_2) / ((t_1-t_0)(t_2-t_1)), \text{ and} \\
 p_2(t) &= (t-t_0)(t-t_1) / ((t_2-t_0)(t_2-t_1)).
 \end{aligned}$$

With this interpolation, we can find the time derivative of  $U(t)$

$$U'(t) = p_0'u_0 + p_1'u_1 + p_2'u_2$$

with

$$\begin{aligned} p_0'(t) &= ((t-t_1)+(t-t_2)) / ((t_1-t_0)*(t_2-t_0)), \\ p_1'(t) &= -((t-t_0)+(t-t_2)) / ((t_1-t_0)*(t_2-t_1)), \text{ and} \\ p_2'(t) &= ((t-t_0)+(t-t_1)) / ((t_2-t_0)*(t_2-t_1)). \end{aligned}$$

The time derivative at  $t_2$  is then,

$$U'(t_2) = p_0'(t_2)*u_0 + p_1'(t_2)*u_1 + p_2'(t_2)*u_2.$$

with:

$$\begin{aligned} p_0'(t_2) &= (t_2-t_1) / ((t_1-t_0)*(t_2-t_0)), \\ p_1'(t_2) &= -(t_2-t_0) / ((t_1-t_0)*(t_2-t_1)), \text{ and} \\ p_2'(t_2) &= (2*t_2-t_1-t_0) / ((t_2-t_0)*(t_2-t_1)). \end{aligned}$$

Now, if we are given an equation

$$dt(U) = F(U)$$

We can write:

$$\begin{aligned} U'(t_2) &= F(U(t_2)) \text{ as a fully implicit equation for } U_2, \text{ and} \\ U_2 &= [F(U_2) - p_0'(t_2)*U_0 - p_1'(t_2)*U_1] / p_2'(t_2). \end{aligned}$$

Solving this system requires either the solution of a linear system or a nonlinear system, depending on the form of  $F(U)$ .

Notice that the core of this solution method is to find a value of  $U_2$  such that the interpolated time derivative at the end of the step matches the specified driving term  $F(U)$  at the end of the time step. There is no time-centering issue involved with  $F(U)$ : it is always at the end of the step.

### 3.5.2 Time step control

The approximation described above implies a constant curvature  $U''$  over the interval  $(t_0, t_2)$ . When we later advance from  $t_2$  to  $t_3$ , we implicitly assume that  $U''$  is constant over the interval  $(t_1, t_3)$ , but with a different  $U''$  than the previous step. There has therefore been an inconsistent assumption about the value of  $U''$  in the overlapped interval  $(t_1, t_2)$ .

The FlexPDE time step control, simply stated, is to measure how much difference the two estimates of  $U''$  imply in  $U_3$ , and to restrict this difference to the requested ERRLIM.

The curvature over  $(t_0, t_2)$  is:

$$U''_1 = 2*U_0 / ((t_1-t_0)(t_2-t_0)) - 2*U_1 / ((t_1-t_0)(t_2-t_1)) + 2*U_2 / ((t_1-t_0)*(t_2-t_1))$$

The curvature over  $(t_1, t_3)$  is:

$$U''_2 = 2*U_1 / ((t_2-t_1)(t_3-t_1)) - 2*U_2 / ((t_2-t_1)(t_3-t_2)) + 2*U_3 / ((t_2-t_1)*(t_3-t_2))$$

The difference is:

$$E''_2 = -2*U_0 / ((t_1-t_0)(t_2-t_0)) + 2(t_3-t_0)U_1 / ((t_1-t_0)(t_2-t_1)*(t_3-t_1)) - 2*U_2*(t_3-t_0) / ((t_2-t_1)(t_3-t_2)(t_2-t_0)) + 2*U_3 / ((t_3-t_2)*(t_3-t_1))$$

Integrating over  $(t_2, t_3)$  produces an error:

$$E_3 = -(t_3-t_2)^2*U_0 / ((t_1-t_0)*(t_2-t_0)) + (t_3-t_2)^2*(t_3-t_0)U_1 / ((t_1-t_0)(t_2-t_1)*(t_3-t_1)) - (t_3-t_2)(t_3-t_0)*U_2 / ((t_2-t_1)(t_2-t_0)) + (t_3-t_2)*U_3 / (t_3-t_1)$$

The  $N$ th power of the nodal values of  $E$  is averaged, rooted and divided by the range of the variable  $U$ .  $N$  is controlled by TNORM in version 5 and ENORM in version 6.  $N=2^{\wedge}TNORM$ . The range of  $U$  is the larger of the declared Threshold or the observed range of values.

$$\begin{aligned} E_{av} &= [SUM((E_3/Range(U))^N)]^{1/N} \\ Deltat_4 &= deltat_3*\sqrt[TERRLIM/E_{av}] \end{aligned}$$

This process applies to variables  $U$  that are defined by an evolution equation  $dt(U)=F(U)$ . Steady equations are not assumed to be polynomial in time, and a different rule is applied to generate  $E_{av}$  for these variables.

In this case,

$$E3 = (U3-U2)/Changelim.$$

Moving nodes adds another criterion, which is that the cell volume cannot change by more than a prescribed percentage on any cycle. The way this is controlled is different in V6 than in V5.

## 4 REFERENCES

- Aachib, M., Mbonimpa, M., and Aubertin, M. 2004. Measurement and prediction of the oxygen diffusion coefficient in unsaturated media, with applications to soil covers. *Water, Air, and Soil Pollution* 156:163-193.
- Bear, Jacob. 1972. *Dynamics of Fluids in Porous Media*. Dover Publications Inc., New York.
- Bear, Jacob, 1979. *Hydraulics of Groundwater*. McGraw-Hill, New York.
- Boufadel, M.C., Suidan, M. T., and Venosa, A.D. 1999. A numerical model for density-and-viscosity-dependent flows in two dimensional variably saturated porous media. *Journal of Contaminant Hydrology*, 37: 1-29.
- Burnett, R. D. and Frind, E. O. 1987. An alternating direction Galerkin technique for simulation of groundwater contaminant transport in three dimensions, 2, dimensionality effects. *Water Resources. res.* 23(4), 695-705.
- Wels, C., Lefebvre, R., and Robertson, A.M. 2003. An overview of prediction and control of air flow in acid-generating waste rock dumps. 6<sup>th</sup> ICARD, Cairns, QLD, 12-18 July 2003.
- Fetter, C. W., 1994. *Applied Hydrogeology*, Third Edition. Prentice-Hall, Inc. Upper Saddle River, New Jersey.
- Fetter, C. W., 1993. *Contaminant Hydrogeology*. MacMillan Publishing Company. New York, New York.
- FlexPDE 6 (2007). *Reference Manual*, PDE Solutions Inc., Spokane Valley, WA 99206.
- FlexPDE 7 (2017). *Reference Manual*, PDE Solutions Inc., Spokane Valley, WA 99206.
- Fredlund, D. G. and Rahardjo, H., 1993. *Soil Mechanics for Unsaturated Soils*. John Wiley & Sons, New York.
- Fredlund, D. G, Rahardjo, H., and Fredlund, M.D. 2012. *Unsaturated Soils Mechanics in Engineering Practice*. John Wiley & Sons, New York.
- Freeze, R. Allan, and Cherry, John A., 1979. *Groundwater*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Gear, C.W., 1971. The Automatic Integration of Ordinary Differential Equations. *Comm. ACM.*, 14, 176-179.
- Millington, R.J. and Quirk, J.P. 1961. Permeability of porous solids, *Trans. Faraday Soc.* 57:1200-1207.
- Molson, J.W., Fala, O., Aubrtin, M., and Bussiere, B. 2004. Numerical simulation of pyrite oxidation and acid mine drainage in unsaturated waste rock piles. *Journal of Contaminant Hydrology*, 78:343-371.
- Nguyen, Thi Minh Thieu. 1999. *Solution of Saturated/Unsaturated Seepage Problems Using a General Partial Differential Equation Solver*. University of Saskatchewan, Saskatoon, SK.
- Pentland, Jason S. 2000. *Use of a General Partial Differential Equation Solver for Solution of Heat and Mass Transfer Problems in Soils*. University of Saskatchewan, Saskatoon, SK. Canada.
- Simpson, M.J. and Clement, T.P. 2003. Theoretical analysis of the worthiness of Henry and Elder problems as benchmarks of density-dependent groundwater flow models. *Advances in water resources*, 26: 17-31.
- Zhen C. and Wang, P. MT3DMS: A modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater system; Documentation and user's guide.