

# **SVAIR SVENVIRO**

**1D/2D/3D FINITE ELEMENT  
PORE-AIR FLOW MODELING**

## **Theory Manual**

**Written by:  
The Bentley Systems Team**

**Last Updated: Wednesday, August 28, 2019**

**Bentley Systems Incorporated**

## **COPYRIGHT NOTICE**

**Copyright © 2019, 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>AIR FLOW THEORY .....</b>	<b>5</b>
2.1	BASIC GOVERNING EQUATION .....	5
2.2	AIRFLOW STEADY-STATE GOVERNING EQUATION.....	6
2.3	UNSATURATED TRANSIENT-STATE AIR FLOW.....	7
2.4	AIRFLOW GOVERNING EQUATIONS COUPLED WITH WATER FLOW .....	8
2.5	AIRFLOW GOVERNING EQUATIONS COUPLED WITH WATER FLOW AND HEAT FLOW.....	9
2.6	NATURAL CONVECTION .....	11
<b>3</b>	<b>MATERIAL PROPERTIES.....</b>	<b>12</b>
3.1	AIR CONDUCTIVITY .....	12
3.1.1	<i>Calculation of air conductivity based on intrinsic or saturated hydraulic conductivity.....</i>	<i>12</i>
3.1.2	<i>Determination of air relative permeability.....</i>	<i>13</i>
3.1.3	<i>Experimental approach to determining air conductivity.....</i>	<i>14</i>
3.2	ANISOTROPY .....	15
3.2.1	<i>2D.....</i>	<i>15</i>
3.2.2	<i>3D.....</i>	<i>15</i>
<b>4</b>	<b>BOUNDARY CONDITIONS AND INITIAL CONDITIONS.....</b>	<b>16</b>
<b>5</b>	<b>WELLS AND TUNNELS.....</b>	<b>17</b>
5.1	INTRODUCTION.....	17
5.2	EQUATIONS .....	17
5.3	GEOMETRY .....	18
5.4	SETTING THE INFLUENCE DISTANCE.....	19
5.5	BOUNDARY CONDITIONS .....	19
5.5.1	<i>Rate .....</i>	<i>19</i>
5.5.2	<i>Air Pressure.....</i>	<i>19</i>
5.6	PARAMETERS .....	19
5.7	REMARKS .....	20
<b>6</b>	<b>NUMERICAL IMPLEMENTATION.....</b>	<b>22</b>
6.1	EQUATION ANALYZER .....	22
6.2	MESH GENERATION .....	22
6.3	MESH REFINEMENT.....	23
6.4	INTERPOLATION ORDER.....	23
6.5	TIME STEPPING .....	23
6.5.1	<i>Time Integration.....</i>	<i>23</i>
6.5.2	<i>Time step control.....</i>	<i>24</i>
<b>7</b>	<b>REFERENCES.....</b>	<b>26</b>

# 1 INTRODUCTION

Air flow 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 air flow 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 air flow. Rather, the intent is to clearly describe details of the theory used in the SVAIR software.

The SVAIR software packages utilize the FlexPDE generic finite element solver to solve the partial differential equation for air flow. 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,
- 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 preconditioned is the diagonal-block inverse matrix.

## 2 AIR FLOW THEORY

The following section presents the air flow equation implemented in SVAIR. Air flow involves the movement of the air phase through a porous media such as soils. Air flow can occur as a result of continuous convection caused by pressure differences, or discontinuous air bubble flow, or the diffusion of air, which is caused by concentration gradients (Ba-Te, 2004). There can also be a combination of more than one of the above mention forms of air flow.

Following is a general outline of the theory needed to solve the air flow model.

### 2.1 BASIC GOVERNING EQUATION

The airflow governing equation can be derived from mass continuity equation:

$$\frac{\partial(\rho_a q_a)}{\partial z} + \frac{\partial(\rho_a \theta_a)}{\partial t} = 0 \quad [1]$$

where:

$\rho_a$	= air density, $kg/m^3$ ,
$q_a$	= air flow specific discharge, $m/s$ ,
$\theta_a$	= volumetric air content, $m^3/m^3$ ,
$t$	= time, $s$ , and
$z$	= elevation, $m$ .

Considering the air compressibility, the air density,  $\rho_a$ , is described as follows:

$$\rho_a = c_v \frac{\omega(u_a + u_{atm})}{RT} \quad [2]$$

where:

		Metric unit	Imperial unit
$\rho_a$	air density	$kg/m^3$	$lbm/ft^3$
$u_a$	pore air pressure (gauge pressure)	kPa	psf
$u_{atm}$	atmospheric pressure	101.325 kPa	2116.173 psf
$\omega$	average molecular weight	0.0288 kg/mol	0.0635 lbm/mol
$R$	universal gas constant	8.3144 J/mol-K	$7.885 \times 10^{-3}$ Btu/(mol-K)
$T$	absolute temperature	K	K
$C_v$	unit conversion	1000	$1.285 \times 10^{-3}$

It is assumed that the airflow specific discharge,  $q_a$ , obeys the Darcy law, which is

$$q_a = -\frac{g \rho_a k_i}{\mu_a} \frac{\partial h_a}{\partial z} \quad [3]$$

where:

$h_a$	= pore air pressure head, $m$ ,
$g$	= gravitational acceleration, $m/s^2$ ,
$k_i$	= intrinsic permeability, $m^2$ , and
$\mu_a$	= dynamic viscosity, $kg/m-s$ , which is the function of temperature.

The pore air pressure head can be expressed as

$$h_a = z + \frac{1}{g} \int_{u_{a0}}^{u_a} \frac{du_a}{\rho} \quad [4]$$

Using equation [ 4 ], the airflow specific discharge can be generally written as following

$$q_a = -\left[ k_a + \frac{C_v k_a}{g \rho_a} \frac{\partial u_a}{\partial z} \right] \quad [5]$$

$$k_a = \frac{g \rho_a k_i k_{ra}}{\mu_a} \quad [6]$$

where:

- $k_a$  = airflow conductivity,  $m/s$ ,
- $k_{ra}$  = relative air permeability, a dimensionless coefficient,
- $C_v$  = unit conversion, for metric  $C_v = 1000$ , for imperial,  $C_v = 1$ .

The relative air permeability,  $k_{ra}$ , is used to include the effect of the degree of water saturation of a material on the air conductivity, which will be discussed in more detail in the section of Material Properties of this document.

Substituting equation [ 5 ] into equation [ 1 ], the mass conservation of airflow governing equation is described as

$$\frac{\partial}{\partial z} \left( \frac{g k_a \rho_a}{C_v} + k_a \frac{\partial u_a}{\partial z} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} - (n - \theta_w) \frac{(u_a + u_{atm})}{T} \frac{\partial T}{\partial t} \right] \quad [7]$$

where:

- $k_a$  = air conductivity,  $m/s$ ,
- $\rho_a$  = gas density,  $kg/m^3$ ,
- $u_a$  = pore air pressure,  $kPa$ ,
- $u_{atm}$  = atmospheric pressure, 101.325  $kPa$ ,
- $g$  = gravity acceleration, 9.807  $m/s^2$ ,
- $\theta_w$  = volumetric water content,  $m^3/m^3$ ,
- $n$  = porosity,
- $\omega_a$  = average molecular weight of air, 0.0288  $kg/mol$ ,
- $R$  = universal gas constant 8.3144  $J/mol-K$ ,
- $T$  = absolute temperature,  $K$ , and
- $T_0$  = Reference temperature, 273.15  $K$ .
- $t$  = time,  $s$ .
- $C_v$  = unit conversion, for metric  $C_v = 1000$ , for imperial,  $C_v = 1$

For isothermal problem (i.e., the temperature gradient effect is neglected), the governing equation becomes

$$\frac{\partial}{\partial z} \left( \frac{g k_a \rho_a}{C_v} + k_a \frac{\partial u_a}{\partial z} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} \right] \quad [8]$$

If the water content is not changed in the airflow (i.e.,  $\partial(\theta_w)/\partial t = 0$ ), the equation [ 8 ] can be further simplified as following

$$\frac{\partial}{\partial z} \left( \frac{g k_a \rho_a}{C_v} + k_a \frac{\partial u_a}{\partial z} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} \right] \quad [9]$$

## 2.2 AIRFLOW STEADY-STATE GOVERNING EQUATION

According to the equation [ 9 ] and neglecting air gravity weight, the partial differential equation governing steady-state airflow in the formulation of the PDE are presented below for one-dimensional two-dimensional, axisymmetric and three-dimensional coordinate.

- **One-Dimensional Horizontal**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) = 0 \quad [10]$$

- **One-Dimensional Vertical**

$$\frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} + \frac{g k_{ay} \rho_a}{C_v} \right) = 0 \quad [11]$$

- **Two-Dimensional**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} + \frac{g k_{ay} \rho_a}{C_v} \right) = 0 \quad [12]$$

- **Axisymmetric**

$$\frac{\partial}{\partial r} \left( k_{ar} \frac{\partial u_a}{\partial r} \right) + \frac{k_{ar}}{r} \frac{\partial u_a}{\partial r} + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{g k_{az} \rho_a}{C_v} \right) = 0 \quad [13]$$

- **Three-Dimensional**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} \right) + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{g k_{az} \rho_a}{C_v} \right) = 0 \quad [14]$$

where:

- $u_a$  = pore-air pressure, *kPa*,
- $k_{ax}(\Psi)$  = air conductivity of the material in the *x* direction as a function of soil suction, *m/s*,
- $k_{ay}(\Psi)$  = air conductivity of the material in the *y* direction as a function of soil suction, *m/s*,
- $k_{az}(\Psi)$  = air conductivity of the material in the *z* direction as a function of soil suction, *m/s*,
- $k_{ar}(\Psi)$  = air conductivity of the material in the *r* direction as a function of soil suction, *m/s*,
- x, y, z, r* = coordinate in *x*, *y*, *z*, or *r* direction, and
- $C_v$  = unit conversion, for metric  $C_v = 1000$ , for imperial,  $C_v = 1$

A functional representation of air conductivity is generally required when solving for correct solution to most steady-state, unsaturated air flow models. As a material dries, there is more air present in the soil matrix. Since air flows only where there is air in the soil, the air conductivity increases as the volumetric air content increases. This behavior will be discussed in the section of material properties in this document.

## 2.3 UNSATURATED TRANSIENT-STATE AIR FLOW

A transient-state air flow model can have both air conductivity and volumetric air content varying as the suction changes in the soil. The derivation of the unsaturated transient-state air flow equation is based on the principles of conservation of mass as described in equation [ 9 ].

- **One-Dimensional Horizontal**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} \right] \quad [15]$$

- **One-Dimensional Vertical**

$$\frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} + \frac{g k_{ay} \rho_a}{C_v} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} \right] \quad [16]$$

- **Two-Dimensional**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} + \frac{g k_{ay} \rho_a}{C_v} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} \right] \quad [17]$$

- **Axisymmetric**

$$\begin{aligned} \frac{\partial}{\partial r} \left( k_{ar} \frac{\partial u_a}{\partial r} \right) + \frac{k_{ar}}{r} \frac{\partial u_a}{\partial r} + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{g k_{az} \rho_a}{C_v} \right) \\ = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} \right] \end{aligned} \quad [18]$$

- **Three-Dimensional**

$$\begin{aligned} \frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} \right) + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{g k_{az} \rho_a}{C_v} \right) \\ = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} \right] \end{aligned} \quad [19]$$

where:

$u_a$	= pore-air pressure, <i>kPa</i> ,
$k_{ax}(\Psi)$	= air conductivity of the material in the <i>x</i> direction, <i>m/s</i> ,
$k_{ay}(\Psi)$	= air conductivity of the material in the <i>y</i> direction, <i>m/s</i> ,
$k_{az}(\Psi)$	= air conductivity of the material in the <i>z</i> direction, <i>m/s</i> ,
$k_{ar}(\Psi)$	= air conductivity of the material in the <i>r</i> direction, <i>m/s</i> ,
$n$	= porosity,
$\theta_w$	= volumetric water content, <i>m<sup>3</sup>/m<sup>3</sup></i> ,
$\omega_a$	= average molecular weight of air 0.0288 <i>kg/mol</i> ,
$g$	= acceleration due to gravity, 9.81 <i>m/s<sup>2</sup></i> ,
$R$	= universal gas constant, 8.3144 <i>J/mol-K</i> , and
$T$	= temperature ( <i>K</i> ).

The formulations presented above state that the difference between the air flow (flux) entering or leaving a unit volume is equal to the change in volumetric air content. Under steady-state conditions, the flux entering and leaving a unit volume is the same so the storage term (right-hand side of the equation) becomes zero.

The above formulations make the following assumptions:

- The water content is not influenced by airflow.
- There is no loading or unloading of the material mass(i.e. no total stress changes),
- Isothermal conditions.

## 2.4 AIRFLOW GOVERNING EQUATIONS COUPLED WITH WATER FLOW

To include the interaction of water flow and airflow, extend the equation [ 8 ] for different coordinates as following:

- **One-Dimensional Horizontal**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} \right] \quad [20]$$

- **One-Dimensional Vertical**

$$\frac{\partial}{\partial x} \left( k_{ay} \frac{\partial u_a}{\partial x} + \frac{g k_{ay} \rho_a}{C_v} \right) = \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial (\theta_w)}{\partial t} \right] \quad [21]$$

- **Two-Dimensional**

$$\begin{aligned} \frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} + \frac{g k_{ay} \rho_a}{C_v} \right) = \\ \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial (\theta_w)}{\partial t} \right] \end{aligned} \quad [22]$$

- **Axisymmetric**

$$\begin{aligned} \frac{\partial}{\partial r} \left( k_{ar} \frac{\partial u_a}{\partial r} \right) + \frac{k_{ar}}{r} \frac{\partial u_a}{\partial r} + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{g k_{az} \rho_a}{C_v} \right) = \\ \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial (\theta_w)}{\partial t} \right] \end{aligned} \quad [23]$$

- **Three-Dimensional**

$$\begin{aligned} \frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} \right) + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{g k_{az} \rho_a}{C_v} \right) = \\ \frac{g \omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial (\theta_w)}{\partial t} \right] \end{aligned} \quad [24]$$

where:

$u_a$	= pore-air pressure, <i>kPa</i> ,
$u_{atm}$	= atmospheric air pressure, 101.325 <i>kPa</i> ,
$k_{ax}(\Psi)$	= air conductivity of the material in the <i>x</i> direction, <i>m/s</i> ,
$k_{ay}(\Psi)$	= air conductivity of the material in the <i>y</i> direction, <i>m/s</i> ,
$k_{az}(\Psi)$	= air conductivity of the material in the <i>z</i> direction, <i>m/s</i> ,
$k_{ar}(\Psi)$	= air conductivity of the material in the <i>r</i> direction, <i>m/s</i> ,
$n$	= porosity,
$\theta_w$	= volumetric water content, <i>m<sup>3</sup>/m<sup>3</sup></i> ,
$\omega_a$	= average molecular weight of air 0.0288 <i>kg/mol</i> ,
$g$	= acceleration due to gravity, 9.81 <i>m/s<sup>2</sup></i> ,
$R$	= universal gas constant, 8.3144 <i>J/mol-K</i> , and
$T$	= temperature ( <i>K</i> ).

## 2.5 AIRFLOW GOVERNING EQUATIONS COUPLED WITH WATER FLOW AND HEAT FLOW

With the equation [ 7 ], the governing equations of airflow coupled with water flow and heat flow in the different coordinates are presented as following.

- **One-Dimensional Horizontal**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) = \frac{g\omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} - (n - \theta_w) \frac{(u_a + u_{atm})}{T} \frac{\partial T}{\partial t} \right] \quad [25]$$

- **One-Dimensional Vertical**

$$\frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} + \frac{gk_{ay}\rho_a}{C_v} \right) = \frac{g\omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} - (n - \theta_w) \frac{(u_a + u_{atm})}{T} \frac{\partial T}{\partial t} \right] \quad [26]$$

- **Two-Dimensional**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} + \frac{gk_{ay}\rho_a}{C_v} \right) = \frac{g\omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} - (n - \theta_w) \frac{(u_a + u_{atm})}{T} \frac{\partial T}{\partial t} \right] \quad [27]$$

- **Axisymmetric**

$$\frac{\partial}{\partial r} \left( k_{ar} \frac{\partial u_a}{\partial r} \right) + \frac{k_{ar}}{r} \frac{\partial u_a}{\partial r} + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{gk_{az}\rho_a}{C_v} \right) = \frac{g\omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} - (n - \theta_w) \frac{(u_a + u_{atm})}{T} \frac{\partial T}{\partial t} \right] \quad [28]$$

- **Three-Dimensional**

$$\frac{\partial}{\partial x} \left( k_{ax} \frac{\partial u_a}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{ay} \frac{\partial u_a}{\partial y} \right) + \frac{\partial}{\partial z} \left( k_{az} \frac{\partial u_a}{\partial z} + \frac{gk_{az}\rho_a}{C_v} \right) = \frac{g\omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} - (u_a + u_{atm}) \frac{\partial(\theta_w)}{\partial t} - (n - \theta_w) \frac{(u_a + u_{atm})}{T} \frac{\partial T}{\partial t} \right] \quad [29]$$

where:

- $u_a$  = pore-air pressure, *kPa*,
- $u_{atm}$  = atmospheric air pressure, 101.325 *kPa*,
- $k_{ax}(\Psi)$  = air conductivity of the material in the *x* direction, *m/s*,
- $k_{ay}(\Psi)$  = air conductivity of the material in the *y* direction, *m/s*,
- $k_{az}(\Psi)$  = air conductivity of the material in the *z* direction, *m/s*,
- $k_{ar}(\Psi)$  = air conductivity of the material in the *r* direction, *m/s*,
- $n$  = porosity,
- $\theta_w$  = volumetric water content, *m<sup>3</sup>/m<sup>3</sup>*,
- $\omega_a$  = average molecular weight of air 0.0288 *kg/mol*,
- $g$  = acceleration due to gravity, 9.81 *m/s<sup>2</sup>*,
- $R$  = universal gas constant, 8.3144 *J/mol-K*, and
- $T$  = temperature (*K*).

## 2.6 NATURAL CONVECTION

The air density is assumed to be incompressible, and it can be applicable with Oberbeck-Boussinesq approximation as following

$$\rho_a = \rho_{a0} [1 - \beta(T - T_0)] \quad [30]$$

where:

$\rho_a$	= pore air density at the given temperature, $kg/m^3$ ,
$\rho_{a0}$	= pore air density at the reference temperature, $kg/m^3$ ,
$T$	= temperature, $^{\circ}C$ ,
$T_0$	= reference temperature, $^{\circ}C$ , and
$\beta$	= air expansion coefficient, $1/^{\circ}C$

The thermal expansion coefficient varies with temperature. Please see the following web site for reference:  
[http://www.engineeringtoolbox.com/air-properties-d\\_156.html](http://www.engineeringtoolbox.com/air-properties-d_156.html)

The governing equation for natural convection becomes:

$$\frac{\partial}{\partial z} \left( k_a \frac{\partial u_a}{\partial z} + \frac{g k_a \rho_a}{C_v} \right) = 0 \quad [31]$$

Please see the SVHEAT Theory Manual for the energy balance equation when coupled with SVHEAT.

### 3 MATERIAL PROPERTIES

The main material properties related to airflow include air conductivity, air density, and air dynamic viscosity, etc.

#### 3.1 AIR CONDUCTIVITY

The air conductivity,  $k_a$ , is the most important material properties in airflow. SVAIR provides the following approaches to determine the air conductivity.

##### 3.1.1 Calculation of air conductivity based on intrinsic or saturated hydraulic conductivity

If the intrinsic permeability is known, the air conductivity can be calculated with the following equation.

$$k_a = \frac{g \rho_a k_i k_r}{\mu_a} \quad [32]$$

where:

- $k_a$  = air conductivity,  $m/s$ ,
- $k_i$  = intrinsic permeability of a material,  $m^2$ ,
- $k_r$  = relative air permeability, a dimensionless coefficient changing from 0 to 1 (see the next section for detail),
- $g$  = gravitational acceleration,  $m/s^2$ ,
- $\rho_a$  = air density,  $kg/m^3$ , and
- $\mu_a$  = air dynamic viscosity,  $kg/m-s$ .

The material intrinsic permeability can be estimated using the saturated hydraulic conductivity, which is

$$k_i = \frac{k_{sat} \mu_w}{g \rho_w} \quad [33]$$

where:

- $k_i$  = intrinsic permeability,  $m/s$ ,
- $k_{sat}$  = saturated hydraulic conductivity,  $m/s$ ,
- $\mu_w$  = water dynamic viscosity,  $kg/m-s$ ,
- $g$  = gravitational acceleration,  $m/s^2$ , and
- $\rho_w$  = water density,  $1000 kg/m^3$ .

Therefore, the air conductivity can be estimated using hydraulic conductivity with the following expression.

$$k_a = \frac{\rho_a \mu_w k_{sat} k_r}{\rho_w \mu_a} \quad [34]$$

The air density is calculated using equation [ 2 ], in which the pore air pressure is determined by the governing equation. In the normal case at temperature = 20 °C and pressure = 1 atm, the air density  $\rho_a = 1.197 kg/m^3$ .

The air dynamic viscosity is mostly dependent of temperature, and it is approximated with the following empirical expression:

$$\mu_a = \mu_{a0} \sqrt{\frac{T}{T_0}} \quad [35]$$

where:

- $\mu_a$  = air dynamic viscosity at the given temperature,  $kg/m-s$ ,
- $\mu_{a0}$  = air dynamic viscosity,  $1.83 \times 10^{-5} kg/m-s$  at a reference temperature,  $T_0$ ,
- $T$  = temperature,  $K$ , and
- $T_0$  = reference temperature,  $293.15 K$ .

The water dynamic viscosity as the function of temperature is:

$$\mu_w = A \times 10^{\frac{B}{(T-C)}} \quad [36]$$

where:

- $\mu_w$  = water dynamic gas viscosity,  $kg/m-s$ ,
- $A$  =  $2.414 \times 10^{-5}$ ,  $kg/m-s$ ,
- $B$  = 247.8 K,
- $C$  = 140 K, and
- $T$  = temperature (K).

### 3.1.2 Determination of air relative permeability

The size of cross section of airflow in the unsaturated soil is decreased because partial cross section is filled with water. As a result, the air permeability will be reduced with the increasing degree of water saturation. The effect of soil water saturation on the air conductivity is evaluated using a coefficient of air relative permeability that is calculated with the following approaches:

- Brooks and Corey (1964) – Burdine (1953) approach

$$k_r = (1 - S_e)^2 \left( 1 - S_e^{(2+\lambda)/\lambda} \right) \quad [37]$$

- Brooks and Corey (1964) – Mualem (1976) approach

$$k_r = (1 - S_e)^{0.5} \left( 1 - S_e^{(1+\lambda)/\lambda} \right)^2 \quad [38]$$

- Van Genuchten (1980) – Mualem (1976) approach

$$k_r = (1 - S_e)^{1/2} \left( 1 - S_e^{1/M} \right)^{2M} \quad [39]$$

- Van Genuchten (1980) – Burdine (1953) approach

$$k_r = (1 - S_e)^2 \left( 1 - S_e^{1/M} \right)^M \quad [40]$$

$$S_e = \frac{(S_w - S_r)}{1 - S_r} \quad [41]$$

$$S_w = \frac{\theta_w}{\theta_{sat}} = \frac{\theta_w}{n} \quad [42]$$

$$S_r = \frac{\theta_r}{\theta_{sat}} = \frac{\theta_r}{n} \quad [43]$$

where:

- $S_e$  = effective degree of water saturation.
- $\lambda$  = fitting parameter with Brooks and Corey SWCC method,
- $M$  = fitting parameter with van Genuchten SWCC method,
- $S_w$  = degree of water saturation,
- $S_r$  = residual degree of water saturation,
- $n$  = porosity,
- $\theta_w$  = volumetric water content,  $m^3/m^3$ ,
- $\theta_r$  = residual volumetric water content,  $m^3/m^3$ , and
- $\theta_{sat}$  = saturated volumetric water content,  $m^3/m^3$ .

The volumetric water content,  $\theta_w$ , of unsaturated soil is governed by soil water characteristic curve (SWCC). Please see **SVFLUX theory manual** for the different approaches such as Fredlund and Xing fitting method to estimate  $\theta_w$  as a function of matric suction.

### 3.1.3 Experimental approach to determining air conductivity

Air conductivity may be determined by different experiments. The following methods are provided in SVAIR for user to specify the air conductivity:

- Constant air conductivity or expression,
- Data table of air conductivity as the function of the degree of water saturation,
- Data table of air conductivity as the function of pore air pressure.

## 3.2 ANISOTROPY

Natural deposition of material layers often results in angled layering. The angled layering results in dominant air flow in a direction parallel to the direction of the layering. This phenomenon can be simulated in SVAIR by specifying anisotropic behavior in either 2D or 3D. The respective formulations are presented in the following sections.

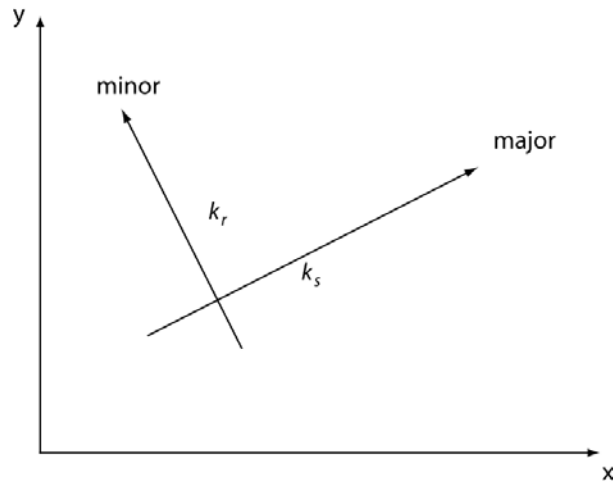


Figure 1

### 3.2.1 2D

A transformation of the specified anisotropic air permeability coefficients must be performed to convert the anisotropic permeability parameters onto the coordinate system used to solve the model. The transformation is shown below.

The general form of the conductivity matrix is as follows Bear (1972):

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \quad [44]$$

where:

$$K_{11} = \frac{kx + ky}{2} + \frac{kx - ky}{2} \cos 2\alpha$$

$$K_{22} = \frac{kx + ky}{2} - \frac{kx - ky}{2} \cos 2\alpha$$

$$K_{12} = \frac{kx - ky}{2} \sin 2\alpha$$

$$K_{21} = K_{12}$$

### 3.2.2 3D

The transformation in 3D is slightly more rigorous than in 2D. Three angles are required to describe the conductivity transformation. The theory for the transformations was not presented at this time.

## 4 BOUNDARY CONDITIONS AND INITIAL CONDITIONS

There are three types of main boundary conditions implemented in SVAIR:

- Essential (or Dirichlet) boundary conditions specify a constant value or expression of air pressure along the boundary of model domain.
- Natural (or Neumann) boundary conditions specify airflow flux, and
- Special boundary conditions: combinations of the above conditions.

## 5 WELLS AND TUNNELS

### 5.1 INTRODUCTION

In a numerical model, tunnels and wells are represented as localized sinks or sources. Numerically, they draw air out of cells (pumping) or add air to cells (injection). For regional scale models, the diameter of a well borehole or tunnel shaft is too small to model efficiently, as it requires a mesh resolution fine enough to model the physical borehole.

A traditional method of representing well and tunnel objects is with a line of nodes in a finite element model. Such a representation is efficient in terms of the mesh, but can result in extremely sharp gradients next to the line of elements. This is especially true if the internal boundary condition applied to the well/tunnel differs greatly from the surrounding domain. Resulting sharp gradients can lead to instabilities in the numerical model.

For these reasons, SVAIR treats wells differently. SVAIR models wells as a sink (or source) term in the partial differential equations, rather than as a physical boundary condition. This eliminates the meshing issues with trying to model a borehole that is very small relative to the extent of the model. This methodology introduces an approximation that affects the distance at which the flux across a closed surface bounding the well screen becomes accurate. The user can set this *influence distance*, and experiment with the level that provides the right mix of accuracy and modeling speed.

In SVAIR, wells are vertical features, whereas tunnels can be inclined at any angle, and be a sequence of straight line segments. They may be used interchangeably. Both can be included in two- and three-dimensional models.

### 5.2 EQUATIONS

Equation [ 45 ] is the air flow equation in three-dimensions.

$$\begin{aligned} & \frac{\partial}{\partial x} \left[ k_{ax} \frac{\partial u_a}{\partial x} \right] + \frac{\partial}{\partial y} \left[ k_{ay} \frac{\partial u_a}{\partial y} \right] + \frac{\partial}{\partial z} \left[ k_{az} \frac{\partial u_a}{\partial z} + \frac{gk_{az}\rho_a}{C_v} \right] + \frac{g\rho_a}{C_v} \bar{Q}_{well} \\ & = \frac{g\omega_a}{RT} \left[ (n - \theta_w) \frac{\partial u_a}{\partial t} \right] \end{aligned} \quad [ 45 ]$$

Where:

$u_a$	= Pore-air pressure,
$k_{ax}$	= air conductivity in the $x$ direction,
$k_{ay}$	= air conductivity in the $y$ direction,
$k_{az}$	= air conductivity in the $z$ direction,
$n$	= porosity,
$\theta_a$	= volumetric water content,
$\omega_a$	= average molecular weight of air (0.0288 kg/mol),
$g$	= acceleration due to gravity,
$R$	= universal gas constant (8.3144 J/mol-K),
$T$	= temperature.
$C_v$	= 1 for imperial units, 1000 for metric units

The source term,  $\bar{Q}_{well}$ , Equation [ 46 ], simulates the wells.

$$\bar{Q}_{well} = \frac{P}{\int_V P dV} Q_{well} \quad [ 46 ]$$

where  $Q_{well}$  is the user-defined pumping/injection rate of the well, and  $P$ , Equation [ 47 ], is a function that scales the source term.

$$P = \exp\left(-\frac{|\mathbf{r}|^2}{\alpha^2} \ln 20\right) \quad [ 47 ]$$

where  $\mathbf{r}$  (described in Section 5.3) is the shortest vector from the well to a point in the model domain, and  $\alpha$  is the user-defined *influence distance*. With this definition of  $P$ , when  $|\mathbf{r}| = \alpha$ ,  $P = 1/20$ . The integral  $\int_V P dV$  in Equation [ 46 ] is taken over the entire volume,  $V$ , of the model, and normalizes  $P$  so that  $\int_V \bar{Q}_{well} dV = Q_{well}$ . Figure 2 shows some examples of  $P$  for some sample values of  $\alpha$ .

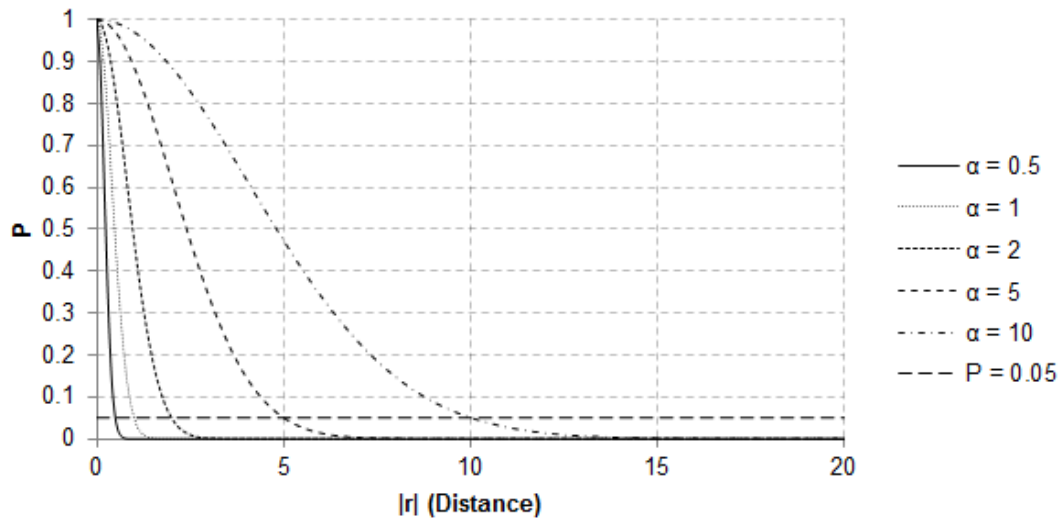


Figure 2: Examples of  $P$  for values of  $\alpha$ .

## 5.3 GEOMETRY

The vector,  $\mathbf{r}$ , in Equation [ 47 ], is either the vector perpendicular to the well screen to the model point, or the vector from the closest end of the screen to the point. Figure 3 shows  $\mathbf{r}$  when the model point is the closest to a point in the screened segment, and Figure 4 shows  $\mathbf{r}$  when the model point is the closest to an end of the screen.

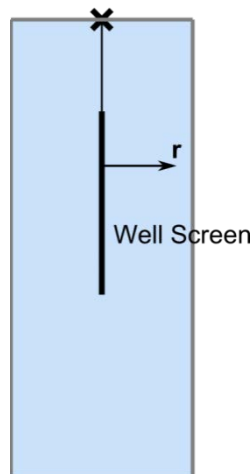


Figure 3: Shortest distance to a model point from the screened segment.

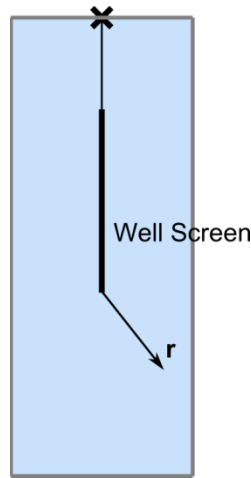


Figure 4: Shortest distance to a model point from an end.

## 5.4 SETTING THE INFLUENCE DISTANCE

The influence distance affects how far away from the screen  $Q_{well}$  becomes realized. Smaller influence distances will cause the well flux to be realized closer to the screen than larger influence distances.

The influence distance should be set to a value that represents the scale at which the air flux is important. In general, if the model has horizontal extents in the hundreds of meters, the flux in a volume immediately surrounding the screen is less important than the overall affect that the well has on the flow in the region. Therefore, a larger value is appropriate (~10 m). This can produce shallower gradients and prevent the computations in the volume surrounding the well from dominating the system.

For simpler models, the influence distance has little effect on the speed of the model. Therefore, the influence distance can be set to a small value (~1 m).

## 5.5 BOUNDARY CONDITIONS

All well and tunnel boundary conditions are implemented through the sink/source term,  $\bar{Q}_{well}$ , in Equation [ 45 ]. SVAIR calculates an appropriate value of  $Q_{well}$  for each type of boundary condition.

### 5.5.1 Rate

The value of  $Q_{well}$  in Equation [ 46 ] is specified.

### 5.5.2 Air Pressure

A specified air pressure,  $u_{well}$ , is maintained by causing the flow necessary to realize  $u_{well}$ , along the screen. The flow is given by:

$$Q_{well} = \begin{cases} K(u_{well} - u_a)^2, & u_a \leq u_{well} \\ -K(u_a - u_{well})^2, & u_a > u_{well} \end{cases} \quad [ 48 ]$$

where  $u_a$  is soil air pressure and  $K$  is the *BIG\_WELL* parameter that can be input.

## 5.6 PARAMETERS

Wells and tunnels are controlled by three user definable parameters: *Influence Distance* ( $\alpha$ ), *Line Mesh Spacing* ( $\zeta$ ), and *Mesh Growth Coefficient* ( $\gamma$ ). The influence distance controls the spread of the sink that simulates the well, and line mesh spacing and mesh growth coefficient control the mesh spacing near the well.

The well sink is controlled by the scale factor formula:

$$P = \exp\left(-\frac{|r|^2}{\alpha^2} \ln 20\right). \quad [49]$$

This is described in detail in Section 5.3.

The mesh spacing is controlled by:

$$m = \max\left\{\zeta, \gamma |r|^{1.2}\right\}. \quad [50]$$

Equation [ 50 ] ensures that  $\zeta$  is the smallest element size, and causes the size to grow at a rate proportional to  $|r|^{1.2}$ . The exponent of 1.2 was selected to allow the mesh to grow at a rate that is generally appropriate for models.

Figure 2 shows the sink scale factor ( $P$ ) for some values of influence distance, and Figure 5 shows the mesh spacing for some values of the mesh growth coefficient.

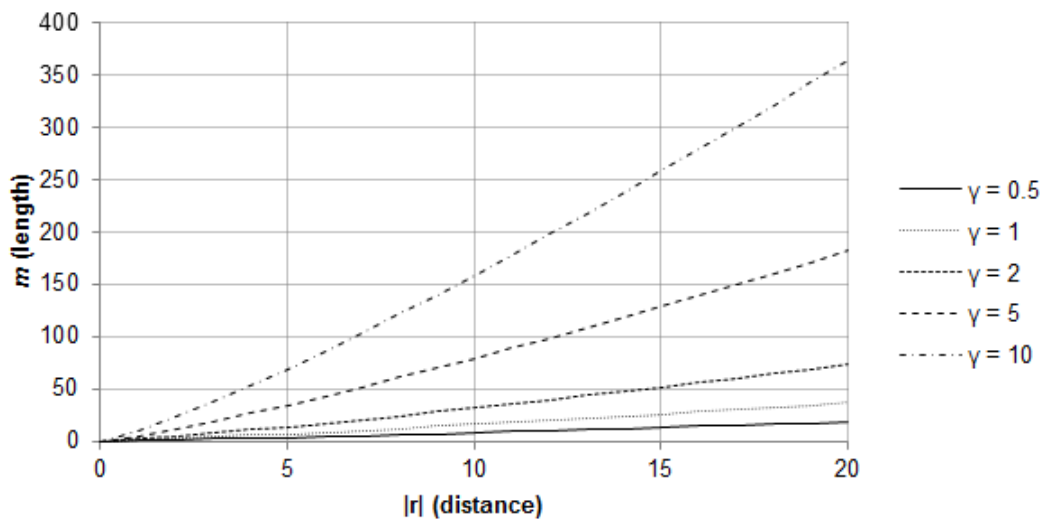


Figure 5: Mesh spacing ( $m$ ) for various values of mesh growth coefficient ( $\gamma$ ).

## 5.7 REMARKS

Changing the default mesh parameters in Equation [ 50 ] is only necessary if the automatically generated mesh is not suitable. The mesh may be too coarse or too fine near the well, and adjustment may be necessary to facilitate accurate and efficient modelling.

It is possible that in large models, the mesh generation will not resolve the well. The mesher may fail to resolve the peak of the sink/source, regardless of how the parameters are set. In this case, it is necessary to increase the mesh density of the model by decreasing the mesh size. It is often best to create a region that contains the well, and, if necessary, decrease the mesh size for that region only. Usually, just creating the region is enough for the mesher to resolve the well. The region mesh size will override the mesh size set by Equation [ 50 ].

When wells or tunnels are directly represented in a model as a borehole, it causes extremely dense meshing near the borehole, which must be resolved out to the model scale mesh. This can generally be accomplished; however, the resulting mesh is extremely dense close to the well, and the model runs very slow. The advantage of incorporating wells and tunnels as sinks/sources in the partial differential equation is that the mesh size is determined at the model scale. This generally gives accurate results in much less time. Equations [ 49 ] and [ 50 ] provide control of the rate of decay of the sink/source and the mesh density so that the model can be fine-tuned.

FlexPDE takes several characteristics of the geometry into consideration when generating the mesh. It is possible that the mesh size set by Equation [ 50 ] will be overridden by another size parameter. Since FlexPDE will select the smallest mesh size out of all the possible sizes under consideration, this should not cause any problems with the simulation.

For complex models, especially those for which the well or tunnel is screened across a contact between materials with a large difference in conductivities, the gradients in the immediate vicinity of the well need to be monitored to ensure they are not unrealistically large.

The *Rate* boundary condition in a two-dimensional model is a volume per unit time. When determining the appropriate rate, the user must consider that the well is being applied to a three-dimensional model with the third dimension having unit length. This means that a rate that is suitable for a three-dimensional model may be too large for a corresponding two-dimensional model.

## 6 NUMERICAL IMPLEMENTATION

The SVAIR 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.

### 6.1 EQUATION ANALYZER

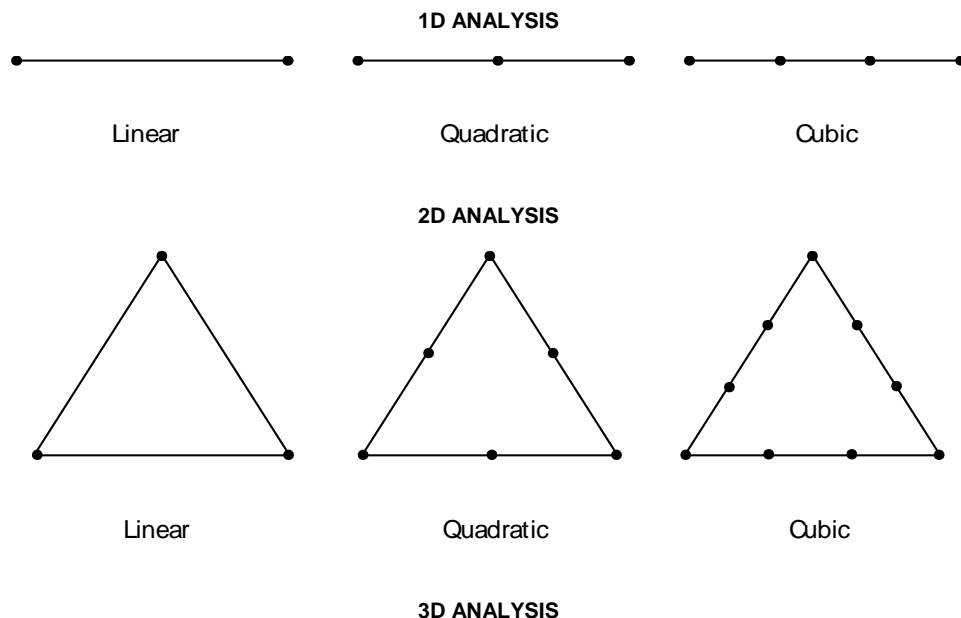
A symbolic *equation analyzer* expands defined parameters and relations, performs spatial differentiation, and symbolically applies integration by parts to reduce second order terms to create symbolic Galerkin equations. It then differentiates these equations to form the Jacobian coupling matrix.

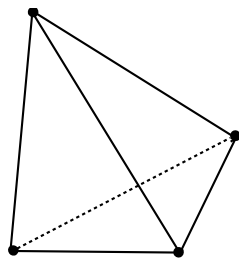
### 6.2 MESH GENERATION

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.

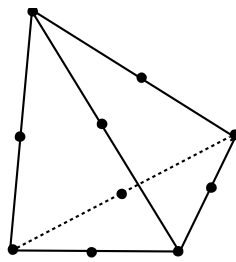
FlexPDE uses an "advancing front" mesh generation method, triangular in 2D and tetrahedral in 3D (the specific algorithm is proprietary, but similar mesh generation methods are widely reported in the literature). The mesh generator allows spatially-varying node density, in order to concentrate cells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

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

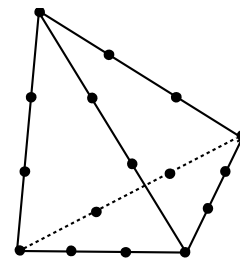




Linear



Quadratic



Cubic

## 6.3 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.

## 6.4 INTERPOLATION ORDER

Finite element basis is quadratic.

## 6.5 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 SVAIR as the most robust method for general application. The research literature supports this decision.

### 6.5.1 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  $U(t)$

$$\begin{aligned}
 U'(t) &= p0'u0 + p1'u1 + p2'u2 \\
 \text{with} \\
 p0'(t) &= ((t-t1)+(t-t2)) / ((t1-t0)*(t2-t0)), \\
 p1'(t) &= -((t-t0)+(t-t2)) / ((t1-t0)*(t2-t1)), \text{ and} \\
 p2'(t) &= ((t-t0)+(t-t1)) / ((t2-t0)*(t2-t1)).
 \end{aligned}$$

The time derivative at t2 is then,

$$\begin{aligned}
 U'(t2) &= p0'(t2)*u0 + p1'(t2)*u1 + p2'(t2)*u2 \\
 \text{with} \\
 p0'(t2) &= (t2-t1) / ((t1-t0)*(t2-t0)), \\
 p1'(t2) &= -(t2-t0) / ((t1-t0)*(t2-t1)), \text{ and} \\
 p2'(t2) &= (2*t2-t1-t0) / ((t2-t0)*(t2-t1)).
 \end{aligned}$$

Now, if we are given an equation,

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

We can write,

$$\begin{aligned}
 U'(t2) &= F(U(t2)) \text{ as a fully implicit equation for } U2, \text{ and} \\
 U2 &= [F(U2) - p0'(t2)*U0 - p1'(t2)*U1] / p2'(t2).
 \end{aligned}$$

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 U2 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.

## 6.5.2 Time step control

The approximation described above implies a constant curvature U'' over the interval (t0, t2). When we later advance from t2 to t3, we implicitly assume that U'' is constant over the interval (t1, t3), 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 (t1, t2).

The FlexPDE time step control, simply stated, is to measure how much difference the two estimates of U'' imply in U3, and to restrict this difference to the requested ERRLLIM.

The curvature over (t0, t2) is,

$$U''1 = 2*U0/((t1-t0)(t2-t0)) - 2*U1/((t1-t0)(t2-t1)) + 2*U2/((t1-t0)*(t2-t1))$$

The curvature over (t1, t3) is,

$$U''2 = 2*U1/((t2-t1)(t3-t1)) - 2*U2/((t2-t1)(t3-t2)) + 2*U3/((t2-t1)*(t3-t2))$$

The difference is,

$$E''2 = -2*U0/((t1-t0)(t2-t0)) + 2(t3-t0)U1/((t1-t0)(t2-t1)*(t3-t1)) - 2*U2*(t3-t0)/((t2-t1)(t3-t2)(t2-t0)) + 2*U3/((t3-t2)*(t3-t1))$$

Integrating over (t2, t3) produces an error,

$$E3 = -(t3-t2)^2*U0/((t1-t0)*(t2-t0)) + (t3-t2)^2*(t3-t0)U1/((t1-t0)(t2-t1)*(t3-t1)) - (t3-t2)(t3-t0)*U2/((t2-t1)(t2-t0)) + (t3-t2)*U3/(t3-t1)$$

The Nth 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^TNORM. The range of U is the larger of the declared Threshold or the observed range of values.

$$\begin{aligned}
 Eav &= [SUM((E3/Range(U))^N)]^{(1/N)}, \text{ and} \\
 deltat4 &= deltat3*sqrt(TERRLLIM/Eav).
 \end{aligned}$$

---

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  $Eav$  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  $V5$ .

## 7 REFERENCES

- Baehr, A.E. and Hult, M.F., (1991). Evaluation of Unsaturated Zone Air Permeability Through Pneumatic Tests. *Water Resources Research*, 27(10): 2605-2617.
- Ba-Te, (2004). Flow of Air-phase in Soils and Its Application in Emergent Stabilization of Soil Slopes. Hong Kong University of Science and Technology, Hong Kong.
- Ba-Te, Zhang, L., and Fredlund, D.G., (2005). A General Air-Phase Permeability Function for Airflow through Saturated Soils. Hong Kong University of Science and Technology, Hong Kong.
- Bear, Jacob, 1972. *Dynamics of Fluids in Porous Media*. Dover Publications Inc., New York.
- Bear, Jacob, 1979. *Hydraulics of Groundwater*. McGraw-Hill, New York.
- DiGiulio, D.C., and Varadhan, R. 2001. Development of recommendations and methods to support assessment of soil venting performance and closure, EPA/600/R-01/070, September 2001
- 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.
- Jordan, D.L., Mercer, J.W., and Cohen, R.M. 1995. Review of mathematical modeling for evaluation soil vapor extraction systems, EPA/540/R-95/513, July 1995.
- Kolditz, O., McDermott, C., Worsch, R. et al. 2008. Numerical Modeling of Heat Storage in Soils. *Journal of Environmental Science for Sustainable Society*, vol 2: 47-56.
- Nield, D., and Bejan, A., 2006. *Convection in Porous Media*.
- Touma, J. and Vauclin, M. 1986. Experimental and Numerical Analysis of Two-Phase Infiltration in a Partially Saturated Soil. *Transport in Porous Media*, vol 1: 27 –55.

---

This page is left blank intentionally.