Numerical Modeling for Flow and Transport

Þröstur Þorsteinsson
Throstur@turdus.net

Based on presentation by:

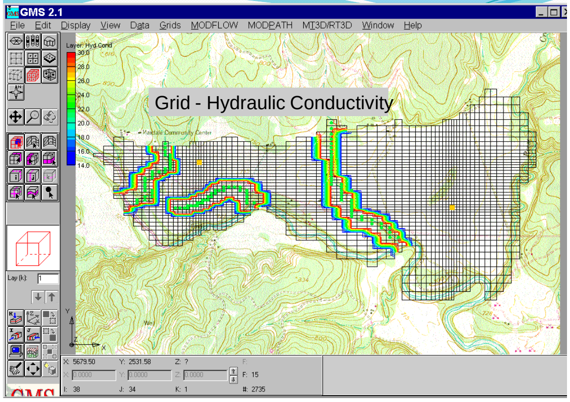
P. Bedient and A. Holder
Rice University, 2005

Uses of Modeling

- A model is designed to represent reality in such a way that the modeler can do one of several things:
  - Quickly estimate certain aspects of a system (screening models, analytical solutions, 'back of the envelope' calculations)
  - Determine the causes of an observed condition (flow direction, contamination, subsidence, flooding)
  - Predict the effects of changes to a system (pumping, remediation, development, waste disposal)

Types of Ground Water Flow Models

- Analytical Models (Exp and ERF functions)
  - 1-D solution, Ogata and Banks (1961)
  - 2-D solution, Wilson and Miller (1978)
  - 3-D solutions, Domenico & Schwartz (1990)
- Numerical Models (Solved over a grid - FDE)
  - Flow-only models in 3-D (MODFLOW)
  - MODPATH - allows tracking of particles in 2-D placed in flow field produced from MODFLOW



Grid - Hydraulic Conductivity

X: 5679.50 Y: 2531.58 Z: ? F:
X: 0.0000 Y: 0.0000 Z: 0.0000 F: 15
I: 38 J: 34 K: 1 #: 2735

Governing Equation for Flow

- For two-dimensional transient flow conditions
  - Transient means that the water level changes with time
  - Steady state means it is constant in time.

$$\boxed{\frac{\partial}{\partial x}\left(T_x\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(T_y\frac{\partial h}{\partial y}\right)=S\frac{\partial h}{\partial t}+Q}$$

- S = storativity [unitless],
- Q = recharge or withdrawal per unit area [L/T]
- T = transmissivity [L^2/T]

Poisson's Equation

- The basic flow equation in a homogeneous, confined, 2-D aquifer at steady-state (S = 0), with sources and/or sinks

$$\boxed{\frac{\partial^2 h}{\partial x^2}+\frac{\partial^2 h}{\partial y^2}=-\frac{Q(x,y)}{T}}$$

- Can be solved analytically or numerically
  - Theis' Analytical solution in cylindrical coordinates
  - Gauss-Seidel with Successive Over-Relaxation (SOR)

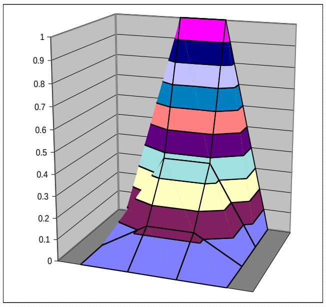
Laplace's Equation

- If there are no source/sink terms, Poisson's equation reduces to Laplace's Equation

$$\boxed{\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0}$$

- Can also be solved analytically or numerically
  - Gauss-Seidel Iteration method
  - Successive Over Relaxation method
- Solutions are generally smooth and well-behaved

Laplace Numerical Solution



Numerical Solutions

- For complex layered, heterogeneous aquifers, a numerical approximation is required in non-uniform flow
- Several types — Finite Difference, Finite Element, and Method of Characteristics
- Finite difference approximations involve applying Taylor's expansions to the equations (flow and transport) and approximating the derivatives in the equation
- Other methods involve different approximations, but all are based generally on the Taylor expansion

Taylor's Expansion from Calculus

Taylor's Series provides a means to predict a function f(x) value at one point in terms of a function value and its derivatives at another point.

Zero Order approximation might be $f(x_{i+1}) = f(x_i)$

Value at new point is same as at old point

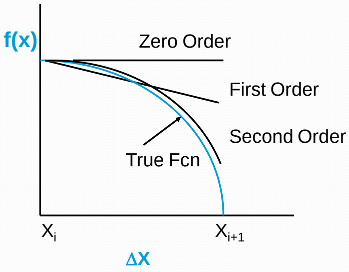
First Order approximation is $f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i)$

Straight line projection to next point

Second Order approximation captures curvature

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + f''(x_i)\,(x_{i+1} - x_i)^2$$

Approximation of f(x) by various orders



Taylor's Expansion from Calculus

Any function $h(x)$ can be expressed as an infinite series:

$$h(x+\Delta x) = h(x) + \Delta x h'(x) + \frac{\Delta x^2}{2} h''(x) + \cdots$$

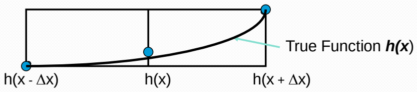
$$h(x-\Delta x) = h(x) - \Delta x h'(x) + \frac{\Delta x^2}{2} h''(x) + \cdots$$

Where *h'(x)* is the first derivative and *h"(x)* is the second derivative and so on.

Taylor's Expansion for Second Derivative

Adding the above two eqns, and neglecting all the higher terms, and rearranging terms gives a useful approximation for *h"(x)* :

$$h''(x) = \left(\frac{d^2h}{dx^2}\right)_x \approx \frac{1}{\Delta x^2}\{h(x+\Delta x) - 2h(x) + h(x-\Delta x)\}$$

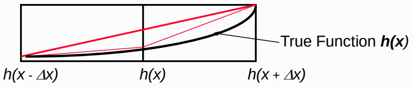


Taylor's Expansion for dh/dx

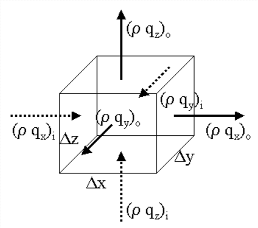
Neglecting second and all higher powers, and rearranging terms gives a forward or backward difference approximation for the first derivative or *h'(x)* :

$$h'(x) = \left(\frac{dh}{dx}\right)_x \approx \frac{1}{\Delta x}\{h(x+\Delta x) - h(x)\}$$ Forward Diff

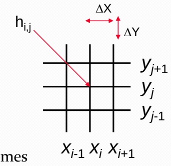
$$h'(x) = \left(\frac{dh}{dx}\right)_x \approx \frac{1}{\Delta x}\{h(x) - h(x-\Delta x)\}$$ Backward Diff



Control Volume



Numerical Solutions to Laplace's Equation $\boxed{\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0}$



- Assume $\Delta x = \Delta y$ [regular square grid]
- Represent $h(x_i, y_j) = h_{i,j}$
- $h(x_i+\Delta x, y_j+\Delta y) = h_{i+1,j+1}$,

Thus replacing terms in Laplace, the F.D.E. becomes

$$h''(x_i, y_j)_x = h''_{i,j} \approx \frac{1}{\Delta x^2}\{h_{i-1,j} - 2h_{i,j} + h_{i+1,j}\}$$

- Do the same for the '*j*' direction, and you have the following:

Approximation to Laplace's Eqn.

$$h''(x_i, y_j)_x + h''(x_i, y_j)_y \approx$$

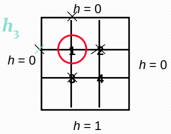
$$\frac{1}{\Delta x^2}\left[\{h_{i-1,j} - 2h_{i,j} + h_{i+1,j}\} + \{h_{i,j-1} - 2h_{i,j} + h_{i,j+1}\}\right] =$$

$$\frac{1}{\Delta x^2}\left[h_{i-1,j} + h_{i,j-1} - 4h_{i,j} + h_{i+1,j} + h_{i,j+1}\right] = 0$$

Summing terms and solving for $h_{i,j}$ gives:

$$\boxed{h_{i,j} = \frac{1}{4}\{h_{i-1,j} + h_{i,j-1} + h_{i+1,j} + h_{i,j+1}\}}$$

Application to a Simple 3x3 Grid



- Start with boundary conditions and initial estimate for h, assume $h_1 = h_2 = h_3 = h_4 = 0$
- Get a new estimate for h_1, call it h_1^{m+1}
- h_1^{m+1} = (top + right + bottom + left)/4
- $h_1^{m+1} = \{0 \quad + \quad h_2 \quad + \quad h_3 \qquad + 0\}/4$
- Repeat four internal points - 2, 3, and 4 using same four-star average calculation

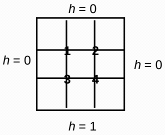
Application to a grid

$h_1^{m+1} = \{0 \quad + h_2^m \quad + h_3^m \quad + 0 \quad \}/4$

$h_2^{m+1} = \{0 \quad + 0 \quad + h_4^m \quad + h_1^m\}/4$

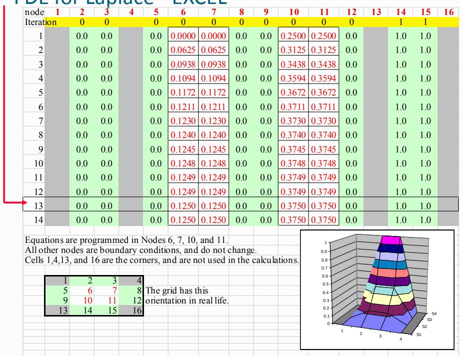
$h_3^{m+1} = \{h_1^m \quad + h_4^m \quad + 1 \quad + 0 \quad \}/4$

$h_4^{m+1} = \{h_2^m \quad + 0 \quad + 1 \quad + h_3^m\}/4$



- Use the initial values and boundary conditions for the first approximation
- Use the updated values (m+1) for further approximations (m+2)
- Continue until the numbers don't change much (convergence!)

FDE for Laplace - EXCEL



node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Iteration		0	0		0	0	0	0	0	0	0	0		1	1	
1		0.0	0.0		0.0	0.0000	0.0000	0.0	0.0	0.2500	0.2500	0.0		1.0	1.0	
2		0.0	0.0		0.0	0.0625	0.0625	0.0	0.0	0.3125	0.3125	0.0		1.0	1.0	
3		0.0	0.0		0.0	0.0938	0.0938	0.0	0.0	0.3438	0.3438	0.0		1.0	1.0	
4		0.0	0.0		0.0	0.1094	0.1094	0.0	0.0	0.3594	0.3594	0.0		1.0	1.0	
5		0.0	0.0		0.0	0.1172	0.1172	0.0	0.0	0.3672	0.3672	0.0		1.0	1.0	
6		0.0	0.0		0.0	0.1211	0.1211	0.0	0.0	0.3711	0.3711	0.0		1.0	1.0	
7		0.0	0.0		0.0	0.1230	0.1230	0.0	0.0	0.3730	0.3730	0.0		1.0	1.0	
8		0.0	0.0		0.0	0.1240	0.1240	0.0	0.0	0.3740	0.3740	0.0		1.0	1.0	
9		0.0	0.0		0.0	0.1245	0.1245	0.0	0.0	0.3745	0.3745	0.0		1.0	1.0	
10		0.0	0.0		0.0	0.1248	0.1248	0.0	0.0	0.3748	0.3748	0.0		1.0	1.0	
11		0.0	0.0		0.0	0.1249	0.1249	0.0	0.0	0.3749	0.3749	0.0		1.0	1.0	
12		0.0	0.0		0.0	0.1249	0.1249	0.0	0.0	0.3749	0.3749	0.0		1.0	1.0	
13		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
14		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	

Equations are programmed in Nodes 6, 7, 10, and 11.
All other nodes are boundary conditions, and do not change.
Cells 1,4,13, and 16 are the corners, and are not used in the calculations.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

The grid has this orientation in real life.

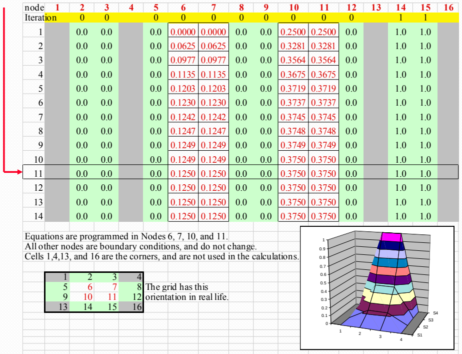
Convergence Criteria

- Convergence for a flow code requires that the change in the solution at each point be less than a specified target, called the convergence criterion, or sometimes epsilon, ε
- If ε is too large, convergence will occur before a solution is reached
- If ε is too small, convergence may take very long, or be impossible due to oscillation
- There is more than one way to define convergence, including global measures, local measures, etc.

Gauss-Seidel Method

- Uses partially completed iteration to estimate values for the rest of the iteration.
- $h_1^{m+1} = \{0 \quad + h_2^m \quad + h_3^m \quad + \quad 0 \quad \}/4$
- $h_2^{m+1} = \{0 \quad + \quad 0 \quad + h_4^m \quad + \quad h_1^m\}/4$
- $h_3^{m+1} = \{\mathbf{h_1^{m+1}} \quad + h_4^m \quad + 1 \quad + 0 \quad \}/4$
- $h_4^{m+1} = \{\mathbf{h_2^{m+1}} \quad + 0 \quad + 1 \quad + h_3^m\}/4$
- Use m+1 update iteration values for h_1 and h_2 when computing h_3 and h_4
- Results in faster convergence over a large grid

Gauss-Seidel



node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Iteration		0	0		0	0	0	0	0	0	0	0		1	1	
1		0.0	0.0		0.0	0.0000	0.0000	0.0	0.0	0.2500	0.2500	0.0		1.0	1.0	
2		0.0	0.0		0.0	0.0625	0.0625	0.0	0.0	0.3281	0.3281	0.0		1.0	1.0	
3		0.0	0.0		0.0	0.0977	0.0977	0.0	0.0	0.3564	0.3564	0.0		1.0	1.0	
4		0.0	0.0		0.0	0.1135	0.1135	0.0	0.0	0.3675	0.3675	0.0		1.0	1.0	
5		0.0	0.0		0.0	0.1203	0.1203	0.0	0.0	0.3719	0.3719	0.0		1.0	1.0	
6		0.0	0.0		0.0	0.1230	0.1230	0.0	0.0	0.3737	0.3737	0.0		1.0	1.0	
7		0.0	0.0		0.0	0.1242	0.1242	0.0	0.0	0.3745	0.3745	0.0		1.0	1.0	
8		0.0	0.0		0.0	0.1247	0.1247	0.0	0.0	0.3748	0.3748	0.0		1.0	1.0	
9		0.0	0.0		0.0	0.1249	0.1249	0.0	0.0	0.3749	0.3749	0.0		1.0	1.0	
10		0.0	0.0		0.0	0.1249	0.1249	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
11		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
12		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
13		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
14		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	

Equations are programmed in Nodes 6, 7, 10, and 11.
All other nodes are boundary conditions, and do not change.
Cells 1,4,13, and 16 are the corners, and are not used in the calculations.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

The grid has this orientation in real life.

Successive Over-Relaxation - SOR

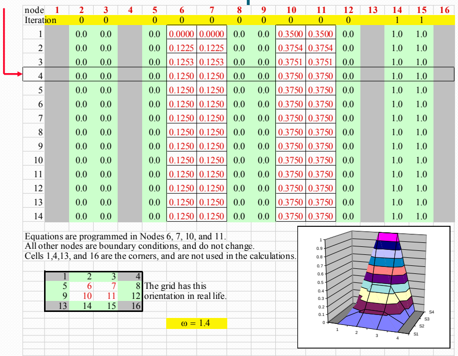
- To speed up the convergence, overshoot the standard model.
- Defining the Residual, $c = h_{i,j}^{m+1*} - h_{i,j}^{m}$
- Using:

$$h_{i,j}^{m+1} = h_{i,j}^{m} + \omega c = (1-\omega)\, h_{i,j}^{m} + \omega h_{i,j}^{m+1*},$$

where ω is the relaxation factor and $h_{i,j}^{m+1*}$ is given by the Gauss-Seidel approximation, we get:

- $h_{i,j}^{m+1} = (1-\omega) h_{i,j}^{m} + (\omega/4)(h_{i-1,j}^{m+1} + h_{i,j-1}^{m+1} + h_{i+1,j}^{m} + h_{i,j+1}^{m})$
- If ω = 1, reduces to Gauss-Seidel
- If 1 < ω< 2, the method is over-relaxed - usually ~ 1.4 - 1.5

SOR - Marked Improvement



node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Iteration		0	0		0	0	0	0	0	0	0	0		1	1	
1		0.0	0.0		0.0	0.0000	0.0000	0.0	0.0	0.3500	0.3500	0.0		1.0	1.0	
2		0.0	0.0		0.0	0.1225	0.1225	0.0	0.0	0.3754	0.3754	0.0		1.0	1.0	
3		0.0	0.0		0.0	0.1253	0.1253	0.0	0.0	0.3751	0.3751	0.0		1.0	1.0	
4		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
5		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
6		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
7		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
8		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
9		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
10		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
11		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
12		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
13		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	
14		0.0	0.0		0.0	0.1250	0.1250	0.0	0.0	0.3750	0.3750	0.0		1.0	1.0	

Equations are programmed in Nodes 6, 7, 10, and 11.
All other nodes are boundary conditions, and do not change.
Cells 1,4,13, and 16 are the corners, and are not used in the calculations.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

The grid has this orientation in real life.

$\omega = 1.4$

Numerical Simulation Models

Numerical models solve approximations of the governing equations of flow and transport - over a grid system

- Based on a grid or mesh laid over the study site
- Helps organize large datasets into logical units
- Data required includes boundary conditions, hydraulic conductivity, thickness, pump rates, recharge, etc.)
- Can be computationally intensive
- Can be applied to actual field sites to help understand complex hydrogeologic interrelationships.

Often-Used Numerical Models

MODFLOW (1988) - USGS flow model for 3-D aquifers
MODPATH - flow line model for depicting streamlines

MOC (1988) - USGS 2-D advection/dispersion code

MT3D (1990, 1998) - 3-D transport code works with MODFLOW

RT3D (1998) - 3-D transport chlorinated - MODFLOW

BIOPLUME II, III (1987, 1998) - authored at Rice Univ 2-D based on the MOC procedures.

Numerical Solution of Equations

Numerically -- *H* or *C* is approximated at each point of a computational domain (may be a regular grid or irregular)

- Solution is very general
- May require intensive computational effort to get the desired resolution
- Subject to numerical difficulties such as convergence problems and numerical dispersion
- Generally, flow and transport are solved in separate independent steps (except in density-dependent or multi-phase flow situations)

MODFLOW Introduction

- Written in the 1984 and updated in 1988
- Solves governing equations of flow for a full 3-D aquifer system with variable K, b, recharge, drains, rivers, and pumping wells.
  - Withdrawal and injection wells (rates may change with time)
  - Constant head boundaries or regions (ponds/rivers/fixed heads)
  - No-flow boundaries or regions (bedrock outcrops/water divides)
  - Regions of diffuse recharge or discharge (rainfall)
  - Observation wells

370 NUMERICAL MODELING OF CONTAMINANT TRANSPORT

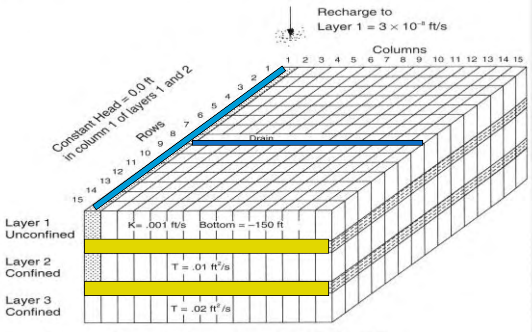


Figure 10.7 Schematic of an aquifer system in MODFLOW.

MODFLOW

MODFLOW is a modular 3-D finite-difference flow code developed by the U.S. Geological Survey to simulate saturated flow through a layered porous media. The PDE solved is for ***h(x,y,z,t):***

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_{yy}\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(K_{zz}\frac{\partial h}{\partial z}\right)-W=S_s\frac{\partial h}{\partial t}$$

- where K_{xx}, K_{yy}, and K_{zz} are defined as the hydraulic conductivity along the *x*, *y*, and *z* coordinate axis, *h* represents the potentiometric head, *W* is the volumetric flux per unit volume being pumped, S_s is the specific storage of the porous material and *t* is time.

MODFLOW Features

- MODLOW consists of a main program and a series of independent subroutines grouped into packages.
- Each package controls with a specific feature of the hydrologic system, such as wells, drains, and recharge. The division of the program into packages allows the user to analyze the specific hydrologic feature of the model independently.

MODFLOW Features

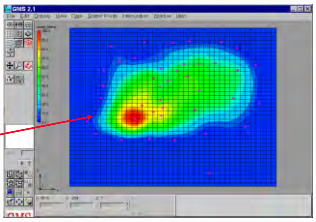
- MODFLOW is one of the most versatile and widely accepted groundwater models
- It is particularly good in heterogeneous regions because it allows for vertical interchange between layers, as well as horizontal flow within the aquifers.
- It also allows for variable grids to speed computation.
- It has been applied to model thousands of field sites containing a number of different contaminants and for a number of different remediation applications.

Solution Methods

- MODFLOW is an iterative numerical solver (SIP or SOR).
- The initial head values are provided and the these heads are gradually changed through a series of time steps, in the case of a transient model run, until the governing equation is satisfied. Time steps can be variable to speed output.
- The primary output from the model is the head distribution in x, y, and z, which can then be used by a transport model.
- In addition, a volumetric water budget is provided as a check on the numerical accuracy of the simulation.

MODFLOW - Input to Transport Models

- Designed to create modern GUI to ease large data entry and output graphical manipulation for applications to complex field sites
  - GMS - 1995
  - Visual MODFLOW
  - Ground Water Vistas

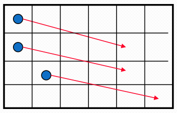


PLUME visualization

MODPATH - Pathlines

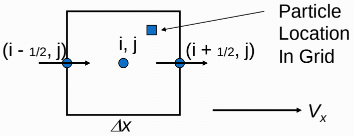
- Designed to use heads from MODFLOW and linear interpolation to compute velocity V_x and or V_y.
- Particles can be placed in areas of known or suspected source concentrations in order to create possible tracks of contaminants in space and time - streaklines

Path results after two time steps

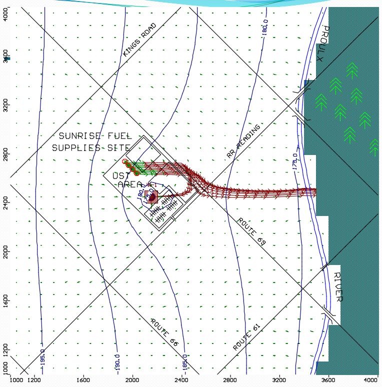


MODPATH

- Designed to use heads from MODFLOW and linear interpolation to compute velocity V_x in the x direction:
- $V_x = (1\text{-}f_x)V_{x(i\,-1/2)} + f_x V_{x(i\,+1/2)}$
- Where $f_x = (x_p - x_{(i-1/2)})/\,\Delta x_{i,j}$
- And x_p is the x coordinate of the particle



Pathlines



Contaminant Transport in 2-D

$$\frac{\partial CB}{\partial t} = \frac{\partial}{\partial x_I}\left(BD_{IJ}\frac{\partial C}{\partial x_J}\right) - \frac{\partial}{\partial x_I}\left(BCV_I\right) - \frac{C'W}{E}$$

C = Concentration of Solute [M/L^3]

D_{IJ} = Dispersion Coefficient [L^2/T] - x and y

B = Thickness of Aquifer [L]

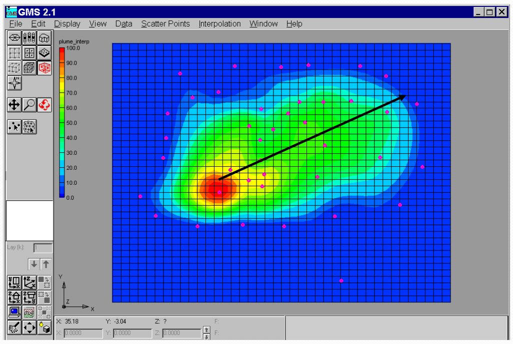
C' = Concentration in Sink Well [M/L^3]

W = Flow in Source or Sink [L^3/T]

E = Porosity of Aquifer [unitless]

V_I = Velocity in 'I' Direction [L/T] - x and y

2-D CONTAMINANT TRANSPORT



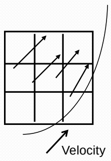
Domenico and Schwartz (1990)

- 3-D solutions for several geometries (listed in Bedient et al. 1999, Section 6.8) - spreadsheets exist
- Generally a vertical plane, constant concentration source.

$$\frac{C(x,y,z,t)}{C_0} = \left(\frac{1}{8}\right)\operatorname{erfc}\left[\frac{x-vt}{2\sqrt{\alpha_x vt}}\right] \left\{\operatorname{erf}\left[\frac{y+Y/2}{2\sqrt{\alpha_y x}}\right] - \operatorname{erf}\left[\frac{y-Y/2}{2\sqrt{\alpha_y x}}\right]\right\} \left\{\operatorname{erf}\left[\frac{z+Z/2}{2\sqrt{\alpha_z x}}\right] - \operatorname{erf}\left[\frac{z-Z/2}{2\sqrt{\alpha_z x}}\right]\right\}$$

Method of Characteristics USGS (MOC)

- Written for USGS by Konikow and Bredehoeft in 1978
- Solves flow equations with Alternating Direction Implicit (ADI) method
- Solves transport equations via particle tracking method and finite difference
- Using velocities calculated from the flow solution, $v_x = k_x\,(\Delta h/\Delta x)$, particles are moved
- Concentrations are based on the average concentration of all particles in a cell at the end of the time step



MOC Concepts

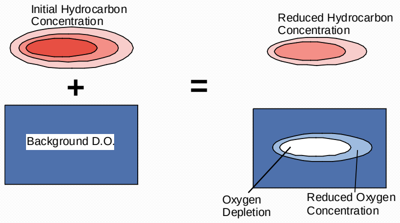
- Partial Differential Equation replaced by equivalent set of ODEs called 'characteristic equations' (approximated with finite difference)
- Particle in a cell is moved a distance proportional to the seepage velocity within the cell
  - Accounts for concentration change due to advection
- Remainder of governing equation solved by finite difference methods
  - Accounts for concentration change due to dispersion, changes in saturated thickness, and fluid sources

MOC/BIOPLUME II Capabilities

- Can simulate effects of natural or enhanced biodegradation:
  - Fast-equilibrium biodegradation
  - First-order biodegradation
  - Monod kinetics
  - Withdrawal and injection wells (pump and treat systems)
  - Injection wells for oxygen enhancement
  - Observation wells within and outside plume area

BIOPLUME II Concepts

- Can simulate effects of natural or enhanced biodegradation
- Adjustment is made at each time step in numerical grid



Initial Hydrocarbon Concentration

+

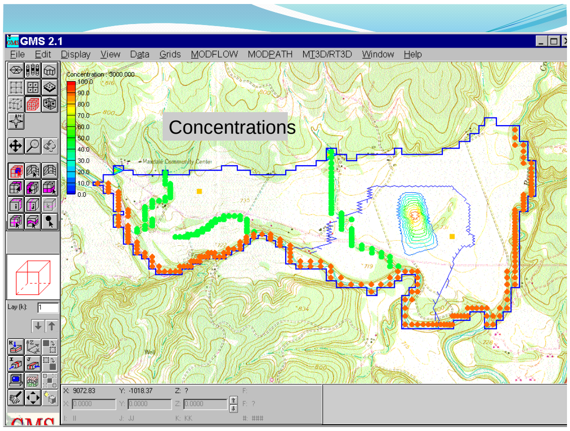
Background D.O.

=

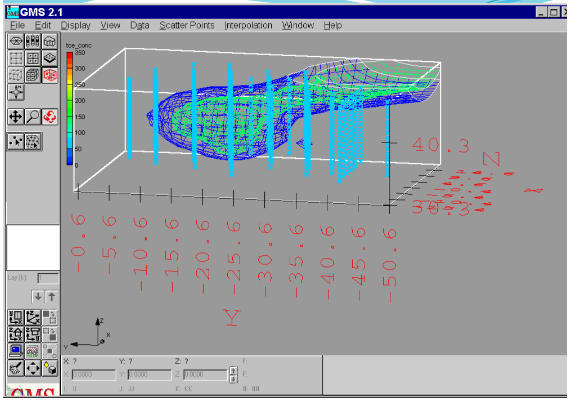
Reduced Hydrocarbon Concentration

Oxygen Depletion

Reduced Oxygen Concentration



Concentrations



Calibration, Validation, and Sensitivity Analysis

- **Calibration** is the process of making the model match real-world data. Involves making several model runs, varying parameters until the 'best fit' is achieved.
- **Validation** is the process of confirming the validity of your calibration by using the model to fit an independent set of data.
- **Sensitivity Analysis** is the process of changing parameters to see the effects on the model results. The most sensitive parameters need to be checked for accuracy to ensure the best model.