SCIENTIFIC NOTEBOOK
#364E

by

Chris Hackert

Southwest Research Institute
Instrumentation and Space Research Division
San Antonio, Texas

August 1, 1999–October 29, 1999
Volume 1

July 1, 2000–September 12, 2000
Volume 2

SCIENTIFIC NOTEBOOK
#364E, Volume 1

by

Chris Hackert

Southwest Research Institute
Instrumentation and Space Research Division
San Antonio, Texas

August 1, 1999–October 29, 1999

SCIENTIFIC NOTEBOOK

by

Chris Hackert

Southwest Research Insitute
Instrumentation and Space Research Division
San Antonio, Texas

August 1, 1999 – October 29, 1999

Table of Contents

INITIAL ENTRIES

Scientific notebook: #364E Vol. 1
Issued to: Chris Hackert
Issue Date: August 3, 1999

This notebook contains some material generated before the official issue date. This notebook will be printed at approximately quarterly intervals. This computerized electronic notebook is intended to address the criteria of CNWRA QAP-001.

Development of Multiflo code benchmarks
Account number: 20-1402-562
Project manager: S. Painter, CNWRA
Objective: Create benchmarks to test the METRA module of Multiflo.

I have previously received the METRA component of Multiflow 1.2β from Scott Painter. This program was compiled without change on the HP C110 workstation terra, using the suggested HP modifications to the makefile.

July 29, 1999

2D steady flow and steady heat transfer in a 2D Cartesian geometry

This benchmark covers fully saturated isothermal flow in a 2D Cartesian geometry, and fully saturated isobaric heat transfer in the same geometry. The solution to each problem (assuming constant uniform properties) is the same, and is a solution of the 2D Laplace equation. Taking the pressure or temperature as a nondimensional potential ϕ, boundary conditions are given by $\phi = 0$ at $x = 0$, $x = L$, and $y = 0$. At $y = W$, $\phi = 1$. The solution to this problem is given in Incropera and DeWitt, *Fundamentals of Heat and Mass Transfer* as

$$\phi(x,y) = \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}+1}{n}\sin\frac{n\pi x}{L}\frac{\sinh(n\pi y/L)}{\sinh(n\pi W/L)}.$$

Fluxes may obtained through differentiation in the appropiate direction.

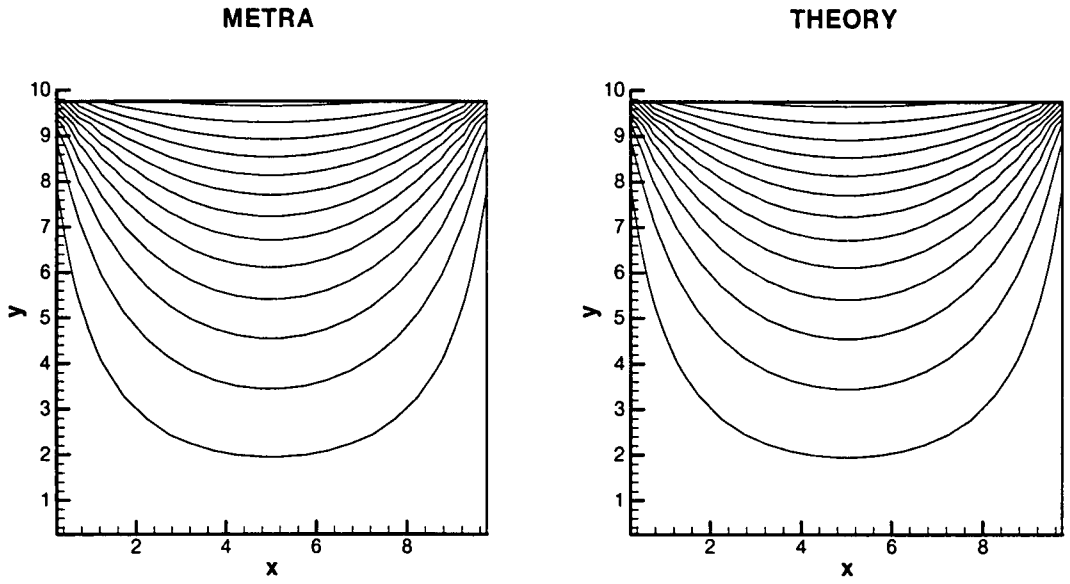
The METRA solution is obtained on a 20 x 20 grid using the following input file

```
METRA test: fully saturated : 2d
                  July 21, 1999
:
RSTART 0
:
:grid  geometry  nx  ny  nz ivplwr ipvcal iout gravity pref tref  href
GRID   XYZ        20 20  1    1      1      3    0      0    0    0
:
:LIQUID : comment this out to check saturated problem as limiting
:        situation of 2-phase problem. should give same result
:
Pckr                            :relative perm and pc keyword
:   i  type-curv swirm  rpmm(lamda)  alpham  swext  sgc  iecm
:
    1  Van-Gen   0.01   0.63    9.7e-5      0.    0.  0
/
:
Thermal-prop
: no rho       cpr   ckdry  cksat    crp  crt   tau  cdiff   cexp  enbd
  1  2.510e+3  847.    1.02   1.76     0     0    .5  2.13e-5  1.8  0.
0
:     igrid   rw      re
DXYZ   0
: (dx(i),i=1,nx)
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
```

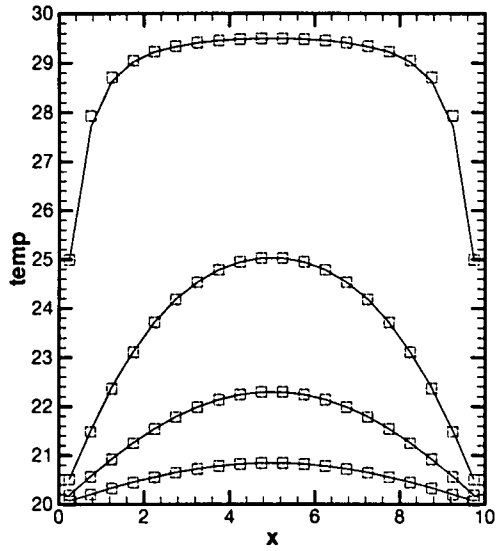
```
:
: (dy(j),j=1,ny)
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
:
: (dz(k),k=1,nz)
1.0
:
PhiK
: i1 i2  j1  j2  k1  k2  iist ithrm  vb porm   permx   permy    permz
  1  20   1  20   1  1   1       1   0.  0.1   1.e-13   1.e-13   1.e-13
   0   0
:
Init
:
: i1  i2  j1  j2  k1  k2         p           t           sg           xg2
pm
   tm          sgm       xgm
:
   1   20   1   20 1   1  101000   20.0000 0.9  000E+00
   0   0
:
Recurrent-data
Bcon  4
:itype iface i1 i2 j1 j2
:time qbc   pbc     tbc   sgbc   xabc
1   LEFT    1   20   1   1
  0.  0.     101000      20.0   0.    0.
.
1   RIGHT   1   20   1   1
  0.  0.     101000      20.0   0.0    0.
.
1   FRONT   1   20   1   1
  0.  0.     101000      20.0   0.0    0.
.
1   BACK   1   20   1   1
  0.  0.     101400      20.0   0.0    0.
.
:
Rstart 1 0
Output   A=1
:  isolve newtnmn newtnmx north nitmx level
Solve 4     1    12    2    100   1
:
:AUTO-step  DPMXE    DSMXE  DTMPMXE   DP2MXe
AUTO-step   1.0E+3   0.04    1.       1.e3
:
:TOLR TOLP  TOLS  TOLT  TOLP2  TOLM  TOLA  TOLE rtwotol rmxtol smxtol
Tolr   0.1 1.e-4  1.e-3  1.e-1 1.e-8 1.e-8 1.e-6 1.e-25 1.e-25 1.e-25
:
:Limit dpmx    dsmx dtmpmx dp2mx dtmn dtmx icutmx
LIMIT  5.e3    .1    3.   .5e4  1.e-8  1.e5 .334
Plots 1 1 1
35
Steady[y] 1.e-7 1.e-7  1.e-7
Ends
```

Identical results to 5 significant figures are obtained if the LIQUID keyword is not commented out. The isobaric heat transfer case is identical except that the BACK boundary condition has pressure 101000 and temperature 30.

Results for isobaric heat transfer:

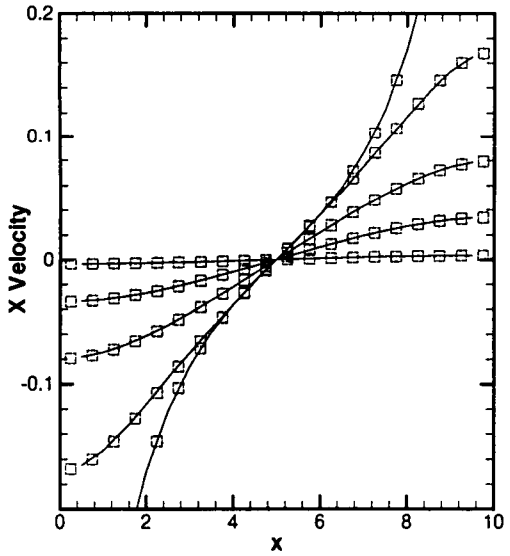


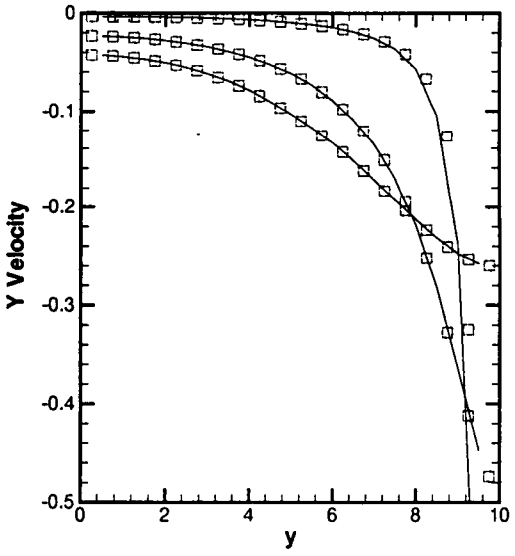
This graph compares the 2D temperature field obtained by METRA to that obtained using the series solution (above) to 40 nonzero terms. A more direct comparison is shown below, where the solid lines represent the METRA solution and the squares represent the series solution. Agreement is good. Each line represents a spacing of 2.5 meters, starting from y=0.25 meters.



Results for isothermal flow:

Due to roundoff, the pressure field output by METRA is choppy. Therefore, a comparison is made of the velocity fields. The x-direction and y-direction velocities (in m/year) are shown below. The solid lines represent the METRA solution and the squares represent the series solution obtained with 40 nonzero terms.





Results are presented for lines spaced every 2 m for the x velocity graph, and 4 m for the y velocity graph. There is again good agreement between theory and METRA result,

deteriorating slightly near the upper corners, which are singular points having very steep gradients.

August 5, 1999

DCM STEADY FLOW

Theoretical solution

In the homogeneous steady DCM formulation, the pressure equations are the following (where f denotes fracture, m denotes matrix, ϕ is the pressure, K is the hydraulic conductivity (permeability divided by viscosity), and C is the connectivity between fracture and matrix):

$$K_f \nabla^2 \phi_f = -C(\phi_f - \phi_m)$$
$$K_m \nabla^2 \phi_m = C(\phi_f - \phi_m)$$

The coefficient C is given by

$$C = K_m A_{fm} \frac{2}{\langle \ell \rangle},$$

in the limit of matrix permeability much less than intrinsic fracture permeability and fracture aperture much less than fracture spacing. Here, A_{fm} is the fracture area per unit volume, and $\langle \ell \rangle$ is the average fracture spacing.

In such a situation, we can define new variables $\phi_t = K_f\phi_f + K_m\phi_m$, and $\phi_d = \phi_f - \phi_m$. Then, $\phi_m = (\phi_t - K_f\phi_d)/(K_m + K_f)$ and $\phi_f = (\phi_t + K_m\phi_d)/(K_m + K_f)$.

By adding the two pressure equations, we get

$$\nabla^2 \phi_t = 0.$$

By subtracting the two pressure equations, we get

$$\nabla^2 (K_m \phi_m - K_f \phi_f) = 2C(\phi_f - \phi_m),$$

which, when transformed to the new variables is

$$\frac{K_m - K_f}{K_m + K_f} \nabla^2 \phi_t - \frac{2K_f K_m}{K_m + K_f} \nabla^2 \phi_d = 2C\phi_d.$$

But since we have already established that the Laplacian of ϕ_t is zero, we have the inhomogeneous equation

$$\nabla^2\phi_d = -\frac{C(K_m + K_f)}{K_m K_f}\phi_d .$$

This makes sense, because the potential difference between matrix and fracture acts to reduce itself through the constant C.
Solutions to the homogeneous equation for ϕ_t are simple, and one has been given earlier in the discussion of steady Cartesian flow. Solutions to the inhomogeneous equation may also be found by the method of separation of variables. For simplicity, consider a case similar to the earlier example. Take a Cartesian isotropic DCM grid defined over x from 0 to L, and y from 0 to L. Apply boundary conditions such that the nondimensionalized pressure $\phi_d = 0$ at $x = 0$, $x = L$, and $y = 0$. At $y = L$, $\phi_d = 1$.

If $\phi_d = X(x)Y(y)$, then

$X''(x)Y(y) + X(x)Y''(y) = BX(x)Y(y)$

where

$$B = -C\frac{K_f + K_m}{K_f K_m}.$$

This equation can be separated so that

$X'' = -\lambda^2 X$, and
$Y'' = (A + \lambda^2)Y$.

The solutions of X are $\sin(\lambda x)$ and $\cos(\lambda x)$, while the solutions of Y are $\sinh[(A+\lambda^2)^{1/2}y]$ and $\cosh[(A+\lambda^2)^{1/2}y]$. Applying the boundary conditions allows the reuse of the series coefficients from the earlier example:

$$\phi_d = \sum_{n=1}^{\infty} \frac{2[1-(-1)^n]}{\pi n}\sin(\lambda_n x)\frac{\sinh\left(y\sqrt{B+\lambda_n^2}\right)}{\sinh\left(L\sqrt{B+\lambda_n^2}\right)},$$

where

$$\lambda_n = \frac{n\pi}{L}.$$

Metra Solution

In order to force a steady flow in a DCM medium, different matrix and fracture pressures are set along one boundary of a Cartesian 2D box. In order to record noticably different matrix and fracture pressures, the coupling area was reduced by a factor of 10^4. A fairly artificial test case is chosen, in which both matrix and fracture pressures on three sides are set equal, to the value 101000 Pa. On the fourth side, the matrix pressure is reduced to 100000 Pa, while the fracture pressure is increased to 102000 Pa. In this case the matrix permeability is equal to the bulk fracture permeability. The advantage of this artificial test case is that the combined pressure, $\phi_t = K_f P_f + K_m P_m$, is everywhere uniform and constant. This provides a quick and accurate check for one of the two independent fields.

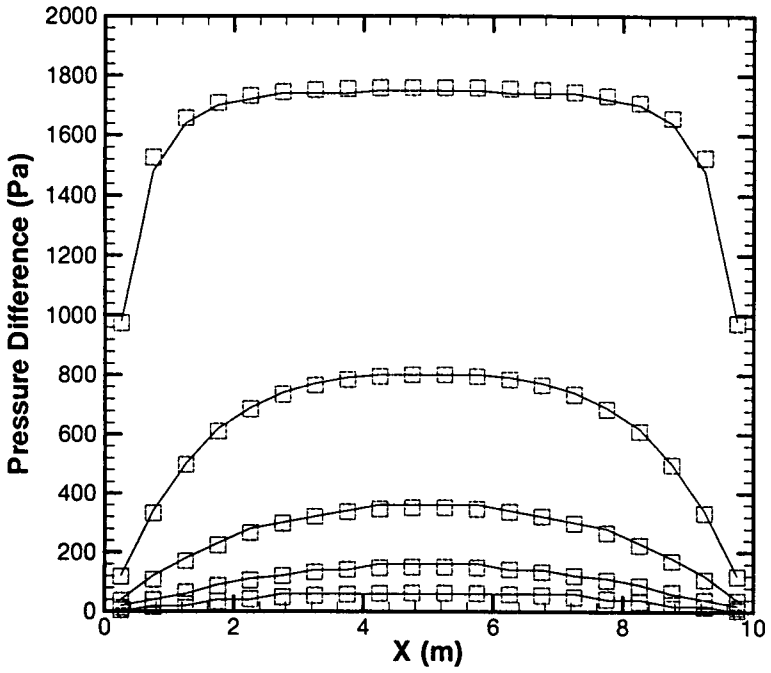
The METRA input file is below:

```
METRA test: DCM steady : 2d
                    July 29, 1999
:
RSTART 0
:
:grid  geometry  nx  ny  nz ivplwr ipvcal iout gravity pref tref  href
GRID   DCMXYZ     20 20  1    1       1      3    0      0    0    0
:
DCMParameters
: I1  I2  J1  J2  K1  K2  sigmaf  areamod  xlm  ylm  zlm
  1   20   1  20  1   1   0.01     1.e-4    0.1  0.1  0.1
/
:
Pckr                              :relative perm and pc keyword
:   i  type-curv swirm  rpmm(lamda)  alpham  swext  sgc  iecm
:
    1  Van-Gen   0.01   0.63    9.7e-5       0.     0.  0
/
:
Thermal-prop
: no rho       cpr   ckdry  cksat    crp  crt   tau  cdiff   cexp  enbd
  1  2.510e+3  847.    1.02   1.76     0     0    .5  2.13e-5  1.8  0.
0
:     igrid   rw      re
DXYZ   0
: (dx(i),i=1,nx)
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
:
: (dy(j),j=1,ny)
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
:
: (dz(k),k=1,nz)
1.0
:
PhiK
: i1 i2  j1  j2  k1  k2  iist ithrm  vb por   permx   permy    permz
: porm  permm    istm  ithermm
  1  20   1  20   1  1   1       1    0.  0.9   1.e-11   1.e-11   1.e-11
0.1 1.e-13 1 1
/
:
Init
:
```

```
: i1  i2  j1  j2  k1  k2      p      t      sg    xa       pm   tm   sgm xgm
:
   1   20   1   20 1   1  101000   20.0 0.9  0.0  101000  20. 0.9 0.
/
:
Recurrent-data
Bcon  4
:itype iface i1 i2 j1 j2
:time qbc    pbc      tbc    sgbc    xabc
:time qbcf    pbcf      tbcf    sgbcf    xabcf
1    LEFT     1   20    1   1
  0.  0.      101000       20.0    0.     0.
  0.  0.      101000       20.0    0.     0.
.
1    RIGHT    1   20    1   1
  0.  0.      101000       20.0    0.0     0.
  0.  0.      101000       20.0    0.0     0.
.
1    FRONT    1   20    1   1
  0.  0.      101000       20.0    0.0     0.
  0.  0.      101000       20.0    0.0     0.
.
1    BACK    1   20    1   1
  0.  0.      100000       20.0    0.0     0.
  0.  0.      102000       20.0    0.0     0.
.
:
Rstart 1 0
Output   A=1
:  isolve newtnmn newtnmx north nitmx level
Solve 4      1     12     2     100    1
:
:AUTO-step  DPMXE     DSMXE  DTMPMXE   DP2MXe
AUTO-step   1.0E+3    0.04     1.        1.e3
:
:TOLR TOLP  TOLS  TOLT  TOLP2  TOLM  TOLA  TOLE rtwotol rmxtol smxtol
Tolr   0.1 1.e-4  1.e-3  1.e-1 1.e-8 1.e-8 1.e-6 1.e-25 1.e-25 1.e-25
:
:Limit dpmx     dsmx dtmpmx dp2mx dtmn dtmx icutmx
LIMIT  5.e3     .1     3.    .5e4  1.e-8  1.e5 .334
Plots 1 1 1
35
Steady[y] 1.e-7 1.e-7  1.e-7
Ends
```

Comparison of METRA and theory

As expected, the total pressure $P_f + P_m$ is constant and equal to 202000 Pa, to within the limits of precision of the METRA output. The pressure difference $P_f - P_m$ matches the theory solution quite well as shown in the graph below. For this case, L = 10, and the solid line is the METRA solution while the squares are the theoretical series solution. Because of roundoff in the output, the METRA solution is only accurate to within 10 Pa.



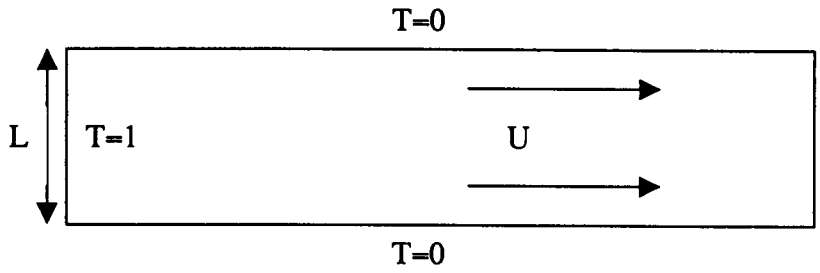
The theoretical solution is obtained using 40 nonzero terms of the series expansion. The lines shown in the graph are constant y, and have the values y = 9.75, 8.25, 6.75, 5.25, and 3.75 (moving from the top down).

August 17, 1999

Thermal diffusion into flow

Theory

Consider a 2-D rectangular porous medium. The top and bottom boundaries are impermeable. Uniform flow with velocity U crosses the left and right boundaries. The top and bottom boundaries have a fixed temperature which is higher than the inlet temperature condition on the left wall. Therefore, heat transfer occurs from the top and bottom surfaces and heat is advected and conducted through the porous medium. The solution uses a scaled temperature, in which the inlet has temperature 1, and the top and bottom surfaces have temperature 0.



The steady-state differential equation governing heat conduction and advection to be solved is

$$k\nabla^2 T = U\frac{dT}{dx},$$

Because the temperature is assumed not to affect the flow, the flow is uniform everywhere and can be satisfied by constant flux boundary conditions. It is also assumed that the thermal diffusivity k is everywhere constant. Applying the method of separation of variables, with the assumption that T(x,y) = X(x)Y(y), leads to the independent equations

$$Y'' = -\lambda^2 Y$$

$$X'' - \frac{U}{k}X' - \lambda^2 X = 0$$

The first has solutions of $\sin(\lambda y)$ and $\cos(\lambda y)$, while the second has solutions of $\exp(m_1 x)$ and $\exp(m_2 x)$ (*CRC Standard Math Tables*). Here, m_1 and m_2 are

$$m_{1,2} = \frac{1}{2}\left(\frac{U}{k} \pm \sqrt{\frac{U^2}{k^2} + 4\lambda^2}\right).$$

Given that U and k are both positive, it can be seen that $m_1 > 0$ and $m_2 < 0$. To satisfy the boundary conditions that T(x,0)=0 and T(x,L)=0, only sin terms are allowed in Y(y), and λ is restricted to $n\pi/L$. To satisfy the boundary condition that $T(x \to \infty, y) = 0$, only the solution with $m_2 < 0$ is allowed.

So, the solution has the form

$$T(x,y) = \sum_n C_n \sin\frac{n\pi y}{L}\exp\left(\frac{x}{2}\left(\frac{U}{k} - \sqrt{\frac{U^2}{k^2} + 4\frac{n^2\pi^2}{L^2}}\right)\right).$$

The C_n are chosen to satisfy the final boundary condition, that T(0,y) = 1. These coefficients are the same as those for a square wave of period 2*L* (Boyce and DiPrima, *Elementary Differential Equations and Boundary Value Problems*):

$$C_n = \frac{4}{\pi n}; \quad n \text{ odd}$$
$$C_n = 0; \quad n \text{ even}$$

METRA Implementation

This configuration is set up on an 80 by 40 grid, with a length of 7.85 meters and height of 1 meter. The inlet temperature is 22 degrees Celsius, with boundaries of 20 degrees C. It was found that assigning uniform flow boundary conditions to the inlet and outlet resulted in instability, a condition not uncommon in flow modeling. This was alleviated by switching to constant pressure boundaries at the inlet and outlet, with a 500 Pa pressure difference. For constant viscosity conditions this will generate a uniform flow in constant permeability media. In practice, the viscosity of water varies by about 5% over the temperature range from 20 to 22 degrees C. The effect of this on the flow field is about the same magnitude, so that the flow is uniform to within +/- 2% of the mean value.

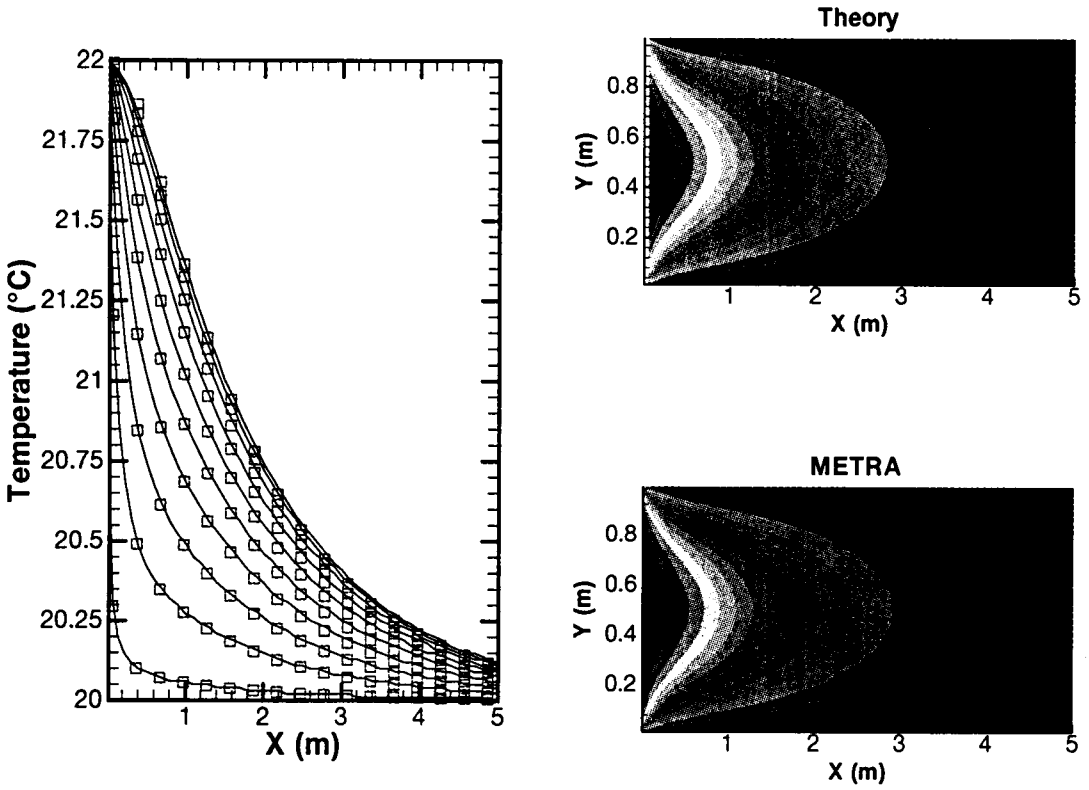
The input file for METRA is below:

```
METRA test: fully saturated : 2d
                  July 21, 1999
:
RSTART 0
:
:grid  geometry  nx  ny  nz ivplwr ipvcal iout gravity pref tref  href
GRID   XYZ        80 40  1    1       1      3    0       0    0    0
:
:
Pckr                              :relative perm and pc keyword
:   i  type-curv swirm  rpmm(lamda)  alpham  swext  sgc  iecm
:
    1  Van-Gen   0.01   0.63    9.7e-5       0.    0.  0
/
:
Thermal-prop
: no rho       cpr   ckdry  cksat    crp  crt   tau  cdiff   cexp  enbd
  1  2.510e+3  847.    1.02   1.76     0     0    .5  2.13e-5  1.8  0.
0
:     igrid   rw      re
DXYZ   0
: (dx(i),i=1,nx)
0.05 0.06 0.07 0.08 0.09 0.10 0.10 0.10 0.10 0.10
0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10
0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10
0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10
0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10
0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10
0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10
```

```
0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10
:
: (dy(j),j=1,ny)
0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025
0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025
0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025
0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025
:
: (dz(k),k=1,nz)
1.0
:
PhiK
: i1 i2  j1  j2  k1  k2  iist ithrm  vb porm   permx   permy    permz
  1  80   1  40   1  1   1       1   0.  0.1   1.e-10   1.e-10   1.e-10
   0   0
:
Init
:
: i1  i2  j1  j2  k1  k2         p          t          sg          xg2
pm         tm          sgm       xgm
:
   1   80   1   40 1   1  101500   21.0000 0.9  000E+00
/
:
Recurrent-data
Bcon  4
:itype iface i1 i2 j1 j2
:time qbc   pbc     tbc   sgbc   xabc
1   LEFT    1   40   1   1
  0.  .0     101500      22.0   0.    0.
.
1   RIGHT   1   40   1   1
  0.  0.0    101000      20.0   0.0    0.
.
1   FRONT   1   80   1   1
  0.  0.     -101000      20.0   0.0    0.
.
1   BACK   1   80   1   1
  0.  0.     -101000      20.0   0.0    0.
.
:
Rstart 1 0
Output   A=1
:  isolve newtnmn newtnmx north nitmx level
Solve 4     1    12    2    100   1
:
:AUTO-step  DPMXE    DSMXE  DTMPMXE   DP2MXe
AUTO-step   1.0E+3   0.04    1.       1.e3
:
:TOLR TOLP  TOLS  TOLT  TOLP2  TOLM  TOLA  TOLE rtwotol rmxtol smxtol
Tolr   0.1 1.e-4  1.e-3  1.e-1 1.e-8 1.e-8 1.e-6 1.e-25 1.e-25 1.e-25
:
:Limit dpmx    dsmx dtmpmx dp2mx dtmn dtmx icutmx
LIMIT  5.e3    .1    3.   .5e4  1.e-8  1.e5 .334
Plots 1 1 1
35
Steady[y] 1.e-7 1.e-7  1.e-7
Ends
```

Comparison of theory to METRA result

The critical factor U/k is computed to be 15.1. The theoretical solution is summed using 40 nonzero terms. Results of the theory (red squares) and METRA (black lines) are shown in the line plot and two field plots below. Each line in the line plot represents a line of constant y, ranging from near the wall to near the midplane at intervals of approximately 0.05.



The comparison between METRA and theory is in general quite good. The simulation is actually continued out to x = 7.85 meters, so that the outlet edge does not significantly affect the result. The tendency of the METRA result to show slightly higher temperatures than the theory may be explained by the change in viscosity with temperature. The viscosity of water increases with decreasing temperature, so the viscosity is highest near the impermeable, cold boundaries. This decreases the flow rate near the thermal boundary which lowers the heat transfer rate. Conversely, the flow rate near the centerline is increased, which allows warm fluid to penetrate farther from the inlet region.

August 20, 1999

Transient DCM solution in radial coordinates

Background

This is intended to be a DCM version of the Theis problem. Specifically, an overpressured DCM aquifer is pumped from fractures at a central well. The matrix pores are not connected directly to the well, and are drained through their connection to the fracture system.

The transient equations governing the behavior of this system are

$$\left(1-\sigma_f\right)\phi_m \frac{1}{c^2}\frac{\partial P_m}{\partial t} - \left(1-\sigma_f\right)\frac{\kappa_m \rho}{\mu}\nabla^2 P_m = \frac{\kappa_m \rho}{\mu}\frac{2}{\langle L\rangle} A_{fm}\left(P_f - P_m\right)$$

$$\sigma_f \phi_f \frac{1}{c^2}\frac{\partial P_f}{\partial t} - \sigma_f \frac{\kappa_f \rho}{\mu}\nabla^2 P_f = -\frac{\kappa_m \rho}{\mu}\frac{2}{\langle L\rangle} A_{fm}\left(P_f - P_m\right)$$

where σ_f is the fractional fracture volume, $\phi_{f,m}$ are the intrinsic fracture and matrix porosities, $\kappa_{f,m}$ are the intrinsic fracture and matrix permeabilities, c is the speed of sound in water (related to the compressibility), μ is the viscosity of water, ρ is the density of water, L is the fracture block spacing, A_{fm} is the fracture-matrix interfacial area per unit volume, and $P_{f,m}$ are the fracture and matrix pressures. To simplify matters a little, write this as

$$D\frac{\partial P_m}{\partial t} - A\nabla^2 P_m = C\left(P_f - P_m\right)$$

$$E\frac{\partial P_f}{\partial t} - B\nabla^2 P_f = -C\left(P_f - P_m\right)$$

In order to solve this system, make the specific assumptions that D = E and A = B. This is equivalent to specifying that the bulk fracture porosity is equal to the bulk matrix porosity, and that the bulk fracture permeability is equal to the bulk matrix permeability. Under these conditions, define new variables

$\theta_t = P_m + P_f$
$\theta_d = P_m - P_f$.

The new variables are described by the independent differential equations

$$\frac{\partial \theta_t}{\partial t} - \frac{A}{D}\nabla^2\theta_t = 0$$

$$\frac{\partial \theta_d}{\partial t} - \frac{A}{D}\nabla^2\theta_d = -2\frac{C}{D}\theta_d$$

The boundary conditions in the original variables are that pumping at a set rate occurs from the fractures at a cylindrical well. At time zero or large distances, the aquifer is undisturbed. The boundary conditions at the well therefore transform to

$$A\frac{\partial P_m}{\partial r}\bigg|_{r=r_o} = 0 \qquad A\frac{\partial \theta_t}{\partial r}\bigg|_{r=r_o} = q_o$$
$$A\frac{\partial P_f}{\partial r}\bigg|_{r=r_o} = q_o \quad \Rightarrow \quad A\frac{\partial \theta_d}{\partial r}\bigg|_{r=r_o} = -q_o$$

To solve the equation for θ_t in radial coordinates, presume that θ_t is a function only of η, which is defined to be

$$\eta = \frac{r}{\sqrt{\alpha t}}$$

where α = A/D. The differential equation is then

$$-\frac{\eta}{2t}\frac{\partial \theta_t}{\partial \eta} - \frac{\alpha}{\alpha t}\frac{1}{\eta}\left(\eta\frac{\partial \theta_t}{\partial \eta}\right) = 0$$

or

$$\theta_t'' + \left(\frac{\eta}{2} + \frac{1}{\eta}\right)\theta_t' = 0.$$

Apparently,

$$\theta_t' = \frac{1}{\eta}e^{-\eta^2/4}.$$

Integrating with respect to η yields

$$\theta_t = \frac{1}{2}\left[\ln\left(\frac{\eta^2}{4}\right) - \frac{\eta^2}{4} + \left(\frac{\eta^2}{4}\right)^2\frac{1}{4} + \cdots + \left(-\frac{\eta^2}{4}\right)^n\frac{1}{(n)(n!)}\right].$$

Any additive or multiplicative constant may also be introduced, since the equation for θ_t is linear.

The equation for θ_d doesn't look so bad, but I haven't been able to find a solution for it in the transient form. Fortunately, the behavior of θ_d fairly rapidly reaches a steady state. The sink term proportional to C damps the solution, and after the spatial distribution equilibrates with the boundary condition the solution is not time-dependent. So, the steady state solution in radial coordinates is solved. This is

$$\frac{\partial^2\theta_d}{\partial r^2}+\frac{1}{r}\frac{\partial\theta_d}{\partial r}=2\frac{C}{A}\theta_d$$

This is the modified Bessel equation of zero order, and its solutions are modified Bessel functions of zero order. The particular solution that fits the boundary conditions is

$\theta_d = K_0(kr)$,

or this multiplied by any constant, where k = sqrt(2C/A).

METRA Implementation

The physical situation is introduced into METRA using the radial DCM formulation and fully saturated media. The center well has a radius of 1 cm, and a liquid mass flux out of the system of 160 kg m^{-3} sec^{-1} is applied at the fracture network at the well. This was intended to correspond to 1 kg per second per meter of wellbore, but it seems that this flux is actually per unit area of fracture, not per unit area of total surface. Therefore the flux actually corresponds to 10 kg per second per meter of wellbore. The matrix is sealed at the well, so there is no flux in the matrix at the well boundary. Initial conditions are that the aquifer is pressurized to a uniform level of 1 MPa. The far boundary at 5000 meters is sufficiently far away to not affect the initial transient behavior.

In order to set the various coefficients above equal to each other (D=E, A=B), the fracture volume is set to 10%, with a porosity of 90%. The matrix porosity is 10%. Similarly, the fracture permeability is 10^{-10} m^2, while the matrix permeability is 1.1×10^{-11} m^2. Thus, the bulk permeability of the fractures and bulk permeability of the matrix are both equal to 10^{-11} m^2. Output of matrix and fracture pressure are taken at 300 seconds after start of pumping, and 3000 seconds. In order to create a visible difference in matrix and fracture pressure, a reduction of interfacial area by a factor of 10^4 is applied. The input file is below:

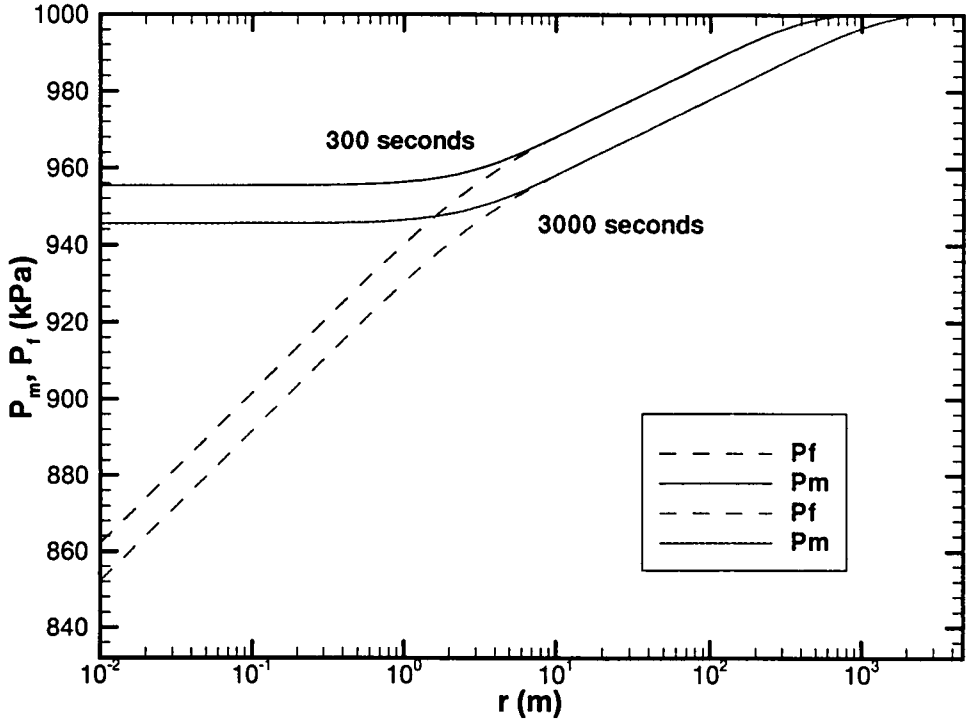
```
METRA test:  DCM version of Theis' solution for pumping from infinite aquifer
             Aug. 9, 1999
:
RSTART 0
:
:grid  geometry  nx  ny  nz ivplwr ipvcal iout gravity pref tref  href
GRID   DCMRAD    100   1  1    1      1      3    0      0    0    0
:
:dcm i1 i2 j1 j2 k1 k2 sigmaf areamod xlm ylm zlm
DCMP
     1 100 1  1  1  1  0.10   1.e-4   0.1 0.1 0.1
/
:
LIQUID
:
Pckr                          :relative perm and pc keyword
:    i  type-curv swirm  rpmm(lamda)  alpham  swext  sgc  iecm
:
     1  Van-Gen   0.01   0.63    9.7e-5      0.    0.  0
/
:
Thermal-prop
: no rho       cpr   ckdry  cksat    crp  crt   tau  cdiff   cexp  enbd
  1  2.510e+3  847.    1.02   1.76     0     0    .5  2.13e-5  1.8  0.
```

```
0
:     igrid   rw      re
DXYZ   0      0.00   5000.
: (dx(i),i=1,nx)
0.01
:
: (dy(j),j=1,ny)
360.
:
: (dz(k),k=1,nz)
1.0
:
PhiK
: il i2  j1  j2  k1 k2  iist ithrm  vb por  permx  permy  permz porm permm istm ithrmm
  1  100  1  1   1  1   1       1   0. 0.9  1.e-10  1.e-10  1.e-10 0.1  1.1e-11 1     1
/
:
Init
: il  i2   j1  j2  k1  k2    p      t    sg   xg2   pm  tm sm xm
   1  100   1   1   1   1  1.e6   20.0  0.0  0E+00 1.e6 20. 0. 0.
/
:
Recurrent-data
BCON 1
: itype iface il i2 j1 j2
: timce  qbc  pbc  tbc  sgbc  xabc  emis
: timbc  qbcf pbcf tbcf sgbcf xabcf emisf
    2   LEFT  1   1  1 1
   0.0    0.0 0.   20.  0.    0.     0.
   0.0   -160. 0.   20.  0.    0.     0.
.
:
Rstart 1 0
Output   A=1
:  isolve newtnmn newtnmx north nitmx level
Solve 4     1    12    2    100   1
:
:AUTO-step  DPMXE    DSMXE  DTMPMXE   DP2MXe
AUTO-step   1.0E+1   0.04    1.       1.e3
:
:TOLR TOLP  TOLS  TOLT  TOLP2  TOLM  TOLA  TOLE rtwotol rmxtol smxtol
Tolr   0.1 1.e-4  1.e-3  1.e-1 1.e-8 1.e-8 1.e-6 1.e-25 1.e-25 1.e-25
:
:Limit dpmx    dsmx dtmpmx dp2mx dtmn dtmx icutmx
LIMIT  5.e3    .1    3.    .5e4  1.e-8  1.e5 .334
Plots 1 1 1
35
Time[s]   300
Time[s]  3000
Ends
```

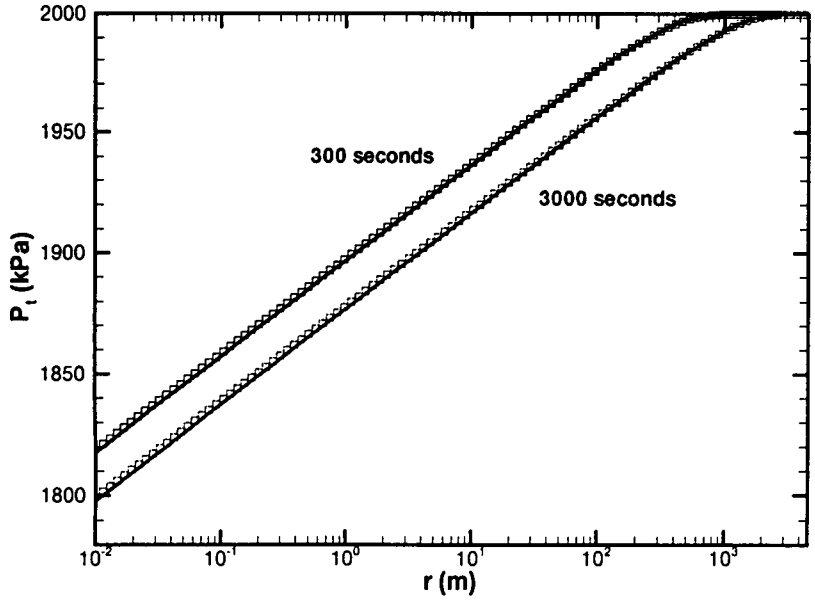
Comparison of METRA to theory

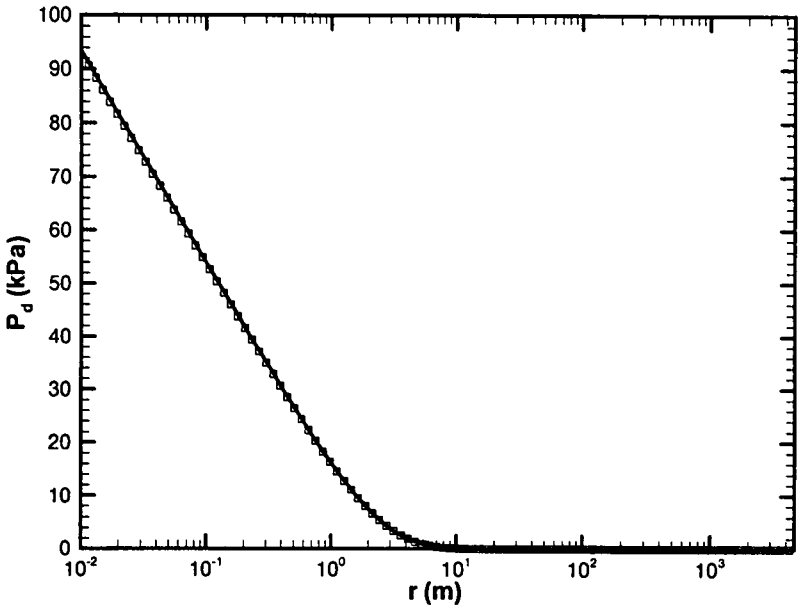
The series solution for θ_t is evaluated with 50 terms. This is sufficient to bring the solution out to about 3000 meters. For θ_d, the modified Bessel function K_0 is computed using the recommended approximation functions given in Abramowitz and Stegun. Approximate values for viscosity and compressibility of water at 20 deg. Celsius were taken from the CRC Handbook of Chemistry and Physics.

First, let us examine the system in the original variables. Since the matrix does not have direct contact with the well, the gradient of the matrix pressure, P_m, is zero near the well. The matrix pressure is drawn down over time by exchange with the fracture network, which is being drained. Far from the well, the matrix and fracture pressures equilibrate, and farther still, the reservoir pressure is undisturbed.



The transformed variables computed from the above METRA results and from the theoretical solutions are compared in the next graphs. METRA results are displayed as solid lines, and theory results as squares. As you can see there is very good agreement between the two results at both computed times. There is no change in the difference pressure between 300 and 3000 seconds, so only one curve is plotted in that figure.





Other

August 10, 1999
Identified bug in grid generation for DCMRAD grid type. METRA assigns radii for the number of original nodes, but DCM requires twice the number of nodes. Result is that the last half of the fracture nodes and last half of the matrix nodes are not assigned a radius. Reported bug to Scott Painter.

August 17, 1999
Actually the above bug was a printing error. Received fix via email from Seth Mohan.

Development of a drift climate model for Multiflo

Account number: 20-1402-562
Description: Near field environment code development – Multiflo
Collaborators: S. Painter, CNWRA

September 15, 1999
I have received model notes and guidance from Scott Painter on the drift climate model. Coding begins for the model following the technique and methods outlined in the "Software Requirements Description" for Multiflo 2.0. A standalone code is expected to be adequate for initial testing.

September 28, 1999
I've written a simple standalone code from Scott Painter's notes and outline. Following is some documentation of the method and code.

Notes:
1. All parameters in the model are in SI units. Choice of units for the inputs is not arbitrary for the user, because Stefan-Boltzmann constant is defined internally in $W/(m^2K^4)$. Temperatures are in Kelvins to facilitate radiation calculations.
2. Output heat is in Watts, that is to say integrated over the node volume. Similarly, output water flux is in kilograms. Signs are as appropriate for METRA source nodes, i.e. heat leaving the drift is positive, water flowing to the drift is negative.
3. Assumption is made that all radiating surfaces are black bodies. Introducing emissivity < 1 means reflecting some energy around the drift, which should raise canister temperatures and lower net radiative heat flux.
4. The saturated vapor pressure, Pvsat, is always determined from the air temperature. The latent heat of vaporization is determined at the wall temperature. I've assumed the Pv is always less than Pvsat. One could imagine a case of humid warm inlet air and cold tunnel walls where water would condense out in the tunnel (assuming low or no heat generation). I'm not sure if the code can handle this, although it may just mean extending plog to positive arguments.
5. Because of the iterative solution path, I'm finding that it takes approximately a number of iterations equal to the number of nodes to come close to converging. CPU time is still only three seconds for 1000 iterations on 500 nodes, and I have optimization off and debugging code on. I'm also using underrelaxation, which may no longer be necessary since I introduced the Newton-Raphson type solver.

Solution method:

1. For given wall and canister temperatures, the air temperature is solved node by node using the analytic solution with piecewise constant boundary conditions.
2. The total heat transport from the canister is determined using both convection and radiation with current temperatures, as

$$\frac{q'}{2\pi r_c} = \sigma\left(T_c^4 - \left\langle T_w^4 \right\rangle\right) + h_c\left(T_c - T_a\right), \qquad (1)$$

where q' is heat transferred per unit length. The canister temperature is adjusted by the Newton-Raphson method to make the lost power more equal to the generated power using a derivative of the above equation,

$$\Delta T_c = \frac{\Delta q'}{2\pi r_c\left(4\sigma T_c^3 + h_c\right)}, \qquad (2)$$

where $\Delta q'$ is the additional heat per unit length needed to make the transferred heat match the power generated. This formulation proved more stable for large h_c than computing T_c directly from equation (1).

Heat and Mass transfer coefficients

Given the known properties of r_w = 2.75 m, and Q_{air} = 2 – 10 m^3/sec, air velocities range from approximately 9 cm/sec to 42 cm/sec. As the air temperature increases these velocities will increase slightly. The Reynolds number of the proposed airflow ranges then from 30000 to 150000. This is well into the range of turbulent flow, for flow in a duct. Kays and Crawford (1993) suggest that for air, the convection coefficient in turbulent internal flow is largely independent of the exact boundary conditions. They suggest a correlation of the form

$Nu = 0.022\ Pr^{0.5}\ Re^{0.8}$.

The Prandtl number of air is about 0.7. Under these circumstances, then, we can expect Nusselt numbers from 70 at low velocity to 250 at the high velocity. This yields convection coefficients from 0.3 to 1.2 W/(m^2 K). I would apply the same number to both the waste package and the tunnel wall in the absence of other information. Surface roughness will increase the convection coefficients to some degree.

For low mass transfer rates, Kays and Crawford (1993) suggest that the mass transfer coefficient may be approximated as h/c_p, which assumes that heat and species diffuse at about the same rate in a gas. Under these circumstances, the mass transfer coefficient β will range from 3 x 10^{-4} kg/(m^2 sec) to 1.2 x 10^{-3} kg/(m^2 sec).

Inputs

The following parameters must be defined in the code. Numbers given are examples only.

tunnrad = 2.75 meters
qair = 10.0 m^3/sec

tinlet = 300. K
pressure = 1.e5 Pa
wppower = 1.e4 W/meter
wprad = 0.5 meter
wpconv = 1. W/(m^2 K)
tunconv = 1. W/(m^2 K)
watconv = .1 kg/(m^2 sec)
pvinlet = 0.5*pvsat(tinlet) (Pa)

Notice here I've chosen an inlet vapor pressure which corresponds to 50% relative humidity. This sort of this is allowed, and can keep Pv < Pvsat.

Also, the boundary conditions in the form of the wall temperature and the wall moisture content must be given along with specifying the node positions and sizes. These are in the arrays:

tw(i) wall temperature in Kelvins for node i
ws(i) product of porosity and saturation for walls at node i
nodcent(i) center of i^{th} node
nodlen(i) length of i^{th} node

Outputs

primary outputs (send to METRA)

hflux(i) Total Watts shed to wall in node i
wflux(i) Total kg/sec of water flowing out of drift node i

secondary outputs (possibly of interest to the user)

tc(i) Waste package (canister) temperature in node i
ta(i) Lumped air temperature in node i
pv(i) H_2O vapor pressure in node i

Currently, the air temperature and vapor pressure are defined at the upstream end of each node to facilitate the analytic solution.

Reference

W. M. Kays and M. E. Crawford (1993) *Convective Heat and Mass Transfer*, 3rd ed., McGraw Hill, New York.

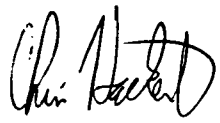
October 6, 1999

Scott Painter noted that the drift program was not handling heat flux to the tunnel ends quite correctly. This has been fixed in the current version.

FINAL ENTRIES

Entries into Scientific Notebook No. 364E for the period of July 29, 1999 to October 29, 1999 were made by Chris Hackert.

No original text entered into this scientific notebook has been removed.



SCIENTIFIC NOTEBOOK

by

Chris Hackert

Southwest Research Institute
Instrumentation and Space Research Division
San Antonio, Texas

July 1, 2000 – September 12, 2000

Table of Contents

INITIAL ENTRIES

Scientific notebook: #364E Vol. 2
Issued to: Chris Hackert
Issue Date: August 3, 1999

This notebook will be printed at approximately quarterly intervals. This computerized electronic notebook is intended to address the criteria of CNWRA QAP-001.

Development of Multiflo code benchmarks

Account number: 20-1402-562
Project manager: S. Painter, CNWRA
Objective: Create benchmarks to test the TVD code in the GEM module of Multiflo.

July 12, 2000

Received info on TVD (Total Variation Diminishing) flux-corrected transport algorithm from Scott Painter. This has been implemented in GEM for better tracking of sharp concentration fronts. The implementation needs to be thoroughly checked, especially in multiple dimensions.

July 13, 2000

Got several files, including tarball and makefile for GEM, databases (master25.v8 for isothermal, and mastertemp.v8.r5 for temperature dependent), and sample input files.

masin2	centered-space 1D
masin21	tvd-1 (IFOR=4)
masin22	tvd-2 (IFOR=5)

and

m3tvd	3D flow-through problem
t	1-D reacting flow, to test if tvd interferes with kinetics calculations

GEM compiles on terra, and runs on terra and gaia, although currently the database must reside in the same directory as the input file and be specified by the full path.

1D flow-through test of TVD implicit algorithm

July 14, 2000

Using the masin* files as baseline cases, the goal of this part is to examine the accuracy and CPU usage of the TVD algorithm while varying control parameters. Output quantities of interest will include some measure of accuracy, total CPU time, number of time steps and number of Newton iterations. While the final CPU time is currently not being output correctly, the "GEM execution time" line does have the right number, as confirmed by the Unix "time" command. Accuracy can be determined by comparison with the 1-D analytical solution provided by S. Painter.

Parameters to vary will include IFOR, IFLXLIM, and maybe some tolerances? Maybe the dx, since a 4 or 5 point scheme should be more spatially accurate.

July 18, 2000

Finished varying IFOR and IFLXLIM for the original 1D flow through case. Results are that:

All values of IFOR (1 through 5) do a reasonably good job of tracking the 50% front (concentration midway between min and max).

All the TVD algorithms are substantially more accurate than central-difference or upwinding.

There is no appreciable difference between IFOR = 4 and IFOR = 5. IFOR = 5 takes longer to run, but less than 2% more CPU time. Except for the Leonard limiter (which has a slight difference), there is no difference in the concentration profiles printed out by GEM when changing between IFOR = 4 and IFOR = 5.

IFLXLIM = 3 and 4 (Fromm and minmod limiters) are the most accurate. The Fromm limiter is slightly more accurate, but takes about 30% more CPU time than the minmod limiter. Curiously, the minmod limiter takes slightly more time steps but slightly fewer iterations than the Fromm. The falloff distance with these limiters for a time of 1.5 y is about 4% longer than the true solution. With the Leonard and Van Leer limiters, it is about 7% longer, central-difference is about 13% longer, upwinding is about 18% longer. I define the falloff distance as the length between the 90% and 10% concentration change fronts.

IFLXLIM = 1 and 2 (Leonard and Van Leer limiters) are the fastest, but least accurate of the TVD algorithms. The Van Leer limiter is about 20% faster than Leonard's limiter, but slightly less accurate. Approximate run-times on gaia.space.swri.edu are: Van Leer, 63 sec; Leonard, 79 sec; minmod, 139 sec; Fromm, 162 sec. The implicit central-difference and upwinding cases run in less than 4 seconds each.

July 25, 2000
S. Painter believes that the run times for the above cases are unacceptably long in comparison to the upwind and central difference cases. After discovering that a great reduction in Newton iterations and run time could be achieved by altering the diffusivity, grid spacing, saturation, or porosity, Scott and Mohan decided that stricter time-step control is probably necessary for the TVD algorithm results.

Mohan Seth proposes a new input file with the following changes:

```
OPTS:
ITMAX cut from 32 to 7
ISTEPDT changed from 1 to 2
DTCUTF reduced from 0.5 to 0.2
```

```
SOLV:
ISOLVE changed from 4 to 3

DTSTep:  max DT reduced from 0.01 to 0.004
```

M. Seth's input file runs in 15 seconds. The unaltered input file runs in 78 seconds. The result from Mohan's file appears to be slightly more accurate, too. Cutting the max timestep reduces the runtime to 42 seconds. Changing only ISTEPDT doesn't make much difference. Reducing DTCUTF to 0.2 reduces the run time to 42 seconds. Reducing ITMAX to 7 cuts the runtime to 30 seconds. Changing the ISOLVE makes it run in 67 seconds (leaving north = 1; changing to north=4 increases the run time to 105 seconds).

So, it appears that the problem was perhaps too large a time step for the TVD algorithm. This leads to an excessive number of Newton iterations, which leads to a large number of time-step cuts. The important parameters seem to be DTSTep, DTCUTF, ITMAX, and ISOLVE. The ISOLVE presumably just takes advantage of a more efficient solution algorithm. Changing DTSTep, DTCUTF, and ITMAX yields a 20 second run time. Changing ISOLV in addition to the others gets us to the 15 second run time.

An important parameter seems to be the number of time step cuts in the problem. Too many mean that the code is wasting time recomputing time steps over again. Too few mean that the maximum time step is too restrictive. Interestingly, the right value for the max DT changes for different flux limiters. IFLXLIM = 2 can take a larger time step than IFLXLIM=3 or 4.

Scott Painter suggests moving forward on this basis and looking at the effects of the grid spacing and flow velocity. He also suggests comparing TVD, upwind, and central-difference results in a diffusion dominated limit. To have some uniformity of time step control, set ISTEPDT=1 and DELCMX = 7.2e-4 (10% of the concentration change). If this doesn't work out try decreasing DELCMX. He suggests avoiding the use of DTSTep to limit the time step, since it is a non-automatic optimization. We'll keep ISOLV=3 for this 1-D case, maybe moving back to ISOLV=4 in 3-D. Set ITMAX to 9.

July 26, 2000
After running this for a while, I think better performance is achieved if I also set DTCUTF to 0.25 and reduce DELCMX to 5% of the concentration change, or 3.6e-4. This puts IFOR=4, IFLXLIM = 1 back into the 15 second runtime regime without having to artificially limit the DTSTep.

I've rerun the basic cases with all the variations of IFOR and IFLXLIM. Result trends are similar, although CPU time is down across the board and accuracy is improved. Interestingly, accuracy is improved for the central-difference and upwinding cases as well. I suspect now that a major component of the accuracy differences among the TVD cases is the time step. The less stable flux limiters force smaller time steps and so of

course wind up being more accurate. The Van Leer limiter appears to be able to handle larger time steps, and so winds up less accurate as a consequence.

I've also checked the CPU time as a function of DELCMX for the IFOR = 4, IFLXLIM = 1 case. There is in fact a minimum in CPU time near DELCMX = 4.2e-4. The solution obviously becomes more accurate as this parameter is reduced, but large values apparently will yield not only less accurate but also slower results.

July 27, 2000
I've now checked the CPU time for the (IFOR, IFLXLIM) = (4,2) and (4,3) cases. There is a minimum in CPU time for each flux limiter. As expected, the Van Leer limiter has its minimum at a less constrained value near DELCMX = 12.6e-4, while the Fromm limiter has a minimum near the more constrained value of 2.7e-4.

It turns out that for this case the central-difference solution displays no oscillations, although there is still substantial numerical diffusion for the upwinding case. We really want a more advection dominated system, so I will reduce the diffusion coefficient until oscillations appear. I will also increase the diffusion coefficient, so that we can make sure in the diffusion dominated limit that central-difference and tvd approach the same result.

Forcing the central difference result to have oscillations is interesting in its own right. What I'm seeing here is that numerical diffusion smoothes out the oscillations unless the time-step control is really cranked down. The central difference doesn't give significant oscillations for D=1.e-6 until DELCMX < 4.e-4. As DELCMX becomes smaller, the front gets steeper and the oscillations more pronounced. The tvd algorithms don't initially seem to provide any sharper front than the central difference, but they do eliminate the oscillations when present.

For this advection dominated case, then, I will set DELCMX to 2% of the maximum difference, or 1.44e-4, and use a diffusion coefficient of 1.e-6 cm^2/s. For this case, IFOR =5 produces exactly identical results as IFOR=4 for all flux limiters. Flux limiters 1 through 3 are marginally more accurate on slope than the central difference, but all tvd cases are free from oscillation. Flux limiters 1 through 3 run slightly slower than central difference, about 20 – 30%. Flux limiter 4 yield a curve with slightly more numerical diffusion than central difference, but is only 5% slower. In terms of overall accuracy, Leonard is the best, and Fromm edges Van Leer for second place.

For the diffusion dominated case, I set a diffusion coefficient of 1.e-2 cm^2/s, and keep the DELCMX to 1.44e-4. Here, all the tvd results are identical regardless of flux limiter or IFOR. The tvd results are very close to the central difference results. Upwinding is slightly less accurate than central difference and tvd. The code is fast in all cases here, about 7.5 seconds for upwind and central difference, and 8.5 seconds for all the tvd cases.

July 28, 2000

I went back to the diffusion coefficient = 1.e-4, and kept the DELCMX at 1.44e-4. With this tight control on the time step, there is essentially no real difference in accuracy or run time between the four flux-limited tvd results and the central difference result.

Here (for reference) is the base input file. I've only been varying DELCMX, DIFF, and of course IFOR and IFLXLIM. The DTSTep value is set very high to force the time step to be computed from DELCMX.

```
  Flow through benchmark: repeat
                July 19, 1999
:
:       geometry nx  ny  nz   ifrq  iprint iwarn idebug ibg1 ibg2
GRID     XYZ     400  1  1     1      1      1    0      1  3
:
DBASe
/home/chackert/multiflo/tvdtest/1dtest/master25.v8
:
:       method  iops  ifor  iflxlim  itmax ihalmx ndteq loglin  istepdt
OPTS      1       0     4      1        9     16    1     0        1
:   isurf  iact   isst  wtup  cournr  dtcutf  delcmax  qkmax  tpulse
     0      0      -1    0     1.e0    .25    1.44e-4    500
:
:      isync  ipor iperm  perm-fac.
COUPle   0     0     0       3.
:
PLTFiles
:iplot   a  s  t  m si sf  v  z  b in  e ex ti  g itex
    1    1  0  0  0  0  0  0  0  0  0  0  0  0   0
:
:     tol    ttol  tolneg tolmin  tolstdste tolc
TOLR  1.d-9 2.e0  1.e0  1.e-10    1.e-12  1.e-12
:
:        mcyc  cc  c   flx r  sp  qk  pk  rk  a1  a2  a3
DEBUg    0       1  1   0   1   1   1   1   1
:
:      isat isothrm iread  por0  phir  sat  w  lambda toldelt  tolpor
ISYStem 0    0       0     .11     1.   0.5  .5   1.    1.e-3    1.e-3
:
:        vx0    vy0    vz0    vgx0    vgy0    vgz0[m/yr]    alphax      alphay
alphaz[m]
FLOW  1.    0.    0.    0.     0.     0.           0.        0.        0.
:
:   d0[cm^2/s]  delhaq[kJ/mol]  dgas[cm^2/s]  dgexp  tortaq  tortg idif
DIFF   1.d-4      12.6            2.13d-1       1.8   1.d0   1.d0    0
:
:      p (Pa) temp flag  a    b    c    d   x0   xlen
PTINit  1.e5   25.   0   0    0    0    0  1000.  2.d3
:
:master species for controlling time stepping
MASTER ALL
:
DXYZ
400*0.1
1.
1.
```

```
:
:   isolv level north nitmax idetail rmaxtol rtwotol smaxtol
SOLV    3   1     1    100      0     1.e-20  1.e-20  1.e-12
:
:initial and boundary conditions: 1-conc., 2-flux, 3-zero gradient
COMP
:
:i1  i2  j1  j2  k1  k2
  1  400   1   1   1 1
.
:species itype   ctot  mineral  diffusion
ca+2       7     8.e-4
.
.
:
BCON
:ibndtyp ifacx tmpbc dist area vell velg  por  sl  porm  slm  imtx
    1      1    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
: i1  i2  j1  j2  k1  k2
   1   1   1   1   1   1
.
:species  itype  guess  ctot  mineral
ca+2       7     8.e-3
:
:ibndtyp ifacx tmpbc dist area vell velg  por  sl  porm  slm  imtx
    1      2    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
: i1  i2  j1  j2  k1  k2
   400 400 1   1  1  1
.
:species  itype  guess  ctot  mineral
ca+2       7     8.e-4
.
:
DTSTep[y]      3 2.e-7 1.e-4 1.e-2
1.e-8            1.d-5   1.e-3  0.1
:
TIME[y] 3 0.5 0.9  1.5
:
ENDS
```

In summary, I would say that the TVD algorithms all behave approximately equally well in the diffusion dominated and mixed regimes, if the time step control is sufficiently restrained. If large time steps are allowed, the Leonard and Van Leer flux limiters are more forgiving than the Fromm or minmod limiters. Good correspondence to the central difference is obtained in the diffusion and mixed regimes. In the advection dominated regime, all the TVD algorithms suppress the oscillations observed in the central difference result. Accuracy in the transition region (as measured by steepness of gradient) is not significantly improved from the central difference result. For the particular test case examined, the Leonard flux-limiter provided marginally more accurate results than the other flux limiters. In no case did changing IFOR = 4 to IFOR = 5 make any real difference.

Overall, then, we could recommend IFOR=4, IFLXLIM = 1 as a good TVD based solution method. It will be important to use relatively tight constraints, as the TVD algorithm seems to be more efficient with a small time step.

Table of 1-D TVD results

IFOR	IFLXLIM	comments	CPU	steps	iters	cuts	x value at concentration 90%	50%	10%
diffusion = 1.e-4, DELCMX = 3.6e-4							time = 1.5 y		
analytic solution		exact values					26.05	27.29	28.54
1	99	central diff	12.3	636	1548	292	25.08	27.24	29.57
2	99	upwind	7.2	398	920	140	23.91	27.24	30.78
4	1		14.9	689	1914	325	25.23	27.27	29.41
4	2		13.8	669	1731	313	25.17	27.28	29.47
4	3		36.5	1056	5658	511	25.53	27.3	29.08
4	4		44.1	1241	7341	604	25.55	27.3	29.06
5	1		15	688	1902	324	25.22	27.27	29.42
5	2		14.2	673	1766	316	25.18	27.28	29.45
5	3	identical to 43	36.5	1056	5658	511	25.53	27.3	29.08
5	4	identical to 44	44.4	1241	7341	604	25.55	27.3	29.06
diffusion = 1.e-6, DELCMX = 1.44e-4							time=0.5 y		
		exact (approx)					8.99	9.09	9.18
1		central diff	56.7	2956	7032	1463	8.63	8.99	9.56
2		upwind	21	1172	2844	570	7.77	9.07	10.45
4	1		66.2	3065	7600	1518	8.68	9.09	9.51
4	2		71.2	3337	8309	1654	8.61	9.12	9.53
4	3		71.6	3037	8420	1504	8.63	9.12	9.51
4	4		59.2	2693	7462	1332	8.48	9.11	9.67
5	1	identical to 41	67.3						
5	2	identical to 42	72.5						
5	3	identical to 43	72.2						
5	4	identical to 44	60						
diffusion = 1.e-2, DELCMX = 1.44e-4							time=0.5 y		
		exact					4.43	10.56	17.3
1		central diff	7.7	428	1051	202	4.36	10.5	17.43
2		upwind	7.5	416	1021	195	4.33	10.54	17.55
4	1		8.5	426	1047	201	4.36	10.51	17.43
4	2		8.4	426	1047	201	4.36	10.51	17.43
4	3		8.4	426	1047	201	4.36	10.51	17.43
4	4		8.4	426	1047	201	4.36	10.51	17.43
5	1		8.4	426	1047	201	4.36	10.51	17.43
5	2		8.4	426	1047	201	4.36	10.51	17.43
5	3		8.4	426	1047	201	4.36	10.51	17.43
5	4		8.4	426	1047	201	4.36	10.51	17.43
diffusion = 1.e-4, DELCMX = 1.44e-4							time=1.5 y		
1		no oscillations	34.7	1845	4443	907	25.72	27.25	28.91
4	1		39.02	1862	4556	915	25.72	27.29	28.9
4	2		37	1742	4290	855	25.66	27.3	28.94
4	3		37.7	1758	4368	864	25.68	27.3	28.92
4	4		38.4	1784	4776	877	25.62	27.31	28.99

3D flow-through test of TVD implicit algorithm

August 1, 2000
For the 3-D case, we want to confirm against the 3-D analytic solution, explore the operating space, and test flow-through in multiple directions to confirm correct operation.

Initially, things look close between TVD (4,1) and central difference, but this is for D = 1.e-3, which is in the diffusion dominated limit. Also, it appears that for time = 1 yr, the outflow boundary may not be far enough away to not influence the downstream result. I'll probably want to look at the shorter time of time = 0.5 y anyway, though. Looks like setting D at 1.e-6 cm^2/s gives the expected instability in the central difference. TVD still looks good.

Possible bug: when D = 1.e-6, the initial concentration profile appears to come out as 8.8e-3 not 8.e-3 like I think it should.

August 2, 2000
This bug (or "undocumented feature") appears to result from two overlapping boundary conditions. GEM seems to be adding the concentrations from each bc, not replacing the first with the second. It is simple to work around this problem.

I'll move on to check consistency, by reversing the flow and changing the flow direction from x to y and z. This requires updating the number of grids, the flow field, the grid spacing, and the boundary and initial condition. I'm using the IFOR=4 , IFLXLIM=2 case with D= 1.e-6. This is in the advection dominated regime. Wrote a shell script to compare profiles through the volume. Results are that each profile is exactly identical to within the limits of precision of the GEM output. For D=1.e-4, all six flow directions also produce identical results. Scott says I probably don't need to check all four flux limiters.

August 3, 2000
Looking at accuracy of solution methods and flux limiters.

D = 1.e-6: Central difference shows strong oscillations at t = 0.5 y. Upwinding shows numerical diffusion. Leonard is most accurate tvd method, with others showing accuracy more or less in proportion to the run time/number of iterations. No discernable difference in accuracy between IFOR = 4 and IFOR = 5.

D = 1.e-4: Upwinding shows slight numerical diffusion, the profile is just a little too smooth. All the tvd cases are very close to the central difference in value. Compared with the analytic solution, the concentration profiles are recovered with adequate accuracy, considering the patch is only 3 grids by 3 grids.

The input file used, varying only IFOR, IFLXLIM, and DIFF is:

```
Flow through benchmark in 3d: repeat
              August, 2000
```

```
:
:       geometry nx  ny  nz    ifrq  iprint iwarn idebug ibg1 ibg2
GRID     XYZ      40 11  11     1      1      1    0      1  3
:
DBASe
/home/chackert/multiflo/tvdtest/3dtest/master25.v8
:
:       method  iops  ifor  iflxlim  itmax ihalmx ndteq loglin  istepdt
OPTS      1       0     5      4        9    16    1     0        1
:   isurf  iact   isst  wtup  cournr  dtcutf  delcmax  qkmax  tpulse
     0      0      -1    0     10.e0    .25    1.44e-4   500
:
:      isync  ipor iperm  perm-fac.
COUPle   0     0     0       3.
:
PLTFiles
:iplot   a  s  t  m si sf  v  z  b in  e ex ti  g itex
    1    1  0  0  0  0  0  0  0  0  0  0  0  0  0   0
:
:     tol    ttol  tolneg tolmin tolexp  dthalf qkmax tolstdste tolc
TOLR  1.d-9 2.e0  1.e0  1.e-10  5.d0    .5     590.   1.e-12  1.e-12
:
:        mcyc  cc  c   flx r  sp  qk  pk  rk  a1  a2  a3
DEBUg    0       1  1   0   1   1   1   1   1
:
:      isat isothrm iread  por0  phir  sat  w  lambda toldelt  tolpor
ISYStem 0    0      0     .11     1.   0.5  .5   1.    1.e-3    1.e-3
:
:        vx0    vy0    vz0    vgx0    vgy0    vgz0[m/yr]    alphax     alphay
alphaz[m]
FLOW  1.   0.   0.   0.    0.    0.          0.       0.       0.
:
:   d0[cm^2/s]  delhaq[kJ/mol]  dgas[cm^2/s]  dgexp  tortaq  tortg idif
DIFF   1.d-4     12.6            2.13d-1       1.8   1.d0   1.d0    0
:
:      p (Pa) temp flag  a    b    c    d   x0   xlen
PTINit  1.e5   25.   0   0    0    0    0  1000.  2.d3
:
:master species for controlling time stepping
MASTER ALL
:
DXYZ
0.25 0.25 0.25 0.25 0.25 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 2.0 2.0 2.0 2.0 3.25
2. 1.0 0.25 0.1 0.1 0.3 0.1 0.1 0.25 1.0
2.
2. 1.0 0.25 0.1 0.1 0.3 0.1 0.1 0.25 1.0
2.
:
:   isolv level north nitmax idetail rmaxtol rtwotol smaxtol
SOLV    3    1     1    100      0     1.e-20  1.e-20  1.e-12
:
:initial and boundary conditions: 1-conc., 2-flux, 3-zero gradient
COMP
:
```

```
:i1  i2  j1  j2  k1  k2
  1   40  1 11  1  11
.
:species itype   ctot  mineral  diffusion
ca+2       7     8.e-4
.
.
:
BCON
:ibndtyp ifacx tmpbc dist area vell velg  por  sl  porm  slm  imtx
: i1  i2  j1  j2  k1  k2
: influx
   1      1    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
   1   1   1   11   1   11
.
:species  itype  guess  ctot  mineral
ca+2       7     8.e-4
:patch influx
    1      1    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
   1  1   5   7  5  7
.
ca+2       7     7.2e-3
:outflux
    1      2    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
   40 40  1  11  1 11
.
ca+2       7     8.e-4
.
:
DTSTep[y]      3 1.e-7 1.e-5 1.e-2
1.e-8            1.d-4   1.e-3  1.e-0
:
TIME[y] 2 0.5 1.0
:
ENDS
```

This same input file was used for the consistency checks, making the indicated changes in GRID, FLOW, DXYZ, COMP, and BCON.

Table of 3-D TVD results

IFOR	IFLXLIM	comments	CPU	steps	iters	cuts
D=1.e-6						
	1	oscillations	328	1009	2065	494
	2	numerical diffusion	178	509	1083	243
	4	1	353	972	2153	476
	4	2	344	953	2117	466
	4	3	363	975	2281	477
	4	4	310	857	1868	417
	5	1 identical to 41	355	972	2153	476
	5	2 identical to 42	347	953	2117	466
	5	3 identical to 43	365	975	2281	477
	5	4 identical to 44	312	857	1868	417
D=1.e-4						
	1		130	309	641	141
	2	slight numerical diffsn	92	209	434	91
	4	1	151	315	701	145
	4	2	143	303	676	139
	4	3	169	336	781	156
	4	4	149	313	695	144
	5	1 identical to 41	153	315	701	145
	5	4 identical to 44	150	313	695	144

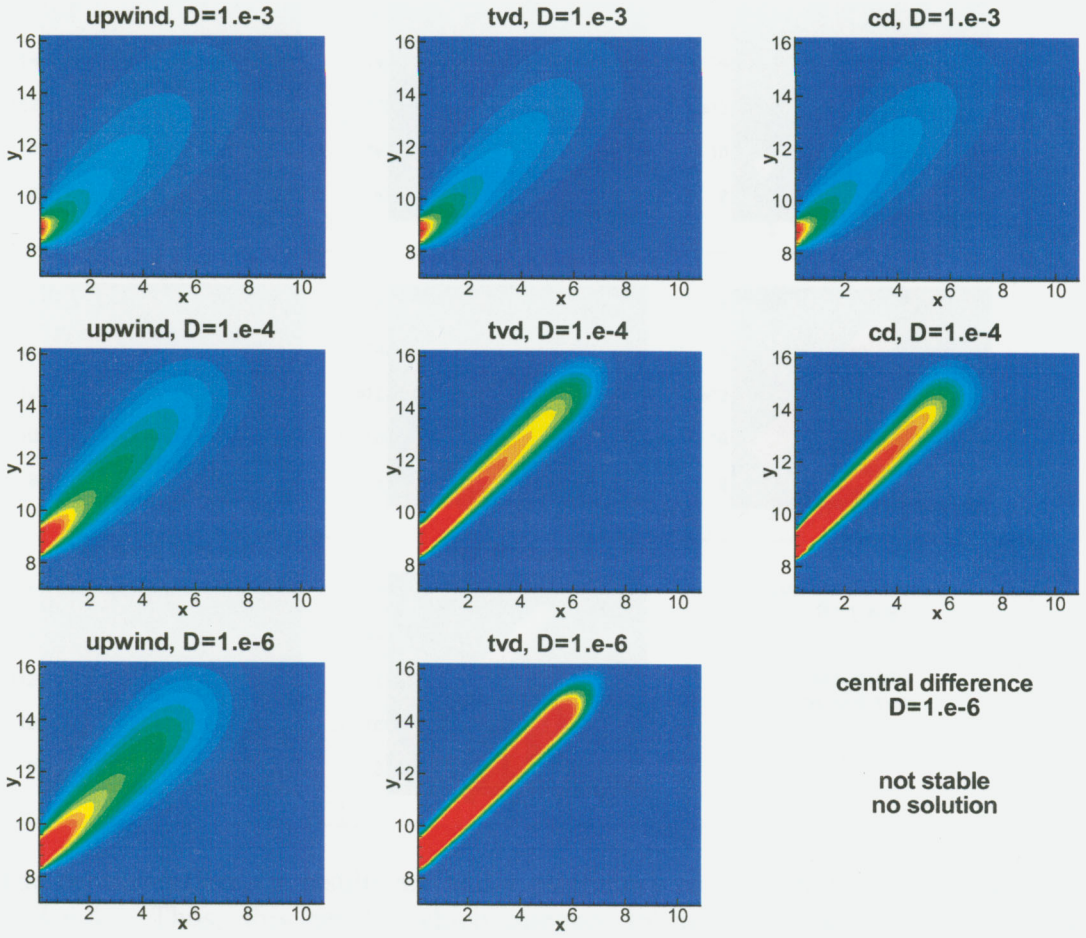
Test of diagonal flow

August 18, 2000

The current implementation of the tvd algorithm lacks some cross terms which are strictly necessary for flux limited modeling of flow which is not parallel to the grid lines. We need to find some cases which can test the importance of neglecting these terms. Initially, I'll try using GEM in 2-D with FLOW in both x and y directions, if GEM will allow such inputs.

GEM will handle this situation. Numerical instability in the central difference is bad at low diffusion coefficients, to the point of creating NaN's in the output file. The tvd methods suffer from relatively high numerical diffusion because diffusion occurs along both the x and y grid lines. Nevertheless, the numerical instability appears to be avoided by using tvd, and numerical diffusion is significantly less than if upwinding is used.

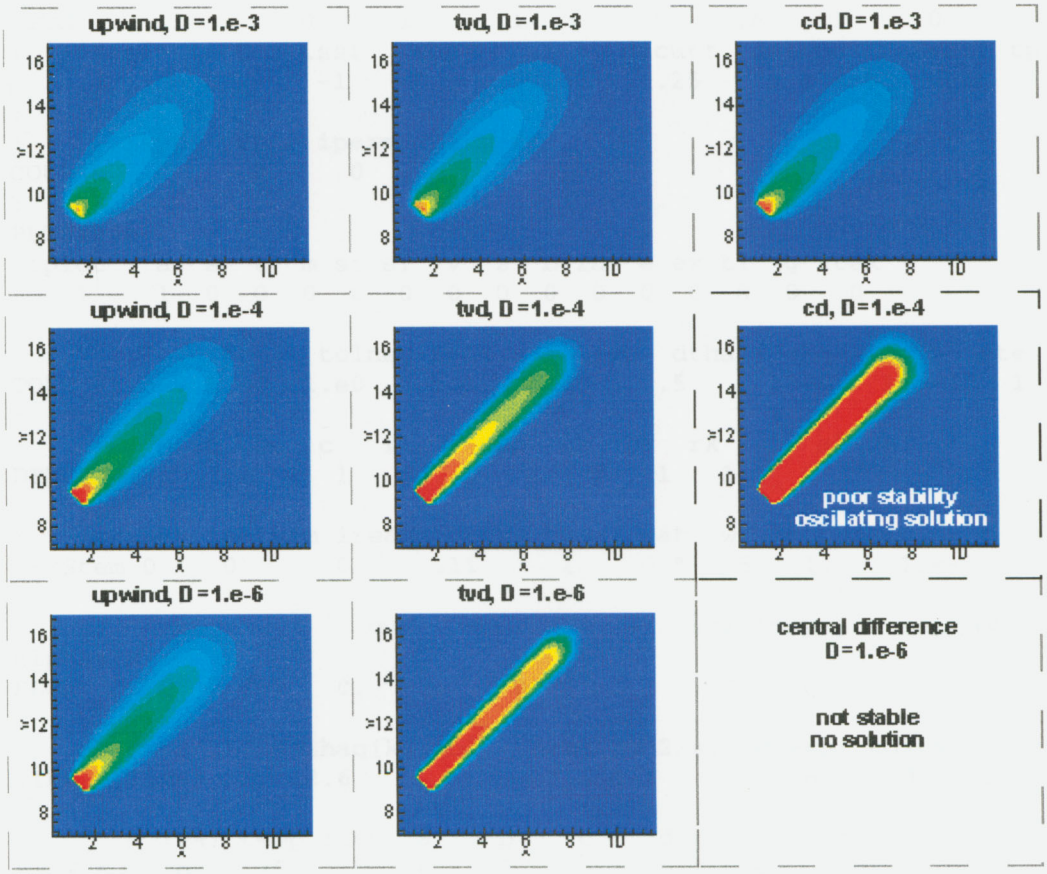
Looking at a plot of some concentration profiles, it seems possible that the tvd algorithm has slightly more numerical diffusion than the central difference for intermediate Peclet numbers. This could be due to neglect of the cross-terms. *(see also August 22, 2000 entry below)* The tvd and central difference results appear identical for high diffusion coefficient, and the central difference is not stable for low diffusion coefficient.



August 22, 2000

S. Painter had an idea to use the source keyword rather that the boundary condition to set up a diagonal flow profile. This has the advantage of allowing us to set the source up orthogonal to the flow, not just aligned with the grid. I think this should allow us to use the patch solution from the 3-D case again, which will provide a nice quantitative check.

Here are some results from the source solution:



The central difference results are badly oscillatory for the central difference result with D=1.e-4. Thus, this result, which appears to be fairly non-diffusive, is actually misleading. It is really anti-diffusive, showing higher concentrations than are really warranted. Something similar, although of lesser degree, may have happened in the diagonal flow boundary condition solution from August 18.

August 28, 2000

I've finished up several runs similar to the above, using injected source but flow parallel to the x-axis. This should provide some quality control for comparing the diagonal flow to the aligned flow. I've also extracted several concentration profiles along streamlines, which should be useful in comparing with analytical solutions.

Here is the diagonal flow, source keyword injection input file:

```
  Flow through benchmark in 2d: diagonal flow with source terms
                August, 2000
:
:       geometry nx  ny  nz    ifrq  iprint iwarn idebug ibg1 ibg2
GRID      XYZ      60 60  1      1      1      1    0      1  3
:
DBASe
/home/chackert/multiflo/tvdtest/3dtest/master25.v8
:
:       method  iops  ifor  iflxlim  itmax ihalmx ndteq loglin  istepdt
```

```
OPTS       1        0      1       1         9     16     1      0         1
:   isurf  iact   isst  wtup  cournr  dtcutf  delcmax  qkmax  tpulse
     0      0      -1     0     10.e0    .25    1.44e-4   500
:
:       isync  ipor iperm  perm-fac.
COUPle   0      0     0       3.
:
PLTFiles
:iplot   a  s  t  m si sf  v  z  b in  e ex ti  g itex
    1    1  0  0  0  0  0  0  0  0  0  0  0  0  0    0
:
:     tol     ttol  tolneg tolmin tolexp  dthalf qkmax tolstdste tolc
TOLR  1.d-9 2.e0  1.e0  1.e-10  5.d0     .5     590.   1.e-12  1.e-12
:
:        mcyc  cc  c   flx r  sp  qk  pk  rk  a1  a2  a3
DEBUg     0        1  1   0   1   1   1   1   1
:
:       isat isothrm iread  por0  phir  sat  w  lambda toldelt  tolpor
ISYStem 0    0       0      .11     1.   0.5  .5   1.    1.e-3    1.e-3
:
:         vx0    vy0    vz0    vgx0    vgy0    vgz0[m/yr]    alphax      alphay
alphaz[m]
FLOW   0.707 0.707    0.    0.     0.     0.           0.         0.         0.
:
:   d0[cm^2/s]  delhaq[kJ/mol]  dgas[cm^2/s]  dgexp  tortaq  tortg idif
DIFF    1.d-3      12.6             2.13d-1       1.8   1.d0   1.d0    0
:
:      p (Pa) temp flag  a     b     c     d    x0    xlen
PTINit  1.e5   25.   0   0     0     0     0  1000.  2.d3
:
:master species for controlling time stepping
MASTER ALL
:
DXYZ
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.50 0.50 1.00 1.00 2.00 2.00
2.00 2.00 1.00 1.00 0.50 0.50 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25
0.25 0.25 0.25 0.25 0.50 0.50 1.00 1.00 2.00 2.00
1.
:
:   isolv level north nitmax idetail rmaxtol rtwotol smaxtol
SOLV    3   1     1    100       0     1.e-20  1.e-20  1.e-12
:
:initial and boundary conditions: 1-conc., 2-flux, 3-zero gradient
COMP
:
:i1  i2  j1  j2  k1  k2
  1   60  1  60  1   1
.
```

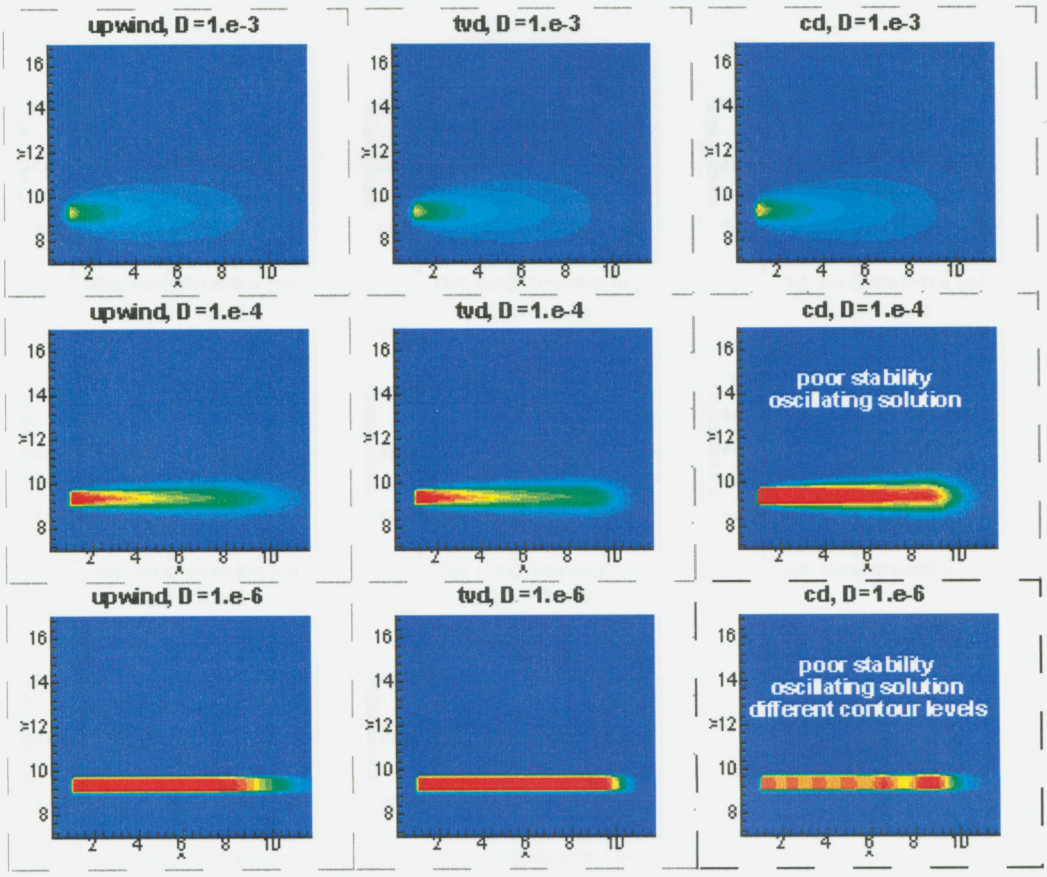
```
:species itype   ctot  mineral  diffusion
ca+2       7     8.e-4
.
:i1  i2  j1  j2  k1  k2
  5  5   17  17  1   1
.
:species itype   ctot  mineral  diffusion
ca+2       7     8.e-3
.
:i1  i2  j1  j2  k1  k2
  6  6   16  16  1   1
.
:species itype   ctot  mineral  diffusion
ca+2       7     8.e-3
.
:i1  i2  j1  j2  k1  k2
  7  7   15  15  1   1
.
:species itype   ctot  mineral  diffusion
ca+2       7     8.e-3
.
.
:
:    ns scale
SOUR 3   1.0
: i1 i2 j1 j2 k1 k2 istype
  5  5  17 17 1  1  11
: time temp q
   0.0 25.0 1.e-5
0
: species itype ctot conspec guess
   ca+2    7    8.e-3
: i1 i2 j1 j2 k1 k2 istype
  6  6  16 16 1  1  11
: time temp q
   0.0 25.0 1.e-5
0
: species itype ctot conspec guess
   ca+2    7    8.e-3
: i1 i2 j1 j2 k1 k2 istype
  7  7  15 15 1  1  11
: time temp q
   0.0 25.0 1.e-5
0
: species itype ctot conspec guess
   ca+2    7    8.e-3
:
BCON
:ibndtyp ifacx tmpbc dist area vell velg  por  sl  porm  slm  imtx
: i1  i2  j1  j2  k1  k2
: x influx
   1      1    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
   1   1   1   60   1   1
.
:species  itype  guess  ctot  mineral
ca+2       7     8.e-4
: y influx
```

```
   1      5    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
   1  60   1   1   1   1
.
:species  itype  guess  ctot  mineral
ca+2       7     8.e-4
: x outflux
    1      2    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
   60 60  1  60  1 1
.
ca+2       7     8.e-4
: y outflux
    1      6    25.    0.   0.   0.   0.   0.   0.   0.   0.   1
   1  60  60 60  1 1
.
ca+2       7     8.e-4
.
:
DTSTep[y]      3 1.e-7 1.e-5 1.e-2
1.e-8            1.d-4   1.e-3  1.e-0
:
TIME[y] 1 0.5
:
ENDS
```

I modified the keywords DIFF and IFOR to adjust the diffusion coefficient and the solution method. To make the straight flow, I reset the source cells to all fall on the same x value, and changed the flow to 1 m/yr along the x-direction.

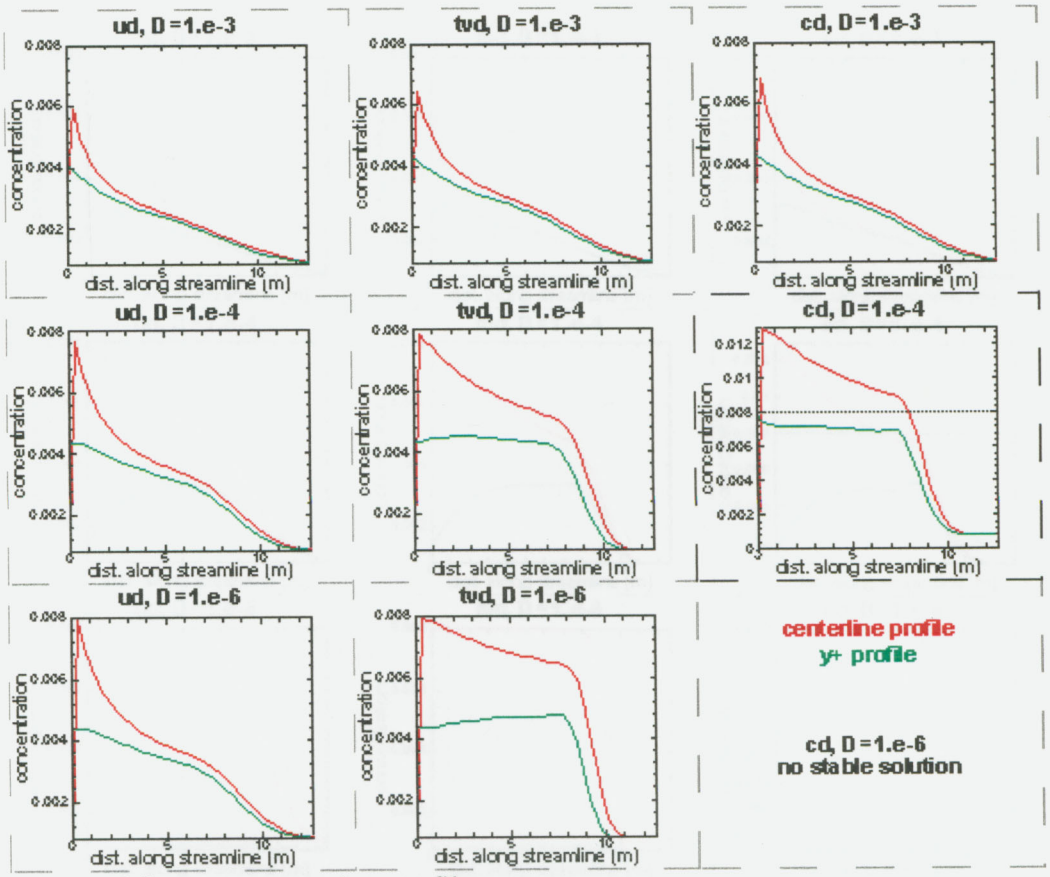
This means while both the diagonal and aligned source patches are 3 grid cells wide, the diagonal source patch is 0.75 m * sqrt(2) = 1.06 m wide, while the aligned source patch is only 0.75 m wide. I could make things more comparable by setting the grid size to 0.25*sqrt(2) for the aligned flow case, but let's hold off on that until we see if it is necessary.

The aligned flow results are in the graph below:

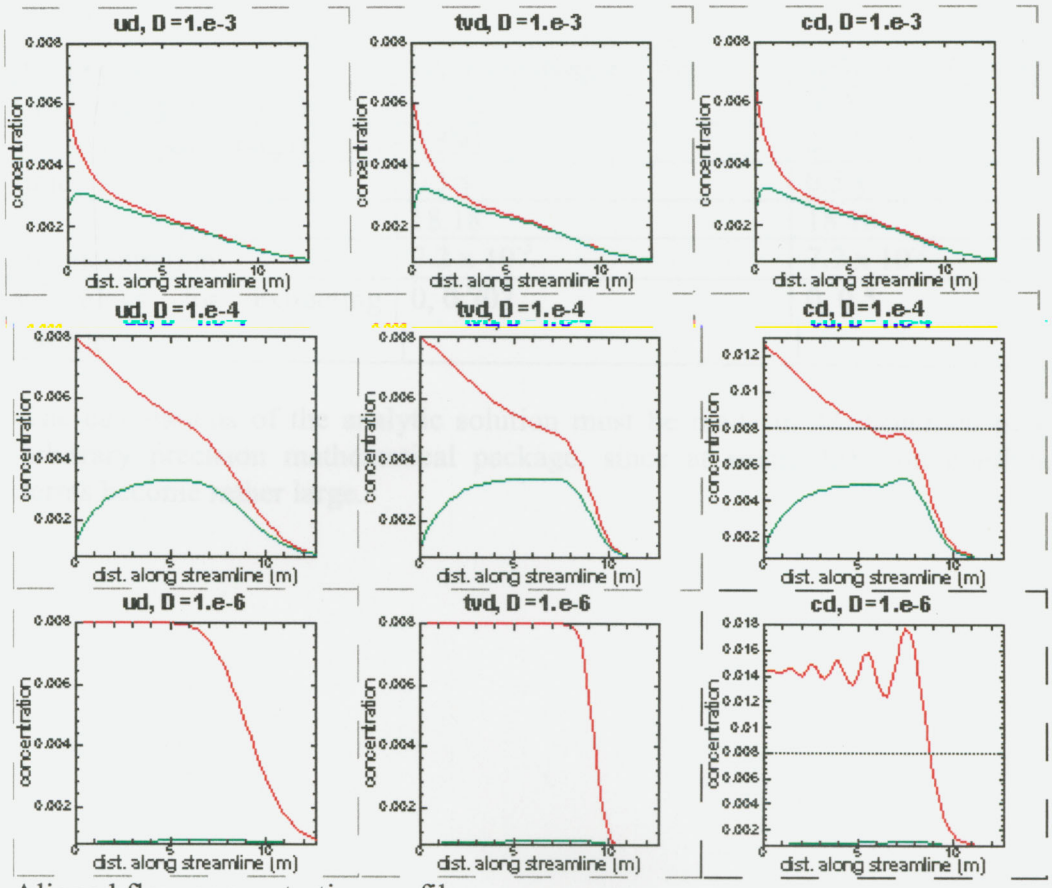


Results are fairly similar to the diagonal flow, except we can now get a solution (even if a very bad one) for the central difference, D=1.e-6 case. I've reset the contours for this image to show the oscillation. In this solution, as well as the central difference, D=1.e-4 case, the peak concentrations are well above the 8.e-3 we are trying to inject.

The next two images show concentration profiles for each case in both the diagonal and straight flow. The red curve is the concentration measured along the streamline running through the center of the source patch. The green line is measured along the streamline which is one cell in the y direction away from the top of the source patch.



Diagonal flow concentration profiles.



Aligned flow concentration profiles.

I should note that the same source injection rate is used for all cases. Thus, the concentration at the injection point may be expected to be slightly lower in the high diffusion cases since more of the injected tracer is diffused away. One of the curious things here is that the concentration at the source injection point is much higher than expected for the central difference low diffusion cases. I presume this is a result of the instability of this solution mechanism since at high diffusion coefficient, the central difference, upwind, and tvd mechanisms all yield pretty much the same result; at low diffusion coefficient, the tvd and upwind mechanisms have injection point concentrations near 8.e-3, as expected.

September 12, 2000

The only real task remaining on the current round of TVD testing would be to compute the analytic solution to compare to the above streamline concentration profiles. This can be done with the existing 3-D source patch solution, setting the z boundaries of the patch near infinity. Taking axes such that the flow is along the x direction and the patch is centered on $y = 0$, the following parameters could be used to compare with the above plots. The diffusion coefficient varies, and must be set for each test case.

parameter	*value for diagonal flow*	*value for aligned flow*
y1 (source patch edge)	-0.530	-0.375
y2 (source patch edge)	0.530	0.375
time	0.5 y	0.5 y
v	18.18	18.18
ΔConcentration	7.2×10^{-3}	7.2×10^{-3}
y values for extracting profiles	0, 0.707	0, 0.5

The calculations of the analytic solution must be made in *Mathematica* or some other arbitrary precision mathematical package, since at small diffusion coefficients some terms become rather large.

FINAL ENTRIES

Entries into Scientific Notebook No. 364E, Vol. 2, on pages 3 – 23 for the period of July 1, 2000 to September 12, 2000 were made by Chris Hackert.

No original text entered into this scientific notebook has been removed.

Chris Hackert

September 15, 2000