SOFTWARE VALIDATION REPORT FOR MULTIFLO VERSION 2.0.1

Prepared by

Scott Painter

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

April 2005

Approved by:

E. C. Pearcy — 4/14/2005

English C. Pearcy, Manager
Geochemistry

Date

TABLE OF CONTENTS

1.0 INTRODUCTION

This Software Validation Report documents software validation results for the computer code MULTIFLO, a numerical model describing two-phase nonisothermal flow and multicomponent reactive transport in variably saturated porous media. This software may be used as part of the review of the potential license application for the high-level waste repository for Yucca Mountain, Nevada.

The code can be used to address the drift scale and repository scale coupled thermal-hydrological-chemical processes that could affect the performance of the protential repository. The code can be applied to assess processes such as

- Isothermal and nonisothermal movement of water through unsaturated rock as liquid and vapor
- Evolution of groundwater compositions near and within the engineered barrier system
- Changes in porosity and permeability of the host rock resulting from mineral alteration and the resulting effects on fluid transport
- Transport of aqueous and gaseous radionuclides from the waste package

The software validation tests described in this report are required by the Quality Assurance Procedure TOP–018 and are a repeat of the previous software validation, which addressed MULTIFLO Version 1.5.2. Validation of MULTIFLO Version 2.0.1 is addressed in this report.

2.0 SCOPE OF THE VALIDATION

This software validation is for MULTIFLO Version 2.0.1, which supersedes the previous version of MULTIFLO. The MULTIFLO software is composed of METRA and GEM components, which can be used separately or in combination. Details of the software and its functions can be found in the MULTIFLO User's Manual (Painter and Seth, 2003) and in supporting technical material. This validation covers the major capabilities of the code that are to be used in regulatory activities. These capabilities include

(1) Nonisothermal multiphase flow and phase-change phenomena in partially saturated porous media

(2) Flow in composite fractured/porous media using a dual continuum formulation

(3) Flow in saturated porous media, including compressibility effects

(4) Advective and diffusive transport of chemicals in the aqueous and gaseous phase

(5) Equilibrium speciation of aqueous and gaseous phase constituents

(6) Kinetically controlled mineral formation and dissolution and resulting effects on porosity, permeability, and flow

(7) Unstructured grid configuration with arbitrary interblock connectivity

New capabilities included in Version 2.0.1 include improved representation of pumping wells and a node reordering scheme for improving the numerical performance for large unstructured grids. This well representation capability is not expected to be used in regulatory reviews and is not tested in this validation.

3.0 ENVIRONMENT

Validation tests were performed on the workstation farm directed by the computer known as Idaho, which uses the Solaris 5.8 operating system. MULTIFLO Version 2.0.1 was compiled using the Sun WorkShop Version 6 (update 2) FORTRAN 77 5.3. The commercial program Mathematica 5.0, running on the desktop computer Gorgon (MicroSoft® Windows XP Professional Version 2002), was used for some calculations to establish target solutions to the underlying mathematical equations. No special peripherals were used.

4.0 PREREQUISITES, ASSUMPTIONS, AND CONSTRAINTS

The GEM database is required for the tests. The location of the GEM database file must be specified in the GEM input datafile.

5.0 TEST CASES

5.1 Test Cases 1–8

Tests 1–8 are identical to the validation test for Version 1.5.2 (regression tests). Please refer to the Software Validation Report for Version 2.0 (Painter, 2003) for a description of the test cases.

Validation tests 1–8 were completed successfully and without error. Results for all validation tests are the same as for the validation for Version 1.5.2 and are consistent with the expected results. Software validation results are documented in Scientific Notebook 282E Volume 15. Relevant excerpts from Scientific Notebook 282E Volume 15 are attached to this report.

5.2 Test Case 9

5.2.1 Description of the Test

Test 9 is new for Version 2.0.1. Test 9 is designed to validate a node reordering scheme that is new to Version 2.0.1. The node reordering option allows the user to provide a node reordering permutation list that is used in the linear solver to reorder the nodes to a computationally more efficient configuration.

The test performs unstructured grid simulations with and without node reordering. The simulations are METRA thermal-hydrology simulations of the potential repository at

Yucca Mountain. The simulations use an unstructured grid and a dual permeability option. The two simulations are identical except that one uses the reordering option and the other does not.

5.2.2 Test Procedure

Test inputs are in the attached disk. To execute the first half of the test type *metra multi* at the unix command prompt. This runs the simulation without reordering. Then type *metra multi1* at the command prompt. This runs the simulation with nodal reordering. The unix *diff* utility is then used to compare the two files *multifldm1.xyp* and *multi1fldm2.xyp*.

5.2.3 Expected Results

Results of the two simulations should be identical except for differences in numerical rounding error. For the test simulation, the tolerances are set tight, and the results are printed in the *.xyp files.

5.2.4 Results

The test completed normally with no reported errors. The *multifldm1.xyp* and *multi1fldm1.xyp* files are identical except for the header information. Following is the results of the unix *diff* utility.

```
texas% diff multifldm1.xyp multi1fldm1.xyp
1c1
< TITLE="  1.000    y  n1= 942  Thu Mar 10 13:29:57 2005"
---
> TITLE="  1.000    y  n1= 942  Thu Mar 10 13:28:47 2005"
texas%
```

6.0 KNOWN PROBLEMS

No problems were identified in this software validation.

7.0 CONCLUSIONS

Results of all validation tests were consistent with the expected results, thereby providing confidence that the major capabilities probed by these tests have been correctly implemented in MULTIFLO.

This validation is considered a partial validation because not all capabilities of MULTIFLO were tested. In particular, the following capabilities were not tested:

- Extraction or injection wells
- Kinetically controlled aqueous phase reactions
- Ion exchange reactions

These capabilities are not expected to be used in regulatory reviews. If they are needed, the software validation should be updated to test the implementation of these capabilities.

8.0 REFERENCES

Painter, S., and M.S. Seth. "MULTIFLO User's Manual." MULTIFLO, Version 2.0: Two-Phase Nonisothermal Coupled Thermal-Hydrological-Chemical Flow Simulator. San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2003.

Painter, S. "Software Validation Test Report." MULTIFLO, Version 1.5.2. San Antonio, Texas: Center for Nuclear Waste Regulatory Analyses. 2003.

Test Case 1: Comparison with Doughty and Pruess's similarity solution

Scott Painter
revised 3-8-05 for Validation of 2.0.1

■ Load packages and set options

```
<< Graphics`Graphics`

$TextStyle = {FontFamily -> "Times", FontSize -> 16}

{FontFamily → Times, FontSize → 16}
```

conversion factor, years to seconds

```
yts = 365 * 24 * 60 * 60;
```

NO Vapor Pressure Lowering

(D&P solution taken from their Figure 6:parameters in Table 2:No Vapor Pressure Lowering)

■ Saturation

```
tmp = OpenRead["H:\\mflo-validation
      \\TestCase1\\novplfld1.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data1 = Transpose[data];

{TITLE="  0.1000      y  nl= 400  Tue Mar  8 15:42:04 2005",
 VARIABLES=" x     "," press"," temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh","  psat","  rhol","  rhog","   por   ",
 ZONE T= "   0.1000    ", I= 400 , F=POINT}
```

```
tmp = OpenRead["H:\\mflo-validation
      \\TestCase1\\novp1fld2.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data2 = Transpose[data];
```

```
{TITLE="   1.000      y  nl= 400  Tue Mar  8 15:47:16 2005",
 VARIABLES=" x      "," press"," temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh"," psat"," rhol"," rhog","   por  ",
 ZONE T= "   1.000    ", I= 400 , F=POINT}
```

```
tmp = OpenRead["H:\\mflo-validation
      \\TestCase1\\novp1fld3.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data3 = Transpose[data];
```

```
{TITLE="   6.000      y  nl= 400  Tue Mar  8 15:51:19 2005",
 VARIABLES=" x      "," press"," temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh"," psat"," rhol"," rhog","   por  ",
 ZONE T= "   6.000    ", I= 400 , F=POINT}
```

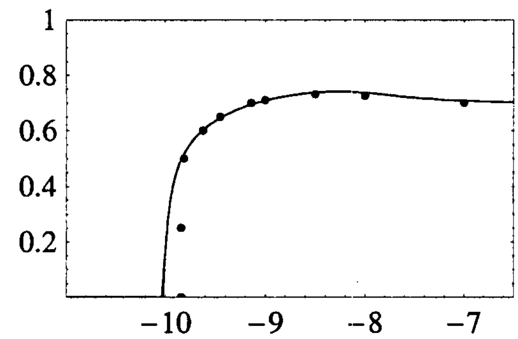
Data from D&P Figure 6

```
simsol = { {-9.84, 0}, {-9.84, 0.25}, {-9.81, 0.5}, {-9.62, 0.6}, {-9.45, 0.65},
  {-9.14, 0.70}, {-9.0, 0.71}, {-8.5, 0.731}, {-8.0, 0.726}, {-7, 0.7}}
```

```
{{-9.84, 0}, {-9.84, 0.25}, {-9.81, 0.5}, {-9.62, 0.6}, {-9.45, 0.65},
 {-9.14, 0.7}, {-9., 0.71}, {-8.5, 0.731}, {-8., 0.726}, {-7, 0.7}}
```

Figure 1

```
DisplayTogether[
   ListPlot[Transpose[{Log[ data1[[1]] / Sqrt[.1*yts]], data1[[4]]}],
  PlotRange -> {{-11, -6.5}, {0, 1.}}, PlotJoined -> True, Frame -> True] ,
 ListPlot[Transpose[{Log[ data2[[1]] / Sqrt[1*yts]], data2[[4]]}],
  PlotRange -> {{-11, -6.5}, {0, 1.}}, PlotJoined -> True, Frame -> True] ,
 ListPlot[Transpose[{Log[ data3[[1]] / Sqrt[6*yts]], data3[[4]]}],
  PlotRange -> {{-11, -6.5}, {0, 1.}}, PlotJoined -> True, Frame -> True] ,
   ListPlot[simsol, PlotStyle -> AbsolutePointSize[4]]]
```




```
- Graphics -
```

Position of dryout region. The value of the Boltzmann variable at the edge of the dryout zone is -9.84 from Doughty and Pruess. Find the value from the simulation and calculate a numerical value for relative error

```
ipos = Count[data1[[4]], 0.]

51

Log[ data1[[1, ipos]] / Sqrt[0.1yts] ]

-10.0399

(% + 9.84) / 9.84

-0.0203142
```

Vapor Pressure Lowering

(D&P solution taken from their Figure 6:parameters in Table 2:Vapor Pressure Lowering)

- **Saturation**

```
tmp = OpenRead["H:\\mflo-
    validation\\TestCase1\\vp1fld1.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data1 = Transpose[data];
```

```
{TITLE="  0.1000      y  n1= 400  Tue Mar  8 15:56:54 2005",
 VARIABLES=" x     "," press"," temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh","  psat","  rhol"," rhog","   por   ",
 ZONE T= "  0.1000     ", I= 400 , F=POINT}
```

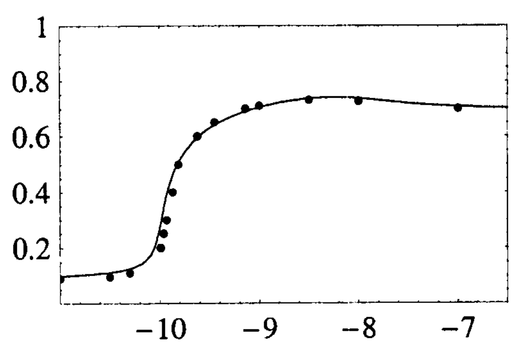
Data from D&P Figure 6

```
simsol = { {-11, 0.09}, {-10.5, 0.095}, {-10.3, 0.11}, {-9.99, 0.20}, {-9.96, 0.25},
  {-9.93, 0.30}, {-9.87, 0.4}, {-9.81, 0.5}, {-9.62, 0.6}, {-9.45, 0.65},
  {-9.14, 0.70}, {-9.0, 0.71}, {-8.5, 0.731}, {-8.0, 0.726}, {-7, 0.7}}
```

```
{{-11, 0.09}, {-10.5, 0.095}, {-10.3, 0.11}, {-9.99, 0.2}, {-9.96, 0.25},
 {-9.93, 0.3}, {-9.87, 0.4}, {-9.81, 0.5}, {-9.62, 0.6}, {-9.45, 0.65},
 {-9.14, 0.7}, {-9., 0.71}, {-8.5, 0.731}, {-8., 0.726}, {-7, 0.7}}
```

Figure 2

```
DisplayTogether[
    ListPlot[Transpose[{Log[ data1[[1]] / Sqrt[0.1*yts]], data1[[4]]}],
  PlotRange -> {{-11, -6.5}, {0, 1.}}, PlotJoined -> True, Frame -> True ],
   ListPlot[simsol, PlotStyle -> AbsolutePointSize[4] ]]
```



- Graphics -

Error in position of dryout region. Define dryout region to be 0.2 saturation for this simulation.

First find value from D&P

```
f = Interpolation[simsol]

InterpolatingFunction[{{-11., -7.}}, <>]

DPvalue = xx /. FindRoot[ f[xx] == 0.2, {xx, -10.1}]

-9.99
```

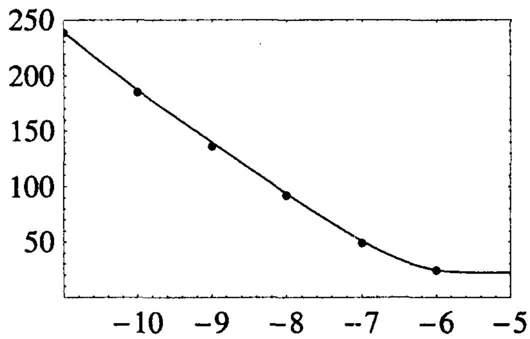
Find the value from the simulation and calculate a numerical value for relative error

```
f = Interpolation[ Transpose[{Log[data1[[1]] / Sqrt[0.1 * yts]], data1[[4]]}]]

InterpolatingFunction[{{-11.3941, -0.587844}}, <>]

metravalue = xx /. FindRoot[ f[xx] == 0.2, {xx, -10.1}]

-10.0546

(metravalue - DPvalue) / DPvalue

0.00646559

f = Interpolation[ Transpose[{Log[data1[[1]] / Sqrt[0.1 * yts]], data1[[4]]}]]

InterpolatingFunction[{{-11.3941, -0.587844}}, <>]

(f[simsol[[All, 1]] ] - simsol[[All, 2]])

{0.0102143, 0.0178139, 0.0167003, 0.0892342, 0.0941085, 0.0918429, 0.0615054, 0.00880573,
 -0.00409341, -0.00946171, -0.00966165, -0.00465352, 0.00458864, 0.0103119, 0.00780329}
```

■ compare temperature

```
temp = {{-11, 238}, {-10, 185}, {-9, 136}, {-8, 92}, {-7, 49}, {-6, 24}};

General::spell1 : Possible spelling error: new symbol name "temp" is similar to existing symbol "tmp". More…
```

```
DisplayTogether[
    ListPlot[
  Transpose[{Log[data1[[1]] / Sqrt[0.1*yts]], data1[[3]]}], PlotJoined -> True,
    Frame -> True, PlotRange -> {{-11, -5}, {0, 250}}] ,
    ListPlot[ temp, PlotStyle -> AbsolutePointSize[4] ]]
```



- Graphics -

Relative error

```
f = Interpolation[ Transpose[{Log[data1[[1]] / Sqrt[0.1 * yts]], data1[[3]]}]]
```

InterpolatingFunction[{{-11.3941, -0.587844}}, <>]

```
(f[temp[[All, 1]] ] - temp[[All, 2]]) / temp[[All, 2]]
```

{0.00241752, 0.0104766, 0.0278815, 0.0183406, 0.0358623, 0.019184}

Test 2:
Dual permeability version of Richard's equation

Solve numerically a dual permeability version of richards equation in 1-D.

Scott Painter
update 3.8.05 for Version 2.0.1 validation

▪ Load packages and set options

```
<< Graphics`Graphics`
```

```
$TextStyle = {FontFamily -> "Times", FontSize -> 16}
```

```
{FontFamily → Times, FontSize → 16}
```

Target solution:
Steady-state richards equation with specified boundary
DKM version

areamodf=1.e-6
areamf = 19.8 1/m;
blocksize=0.1 m;
transmssivity_mf = areamodf * areamf/(blocksize/2) = 0.000396
bulk fracture perm=matrix perm / 100

Van Genuchten parameters and relperm curve. Alpha given in terms of inverse meters.

```
a1 = 0.195
m1 = 6.7;
lambda1 = 1 - 1. / m1
krel1[h_] := If[ h >= 0, 1, Sqrt[(1 + (-a1 h) ^m1) ^-lambda1]
             (1 - ( (-a1 h) ^m1 / (1 + (-a1 h) ^m1)) ^lambda1) ^2]

0.195

0.850746
```

target matrix saturation at upper boundary and
corresponding cap pressure

```
tsat1 = 0.8;
tpc1 = (1 / a1) ( tsat1^ (-1 / lambda1) - 1) ^ (1 / m1);
```

Van Genuchten parameters and relperm curve. Alpha given in terms of inverse meters.

```
a2 = 0.195;
m2 = 6.7;
lambda2 = 1 - 1. / m2;
krel2[h_] := If[ h >= 0, 1, Sqrt[(1 + (-a2 h) ^m2) ^-lambda2]
             (1 - ( (-a2 h) ^m2 / (1 + (-a2 h) ^m2)) ^lambda2) ^2]
```

target fracture matrix saturation at upper boundary and
corresponding cap pressure

```
tsat2 = 0.4;
tpc2 = (1 / a2) ( tsat2^ (-1 / lambda2) - 1) ^ (1 / m2);
```

Solve DE by forward integration from water table up:

```
h0[ q01_, q02_] := Module[ {},
 hz = NDSolve[
   { {krel1[h1[z]] (h1'[z] - 1) + q1[z] == 0,  h1[10.] = -2},
     {q1'[z] - 0.000396 krel1[h1[z]] * (h2[z] - h1[z]) == 0, q1[10.] == q01},
    {0.01 krel2[h2[z]] (h2'[z] - 1) + q2[z] == 0,  h2[10.] ==  -2 },
    {q2'[z] + 0.000396 krel1[h1[z]] * (h2[z] - h1[z]) == 0, q2[10.] == q02}},
      { h1, q1, h2, q2},  {z, 00, 10}];
    { h1[0.] /. hz[[1, 1]] ,  h2[0.] /. hz[[1, 3]] }
   ]

{tpc1, tpc2}

{4.28453, 5.65961}
```

Initial guess must be in the feasible region
Find it by trial and error

```
h0[0.452121, 0.0022367]

{-4.28453, -5.6596}
```

Define saturation curve

```
sr = 0.;
sat1[h_] := If[h >= 0, 1, (1 - sr) (1 + (-a1 h) ^m1) ^-lambda1 + sr]

General::spell1 :
 Possible spelling error: new symbol name "sat1" is similar to existing symbol "tsat1". More...

sr = 0.;
sat2[h_] := If[h >= 0, 1, (1 - sr) (1 + (-a2 h) ^m2) ^-lambda2 + sr]

General::spell1 :
 Possible spelling error: new symbol name "sat2" is similar to existing symbol "tsat2". More...
```

Saturation associated with the specified capillary suction at depth h=10
use this as boundary condition in metra

```
sat1[ Evaluate[h1[10.0 ] /. hz[[1, 1]]] ]

0.998454

sat2[Evaluate[h2[10.0] /. hz[[1, 3]] ]]

0.998454
```

Saturation associated at top of boundary
use this as boundary condition in metra

```
sat1[ Evaluate[h1[0.0 ] /. hz[[1, 1]]] ]

0.8

sat2[Evaluate[h2[0.0] /. hz[[1, 3]] ]]

0.400002
```

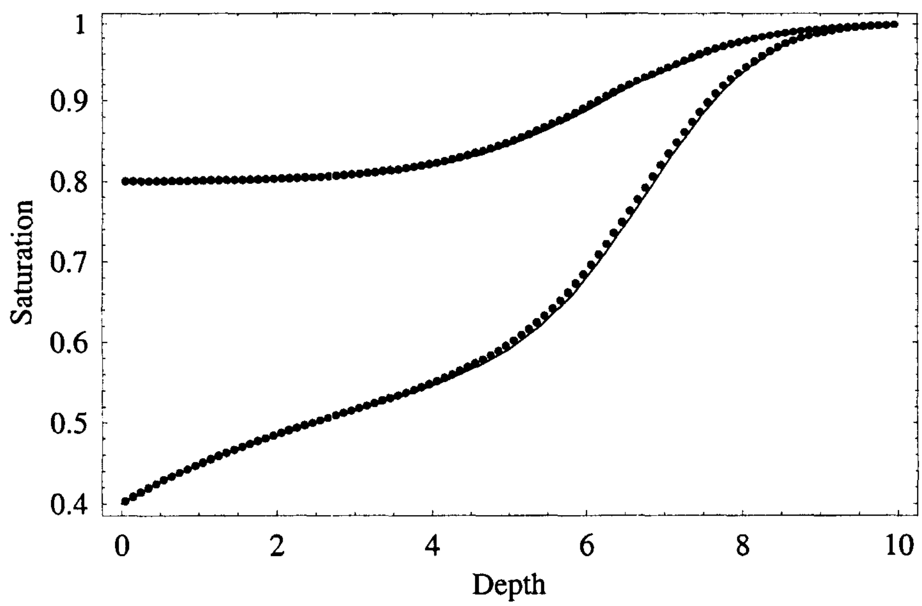
load the results from METRA and plot together

```
tmp = OpenRead["H:\\mflo-validation\\TestCase2\\sstate1f1df2.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
dataf = Transpose[data];

{TITLE="    152.4       y  n1= 200  Tue Mar  8 15:56:57 2005",
 VARIABLES=" depth"," press","  temp","    sl","    sg"," xairl"," xairg","   pcap","
      rh","  psat","  rhol","  rhog","   por  ",
 ZONE T= "    152.4      ", I= 200 , F=POINT}

General::spell1 :
 Possible spelling error: new symbol name "dataf" is similar to existing symbol "data". More...
```

```
tmp = OpenRead["H:\\mflo-validation\\TestCase2\\sstate1f1dm2.xyp"];
Read[tmp, {String, String, String}];
data = ReadList[tmp, Number];
data = Partition[data, 13];
datam = Transpose[data];
```

```
General::spell :
 Possible spelling error: new symbol name "datam" is similar to existing symbols {data, dataf}. More…
```

```
DisplayTogether[
    Plot[sat2[ Evaluate[h2[z] /. hz[[1, 3]]] ], {z, 0, 10}, PlotRange -> All],
     Plot[sat1[ Evaluate[h1[z] /. hz[[1, 1]]] ], {z, 0, 10}, PlotRange -> All],
    ListPlot[ Transpose[{datam[[1]], datam[[4]]}],
            PlotStyle -> {AbsolutePointSize[4]} ],
    ListPlot[ Transpose[{dataf[[1]], dataf[[4]]}],
            PlotStyle -> {AbsolutePointSize[4]}],
    Frame -> True,
    FrameLabel -> {"Depth", "Saturation"}
]
```



- Graphics -

Test 3:

Comparision with Theis' solution

Scott Painter
revised 3.8.05 for Version 2.0.1 Validation

■ Load packages and set options

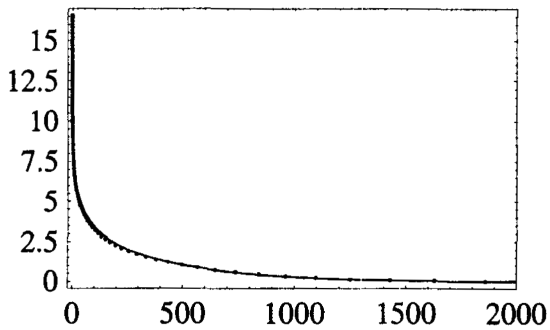
```
<< Graphics`Graphics`

$TextStyle = {FontFamily -> "Times", FontSize -> 16}

{FontFamily → Times, FontSize → 16}
```

compare drawdown

■ early time

```
tmp = OpenRead["H:\\mflo-validation
     \\TestCase3\\theisfld1.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data = Transpose[data];

{TITLE=" 1.0000E-04 y  n1= 100  Tue Mar  8 15:56:58 2005",
 VARIABLES=" x    "," press"," temp","  sl","   sg"," xairl"," xairg","  pcap","
      rh"," psat"," rhol"," rhog","  por  ",
 ZONE T= "  1.0000E-04", I= 100 , F=POINT}

yts = 365 * 24 * 60 * 60;

s = 9.4 10^-7;
hc = 10^-4;
Q = 1. / 1000.;
```

```
DisplayTogether[
ListPlot[Transpose[{ data[[1]], (10^6 - data[[2]]) / 9810. }],
  PlotRange -> {Automatic, Automatic}, Frame -> True],
    Plot[ -Q / (4 * Pi * hc ) ExpIntegralEi[ -r^2 s / (4 * hc 0.0001 yts ) ],
  {r, 0.1, 2000}],
        PlotRange -> { {-20, 2000}, Automatic} ]
```




```
- Graphics -
```

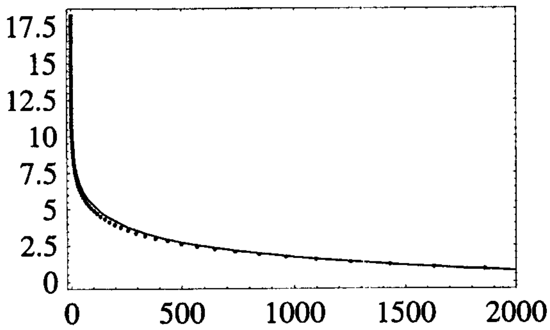
- **later time**

```
tmp = OpenRead["H:\\mflo-validation
      \\TestCase3\\theisfld2.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data = Transpose[data];
```

```
{TITLE="  1.0000E-03 y  nl= 100  Tue Mar  8 15:56:58 2005",
 VARIABLES=" x     "," press","  temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh","  psat","  rhol","  rhog","   por  ",
 ZONE T= "  1.0000E-03", I= 100 , F=POINT}
```

```
yts = 365 * 24 * 60 * 60;

s = 9.4 10^-7;
hc = 10^-4;
Q = 1. / 1000. ;
```

```
DisplayTogether[
ListPlot[Transpose[{ data[[1]], (10^6 - data[[2]]) / 9810. }],
  PlotRange -> {Automatic, Automatic}, Frame -> True],
    Plot[ -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.001 yts ) ],
  {r, 0.1, 2000}],
       PlotRange ->  { {-20, 2000}, Automatic} ]
```



- Graphics -

■ show together

```
tmp = OpenRead["H:\\mflo-validation
      \\TestCase3\\theisfld2.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data2 = Transpose[data];
```

```
{TITLE="  1.0000E-03 y  nl= 100  Tue Mar  8 15:56:58 2005",
 VARIABLES=" x      "," press","  temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh","  psat","  rhol","  rhog","   por  ",
 ZONE T= "  1.0000E-03", I= 100 , F=POINT}
```

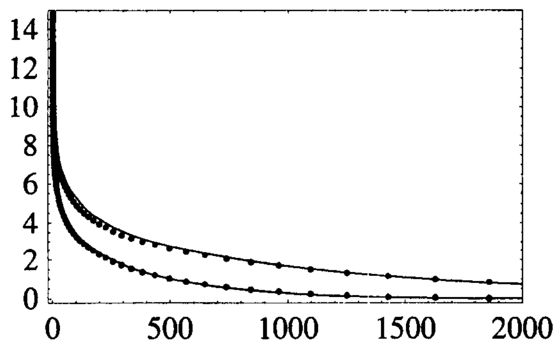
```
tmp = OpenRead["H:\\mflo-validation
     \\TestCase3\\theisfld1.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data1 = Transpose[data];
```

```
{TITLE=" 1.0000E-04 y  nl= 100  Tue Mar  8 15:56:58 2005",
 VARIABLES=" x    "," press","  temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh","  psat","  rhol","  rhog","   por  ",
 ZONE T= "  1.0000E-04", I= 100 , F=POINT}
```

```
yts = 365 * 24 * 60 * 60;
s = 9.4  10^-7 ;
hc = 10^-4;
Q = 1. / 1000. ;
```

Figure 5

```
DisplayTogether[
ListPlot[Transpose[{ data1[[1]], (10^6 - data1[[2]]) / 9810. }],
  PlotRange -> {Automatic, Automatic},
  Frame -> True, PlotStyle → AbsolutePointSize[3]],
    Plot[ -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.001 yts ) ],
  {r, 0.1, 2000}],
    ListPlot[Transpose[{ data2[[1]], (10^6 - data2[[2]]) / 9810. }],
  PlotRange -> {Automatic, Automatic},
  Frame -> True, PlotStyle → AbsolutePointSize[3]],
    Plot[ -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.0001 yts ) ],
  {r, 0.1, 2000}],

PlotRange -> { {-20, 2000}, {-0.2, 15} } ]
```



- Graphics -

Maximum error

```
Max[ ( (10^6 - data1[[2]]) / 9810.) ]
```

16.5851

```
Theis1[r_] :=  -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.0001 yts ) ]
```

```
(Theis1[ data1[[1]] ] - (10^6 - data1[[2]]) / 9810. ) / 16.56
```

{0.0913647, 0.0902601, 0.0890999, 0.0879988, 0.0868342, 0.0856707, 0.0845677, 0.0834044, 0.0822424, 0.0811422, 0.0799768, 0.0788761, 0.0777114, 0.0765483, 0.075447, 0.0742833, 0.073121, 0.0720186, 0.070853, 0.0697567, 0.0685932, 0.0674265, 0.0663261, 0.0651636, 0.0639992, 0.0628951, 0.0617323, 0.0606313, 0.0594681, 0.0583052, 0.0572034, 0.0560404, 0.0548766, 0.0537744, 0.0526118, 0.051513, 0.0503452, 0.0491844, 0.0480789, 0.046915, 0.0457527, 0.044653, 0.0434886, 0.0423861, 0.0412237, 0.0400604, 0.0389587, 0.0377971, 0.0366328, 0.0355325, 0.0343694, 0.0332672, 0.0321056, 0.0309417, 0.029846, 0.0286787, 0.0275192, 0.02642, 0.0252614, 0.0241013, 0.0230078, 0.0218501, 0.0206975, 0.0195486, 0.0184028, 0.0173242, 0.0161908, 0.0150056, 0.0138929, 0.0127958, 0.0116561, 0.0105479, 0.00940301, 0.00830617, 0.0072669, 0.00617484, 0.00512029, 0.00412379, 0.00310235, 0.00221763, 0.0012852, 0.000494963, -0.00025337, -0.000856727, -0.00138304, -0.00176096, -0.00191196, -0.00194644, -0.00187425, -0.00157244, -0.00123323, -0.000897095, -0.000574704, -0.000273443, -0.000128589, -0.0000364577, 3.57742×10^{-6}, 3.02523×10^{-7}, 1.299×10^{-8}, 2.31171×10^{-10}}

```
Max[ ( (10^6 - data2[[2]]) / 9810.) ]
```

18.2538

```
Theis2[r_] := -Q / (4 * Pi * hc ) ExpIntegralEi[ -r^2 s / (4 * hc 0.001 yts ) ]

(Theis2[ data2[[1]] ] - (10^6 - data2[[2]]) / 9810. ) / 18.25
```

{0.0918703, 0.0908122, 0.0897594, 0.0887602, 0.0877035, 0.0866477, 0.0856469, 0.0845913, 0.0835927, 0.0825386, 0.0814811, 0.0804823, 0.0794255, 0.0783701, 0.0773707, 0.0763149, 0.0753161, 0.0742598, 0.0732022, 0.0722074, 0.0711517, 0.0700929, 0.0690945, 0.0680396, 0.0670389, 0.0659812, 0.0649261, 0.063927, 0.0628715, 0.0618164, 0.0608166, 0.0597612, 0.0587611, 0.0577051, 0.0566501, 0.0556531, 0.0545934, 0.0535401, 0.0525369, 0.0514809, 0.050482, 0.0494283, 0.0483716, 0.0473712, 0.0463163, 0.0452606, 0.0442609, 0.0432067, 0.0422058, 0.0411513, 0.0400955, 0.0390949, 0.0380403, 0.0369833, 0.0359881, 0.0349275, 0.0339295, 0.0328739, 0.0318196, 0.0308189, 0.0297658, 0.0287087, 0.0277101, 0.0266565, 0.0256022, 0.0246047, 0.0235516, 0.0225, 0.0215046, 0.020455, 0.0194062, 0.0183649, 0.0173745, 0.0163363, 0.0153041, 0.0142759, 0.0132605, 0.0121981, 0.0112088, 0.0101794, 0.00917762, 0.00815255, 0.00716944, 0.00618666, 0.00521878, 0.0042871, 0.00336237, 0.00242301, 0.00156982, 0.000794801, 7.07078×10^{-7}, -0.000672347, -0.00124472, -0.00176916, -0.00218309, -0.00246769, -0.00278427, -0.00309511, -0.00352622, -0.00449916}

Test 4:
Equilibrium Speciation in GEM

Scott Painter
revised 3.8.05 for Version 2.0.1 Validation

Equilibrium Speciation

The following values are taken from the MULTIFLO out file. Values are activities (concentration*activity coefficient), except for gas, which is in partial pressure consistent with standard convention for denoting the log K for henry's law.

```
co2aq = 3.39703 10^-4;
hplus = 6.7183 10^-8 * 0.9227;
hco3minus = 2.7136 10^-3 * 0.9131;
caco3aq = 7.03072 10^-6 ;
ohminus = 1.78881 10^-7 * 0.91207;
co3minus2 = 2.68987 10^-6 * 0.69703;
caplus2 = 2.5 10^-3 * 0.7063;
pco2gas = 0.01;
h2o = 1.0;
```

Use the concentration values to calculate equilibrium constants. These can be compared to input values as reported from MULTIFLO output file.

Aqueous reactions:

Reaction 1

```
Log[ 10, h2o / ohminus / hplus ]
```

```
13.9951
```

Reaction 2

```
Log[10, ( co2aq^-1 * hplus * hco3minus / h2o)]
```

```
-6.34472
```

Reaction 3

```
Log[10, co3minus2^-1 / hplus hco3minus]
```

```
10.3288
```

Reaction 4

```
Log[10, caco3aq ^-1 * caplus2/hplus•hco3minus]
```

```
7.00167
```

Gas Reaction 5

```
Log[10, pco2gas^-1 * hplus * hco3minus /h2o]
```

```
-7.81362
```

Calcite mineral

```
Log[ 10, caplus2*hco3minus/hplus]
```

```
1.84867
```

Activity coefficient (B-dot)

use the B-dot formula to calculate the activity coefficients, using ionic strength as caculated by multiflo. These hand calccula-tion can then be compared to the MULTIFLO reported values for activity coefficients

```
A = 0.5114;
B = 0.3288;
bdot = 4.1000 10^-2;
ionic = 7.5028 10^-3;
```

```
GammaBdot[ionic_, a0_, z_] :=
 10^( -z^2 A Sqrt[ionic] / (1 + a0 B Sqrt[ionic]) + bdot ionic )
```

ca+2;

```
a0 = 6;
z = 2;
GammaBdot[ ionic, a0, z]
```

```
0.706284
```

h+;

```
a0 = 9;
z = 1;
GammaBdot[ ionic, a0, z]
```

```
0.922674
```

hco3-;

```
a0 = 4;
z = 1;
GammaBdot[ionic, a0, z]
```

```
0.913148
```

cl-;

```
a0 = 3;
z = 1;
GammaBdot[ionic, a0, z]
```

```
0.910957
```

Test 5:
Transport in Dual Permeability Media

Solves for transport in dual permeability media by semi-analytical method

Scott Painter
update 3/8/05 for Version 2.0.1 Validation
Note: Quit kernel and restart before running the mathematica script.

▪ Load packages and set options

```
<< Graphics`Graphics`

$TextStyle = {FontFamily -> "Times", FontSize -> 16}

{FontFamily → Times, FontSize → 16}

<< "D:\\MULTIFLO Project\\Validation\\Mathematica Notebooks\\NLapInv.m"
```

Compare with analytical solution
numerical solution = masin.2 file
Coupling between fractures and matrix areamodf=1.e-6

▪ First Fractures

diffusion coef m^2/y

```
d = N[ 10^-6 (60 60 24 365)]

31.536
```

mean position (porosity=0.11, sat=0.5)

```
v = 1. / (0.11 0.5)

18.1818
```

coupling terms (matrix block size = 0.1, sigmaf=0.01, areamodf=1.e-6)

```
lam = d 1.98 10^-5 100. / 0.05
lamp = d 1.98 10^-5 / 0.05
```

```
1.24883
```

```
General::spell1 : Possible spelling error: new symbol name "lamp" is similar to existing symbol "lam". More...
```

```
0.0124883
```

analytical solution

```
bigresult = Table[
        result = Table[
A = {
        { 0, 1, 0, 0},
        { (s + lam) / d, v / d, -lam / d, 0},
        {0, 0, 0, 1},
     {-lamp / d, 0, (s + lamp) / d, 0}
};
eval = Eigenvalues[A];

evect = Eigenvectors[A];
sol[x_] := c1 evect[[2]] Exp[ eval[[2]] x ] + c2 evect[[4]] Exp[eval[[4]] x];

foo = sol[xx] /. Solve[ { sol[0.][[1]] == 1. / s, sol[0.][[3]] == 1. / s}, {c1, c2}] ;
{s, foo[[1, 1]]},
        {s, 0.01, 30.01, 0.02}];
chat = Interpolation[result, InterpolationOrder -> 8];
chat2[s_] := If[s > 55.005, 0, chat[s]] ;

        {xx, NLInvSteh[ chat2[s], s, 0.9, 20] }, {xx, 0., 40., 2}];


bigresult1 = Transpose[
   {Transpose[ bigresult] [[1]], (0.008 - 0.0008) Transpose[bigresult] [[2]] + 0.0008}];

conc[x_, t_] := 0.0008 + 0.5 * (0.008 - 0.0008)
   (Erfc[ (x - v t) / Sqrt[4 d t]] + Exp[ x v / d] Erfc[(x + v t) / Sqrt[4 d t]])
```

```
tmp = OpenRead["H:\\mflo-validation
     \\TestCase5\\masin5_aqf2.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 3];
data1 = Transpose[data];
```

```
{!time=  0.9000      y     Tue Mar  8 15:57:00 2005,
 !distance     1-ca+2               2-         dliq,
   2.500000E-01       7.9803E-003       1.0000E-006}
```

▪ For the matrix

```
vm = 0. / (0.5 0.5)
```

```
0.
```

analytical solution

```
conc[x_, t_] := 0.0008 + 0.5 * (0.008 - 0.0008)
   (Erfc[ (x - vm t) / Sqrt[4 d t]] + Exp[x vm / d] Erfc[(x + vm t) / Sqrt[4 d t]])
```

```
lam = d 1.98 10^-5 100. / 0.05
lamp = d 1.98 10^-5 / 0.05
```

```
1.24883
```

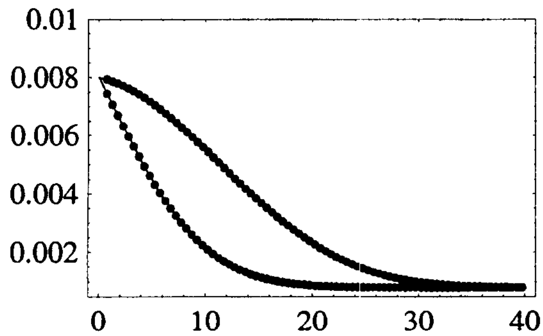
```
0.0124883
```

analytical solution

```
bigresult = Table[
        result = Table[
A = {
        { 0, 1, 0, 0},
        { (s + lam) / d, v / d, -lam / d, 0},
        {0, 0, 0, 1},
     {-lamp / d, 0, (s + lamp) / d, 0}
};
eval = Eigenvalues[A];
evect = Eigenvectors[A];
sol[x_] := c1 evect[[2]] Exp[ eval[[2]] x ] + c2 evect[[4]] Exp[eval[[4]] x];

foo = sol[xx] /. Solve[ { sol[0.][[1]] == 1. / s, sol[0.][[3]] == 1. / s}, {c1, c2}] ;
{s, foo[[1, 3]]},
        {s, 0.01, 30.01, 0.02}];
chat = Interpolation[result, InterpolationOrder -> 8];
chat2[s_] := If[s > 55.005, 0, chat[s]] ;

        {xx, NLInvSteh[ chat2[s], s, 0.9, 20] }, {xx, 0., 40., 2}];


bigresult2 = Transpose[
   {Transpose[ bigresult] [[1]], (0.008 - 0.0008) Transpose[bigresult] [[2]] + 0.0008}];

tmp = OpenRead["H:\\mflo-validation\\TestCase5\\masin5_aqm2.xyp"];
Read[tmp, {String, String, String}]
data9 = ReadList[tmp, Number];
data9 = Partition[data9, 3];
data9 = Transpose[data9];

{!time=  0.9000      y     Tue Mar  8 15:57:00 2005,
 !distance      1-ca+2              2~         dliq,
   2.500000E-01       7.8107E-003        1.0000E-006}
```

Figure 6 : concentration versus distance from inlet for fractures and matrix

```
DisplayTogether[
    ListPlot[Transpose[{data9[[1]], data9[[2]]}],
PlotRange -> {0.0005, 0.01}, PlotStyle -> AbsolutePointSize[4] ],
    ListPlot[bigresult2, PlotJoined -> True],
 ListPlot[Transpose[{data1[[1]], data1[[2]]}],
PlotRange -> {0.0005, 0.01}, PlotStyle -> AbsolutePointSize[4] ],
ListPlot[bigresult1, PlotJoined -> True] ,


        Frame -> True, Axes -> False]
```




```
- Graphics -
```

Error Fractures

```
f = Interpolation[bigresult1];
foo = f[data1[[1]] ] ;
Max[Abs[(foo - data1[[2]]) / foo ]]
```

```
0.0456528
```

Error matrix

```
f = Interpolation[bigresult2];
foo = f[data9[[1]] ] ;
Max[Abs[(foo - data9[[2]]) / foo ]]
```

```
0.0117006
```

Test 6:

Solute Transport in 3-D

Problem is a "Patch" source in a constant velocity field
Scott Painter
revised 3/8/05 for Version 2.0.1 validation

■ Load packages and set options

```
<< Graphics`Graphics`

$TextStyle = {FontFamily -> "Times", FontSize -> 16}

{FontFamily → Times, FontSize → 16}
```

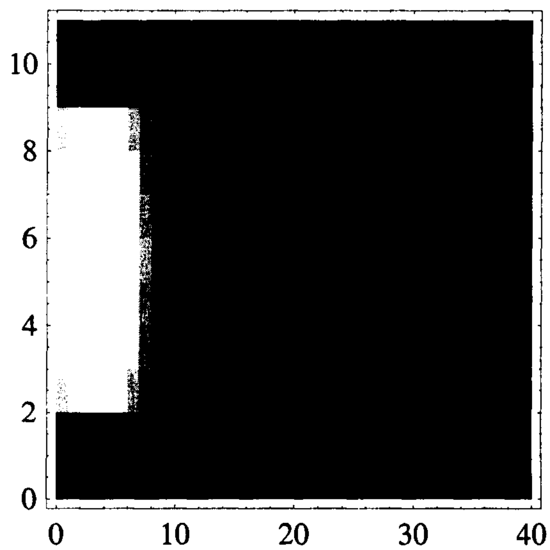
---

masin21a:

```
tmp = OpenRead["H:\\mflo-validation\\TestCase6\\masin21a_aq3.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
Close[tmp] ;
data = Partition[data, 4];
data = Transpose[data];
ca = data[[4]];

{TITLE= "    1.000      y     Tue Mar  8 15:57:26 2005, VARIABLES="      x-
   coord","      y-coord","      z-coord","ca+2          ","          dliq  ",
 ZONE T= "    1.000              y    " I=   40  , J=   11 K=   11F=POINT}

ca = Partition[ca, 40];
ca10 = Partition[ca, 11];
```

2-d slice at midplane

```
ListDensityPlot[ca10[[6]], Mesh -> False]
```




```
- DensityGraphics -
```

Diffusion coefficent (m^2/s)

```
dx = dy = dz = N[ 10^-7 (60 60 24 365)]
3.1536
```

Dimensions of patch source

```
z1 = -0.25;
z2 = 0.25;
y1 = -0.25;
y2 = 0.25;
```

Analytical solution (green's function)

```
cpatch[x_, y_, z_, t_] := (x*Exp[v*x/(2*dx)]/(8*Sqrt[Pi*dx]))*
    Exp[(-v^2)*(t/(4*dx)) - x^2/(4*dx*t)]*(Erfc[(y1 - y)/(2*Sqrt[dy*t])] -
     Erfc[(y2 - y)/(2*Sqrt[dy*t])])*(Erfc[(z1 - z)/(2*Sqrt[dz*t])] -
     Erfc[(z2 - z)/(2*Sqrt[dz*t])])/t^(3/2)
```

velocity (porosity=0.11, sat=0.5)

```
v = 1. / (0.11 0.5)
18.1818
```

Concentration at source - Initial concentration

```
co = 0.0072
```

```
0.0072
```

```
? cpatch
```

```
Global`cpatch
```

cpatch[x_, y_, z_, t_] :=
$$\frac{\left(x\, e^{\frac{v x}{2 dx}}\right) e^{-\frac{v^2 t}{4 dx}-\frac{x^2}{4 dx\, t}} \left(\text{Erfc}\left[\frac{y1-y}{2\sqrt{dy\, t}}\right]-\text{Erfc}\left[\frac{y2-y}{2\sqrt{dy\, t}}\right]\right) \left(\text{Erfc}\left[\frac{z1-z}{2\sqrt{dz\, t}}\right]-\text{Erfc}\left[\frac{z2-z}{2\sqrt{dz\, t}}\right]\right)}{\left(8\sqrt{\pi\, dx}\right) t^{3/2}}$$

Integrations to get solution. Various 1-D traces out of the 3-d solution.

Pick trace at(y=0,z=0.)

```
aresult1 = Table[
        {x, co*NIntegrate[cpatch[x, 0., 0., t], {t, 0, 1},
            MaxRecursion -> 16, MinRecursion -> 6] + 0.0008},
        {x, 0.125, 19.125, 0.25}];
```

Pick trace at(y=0,z=0.3)

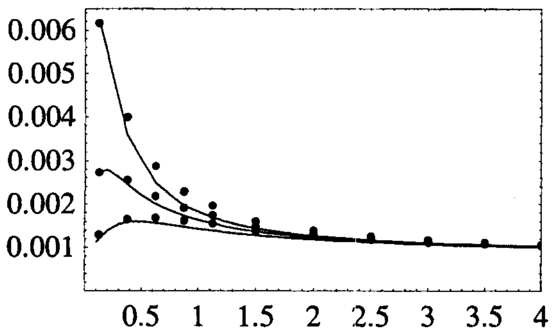
```
aresult2 = Table[
        {x, co*NIntegrate[cpatch[x, 0., 0.3, t], {t, 0, 1},
            MaxRecursion -> 16, MinRecursion -> 6] + 0.0008},
        {x, 0.1, 10, 0.1}];
```

Pick trace at(y=0,z=0.475)

```
aresult3 = Table[
        {x, co*NIntegrate[cpatch[x, 0., 0.475, t], {t, 0, 1},
            MaxRecursion -> 16, MinRecursion -> 6] + 0.0008},
        {x, 0.1, 15, 0.1}];
```

```
xpos = Take[data[[1]], 40];
```

```
DisplayTogether[
 ListPlot[Transpose[{xpos, cal0[[6, 9]]}],  PlotStyle -> AbsolutePointSize[4] ],
ListPlot[aresult3, PlotJoined → True],
 ListPlot[aresult2, PlotJoined → True],
 ListPlot[Transpose[{xpos, cal0[[6, 8]]}], PlotStyle -> AbsolutePointSize[4] ],
 ListPlot[Transpose[{xpos, cal0[[6, 6]]}], PlotStyle -> AbsolutePointSize[4]],
ListPlot[aresult1 , PlotJoined -> True ],
 PlotRange -> {{0, 4}, {0, 0.0065}} , Frame → True
]
```




```
- Graphics -
```

Check error

```
foo = Interpolation[aresult1]

InterpolatingFunction[{{0.125, 19.125}}, <>]

(foo[Take[xpos, 8]] - Take[cal0[[6, 6]] , 8]) / 0.008

{0.0104312, -0.0502396, -0.049765, -0.0399425,
 -0.0349595, -0.0194968, -0.015421, -0.0116214}

foo = Interpolation[aresult3]

InterpolatingFunction[{{0.1, 15.}}, <>]

(foo[Take[xpos, 8]] - Take[cal0[[6, 9]] , 8]) / 0.008

{-0.0104238, -0.00767424, -0.0149356,
 -0.0184627, -0.0200637, -0.0133678, -0.0107308, -0.00818524}

foo = Interpolation[aresult2]

InterpolatingFunction[{{0.1, 10.}}, <>]

(foo[Take[xpos, 8]] - Take[cal0[[6, 8]] , 8]) / 0.008

{0.00621939, -0.0124876, -0.0211421,
 -0.0225969, -0.0236019, -0.0146523, -0.012211, -0.0094089}
```

Test 7: Flow through with dissolution and permeability modifications

Scott Painter
Revised 3/8/05 for Version 2.0.1 Validation

Preliminaries

```
<< Graphics`Graphics`

$TextStyle = {FontFamily -> "Times", FontSize -> 16}

{FontFamily → Times, FontSize → 16}
```

Scenario 1: Advection and Diffusion: not fully dissolved

■ Get multiflo solutions

```
tmp = OpenRead["H:\\mflo-validation\\TestCase7\\masin92_vol3.xyp"];
Read[tmp, {String, String}]

{!time=  3.0000E+04 y    Tue Mar  8 15:57:41 2005,
 !distance      1-quartz*          2-    porosity     3-          por}

data = ReadList[tmp, Number];
data = Partition[data, 4];
data2 = Transpose[data];
quartz2 = data2[[2]];

Close[tmp]

H:\mflo-validation\TestCase7\masin92_vol3.xyp
```

■ Analytical with perm mod m=2

```
diff = 0.001 * 365 * 24 * 3600;
phi0 = 0.1;
```

```
phis0 = 0.5;
vsbar = 22.7;
delc = 9 10^-7;
v0 = 310.0;
L = 4000.;
kps = 3.15 / 4.;
```

```
General::spell1 :
 Possible spelling error: new symbol name "phis0" is similar to existing symbol "phi0". More...
```

```
a = 0.6 - phis0;
```

```
taus = phis0 / (kps delc vsbar)
```

```
31077.9
```

```
t = 30000 ;
b = t vsbar kps delc;
a2 = b + a;
```

```
    F[v_] := Module[ {},
    w = Sqrt[ 1 + 4 kps phi0 diff / (v) ^2];
  q = v / (2 phi0 diff) * (w - 1);
 lt = 0. ;
    a1 = b + a Exp[q * (L)];
    L (v0 / (2 v - v0) ) - ( b + a1 Log[a1]) / (a1 q) + (b + a2 Log[a2]) / (a2 q)  ];
```

Calculate velocity at 30000 years (cm/year). To be used later in analytical solution (Fsol[ ])

```
FindRoot[ F[v] == 0, {v, 300}]
```

```
{18.1818 → 382.869}
```

■ Compare with analytical solution

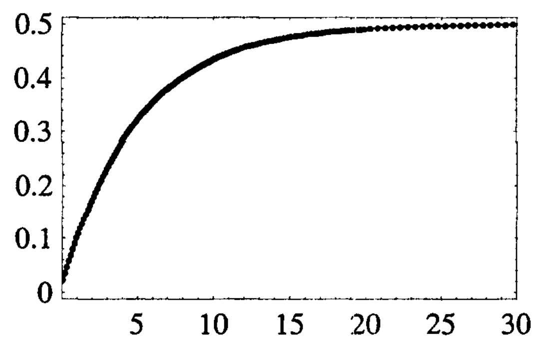
```
phis0 = 0.5;
vsbar = 22.69;
delc = (10^-3 - 10^-4) * 10^-3 ;
kps = 3.15 / 4.;
taus = phis0 / (kps * vsbar * delc) ;
diff = 0.001 * 365 * 24 * 3600 / 10^4 ;
phi = 0.1;
time = 30000.;
```

The following module is the analytical solution once the velocity is known

```
fsol[x_, v_] := Module[ {w, q, lt},
w = Sqrt[ 1 + 4 kps phi diff / (v) ^2];
q = v / (2 phi diff) * (w - 1);
lt = vsbar delc kps / (phis0 q) * (time - taus);
        lt = -1;
            If[ x > lt, phis0 * (1 - Exp[-q ( x )] * time / taus) , 0.] ]
```

General::spell1 : Possible spelling error: new symbol name "fsol" is similar to existing symbol "sol". More...

```
DisplayTogether[
ListPlot[ Transpose[{data2[[1]], quartz2}] , PlotStyle → AbsolutePointSize[3]],
Plot[ fsol[x,  3.82], {x, 0, 30}],
 Frame -> True, PlotRange -> {{0, 30}, Automatic}, Axes -> False ]
```



- Graphics -

■ Relative Error

```
target = Table[ fsol[ data2[[1, i]], 3.82], {i, 1, Length[data2[[1]]]}];
```

```
Max[ Abs[ (target - quartz2) / 0.5]]
```

0.00855967

Scenarios 2 and 3:
Advection dominated and fully dissolved at inlet

■ Get Multiflo Solutions

```
tmp = OpenRead["H:\\mflo-validation\\TestCase7\\masin80_vol4.xyp"];
Read[tmp, {String, String, String}]

{!time=  4.0000E+04 y    Tue Mar  8 15:58:14 2005,
 !distance     1-quartz*          2-    porosity     3-         por,
   6.250000E-02       0.0000E+00      6.0000E-01      1.0000E-01}

data = ReadList[tmp, Number];
data = Partition[data, 4];
data80 = Transpose[data];
quartz80 = data80[[2]];

Close[tmp]

H:\mflo-validation\TestCase7\masin80_vol4.xyp

tmp = OpenRead["H:\\mflo-validation\\TestCase7\\masin81_vol4.xyp"];
Read[tmp, {String, String, String}]

{!time=  4.0000E+04 y    Tue Mar  8 15:58:00 2005,
 !distance     1-quartz*          2-    porosity     3-         por,
   6.250000E-02       0.0000E+00      6.0000E-01      1.0000E-01}

data = ReadList[tmp, Number];
data = Partition[data, 4];
data81 = Transpose[data];
quartz81 = data81[[2]];

General::spell1 :
 Possible spelling error: new symbol name "data81" is similar to existing symbol "data1". More…

Close[tmp]

H:\mflo-validation\TestCase7\masin81_vol4.xyp
```

■ Analytical with perm mod m=2

```
phi0 = 0.1;
```

```
phis0 = 0.5;
vsbar = 22.7;
delc = 9 10^-7;
v0 = 310.;
L = 4000.;
    kps = 3.15 / 4.;

a = 0.6 - phis0;

taus = phis0 / (kps delc vsbar)

31077.9

time = 40000 ;
b = taus vsbar kps delc;
a2 = b + a;


        F[v_] := Module[ {},
        q = kps / v;
  lt = vsbar delc kps / (phis0 q) * (time - taus);
        a1 = b + a Exp[q * (L - lt)];
        L (v0 / (2 v - v0) ) -
   ( b + a1 Log[a1]) / (a1 q) + (b + a2 Log[a2]) / (a2 q) - (phi0 / 0.6) ^2 lt ];
```

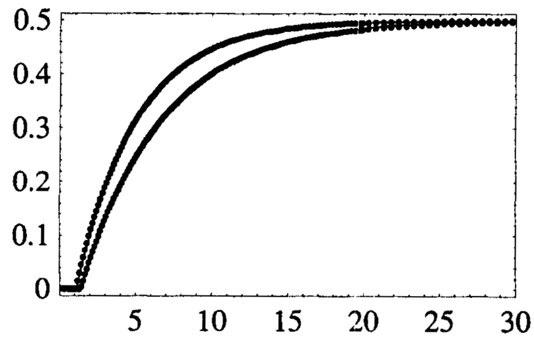
Calculate velocity at 40000 years (cm/year). To be used later in analytical solution (Fsol[ ])

```
FindRoot[F[v] == 0, {v, 400}]

{18.1818 → 400.479}
```

■ Compare with Analytical Solution

```
phis0 = 0.5;
vsbar = 22.69;
delc = (10^-3 - 10^-4) *10^-3;
kps = 3.15 / 4.;
taus = phis0 / (kps * vsbar * delc) ;
phi = 0.1;
time = 40000.;



fsol[x_, v_] := Module[
            {w, q, lt},
q = kps / v;
lt = vsbar delc kps / (phis0 q) * (time - taus);
            If[x > lt, phis0 * (1 - Exp[-q (x - lt)]) , 0.] ]
```

```
DisplayTogether[
    ListPlot[Transpose[{data80[[1]], quartz80}], PlotStyle → AbsolutePointSize[3] ],
    ListPlot[Transpose[{data81[[1]], quartz81}], PlotStyle → AbsolutePointSize[3]],
    Plot[fsol[x, 3.10], {x, 0, 40}],
    Plot[fsol[x, 4.00], {x, 0, 40}],
    Frame -> True, PlotRange → { {0, 30}, Automatic}]
```




```
- Graphics -
```

- **Error (normalized by initial mineral abundance)**

```
target = Table[fsol[data80[[1, i]], 3.1], {i, 1, Length[data80[[1]]]}];

Max[Abs[ (target - quartz80)] / 0.5]

0.0152697
```

- **Error (normalized by initial mineral abundance)**

```
target = Table[fsol[data81[[1, i]], 4.0], {i, 1, Length[data81[[1]]]}];

Max[Abs[ (target - quartz81)] / 0.5]

0.0143264
```

Test 8:
Comparison with Theis' solution using unstructured grid

Scott Painter
Revised 3/8/05 for Version 2.0.1 Validation

■ Load packages and set options

```
<< Graphics`Graphics`

$TextStyle = {FontFamily -> "Times", FontSize -> 16}

{FontFamily → Times, FontSize → 16}
```

---

compare drawdown

■ get the nodal positions

```
tmp = OpenRead["H:\\mflo-validation
      \\TestCase3\\theisfld1.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data = Transpose[data];
xpos = data[[1]];

{TITLE=" 1.0000E-04 y  nl= 100  Tue Mar  8 15:56:58 2005",
 VARIABLES=" x      "," press","  temp","    sl","    sg"," xairl"," xairg","   pcap","
       rh","  psat","  rhol","  rhog","   por  ",
 ZONE T= "  1.0000E-04", I= 100 , F=POINT}
```

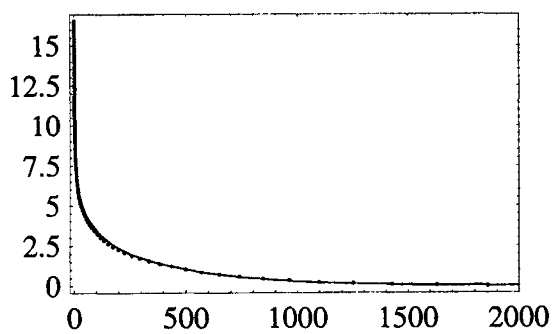
▪ early time

```
tmp = OpenRead["H:\\mflo-validation
      \\TestCase8\\unstructfld1.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data = Transpose[data];
```

```
{TITLE="  1.0000E-04 y  n1= 100  Tue Mar  8 15:58:16 2005",
 VARIABLES=" x     "," press","  temp","    sl","     sg"," xairl"," xairg","  pcap","
      rh","  psat","  rhol","  rhog","   por  ",
 ZONE T= "  1.0000E-04", I= 100 , F=POINT}
```

year to second conversion

```
yts = 365 * 24 * 60 * 60;

 s = 9.4  10^-7 ;
hc = 10^-4;
Q = 1. / 1000. ;

DisplayTogether[
ListPlot[Transpose[{ xpos,  (10^6 - data[[2]]) / 9810. }],
  PlotRange -> {Automatic, Automatic}, Frame -> True],
   Plot[  -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.0001 yts ) ],
  {r, 0.1, 2000}],
       PlotRange ->  { {-20, 2000}, Automatic} ]
```



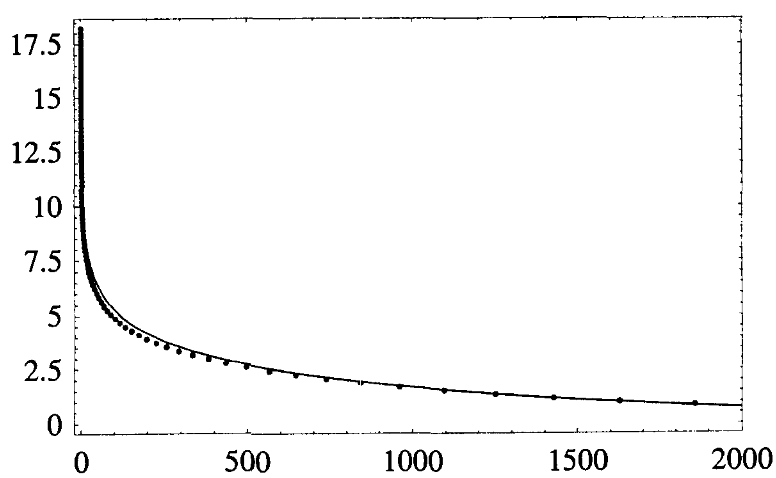
- Graphics -

- **later time**

```
tmp = OpenRead["H:\\mflo-validation
    \\TestCase8\\unstructfld2.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data = Transpose[data];
```

```
{TITLE="  1.0000E-03 y  nl= 100  Tue Mar  8 15:58:16 2005",
 VARIABLES=" x     "," press","  temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh","  psat","  rhol","  rhog","   por  ",
 ZONE T= "  1.0000E-03", I= 100 , F=POINT}
```

```
yts = 365 * 24 * 60 * 60;

s = 9.4  10^-7 ;
hc = 10^-4;
Q = 1. / 1000. ;

DisplayTogether[
ListPlot[Transpose[{ xpos,  (10^6 - data[[2]]) / 9810. }],
  PlotRange -> {Automatic, Automatic}, Frame -> True],
   Plot[  -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.001 yts ) ],
  {r, 0.1, 2000}],
       PlotRange ->  { {-20, 2000}, Automatic} ]
```



- Graphics -

- **show together**

```
tmp = OpenRead["H:\\mflo-validation
     \\TestCase8\\unstructfld2.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data2 = Transpose[data];
```

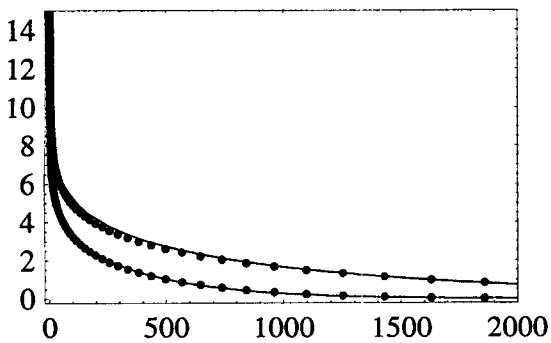
```
{TITLE="  1.0000E-03 y  n1= 100  Tue Mar  8 15:58:16 2005",
 VARIABLES=" x     "," press"," temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh"," psat"," rhol"," rhog","  por  ",
 ZONE T= "  1.0000E-03", I= 100 , F=POINT}
```

```
tmp = OpenRead["H:\\mflo-validation
     \\TestCase8\\unstructfld1.xyp"];
Read[tmp, {String, String, String}]
data = ReadList[tmp, Number];
data = Partition[data, 13];
data1 = Transpose[data];
```

```
{TITLE="  1.0000E-04 y  n1= 100  Tue Mar  8 15:58:16 2005",
 VARIABLES=" x     "," press"," temp","    sl","    sg"," xairl"," xairg","  pcap","
      rh"," psat"," rhol"," rhog","  por  ",
 ZONE T= "  1.0000E-04", I= 100 , F=POINT}
```

```
yts = 365 * 24 * 60 * 60;
s = 9.4 10^-7;
hc = 10^-4;
Q = 1./1000.;
```

```
DisplayTogether[
ListPlot[Transpose[{ xpos, (10^6 - data1[[2]]) / 9810. }],
  PlotStyle → AbsolutePointSize[4]],
    Plot[ -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.001 yts ) ],
  {r, 0.1, 2000}],
    ListPlot[Transpose[{ xpos, (10^6 - data2[[2]]) / 9810. }],
  PlotStyle → AbsolutePointSize[4]],
    Plot[ -Q  / (4 * Pi * hc ) ExpIntegralEi[ -r^2  s / (4 * hc  0.0001 yts ) ],
  {r, 0.1, 2000}],
PlotRange -> { {-20, 2000}, {-0.2, 15} }, Frame → True ]
```



- Graphics -