ADAPTIVE HYDRAULICS

A TWO-DIMENSIONAL MODELING SYSTEM
DEVELOPED BY THE COASTAL AND HYDRAULICS LABORATORY
ENGINEER RESEARCH AND DEVELOPMENT CENTER

A PRODUCT OF THE SYSTEM-WIDE WATER RESOURCES PROGRAM

SAMPLE PROBLEM

Using AdH to develop boundary conditions
For a simple rectangular flume

Sample problem

This document develops setting up a problem using AdH by demonstrating the development of the boundary condition file for a simple rectangular flume, Fig. 2.1, composed of triangular elements. The flume is 1000 m long and 50 m wide. The model slopes downward in the positive x-direction with a slope of 0.0009. The upstream (left end) bed elevation is 0.0 m. The downstream (right end) bed elevation is -0.9 m. The problem solved is for flow imposed on the upstream face with a velocity of 1.0 meter per second in the positive x direction. Flow exits directly downstream. The sidewalls have an imposed drag. The downstream tailwater elevation is imposed at an elevation of 0.1 m. Just for the sake of an example the mesh is composed of two material types. The right half of the grid is material 2 and the left half is material 1 (see Fig. 2.2).

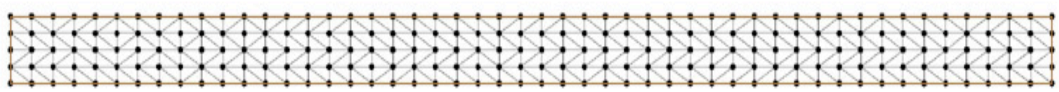

Figure 2.1: Rectangular Flume Mesh.

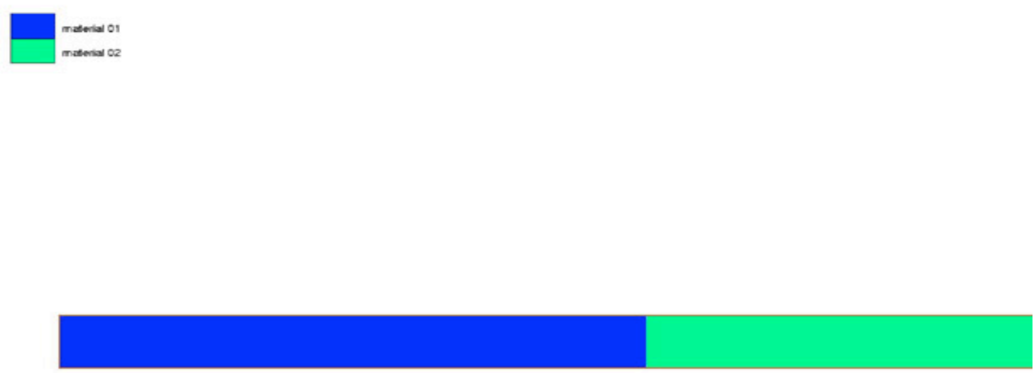

Figure 2.2: Rectangular Flume Materials.

Sample problem A: Geometry file

A portion of the geometry file is shown here. It is composed of triangular elements denoted by the **E3T** card. The term "MESH2D'' starts the file. Each element is "E3T Element-# 3-Node-#'s Material ID''. This particular example has 392 triangular elements. This list is followed by the list of nodal coordinates. These are "ND Node-# X, Y, and Z coordinates''.

```
MESH2D
E3T 1 2 1 6 1
E3T 2 7 2 6 1
E3T 3 8 2 7 1
E3T 4 8 3 2 1
E3T 5 4 3 8 1
E3T 6 9 4 8 1
E3T 7 5 4 9 1
E3T 8 10 5 9 1
E3T 9 7 6 11 1
E3T 10 12 7 11 1
.   .   .   .   .   .
.   .   .   .   .   .
.   .   .   .   .   .
E3T 381 239 238 243 2
E3T 382 244 239 243 2
E3T 383 240 239 244 2
E3T 384 245 240 244 2
E3T 385 242 241 246 2
E3T 386 247 242 246 2
E3T 387 248 242 247 2
E3T 388 248 243 242 2
E3T 389 244 243 248 2
E3T 390 249 244 248 2
E3T 391 245 244 249 2
E3T 392 250 245 249 2
ND 1 0.00000000e+00 0.00000000e+00 0.00000000e+00
ND 2 0.00000000e+00 1.25000000e+01 0.00000000e+00
ND 3 0.00000000e+00 2.50000000e+01 0.00000000e+00
ND 4 0.00000000e+00 3.75000000e+01 0.00000000e+00
ND 5 0.00000000e+00 5.00000000e+01 0.00000000e+00
ND 6 2.04081633e+01 0.00000000e+00 -1.83673469e-02
ND 7 2.04081633e+01 1.25000000e+01 -1.83673469e-02
.   .   .   .   .
.   .   .   .   .
.   .   .   .   .
ND 241 9.79591837e+02 0.00000000e+00 -8.81632653e-01
ND 242 9.79591837e+02 1.25000000e+01 -8.81632653e-01
ND 243 9.79591837e+02 2.50000000e+01 -8.816326531-01
ND 244 9.79591837e+02 3.75000000e+01 -8.81632653e-01
ND 245 9.79591837e+02 5.00000000e+01 -8.81632653e-01
ND 246 1.00000000e+03 0.00000000e+00 -9.00000000e-01
ND 247 1.00000000e+03 1.25000000e+01 -9.00000000e-01
ND 248 1.00000000e+03 2.50000000e+01 -9.00000000e-01
ND 249 1.00000000e+03 3.75000000e+01 -9.00000000e-01
ND 250 1.00000000e+03 5.00000000e+01 -9.00000000e-01
```

Sample problem: Hotstart file

A hotstart file is given below. The first part is the initial estimate of depth. In this case they are all set to 1.0. If either depths and/or velocities are not in the hotstart file their values are assumed to be zero. These hotstart files can easily be developed from output from previous runs by simply copying the timestep that you want to use from the output file. The "NAME'' row then is modified to match the hotstart designation. This is the depth part of the hotstart file, as indicated by the name "ioh''.

```
DATASET
OBJTYPE "mesh2d"
BEGSCL
ND 250
NC 392
NAME ioh
TS 0 0.00000000e+00
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
ENDDS
```

The velocity portion ("iov'') of the hotstart file then is:

```
DATASET
OBJTYPE "mesh2d"
BEGVEC
ND 250
NC 392
NAME iov
TS 0 0.00000000e+00
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
.   .   .
.   .   .
.   .   .
1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0
ENDDS
```

Sample problem A: Boundary condition file

Next is shown the boundary condition file. The first portion of the file contains the **OP**, **IP**, and **MP** cards. The **OP** cards designate that this is a two-dimensional Shallow Water run, that memory is added in blocks of 40, no transport constituents, there are to be 1 block per processor, and the preconditioner uses the two-level additive Schwarz approach. The **IP** cards indicate that the maximum number of linear iterations allowed are 200, the maximum non-linear iterations is 8, and the tolerance on the non-linear iterations is 1.0E-5. The **MP** cards designate the other properties of the fluid for this particular problem.

```
OP  SW2
OP  INC  40
OP  TRN 0
OP  PRE  2
OP  BLK 1

IP  IT 200
IP  NIT 8
IP  NTL  1.0E-5

MP ML  1  0
MP COR  1  0
MP SRT   1  10
MP  EVS  1  0.010  0.010  0.010
MP ML  2  0
MP COR  2  0
MP SRT   2  10
MP EVS  2  0.010  0.010  0.010
MP MUC 1.0
MP MU  0.0
MP G  9.8
MP RHO  1000.0
```

The "strings'' are designated by **NDS** for node strings, **EGS** for edge strings, and **MTS** for material strings. Material 1 and 2 make up strings 1 and 6, respectively. Strings 2, 3, and 4 are edges that compose the upstream, sidewalls, and downstream edges of the domain, respectively. Note that both the left and right sidewalls have been designated as the same string. They can be separate strings. String 5 is a list of the upstream boundary nodes.

```
!UPSTREAM INFLOW
NDS 1 5
NDS 2 5
NDS 3 5
NDS 4 5
NDS 5 5

!UPSTREAM EDGE
EGS 1 2 2
EGS 2 3 2
EGS 3 4 2
EGS 4 5 2

!RIGHT SIDEWALL
EGS 1 6 3
EGS 6 11 3
EGS 11 16 3
EGS 16 21 3
EGS 21 26 3
EGS 26 31 3
EGS 31 36 3
EGS 36 41 3
EGS 41 47 3
EGS 47 51 3
EGS 51 56 3
EGS 56 61 3
EGS 61 66 3
EGS 66 71 3
EGS 71 76 3
EGS 76 81 3
EGS 81 86 3
EGS 86 92 3
EGS 92 98 3
EGS 98 101 3
EGS 101 106 3
EGS 106 111 3
EGS 111 116 3
EGS 116 121 3
EGS 121 126 3
EGS 126 131 3
EGS 131 136 3
EGS 136 142 3
EGS 142 146 3
EGS 146 151 3
EGS 151 156 3
EGS 156 161 3
EGS 161 166 3
EGS 166 172 3
EGS 172 177 3
EGS 177 182 3
EGS 182 188 3
EGS 188 191 3
EGS 191 196 3
EGS 196 201 3
EGS 201 206 3
EGS 206 211 3
EGS 211 216 3
EGS 216 221 3
EGS 221 226 3
EGS 226 231 3
EGS 231 236 3
EGS 236 241 3
EGS 241 246 3
```

```
!LEFT SIDEWALL
EGS 5 10 3
EGS 10 15 3
EGS 15 20 3
EGS 20 25 3
EGS 25 30 3
EGS 30 35 3
EGS 35 40 3
EGS 40 45 3
EGS 45 50 3
EGS 50 55 3
EGS 55 60 3
EGS 60 65 3
EGS 65 70 3
EGS 70 75 3
EGS 75 80 3
EGS 80 85 3
EGS 85 90 3
EGS 90 95 3
EGS 95 100 3
EGS 100 105 3
EGS 105 110 3
EGS 110 115 3
EGS 115 120 3
EGS 120 125 3
EGS 125 130 3
EGS 130 135 3
EGS 135 140 3
EGS 140 145 3
EGS 145 150 3
EGS 150 155 3
EGS 155 160 3
EGS 160 165 3
EGS 165 170 3
EGS 170 175 3
EGS 175 180 3
EGS 180 185 3
EGS 185 190 3
EGS 190 195 3
EGS 195 200 3
EGS 200 205 3
EGS 205 210 3
EGS 210 215 3
EGS 215 220 3
EGS 220 225 3
EGS 225 230 3
EGS 230 235 3
EGS 235 240 3
EGS 240 245 3
EGS 245 250 3

!DOWNSTREAM EDGE
EGS 246 247 4
EGS 247 248 4
EGS 248 249 4
EGS 249 250 4

MTS 1 1
MTS 2 6
```

The XY-series are shown next. These **XY1** cards are followed by the series number, the number of points in the series, and two unit specifiers. The first value in the series is the independent timestep and the second number is the dependent variable describing the condition (depth, velocity, etc.). The series are referenced in the boundary condition cards. Note that series 4 includes the **XY1** card but just before it is the **XYT** card. This **XYT** card simply tells ADH the tolerance for which values will be interpolated in this series. This means that it will only cut the time step to the point that it has an accuracy of 1.0 in interpolating this series.

```
XY1   1   2   0   0
0.0     0.0
2000.0   0.0

XY1   2   2   0   0
0.0     1.0
2000.0   1.0

XY1   3   8   0   0
100.        0.0
200.        0.0
400.        0.0
600.        0.0
800.        0.0
1200.       0.0
1600.       0.0
2000.       0.0

XYT   4  1.0
XY1   4   4   0   0
0.0       20.0
100.0     40.0
200.0     100.0
2000.0   100.0

XY1   5   2   0   0
0.0     0.1
2000.0   0.1
```

The following lines represent the applied boundary conditions. The first 6 lines containing **FR MNG**, all have Manning's n of 0.03. Strings 1, 3, and 6 have roughness that is non-zero. Strings 1 and 6 are the 2D elements associated with material types 1 and 2. String 3 is the sidewall of the flume. The next 3 lines are Neumann boundary conditions. These are the **NB OVL** cards. Each is followed by a string number, in this case the 3 strings are 1,3, and 6. The final number represents which series is to be used to get the flux into or out of this string. All of them are associated with series number 1. XY1 series 1 has 0.0 throughout all time. So there is no flux into or out of each of these strings.

The next boundary condition card **NB OTW 4 5** signifies that string 4 will have a tailwater elevation specified. And that elevation is given on series # 5. If we check that series we see that the elevation specified there has a value of 0.1 for the entire simulation.

The next two cards deal with the upstream boundary condition. The first, given by the DB OVL 5 2 1 card, says that we intend to enforce a velocity Dirichlet boundary condition on string # 5. These are the upstream nodes. The values of the x and y-components of velocity are given on series 2 and 1, respectively. The next card is **OB OF** 2, which tells AdH to process this string so that flow is allowed to pass through it.

```
FR MNG  1  0.03
FR MNG  2  0.03
FR MNG  3  0.03
FR MNG  4  0.03
FR MNG  6  0.03

NB OVL    1   1
NB OVL    3   1
NB OVL    6   1
NB OTW    4   5
DB OVL    5   2   1
OB OF     2
```

The **OC** card designates series 3 to contain the list of timesteps to be output. Series 3 requests that output be generated at times 100.0, 200.0, 400.0, 600.0, 800.0, 1200.0, ... seconds. **TC T0** and **TC TF** designate the starting and ending times to be 0.0 and 1200.0 respectively. The timestep size is setup by the **TC IDT** card. The timestep size is set by series 4, in this case. Series 4 then indicates that a time step size progresses from 20.0 seconds to 100.0 seconds throughout the run.

```
OC 3

TC T0 0.0
TC IDT 4
TC TF 1200.0
TC NDP

END
```

Running AdH

ADH is run after a setup program called PRE_ADH. PRE_ADH reads the input files and checks that they are somewhat consistent. It also checks the geometry to make sure that the elements are properly formed. The actual command to run PRE_ADH on this sample problem is:

Pre_adh uniform

At this point the program itself can be run. Depending upon the flags set during compilation the program can be run with or without "MPI''. When running ADH on the pc, the command is:

Adh uniform

This will create files called "uniform_ovl.dat'' ,"uniform_dep.dat'', and "uniform_err.dat'', for velocities, depths, and errors for adaption, respectively. These can be directly imported into GMS or SMS.
If instead the compiler flags are set to run MPI, the run command is as follows:

Mpirum –np 2 adh uniform

This will create files called "uniform_ovl.dat'' ,"uniform_dep.dat'', and "uniform_err.dat'', for velocities, depths, and errors for adaption, respectively. These can be directly imported into GMS or SMS.

When including transport quantities, output files will be generated for the concentration of each constituent. If using the sediment transport option, bed displacements and other variables will be given along with the concentrations for each sediment.

These results are shown in the next three figures. Fig. 2.3 shows velocity results, Fig. 2.4

shows overland head results, and Fig. 2.5 shows the error used for adapting at time 1200.0 seconds.

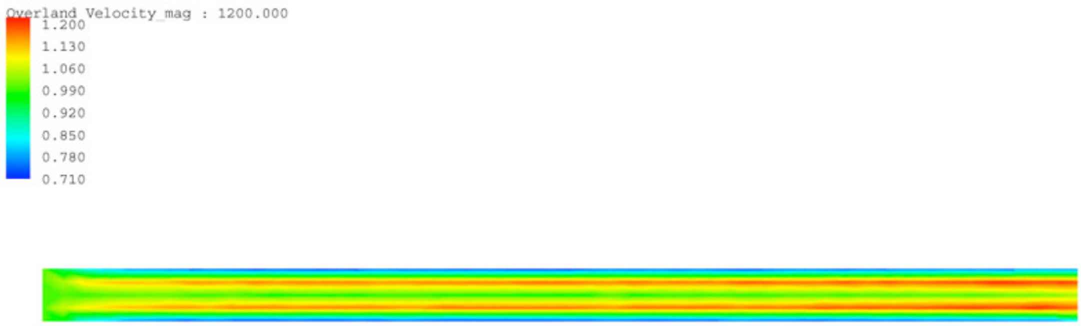

Figure 2.3: Velocity Results.

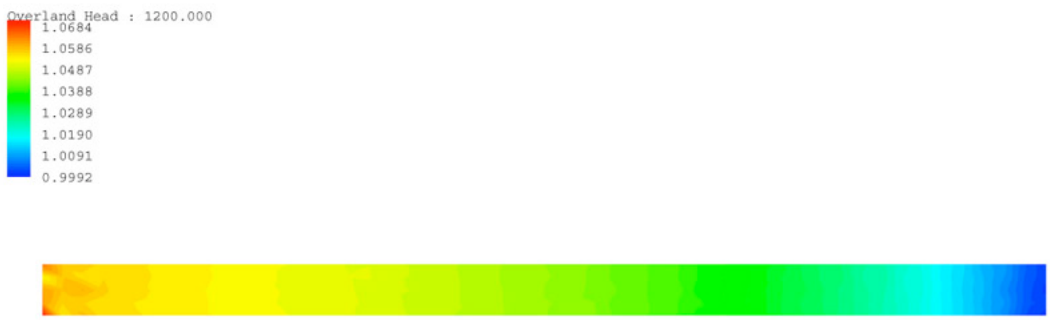

Figure 2.4: Head Results.

Figure 2.5: Error Results.

Modifying the boundary conditions

The previous example can be modified easily in the boundary condition (.bc) file. The maximum timestep can be increased or decreased by changing the **TC TF** card. Be certain to modify the **XY1** series cards if necessary so that they also reflect the change in run length. The Manning's "n'', or roughness, can be manipulated in the FR MNG card for each edge or material.

Supercritical flow

The following change will generate supercritical flow and a hydraulic jump will be observed. To properly pose a supercritical inflow boundary, we need to specify velocity and depth. For subcritical inflow we only need velocity specified. Suppose the Froude number is 2. The corresponding horizontal velocity is 3.333 m/s and the corresponding tailwater height is 0.3 m. Open the uniform.bc file and change the **XY1** series 2 card to 3.333 since it corresponds to the x-velocity component at all timesteps. Then add an X**Y1** series 6 card to equal 0.3 at all timesteps. It will represent the tailwater head. Also replace the **DB OVL** card with a **DB OVH** card making series 6 the depth for node string 5. Re-run PRE_ADH and ADH with the modified boundary condition file. The results should show a downward jump as seen in Fig. 2.6 and Fig. 2.7. Since the refinement level (MP ML) is set to zero meaning no adaption is taking place, the error files will not be shown for this example.

Figure 2.6: Velocity Results for Supercritcal Flow.

Figure 2.7: Overland Head Results for Supercritcal Flow.

Rain and evaporation

As noted previously, rain and evaporation can be modeled with AdH. Since the example mesh is divided into two material types, rain will be applied on material 1 and evaporation on material 2. This can be done easily by modifying the **XY1** series cards in the boundary condition file. Using the original .bc file, add an **XY1** series 6 card so that it equals 0.0001 at all timesteps. Also add an **XY1** series 7 card that is -0.0001 at all timesteps. Keep in mind that the volume raining in or evaporating out cannot greatly exceed the volume of the flume. Now modify the **NB OVL** card so that string 1 (material 1) uses series 6 and string 6 (material 2) uses series 7. The velocity results of this case at 1200 seconds can be seen in Fig. 2.8 and the overland head results in Fig. 2.9.

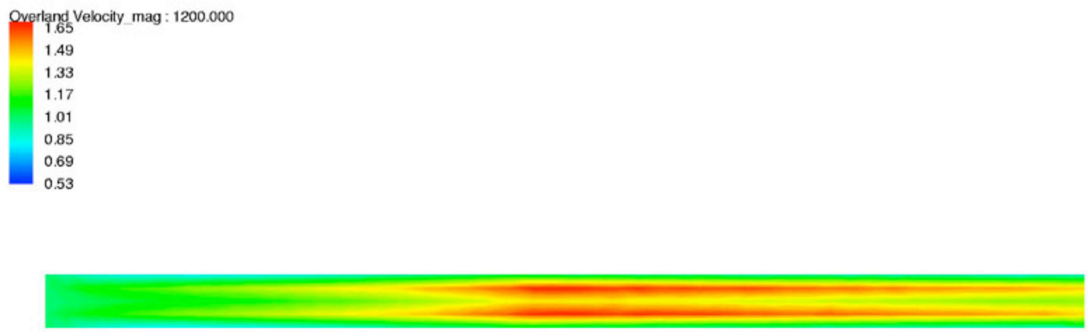

Figure 2.8: Velocity Results for Rain and Evaporation.

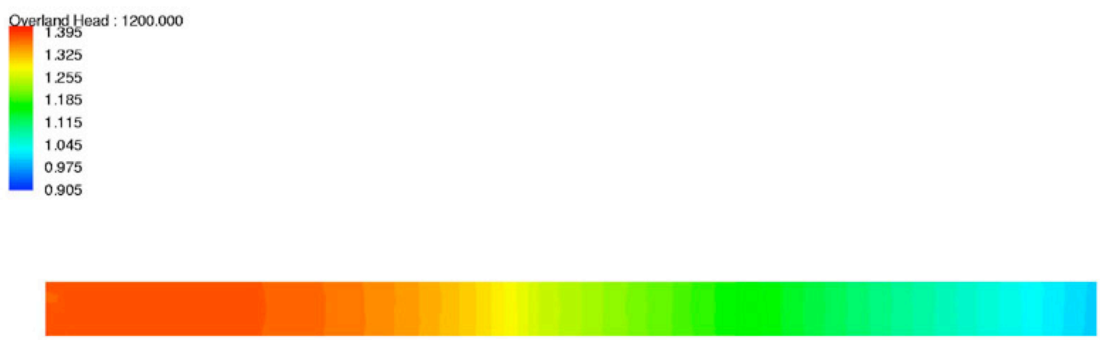

Figure 2.9: Overland Head Results for Rain and Evaporation.

This example was divided by material type and the symmetry of the rain and evaporation can be seen. Typically, rain and evaporation are varied over time for long runs. This is done by simply adding more points in the **XY1** series card that describes the conditions at different timesteps. Remember that rain is assumed positive and evaporation is assumed negative.

Transport Quantities

Concentration

Transport quantities and concentrations can be modeled in AdH with a few additions to the boundary condition and hotstart files. The **TRN** card at the beginning of the boundary condition file gives the number of transport items being modeled. Change this value to one (1) for this example. The attributes for the constituent must then be given for each material type. The values necessary for this example with a reference concentration of one are given below. Also, change the flow conditions back to those given initially so that rain, evaporation, or supercritical flow are not occurring.

```
MP DF    1   1   0.02
MP TRT   1   1   0.1

MP DF    2   1   0.02
MP TRT  2    1   0.1

! Reference Constituent: Constituent 1, Concentration
CN  CON 1  1   0   0   0
```

The hot start file must also include the initial concentrations at the necessary nodes. To apply an initial concentration cloud having a maximum concentration of one near the left boundary, add the following lines to the hotstart file:

```
DATASET
OBJTYPE "mesh2d"
BEGSCL
ND 250
NC 392
NAME icon 1
TS  0  0.00000000e+00
0
0
0
0
0
0
0
0
0
0
0
0
0.25
0
0
0
0.25
0.5
0.25
0
0.25
0.5
1.0
0.5
0.25
0
0.25
0.5
0.25
0
0
0
0.25
0
.
.
.
0
ENDDS
```

There should be 250 entries so that there is one for each node. The same process would be continued for multiple transport quantities, changing the initial values and the concentration number in the "NAME'' field.

Run Pre_ADH and ADH in the same manner as stated previously. The overland velocity, overland head, and error output files will be generated along with a constituent file for each transport quantity. The additional output files will be named in the same form, with the number of the quantity included.

Uniform_con1.dat

Upon viewing the results of the run (Figures 2.10- 2.11), the concentration is initially a tight cloud. As the flow pushes it downstream, the cloud begins to spread out, due in part to numerical approximation. With ADH, however, this error can be minimized by applying the mesh adaption around the concentration.

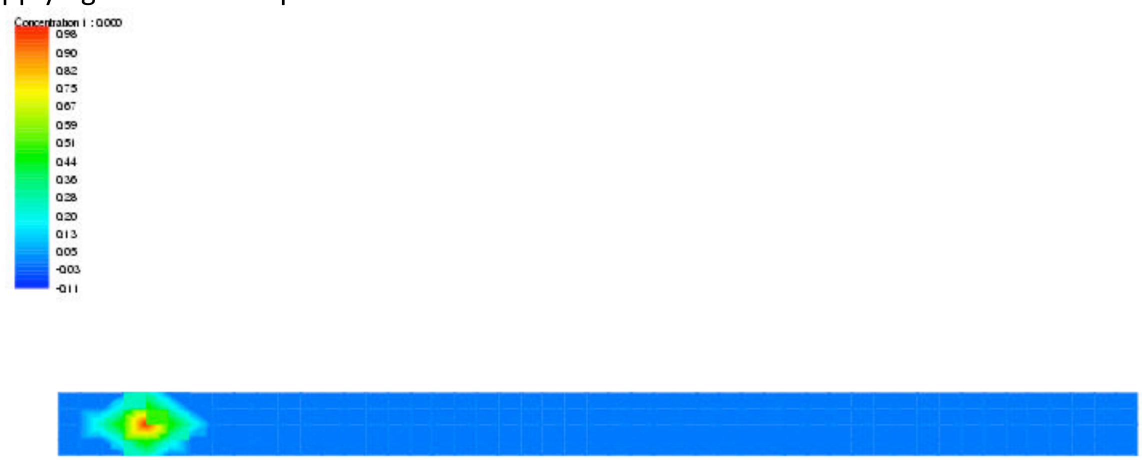

Figure 2.10: Concentration Results at Time = 0 sec.

Figure 2.11: Concentration Results at Time = 4800 sec.

Mesh adaption

The unique aspect of AdH is its ability to adapt the mesh in areas where more resolution is needed and then unrefine when necessary. This process is done by normalizing the results so that an error quantity is determined for each element. If this error exceeds the tolerance set by the user, the element is then split. In order to apply adaption, the refinement level for each material must be greater than zero. This value sets a limit on the number of times an element can be divided.

To invoke adaption around the concentration cloud in this example, change the level of refinement from 0 to 4 for each material. The lines should now look like this:

```
MP ML 1 4
. . .
MP ML 2 4
```

The next step is to set the refinement tolerance for the adaption. This is done on the **TRT** card found in the section describing the concentration. Set this value to 0.1 for this example. In order to ensure that the only adaption occurring is around the concentration, the shallow water refinement tolerance, the **SRT** card, can be set to a large value.

```
MP TRT 1 1 0.1
. . .
MP TRT 2 1 0.1
```

Run Pre_ADH and ADH again with the new boundary condition file. The screen output will show the number of original nodes and the number of new nodes when adaption takes place. If running on multiple processors, the problem will be repartitioned if one processor has 10% more nodes than another. This step allows for better time management by not making one processor work harder than another. The solutions are found at all nodes and then interpolated back to the original nodes, so the user never sees the adapted mesh and the visual results in SMS or GMS may not appear to be improved. However, the results are more accurate due to the smaller elements in the problem areas and the increased number of nodes at which the equations are solved. An example of the adaption taking place in a similar problem can be seen in Figure 2.12.

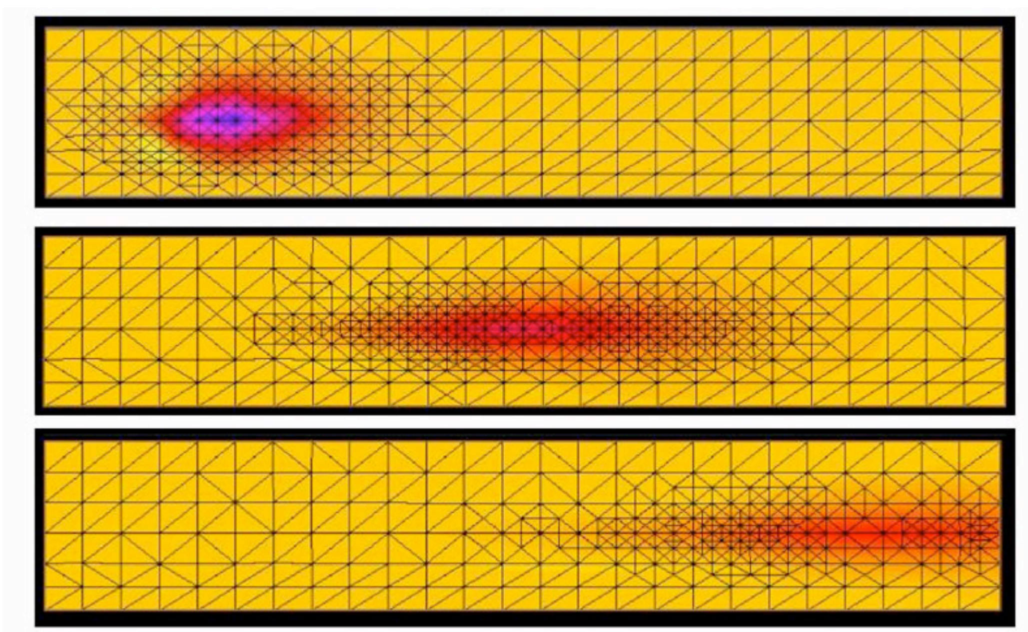

Figure 2.12: Example of Mesh Adapting Around a Concentration Cloud.