TUTORIAL

Dr. Ahmad Fakheri
Department of Mechanical Engineering
Bradley University

Tutorial 1
Laminar Pipe Flow

Objective

This tutorial is intended to explore the basic capabilities of GAMBIT and FLUENT by solving the classical problem of pipr flow. The tutorial is also intended to show how problems can be nondimensionalized and how Fluent can be used to solve the nondimensional form of the governing equations and boundary conditions to obtain more general solutions. This tutorial consists of three parts. Part 1 is the nondimensionalization of the governing equations and the boundary conditions. Part 2 is the mesh generation using GAMBIT. Part 3 is solving the governing equations for the flow field by FLUENT.

Problem Statement

Consider laminar flow between two plates shown in Fig. 1. The fluid approaches the plate with a uniform velocity u_∞. First, you will need to nondimensionalize the governing equations to determine the nondimensional values for the size of the computational domain, the boundary conditions, and the properties that need to be inputted to GAMBIT and FLUENT. Generate a mesh by GAMBIT following the steps provided. Then, solve the problem by FLUENT and present the results.

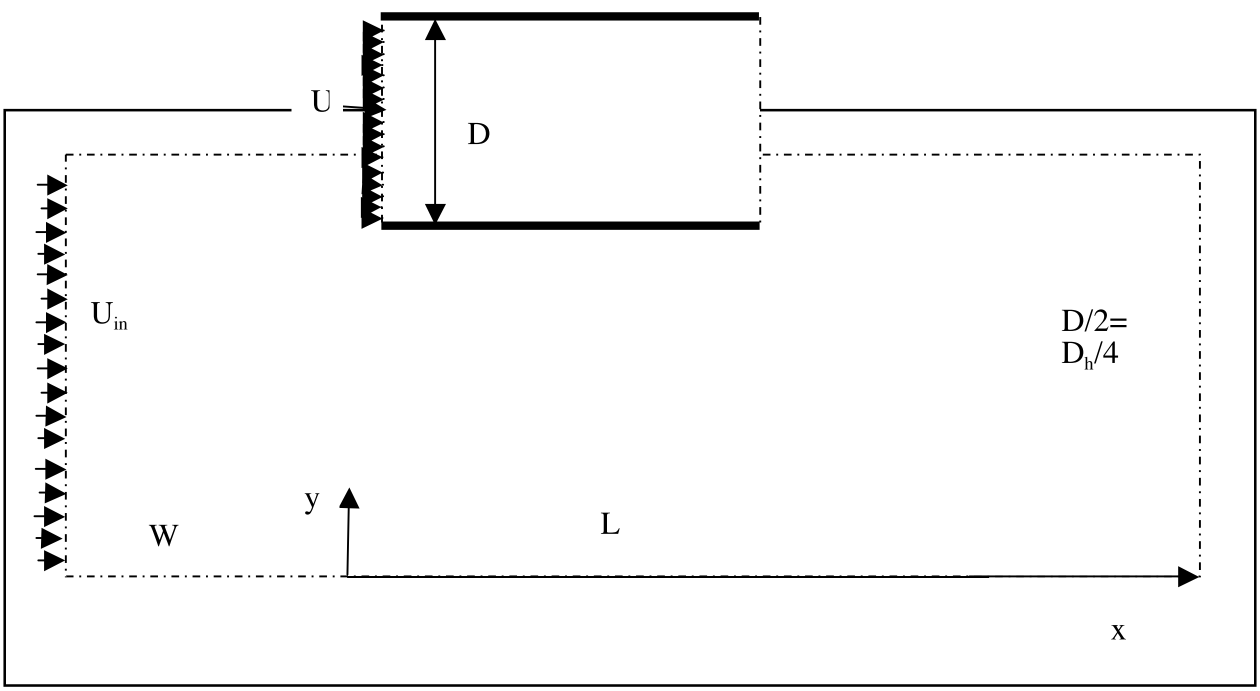


Figure 1 The problem geometry

The governing equations reduce to

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0$$

$$\rho\left(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=-\frac{\partial P}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

$$\rho\left(u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right)=-\frac{\partial P}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)$$

The solution to these equations provide the x and y components of the velocity in the flow field. To obtain general solutions, the governing equations are nondimensionalized first by defining the nondimensional variables

$$u^*=\frac{u}{u_{in}},v^*=\frac{v}{u_{in}},x^*=\frac{x}{D_h},y^*=\frac{y}{D_h}$$

The nondimensional form of the boundary layer equations reduce to

$$\frac{\partial u^*}{\partial x^*}+\frac{\partial v^*}{\partial y^*}=0$$

$$u^*\frac{\partial u^*}{\partial x^*}+v^*\frac{\partial u^*}{\partial y^*}=-\frac{\partial P^*}{\partial x^*}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u^*}{\partial x^{*2}}+\frac{\partial^2 u^*}{\partial y^{*2}}\right)$$

$$u^*\frac{\partial v^*}{\partial x^*}+v^*\frac{\partial v^*}{\partial y^*}=-\frac{\partial P^*}{\partial y^*}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v^*}{\partial x^{*2}}+\frac{\partial^2 v^*}{\partial y^{*2}}\right)$$

$$x^*=-w^* \quad u^*=1 \quad v^*=0$$

$$x^*<0,\quad y^*=0 \quad or \quad y^*=\frac{1}{4} \quad \frac{\partial u^*}{\partial y^*}=0 \quad v^*=0$$

$$x^*>0,\quad y^*=0 \quad u^*=0 \quad v^*=0$$

$$x^*>0,\quad y^*=\frac{1}{4} \quad \frac{\partial u^*}{\partial y^*}=0 \quad v^*=0$$

Comparing this set of equations with the set in the primitive variables that Fluent solves, we can see that by assigning

ρ=1

μ=1/Re

then solution obtained from Fluent will be dimensionless

<u>Part 2: Mesh Generation</u>

Draw the Geometry:

1. Create a directory on the Desktop to keep all your files and name it something like LAMIARPIPEFLOW
2. On PC double click *gambit* and Browse to select LAMIARPIPEFLOW on the Desktop **pipeflowp** and for the session ID enter a name like pipeflow **Run**

3. After Gambit window appears, if need be, increase the size of command panel by clicking on the line located on the left of command panel and then hold and drag it to the left.

4. Because all dimensions are based on the hydraulic diameter, it would be easier to create a scale table first. To do so, you can click on **Tool Command Button** under **Operation**, and then click on **Coordinate-System Command** button under **Tools**. Finally, clicking on **Display Grid** button under **Coordinate System** will open **Display Grid** window in which you can put in maximum, maximum, and increment values for x and y axis (Fig. 2). In this case, for x-axis, the minimum value is -0.5, the maximum value is 2, and the increment value is 0.5 Then click on **Update List** button to save your input. All x-coordinates of every node will appear.

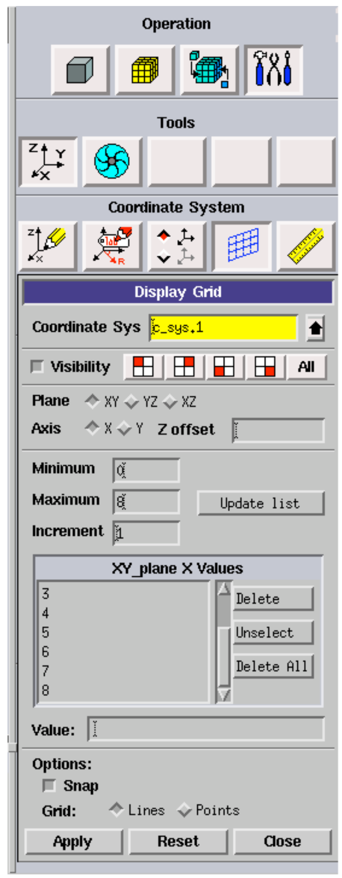


Figure 2 Display Grid panel

For y-axis, click on **Y** on the right of **Axis**. The minimum value is 0, the maximum value is 0.25, and the increment value is 0.25. Click on **Update List**, and then **Apply**, and **Close** then Gambit will create a scale table with the specific size for you.

Note: Name of the button appears under the **Description** window.

You can enlarge the picture by holding Ctrl button and then dragging the left-click to create a box covering the focused area.

5. Save the file by going to **File Save**
6. The next step is to create the actual nodes of the geometry. To do so, you can click on **Geometry Command Button** under **Operation**, and then click on **Vertex Command** button under **Geometry**. Finally, click on **Create Vertex** button under **Vertex** to open **Create Real Vertex** window (Fig. 3).

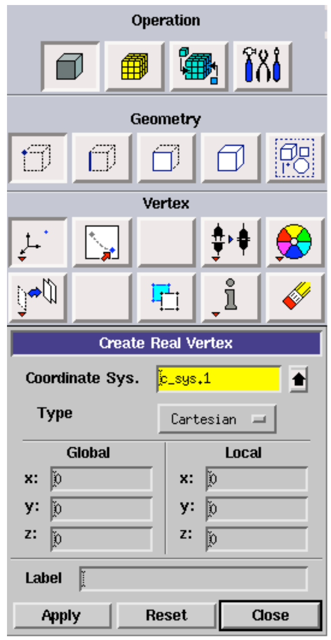


Figure 3. Create Real Vertex panel

To create a node, you can either put coordinate of the node in this window, or Ctrl-right click on the created scale table. In this case, it would be easier if you just Ctrl-right click on the created scale table to create 8 grid points, upstream, inlet, halfway in the pipe and exit along the centerline and wall.

Now, you can turn off your scale table by opening **Display Grid** window again, but this time turn off the square on the left of **visibility** by clicking on it. Then click **Apply**, **Close**

7. At this point, it is wise to zoom in somewhat to see the vertices more clearly. There are two ways to do this. You can zoom in by holding Control and drawing a box by left clicking the mouse. Or, to fit all the geometry into the available screen space (a very handy tool!), in *Graphics/Windows Control* (near the bottom right), Fit to Window.

8. In the main Gambit window near the upper left, File-Save. This will save your work so far. It is a good idea to do this every so often, especially after a major task is completed.

9. Now, you have to connect all vertexes with edges. To do so, you can click on **Geometry Command Button** under **Operation**, and then click on **Edge Command** button under Geometry. Finally, clicking on **Create Edge** button under **Edge** will open **Create Straight Edge** window (Fig. 4).

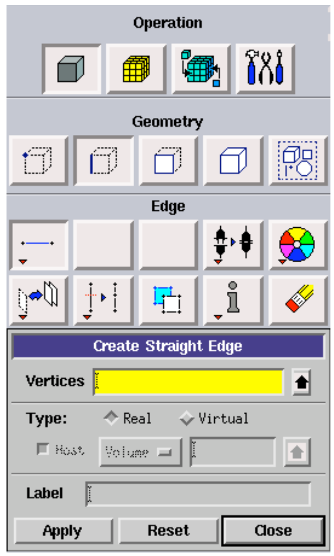


Figure 4 Creating Straight Edge panel

To create an edge, Shift-left click on the starting vertex, and then Shift-left click on the ending vertex, it is also helpful to assign a descriptive name to each edge like wall, inlet, symmetry,.. etc. Finally click on **Apply**.

Alternatively,

a. Under *Geometry*, Edge Command Button. (Note: If the button is already open, clicking it again will make the options disappear! If this happens, click it again.)
b. Under *Edge*, right mouse click the top left icon, which is called Create Edge, and then Create Straight Edge. (In this learning module from now on, this type of command which requires a right click of the mouse will be denoted by R-Create Edge-Create Straight Edge.)
c. In *Create Straight Edge*, make sure the text area called *Vertices* is highlighted - if not, click in that area. Vertices are selected by Shift + left mouse click. Select the two rightmost vertices, first the one we called lower right, and then the one we called upper right.
d. Type in an appropriate label for the edge about to be created, like "outlet" or something equally descriptive. Apply. Gambit will create a straight edge between these two vertices. A yellow line should be drawn on the screen connecting the vertices.

e. In similar fashion, create two separate edges along the top, one along the top left of the computational domain, and one along the centerline of the pipe. If desired, label these "top left", "top center" respectively.
f. Generate an edge along the left side of the domain, and call it "inlet", and another at he inlet of the pipe and call it “pipe inlet”
g. For the bottom of the domain, create two separate edges - one from the lower left vertex to the origin (inlet of the pipe), and one from the origin to the lower right vertex. Recommended labels are "bottom left" and "wall".
h. Now Close the *Create Straight Edge* window.
i. In the main Gambit window near the upper left, File-Save. This will save your work so far. It is a good idea to do this every so often, especially after a major task is completed.

Note: You cannot choose one vertex twice. If you need to, click **Apply** first, and then create another set of lines.

10. Finally, you have to create a face from all edges. To do so, you can click on **Geometry Command Button** under **Operation**, and then click on **Face Command** button under **Geometry**. Finally, clicking on **Form Face** button under **Face** will open **Create Face From Wireframe** window (Fig. 5).

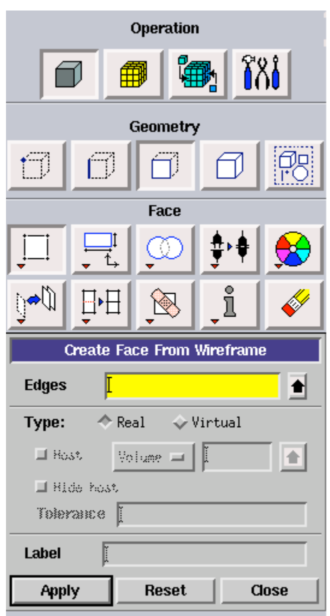


Figure 5 Create Face From Wireframe panel

To create a face, you can Shift-left click on all the edges, and then click on **Apply**. Alternatively, you can drag Shift-left click to create a box covering all the edges. Then all the edges will be selected. Also, you should notice that color of all lines changes from yellow to blue. The color change means that the separate edges are now forming a face.

Note: All vertexes, edges, and faces have their own name, and they still can be considered individually. Also, it is important to select the edges *in order* when creating a face from existing edges. A consistent method is to select them in mathematically positive counterclockwise order. Select the rightmost edge first (using shift+left mouse button), followed by the top right edge, the top plate, top left edge, the leftmost edge, the edge upstream of the plate, then the edge representing the plate itself, followed by the lower right edge. These edges outline a closed face.

Apply the Mesh:

1. After creating the geometry, you can start applying the mesh. Usually, you start with meshing on each edge first, then apply face meshing, and finally volume meshing. To apply edge meshing, you can click on **Mesh Command Button** under **Operation**, and then click on **Edge Command** button under **Mesh**. Finally, clicking on **Mesh Edges** button under **Edge** will open **Mesh Edge** window.
2. Select the leftmost (inlet) edge (using shift+left mouse button), and in *Mesh Edges*, change the *Spacing* option from Interval Size to Interval Count. Enter the number 20 as the *Interval Count*. If it is desirable to cluster or bunch nodes close to the plate, the *Ratio* (in the *Grading* section of the *Mesh Edges* window) can be set to a number different than 1.0. For now just leave it as 1. Note: when using variable spacing, if the bunching is backwards from the way you desire, click on the Reverse button. Apply. Blue circles should appear at each created node point along that edge. Put the same node distribution on the rightmost edge (the outlet). You may have to Reverse the distribution. Don't forget to Apply; otherwise the node distribution is not saved.
3. Do the same at the outlet
4. Select the left portion of the bottom edge (the part upstream of the plate). Change *Interval Count* to 20. Keep *Ratio* at 1.0. **Apply**.
5. Similarly, define 100 uniformly spaced nodes along the pipe itself
6. Define the same distribution on the two top edges as on the two bottom edges so that the mesh lines will be vertical. This is not necessary, but makes the mesh look nicer, and rectangular cells lead to better convergence. Don't forget to Apply each time, or the nodes will not actually be saved.
7. When all edges have been assigned nodes, Close the Mesh Edges window.

Specify the boundary types on all edges:

1. In order for the mesh to be properly transferred to Fluent, the edges must be assigned boundary types, such as wall, inlet, outlet, etc. Actual numerical values, such as inlet velocity, will be specified as boundary conditions from within Fluent itself. In *Operation*, Zones Command Button-Specify Boundary Types Command Button.
2. In the *Specify Boundary Types* window, change *Entity* to Edges (the default is usually Faces). In this problem, which is 2-D, boundary conditions will be applied to edges rather than to faces.

3. Select the left-most edge of the computational domain, which will become an inlet. Change *Type* to **Velocity_inlet**, and type in the name "inlet". **Apply**. Some words indicating this boundary condition will appear on the screen.
4. Select the right-most edge. In similar fashion, make this an Outlet named " Outflow". Be sure to Apply, or the boundary condition will not actually be changed.
5. Select the two top-most edges. These edges will be combined into one boundary for Fluent, which will also be a “Symmetry”. The label "top" is suggested. Apply.
6. Select the horizontal edge running from the lower left to the origin and select its Type as Symmetry, and label it "sym1". **Apply**.
7. Finally, Select the edge defining the plate itself. Name this boundary type "plate", and change *Type* to Wall. **Apply**.
8. Now Close the *Specify Boundary Types* window.

Generate the mesh on the face:

1. Under *Operation*, Mesh Command Button-Face Command Button. The default window that pops up should be *Mesh Faces*. If not, Mesh Faces.
2. Select the face by shift clicking on one of its edges. *Elements* should be Quad by default; if not, change it. Also change *Type* to Map if necessary. The *Spacing* options will be ignored since nodes have already been defined on all edges of this face.
3. Generate the mesh by Apply. If all goes well, a structured mesh should appear similar to the one shown below.

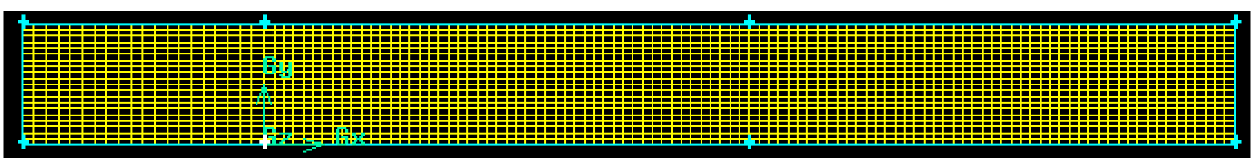


Figure 6 Mesh

4. Zoom out so that the entire mesh can be clearly seen. This is most easily accomplished by clicking on Fit to Window in the *Graphics/Windows Control*(near the bottom right of the screen).
5. You can now Close the *Mesh Faces* window.

Write out the mesh in the format used by Fluent, and then exit Gambit:

1. Click on **File/Save** to save your mesh. Gambit will create three files with extensions .dbs, .jou, and .trn.

2. Click on **File/Export/Mesh** to export your mesh to Fluent by creating ‘msh’ file. Unlike when you save the file, you have to specify the correct path to your folder. Check Export 2-D(X-Y) Mesh box. **Accept**

3. **Note:** Don't forget to delete the 'lok' file before calling back a Gambit's 'dbs' file to make changes.

4. When Gambit exits, see the files created by Gambit. The pipeflow.msh file is the one to be used by Fluent.

Part 3: Solving the Governing Equation by FLUENT

Set the Conditions:

1. Open Fluent and select the **2d Run**.

2. Select File-Read-Case. Find the file pipeflow.msh. Highlight this file (i.e. click on it), and OK. Fluent will read in the grid geometry and mesh that was previously created by Gambit. Some information is displayed on the main screen. If all is read well, it should give no errors, and the word "Done" should appear. **Note - If you have trouble reading your grid, you probably made a mistake in Gambit somewhere. Go back and try to either modify your grid or remake it following the module above.**
3. Check the validity of the grid: Grid-Check. If the grid is valid, no errors should appear. If there are errors, you may have done something wrong in the grid generation, and will have to go back and regenerate the grid.
4. Look at the grid to make sure it is correct. Display-Grid-Display . A new graphical display window opens up showing the grid. If this window is too big, re-scale it by dragging the edges of the window. It is best if the graphical display window is small enough that both it and the *Fluent* window are visible simultaneously. The *Fluent* window and/or the graphical display window may need to be moved to accomplish this.
5. The graphical display can be zoomed-in or zoomed-out with the *middle* mouse button. If you start on the *lower left* and draw a rectangle with the middle mouse button towards the upper right, the display will zoom in on what is included in the rectangle. Meanwhile, the *left* mouse button can be used to drag the image to a new location. If you draw a rectangle *backwards* with the middle mouse button, i.e. from right to the left, it will zoom out. Zoom in if necessary until the grid is shown nicely in the window. Close the *Grid Display* window; the display itself will remain. Note that each boundary condition type should have it's own color (red for pressure, violet inlet, yellow symmetry and white for the plate).
6. Click on **Grid/Scale** to specify the length scale used in the problem. In this case, the length scale is meter. You should check that the maximum x and y values are 2 and 0.25 meters respectively.

   Note: you can use the right mouse click to check which zone number corresponds to each boundary. If you click on one of the boundaries in the graphic window, its zone number, name, and type will be printed in the FLUENT console window.

Generate lines at x= -0.10 , 0.0 , 0.5, 1. and at x = 1.90:

1. The velocity profiles will be examined in detail at five x locations along the plate, namely at x = -0.10 , 0.0 , 0.5, 1. and at x = 1.90, so lines need to be defined within Fluent for these.
2. In the main *Fluent* window, Surface-Line/Rake. Type in the desired starting and ending x and y locations of the vertical line, i.e. a vertical line going from (-0.10,0) to (-0.10,0.25).
3. The *New Surface Name* should be assigned at this point. It is suggested that this line be called "profile-0.10" or something descriptive of its intended purpose.
4. Click on Create to create the line.
5. Similarly, create the other lines at the other x locations using a consistent labeling convention.
6. To view these newly created lines, return to the main *Fluent* window, and Display-Grid-Display. Unselect (by left mouse click) the default interior, and select the newly created lines instead. Display. The lines should be visible at the appropriate locations. If not, create them again more carefully.
7. Now Close both the *Line/Rake Surface* window and the *Grid Display* window.

Define the Model

1. To choose Solver, you can click on **Define/Models/Solver**. In this problem, keep the default values in **Solver** setting (Fig. 7).

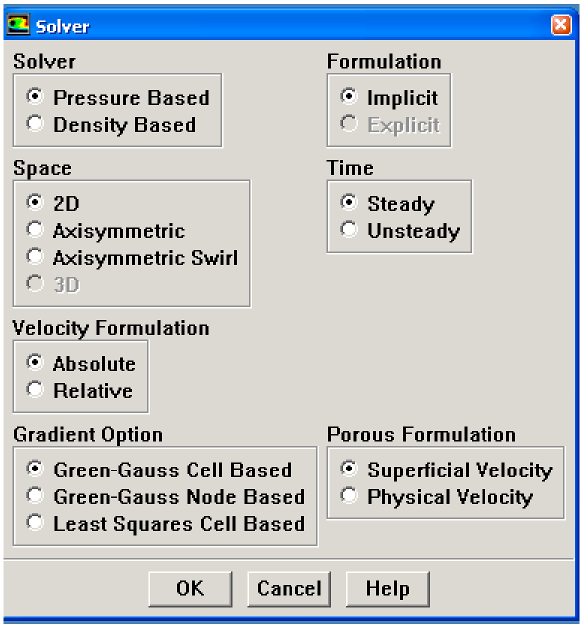


Figure 7 . Solver panel

Define the Fluid

Define Materials using **Define/Materials.** Input the pertinent values in the materials panel. Note that we are solving the nondimensional form of the governing equation, so even though the material is stated to be air, that label is not relevant since we enter the properties manually. We specify density to be 1, and since we are obtaining the solution for Re=100, the viscosity which is equal to 1/Re is set at 1e-2. **Change/Create Close**

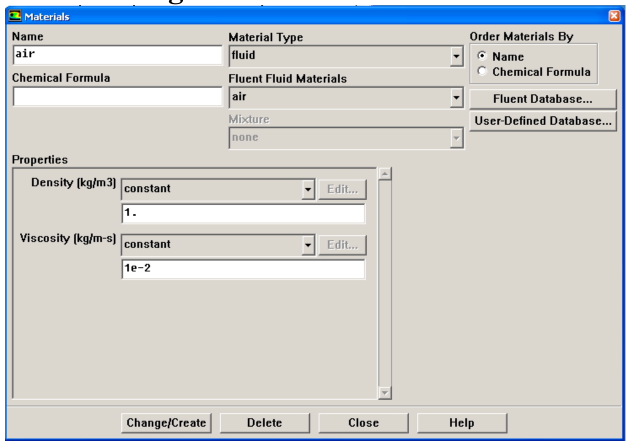
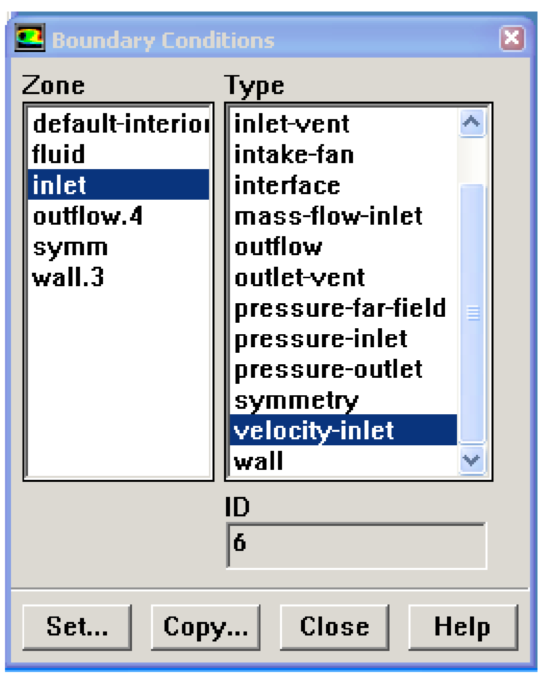


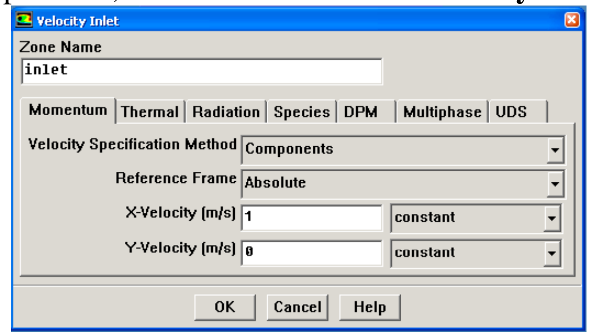
Fig. 8. Material panel

Define Boundary conditions

1. Now the boundary conditions need to be specified. In Gambit, the boundary conditions were declared, i.e. wall, velocity inlet, etc., but actual values for inlet velocity, etc. were never defined. This must be done in Fluent. In the main *Fluent* window, click on Define-Boundary Conditions, and a new *Boundary Conditions* window will pop up.
2. The default boundary conditions for the symmetry planes (symmetry) and the pressure outlets are okay, so nothing needs done to them.
3. First, specify velocity at the inlet by clicking on its name (here we name it inlet). After clicking on the name, its **Type** and **ID** will show up.



4. Click on **Set**, and you will be on **Velocity Inlet** panel. There are three ways to specify the inlet velocity as you can see in **Velocity Specification Method** dropdown list. Choose **Components** for this problem, and use 1 m/s for the **x-velocity**.



5. Check the other boundaries to make sure they are the right type

Fig. 9. Wall Boundary Condition panel

6. Boundary conditions are complete, so Close the *Boundary Conditions* window.

The next step is to choose the **Under Relaxation** factor and the **Discretization** method of each governing equation by clicking on **Solve/Control/Solution** .

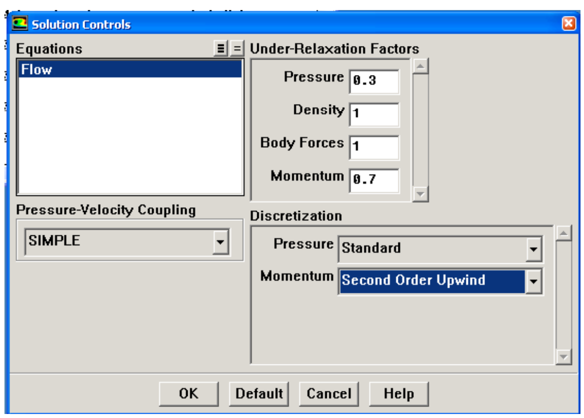


Fig. 11. Solution Controls panel

Under Relaxation factors control convergence of the equations. The higher this number is, the faster convergence, but also the higher chance of divergence. In this case, set the values to half the default values.

The discretization method tells how the partial differential equation is changed into finite difference equation. For this problem, you can use the **Second order upwind** method for momentum, because it has higher order of accuracy than the **First order upwind** method.

7. In order to understand the behavior of the calculation, you will need to turn on the residual graph by clicking on **Solve/Monitor/Residual** (Fig. 12). Then turn on **Plot** under **Options**.

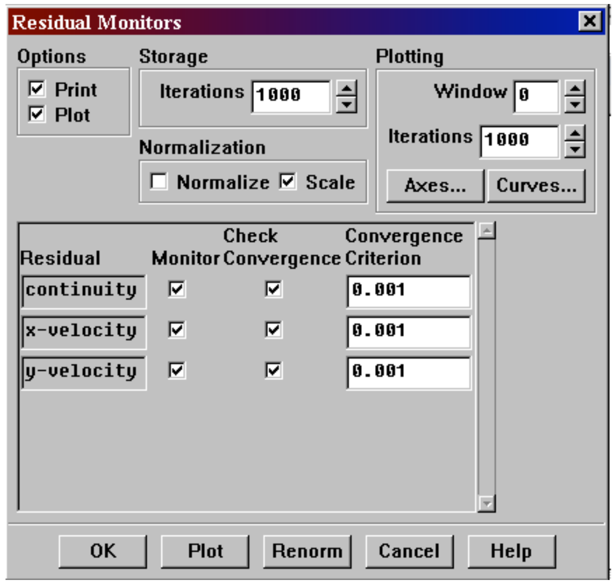


Fig. 12. Residual Monitors panel

Also, in this panel, you can specify **Convergence Criterion** for each governing equation. If you want higher accuracy, you can reduce this number. However, keep in mind that, for many problems, you cannot go beyond a certain value.

8. Initialize the problem by using **Solve/Initialize**

   You select inlet boundary for initialization and initialize x-velocity component to1m/s

9. Now you can start doing the iterations by clicking on **Solve/Iterate** .Do 1000 iterations (267 iterations should be enough).

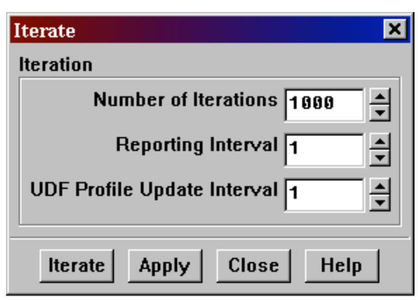


Fig. 13. Iterate panel

10. Save your case and data files by clicking on **File/Write/Case&Data**.

Present the Results:

1. To display a filled contour plot, you can click on **Display/Contours**, then the **Contour**s panel will show up (Fig. 14).

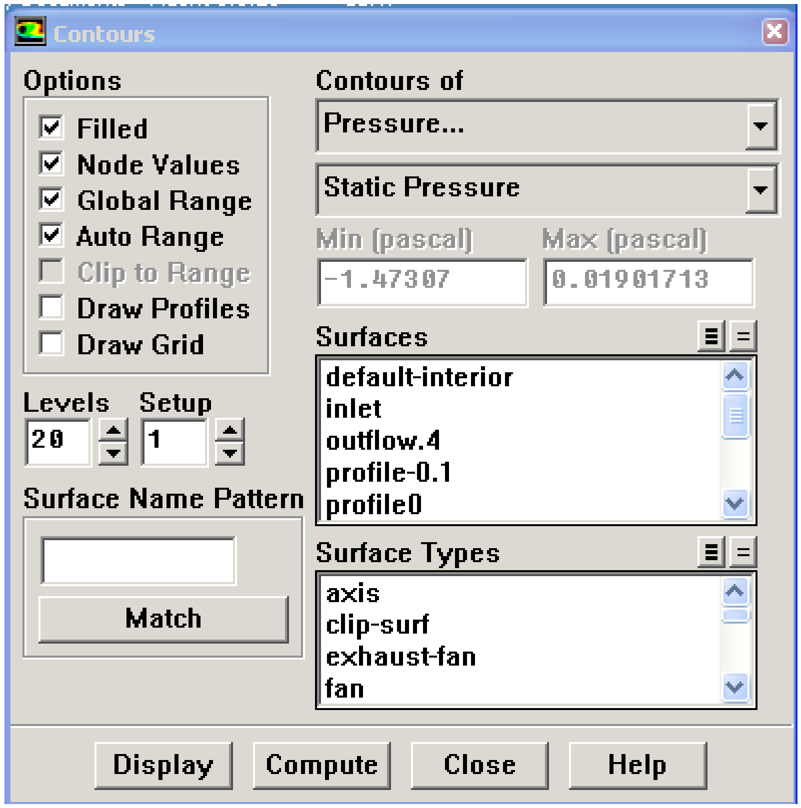
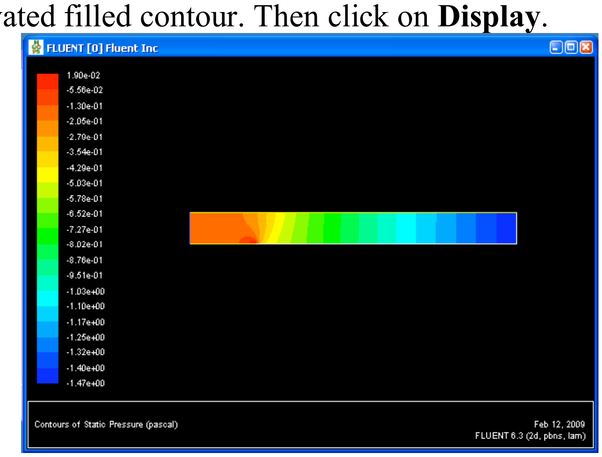


Fig. 14. Contours panel

Pressure and **Static Pressure** are the default values under **Contours Of**. Click on **Filled** under **Options** to activated filled contour. Then click on **Display**.



For contours of streamlines, you can change the first dropdown list under **Contours Of** to **Velocity**, and then change the second dropdown list to **Streamlines**.

Note: you can close up the picture by dragging the middle click from left to right to create a box covering the Tee component. Also, you can zoom out by dragging the middle click from right to left.

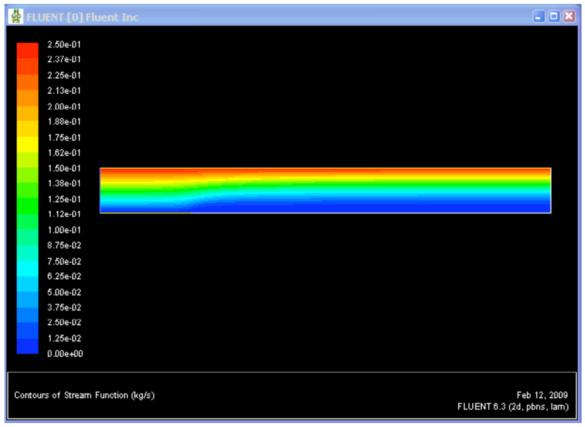


Figure 15 Stream Function

2. To display a velocity vector plot, you can click on **Display/Velocity Vector**, then the **Velocity Vectors** panel will show up (Fig. 15).

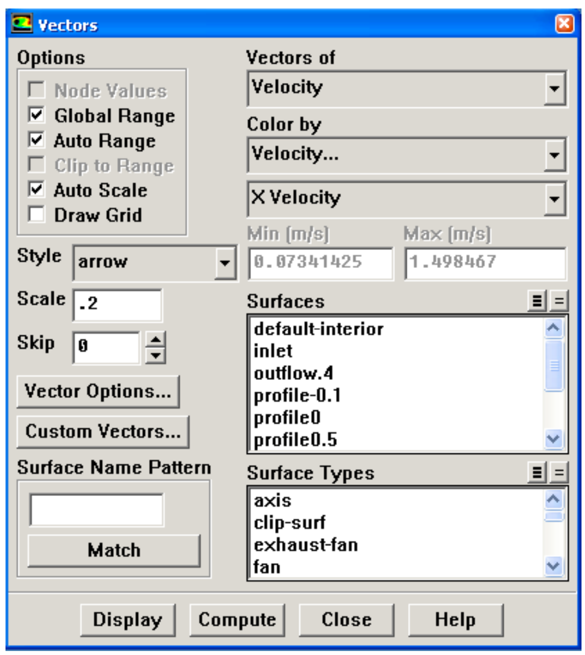
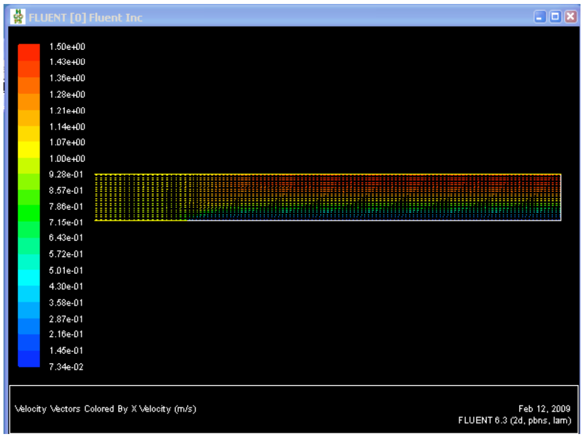


Fig. 16. Velocity Vectors panel

Velocity and **Velocity Magnitude** are the default values under **Color By**. Change **Style** to **Arrow** to see the head of the vector more clearly. **Scale** 2 means the vector is twice as long as its real magnitude. **Skip** 2 means show vector of one cell, and then skip two cells before showing vector of the next cell. Then click on **Display**.



To plot the velocity profiles, go to **Pot xy Plot** and the following window appears

Select the rakes and plot and see the velocity profiles that look like below