

GAMBIT Demo – Tutorial

Wake of a Cylinder.

1.1 Problem Description

The problem to be considered is schematically in fig. 1. We consider flow across a cylinder and look at the wake behind the cylinder.

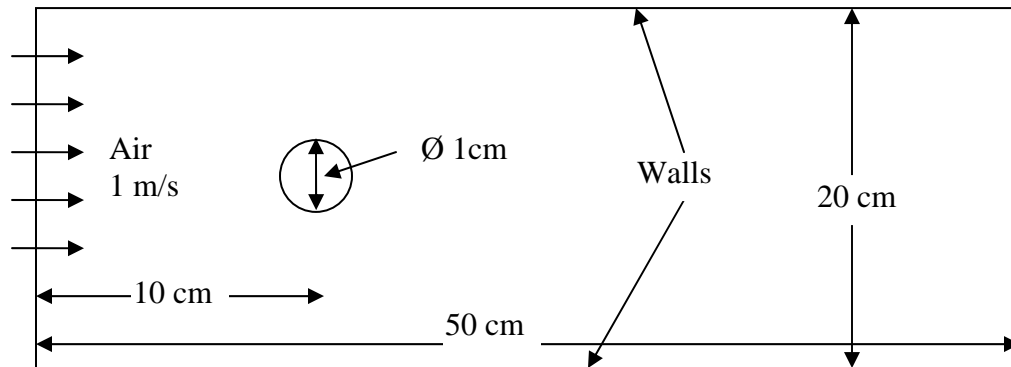


Figure 1: Schematic of the Problem (not to scale).

1.2 Procedure

Start GAMBIT: go to Start→Programs→ Accessories → Command Prompt.

This opens up the DOS command window. Change the path to the directory you want to work in.

At the command prompt, type: `gambit <filename>`

This should open up the gambit interface that would look like in fig 2.

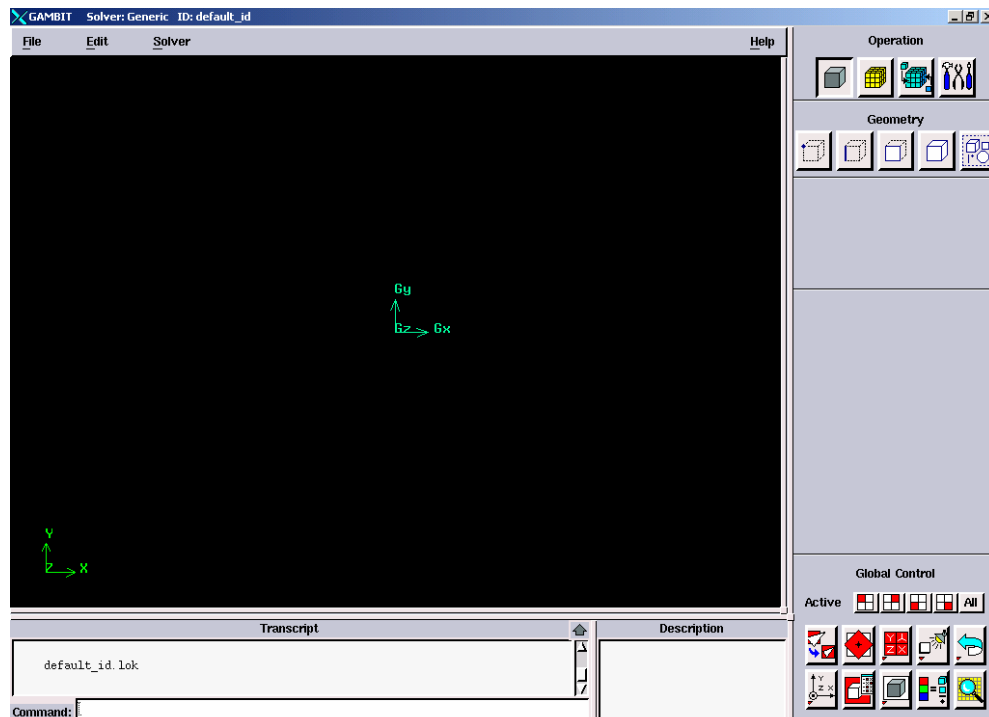


Figure 2: Gambit Interface.

Step 1: Select a Solver


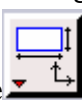
1. Choose the solver you will use to run your CFD calculation by selecting the following from the main menu bar:

Solver —> **FLUENT 5/6**


*This selects the **FLUENT 5/6** solver as the one to be used for the CFD calculation. The choice of a solver dictates the options available in various forms (for example, the boundary types available in the **Specify Boundary Types** form). The solver currently selected is indicated at the top of the GAMBIT GUI.*

Step 2: Create the domain


1. Create the outer domain by drawing a rectangle

Face command button  → **Create face** 


In the create rectangle box, enter the height and width as $w = 50$, $h = 20$ and apply.

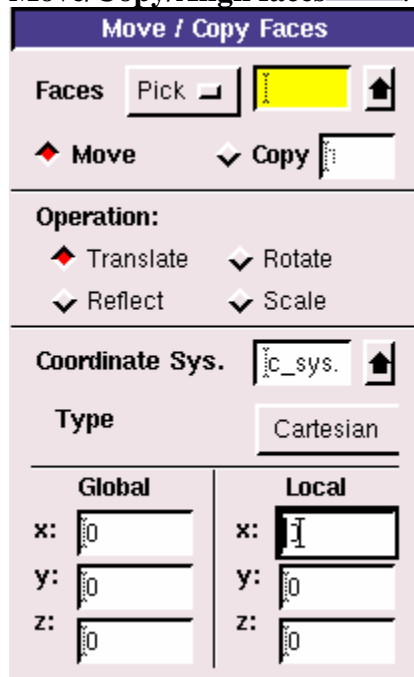
If you do not see the rectangle, fit the display to screen by using  at the bottom.

2. Similarly create a circle for the cylinder.

Right click on  and choose the circle option. In the radius panel enter $r = 0.5$.

3. Offset the circle by 10 units to the left by using the move command.

Move/Copy/Align faces . To select the circle, use shift + left click on the circle.



In the local panel, enter -10 for the x value. This will move the circle by 10 units to the left.

4. Since we only need the region of the domain where the flow occurs, we can subtract the circle from the rectangle.

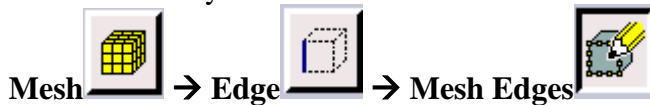


Right click on **Boolean operations** and choose 'subtract' option. This will open up the 'subtract real faces' window. For the first selection, choose the rectangle by using shift + left click on any of the boundaries of the rectangle. Once the rectangle is selected, it will be displayed in red. Now click in the 'subtract faces' box and choose the circle by following the same procedure as before. Click apply.

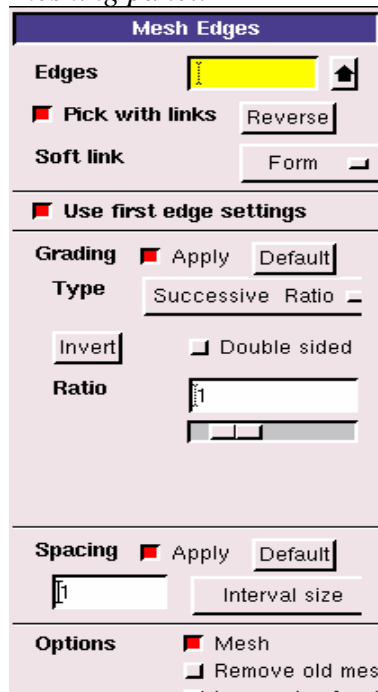
The transcript window will give you a description of the operations performed. Any error will also be displayed here.

Step 3: Meshing the edges.

While a face can be directly meshed in gambit, specifying the node distribution on the edges gives a better control on the grid distribution. This would be especially important when your problem requires a finer mesh in some regions of the domain and not so fine in other regions where there is not much 'action' taking place. In the present example it is clear that most of the interesting physical phenomena take place near the surface of the cylinder and hence we need a fine mesh close to the cylinder.



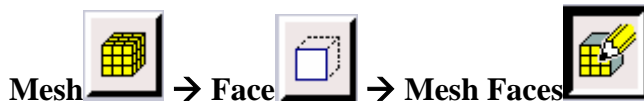
Mesh → Edge → Mesh Edges
This will open up the edge meshing panel.



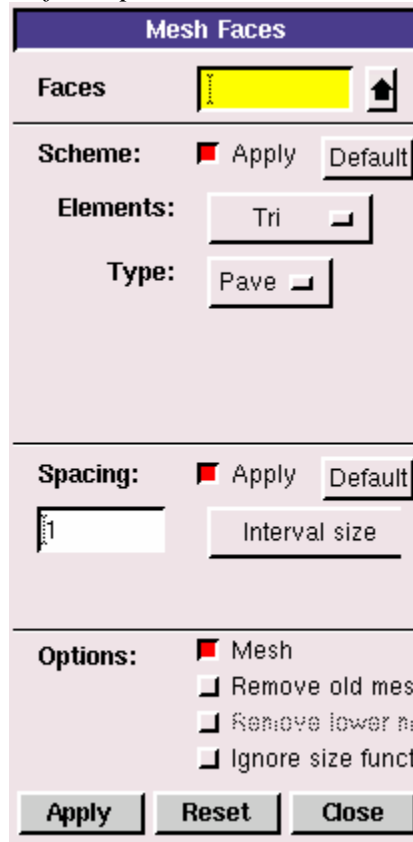
1. Select the edge that makes the cylinder using the same procedure described previously. Leave the value of the ratio at the default value of 1.

Right click on the **interval size** button and choose **interval count** option. This will allow you to specify the number of nodes you want on the cylinder surface. Enter a value of 50 in the corresponding box to the left and apply.

2. Select all the other edges. Right click on the **interval count** button and bring it back to **interval size** and enter 1 in the box. Click apply. This will create an edge mesh for all the edges. Now we are ready to mesh the face.



This will open the mesh faces panel



Select the face of the domain. In the **Elements** option, right click on the **Quad** option and choose **Tri**. The type will be set to **Pave** by default. Click apply. The face will be meshed with triangular mesh elements. The display should look as shown in the following figure. As was desired, we have created a finer mesh close to the cylinder surface and it gets coarser as we move away from the surface.

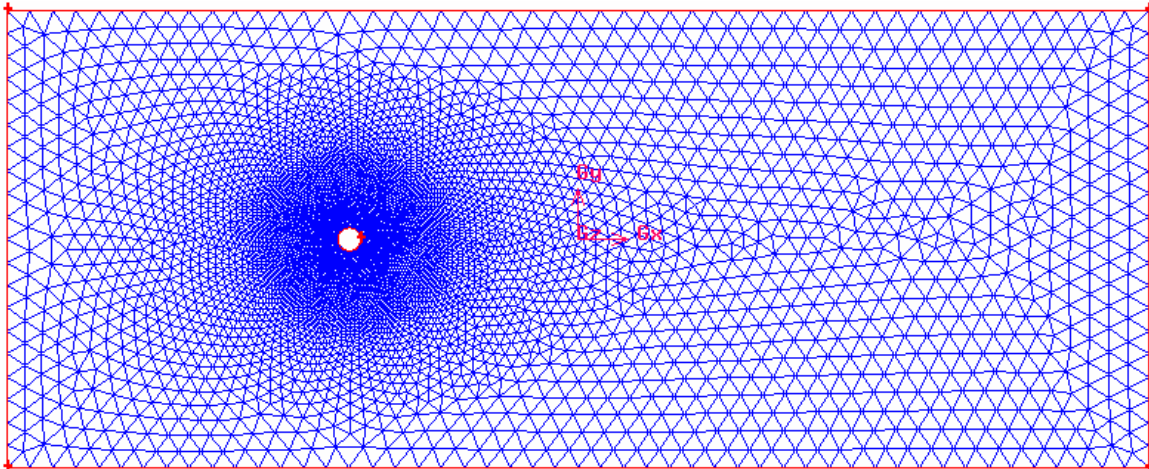


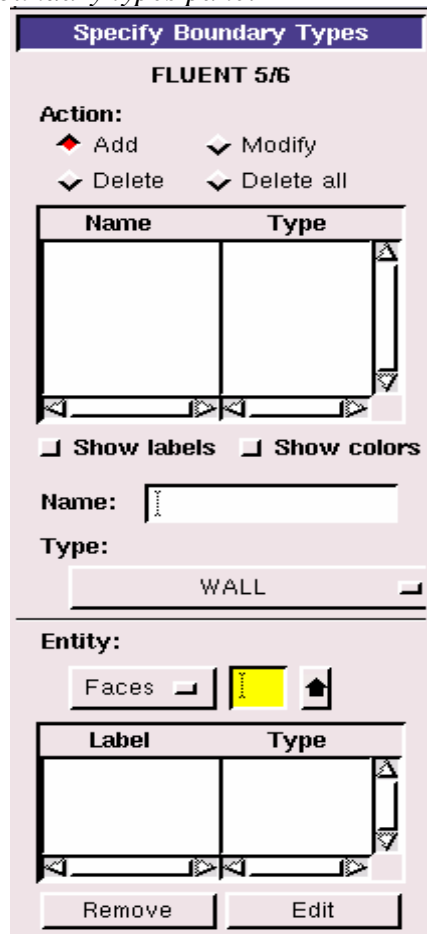


Figure 3: The Meshed Domain.

Step 4: Set Boundary Types.

Zones  → Specify Boundary Types 
This will open the Boundary types panel



1. Under **Entity** change the option from **Faces** to **Edges** by right clicking.
2. Choose the cylinder edge and under **Type** leave it as default 'wall'. In the **Name** box, enter 'cylinder' for the name of the edge (any name can be specified) and apply.
3. Similarly select the top edge of the rectangular domain and name it as 'Top wall'. Name the bottom edge as 'Bottom wall'.
4. Select the left boundary and choose **velocity inlet** from the options under **Type**. Name it as 'inlet' and apply.
5. Select the right boundary and choose **Outflow** from the options under **Type**. Name it as 'outlet' and apply.

Now we are done specifying the boundary types.

Step 5: Exporting the mesh to be read by FLUENT.

File → Export → Mesh

Will open the **Export Mesh File** panel. Here you can specify the path and the filename of the mesh to be exported. Specify 'Cylinder.msh' as the filename. Make sure to click on the **Export 2-D mesh** radio button to make it active. Click apply. The mesh will get exported in a format that FLUENT reads.

File → Save as → Enter path and filename. This will save the .dbs file.

File → Exit

FLUENT Demo – Tutorial

Step 1: Open FLUENT.

Start → Programs → Fluent Inc → Fluent 6.0 Select 2D and run.

Step 2: Read Case file.

File → Read → Case. Go to the directory where you saved your mesh file and open the file with extension .msh.

*As this grid is read by **FLUENT**, messages will appear in the console window reporting the progress of the conversion. After reading the grid file, **FLUENT** will report that 12288 triangular fluid cells have been read, along with a number of boundary faces with different zone identifiers.*

Grid → Check

Grid Check

Domain Extents:

x-coordinate: min (m) = -2.500000e+01, max (m) = 2.500000e+01

y-coordinate: min (m) = -1.000000e+01, max (m) = 1.000000e+01

Volume statistics:

minimum volume (m3): 1.658685e-03

maximum volume (m3): 5.699425e-01

total volume (m3): 9.992167e+02

Face area statistics:

minimum face area (m2): 5.737170e-02

maximum face area (m2): 1.208482e+00

Checking number of nodes per cell.

Checking number of faces per cell.

Checking thread pointers.

Checking number of cells per face.

Checking face cells.

Checking bridge faces.

Checking right-handed cells.

Checking face handedness.

Checking element type consistency.

Checking boundary types:

Checking face pairs.

Checking periodic boundaries.

Checking node count.

Checking nosolve cell count.

Checking nosolve face count.

Checking face children.

Checking cell children.

Checking storage.

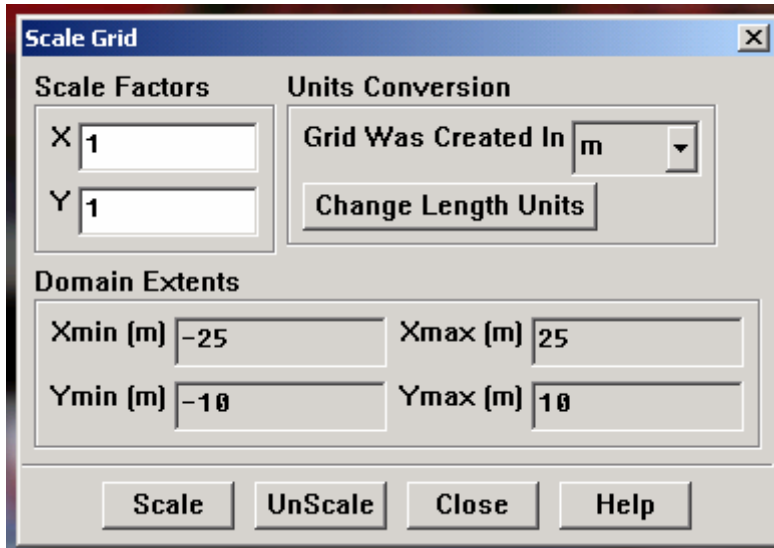
Done.

Note:

The grid check lists the minimum and maximum x and y values from the grid, in the default SI units of meters, and reports on a number of other grid features that are checked. Any errors in the grid would be reported at this time. In particular, you should always make sure that the minimum volume is not negative, since **FLUENT** cannot begin a calculation if this is the case. To scale the grid to the correct units of centimeters, the **Scale Grid** panel will be used.

Step 3: Scale the grid

Grid → Scale



- (a) Under **Units Conversion**, select **cm** from the drop-down list to complete the phrase **Grid Was Created In cm** (centimeters).
- (b) Click **Scale** to scale the grid.
*The reported values of the **Domain Extents** will be reported in the default SI units of meters.*

Note: Do not press scale more than once. If you press by mistake, you can undo this operation by pressing **unscale**.

Step 4: Display the grid

Display → Grid

Make sure that all of the surfaces are selected and click **Display**.

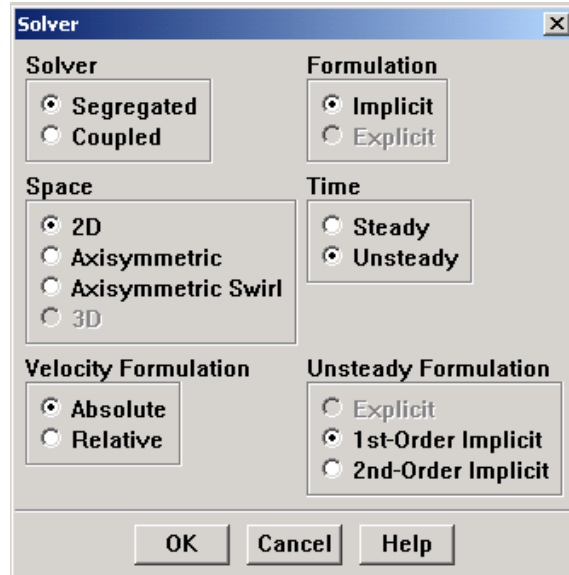
Extra:

*You can use the right mouse button to check which zone number corresponds to each boundary. If you click the right mouse button on one of the boundaries in the graphics window, its zone number, name, and type will be printed in the **FLUENT** console window. This feature is especially useful when you have several zones of the same type and you want to distinguish between them quickly.*

Step 5: Select Solver

Define → Models → Solver

Under **Time** change the option to **Unsteady** since we will be solving an unsteady case.



Define → Models → Viscous...

Leave the option as the default laminar option. In the present problem, the Reynolds number comes out to be about 700 which makes the flow laminar.

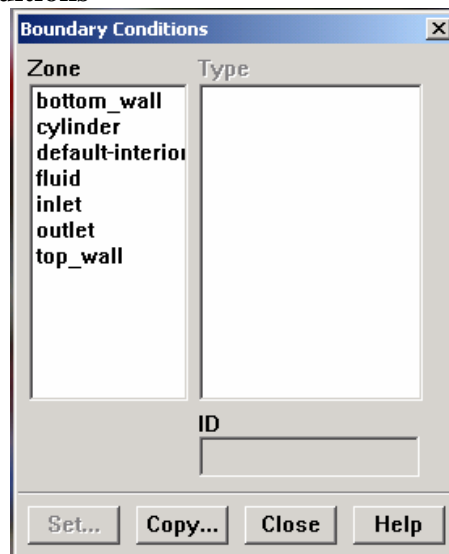
Step 6: Defining Materials

Define → Materials...

Use the default material air. The properties of the material are predefined and can be changed if required.

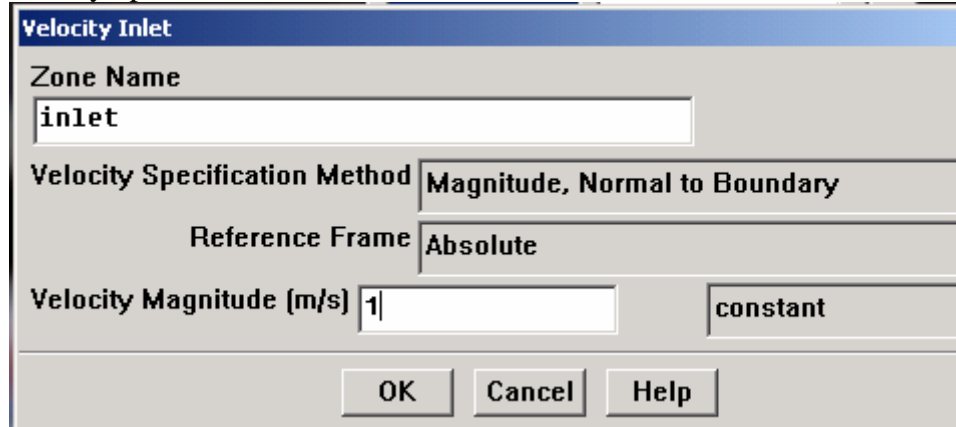
Step 7: Boundary Conditions

Define → Boundary Conditions



Click on the boundary zone and click on set to set the conditions.

- a) select **inlet** from the options and enter a value of 1 m/s for the velocity normal to boundary option.



The image shows a 'Velocity Inlet' dialog box with the following settings:

- Zone Name: inlet
- Velocity Specification Method: Magnitude, Normal to Boundary
- Reference Frame: Absolute
- Velocity Magnitude (m/s): 1
- Velocity Type: constant

Buttons: OK, Cancel, Help

- b) Leave the other boundary zones at their default values. The conditions can be checked (and changed if required) by clicking on each one of the boundaries separately.

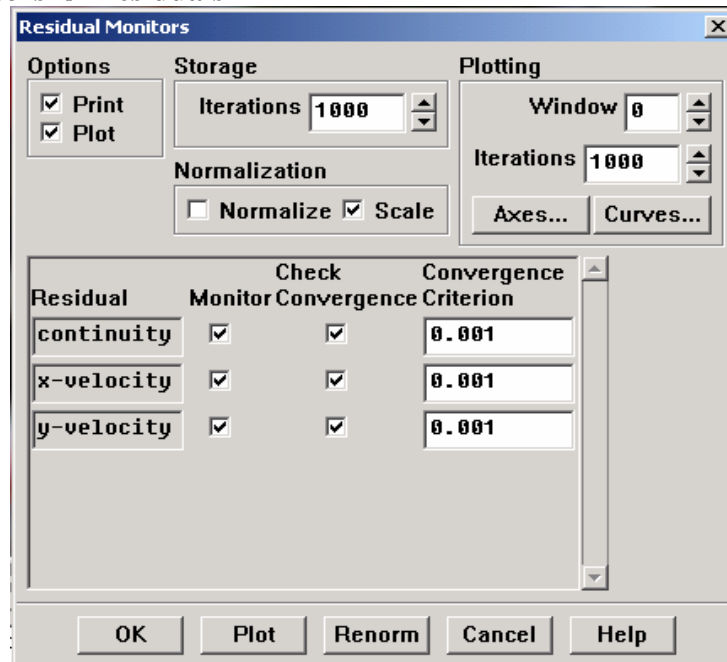
Step 8: Solution

Solve → Initialize → Initialize

- (a) Choose **inlet** from the **Compute From** list.
 (b) Click init and close the panel.

This is just an initial guess for the flow domain. If you know a good guess for the solution, you can enter the values in the panel. This would help faster convergence.

Solve → Monitors → Residuals



The image shows a 'Residual Monitors' dialog box with the following settings:

- Options: Print, Plot
- Storage: Iterations 1000
- Normalization: Normalize, Scale
- Plotting: Window 0, Iterations 1000, Axes..., Curves...

Residual	Check Monitor	Convergence	Convergence Criterion
continuity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.001
x-velocity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.001
y-velocity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.001

Buttons: OK, Plot, Renorm, Cancel, Help

Check the plot option. This will plot the residuals which gives an idea of whether the solution is progressing towards convergence.

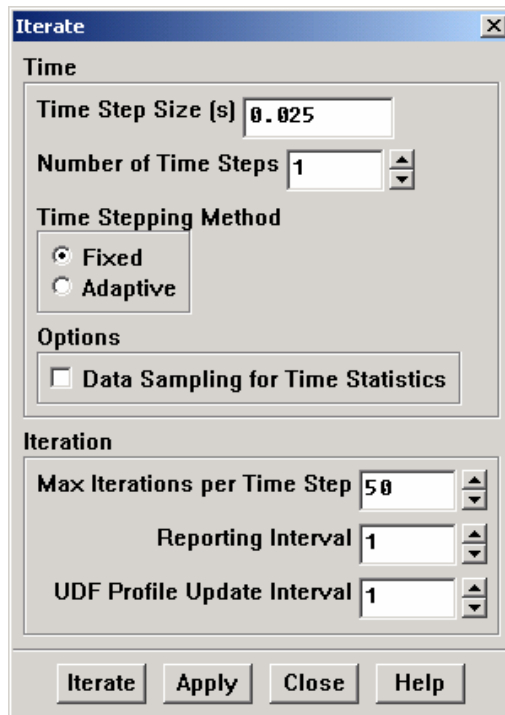
Step 9: Write case

File → Write → Case...

Save the case as cylinder.cas

Step 10: Solve

Solve → Iterate...



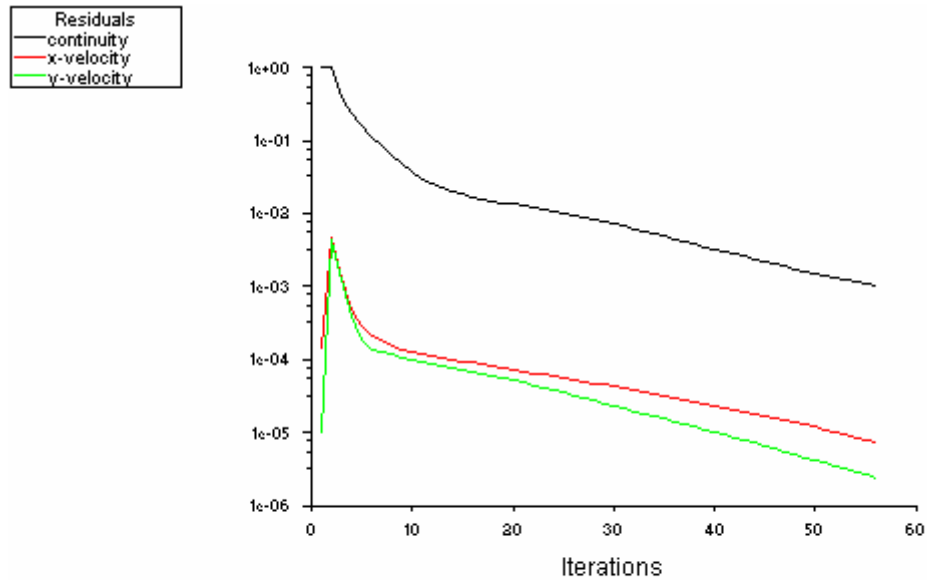
Enter a value of 0.025 for the **Time Step Size** and enter 1 under **Number of time steps**. It is always better to first check the convergence for 1 time step before proceeding with the full solution. This will give you a chance to change say, the time step if required. Increase the **Maximum Iterations per Time Step** to 100. Click Iterate.

Fluent will display the residuals in the display window. It will also print the values of the residuals in the console window. Fluent will display a message

! Solution is converged

at the end of each converged time step.

Note: Make sure that the solution converges at each time step. Otherwise the errors will get accumulated and the solution can be completely inaccurate.

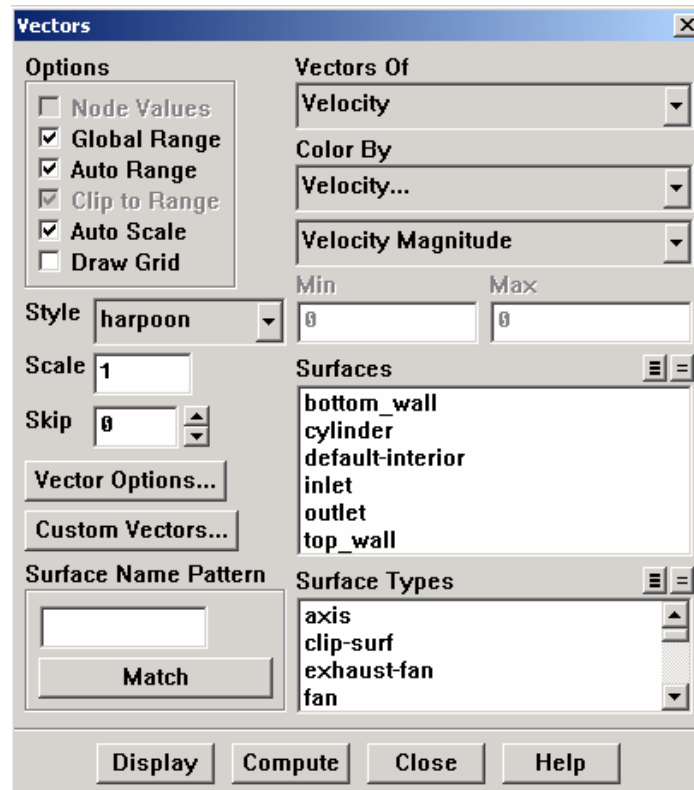


Scaled Residuals (Time=2.5000e-02) Nov 19, 2004
FLUENT 6.0 (2d, segregated, lam, unsteady)

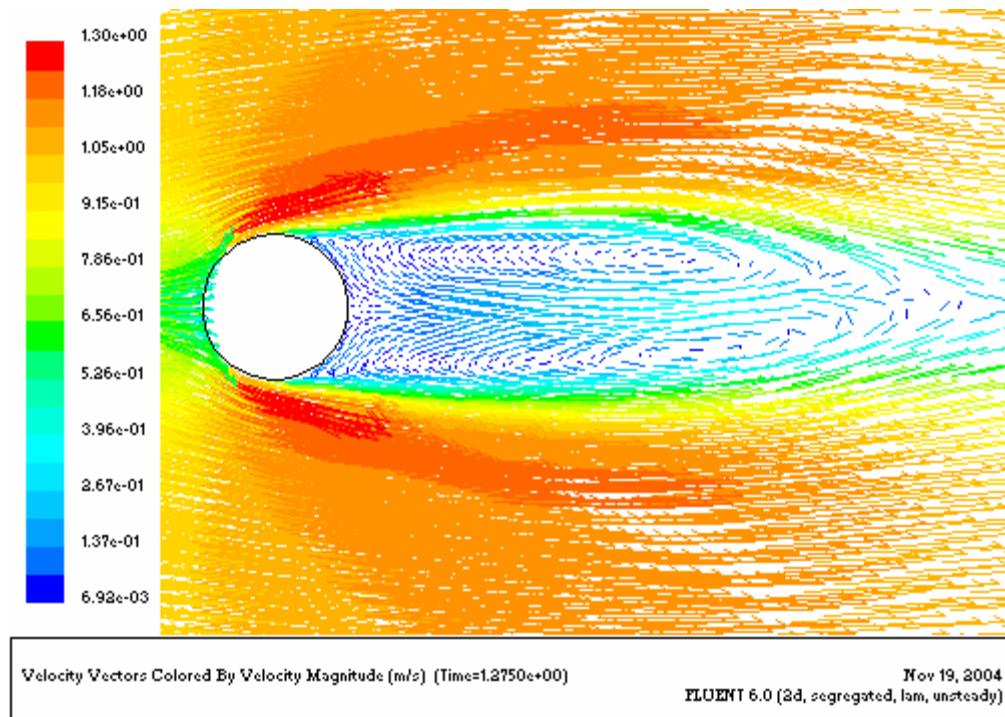
After the first time step converges, increase the number of time steps to 50 and iterate.

Step 11: Postprocessing

Display → Vectors...



Increase the scale to 5 and click display. You can zoom into any part of the display by using the middle mouse button to drag a box. To zoom out, middle click and drag the box from right to left.



To look at contours, **Display** → **Contours...** and select any option from the drop down menu.

Step 12: Save the solution

File → **Write** → **Case & Data...**

Save the case and data with a filename.

File → **Exit**

Validation:

The drag across a cylinder is a well established result and there are various charts available for the same. Referring to one such chart, we get $C_d = 1.131$. Using this to calculate the drag force using $F_D = 0.5\rho u^2 A_p C_d$, we get $F_D = 0.0068$ N/m.

We can get the drag force in Fluent by **Report** → **Forces...** enter 1 for 'x-direction' and 0 for y and z directions. Choose 'cylinder' as the surface from the list and print. Fluent will print the various forces. The total drag force is the one listed under **total force**. Fluent reports a value of $F_D = 0.005$ N/m which is pretty close.