

Gambit – Generation of grid for a cylinder or sphere

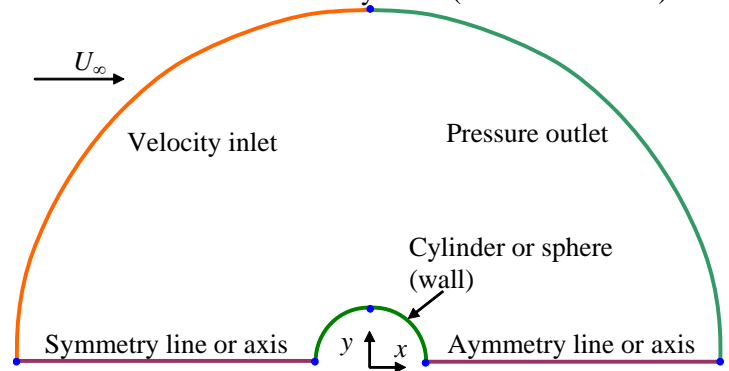
Author: John M. Cimbala, Penn State University
Latest revision: 16 January 2008

Introduction and Instructions:

In this document is a procedure that enables you to generate a two-dimensional, structured grid around a half circle, which will be used with the CFD program, Fluent to solve for flow around a circular cylinder (two-dimensional) or around a sphere (axisymmetric). You should already be familiar with the notation used in this module, as described in a separate learning module.

Overall description of the problem and grid:

1. Because of symmetry, only half of the geometry needs to be generated. The grid will be formed by two half circles, the smaller one representing the cylinder or sphere, and the larger one representing the limits of the computational domain, as sketched (not to scale):
2. In the sketch, x is in the direction of the freestream, and y is normal to the freestream flow. In the case of a 2-D cylinder, the x - y plane represents one slice through an infinite domain in the z -direction (perpendicular to the drawing). For the spherical case, the x -axis is the line of rotational symmetry, and y is actually radius r from the x -axis.
3. In order to adequately resolve the boundary layer along the body surface, grid points will be clustered exponentially near the body. Far away from the body, where the flow is basically uniform, the grid points can be very far apart.



Log on and start Gambit

1. Log onto any Windows or Linux computer that has a valid license for the Ansys-Fluent family of programs – Gambit (grid generator) and Fluent (CFD solver).
2. Start Gambit. *Note:* If Gambit does not open properly, you may not have the proper path set up in your user files. If this is the case, check with the system administrator.
3. In Gambit, when the mouse cursor is placed over a button, a short description of the function of that button appears in the lower right window under *Description*. This is useful in implementing the commands given below.

Create some data points (vertices) for the desired geometry:

1. Create a vertex at the origin, which is the center of the cylinder. In *Geometry*, Vertex Command Button-R-Create Vertex-From Coordinates. In *Create Real Vertex-Global*, enter 0, 0, and 0 for x , y , and z coordinates respectively. Type an appropriate label for this vertex, e.g., “origin”, and Apply. A vertex is created, as indicated by a small white plus sign at this location.
2. The diameter of the cylinder or sphere will be 0.10 m. Vertices need to be created at the front, top, and back of the cylinder. Create another vertex with coordinates (-0.05,0,0). You can call it “front cylinder” or something like that. Similarly, create a vertex at (0.0,0.05,0) called “top cylinder”, and one at (0.05,0,0) called “rear cylinder”. These will be displayed on the screen, but they will hardly be distinguishable from the vertex at the origin since the dimensions are so small.
3. In *Global Control* (bottom right), Fit to Window. This fits the geometry into the available screen space.
4. As shown in the sketch, the outer domain will also be a half circle, in our case 100 times bigger than the small half circle just created. In similar fashion, enter the coordinates of the other points defining the limits of the computational domain, i.e. (-5,0,0), (0,5,0), and (5,0,0). You can name them appropriately (“left domain”, “top domain”, and “right domain” are suggested), or you can let Gambit label these vertices for you by default.
5. Fit to Window again since we have defined the extent of the computational domain.
6. 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.

Create edges from these vertices to define the computational domain:

1. Now some edges will be created to define the cylinder and the outer boundary. 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.)
2. Under *Edge*, right mouse click the top left icon, which is called Create Edge, and then Arc. (From now on, this type of command which requires a right click of the mouse will be denoted by R-Create Edge-Arc.)
3. In *Create Circular Arc*, make sure the text area called *Center* under the *Vertices* section is highlighted - if not, click in that area. Vertices are selected either by Shift + left mouse click on the vertex or from the *Vertex List*, which can be obtained via a click on the up arrow to the right of the highlighted area. Select the vertex at the origin.
4. Now two vertices along the outer domain need to be selected to create an arc. Here we will first create a quarter-arc along the rear of the domain (the edge labeled “pressure outlet” in the sketch). Click in the area called *End-Points*, and select both the right domain and top domain vertices.
5. Type in an appropriate label for the edge about to be created. I suggest “outlet”. Apply. Gambit will create an arc between these two vertices. A yellow arc should be drawn on the screen connecting the vertices.
6. In similar fashion, create another quarter-arc called “inlet” to define the inlet of the domain.
7. Now Close the *Create Real Circular Arc* window.
8. An edge along the bottom left of the domain needs to be generated. R-Create Edge-Straight. Select the “left domain” and “front cylinder” vertices, type in a label (“upstream” is suggested), and Apply. A yellow line should appear indicating successful creation of this edge.
9. Similarly, create a straight edge called “downstream” from “rear cylinder” to “right domain”.
10. Now Close the *Create Straight Edge* window.
11. At this point it is useful to zoom in on the cylinder region: move by dragging the middle mouse and zoom by dragging the right mouse. Two quarter circles also need to be created to define the edges of the small cylinder. R-Create Edge-Arc. Again select the origin vertex as the center of the arc. Select the vertices called “rear cylinder” and “top cylinder” as the end points. Label this arc “rear cylinder”, and Apply.
12. In exactly similar fashion, create an arc called “front cylinder”. At this point you may wish to zoom out to see the entire computational domain.
13. Finally, Close the *Create Real Circular Arc* window.
14. It is a good idea to save your work again just in case something goes wrong. In the main Gambit window near the upper left, File-Save.

Generate a face defining the computational domain:

1. Under *Geometry*, Face Command Button-R-Form Face-Wireframe.
2. It is important to select the edges *in order* when creating a face from existing edges. (I like to select them in mathematically positive counterclockwise order). Select the “downstream” edge first, followed by the “outlet” edge, the “inlet” edge, the “upstream” edge, the edge called “front cylinder”, and then finally the edge called “rear cylinder”. These edges outline a closed face. You can zoom in or out any time as needed.
3. In *Create Face From Wireframe*, type in a label for this face if desired (“domain” is suggested), and Apply. If all went well, a pretty blue outline of the face should appear on the screen; this is a face, which is now ready to be meshed.

Define node points along the edges:

1. In *Operation*, Mesh Command Button-Edge Command Button. *Mesh Edges* should be the default window that opens; if not, Mesh Edges.
2. Select the lower right edge called “upstream”, and in *Mesh Edges*, change the *Spacing* option from Interval Size to Interval Count. Enter the number 80 as the *Interval Count*. It is desirable to cluster or bunch nodes close to the small cylinder. This is accomplished by changing *Ratio* (in the *Grading* section of the *Mesh Edges* window). A ratio of 1.1 is recommended so that there are lots of nodes near the cylinder. *Note*: If the bunching is backwards from the way you desire, click on the Invert button. Apply. Blue circles should appear at each created node point along that edge.
3. If at any time you screw up, hit the Undo button on the bottom right of the screen.
4. Put the same node distribution on the “downstream” edge. You may or may not have to Invert the distribution. Don’t forget to Apply; otherwise the node distribution is not saved.
5. Zoom in so the cylinder is clearly visible. Select the “front cylinder” edge. Set *Interval Count* to 50. This time, we do not need to cluster the nodes, so set *Ratio* to 1.0. Apply.
6. Similarly, define 50 evenly distributed node points along the rear of the cylinder.

7. In a similar fashion, define node distributions along the two large quarter arcs defining the inlet and outlet. Since we are creating a structured mesh, the number of nodes on the large quarter arcs needs to match those on the small quarter arcs. So, generate 50 evenly distributed nodes along each of these large quarter arcs.
8. Now that all the necessary edges have been assigned nodes, Close the *Mesh Edges* window. Save your work.

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 (which may or may not be the default). 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 was labeled “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, called “outlet”. In similar fashion, make this a Pressure Outlet named “outlet”. Be sure to Apply, or the boundary condition will not actually be changed.
5. Select the edge called “upstream” *and* the one called “downstream” (the two straight bottom edges). These can be combined into one boundary. Label it “symmetry” and make its *Type* Symmetry. Apply.
6. You may need to zoom in to do this next step. Select *both* of the quarter-arc edges which define the cylinder. These two edges will be combined into one boundary for Fluent, which will be a wall. The label “wall” is suggested. Make its *Type* Wall, and Apply.
7. Now Close the *Specify Boundary Types* window, and save your work.

Generate the mesh on the face:

1. If you have zoomed in, you will now want to zoom out. Zoom back 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).
2. Under *Operation*, Mesh Command Button-Face Command Button. The default window that pops up should be *Mesh Faces*. If not, Mesh Faces.
3. 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 the appropriate edges of this face.
4. Generate the mesh by Apply. If all goes well, a structured mesh should appear. Zoom in to see how the cells are nicely clustered close to the cylinder wall. This will help Fluent to resolve the boundary layer along the cylinder.
5. You can now Close the *Mesh Faces* window.

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

1. In the main Gambit window, File-Export-Mesh. (The default file name can be changed at this point if desired.) Check the option to export a 2-D mesh, and Accept.
2. When the Transcript (at lower left) informs you that the mesh is done, File-Exit-Yes.
3. The mesh file should now be ready for use by Fluent.