Chapter 3

Creating a curved geomtery in Open Foam

In this chapter we will learn how to create a 2-D geometry for a flow over a cylinder. As it is an axis-symmetric geometry, we will consider the cylinder as a semi-circle and a rectangular domain around it. For meshing this geometry you should divide the domain into small hexahedral blocks. In this particular problem we have body fitted grid, as shown in Fig 3.1 and Fig 3.2:

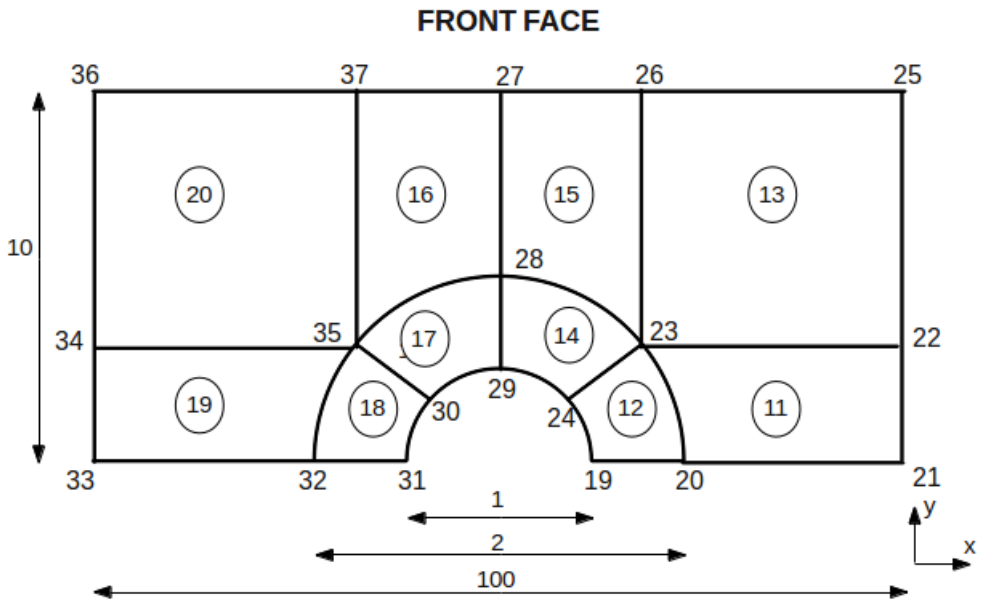

Figure 3.1: Geomtery points in the front face for 2-D flow over a cylinder

Now, as mentioned in the previous chapter create a new blockMeshDict file and open it. In this geometry file copy the initial few lines till “convertToMeters” from previous lid-driven cavity problem “blockMeshDict” file and paste it. Here we will keep “convertToMeters” as 1 as we have given the geometry in meters. After this write down the coordinates of the vertices of the geometry in a similar fashion as given in the previous chapter.

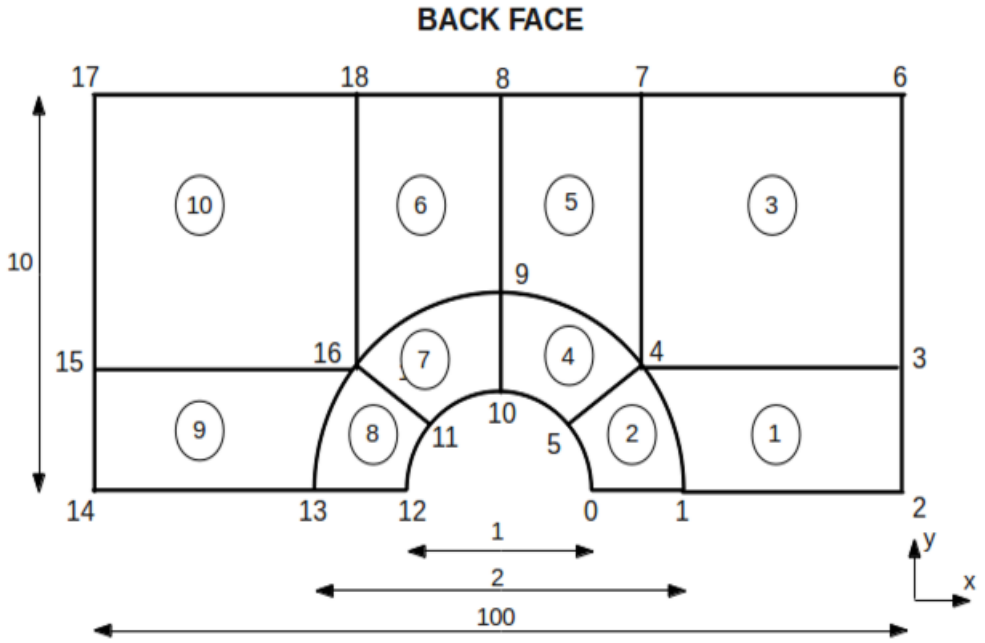

Figure 3.2: Geomtery points in the back face for 2-D flow over a cylinder

For this particular problem we have used cosine and sine function to calculate the vertices on the curved edges as shown below:

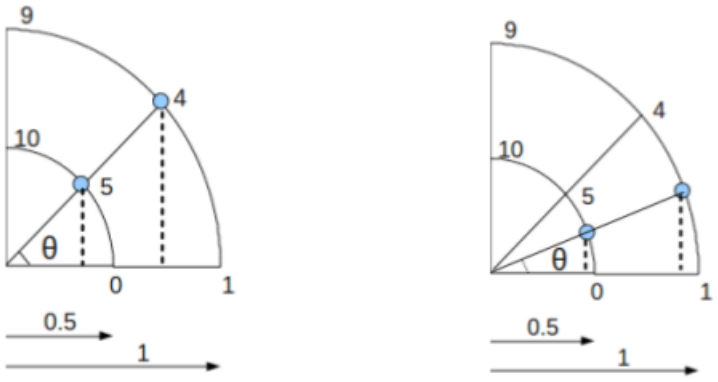

Figure 3.3: Geomtery points for 2-D flow over a curved body

where angle θ is calculated as:

$\sin(\theta)$ = perpendicular/hypotenuse

$\cos(\theta)$ = base/hypotenuse

Note that you should be very careful about the order of the vertex coordinates. It should start from 0 and be continued as 1,2,3.., as shown below:

vertices

(

(0.5 0 0) //0

(1 0 0) //1

.

.

.

.

);

After this enter the details within blocks in a similar manner as shown in the previous chapter. Here in this problem, we have divided the geometry domain into ten small hexahedral blocks having structured body fitted grid points, as shown in fig 3.4. For further details on how to create blocks you may refer to chapter 2.

```
blocks
(
    hex (5 4 9 10 16 15 20 21) (10 10 1) simpleGrading (1 1 1)
    hex (0 1 4 5 11 12 15 16) (10 10 1) simpleGrading (1 1 1)
    hex (1 2 3 4 12 13 14 15) (20 10 1) simpleGrading (1 1 1)
    hex (4 3 6 7 15 14 17 18) (20 20 1) simpleGrading (1 1 1)
    hex (9 4 7 8 20 15 18 19) (10 20 1) simpleGrading (1 1 1)
);
```

Figure 3.4: Block details for the blockMeshDict file

In the previous chapter we have seen, the geometry domain had staright edges, thereby we had kept the details within edges keyword empty (which is the default condition). But here, in this geometry we have curved edges. In OpenFOAM we can use the following types of edges in blockMeshDict file, as shown in fig 3.5:

Keyword selection	Description	Additional entries
arc	Circular arc	Single interpolation point
simpleSpline	Spline curve	List of interpolation points
polyLine	Set of lines	List of interpolation points
polySpline	Set of splines	List of interpolation points
line	Straight line	—

Figure 3.5: Types of edges used in BlockMeshDict dictionary

For this problem we have used arc edges. Therfore the details within edges can be given as:

edges

(

<keyword> <vertices joining the edge> (interpolation points)

);

The details of edges for our current problem is given in fig 3.6:

```
edges
(
    arc 0 5 (0.469846 0.17101 0)
    arc 5 10 (0.17101 0.469846 0)
    arc 1 4 (0.939693 0.34202 0)
    arc 4 9 (0.34202 0.939693 0)
    arc 11 16 (0.469846 0.17101 0.5)
    arc 16 21 (0.17101 0.469846 0.5)
    arc 12 15 (0.939693 0.34202 0.5)
    arc 15 20 (0.34202 0.939693 0.5)
);
```

Figure 3.6: Edge details for the blockMeshDict file

After this enter the boundary patches under the keyword boundary. You may refer to the previous chapter to get know more details regarding how to write the boundary patches.

Similar to the lid-driven cavity problem , even this geometry does not have any patches to be merged. Therefore we will keep the mergePatchPairs empty.

After completing the blockMeshDict file, you can save it within the required case file in polymesh folder inside constant folder. Thereafter switch back to the command terminal and open the required case file as mentioned in the previous chapter.

After this enter blockMesh in the command terminal and press $< enter >$. Thus you can see geometry is meshed. After this type paraFoam in the command terminal and press<enter>. This opens the ParaView window. Now on the ParaView window press apply on the left hand side of the Object Inspector Menu to view the new geometry.

Now in the ParaView window you can check or uncheck the different regions within the mesh region in the Object Inspector Menu to visualize differnt regions on the geomtery. You can also visualize the geometry in wire-frame instead of surface by changing it from the down-down Active Variable Control Menu.

Thus in this chapter we learned how to create a curved geometry in OpenFOAM and visualize it in different ways using ParaView.

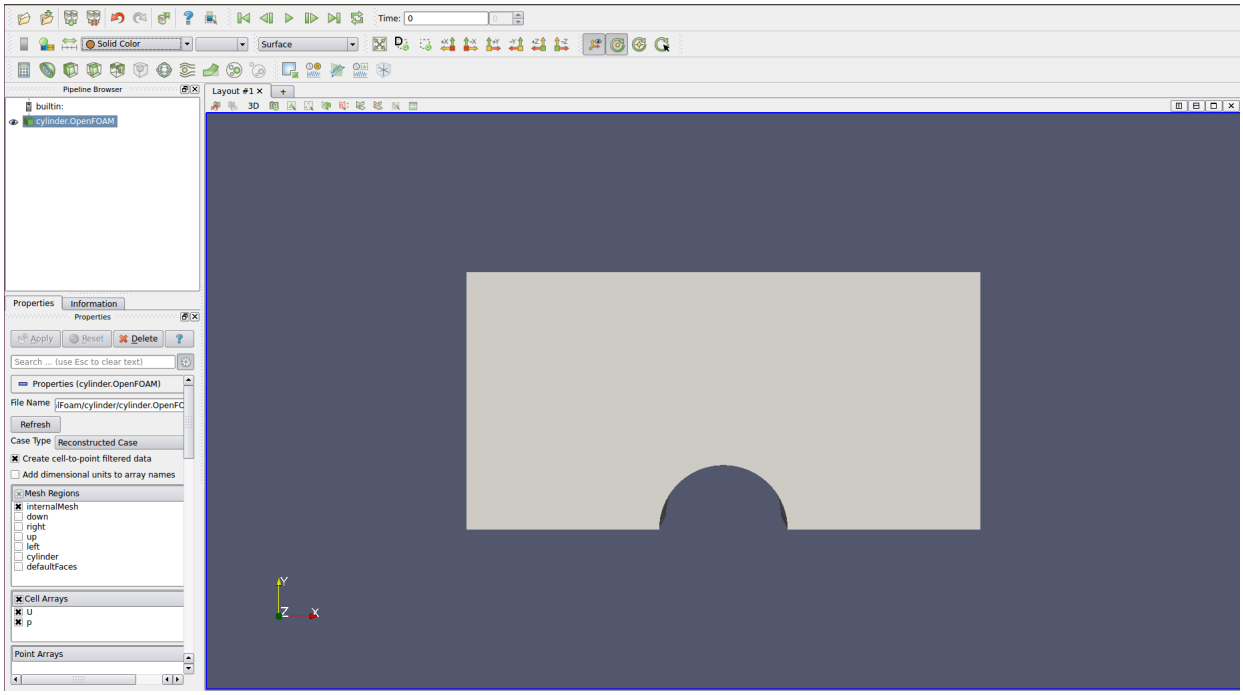

Figure 3.7: ParaView window showing the 2-D geometry