How to build a high-quality mesh using blockMesh in OpenFOAM

By: Ahmed Hamada

September 22, 2022
Source : http://ocean-cfd.engr.tamu.edu/doc/foilmesh

Attachment : blockMeshDict

Report No.: **OCEN CFD Technical Report #220004**

---

1 Introduction and problem statement

In this article, we will discuss a schematic methodology that is recommended to be followed in designing a C-section structured body-fitted high-orthogonal mesh with quadratic elements. This is implemented using the *blockMesh* technique in OpenFOAM. The study case, that will be used to explain the procedure for required mesh, is mirrored cambered foils at a high angle of attack, as shown in figure 1. The foil has a chord, C, separated with the mirrored one with a distance, L, and each one of them is tilted from the free-stream velocity, U, with angle of attach, AOA.

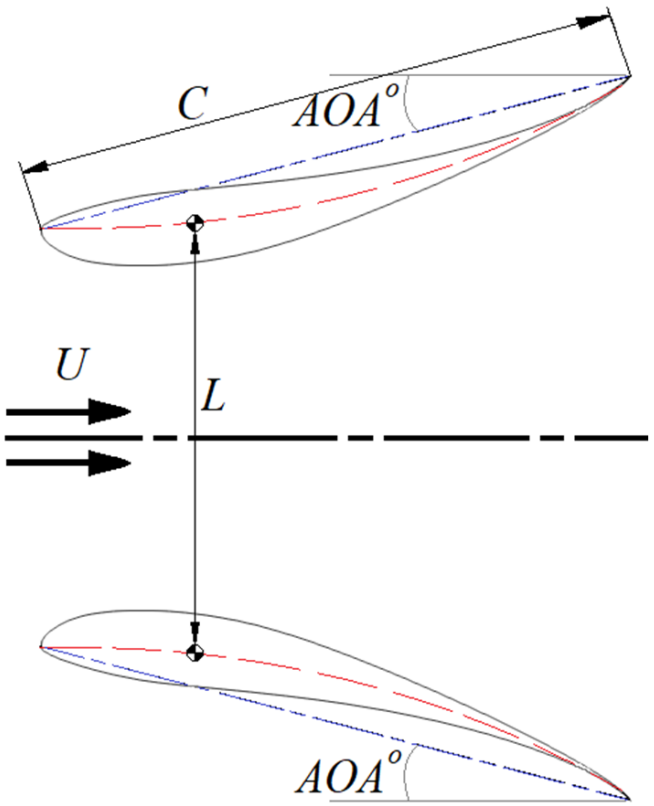

Figure 1: Schematic of mirrored foils.

2 Procedure (DO NOT RUSH ANY STEP)

Part I: Preliminary design of the mesh, by doing a simple sketch.

1. Define the main vertices that defines the boundary surfaces of the computational domain.

   - Add at least 3 points (red ones in figure 2) at any quarter circle.
   - Add at least 6 points (blue ones in figure 2) at the foil body.
   - Add points (green ones in figure 2) along any symmetric lines.
   - Add points (purple ones in figure 2) that bound the bodies on the domain edges.
   - Add points (brown ones in figure 2) on long straight lines.
   - Add points (Grey ones in figure 2) on corners.

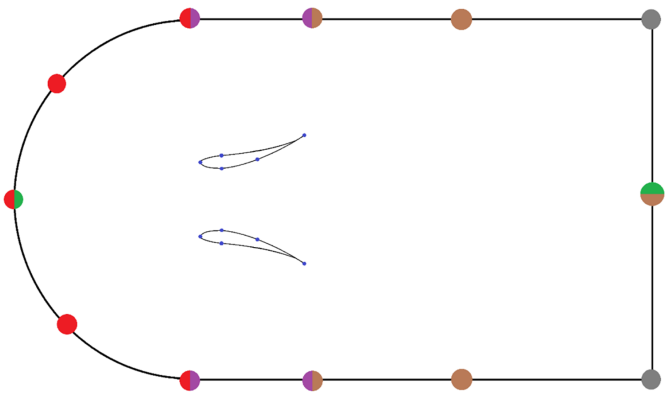

Figure 2: The computational domain including the main points with colors to define each one.

2. Define the secondary vertices that defines the interior blocks of the computational domain. Plot straight lines to design blocks with 4 edges, defining new vertices. These vertices are defined when two plots or more intersect in a point. The following design has 35 points.

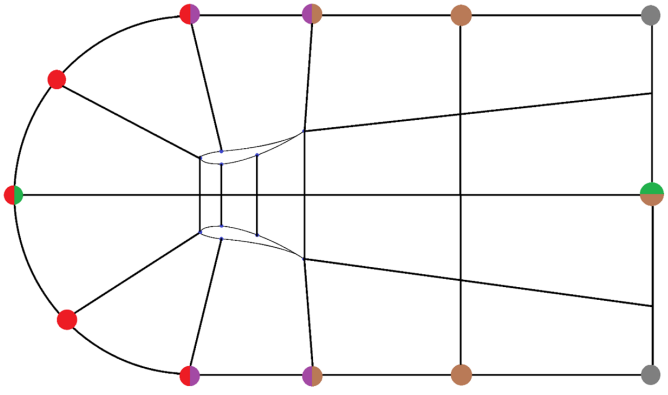

Figure 3: The computational domain including the interior blocks.

Part II: Preliminary building of the mesh using *blockMesh*.

1. Add the vortices to the blockMeshDict with the help of the sketch that you made before. Assuming the x-direction is along the horizontal flow direction and the y-direction is the transverse direction. Then, the z-direction is the outward direction along the span of wing.

- Starting with the negative z-plane, give a name to the vertices in the following format, name nv# $(x, y, -z)$, where n is an abbreviation of negative z plane and v is an abbreviation of vertex. Fill the vertices from the bottom left corner point and moves to the right until the row is finished. Then move up to the higher row starting from the left side again. Further, you may add spaces to align the start of each component, as this would help in the review process if there were mistakes. Remember the numbering in OpenFOAM starts with 0.
- After finishing the vertices of the negative z-plane, copy and past them. Then, start to remove the negative sign in the z-plane to make the positive one and change the naming from nv# to pv#.

Code 1: Vertices in blockMeshDict to generate a grid for the c-section domain

```
vertices
(
//-Z direction

    name vn0  (-2.15         0           -0.1)//Vn0
    name vn1  (-1.225        0           -0.1)//Vn1
    name vn2  (-0.1          0           -0.1)//Vn2
    name vn3  ( 0.0004       0.1735      -0.1)//Vn3
    name vn4  ( 0.0004      -0.1735      -0.1)//Vn4
    name vn5  ( 0.0946      -0.2063      -0.1)//Vn5
    .
    .
//+Z direction
    name vp0  (-2.15         0            0.1)//Vp0
    name vp1  (-1.225        0            0.1)//Vp1
    name vp2  (-0.1          0            0.1)//Vp2
    name vp3  ( 0.0004       0.1735       0.1)//Vp3
    name vp4  ( 0.0004      -0.1735       0.1)//Vp4
    name vp5  ( 0.0946      -0.2063       0.1)//Vp5
    .
    .
);
```

2. Construct the blocks with the help of the sketch that you made before. Start with the bottom left corner one and move to the right finishing the row. Then, move to the second row starting from the left side again. Start building the block from the same relatively vertices and moves along the local x-direction of the block then the y-direction, following the main picture of blocking. Further, you may add spaces to align the start of each vertex, as this would help in the review process if there were mistakes. In this level, use a single division in all directions and a simple grading of 1. In addition, you may add a comment of the number of the block, as this would help in the review process if there were mistakes.

Code 2: Blocks in blockMeshDict to generate a grid for the c-section domain

```
blocks
(
    hex (vn0  vn1  vn17 vn16 vp0  vp1  vp17 vp16) (35  70 5)
    simpleGrading (1 1 1)//B1
    hex (vn35 vn36 vn1  vn0  vp35 vp36 vp1  vp0 ) (35  70 5)
    simpleGrading (1 1 1)//B2
    hex (vn1  vn2  vn3  vn17 vp1  vp2  vp3  vp17) (170 70 5)
    simpleGrading (1 1 1)//B3
    .
    .
);
```

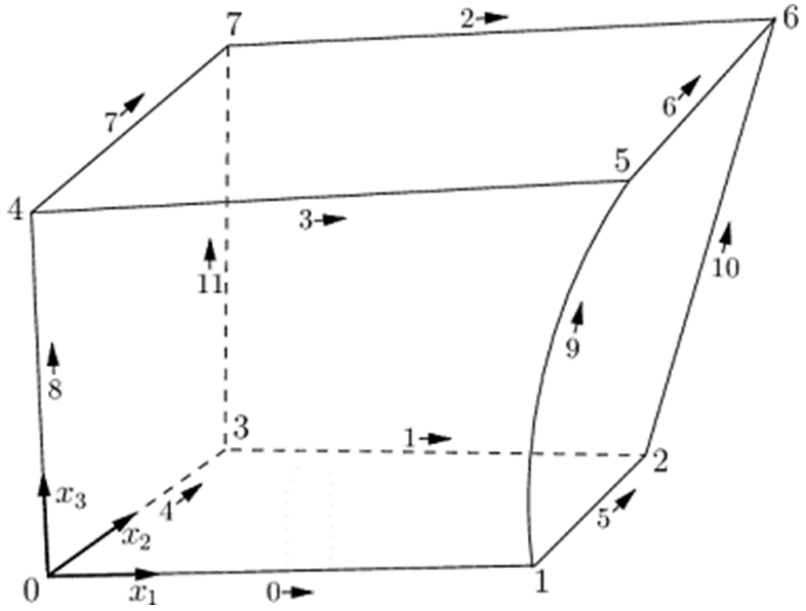

Figure 4: A single block, showing the vertices, and the edges.

3. Leave the edges blank, as there is no need to stress about the arcs or the curved edges in the domain in this step of building the mesh. All edges will be straight lines, and this would help in the review process if there were mistakes.

Code 3: Edges in blockMeshDict to generate a grid for the c-section domain

```
edges
(
);
```

4. Fill faces of the domain's boundaries with the help of the sketch that you made before. Further, you may add spaces to align the start of each vertex, as this would help in the review process if there were mistakes.

Code 4: Boundary in blockMeshDict to generate a grid for the c-section domain

```
boundary
(
     Inlet
    {
        type patch;
        faces
        (
            (vn16 vp16 vp31 vn31)
            (vn0  vp0  vp16 vn16)
            (vn35 vp35 vp0  vn0 )
            (vn50 vp50 vp35 vn35)
        );
    }
    .
    .
);
```

5. Run the command *blockMesh* on the console window and open the Paraview of the case to see the wireframe of the domain.

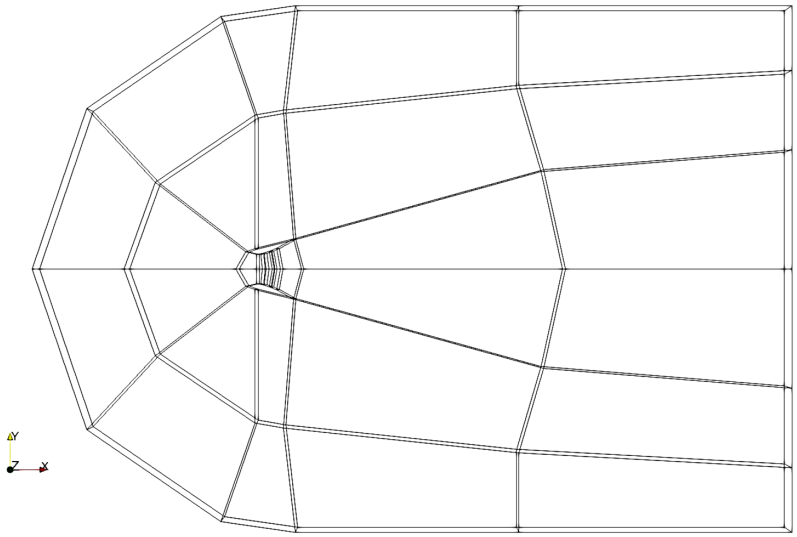

Figure 5: Testing #1 the grid of the C-section domain.

6. Increase the number of divisions to (20 20 1) in (x, y, z), respectively.

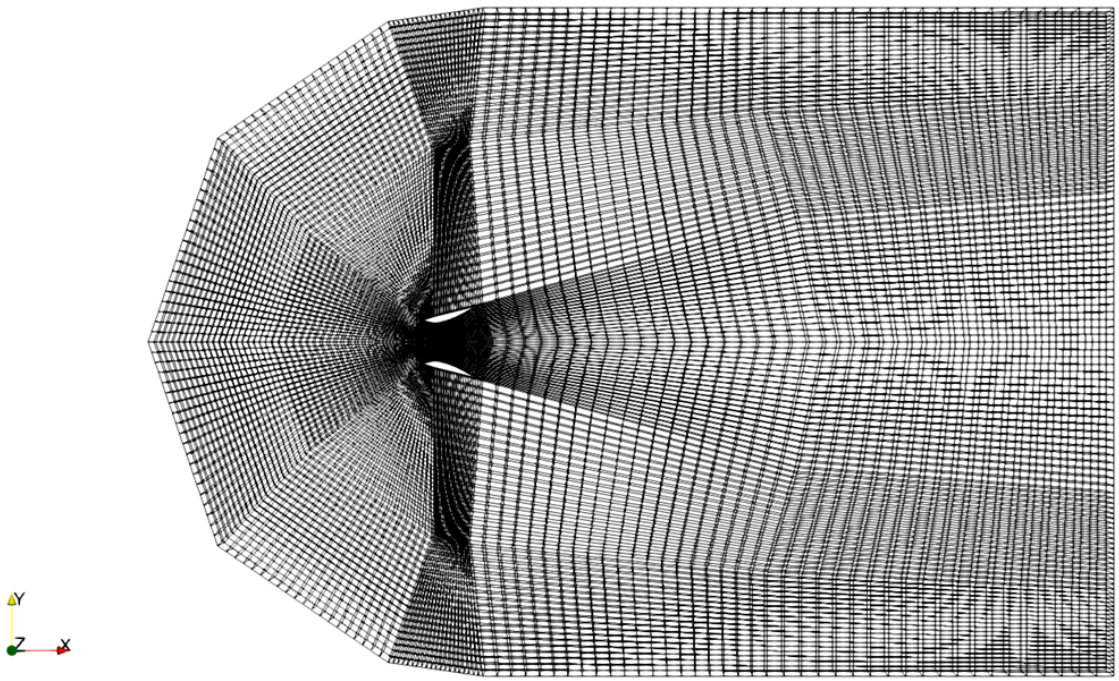

Figure 6: Testing #2 the grid of the C-section domain.

7. Add the arcs and polylines one by one. Start with exterior edges then do the interior ones. At each edit, check the change by repeating step (5) in Part II.

8. Apply the clustering on the blocks using simple grading and edge grading. You may use the OpenFoam User Guide Greenshields (2022) and equations to calculate the number of divisions in each direction.

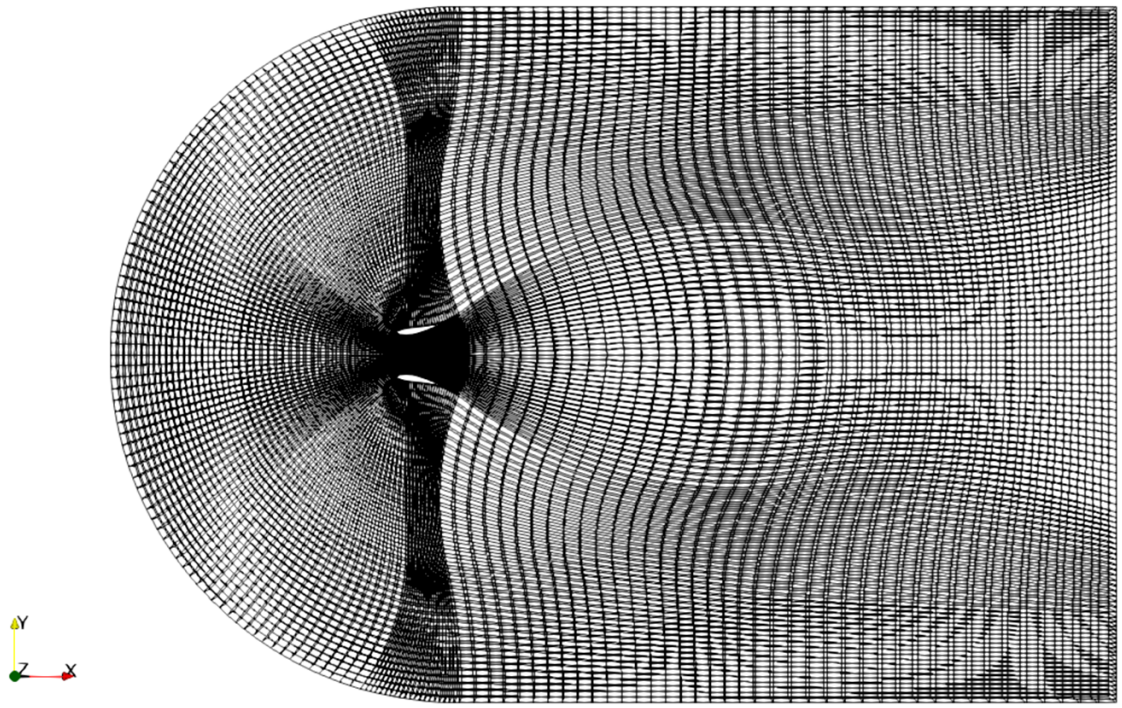

Figure 7: Testing #3 the grid of the C-section domain.

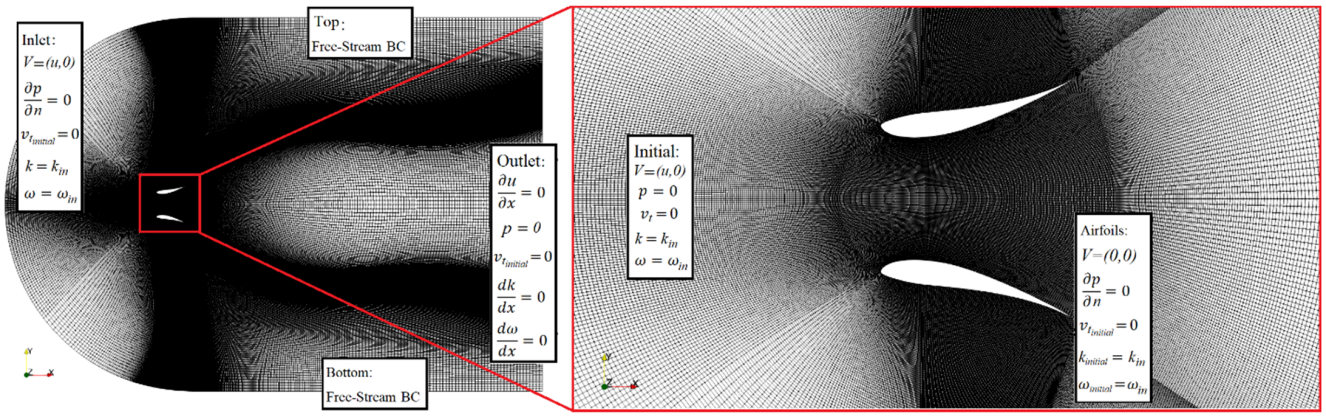

Figure 8: The grid of the C-section domain.

blockMesh calculates the cell sizes using a geometric progression to generate a non-uniform mesh. Along a length l, if n cells are requested with a ration of R between the last and first cells, the size of the smallest cell, Δs, is given by:

$$\Delta s = l\frac{r-1}{\alpha r - 1} \tag{1}$$

where r is the ratio between one cell size and the next which is given by:

$$r = R^{\frac{1}{n-1}} \tag{2}$$

$$\alpha = \begin{Bmatrix} R & for & R>1 \\ 1-r^{-n}+r^{-1} & for & R<1 \end{Bmatrix} \tag{3}$$

3 Mesh Quality Results

9. Finally, it is important to check and visualize the quality of the designed mesh. This is done by running the *checkMesh* command after generating the mesh. Then, visualize the mesh quality using Paraview.

The suggested options for *checkMesh* command are shown in the following line:

Code 5: Check mesh command on OpenFOAM to analyze and visualize the mesh quality

```
checkMesh -allGeometry -allTopology -writeAllFields -writeSets vtk |
tee log.checkMesh
```

The mesh quality parameters, such as skewness, aspect ratio, orthogonality, and so on, and topology quality parameters are explained in details in wolfdynamics.com (2017). The standard limiting values for mesh quality with *checkMesh* can be found in the file *primitiveMeshCheck.C*, located in the directory: *$WM_PROJECT_DIR/src/OpenFOAM/meshes/primitiveMesh/primitiveMeshCheck*. The standard values, that can be changed by the user, are defined as:

Code 6: Values of the standard mesh quality parameters in OpenFOAM

```
36 Foam::scalar Foam::primitiveMesh::closedThreshold_ = 1.0e-6;
37 Foam::scalar Foam::primitiveMesh::aspectThreshold_ = 1000;
38 Foam::scalar Foam::primitiveMesh::nonOrthThreshold_ = 70; // deg
39 Foam::scalar Foam::primitiveMesh::skewThreshold_ = 4;
40 Foam::scalar Foam::primitiveMesh::planarCosAngle_ = 1.0e-6;
```

Even if the results from the *meshCheck* were *okay*, it is highly recommended to follow the suggested following values by SimScale (2022) and CFDSupport (2022) to avoid low-quality meshes, because they will almost always result in numerical inaccuracies, causing the simulations to fail or produce misleading results.

Code 7: Recommended values of the mesh quality parameters

```
Aspect ratio < 10
Non-orthogonality < 60
skewness < 0.95
```

The mesh quality of the built mesh for the problem of reflected airfoils are shown in the figures 9-12. The mesh quality results meet the recommended values, expect for the aspect ratio. The aspect ratio can be improved by increasing the number of divisions, which will increase the computational time. Otherwise, the convergence speed will significantly decrease with the high aspect ratio, but likely it is not fatal for the solver stability. Thus, the next steps to be done are the mesh independent analysis and validation study.

4 Conclusion

The article provided a schematic procedure to construct a structured mesh over simple geometries using *blockMesh* in OpenFOAM. Further, the article test the procedure over a two-reflected-airfoils problem. The results of proposed procedure showed a good mesh quality, which was analyzed by *meshCheck* in OpenFOAM. A future work of constructing a similar procedure for the complex geometries using *snappyHexMesh* in OpenFOAM will be conducted.

5 Peer-Review

This article was firstly uploaded on Linked-In among the CFD and OpenFOAM community. The post can be found here. They discussed the article and provided suggestions for improvement.

```
Checking geometry...
    Overall domain bounding box (-2.15 -2.65 -0.1) (5.5 2.65 0.1)
    Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
    Mesh has 3 solution (non-empty) directions (1 1 1)
    Boundary openness (6.7515146e-17 5.7225872e-17 2.1139632e-14) OK.
    Max cell openness = 7.8486232e-16 OK.
    Max aspect ratio = 61.237417 OK.
    Minimum face area = 1.4175703e-06. Maximum face area = 0.0032470223.  Face area magnitudes OK.
    Min volume = 5.6702812e-08. Max volume = 0.00012988089.  Total volume = 7.4988797.  Cell volumes OK.
    Mesh non-orthogonality Max: 34.712564 average: 9.3489758
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 0.93734832 OK.
    Coupled point location match (average 9.704423e-17) OK.
    Face tets OK.
    Min/max edge length = 0.00097458738 0.074246292 OK.
    All angles in faces OK.
    Face flatness (1 = flat, 0 = butterfly) : min = 1  average = 1
    All face flatness OK.
    Cell determinant (wellposedness) : minimum: 9.824421e-05 average: 0.11714375
***Cells with small determinant (< 0.001) found, number of cells: 13170
  <<Writing 13170 under-determined cells to set underdeterminedCells
    Concave cell check OK.
    Face interpolation weight : minimum: 0.24149935 average: 0.49746866
    Face interpolation weight check OK.
    Face volume ratio : minimum: 0.31835939 average: 0.98936458
    Face volume ratio check OK.
```

Figure 9: Mesh quality Results of the built mesh for the problem of reflected airfoils.

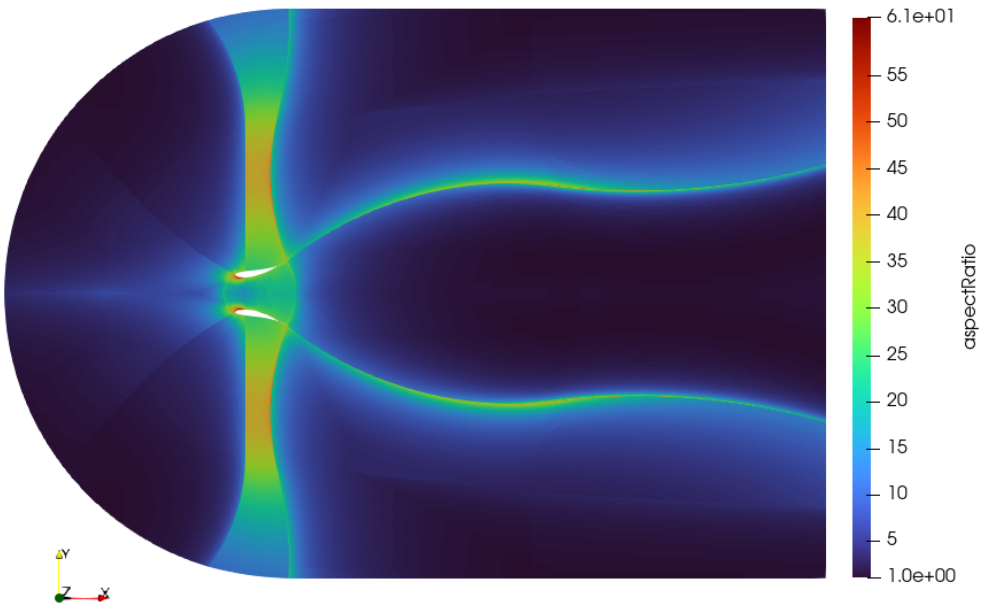

Figure 10: Aspect ratio quality of the built mesh for the problem of reflected airfoils.

6 Acknowledgments

The author thanks each of the Ocean TAMU professors Björn Windén, Mirjam Fürth, and Sharath Girimaji, for their technical support and invitation to write this article for the CFD users community for the Ocean Engineering department at Texas A&M University, ocean-cfd.engr.tamu.edu.

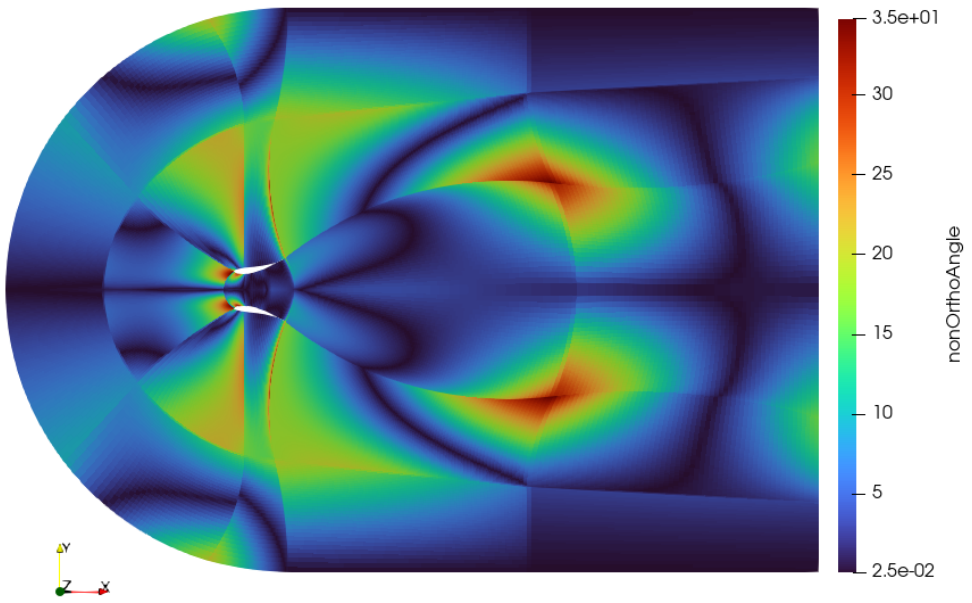

Figure 11: Non-orthogonality quality of the built mesh for the problem of reflected airfoils.

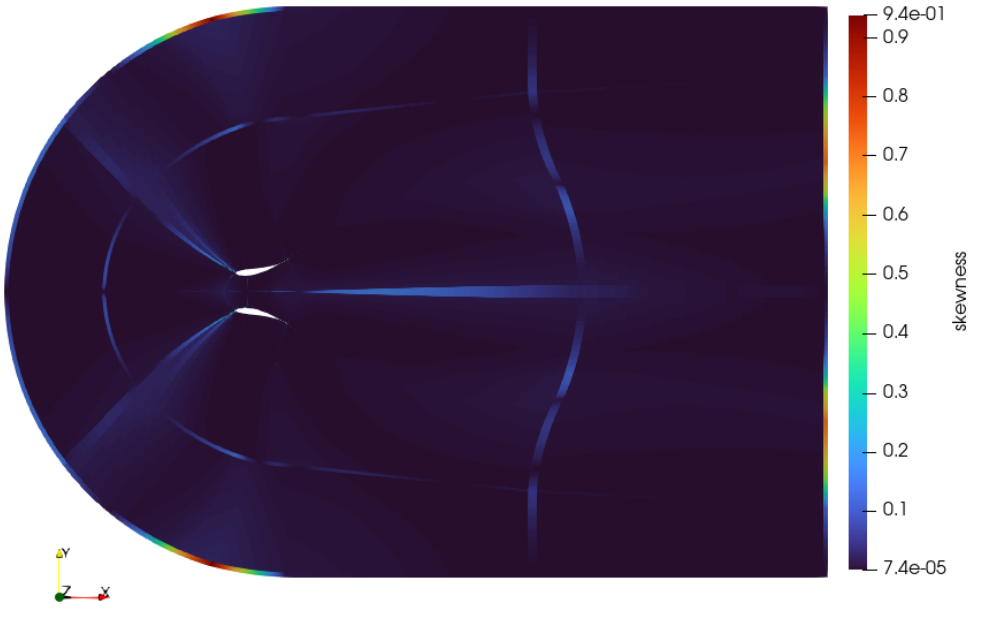

Figure 12: Skewness quality of the built mesh for the problem of reflected airfoils.

References

CFDSupport (2022). Mesh quality check. https://www.cfdsupport.com/OpenFOAM-Training-by-CFD-Support/node131.html. Accessed: 2022-09-04.

Greenshields, C. (2022). Mesh generation with the blockmesh utility. https://doc.cfd.direct/openfoam/user-guide-v10/blockmesh. Accessed: 2022-07-14.

SimScale (2022). Mesh quality, quality assessment. https://www.simscale.com/docs/simulation-setup/meshing/mesh-quality/. Accessed: 2022-09-04.

wolfdynamics.com (2017). What is a good mesh? meshing preliminaries and quality assessment. http://www.wolfdynamics.com/wiki/meshing_preliminaries_and_quality_assessment.pdf. Accessed: 2022-09-04.