© 2020 IJRTI | Volume 5, Issue 7 | ISSN: 2456-3315

Detailed Study in Grid Generation and its Topology Optimization

Karan Kumar Shaw

B. Tech Student Batch-21,
Aerospace Department of Engineering,
SRM Institute of Science & Technology, Kattankulathur-603203, Tamil Nadu, India

***Abstract*: A quality of the mesh plays a significant role in the accuracy and stability of the numerical computation. We should strive to maintain an optimal balance between the computational cost and the level of fineness achieved. The grid generation by algebraic methods based on mathematical interpolation function provides advanced geometry/mesh generation as well as mesh diagnostics and repair functions useful for in-depth analysis. This comparative study was intended to show the usefulness of various type of grid generation surrounding airfoil contours depending upon its application and the extent to which it captures the flow behind the trailing edge. Various Grid types like C-type, H-type & O-type were compared and optimized hybrid mesh is obtained. Critical Meshing techniques like sliding mesh & CHIMERA techniques were also looked upon.**

***Index Terms*: Meshing, Grid, Topology, Structural, CFD, Cell, Block.**

I. INTRODUCTION

All Airfoil geometry is seen as the stepping stone in aerospace / turbomachinery field, before diving deep into CFD. Irrespective of the gridding software and the gridding methodology adopted to mesh, everyone should understand how to achieve this goal of meshing an airfoil first. Despite the fact that meshing strategies for an airfoil have come a long way, newer, smarter and more efficient strategies continue to evolve, to capture the subtle physics in the most accurate and optimal way.

This article focuses on the traditional gridding strategies along with improved blocking techniques around airfoils for optimal CFD results.

II. BASIC GRID

Grid generation is a challenging operation during which the engineer has to maintain accurate mesh density as well as ensure that the mesh count is not impractically high. Computational fluid dynamics is a method of solving governing equations using numerical methods like finite difference, finite element or finite volume. The governing equations (that are either in differential form or integral form) are discretized and rewritten either at a point, an element or at a control volume (also called as cell). As the equations are written and solved in discretized form the domain or the geometry in which these equations will be solved also needs to be discretized. Once the governing equations have been discretized, a suitable grid needs to be generated to represent the nodes and/or cells within the computational domain.

Structural & Unstructural Mesh

There are two major types of grids in CFD: structured and unstructured.

Structured Grid: It consists of rectangles in 2D plate and Hexahedrons in 3D model.

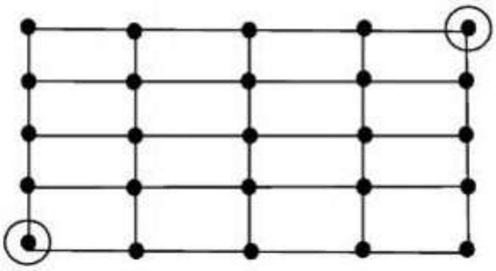

Figure 1: Structured Mesh

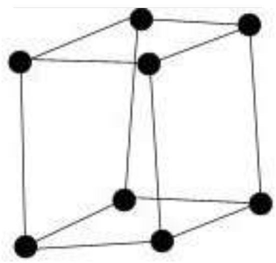

Figure 2: Hexahedral Element

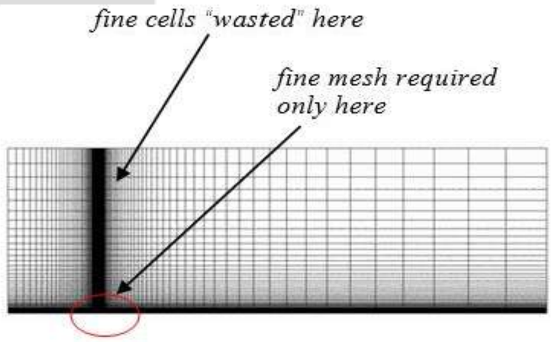

Figure 3: Structural Mesh Problem

Advantage

- The connectivity is clear from indexing alone
- It is easy to manipulate and store in solver.

Disadvantage

- Restrictions on orthogonality & aspect ratio and difficult to wrap around complex geometries
- Less efficient for locally refined meshes and manual mesh generation is difficult.

Unstructured Grid: It consists of triangles in 2D plate and tetrahedrons in 3D model.

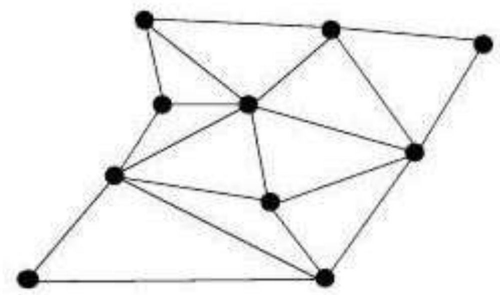

Figure 4: Unstructured Mesh

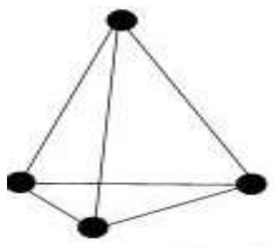

Figure 5: Hexahedral Element

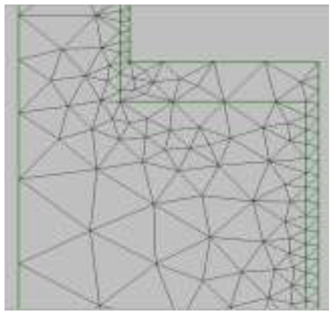

Figure 6: Unstructured Mesh Corner

Advantage

- It is easier to wrap around complex geometries.
- It is very for locally refined meshes and manual mesh generation is easy.

Disadvantage

- Connectivity info for each cell needs to be stored.
- It is difficult to store and manipulate in solver.

Grid Transformation

For most practical problems, the body geometry is usually curved and the flow exhibits regions of strong gradients. To accommodate these effects, the grid needs to be:

Boundary-fitted: Grid wrapped around the body, e.g. around the airfoil.

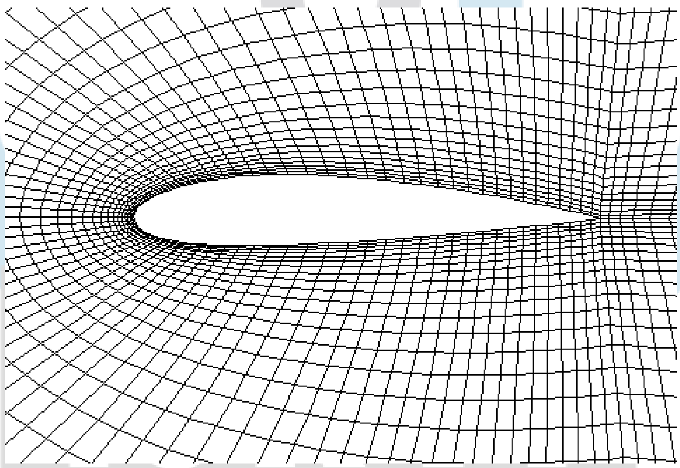

Figure 7: Boundary-fitted Structured Mesh around an airfoil

Curvilinear: This concept has cells are rotated relative to the x-y coordinates.

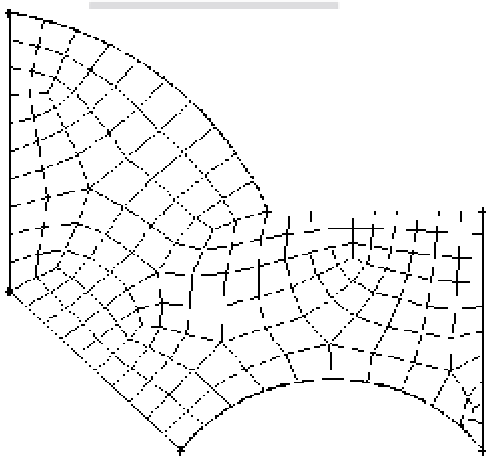

Figure 8: Illustration of a curvilinear Structured Mesh

Non-Uniform: This design has unequal spacing of cells inn any one direction

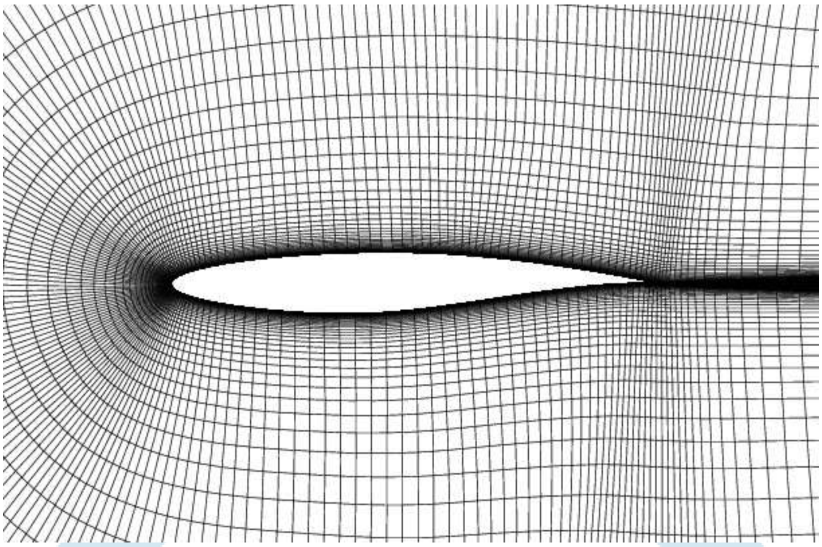

Figure 9: Curvilinear, body-fitted, non-uniform structured mesh around an airfoil

In such cases, the conventional finite difference quotients, $\frac{\partial f}{\partial x} = \frac{\Delta f}{\Delta x}$, are difficult to express and hence, the grid has to be transformed from the

Physical plane

Curvilinear non-uniform x-y coordinates

to the

Computational plane

Rectangular uniform ξ-η coordinates

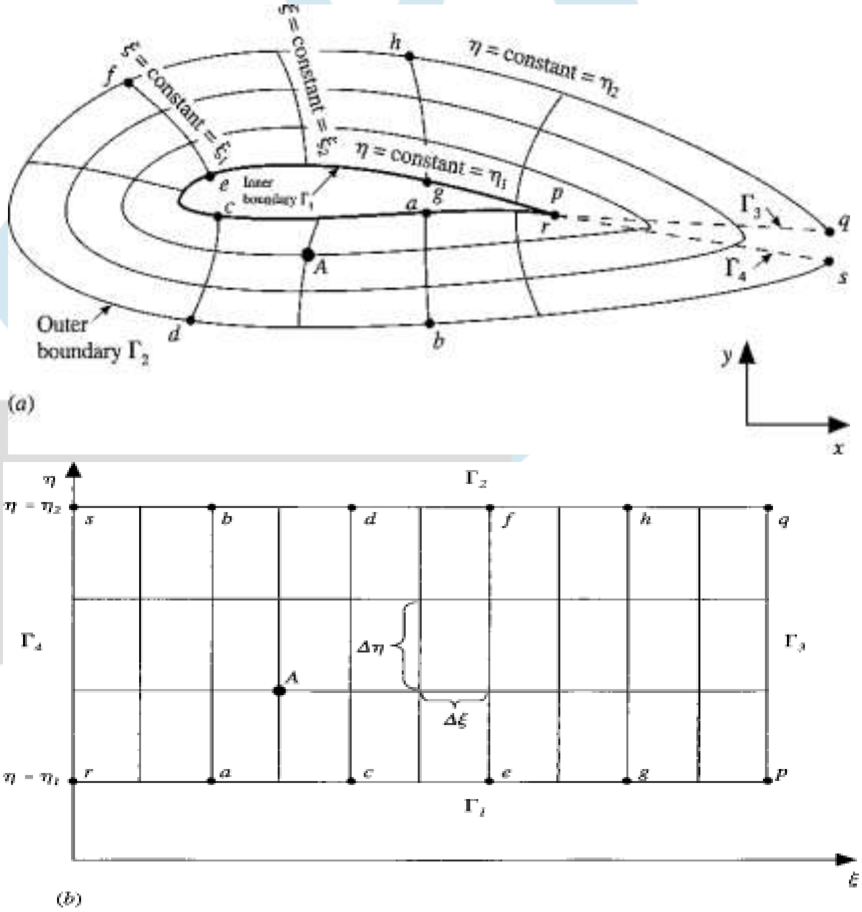

Figure 10: Illustrations of a grid transformation around an airfoil

i.e. we need to transform the independent variables (ρ, u, v, p, t) from the physical space (x, y, t) to the computational space (ξ , η, τ), where:

$$\xi = \xi\,(x, y, t)$$
$$\eta = \eta(x, y, t)$$
$$\tau = \tau(t)$$

In essence, we need to replace all 1st and 2nd order derivatives in the governing equations by:

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x} \quad \text{………………………………….. (i)}$$

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial y} \quad \text{…………………………………. (ii)}$$

Using these expressions makes the governing equations in the computational space (ξ, η, τ) to look awkward and way too involved mathematically. So, a more convenient form of the transformation is obtained by using inverse metrics or Jacobians:

Inverse Metrics:

$$\frac{\partial u}{\partial \xi} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \xi} \quad \text{………………………………. (iii)}$$

$$\frac{\partial u}{\partial \eta} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \eta} \quad \text{………………………………. (iv)}$$

We can find its solution by using Cramer's rule:

$$\frac{\partial u}{\partial x} = \frac{\begin{vmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}}{\underbrace{\begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}}} \Rightarrow J \equiv \frac{\partial(x, y)}{\partial(\xi, \eta)} \equiv \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}$$

Jacobian matrix denoted as J and one can then express the governing equations in the computational space as shown the example below:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \Rightarrow \frac{\partial U_1}{\partial t} + \frac{\partial F_1}{\partial \xi} + \frac{\partial G_1}{\partial \eta} = 0$$

With:

$$U_1 = JU$$

$$F_1 = JF\frac{\partial \xi}{\partial x} + JG\frac{\partial \xi}{\partial y}$$

$$G_1 = JF\frac{\partial \eta}{\partial x} + JG\frac{\partial \eta}{\partial y}$$

Cartesian Meshes

A special class of structured meshes are Cartesian meshes, which are grids with maximum degree of structure. For Cartesian meshes, *all cells are square (or in 3D cube) shaped, which sides are aligned with the exes of the coordinate system* (Fig. 11).

Since Cartesian meshes combine the best of both worlds from the unstructured and structured grids, i.e.

- easy mesh generation (typical for unstructured grids)
- easy storage and manipulation of cell connectivity (typical of structured meshes)

They have become popular recently (since about 2000) for complex and largescale problems where frequent re-meshing is required during the calculation (such as predicting the near-aircraft path of a missile launched from an F-18 fighter jet).It is really the evolution of the Finite Volume method, which opened the avenue for this type of grids, since the *truncated cells near the surface can be easily solved by Finite Volume methods* (for which the cell shape is notimportant).

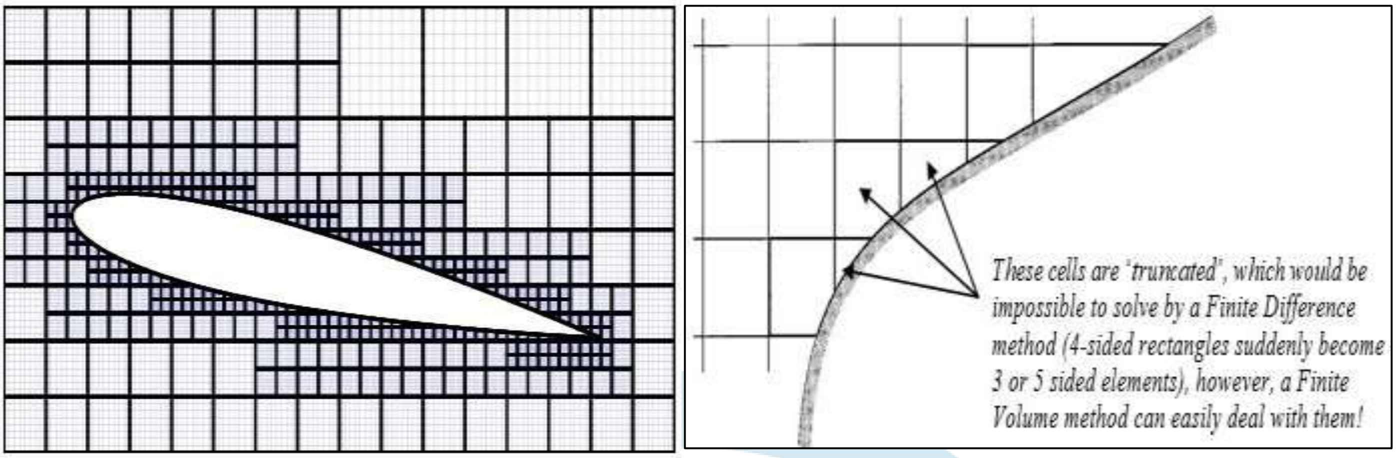

Figure 12: Cartesian Mesh around the boundary of an airfoil with refinement close to the surface

Cartesian meshes are used for very complex geometries (full F-18 fighter jet model with extended landing gears, missiles and auxiliary fuel tanks at wing tips) where grid generation with other techniques would take ages. Combined with *"adaptive multigrid"* techniques, it forms a very powerful tool, although often limited to inviscid flows only. This is the method employed for example the commercial VECTIS code as well.

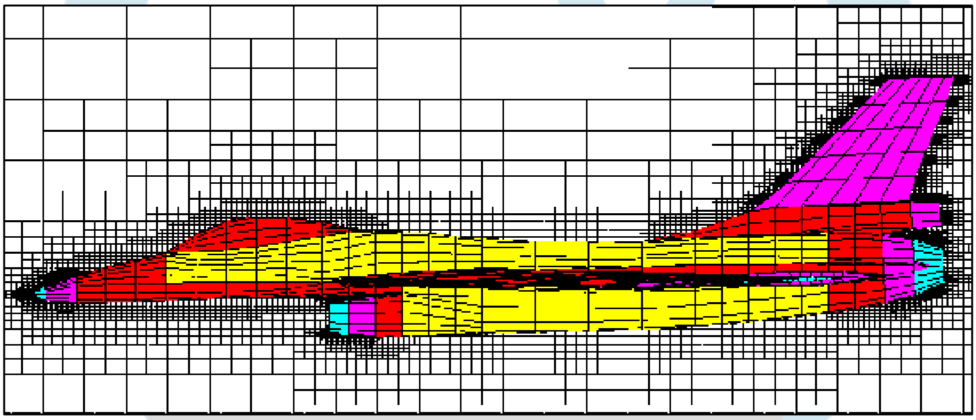

Figure 13: cartesian Mesh around F-16 aircraft

Zonal or Block-Structured Grid

Many solvers are set up to work with meshes divided into zones (the term frequently reserved for unstructured meshes) or blocks (the same term used for structured meshes):

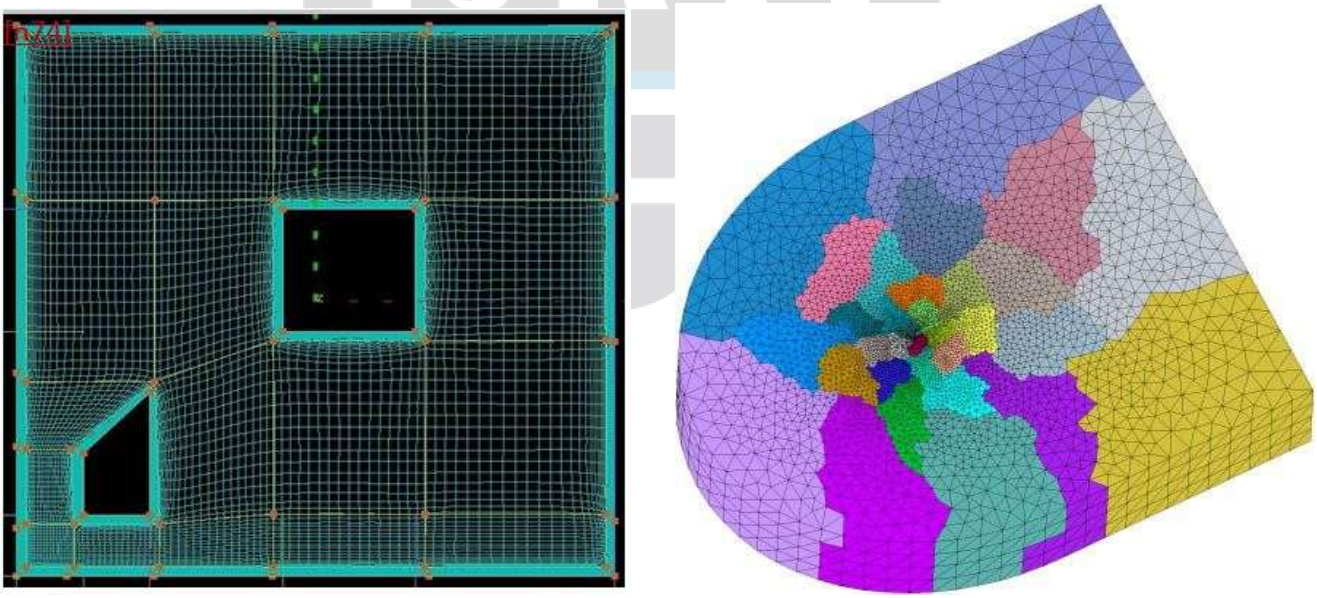

Figure 14: Multi-block structured mesh (left) and unstructured zonal mesh (right). The different rectangles/trapezoids on the left and the different colors on the right represent different blocks (or zones).

Advantages

- It has easy control over local mesh refinement.
- It is suitable for parallel processing (each zone/block run on a different processor).

Disadvantages

- Connectivity of zones/blocks has to be defined/stored in the solver.
- The boundary conditions for a block face reads from the neighboring block's corresponding face.
- Each block/zone face must form one type of boundary, i.e. mixed boundary conditions on any one face are typically not allowed.

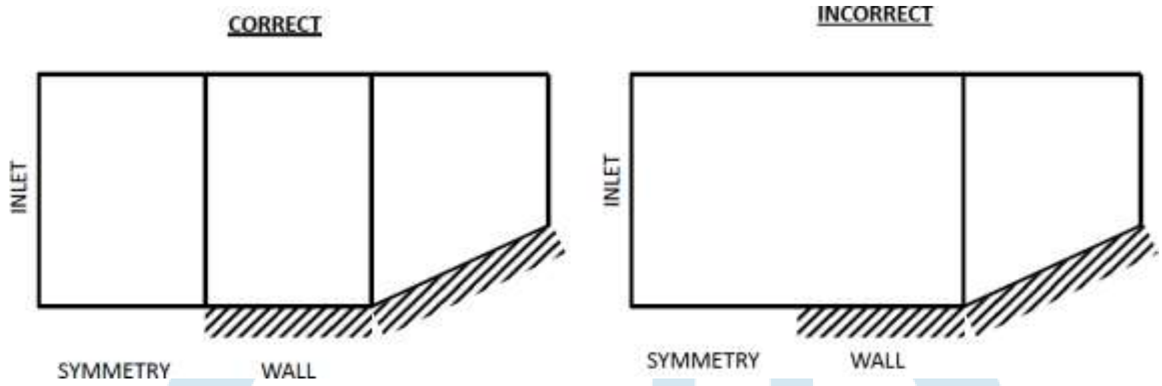

Figure 15: Rules for block connectivity: an edge of a block usually must be of the same boundary type.

For many block-structured solvers, only compatible block sizes are allowed.

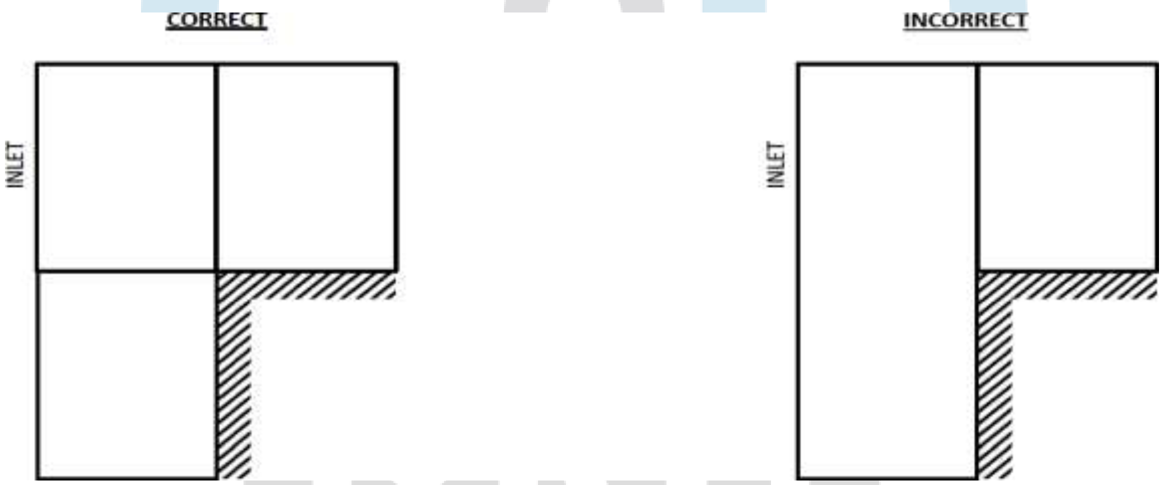

Figure 16: Rules for block connectivity: only compatible block sizes are allowed.

Hybrid Meshes

To capture boundary layers in viscous (laminar or turbulent) simulations, one needs to introduce a quite a fine mesh near solid surfaces. The general rule is:

- To have atleast 10-15 points inside the boundary layer.

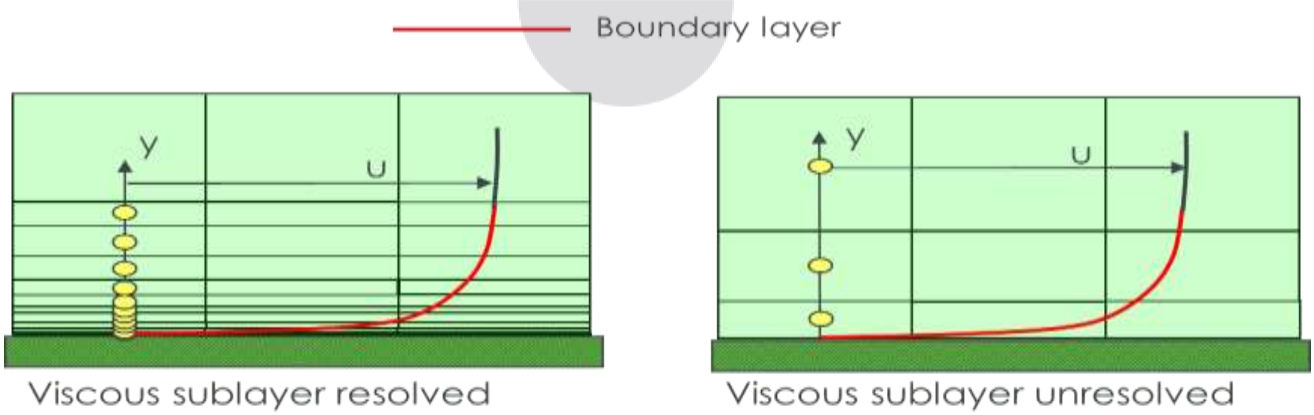

Figure 17: In order to resolve a boundary layer properly, atleast 11-15 cells need to be inside the boundary layer

- The first spacing near the wall shall be in the order of $yP^+ = (1\sim10)$, i.e. inside the viscous sublayer for models without wall functions.
- The first spacing near the wall shall be in the order of $yP^+ > 30$, i.e. outside the viscous sublayer for models with wall functions.

Achieving such grid density near the wall is often not efficient - the node count is typically much higher than for structured meshes. Hence, many solvers, such as ANSYS-CFX, offer the possibility of using hybrid meshes, which combine the advantages and disadvantages of structured and unstructured meshes, i.e.

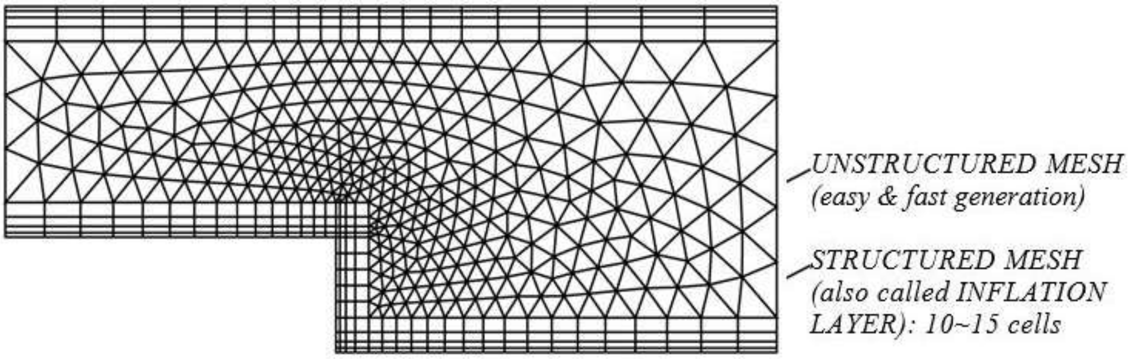

Figure 18: Hybrid Structured-Unstructured Mesh. Structured mesh is used around the solid walls to capture the boundary layer, while Unstructured mesh in the freestream.

A h mesh hybrid mesh cells calls for Prism type cells in the inflation layer and Tetrahedron type cells elsewhere.

Figure 19: Cells created in a hybrid Structured-Unstructured Mesh. The Prism cells occurs in the structured mesh part, while the Tetrahedron in the unstructured mesh part of the domain.

III. Mesh Techniques

Moving Mesh Techniques

Many applications require that the body of interest moves across the stationary computational domain. Examples involve releasing a missile from an aircraft, tracking the debris falling off from the Space Shuttle, rotating blades of a helicopter rotor, etc.

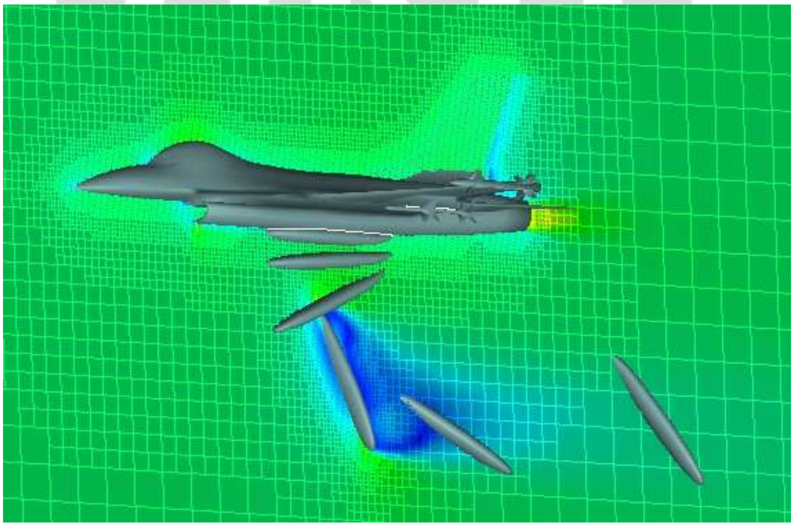

Figure 20: CFD simulations of releasing a missile from an F-16 aircraft.

For such cases, various moving mesh techniques need to be applied, from which we will review the 2 most popular techniques:

- Sliding Mesh Technique
- CHIMERA Technique

Sliding Mesh Technique

Commonly used for problems, where the motion of the body is not arbitrary. Sliding meshes are used for simulating geometries moving inside a stationary domain. This is a very popular technique for rotating bodies, such as turbines, impellers, propellers, or helicopter rotors. For the sliding mesh interface, we need special treatment of the fluxes for the instantaneously non-overlapping cells.

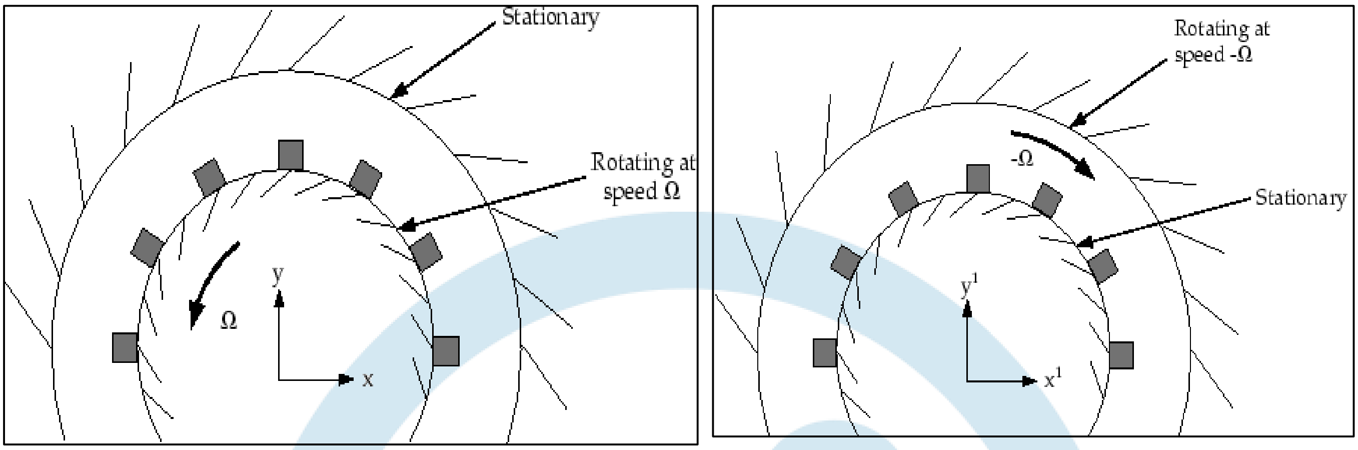

Figure 21: Original Reference Frame (left) and Rotating Reference Frame (right)

CHIMERA Technique

In CFD, the CHIMERA technique involves using 2 overlapping grids, one which is fixed to the moving body (Minor Grid), and one, which is fixed in the coordinate system (the Major Grid). Then, the flow variables are calculated with a relatively complex interpolation technique between the two grids.

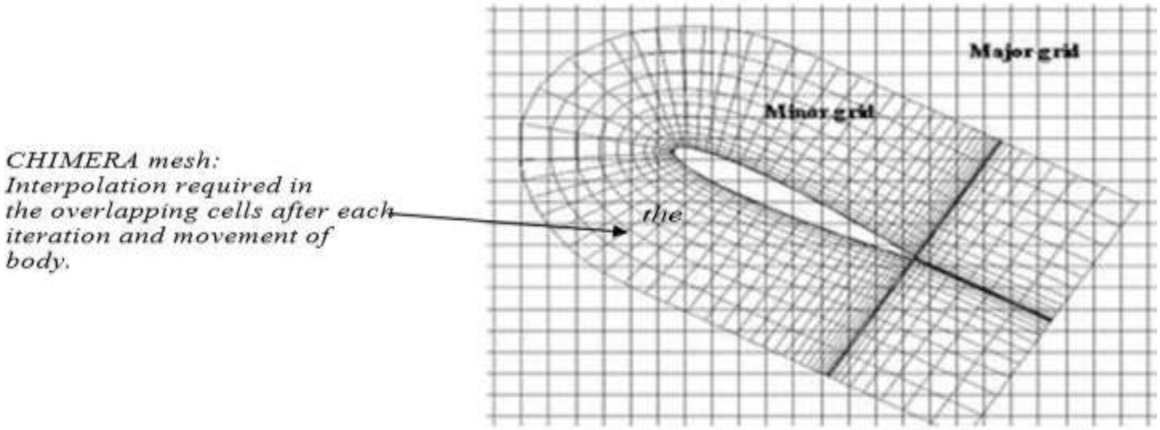

Figure 22: A Chimera Mesh for an airfoil simulation with time-varying angle of attack

CHIMERA is ideal for problems where the trajectory of the moving body is not known *a priori*. An example is the missile release from an F-16, in which the missile has 6 degrees of freedom movement. It requires a mechanism to cut-off or discard those cells of the Major Grid which are covered by the moving body.

Deforming Mesh Technique

When the body of interest moves only slightly relative to the computational domain, it is possible to deform the cells of the computational domain via an automatic regeneration technique. This is called the Deforming Mesh method. One example is the automatic regeneration of the mesh based on the wall surface motion, for which for example the so-called Trans-Finite-Interpolation (TFI) technique can be used.

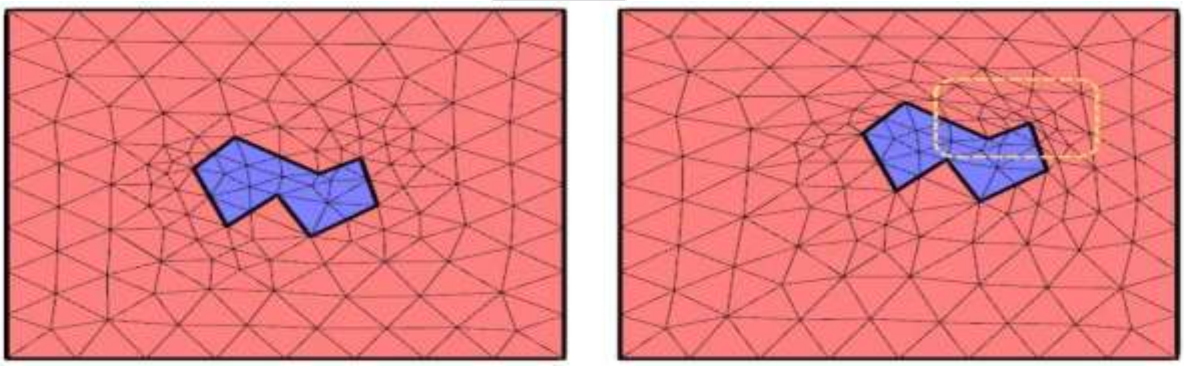

Figure 23: Deforming Mesh showing in the yellow box.

The total number of cells has not changed but their topology has been deformed in the pink zone in order to accommodate the motion of the purple zone.

Adaptive Mesh Technique

An adaptive mesh is a grid network that automatically clusters grid points in regions of high gradients. It evolves in steps of time in conjunction with a time dependent solution of the governing equations. Adaptive meshing usually means redistribution of the existing cells, instead of introducing new ones.

Advantages

- There is no need to know the flow details such as location of shock waves, thickness of boundary layer etc. in advance
- It has increased accuracy in comparison to fixed-grid approaches.

Disadvantages

- The grid generation increases computational cost.

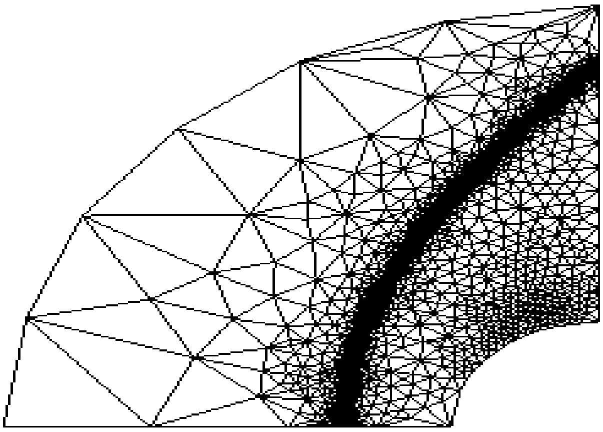

Figure 24: Adaptive Mesh technique automatically clusters cells around the region of high -pressure gradients

Adaptive mesh technique applied to high speed flow over a blunt body. Note that most of the cells have been automatically clustered by the code around the region of high-pressure gradient, i.e. the shock wave.

IV. Grid Type

In order to achieve efficient solutions of large systems of equations using the philosophy of the multigrid method by gaining superior convergence rates by carrying out the early iterations on fine grid and then to progressively transferred these results onto a coarse grid, the grid is categorized into three types:

H-type Grid

One of the classical gridding approaches is the H-type pattern. It has finely clustered cells at the leading and trailing edges as shown in *figure 25*. This pattern is simple to construct and holds good for a biconvex airfoil with sharp leading and trailing edges. However, when applied to an airfoil with curved leading edge, the H-pattern creates a singular point *(figure 25)*. This can partly be alleviated by splitting the singularity into two weaker singularities as shown in *figure 26*. But still, it falls short of capturing the leading-edge curvature accurately and also propagates the boundary layer in the transverse direction *(figure 26)*.

Alongside the wasted use of point clusters and high aspect ratio cells, the high clustered cells both parallel and perpendicular in regions where the flow accelerates, can result in significant time step reductions due to CFL conditions, leading to slowing of solver convergence.

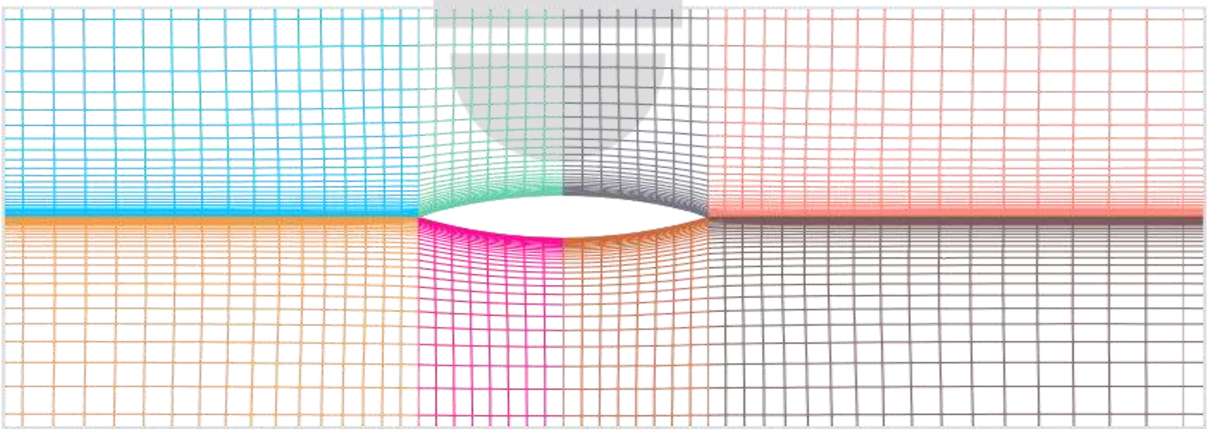

Figure 25: H-type pattern for a bi-convex airfoil

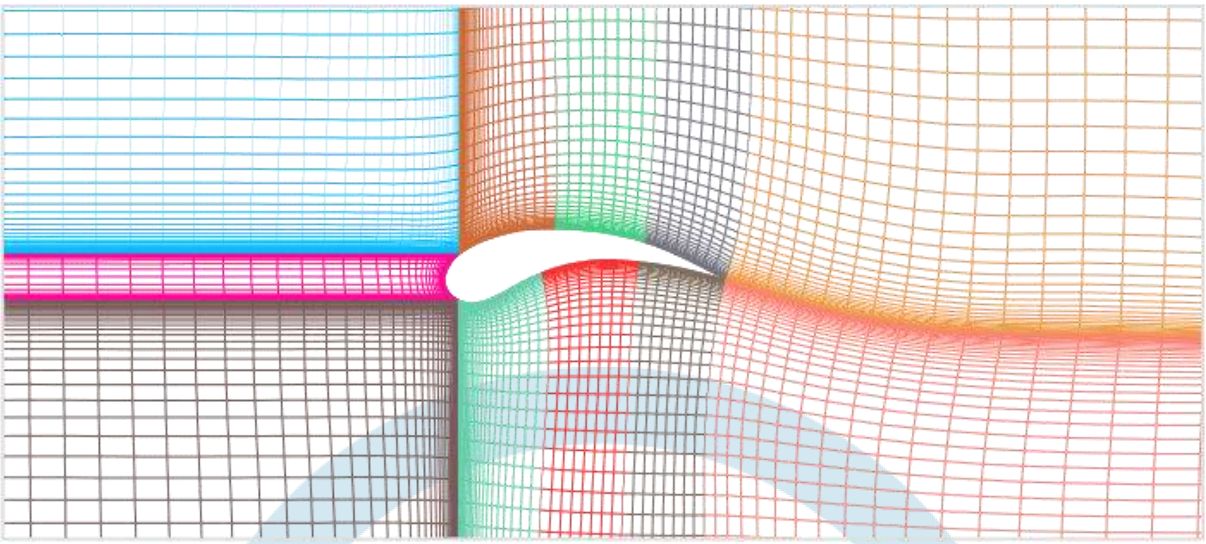

Figure 26: H-type pattern with boundary layer clustering

C-type Grid

An improved variant of H-type topology is the C-type pattern, which captures the leading-edge curvature without any singularities. Though, the C-type pattern avoids the propagation of boundary layer fineness upstream, it fails to do so in the downstream of the airfoil trailing edge. In a way, this downstream fineness proves beneficial as it helps to capture the shear layer for solver runs at low angles of attack.

In practice however, most applications of the C-type pattern, have not taken advantage of the wonderful alignment of the grid lines along the shear layer. To take full advantage of this pattern, the CFD analysis would have to continually shift the grid curves to dynamically align with the shear layer and this would result in greater conformity to the flow physics. While not doing this alignment results in algorithmic simplicity, it does result in the wastage of grid points downstream and falls short of being an efficient grid type.

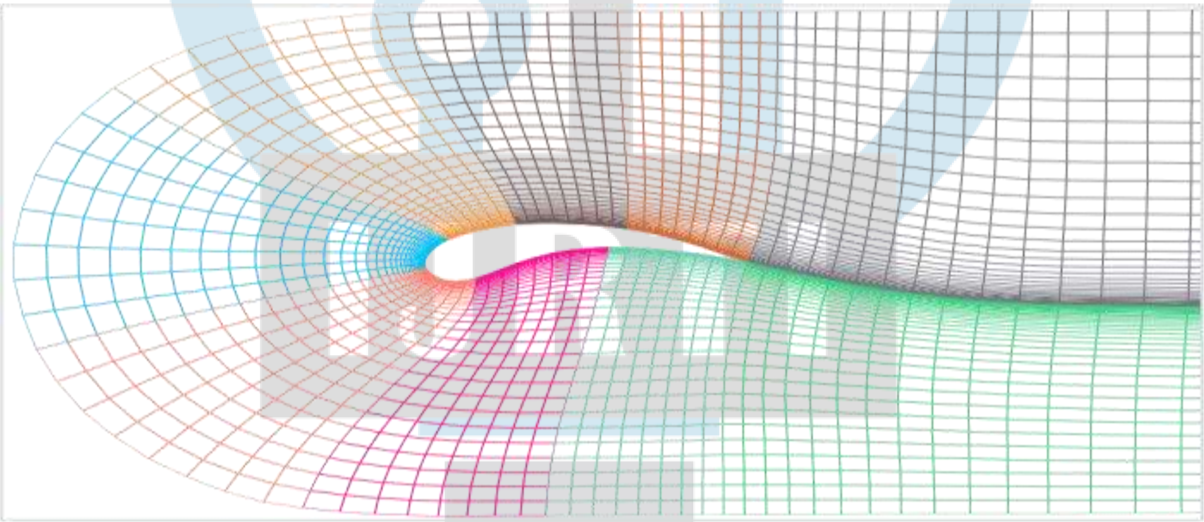

Figure 27: C-type pattern for a blunt leading-edge airfoil with boundary layer clustering

O-type Grid

The next classical blocking strategy is the O-type, which almost overcomes the disadvantages of the H-type and C-type grids. In O-type the entire grid sheet is wrapped-up around the airfoil without any propagation of the boundary-layer fineness into the field. This helps in getting an optimal cell count, eliminating the redundant propagation of cells as seen in the H-type and C-type patterns.

However, this pattern has its inherent limitations. It creates highly skewed cells for zero thickness trailing edge airfoil. The grid cell on the upper airfoil surface directly connects to the cell on the lower airfoil surface across the horizontal grid line emanating from the trailing edge. For a single cell-to-cell step, it nearly creates a 360 degree turn in one step. Thus, each cell will have an angle of slightly less than 180 degrees which is not exactly of good quality.

The level of skewness created in such cases is acceptable for Euler grids, but is extremely high in viscous grids. Singular points like the trailing edge are critical for accurate CFD computations. High skewness will affect the solver's robustness and also the quality of the solutions. For this reason, many prefer to go with the C-type grid, and live with the excessive cells in the wake region.

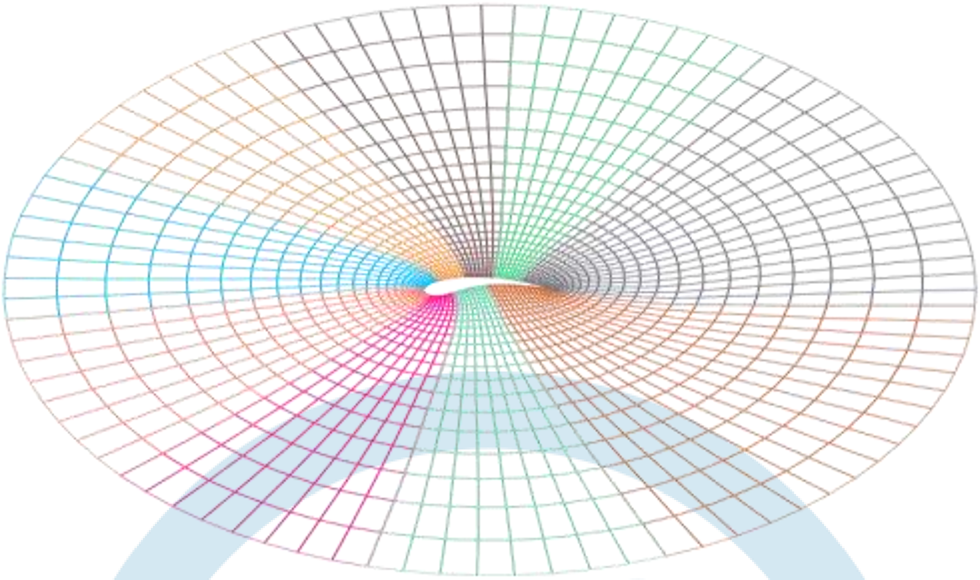

Figure 28: O-type pattern for a blunt leading-edge airfoil with boundary layer clustering

V. Conclusion

Major flow phenomenon like the wake and shocks can be effectively captured by smart topology building without the limitations posed by traditional-blocking strategies like the C- type and O-type topologies.

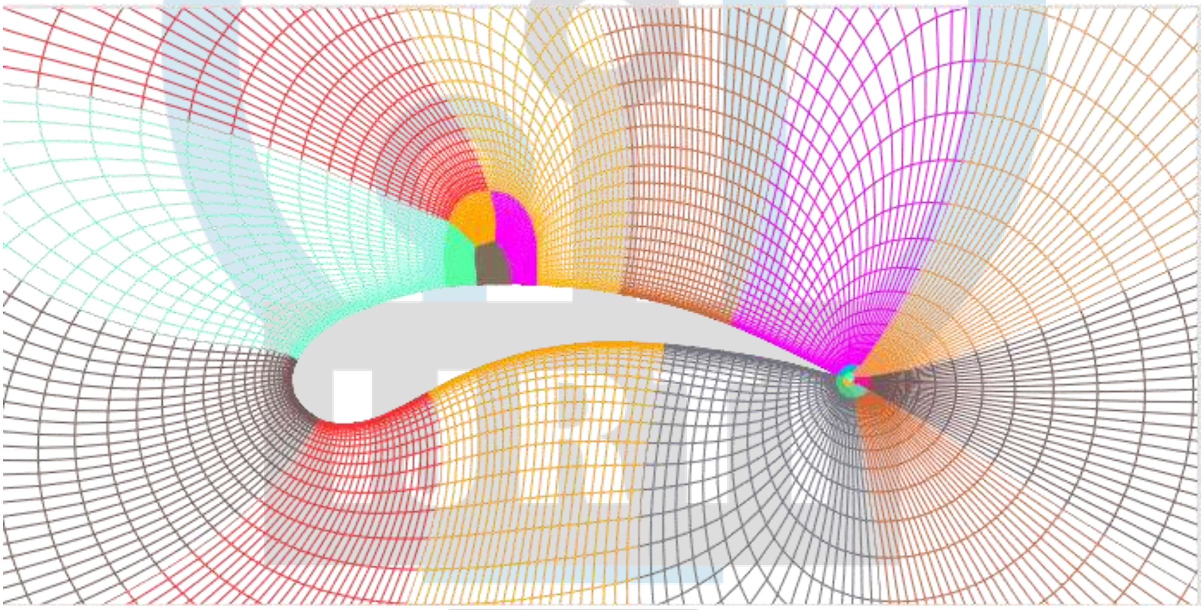

Figure 29: Topology for shock capturing on the upper surface of the airfoil

Figure 29 shows the grid block layout for shock capturing with a known shock position. As it can be observed, the blocks are placed right at the discontinuity, capturing the shock with a dense cloud of grid points and gradually smoothing out as we move away from the location. The grid density in the central cocoon of blocks can be varied to the users desired level of resolution, without affecting the neighboring block grid size.

If CFD was able to capture the gross flow-physics and make modest prediction of the flow parameters, it was perceived as an accomplishment. CFD now is no more seen as an aid to Experiments for data cross-verifications, but is widely accepted as a precursor and a reliable design tool. The innovative strategies discussed here are not out of context. They are very effective and they enhance the CFD results, making them more reliable and accurate. Recent advances in the computer technology has paved the way for transitioning from RANS to higher order CFD methods, LES and even DNS. And they all demand low aspect ratio, high quality quad/hex meshes. These unique blocking strategies help to cater to their needs, without the hidden cost of cell count shoot-up, loss of flow-alignment and orthogonality, poor cell quality as seen by conventional gridding techniques.