TMP UNIVERSAL JOURNAL OF RESEARCH AND REVIEW ARCHIVES

VOLUME 4 | ISSUE 2 | YEAR 2025 | APRIL – JUNE 2025

RECEIVED DATE	ACCEPTED DATE	PUBLISHED DATE
15/03/2025	05/04/2025	15/04/2025

Article Type: Research Article

Available online: www.tmp.twistingmemoirs.com

ISSN 2583-7214

DEVELOPMENT OF AN EFFICIENT SIMPLIFICATION METHOD FOR 2D AND 3D HYBRID MESHES IN COMPUTER GRAPHICS APPLICATIONS

[1]Kamyar Safari

[1]Department of Computer Engineering and Information Technology, Payame Noor University (PNU), P.O. Box, 19395- 3697, Tehran, Iran

Corresponding Author: Kamyar Safari

ABSTRACT

In the preprocessing stages of the computer graphical software. The Meshes are used to approximate and describe the models. These Meshes follow specific structures and they are made up of various elements. In many applications, there is no need to use massive and very detailed meshes. To simplify meshes, some methods are used which are known as coarsening methods. These methods simplify a mesh in such a way as to has the most similarity to the original shape. However, degradation of the mesh must be kept to a minimum and the structure should be maintained as much as possible. The objective includes creating a method for coarsening 2- and 3-dimensional hybrid meshes with acceptable accuracy; providing an appropriate platform for conducting research on hybrid meshes; reducing the final size of the designed meshes for rendering operations; accelerate the transfer of meshes on the network platform; increasing network data transfer speeds because of the mesh size reduction; facilitation and acceleration of convergence in solving equations of calculating external flows by simplifying the computational meshes.

Keywords: 3D Mesh, 2D Mesh, Graphics Software, Optimization, Simulation, Model Simplification, Model Coarsening, Metric Fields, Edge Contraction

INTRODUCTION

The visual quality of a mesh depends on two factors. Number of mesh elements (if higher, the quality is better) And the quality of each cell and Face separately (The number of large angles should be at least). Although the number of cells and more elements makes visual images more attractive and more qualitative, on the other hand, in some cases it causes overhead due to the use of more resources which in interactive applications, makes the performance unacceptably

low. In the field of Computer Graphics, 3D models are used to describe the shape of a 3D space object that they represent. Different tools are used to represent these shapes, for example, the surface of curve modeling, constructive solid geometry, voxels, etc [11].

To improve the rendering performance, we usually need simplification of the original terrain mesh. More attention has been paid to how to improve the efficiency of mesh simplification and maintain geometric characteristics of terrain effectively. The existing mesh simplification algorithms include vertex deletion, edge collapse, vertex clustering, resampling and so on [12]. Meshes are generally divided into three categories. Structured, Unstructured and Hybrid meshes. The type of mesh that is used depends on the process to be performed.

A hierarchical structure of a mesh is such that the mesh consists of a number of cells or unit volumes. Each volume or cell consists of joining a number of faces and each face also consists of a number of vertices. These vertices are connected using the Edges. Any connection between two vertices is called one edge. Simplifying or coarsening of a mesh, replacing the current complex mesh by a simpler model, so that the overall shape of the original mesh is maintained. This operation is one of the most important steps in preparing the meshes for processing in various applications.

Multi-grid methods: are one of the most effective methods for solving linear and nonlinear algebraic equations in computational science. In these methods, a mesh sequence can be created to make each mesh simpler than its previous mesh. Generally, in these methods, the generation of the sequence of meshes is controlled by two conditions, First, Until the mesh can be simplified. Second, mesh quality does not exceed the threshold [4].

Metric Fields: In this study, we use some functions that called metric fields to coarse a fine mesh with a high anisotropy in the boundary layer. These functions convert the problem-solving field and graph generated from the mesh into a metric field. The general method is that at first, characteristics of the cells stretching size and stretching directions will be generated, Then the generated metrics are interpolated onto the nodes and the metric tensor will be generated on the nodes.

In the selection stage for detect removable mesh elements, the purpose of using the metric fields is to increase the precision of the mesh coarsening with a high degree of anisotropy (aspect ratio of order 104) [7].

The proposed solution is the mesh coarsening in the direction of perpendicular to the maximum elongation. Metric mapping is used to determine the direction of the most elongation of the elements. After defining the metric fields of the cells and interpolating its values on the vertices, the eigenvalues of this metric tensor, which represent the direction and size of the elements, changes in such a way that the shape of the element approaches to a equilibrium [7].

The bug of this method is in the case when the mesh is becoming completely anisotropic to an isotropic state, In this case, the mesh may cause deformation in the isotropic area. This transition zone is known as the "H-Shock". To solve this problem, a solution to limit mesh changes should be implemented. This can be accomplished by two ways of limiting the size of the characteristic function changes and limiting the Euclidean length ratio of the two adjacent edges.

IMPORTANCE AND NECESSITY OF RESEARCH

In many numerical computing software, using these algorithms increases the convergence rate of solving equations. Also, in addition to the accuracy of the algorithm, in the development of

coarsening algorithms, the reduction of its operational overhead is very important. An accurate and fast model simplification algorithm can increase operational efficiency, such as rendering images in graphical software, as well as solving equations in computational applications. At present, there are several implementations of algorithms for coarsening meshes in commercial programs. But due to the exclusivity, the algorithms and those codes are not available, therefore can't be developed and used in other studies. This issue proves the necessity of developing these types of algorithms, so that the results can be used in different researches. In the embedded systems that calculate flow equations for the devices in a dynamic and instantaneous manner, such as computers in aircrafts, as well as missile and other systems, the proposed method can improve the response time, processing time and the accuracy of the required processings.

Computational Fluid Dynamics (CFD) commercial applications are the most commonly used method of simulating flow in the field of fluid mechanics. Considering that the design and construction of any mechanical device requires extensive and costly testing in well-equipped laboratories. By solving basic equations and stream simulations, we can make significant savings in time and cost.

The data for microstructure computational models can come from a variety of sources. Three principal sources are analytic geometry, voxel data, and volume fraction data. Analytic geometry may be given by a series of geometric primitive specification. Voxel data is provided as a dense set of integers on a Cartesian grid where the integer represents the dominant material in each cell. For analytic geometry and voxel data the input must first be converted into a volume fraction representation to be processed. For analytic geometry, a series of inside-outside tests are performed on a uniform grid of sampled locations within each cell to approximate the volume fraction of each material present [8].

General Concepts:

It is possible to create meshes that are structured in some areas and are unstructured in some other areas. These types of meshes are called hybrids. This type is most commonly used in boundary layer meshes and in computing software that requires very accurate computing in the boundary regions. Most of the algorithms for eliminating and simplifying the elements of a mesh, remove elements, points, or edges, based on one of the three methods of vertex clustering, edge contraction, and vertex removal.

Vertex clustering: A cell that can be 2D or 3D depending on the type of mesh, is placed around the input model. Then this large cell is divided into a number of smaller cells. An element of each cell is selected and the rest of the cell elements are merged with that element and then deleted [3].

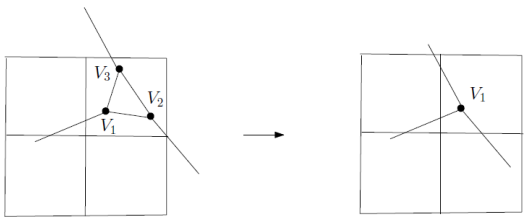

Figure 1: Simplification by vertex clustering

Edge contraction: In this method, all of the edges are selected repeatedly, and at each time it is repeated, the desired edge contracts toward one of the two vertices attached to it. By contracting the edge, all the elements connected to the desired edge will be updated. Many implementations of this algorithm can be made [3].

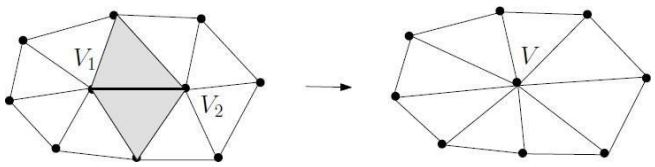

Figure 2: Simplification by Edge contraction

Vertex removal: In this method, each vertex will be removed with its adjacent faces. In this method, like the edge contraction method, a cost function determines which vertices are best to be selected to minimize the mesh. After selecting the vertices and removing each vertex with its adjacent faces, a hole arises that is surrounded by a circuit of the edges; therefore, at this step, the resulting hole needs to be restored. The removal of the cavity is done by a series of operations called "re-triangulation" [3].

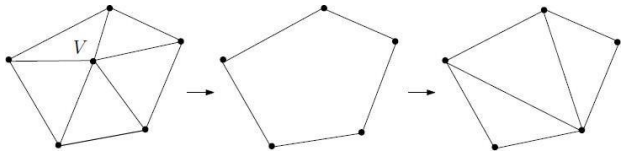

Figure 3: Simplification by Vertex Removal Method

Deciding which type of element is optimal to use:

Suitability of the four most commonly used element types along with varying mesh resolution in CPFE modeling of grain structures are investigated using large-scale simulations. A voxel-based polycrystalline grain structure is generated by a phase field grain growth simulation and converted to interface conformal hexahedral and tetrahedral element meshes of variable resolution. Procedures for such interface-conformal mesh generation over complex shapes relying on Patran for tetrahedral and Cubit/Sculpt for hexahedral elements are described. CPFE simulations of simple tension and simple shear deformation conditions are performed. Minor sensitivity of models to these boundary conditions is observed. The computational time per CPU was measured to scale with the degrees of freedom for every element type. As the time per CPU scales with the number of degrees of freedom, linear elements are faster per CPU than the corresponding quadratic elements. However, hexahedral elements exhibit a better convergence rate than quadratic tetrahedral elements and arrive at a solution of equivalent accuracy to quadratic tetrahedral elements at less computational cost. Simulation results suggest that linear tetrahedral element is not appropriate for CPFE modeling of grain structures as these elements are overly stiff. Mesh refinement only moderately improve such simulation results. Results also show that quadratic brick elements are not suitable for large plastic straining of complex geometries due to their propensity to volumetric locking.

Similarly, mesh refinement only reduces the checkerboard pattern in pressure fields. The mesh resolution studies in capturing mechanical fields shows that brick elements are more sensitive than tetrahedral elements, which can be associated to geometric adaptability of tetrahedral elements. Geometry features can be lost with brick element coarsening. Moreover, tetrahedral elements are suitability for rapid and automatic meshing algorithms. In summary, quadratic tetrahedral and linear hexahedral elements are more accurate for crystal plasticity finite element modeling than linear tetrahedral and quadratic hexahedral elements. Furthermore, tetrahedral elements are more desirable due to fast mesh generation and flexibility to describe complex grain structure geometries. It is anticipated that the results from this study provide useful guidance for future CPFE modeling of grain structures. This guidance should be applicable to other crystal plasticity models as constitutive models and hardening formulations have minor effects on mesh sensitivity [2].

Function-based coarsening methods:

These methods are the best ways to simplify meshes. In the function-based methods, how to define a size function has an important role in the final mesh quality. Based on [13], a function called Sp(v) will be defined on the vertices u ε V of M**.** The function Sp(v) will approximate the local size of the mesh elements**.** The nearest neighbor's distance can be a good measure criterion for approximation [9]. A parameter β ensures that the distance of the function Sp(v) for the vertex v after the coarsening will be at least β times to the initial state. This method will use the following:

$$\beta > \beta_0$$

$$\beta_0 = \sqrt{2} \in twodimensions$$

$$\beta_0 = \sqrt{3} \in threedimensions$$

We say that a coarsened subset of the set of points *M, V^{FBC}* follow*C*s the distance condition if:

$$\beta\ (Sp(v_i) + Sp(v_j)) < dist(v_i, v_j) for all pairs (v_i, v_j) \epsilon V_C^{FBC}$$

Coarsening will either be performed on a specified number of steps or will be performed until the mesh reaches the maximum specified thickness. In this way, the quality of the generated mesh will be a limiting factor [5]. For example, the minimum aspect ratio should be limited by a constant.

A function-based method consists of the following steps:

- Define a initial mesh size function
- To form a new distance function, increase the size as needed
- Depending on the new function, remove the removable points, edges and elements
- Create a link between the points that were remained in the previous step

In related to the size function, we can say that, the set of points P by the function f will be spaced to β if:

$$p, q \in P, f(p_1) + f(p_2) < \beta \|p - q\|$$

And then the function f will be known as the spacing function of the set P of degree β [5, 13].

Defining the metric fields, on the fine mesh elements:

One of the main concepts in mesh computing is to define the metric fields on the mesh elements. The metric field specifications will ultimately be effective in calculating the size of the elements and the lengths of the edges. In the process of detecting removable points, in order to calculate the metric of the points, we first need to generate it on the cells. Generally, the metric fields are positive and symmetric positive matrices, and Their dimensions are equal to the dimension of the mesh space [6].

H-Shock:

The metric fields defined on mesh nodes can performs a good coarsening in normal areas of the mesh, but the problem occurs in an area called the H-Shock. If the mesh becomes a completely non-isotropic state to an isotropic state, it is called the H-Shock area.

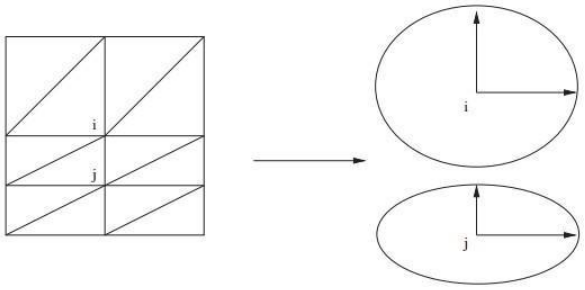

Figure 4: An image of a H-Shock event

On the left side, the area between the boundary layer and the isotropic mesh region and on the right side, the elliptical state of the metric field on the nodes i and j are shown [4]. As you can see, the radius of both ellipsoids is equal to x. Using the correction algorithm, the node j is detected in the boundary layer region and will be coarse only in y direction. While the node i is in the isotropic region, it will be coarse in both x and y axis directions.

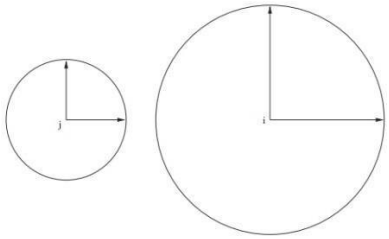

Figure 5: Comparison of the metric size of a node in isotropic and anisotropic regions

We find that a quick change in the mesh coarse is happening. The solution provided for solve this problem is called H-Shock correction.

Mesh Layering:

In the process of coarsening a boundary layer mesh, in order to preserve the overall shape of the anisotropic region of the boundary layer, the anisotropy and boundary layer must be detected and layered. Each layer must be identified with an integer number. The areas to be layered are the boundary layer around the airfoil and the wake area [4].

- At first, the anisotropic points of the mesh will be detected. Then the initial seed points will be identified**.**
- The initial layering will then be done and the points that are inaccurately and discretely layered will be removed from the layering.
- The boundary layer regions will be corrected, such that the points that are not even anisotropic, but are in a layer i that part of that layer contains anisotropic points, will be added to the layer i
- New seed points will be detected to begin the wake area layering
- Finally, the points of the wake area that are incorrectly layered will be removed from the layering.

Multi-grid methods on complicated unstructured meshes:

In [5], introduced a method for coarsening complicated unstructured meshes. The proposed method is a multi-grid approach. The purpose of this method is to simplify and increase the convergence rate of solving equations on unstructured meshes. In this method, a circle (or a three- dimensional sphere) is used around each of the mesh nodes. The radius of this circle is considered as a spacing function. At each time, only those nodes of the mesh are deleted that

their circles overlap. The circles around each node are completely separate at the beginning that will be coarse. By coarsening, those that are close enough to each other will overlap. so, the algorithm can delete them. The steps are shown in the following:

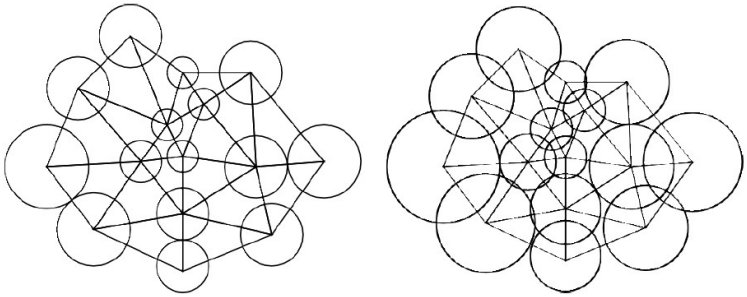

Figure 6: Distance of the nearest neighbor to each node before and after coarsening

In the above figure(left) we see a mesh with the initial circles. We see that in this case none of the circles are not overlapping with each other. In the approach used in this survey, the closest Euclidean distance to each node is used to create these circles. The right-hand figure shows the circles after coarsening the distance function. We find that there are a number of overlaps that can be removed from the mesh as follows.

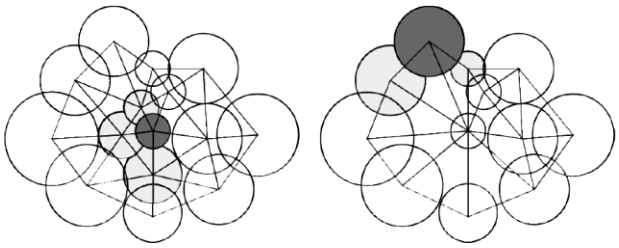

Figure 7: Deciding to remove one edge according to the size of the criteria

According to the above figure, we see that the central node of the mesh is selected as the primary node that remaining in the mesh, and in the right-hand side we see that the adjacent nodes that circles overlap with the central node circle have been removed from the mesh. Also, in the right- hand side, after the initial contraction in the mesh center, another node is selected to remain in the mesh. The node is marked with a gray color. Two nodes overlapping their circles with the selected node circle are also displayed using a pale gray color. In this method, the order of selecting nodes is very important. To arrange the nodes, we can use many factors such as boundary nodes, apex nodes or nodes in particular areas. This method cannot properly coarse meshes with an anisotropic boundary layer.

Automatic coarsening of anisotropic unstructured meshes:

This method is used to detect the removable elements of the meshes. In this algorithm, it is possible to simplify the high-anisotropy boundary layer meshes. This will be done by Metric Fields. Each of the metrics, expresses an ellipse around each of the mesh nodes by a rotation matrix. The defined ellipses, determines the direction and elongation of each element. In this method, coarsening is done in two ways. Coarsening in anisotropic regions will be based on the Metric Fields and only in the direction of the least elongation. This kind of coarsening is called semi- coarsening. In other isotropic regions of the mesh, coarsening is done by the metric fields and in all directions, which is referred to as full coarsening. After defining the metrics, the metric field coarsening will be done by the following algorithm.

An algorithm for coarsening metrics defined on the mesh nodes

Algorithm 2: Modified anisotropic coarsening algorithm
Input: $\lambda_1, \cdots, \lambda_d$ eigenvalues of a $\mathcal{M}_k$
Output: Update $\lambda_1, \cdots, \lambda_d$ coarsened eigenvalues
For each $i \in \mathcal{N}$
Set: $h_k^i = (\lambda_k^i)^{-1/2}, k = 1, \ldots, d, h_1^i \leqslant \ldots \leqslant h_d^i$
Set $h_0 = C_{CF} \cdot h_1$
Store the initial size: $h_{k,\text{old}}^i = h_k^i$
// compare the current node size specifications with its neighbors
if $\max\left(h_d^i, \min\left(C_{cf} \cdot h_d^i, h_{d-1}^i\right)\right) = C_{cf} \cdot h_d^i$
and $\exists j_0 < i \in V(i)$ such as
$h_1^{j_0} < h_1^i$. then $h_k^i = h_{k,old}^i, k = 1, \ldots d$
Define new eigenvalues $\lambda_k \leftarrow h_k^{-2}$
End for

This method is in the category of multi-grid methods and the final mesh that coarsened by this method has an acceptable quality, especially in boundary layers. In the metric fields, the inverse of the eigenvalue of each matrices, defines the size of the mesh in each direction. In this method, first, the metric field matrices will be defined on the mesh elements, and then they will be interpolated on the mesh nodes [4].

Define metric fields on the fine mesh elements:

The following formula will be used to define a metric on the mesh cells or elements.

$$M_T = C_M \cdot \left(\sum_{\substack{i,j=1 \\ i<j}}^{d+1} \vec{x}_{ij} \vec{x}_{ij}^T \right)^{-1}$$

In the above relation we have $C_M = \frac{d+1}{2}$ and the d parameter is equal to the dimensions of mesh. In this regard, the multiplication of the matrix $\vec{x}_{ij}$ with dimensions $d \times 1$ and $\vec{x}_{ij}^T$ with dimensions $1 \times d$ will create a matrix with dimensions $d \times d$. In front of the Σ mark, depending on the dimensions of the mesh, 1, 3 or 6 matrices are obtained for 1, 2, or 3 dimensional meshes. After summing the matrices together, the inverse of it will be multiplied by C_M. In this way, M_T tensor metric will be generated on the element T [6].

Metric fields interpolation on the fine mesh nodes:

To define a metric on a mesh, it will first be necessary to define a metric tensor for each element. Suppose Th represents a triangular mesh of a multi- dimensional domain Ω, which consists of a set of nodes called N and a set of two-dimensional or three-dimensional elements called T which that nodes are members of the set N. In the next step, the tensor metrics that generated on the mesh elements should be interpolated over mesh nodes. So as long as the tensors defined on the elements, on each point, define the length and direction of elongation of the points. The set *T (i)* is defined as: its members include T elements that are common to *i, i*ϵ*T*. The metric matrix on node i is calculated using the following equation:

$$M_i = \left(\frac{1}{card(T(i))} \sum_{T \epsilon T(i)} M_T^{\frac{-1}{2}}\right)^{-2}$$

$card(T(i))$ represents the number of members in collection $T(i)$. $M^{\frac{-1}{2}}$ is a function of a matrix, based on its diagonal form $M = V.D.V^T$ which is calculated as $M^{\frac{-1}{2}} = D^{\frac{-1}{2}} . V^T$ with the same definition for M^{-2}. The exponents of $\frac{-1}{2}$ and -2 re used to averaging the mesh size characteristics [7].

The division of mesh nodes and elements into two isotropic and anisotropic categories:

The nodes of a boundary layer model can be coarse in two isotropic and anisotropic methods. If the ratio of the maximum metric stretch size to the least size is greater than a threshold, then the node will be considered as a node that belongs to the anisotropic region of the mesh. The division of nodes into two isotropic and anisotropic categories can be useful in coarsening of different regions, and can also be used for layering on the boundary layer of anisotropic regions.

Metric correction for more accurate estimation in regions with high anisotropy:

The metric fields that are defined on nodes and cells, locally determine the size and direction of elongation of cells and elements. Experimental results show that under certain conditions, these metrics cannot maintain the characteristics of the mesh. For example, if the aspect ratio of the adjacent edges to the smallest mesh characteristics be more than a threshold (such as the thickness of the boundary layers). To solve this problem, after the metric fields are defined on the mesh nodes, it is necessary to correct the nodal metric eigenvalues. So that we can identify the smallest mesh characteristics, such as the thickness of the boundary layers and keep the changes in place.

To do this, based on [10], for the metrics of each node, we will to divide it into the form below, in which the matrix D is a square matrix with eigenvalues of the matrix on the main diameter arranged in ascending order. Also, V is an orthogonal matrix of eigenvector matrices of the main matrix. VT also is a transposition of the matrix V.

$$M_k = V_k . D_k . V_k^T$$

It is also explained in the article [10] that After decomposing the main matrix, we need to replace the main eigenvalues with a matrix of corrected eigenvalues. A new eigenvalue matrix for each vertex will be considered as *D* and will be calculated from the following equation:

$$\overline{D}_i = \left(\frac{1}{card(T)} \sum_{T_k \in T} D_{T_K}^{\frac{-1}{2}}\right)^{-2}$$

In this equation, T is a cell, and T_k is each of the cells connected to the processing point. Also, card (T) means the number of cells which connected to the point. After calculating the new corrected diagonal matrix, the final eigenvalues must be multiplied in the old eigenvectors.

$$M_i = V_i . \overline{D_i} . V_i^T$$

In the above equation, M_i means the node metric which to be processed. Also, *V I* is a matrix of old eigenvectors [4].

Using Unsupervised Segmentation Techniques:

The segmentation method that we use in this research is a unsupervised Segmentation Technique which itself includes different methods. This means that the segmentation of different areas is done automatically based on the metric fields. One can say that a simple unsupervised algorithm can build a model of the relevant data, where data points are independently and identically chosen from some distribution $P\ (\ X\ )^2$. In the field of mesh segmentation, unsupervised learning is one of the methods used [1]. According to the research done in [1] Our chosen method is a partition-based method.

Considering the center of the data points as the center of the corresponding cluster is the

fundamental idea of this kind of clustering algorithm. Since clustering depends on the center of data points, partition-based methods are called centroid-based methods [1].

In [1] research, various algorithms have been mentioned that we can use all of these algorithms. The only difference is in the size criterion chosen.

H-Shock Correction:

In the previous sections, the concept of H-Shock was explained. In this section, we can use the following algorithm to correct the H-Shock problem.

- Until the h-shock for one edge is greater than the threshold β, repeat it:
  - Suppose that we are processing a PQ edge
  - Calculate $h(P)$ (and so $h(Q)$) size function in the vicinity of point P (And so Q) in the direction $\overrightarrow{P^{\rightarrow\rightarrow\rightarrow}Q^{\rightarrow\rightarrow\rightarrow}}$ (assuming $h(P) \leq h(Q)$)
  - Calculate $l(PQ)$; as the length of the PQ, in the metric space
  - Calculate $c(PQ)$; as h-shock value in the direction of the $\overrightarrow{P^{\rightarrow\rightarrow\rightarrow}Q^{\rightarrow\rightarrow\rightarrow}}$ edge
  - If $c(PQ) \geq \beta$, then the metric $M_d^{(Q)}$ will be replaced by $\frac{M_d(Q)}{\eta^2}$, where $\eta = \left(\frac{\beta}{c(PQ)}\right)^{l(PQ)}$

Given the necessity of calculating the function size h, in the vicinity of a point and in the direction of the edge connected to it, using the generic metric definition, we can calculate the following equation.

$$h\left(\vec{x}, \vec{v}\right) = \frac{\sqrt{\vec{v}^T \times \vec{v}}}{\sqrt{\vec{v}^T \times M_d(\vec{x}) \times \vec{v}}}$$

This equation returns the metric size adjacent to a point in the direction of a connected edge to it. In this function, x is the coordinate of the point in Euclidean space and v is a unit vector in the direction of the desired edge, Also, M is a metric field associated with the point [4].

Layering in the boundary layer regions:

For each node i, we can define an index $L(i)$, which specifies the number of the layer. This means that the point i belongs to a layer with index (i). As an example in the figure below, a simple layering is specified on the nodes of a mesh.

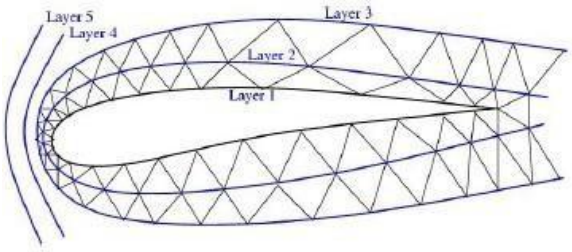

Figure 8: An example for boundary layer layering of a mesh

For layering of the vertices of a mesh, we can use the connection between points, First, we choose a layer to get started, which is sure that its points belong to a common layer. To do this, we can choose the internal boundary nodes on the airfoil and consider all of them in a specific layer. Then, depending on the chosen start layer, the rest of the nodes can be layered.

In this survey, we consider the stretched boundary points to identify the first points of the seed, and consider all the boundary points that connected to these points as seed. This set of points will be Layer #1.

Table 1: Initial mesh layering steps

	Basic Mesh Layering Steps
1	Detecting Stretching Nodes
2	Detecting Seed Nodes
3	Perform initial layering
4	Remove discrete layer Indexes from Layering System
5	Correcting Boundary Layer Node Layer Indexes
6	Re-Define Seed Nodes
7	Perform Layering Again
8	Remove discrete layer Indexes from Layering System Again

Coarsening the metric fields defined on the initial fine mesh:

The metric field defined on the initial fine mesh must be coarse enough to select the appropriate elements to be removed and be able to check the distances between nodes on each edge.

The coarsening algorithm works as follows:

For all nodes $i = 1, \ldots, n$ belonging to the main fine mesh, we assume that the metric M_i is defined on these nodes. Coarsening will be done in the following way:

- M_i will be decomposed to its eigenvalues in descending order, $\sigma(M) = \{\lambda_1, \ldots, \lambda_d\}$, $\lambda_1 \geq \cdots \geq \lambda_d$, and eigenvectors corresponding to each eigenvalue $\{\vec{v}_1, \ldots, \vec{v}_d\}$; d is equal to the dimensions of the space and accordingly equal to the tensor dimensions (for example, in the 2-dimensional tensor space we have 2 × 2 matrices).
- Define the size function according to $h_k = \lambda_k^{\frac{-1}{2}}$ $k = 1, \ldots, d$, $h_1 \leq \cdots \leq h_d$ and also $h_0 = C_{CF}.\, h_1$
- For $k = 1, ..., d$, the following steps will be done:
  - Change the metric specifications as:
    $$h_k \leftarrow max\,(h_k, min\,(C_{CF}.\, h_k, h_{k-1}))$$
  - Define new eigenvalues as $\lambda_k \leftarrow h_k^{-2}$
  - Calculating the coarsened metric matrix by multiplying the new eigenvalues $\{\lambda_1, ..., \lambda_d\}$ in the old eigenvectors $\{\vec{v}_1, ..., \vec{v}_d\}$

In the anisotropic regions, the coarsening will only be done at the minimum stretching direction, while in isotropic regions, coarsening will be done in all directions [4].

Identify removable edges and edge contraction:

A function f, on the domain Ω is $1 - lipschitz$, if for every x, y belongs to Ω:

$$|f(x) - f(y)| \leq \|x - y\|$$

The set of points P are called spaced if:

$$p, q \epsilon P, f(p_1) + f(p_2) < \beta \|p - q\|$$

In this way, the function f is known as the spacing function of the set P. Suppose $\beta > 1$ is a real number and f is a $1 - lipschitz$ function. A set of points P, are $f^{\beta} - spaced$ [5]. If for every point p, q have:

$$\frac{f(p)}{\beta} + \frac{f(q)}{\beta} < \|p - q\|$$

In this research, in the isotropic regions, the nearest neighbor function is used as a criterion. Also in the anisotropic regions, the ellipse radius defined by the metric fields is used as the size function.

The proposed algorithm, in brief:

The process steps of the proposed algorithm are briefly described below:

- Get the input mesh data
- Identify the edges that can be deleted from the mesh
    - Identify mesh boundary layer points
    - Define metric fields on the mesh points
        - Define metrics on the mesh cells
        - Interpolation of cell metrics, on the mesh points
        - Correcting eigenvalues for the point metrics
        - H-Shock correction of the mesh point metrics
        - Coarsening the final metric sizes
    - Mesh Layering
        - Identify the points that belongs to the anisotropic regions
        - Identify the seed points for layering
        - Perform the basic layering
        - Remove points and layers that are discretely layered from set of layered points
        - Complete the layers that are not completely layered
        - Define new seed points for wake region layering
        - Repeat layering operation for wake region layering
        - Remove points and layers that are discretely layered from set of layered points
    - Sorting the mesh points based on the defined layer indexes
    - Define a criterion for coarsening the isotropic regions (half of the nearest neighbor distance)
    - Identify removable mesh nodes
        - Select a point (P1)
        - Identify the point(P2) that is in the minimum stretch direction of the P1
        - Checking the $\overrightarrow{P1}\overrightarrow{P2}$ edge removal capability based on defined criteria
        - Controlling the amount of possible coarsening
        - Store P1 and P2 as the points of the beginning and the end of the removable edge
        - Update the modified mesh connections
    - Removing the detected removable edges from the mesh by the edge contraction method
        - Select the edges that were detected in the previous steps as removable edges
        - Check whether the mesh has a negative volume or not
        - Checking invalid cell creation
        - Remove the selected edge
            - Identify the points and edges that are adjacent to the selected edge
            - Deletion or deformation the adjacent cells of the selected edge
            - Deforming or removing the removable faces
            - Updating mesh information about the edges of each region
    - Update the mesh connectivity graph data
        - Removing the deleted edges of each region from the entire mesh data
        - Update the mesh connectivity matrix and also general information about it
    - Rewrite the mesh data in an output file

Accuracy of the algorithm in detecting the boundary layers and anisotropic regions:

One of the works that must be done in mesh coarsening algorithms is to identify the anisotropy of the points and elements, but this is not enough to achieve a good precise mesh. Usually in anisotropic regions of a mesh, the points and elements are in the form of a layered structure [11], therefore, after the coarsening operation, the structure of each layer should be maintained as much as possible. As the number of nodes identified as anisotropic nodes differ by the number of layered points, it means that layers may be deformed in the main mesh. in this case, the simplified mesh may not have acceptable accuracy compared to the fine mesh. In this way, we can use it as a criterion for evaluating the results.

In the previous sections we said that a mesh in anisotropic regions would be simplified only at the least stretching direction; Therefore, in the anisotropic regions and the boundary layers of the mesh, in the other directions, it is not permissible to simplify. Also, due to the fact that in an appropriate simplification the layers of the boundary layer will be eliminated in a one-to-one order, we can expect that half of the points belonging to the boundary layers be eliminated after the simplification.

In [4], it has been explained that in boundary layer meshes, layering can be done more accurately by dividing the boundary layer into several different classes. In this study, we will use this technique to evaluate the method. we will divide the mesh nodes into several classes in terms of dimensions and metrics, and we will examine the degree of coarsening done in each class. As we said, the mesh in the anisotropic regions will only be coarsened in a minimum stretching direction; while in other regions it will be coarse in all directions. Therefore, at each step of a proper coarsening process, the aspect ratio of the elements cannot be increased. Regarding this issue, we use the following equation to examine the number of nodes allowed to be in each class after coarsening:

$$Max_{C_x} = RN_{C_x} + \sum_{i=x+1}^{Max(C)} (TN_{C_i} - CN_{C_i} - RN_{C_i})$$

In the above equation, RN is the number of remaining nodes, CN is the number of simplified nodes in the class i after the coarsening, and also TN is the total number of nodes in the class i before doing the coarsening. In summary, the above equation ensures that a new node will not be created in the anisotropic regions [10]. As a result, in order to evaluate the accuracy of the algorithm in identifying the anisotropic regions and the boundary layers, we can examine the following criterion: Total number of nodes which identified as anisotropic must be greater than or equal to the sum of total number of the nodes in the boundary region. In the following, we see the calculated data for a sample mesh:

Table 2: Calculation of the data for a sample mesh:

	Class C_3	Class C_2	Class C_1
Number of class nodes in the fine mesh	520	4049	1531
The number of nodes that have been removed from the class after coarsening	260	1705	215
Number of class nodes in the coarse mesh	192	1950	1316
Maximum number of nodes in each class in the coarsened mesh	**260**	**2018**	**1778**

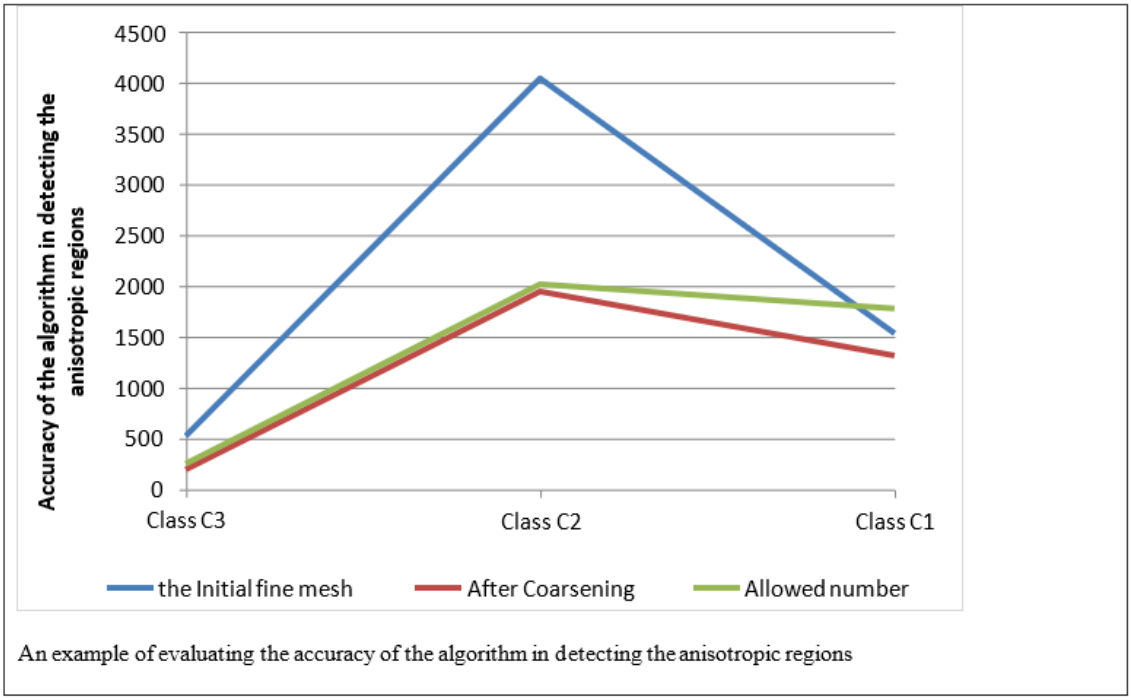

Figure 9: Example of evaluating accuracy of algorithm in detecting the anisotropic

In this example, the nodes of a mesh are divided into three distinct classes. The C_3 class contains the most stretching nodes and the C_1 class contains nodes with the least stretching. We can see that the number of nodes in each class after coarsening is less than the number of the fine mesh nodes and also lower than the allowable number of nodes in each class.

Evaluating the geometric quality of mesh elements after coarsening:

This type of evaluation consists of two parts. The first part is a geometric quality calculation of all the coarsened mesh elements. Geometric quality is calculated individually for each element of the coarsened mesh. it is a function of the mesh element geometry and the corresponding metric fields of the fine mesh and shape, volume and the size of the mesh elements. For evaluating the quality of the elements, aspect ratio changes can be used. In this way, the aspect ratio of the largest metric size of a point to its minimum after coarsening should be greater than or equal to the coarse metric aspect ratio and smaller than the ratio of the initial fine metric. In other words, this condition can be used to evaluate the following two states:

- The mesh in anisotropic regions should be coarse along the least stretching direction.
- the coarsening in the vicinity of each point should be done in a controlled manner. In other words, the mesh size adjacent to any point should not exceed the limit specified by its metric fields.

The second criterion can examine the overlapping of the coarse mesh elements. By examining the overlap of the elements, we can examine the invalid states of a mesh, such as negative volumes, or change of the boundaries. One of the important issues in evaluating by this method is to detect the threshold or the limit of allowed changes in the coarse mesh rather than the fine mesh. Removing the boundary points can reduce the overall volume of the mesh. Also, given that apex points cannot be removed from the mesh. the volume that can be lost by deleting all non-apex points from the mesh can be considered as the maximum amount of volume variation allowed. In the boundary faces, the points can only contract toward one of the other boundary points, so the second point will definitely remain in the mesh. Therefore, it can be concluded that the maximum boundary points that can be excluded from the mesh can be calculated from

the following equation:

$$maxCB = \frac{BoundryNodes - appexes - folds}{2}$$

In the above equation, $maxCB$ is the maximum number of boundary nodes that can be removed from a mesh, $BoundryNodes$ are the total number of nodes in all regions, $appexes$ are the total number of nodes and $folds$ are the number of boundary curve points. We can only use the number of boundary points that have been eliminated to evaluate the method. we can also calculate the maximum volume that can be lost from the mesh by coarsening.

$$maxLostVol = \sum_{i=1}^{maxCB(I)} (lostAppexesVol(maxCB(i)))$$

In the above equation $lostAppexesVol(maxCB(i))$ is the maximum volume that is allowed to reduce from the mesh when deleting the $maxCB(i)$ point. Clearly, at least changes are zero. This means that in a proper simplification process, volumes cannot be added to the mesh. According to the above, adding a volume can mean an overlap between the elements, which is an invalid state in the mesh. In the following, how to set the threshold for volume changes will be explained as an example.

Table 3: General Specifications of a fine mesh

Specifications	Mesh M1
Number of boundary nodes	500
Number of apex nodes	20
Number of boundary curve nodes	50
Sum of element volumes in Euclidean space	15
Average length of the boundary edges in Euclidean space	1

As the first case, we need to calculate the amount of lost volume by deleting a non-apex node in the mesh. The coordinates of one of the non-apex points of the mesh and the coordinates of its adjacent points are as follows. Note that for simplicity in understanding the subject, it is assumed that only two edges are connected to the apex point.

Point coordinates: P_a (−5,10,0), P_1(−4.3,11,0), P_2(−5,9,0)

Therefore, the lost volume is a triangular shape between the three edges:

$Edges = \overrightarrow{P\overrightarrow{a}}, \overrightarrow{P\overrightarrow{1}}, \overrightarrow{P\overrightarrow{a}}, \overrightarrow{P\overrightarrow{2}}, \overrightarrow{P\overrightarrow{1}}, \overrightarrow{P\overrightarrow{2}}$

At this point, we have the points and their coordinates, so we can calculate the maximum volume that can be lost by the contraction of each of the boundary nodes. To calculate the volume according to the method presented in (Robert Nurnberg) we use the following equation:

$$Volume = \frac{1}{2} \sum_{i=1}^{Edges=\overrightarrow{P\overrightarrow{i}}, \overrightarrow{P\overrightarrow{i+1}}} X(P_i)Y(P_{i+1}) - X(P_{i+1})\,Y(P_i)$$

In the three-dimensional meshes, the following equation will be used:

$$Volume = \frac{1}{6} \sum_{i=1}^{Face=a_i b_i c_i} a_i . n\hat{} _i$$

In the above equation $n\hat{}_i$is calculated by the following tensor multiplication:

$$n\hat{}_i = (b_i - a_i) \otimes (c_i - a_i)$$

According to the data, the maximum volume that can be lose from the mesh, in case that we remove the point P_a will be calculated as follows:

$$lostAppexesVol = -12 + 5 + 16.3 = 9.3$$

So, the number of removable boundary points is equal to the following value:

$$maxCB = \frac{500 - 20 - 50}{2} = 215$$

The calculation of the maximum lost volume should be done for all 215 points to achieve the maximum removable volume.

RESULTS

To ensure good representation of the fluid-solid interface or the existing mesh has to be allowed to deform to track the moving geometry [10]. In this section to evaluating the proposed method, the results of the algorithm will be examined on multiple meshes, along with their initial and the coarse states.

A hybrid mesh around an airfoil a with boundary layer region and a wake area:

This mesh has an anisotropic boundary layer region and a wake area. Other areas are isotropic. The elements of the boundary layer section are in the shape of a quadrangle.

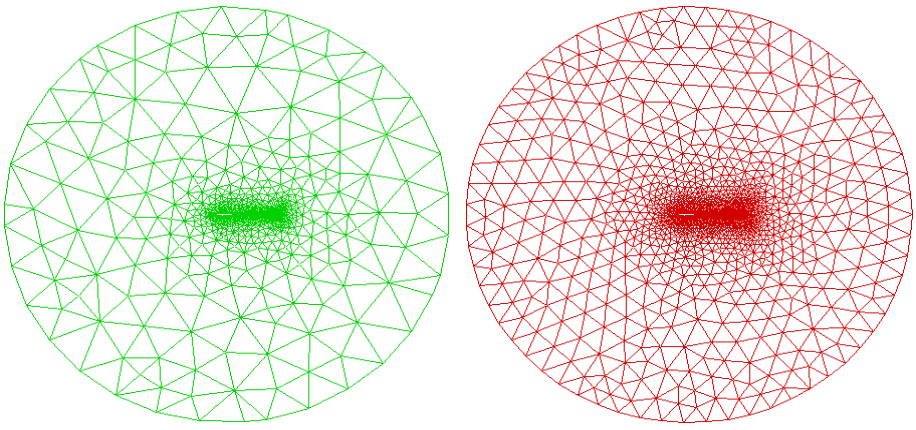

Figure 10: The mesh (far view) in fine mode(red) and after coarsening(green)

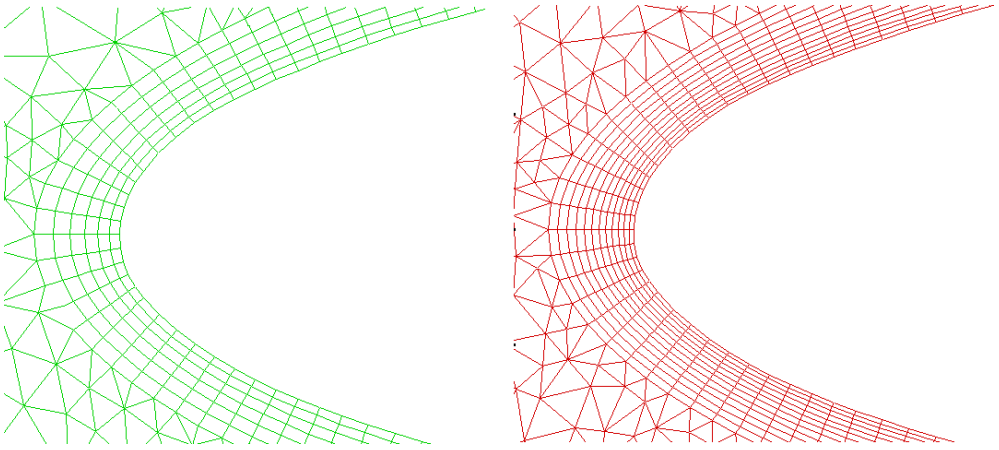

Figure 11: Part of the boundary layer in the fine mesh(red) and after coarsening(green)

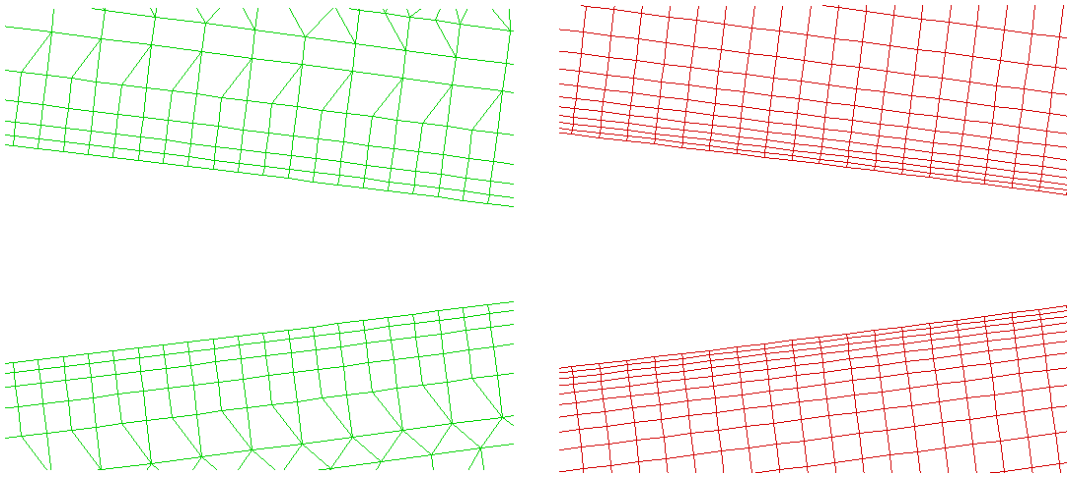

Figure 12: Part of the boundary layer in the fine mesh(red) and after coarsening(green)

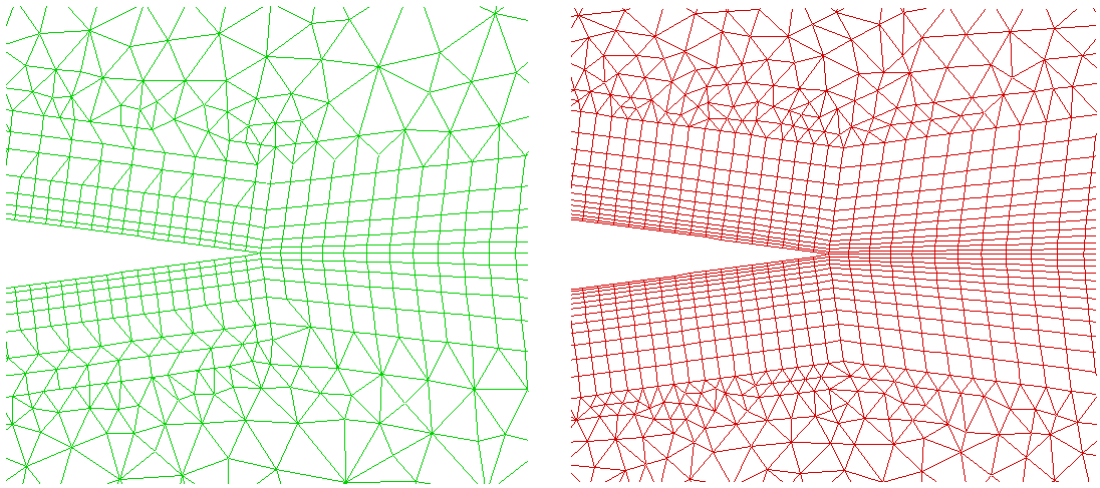

Figure 13: The wake area and boundary layer of the mesh in the fine mesh(red) and after coarsening(green)

Table 4: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	16459	7612
Number of Faces	28436	13306
Number of Regions	3	3
Interior faces of zone FLUID	27949	12848
Faces of zone AIRFOIL	424	424
Faces of zone INLET	63	34

Description:

- As you can see in the figure, anisotropic regions are coarsened to a minimum stretching direction. While the isotropic regions are coarse in all directions. The overall shape of the layers is also preserved.
- In this mesh, smoothing is also performed 2 times.
- Average program run time: 10 seconds

A mesh around an airfoil without the anisotropic boundary layer:

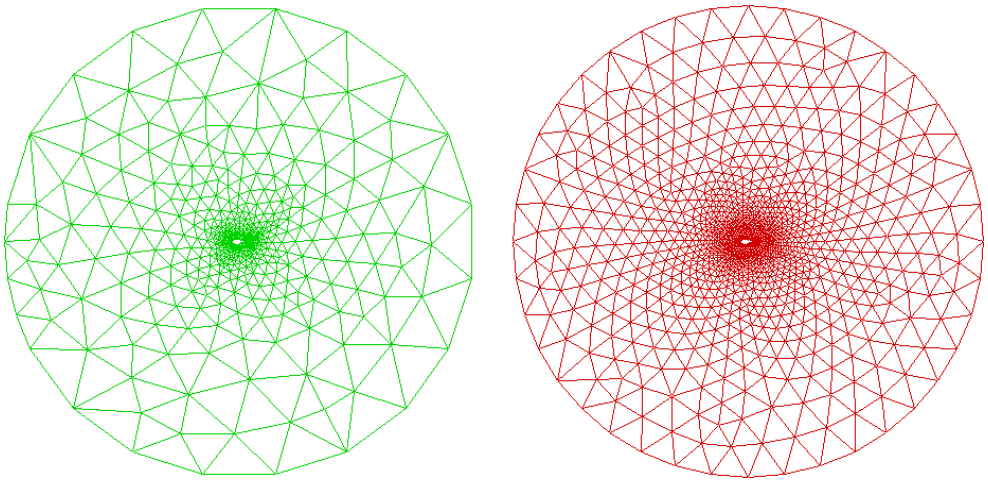

Figure 14: The mesh (far view) in fine mode(red) and after coarsening(green)

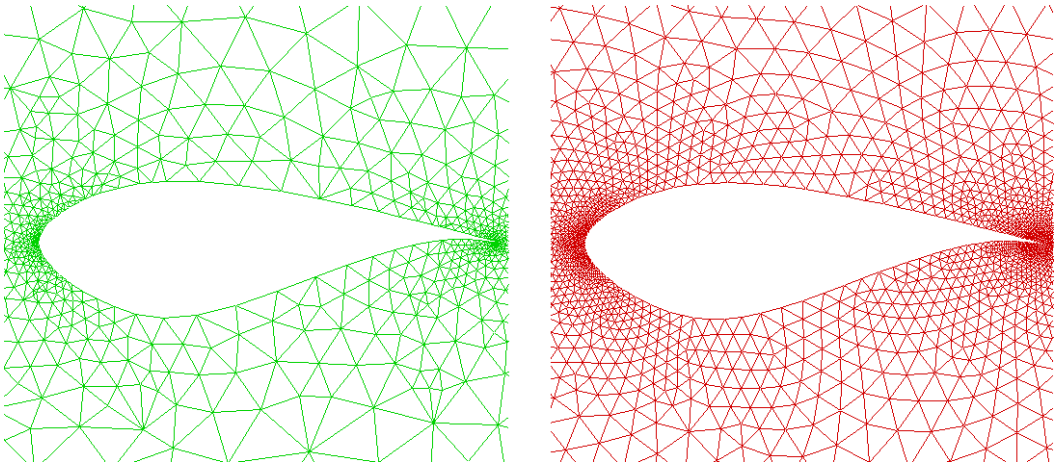

Figure 15: The mesh (far view) in fine mode(red) and after coarsening(green)

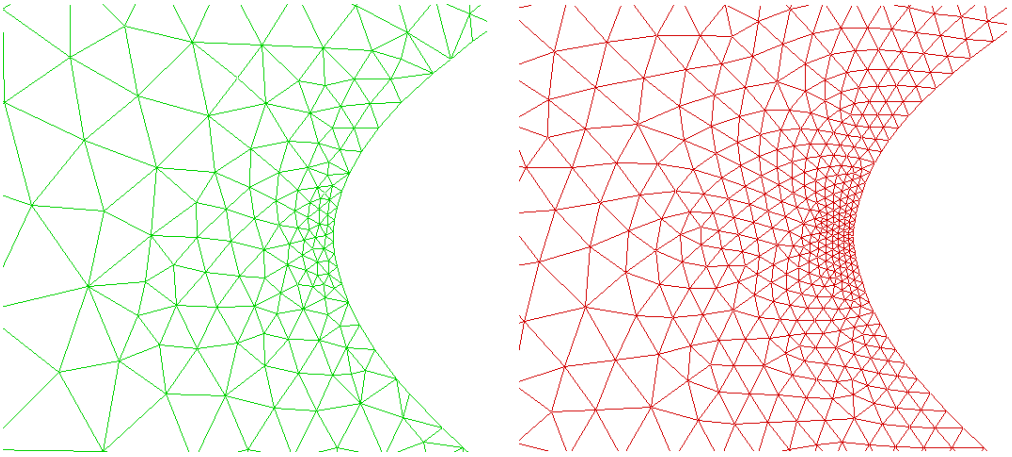

Figure 16: Part of the fine mesh(red) and after coarsening(green)

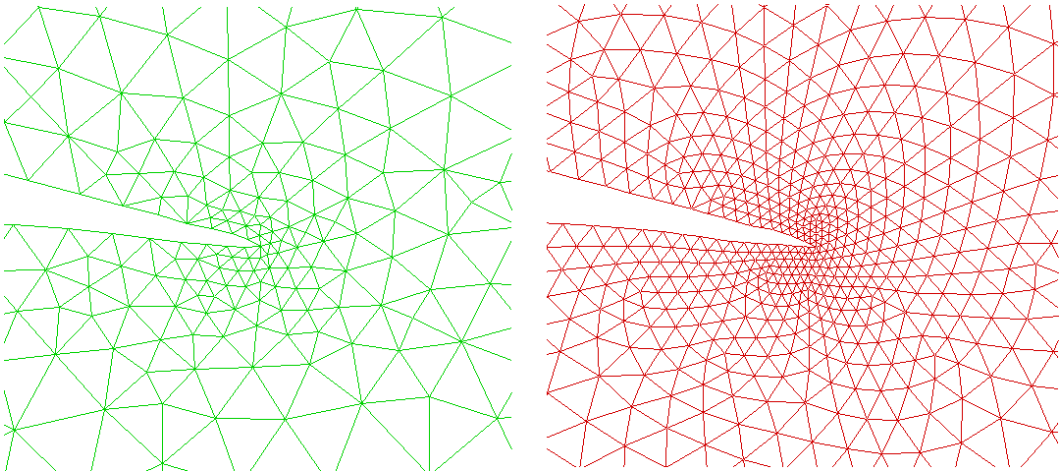

Figure 17: Part of the fine mesh(red) and after coarsening(green)

Table 5: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	6824	2235
Number of Faces	10332	3406
Number of Regions	4	4
Interior faces of zone DUCT	10140	3301
Faces of zone DU30 Low	76	40
Faces of zone DU30 Up	76	43
Faces of zone INLET	40	22

Description:

- In this mesh, smoothing is also performed 1 time.
- Average program run time: 10 seconds

A triangular isotropic mesh without any anisotropic boundary layer:

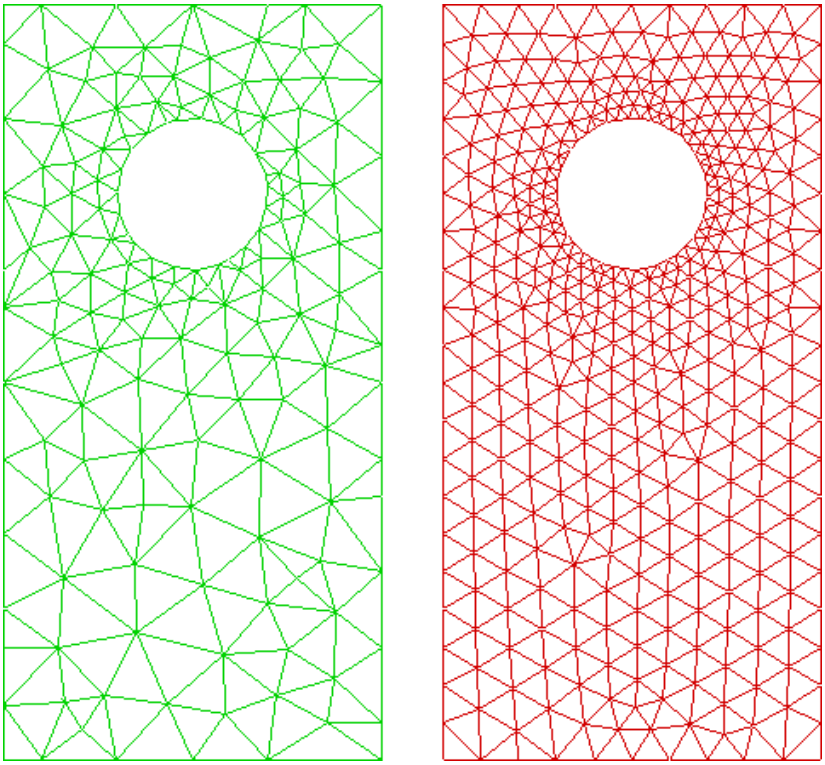

Figure 18: The mesh (far view) in the fine mode(red) and after coarsening(green)

Table 6: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	792	270
Number of Faces	1239	433
Number of Regions	3	3
Region 1 Faces	60	31
Region 2 Faces	42	23
Region 3 Faces	1137	379

Description:

- In this mesh, smoothing is also performed 1 time.
- Average program run time: 5 seconds

A hybrid mesh which has 3 distinct airfoils and anisotropic boundary layers

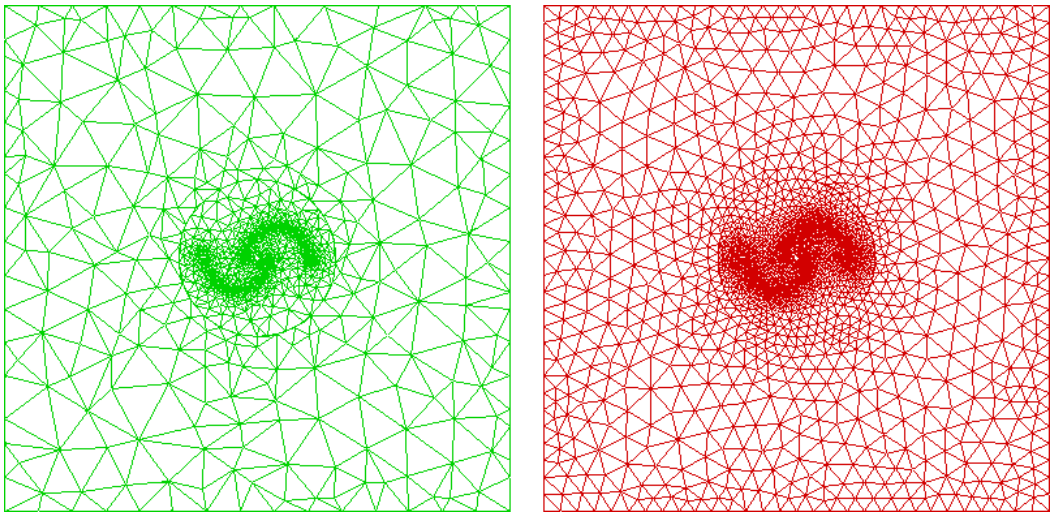

Figure 19: The mesh (far view) in the fine mode(red) and after coarsening(green)

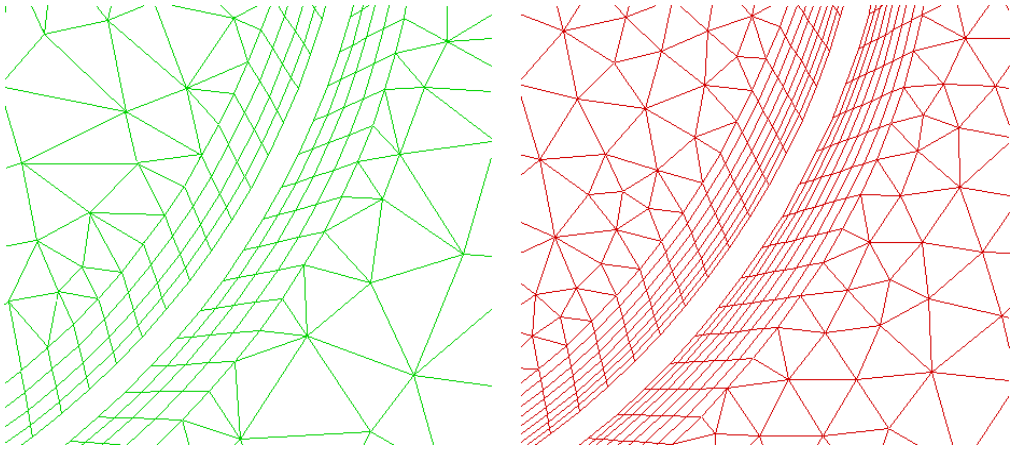

Figure 20: Part of the boundary layer in the fine mesh(red) and after coarsening(green)

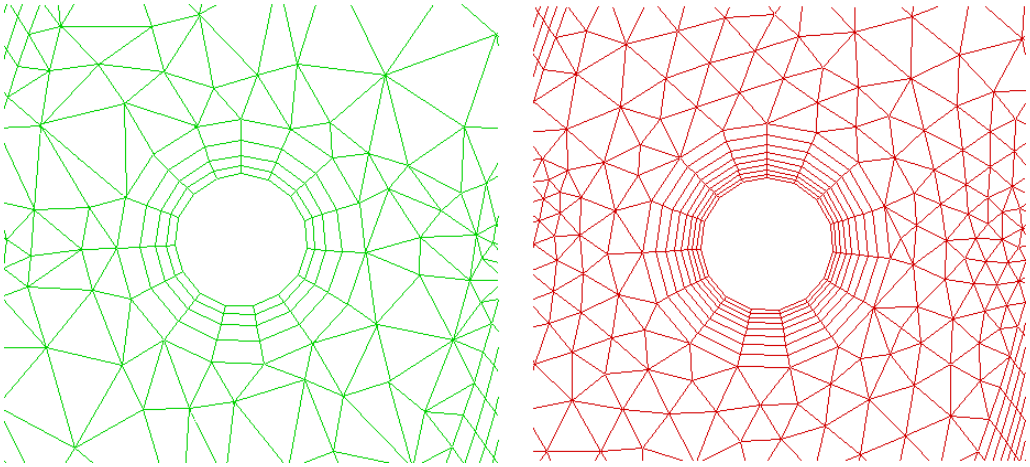

Figure 21: Part of the boundary layer in the fine mesh(red) and after coarsening(green)

Table 7: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	7031	2457
Number of Faces	14483	6786
Number of Regions	11	11
Interior faces of zone FACE.4	11800	5605
Interior faces of zone FACE.5	2071	682
Faces of zone WALL-OBSTACLE	13	7
Faces of zone VELOCITY-INLET	30	15
Faces of zone VELOCITY-INLET 2	60	33
Faces of zone OUTLET	30	17
Faces of zone INTERFACE1	50	27
Faces of zone INTERFACE2	62	33
Faces of zone WALL-FACE.4	15	15
Region 10 Faces	176	176
Region 11 Faces	176	176

Description:

- As seen in the figure, this mesh has three airfoils, which have their own boundary layers. We see that all three boundary layers are coarse.

Coarsening in other regions has been done in an isotropic state and has been done in all directions.

- In this mesh, smoothing is also performed 3 time.
- Average program run time: 15 seconds

A triangular boundary layer meh:

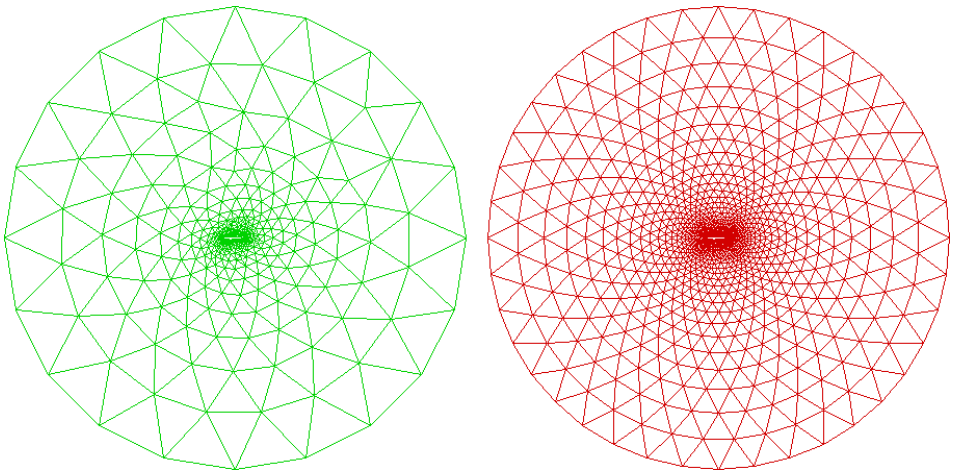

Figure 22: The mesh (far view) in the fine mode(red) and after coarsening(green)

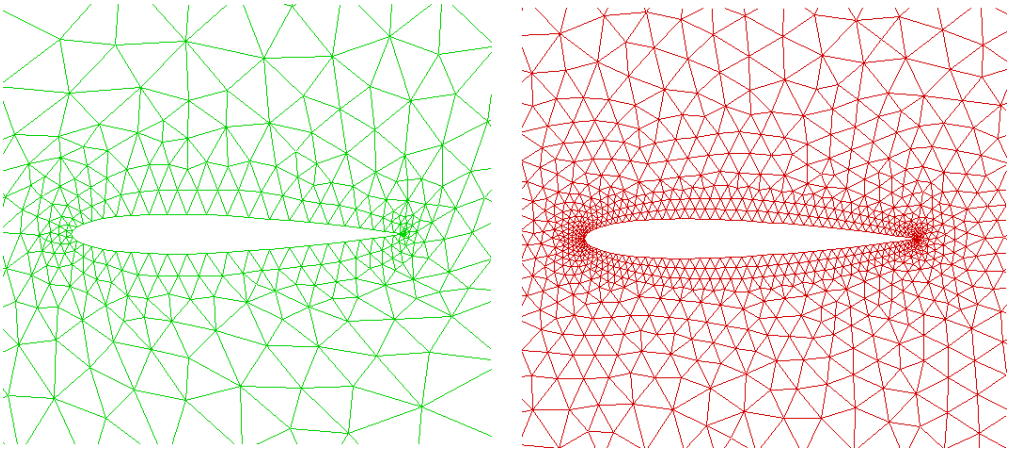

Figure 23: Part of the boundary layer in the fine mesh(red) and after coarsening(green)

Table 8: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	3218	940
Number of Faces	4897	1445
Number of Regions	3	3
Interior faces of zone FACE.4	100	50
Interior faces of zone FACE.5	40	20
Faces of zone WALL-OBSTACLE	4757	1375

Description:

- In this mesh, smoothing is also performed 1 time.
- Average program run time: 10 seconds

A hybrid mesh around an airfoil (least stretching in the z axis direction):

This mesh has an anisotropic boundary layer and an anisotropic wake area. Other regions are isotropic. This mesh has been extruded from a 2D mesh in z direction. Its minimum stretching is in the z axis direction. Therefore, it is expected that the anisotropic regions will be coarse in the direction of the z axis. The elements of the boundary layer regions of this mesh are hexagonal and elements of other regions are in the form of prism cells.

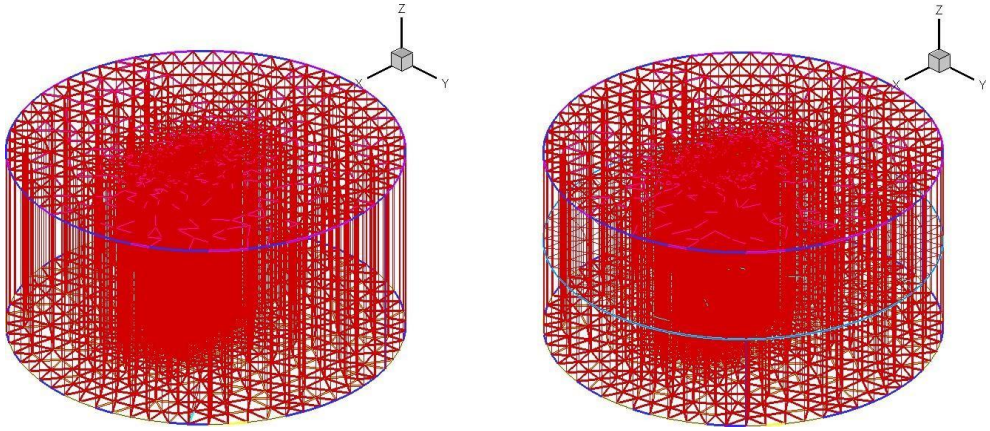

Figure 24: The far view of the mesh in the fine state(right) and after the coarsening(left)

We see in the above figure that there is a middle layer in the fine mesh. The least stretching of the mesh is in the direction of the middle layer(z-axis). Therefore, the mesh will be coarse in that direction. we see that in the left-hand side (after coarsening) the middle layer is removed.

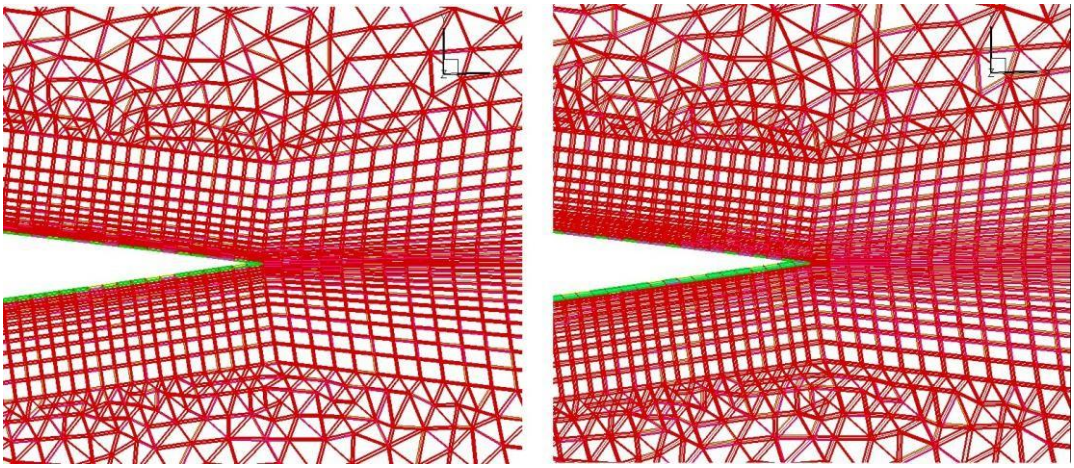

Figure 25: Part of the boundary layer in the fine mesh(right) and after coarsening(left)

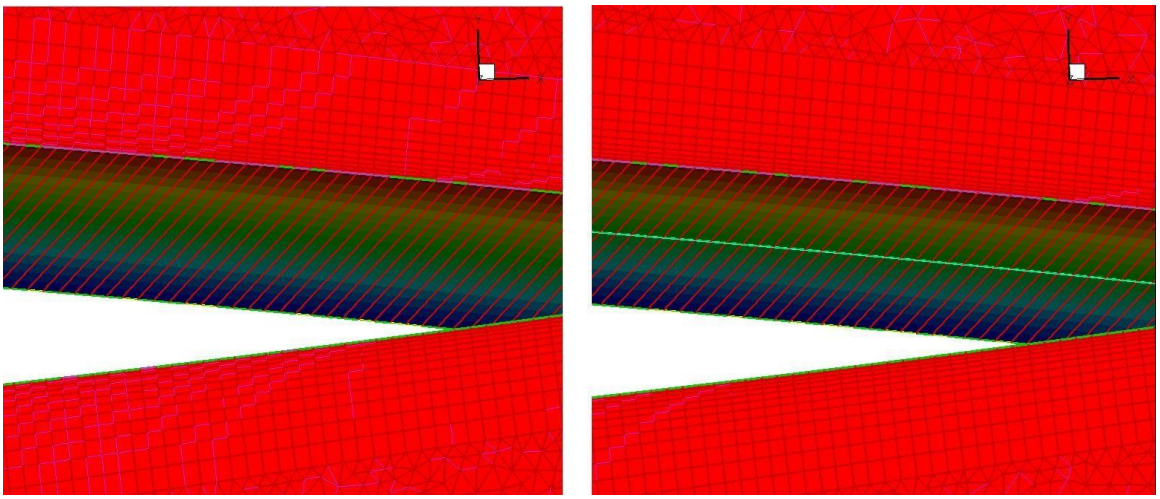

Figure 26: Part of the boundary layer in the fine mesh(right) and after coarsening(left)

To make it clearer, we have used Tecplot's Contour mode. We see that the midpoints of the mesh, which appear in the right figure in the form of a green line, are removed in the left-hand side.

Table 9: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	32918	16556
Number of Faces	106249	61557
Number of Regions	6	6
Faces of Region1	55898	28053
Faces of Region2	848	426
Faces of Region3	126	63
Faces of Region4	16459	97
Faces of Region5	16459	16459
Faces of Region6	16459	16459

Descriptions:

- As shown in the figure, the middle layer is completely deleted from the mesh.
- Average program run time: 10 seconds

A hybrid mesh around an airfoil (least stretching in the direction of x, y):

This mesh is the same as the previous mesh, with a difference in stretching direction which have the least stretching in the direction of x and y axis. Therefore, it is expected that the coarsening operation will be done in the x or y axis direction. This mesh has a boundary layer and a wake area.

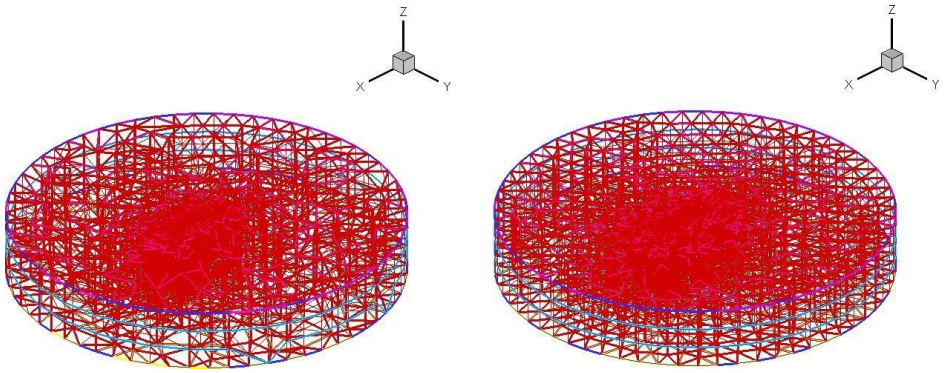

Figure 27: The far view of the mesh in the fine state(right) and after the coarsening(left)

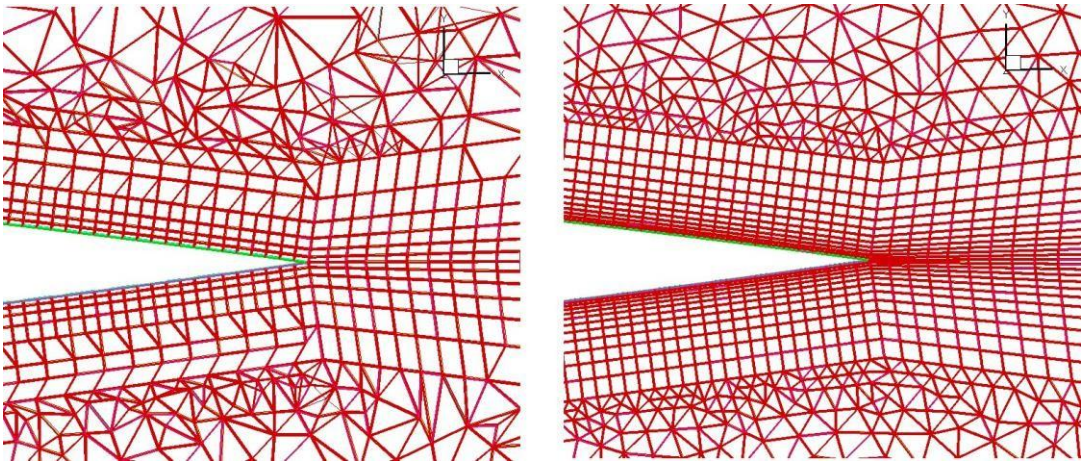

Figure 28: Part of the boundary layer in the fine mesh(right) and after coarsening(left)

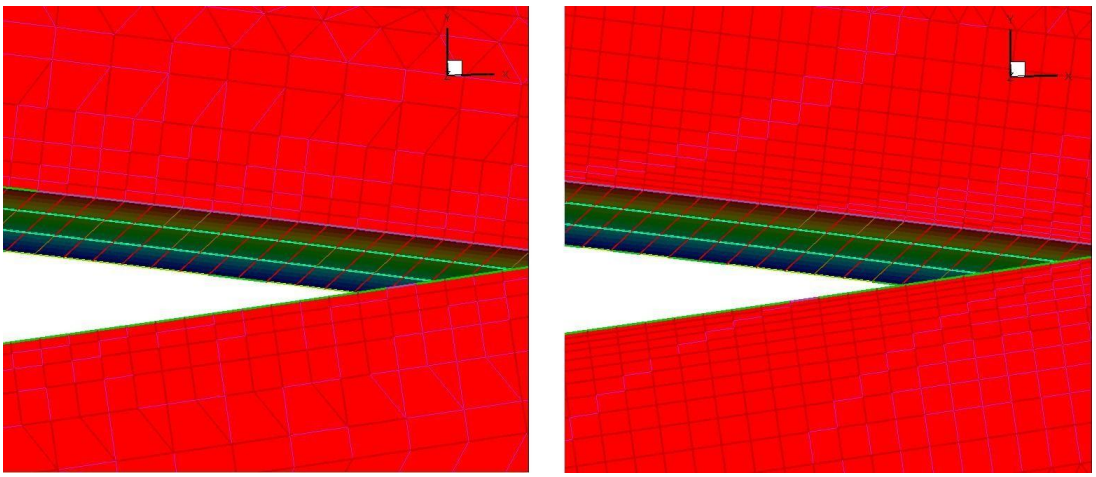

Figure 29: Part of the boundary layer in the fine mesh(right) and after coarsening(left)

The above figure shows some of the midpoints in the mesh boundary layer area. The right figure is the initial fine mesh and the left side figure is the coarse mesh.

Table 10: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	49377	28130
Number of Faces	151144	84676
Number of Regions	6	6
Faces of Region1	83847	46294
Faces of Region2	1272	1272
Faces of Region3	189	170
Faces of Region4	32918	18517
Faces of Region5	16459	9099
Faces of Region6	16459	9324

Descriptions:

- This mesh was created by extruding a 2D mesh in the z axis direction.
- Average program run time: 10 seconds

A 3D mesh within a hemisphere:

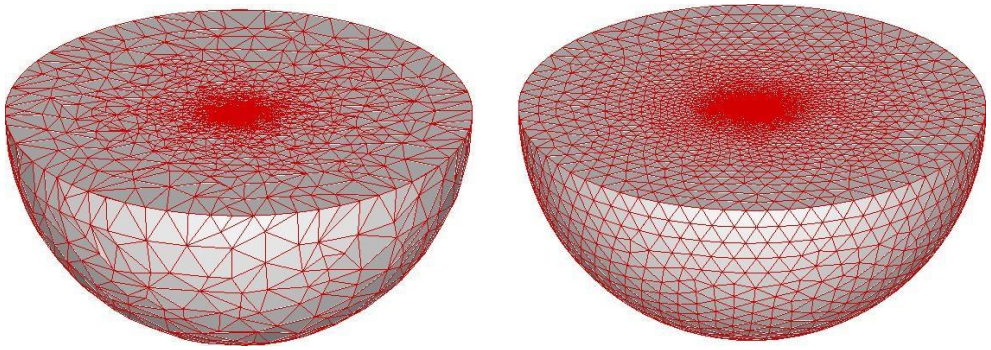

Figure 30: The far view of the mesh in the fine state(right) and after the coarsening(left)

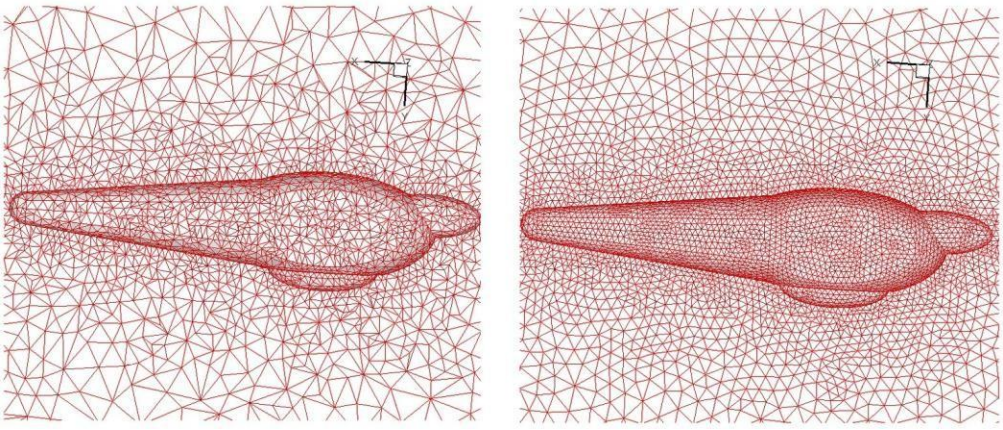

Figure 31: The far view of the mesh in the fine state(right) and after the coarsening(left)

Table 11: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	385175	184357
Number of Faces	777619	372701
Number of Regions	5	5
Interior faces of zone INTERIOR	763131	364787
Faces of zone WALL-INTERIOR	83	83
Faces of zone INLET	9138	5017

Faces of zone SOLID2	3679	2038
Faces of zone SYMMETRY	1588	776

Descriptions:

- before to the contraction operation, 49,854 edges were selected to be removed
- Average program run time: 20 seconds

A 3D mesh within a hemisphere:

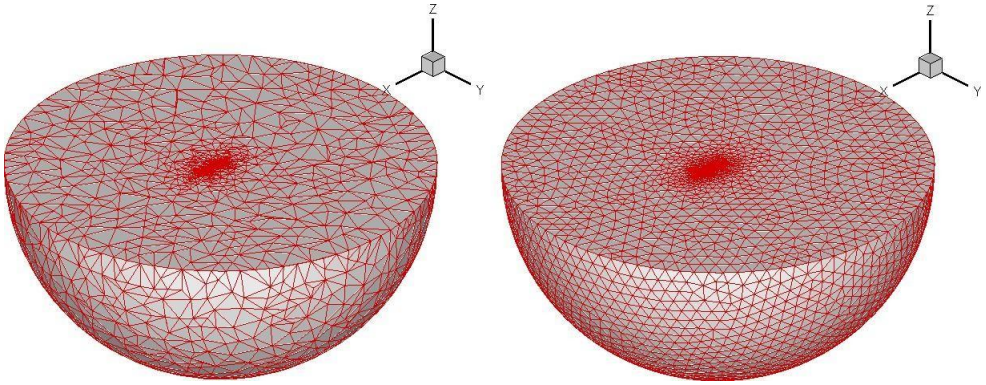

Figure 32: The far view of the mesh in the fine state(right) and after the coarsening(left)

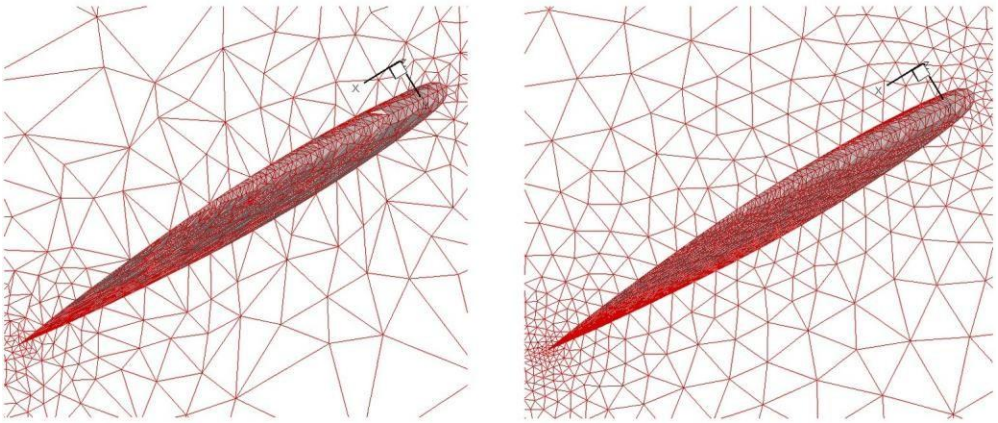

Figure 33: close image of the top view of the mesh in the fine state(right) and after the coarsening(left)

Table 12: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	281966	139701
Number of Faces	670624	283269
Number of Regions	5	5
Interior faces of zone SOLID	557240	275543
Faces of zone WALL-SOLID	260	198
Faces of zone CIRCLE	2939	1696
Faces of zone SEMI_SPHERE	3349	1640
Faces of zone ONERA_M6	6836	4192

Descriptions:

- before to the contraction operation, 67,387 edges were selected to be removed
- Average program run time: 15 seconds

A hybrid mesh around an airfoil (least stretching in the direction of z axis):

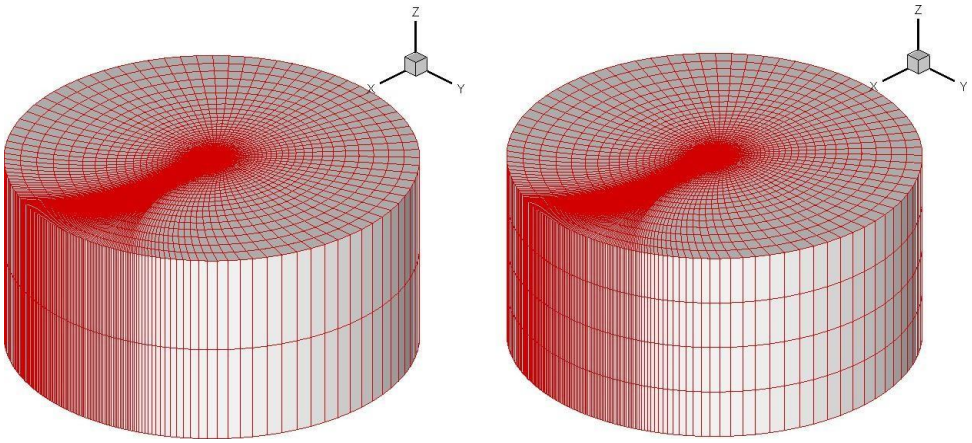

Figure 34: Far view of the mesh in the fine state(right) and after the coarsening(left)

Table 13: Specification before and after coarsening

Specification	Before Coarsening	After Coarsening
Number of Cells	76464	49867
Number of Faces	249216	166437
Number of Regions	6	6
Faces of Region1	152220	96987
Faces of Region2	708	354
Faces of Region3	708	708
Faces of Region4	57348	31132
Faces of Region5	19116	18660
Faces of Region6	19116	18596

Descriptions:

- Average program run time: 70 seconds
- As shown in the figure, the middle layers were in the direction of z axis and were in the direction of the least mesh stretching were completely removed from the mesh

---

SUGGESTIONS AND FUTURE WORK

- In this research, two different criteria are used to coarse the isotropic and anisotropic regions. Given that the metric fields will become a circle or a sphere in isotropic regions, if in the isotropic regions of the mesh the metric field be used; then the coarsening of isotropic regions will be more accurate
- In [8], a new method for eliminating non-manifold conditions is introduced, in which the nodes and edges that can be removed are identified by

Euler's criterion and removed if possible. This criterion can be used along with the presented algorithm to increase the accuracy.

- According to [11], The performance of these type of algorithms can be improved by segmenting the mesh and parallel processing. To do this, you can use clustering algorithms like FCM or K- Means, etc. We suggest that, for simplicity, first create a Voronoi graph of the mesh elements, which that nodes will be the centers of each element. Then clustering will be done on the Voronoi graph nodes. It is better to use the metric fields as a distance criterion
- The layering method introduced in this study can be used in studies that needs mesh

boundary layering.

DISCUSSION AND CONCLUSION

In the past, many algorithms have been developed to simplify the meshes. The most effective algorithms are based on the multi-grid methods. In these methods, a sequence of meshes will be created. Which will make each mesh coarser than its previous mesh. In addition, we concluded that to detect the mesh removable elements, we should use two different criteria. One of those is used to coarse the isotropic regions, and the other is used for the anisotropic regions. In the isotropic states, we used the "nearest neighbor distance" criterion, However, In the anisotropic regions, we used the metric fields to coarsening. The results show that by using these two criteria together, the final mesh will have better quality. We've improved the previous layering algorithms and created a new mesh boundary layering method. We did this by correcting the layers in each step. This will improve the mesh boundary layering accuracy.

In this survey, after detecting the removable elements, we used the edge contraction method to eliminate them. In this algorithm, all types of elements can be coarse. In other words, at each step, by removing each edge, the element type or the face type will be changed. we can say, at each step the invalid states of each element are identified and prevented from occurring. In order to identify the type of elements, we would use a metric called the metric fields. Based on the experimental results, using the field metric, we can do a better layering and a more accurate coarsening in the anisotropic regions.

REFERENCES

[1] Sivri, Talya Tümer, and Yusuf Sahillioğlu. "A data-centric unsupervised 3D mesh segmentation method." The Visual Computer 40.4 (2024): 2237-2249.

[2] Feather, William G., Hojun Lim, and Marko Knezevic. "A numerical study into element type and mesh resolution for crystal plasticity finite element modeling of explicit grain structures" *Computational Mechanics* 67 (2021): 33-55.

[3] Sumanta Guha (2015). "Mesh Simplification Via A Volume Cost Measure", International Journal of Computer Graphics & Animation (IJCGA) Vol.5, No.2, April 2015, PP 53-64.

[4] Mesri, Y., Guillard, H., & Coupez, T. (2012). "Automatic coarsening of three dimensional anisotropic unstructured meshes for multigrid applications", Applied Mathematics and Computation, 218(21), PP 10500-10519.

[5] Brune, P. R., Knepley, M. G., & Scott, L. R. (2013). "Unstructured geometric multigrid in two and three dimensions on complex and graded meshes", SIAM Journal on Scientific Computing, 35(1), PP A173- A191.

[6] Coupez, T. (2011). "Metric construction by length distribution tensor and edge-based error for anisotropic adaptive meshing. Journal of computational physics", 230(7), PP 2391-2405.

[7] Alauzet, F., Li, X., Seol, E. S., & Shephard, M. S. (2006). "Parallel anisotropic 3D mesh adaptation by mesh modification. Engineering with Computers", 21(3), PP 247-258.

[8] Owen, S.J., Brown, J.A., Ernst, C.D., Lim, H. and Long, K.N., 2017. "Hexahedral mesh generation for computational materials modeling", Procedia Engineering, 203, pp.167-179.

[9] Shi, Zhuo, Yalei An, Songhua Xu, Zhongshuai Wang, Ke Yu, and Xiaonan Luo. "Mesh Simplification Method Based on Reverse Interpolation Loop Subdivision", In Proceedings of the 8th International Conference on Computer Modeling and

Simulation, pp. 141-145. ACM, 2017.

[10] Daldoul, Wafa. "Moving mesh for complex geometries", PhD diss., Ecole Nationale Supérieure des Mines de Paris, 2017.

[11] Khattab Dina. Ebeid Hala.M. Hussein Ashraf.S. Tolba Mohamed.F. (2016). "3D Mesh Segmentation Based on Unsupervised Clustering", Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016. AISI 2016. Advances in Intelligent Systems and Computing, Vol 533, PP 598-607. Springer, Cham.

[12] Chao, Yin, Wang Jiateng, Qiu Guoqing, and Dong Kun. "A mesh simplification algorithm based on vertex importance and hierarchical clustering tree", In Eighth International Conference on Digital Image Processing (ICDIP 2016), vol. 10033, p. 1003364. International Society for Optics and Photonics, 2016.

[13] Jiang, Yunliang, Wuyang Nie, Liang Tang, Yong Liu, Ronghua Liang, and Xiulan Hao. "Vertex Mesh Simplification Algorithm Based on Curvature and Distance Metric", In Transactions on Edutainment XII, pp. 152-160. Springer, Berlin, Heidelberg, 2016.

[14] Lai, S., Chen, K. (2016). "3D Mesh Coarsening via Uniform Clustering. World Academy of Science, Engineering and Technology", International Journal of Computer, Electrical, Automation, Control and Information Engineering, 10(6), PP 1059-1063.

[15] Fortunato, Meire, and Per-Olof Persson. "High-order unstructured curved mesh generation using the Winslow equations", Journal of Computational Physics 307 (2016): 1-14.

[16] David Bommes, Bruno Levy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, Denis Zorin (2013). "Quad-Mesh Generation and Processing: A Survey", Computer Graphics forum Vol:32 (2013), number 6, PP 51-76.