12-13-2003

Quality Improvements in Extruded Meshes Using Topologically Adaptive Generalized Elements

Satish Chalasani

Recommended Citation

Chalasani, Satish, "Quality Improvements in Extruded Meshes Using Topologically Adaptive Generalized Elements" (2003). *Theses and Dissertations*. 3684.
https://scholarsjunction.msstate.edu/td/3684

QUALITY IMPROVEMENTS IN EXTRUDED MESHES USING TOPOLOGICALLY ADAPTIVE GENERALIZED ELEMENTS

By

Satish Chalasani

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computational Engineering
in the College of Engineering

Mississippi State, Mississippi

December 2003

QUALITY IMPROVEMENTS IN EXTRUDED MESHES USING TOPOLOGICALLY ADAPTIVE GENERALIZED ELEMENTS

By

Satish Chalasani

Approved:

David S. Thompson Associate Professor of Aerospace Engineering (Director of Dissertation)	Edward A. Luke Assistant Professor of Computer Science (Committee Member)
Clarence O. E. Burg Assistant Research Professor of Computational Engineering (Committee Member)	Michael G. Remotigue Assistant Research Professor of Computational Engineering (Committee Member)
Boyd Gatlin Associate Professor of Aerospace Engineering (Committee Member and Graduate Coordinator, Department of Computational Engineering)	A. Wayne Bennett Dean of the College of Engineering

Name: Satish Chalasani

Date of Degree: December 13, 2003

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr. David S. Thompson

Title of Study: QUALITY IMPROVEMENTS IN EXTRUDED MESHES USING TOPOLOGICALLY ADAPTIVE GENERALIZED ELEMENTS

Pages in Study: 97

Candidate for Degree of Doctor of Philosophy

Abstract

In this dissertation, a novel method to extrude near-body meshes from surface meshes of arbitrary topology that exploits topologically adaptive generalized elements to improve mesh quality is presented. Specifically, an advancing layer algorithm to generate near-body meshes which are appropriate for viscous fluid flows is discussed. First, an orthogonal two-layer algebraic reference mesh is generated. The reference mesh is then smoothed using a locally three-dimensional Poisson-type mesh generation equation that is generalized to smooth extruded meshes of arbitrary surface topology. Local quality improvement operations such as edge collapse, face refinement, and local reconnection are performed in each layer to drive the mesh toward isotropy. An automatic marching thickness reduction algorithm is used to extrude from multiple geometries in close proximity. A global face refinement algorithm is used to improve the transition from the extruded mesh to the void-filling tetrahedral mesh. A few example meshes along with quality plots are presented to demonstrate the efficacy of the algorithms developed.

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to Dr. David S. Thompson, who acted as a major advisor and the director of my dissertation. His constant support and encouragement has given me an opportunity to learn, and grow professionally. I am indebted for his suggestions and guidance through out the dissertation work. It is hard to find enough words to express my appreciation for his patience shown towards me in many issues and guiding me to go in proper direction.

I want to thank my committee members Dr. Edward A. Luke, Dr. Clarence O. E. Burg, Dr. Michael G. Remotigue, and Dr. Boyd Gatlin for their invaluable support, and time. I want to thank Dr. Edward A. Luke for letting me to use his solution plots. I want to thank Dr. Michael G. Remotigue for giving the X38 geometry.

I want to thank every one in ERC who helped me since fall 1999 to date. I want to thank all my friends with whom I had nice experience in starkville, especially Thomas (Jumbo), Raj, Mukti, Jaya and many others. I want to thank my parents and my wife Jyostna for their support during this work.

The partial support provided by National Science Foundation (ACS-0058969 and EEC-0121807), by the Army Research Office (DAA D19-00-1-0155), and by NASA Glenn Research Center (NAG3-2562) are greatfully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

In computational fluid dynamics and solid mechanics, algorithms using finite difference, finite volume, and finite element methods need discrete meshes for approximating their respective governing equations. Meshes can be categorized as one of the following types depending on how neighboring cell information is accessed and the types of cells used:

- structured mesh
- unstructured mesh
- hybrid mesh
- meshless method.

1.1 Structured mesh

Thompson et al. [1] describe different methods to generate a structured mesh. A structured mesh is generated by mapping the coordinates in physical space (x, y, z) onto a hexahedral computational space (ξ, η, ζ). An example of a two-dimensional structured mesh is shown in figure 1.1.

The structured mesh information lends itself to storage in an array. The neighboring points and cells are accessed by incrementing or decrementing an array index. This avoids extra memory needed to identify and access the neighbor information.

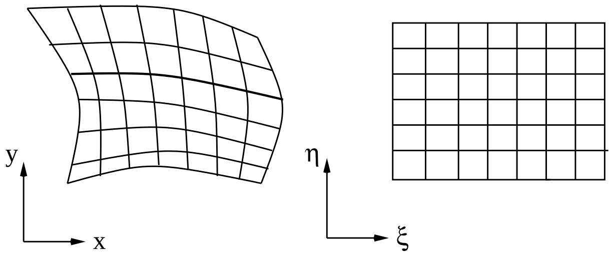


Figure 1.1: Example of a two-dimensional structured mesh

If the geometry is complex in shape, it is difficult to map the computational domain to a single computational block. Allwright [2] and Dannenhoffer [3] described a multi-block decomposition strategy which divides the domain into smaller sub-domains or blocks and each block is mapped individually to a computational space. The disadvantage of the multi-block approach is that it is labor-intensive and time consuming. Structured meshes are not suitable for adaption through h-refinement where more points are to be added in a particular region. On the other hand, structured meshes permit highly anisotropic cells that can resolve viscous flows. Cells generated for this type of mesh are quadrilaterals in two dimensions and hexahedra in three dimensions. Software such as GUM-B [4], Fluent [5], Gridgen [6], and GridPro [7] are a few state-of-the-art structured mesh generators that are based on the multi-block approach.

1.2 Unstructured mesh

As compared to a structured mesh, an unstructured mesh [8, 9, 10, 11] can be generated independent of the geometric complexity of the domain. Figure 1.2 shows an example unstructured mesh made of triangular elements in a circle geometry. The following are methods that are commonly used to generate unstructured meshes:

- Delaunay triangulation [8]
- advancing front method [10]
- quadtree and octree methods [11].

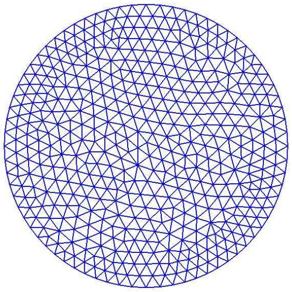


Figure 1.2: Example of an unstructured mesh

The points and cells in an unstructured mesh are stored with explicit neighbor information. Therefore, extra memory is required to access and store the neighbor information. The cells generally employed are triangles in two dimensions and tetrahedra in three dimensions. Unlike structured meshes, solution-adaptive meshing through h-refinement is possible for unstructured meshes. The main disadvantages associated with unstructured meshes when compared with structured meshes are:

- More memory is required (which is much larger than the memory required for a structured mesh of the same point density). Shaw [12] indicated that the storage requirements per point for a flow solution on an unstructured mesh are about four times those for a structured mesh.
- The CPU time for flow solvers is larger per mesh point when compared to structured mesh flow solvers [12].

If an isotropic unstructured mesh is used to resolve the boundary layer region, the number of resultant points and cells will be prohibitive. As alternatives, Pirzadeh [13], Marcum [14], and Garimella et al. [15] described methods that use advancing normal point placement strategies for generating highly anisotropic tetrahedral cells in the boundary layer region. It should be noted that there are questions regarding the accuracy of computations on a highly anisotropic tetrahedra [16].

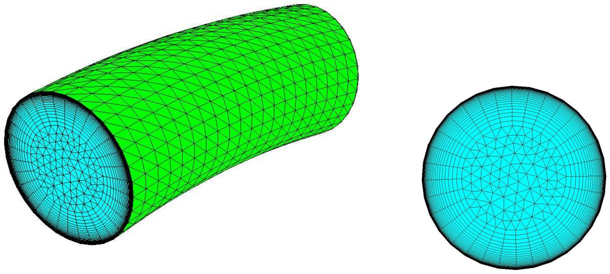
1.3 Hybrid mesh

Hybrid meshes combine the advantages of structured and unstructured meshes. Shaw [12] described a methodology to generate structured and unstructured meshes in a multi-block environment according to the geometric complexity. Nakahashi [17] used a zonal method to generate a hybrid mesh. In his method, a structured mesh is generated around the body and remaining domain is filled with a void-filling mesh. Chappell et al. [16] and Kallinderis [18] described methods to generate extruded meshes in viscous regions using advancing layer schemes. The layer definition is kept intact through out the extrusion process, i.e., the topology of the initial surface mesh and final extruded surface mesh will be identical. Laplacian smoothing is used to smooth the normals in concave regions as the mesh is extruded. The remainder of the domain is filled with a tetrahedral mesh. If the extruded surface has quadrilaterals, Owen [19] used pyramids as transition elements between the extruded mesh and the tetrahedral mesh. Wang [20] and Ito et al. [21] generated a void-filling mesh around the geometry and inserted a viscous layer between the void-filling mesh and the geometry.

A simple example of a hybrid mesh is shown in figure 1.3. In this example, the initial surface consists of triangles. The extruded volume mesh has prismatic elements near the body. The remainder of the domain is filled with tetrahedral elements. This example mesh has a uniform spacing variation at the interface of the near-body and tetrahedral meshes. Typically, cells generated in a hybrid mesh are of standard shapes: triangular prisms, tetrahedra, hexahedra, and pyramids.

1.4 Meshless methods

Meshes generating using the above methods pose numerous difficulties when used for simulating problems that are dynamic with large deformations with respect to time. The difficulties are mainly due to the necessity of connectivity information. These methods



(a) Volume mesh

(b) Mesh at the inlet of the pipe

Figure 1.3: Example of a hybrid mesh – curved pipe model

also require interpolation of the solution onto a new mesh as it is regenerated in successive stages of the problem which may reduce numerical accuracy of the solution. A meshless method that doesn't put any restriction on mesh connectivity is devised in reference [22]. In this method, the entire domain is approximated by cloud of points. This method is easy to implement as there is no need to specify connectivity between nodes. The nodes are distributed in the domain subject to certain constraints such as no node should lie within a certain radius of any other. Belytschko et al. [22] discussed this method in more detail.

1.5 Motivation

In an advancing layer scheme, a near-body mesh is generated by extruding from a given surface mesh in the direction of local surface normals. As the mesh is extruded, layer by layer, smoothing is required in the concave regions of the geometry to avoid crossing. To address mesh crossing, Kallinderis [18] used Laplacian smoothing to smooth the normals. However Laplacian smoothing doesn't guarantee a valid mesh [23] and doesn't provide a mechanism to control the mesh quality. Therefore, smoothing is needed which provides better control than Laplacian smoothing.

The example hybrid mesh shown in figure 1.3 has a smooth variation in the cell size near the interface of the extruded and tetrahedral meshes. Unfortunately, volume meshes generated using advancing layer methods may have poor quality elements near convex and concave regions of the geometry. Figure 1.4 shows an example of such poor quality elements. These elements are of poor quality because they may adversely affect the transition between the extruded mesh and tetrahedral mesh.

To address this shortcoming, several researchers [24, 25, 26] have considered the possibility of using other element types to improve mesh quality based on the concept of topological adaptivity. In a topologically adaptive mesh, cells are used that are locally appropriate to the region being discretized. The resulting meshes, which may contain arbitrary polyhedra with no restrictions placed on the number of edges a face can have or the number of faces a element can have, have been dubbed generalized meshes [27].

Thompson and Soni [24] employed face termination and face refinement to improve mesh quality near convex and concave regions of the geometry. Their strategy resulted in the production of general elements with "hanging nodes" and was limited to structured quadrilateral surface meshes.

Leatham et al. [25] employed edge collapse and edge refinement to achieve smooth transition. Their approach utilized non-face matching, i.e., a face of one element need not be coincident with a face in the adjacent element. The mismatch is caused by the application of face refinement to the initial surface mesh before extrusion step and imposes additional requirements on the flow solver.

The concept of the generalized prism is introduced in reference [26] to improve the quality of the mesh. A cell called a "filler cell", obtained by collapsing one of the triangular faces of a prism to a line segment, is used to improve quality near concave and convex regions. In order to control the resolution of the mesh near convex regions, degenerate triangles (lines and points) are added on the initial triangulated surface mesh so that, when extruded from these entities, the resultant faces will have finite area and the “filler cells” formed out of these degenerate faces will have finite volumes. Due to the addition of degenerate faces on the initial surface, new extrusion directions must be computed.

For a hybrid mesh with basic element types, Marcum [14] and Pirzadeh [13] developed a technique to provide a smooth transition between the extruded mesh and the tetrahedral mesh by terminating the extruded mesh locally when the algorithm tries to insert a new point close to an existing point or if the spacing is comparable to the initial surface mesh spacing. However, this approach may result in a premature transition to tetrahedral elements. Additionally, this approach was developed in the context of an advancing front algorithm and is not appropriate for an advancing layer algorithm. Therefore, a near-body mesh generation method independent of initial surface mesh topology which improves mesh quality near concave and convex regions of the geometry and doesn’t put constraints like non-face matching between layers is required.

Extruding from multiple bodies which are in close proximity, may lead to mesh overlap or crossing. Kallinderis [18] developed a method which automatically reduces the extruded mesh thickness to avoid overlapping and crossing. Due to the reduction of near-body mesh thickness, the resultant transition from the extruded mesh to a tetrahedral mesh may not be smooth. Therefore, a new method which improves the transition from the extruded mesh to the tetrahedral mesh is needed.

1.6 Dissertation statement

Given a surface mesh, the objective of this dissertation is to develop the techniques needed to generate a near-body mesh that uses topologically adaptive elements to produce a smooth transition from the near-body mesh to the void-filling mesh. The efficacy of the developed techniques will be evaluated using appropriate quality measures.

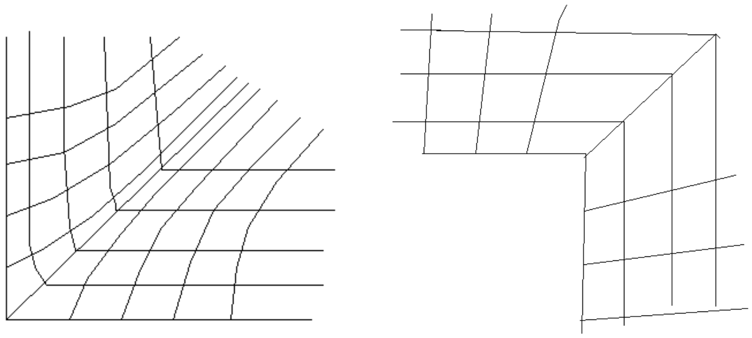
1.7 Contributions

In this dissertation, three unique contributions are made. They are:

- Development of a new mesh smoothing algorithm which uses a Poisson equation to smooth the mesh point locations instead of the extrusion normals. The advantage of this method over other methods in the literature is that it allows the use of control functions. The control functions are used to drive the mesh toward the desired properties [28].

- Development of quality improvement operations such as edge collapse and face refinement in extruded meshes generated by an advancing layer algorithm. These operations are applied in each layer to improve the transition from the extruded mesh to the void-filling tetrahedral mesh. These operations typically result in generalized topologies. Face matching is enforced by applying all the operations on the extruded surface mesh and using the modified extruded surface mesh as the initial surface mesh to build the next layer. This doesn't place any constraints on the flow solver to handle non-face matching topologies. There is no need to compute new extrusion directions as degenerate faces are not added on the initial surface. Additionally, generalized cell topologies eliminate the need for a separate transition layer to the tetrahedral mesh.

- Development of a technique to handle extrusion from multiple geometries in close proximity coupled with a face refinement operation to get smooth transition from near-body mesh to tetrahedral mesh. The near-body mesh thickness is reduced to avoid overlapping or crossing of the meshes. This technique is applied before starting the extrusion process. As a result, the transition may not be good in the regions where the mesh thickness is reduced. This transition is improved by using the face refinement operation mentioned above.



(a) Mesh in concave regions (b) Mesh in convex regions

Figure 1.4: Extruded mesh near concave and convex regions

1.8 Outline of the dissertation

Chapter 2 outlines the algorithm used to generate the near-body mesh and smoothing procedure. Chapter 3 describes topological operations used to improve the local mesh quality near convex and concave regions of the geometry. Chapter 4 describes a methodology to address problems that could be encountered in the case of multiple bodies in proximity with or without similar node spacings. A few results with quality plots are presented in chapter 5 to show the efficacy of the algorithms discussed in the dissertation. Chapter 6 includes conclusions and recommendations for future work.

CHAPTER II

NEAR-BODY MESH GENERATION

This chapter describes a new method for generating a near-body mesh given a surface mesh of arbitrary topology. An advancing layer scheme is used to extrude the near-body mesh. The following section explains the algorithm used to generate the near-body mesh. Noack [29] described a parabolic mesh generation algorithm for structured meshes. His algorithm is modified to extrude volume meshes from surface meshes of arbitrary topology.

2.1 Outline of the algorithm

The following three-step process is used to generate each layer of the mesh. Please refer to figure 2.1.

(i) *Generate the reference mesh* – Generate two layers of a reference mesh, k and $k+1$, by extruding from an initial surface mesh, points 1, 2, and 3, in the direction of the local surface normals. A simple way to compute the extrusion direction at a given node is to take an average of face normals sharing the node under consideration. However, after extrusion, the resulting point location may violate the visibility condition [18] if the surface has sharp discontinuities, for example, at the intersection of a wing trailing edge and a fuselage. Kallinderis [18] describes a method to find a normal direction that satisfies the visibility condition. This method is used to calculate the normal if the simple average normal fails to satisfy the visibility condition.

(ii) *Smooth the reference mesh* – Smooth the interface between the first and second layers of the reference mesh, points 4, 5, and 6, using a Poisson equation. Update the outer surface of the second layer of the reference mesh, points 7, 8, and 9, after every iteration. The Poisson equation is obtained by modifying the method described in [30] for smoothing surface meshes of arbitrary topology. Control functions are used to maintain the characteristics of the initial surface. Note that the actual point positions are adjusted using this smoothing.

(iii) *Perform quality improvements* – After smoothing, perform local mesh quality improvements on the extruded surface of the first layer, points 4, 5, and 6, and discard the outer surface of the second layer, points 7, 8, and 9. Then, update the face and element connectivities to reflect the changes due to quality improvements. The outer surface of the first layer of the mesh, points 4, 5, and 6, is used as the initial surface for the next layer.

The above three steps are repeated until the desired number of layers is extruded. This method can be used independently of the initial surface mesh topology. The advantage of using Poisson smoothing over Laplacian smoothing is the use of control functions [28]. The control functions can be used to drive the mesh toward desired characteristics.

The following explains in detail the methodology employed in building and smoothing the reference mesh. The next two chapters discuss the quality improvement operations performed after smoothing.

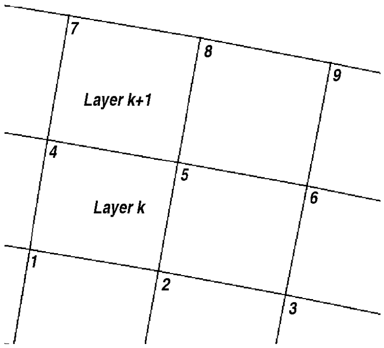


Figure 2.1: Reference mesh

2.2 Reference mesh

The methodology for building a reference mesh is explained with the aid of figure 2.2. Given the initial surface mesh, the first surface mesh nodes are defined using

$$r_{m,2} = r_{m,1} + \delta_m^k n_{m,1}^k \tag{2.1}$$

where δ_m^k is the marching distance in layer k, and $n_{m,1}$ is the computed unit normal. Before generating a node position in the second surface mesh, a visibility check [18] is performed to ensure is a valid position. The connectivity and face topology of the initial surface mesh are maintained. The positions of second surface mesh are computed by using

$$r_{m,3} = r_{m,2} + \delta_m^{k+1} n_{m,2}^{k+1}. \tag{2.2}$$

In Eq. 2.2, $r_{m,3}$ is the position in the second surface mesh corresponding to the position $r_{m,2}$ in first surface mesh, δ_m^{k+1} is the marching distance in layer k+1, and $n_{m,2}$ is the computed unit normal. The following subsections explain the normal computation and the visibility condition in detail.

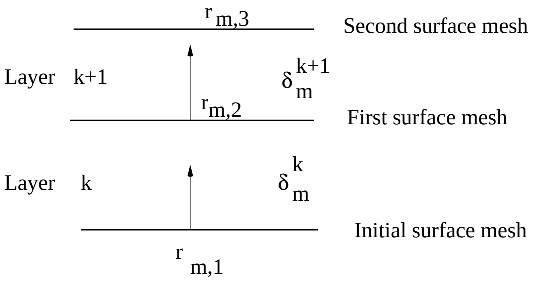


Figure 2.2: Terminology used for computation of position vector in two layer reference mesh

2.2.1 Valent faces & valent face nodes

The valent faces of a node are those faces which share that node. The valent face nodes are the nodes of the valent faces. The valence of a node is given as the number of its valent face nodes. In figure 2.3, the valence of node 1 is 6 and the valent face nodes are 2, 5, 7, 9, 4, and 6. The valent face nodes and valent face information are used to compute the normal at each node.

Depending on the topology of the domain being meshed, i.e., if the surface being extruded from is not closed, the extrusion may have "side" boundaries. Nodes and edges located on these side boundaries are called boundary nodes and edges, respectively. If the node is a boundary node then the valent face nodes are determined according to the boundary condition. Generally there are three kinds of boundary conditions:

- *Constrained* – The boundary normals are constrained to lie within a specified plane. In this case, the interior nodes are mirrored about the user specified plane and the resultant nodes are also considered as valent face nodes. For example, in figure 2.4(a) the node 6 has valent nodes 2, 3, 4, 8, 13, 11, 10, and 9 out of which nodes 13, 11, and 10 are the mirror positions of nodes 4, 3, and 2, respectively, about the user specified boundary plane.

- *Floating* – The boundary is allowed to float based on the local surface normal. In this case, the plane of about which the interior nodes are mirrored is computed

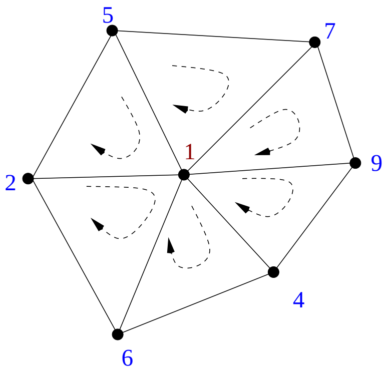
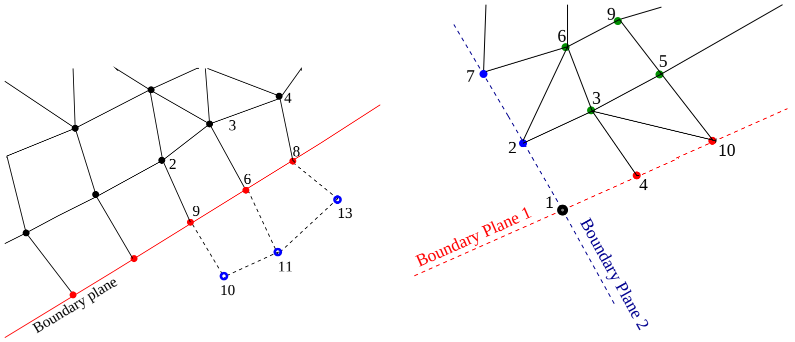


Figure 2.3: Valent faces and valent face nodes of node 1



(a) Constrained boundary condition

(b) Fixed boundary condition

Figure 2.4: Different boundary conditions

using the local surface mesh topology. The mirrored nodes are also considered as valent face nodes.

- *Fixed* – In this case the normal is set in a fixed direction. For example, in figure 2.4(b), node 1 is at the intersection of two boundary planes. Therefore, the normal direction emanating out from the node 1 should point in the same direction as the intersection vector of the two boundary planes.

2.2.2 Normal calculation

A normal at each node is computed as the average of the face normals that share the node under consideration. For example, in figure 2.3, planes formed by the node 1 with two of its consecutive valent nodes are listed below.

plane	normal
plane $(1,6,2)$	n_1
plane $(1,2,5)$	n_2
plane $(1,5,7)$	n_3
plane $(1,7,9)$	n_4
plane $(1,9,4)$	n_5
plane $(1,4,6)$	n_6

Therefore, the normal at node 1 is given by

$$\frac{1}{6}\sum_{i=1}^{6} n_i. \tag{2.3}$$

The computed normal is normalized to a unit magnitude. In complex geometries, the simple average algorithm for normal calculation may not result in a position which satisfies the visibility check [18]. Even an area weighted normal computation may also fail the visibility check as it depends on the initial surface mesh discretization. An uneven discretization causes the normal to drift. Therefore, the special method described in [18] is used to compute the normal. This method will be used at the intersection of two or more surfaces. A few problems that may be encountered using this method are stated and measures taken to overcome those are suggested. With reference to figure 2.5, the following steps explain the special method:

- The two valent faces f_{k1} and f_{k2} (figure 2.5(a)) of node 1 which intersect to form the most acute wedge are determined such that

$$\hat{n}_{k1} \cdot \hat{n}_{k2} \leq \hat{n}_i \cdot \hat{n}_j \tag{2.4}$$

where $\hat{n}_{k1}$, $\hat{n}_{k2}$, $\hat{n}_i$, and $\hat{n}_j$ are the unit normals of valent faces f_{k1}, f_{k2}, f_i and f_j, respectively.

- An intersection vector for the valent faces f_{k1} and f_{k2} is computed using the following equation

$$I_1 = \hat{n}_{k1} \times \hat{n}_{k2}. \tag{2.5}$$

- A vector $\hat{B}$ lying on the bisection plane of the valent faces f_{k1} and f_{k2} and which is perpendicular to I_1 is computed using

$$\hat{B} = \frac{\hat{n}_{k1} + \hat{n}_{k2}}{2}. \tag{2.6}$$

- The intersection positions ($r_{t^\pm}$) (figure 2.5(b)) of the remaining valent faces f_k ($k \neq (k_1, k_2)$) with the line whose unit vector is I_1 are computed by

$$\hat{n}_k \cdot (\hat{B} + tI_1) = 0. \tag{2.7}$$

The $t^\pm$ in intersection positions ($r_{t^\pm}$) denote positive and negative values of the parameter t in Eq. 2.7. The calculation of the intersection parameter t may lead to a numerical inconsistency if the valent face (f_k) normal $\hat{n}_k$ and the vector I_1 are perpendicular to each other. In order to avoid this numerical inconsistency, Eq. 2.7 is modified by adding a small quantity ϵ. The modified equation is given by

$$\hat{n}_k \cdot \hat{B} + t(\hat{n}_k \cdot I_1 + \epsilon) = 0. \tag{2.8}$$

- Finally, the unit normal at node 1 $\hat{n}_1$ in (figure 2.5(b)) is given as the the bisector of

$$(r_{max(t)} - r_{m,1}) \text{ and } (r_{min(t)} - r_{m,1}). \tag{2.9}$$

2.2.3 Visibility condition check

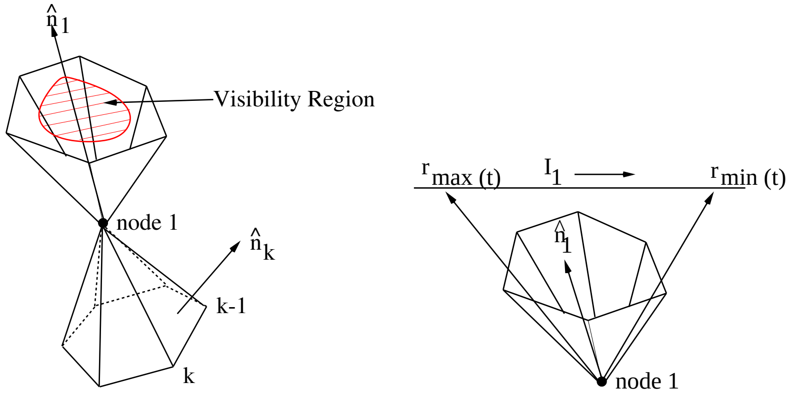
The position vector $r_{m,2}$ of a node in the first surface mesh (figure 2.2) is accepted in the list of first surface mesh nodes if it is visible to all the valent face nodes of $r_{m,1}$ of the initial surface mesh. This is called the visibility condition [18] and ensures the following:

- The computed point is above the initial surface mesh
- The line connecting any of the valent face nodes in the initial surface mesh to the computed point doesn't intersect any of its valent faces.

Figure 2.5(a) shows the region in which the normal at node 1 should fall to satisfy the visibility condition. The visibility check for each computed node position in the first layer is performed by determining the sign of the tetrahedral volume. The tetrahedra is formed by the following three position vectors:

1. A first vector is formed by $r_{m,1}$ and $r_{m,2}$.
2. A second vector is formed by $r_{m,1}$ and the first valent face node of the edge which is made up of two valent face nodes.
3. A third vector is formed by $r_{m,1}$ and the second valent face node of the edge considered in the second vector.

This test is performed repeatedly by replacing the edge considered in the second vector with the remaining edges that share any two consecutive valent face nodes of $r_{m,1}$ in the initial surface mesh. A positive sign of the volume indicates that the point is visible to the valent face nodes of $r_{m,1}$. The same check is performed after computing the



(a) Visibility region for the normal at node 1

(b) Intersection of the face planes with the line I_1

Figure 2.5: Normal calculation at node 1

positions of the nodes in second surface mesh. After computing positions in the first and second layer surface meshes, the next step is smoothing the mesh. The following section describes in detail the need for smoothing and the methodology used to smooth the mesh.

2.3 Smoothing the reference mesh

Smoothing is required in regions where the mesh faces are converging or diverging. For example, in concave regions extrusion may result in mesh crossing. The basic assumption here is that the initial surface mesh has desirable characteristics and the generated reference mesh is locally orthogonal. Laplacian smoothing can be used to smooth the mesh normals before extrusion but it is not as resistant to crossing as Winslow smoothing [23]. Further, Winslow smoothing provides no mechanism to propagate the initial surface mesh characteristics in to the mesh. On the other hand, Poisson smoothing, which is not as resistant to crossing as Winslow, permits the use of control functions to influence the spacing of the nodes on the extruded surfaces [28].

In a structured mesh generation, a global mapping of the form

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z) \end{aligned} \tag{2.10}$$

is typically employed. The inverse mapping

$$\begin{aligned} x &= x(\xi, \eta, \zeta) \\ y &= y(\xi, \eta, \zeta) \\ z &= z(\xi, \eta, \zeta) \end{aligned} \tag{2.11}$$

is also assumed to exist. Assuming that ζ is the marching (extrusion) direction and ζ lines are orthogonal to the extruded $\zeta=$ constant surface, the Poisson equation commonly used for structured mesh generation becomes

$$\frac{g_{22}g_{33}}{g^2}(r_{\xi\xi}+\Phi r_\xi)+\frac{g_{11}g_{33}}{g^2}(r_{\eta\eta}+\Psi r_\eta)-2\frac{g_{12}g_{33}}{g^2}r_{\xi\eta}+\frac{g_{11}g_{22}-g_{12}^2}{g^2}(r_{\zeta\zeta}+\Theta r_\zeta)=0 \tag{2.12}$$

where $r(x, y, z)$ is the position vector,Φ, Ψ, and Θ are the control functions for the ξ, η, and ζ directions, respectively, and

$$\begin{aligned} g_{11} &= x_\xi^2 + y_\xi^2 + z_\xi^2 \quad & g_{22} &= x_\eta^2 + y_\eta^2 + z_\eta^2 \\ g_{12} &= x_{\xi\eta} + y_{\xi\eta} + z_{\xi\eta} \quad & g_{33} &= x_\zeta^2 + y_\zeta^2 + z_\zeta^2 \end{aligned} \tag{2.13}$$

$$g = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix} .$$

In a standard structured mesh generation algorithm, the partial derivatives appearing in Eq. 2.12 and Eq. 2.13 are approximated using standard second-order central differences. The same approach can't be used for smoothing unstructured meshes. Knupp [23] described an approach for applying Poisson smoothing to an unstructured quadrilateral mesh. In this approach, a global coordinate mapping is abandoned and a local, discrete uniform logical space (ξ_m, η_m) is defined at each node using

$$\begin{aligned} \xi_m &= \cos\theta_m \\ \eta_m &= \sin\theta_m \end{aligned} \tag{2.14}$$

where

$$\theta_m = \frac{2\pi m}{M} \tag{2.15}$$

with M being the number of valent face nodes and m=0,1,...M-1. By assuming a smooth, C^∞ function $f(\xi, \eta)$ on the local logical space and using integral powers of the roots of unity, an approximation of $f(\xi, \eta)$ about $f(0,0)$ using a Taylor series expansion is given as

$$f(\cos\theta_m, \sin\theta_m) = \sum_{l=1}^{L} \frac{1}{(l-1)!}(\cos\theta_m \frac{\partial}{\partial\xi} + \sin\theta_m \frac{\partial}{\partial\eta})^{l-1} f|_{0,0}. \tag{2.16}$$

Eq. 2.16 is manipulated to derive first and second derivatives approximations. A detailed description is given in [31]. The following are the difference approximations for first and second derivatives that are obtained by manipulating Eq. 2.16.

- For valent face nodes $M \geq 5$

$$\begin{aligned}
f_\xi &= \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_o) \cos\theta_m \\
f_\eta &= \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_o) \sin\theta_m \\
f_{\xi\xi} &= \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_o)(4\cos^2\theta_m - 1) \\
f_{\xi\eta} &= \frac{8}{M} \sum_{m=0}^{M-1} (f_m - f_o) \cos\theta_m \sin\theta_m \\
f_{\xi\xi} &= \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_o)(4\sin^2\theta_m - 1)
\end{aligned} \tag{2.17}$$

  where f_o is the value of the function at the node under consideration.

- For valent face nodes M=4 and M=3, the mid point of the edge connecting two valent face nodes is also considered as a valent face node and accordingly approximations in Eq. 2.17 are used for M=8 and M=6 respectively.

- For valent face nodes M=2, Laplacian smoothing is used.

Knupp [23] derived the expressions given in Eq. 2.17 to use in the Poisson equation to smooth quadrilateral surface meshes. The Poisson smoother is not appropriate for smoothing a surface with significant convex and or concave regions unless some method is employed to constrain the resulting surface deformations. Therefore, the surface Poisson smoother is modified by including a three-dimensional term which constrains the deformation of the surface mesh points. Therefore, the expressions in Eq. 2.17 are used only in a ζ=constant surface and standard finite-difference approximations are used for ζ derivatives. This approach is possible due to the structure of the mesh in the extrusion direction. The first surface mesh (figure 2.2) is smoothed using a point Jacobi algorithm. After each iteration, the second surface mesh node positions are updated to reflect the changes in the first surface mesh nodes.

2.3.1 Control functions

The control functions Φ, Ψ, and Θ play critical roles in determining the quality of the mesh. The given initial surface mesh is assumed to be of good quality and g_{33} is taken as a constant within the surface. The control functions are derived such that the mesh characteristics of the initial surface mesh are propagated into the interior of the domain assuming orthogonality of the mesh within each layer, i.e., $g_{13} = g_{23} = 0$. The control functions Φ and Ψ are computed on the initial surface mesh and used for smoothing the first surface mesh. The control function Θ is computed using the marching distance distribution and by neglecting the variation of other metrics in the marching direction. The three control functions Φ, Ψ, and Θ in Eq. 2.12 are coupled. In order to decouple the control functions [28], the dot products of Eq. 2.12 with r_ξ, then r_η, and finally with r_ζ are taken. The resultant equations, after substituting metric identities are given below.

$$\begin{aligned}
&\frac{g_{22}g_{33}}{g^2}(\frac{g_{11_\xi}}{2} + \Phi g_{11}) + \frac{g_{11}g_{33}}{g^2}(g_{12_\eta} - \frac{g_{22_\xi}}{2} + \Psi g_{12}) \\
&\qquad -2\frac{g_{12}g_{33}}{g^2}g_{11_\eta} - \frac{g_{11}g_{22} - g_{12}}{g^2}\frac{g_{33_\xi}}{2} = 0 \\
&\frac{g_{22}g_{33}}{g^2}(g_{12_\xi} - \frac{g_{11_\eta}}{2} + \Phi g_{12}) + \frac{g_{11}g_{33}}{g^2}(\frac{g_{22_\eta}}{2} + \Psi g_{22}) \\
&\qquad -2\frac{g_{12}g_{33}}{g^2}\frac{g_{22_\xi}}{2} - \frac{g_{11}g_{22} - g_{12}^2}{g^2}\frac{g_{33_\eta}}{2} = 0 \\
&-\frac{g_{22}g_{33}}{g^2}(\frac{g_{11_\zeta}}{2}) - \frac{g_{11}g_{33}}{g^2}(\frac{g_{22_\zeta}}{2}) - 2\frac{g_{12}g_{33}}{g^2}g_{12_\zeta} \\
&\qquad +\frac{g_{11}g_{22} - g_{12}^2}{g^2}(\frac{g_{33_\zeta}}{2} + \Theta g_{33}) = 0
\end{aligned} \tag{2.18}$$

In Eq. 2.18, ζ lines are assumed to be orthogonal in the layer. With this assumption, the control functions Φ, and Ψ in the ζ= constant surface decouple from the control function Θ. Through algebraic manipulation of Eq. 2.18, the following equations are obtained to compute the control functions.

$$\begin{aligned}\Phi + \left(\frac{g_{12}}{g_{22}}\right)\Psi = -\frac{1}{2}\left(\frac{g_{11_\xi}}{g_{11}} - \frac{g_{22_\xi}}{g_{22}} - \frac{g_{11}g_{22} - g_{12}^2}{g_{11}g_{22}}\frac{g_{33_\xi}}{g_{33}}\right) \\ + \left(\frac{g_{12}}{g_{22}}\right)\left(\frac{g_{11_\eta}}{g_{11}} - \frac{g_{12_\eta}}{g_{12}}\right) = 0\end{aligned}$$

$$\begin{aligned}\left(\frac{g_{12}}{g_{11}}\right)\Phi + \Psi = -\frac{1}{2}\left(\frac{g_{22_\eta}}{g_{22}} - \frac{g_{11_\eta}}{g_{11}} - \frac{g_{11}g_{22} - g_{12}^2}{g_{11}g_{22}}\frac{g_{33_\eta}}{g_{33}}\right) \\ + \left(\frac{g_{12}}{g_{11}}\right)\left(\frac{g_{22_\xi}}{g_{22}} - \frac{g_{12_\xi}}{g_{12}}\right) = 0\end{aligned} \tag{2.19}$$

$$\begin{aligned}\Theta = -\frac{1}{2}\left(\frac{g_{33_\zeta}}{g_{33}} - \frac{g_{11}g_{22}}{g_{11}g_{22} - g_{12}^2}\left(\frac{g_{11_\zeta}}{g_{11}} + \frac{g_{22_\zeta}}{g_{22}}\right)\right) \\ - \frac{g_{12}^2}{g_{11}g_{22} - g_{12}^2}\left(\frac{g_{12_\zeta}}{g_{12}}\right) = 0\end{aligned}$$

The control function Θ which governs layer spacing is computed using a central finite-difference scheme. ξ and η derivatives are computed using Eq. 2.17. The advantage of this method over the other methods in the literature is the control of mesh quality. All the advantages of control functions [1], like clustering mesh lines towards a point or a line, or to a any user specified function, can be used.

2.3.2 Modification to Poisson smoother in concave regions

The Poisson equation is modified to avoid mesh folding in strongly non-convex regions. The approach used is an extension of the method described in [24]. In the regions where $(g_{11}, g_{22}) >> g_{33}$, the term containing ζ derivatives in Eq. 2.12 is dominant and it dictates the node position [1]. In non-convex regions, the resulting lack of smoothing in the ζ=constant surface can lead to mesh folding. The approach employed here is to modify the first two terms in Eq. 2.12 in order to reduce the dominance of the ζ derivative term. This can be thought as adding additional smoothing in highly non-convex areas. and can be accomplished by multiplying the coefficients of $r_{\xi\xi}$ and $r_{\eta\eta}$ in Eq. 2.12 by a factor. After modification, the Poisson equation is given by

$$\begin{aligned} &\frac{g_{22}g_{33}}{g^2}((1+v_\xi)r_{\xi\xi}+\Phi r_\xi)+\frac{g_{11}g_{33}}{g^2}((1+v_\eta)r_{\eta\eta}+\Psi r_\eta) \\ &\qquad -2\frac{g_{12}g_{33}}{g^2}r_{\xi\eta}+\frac{g_{11}g_{22}-g_{12}^2}{g^2}(r_{\zeta\zeta}+\Theta r_\zeta)=0. \end{aligned} \tag{2.20}$$

where v_ξ and v_η are directional smoothing coefficients. Using local the coordinate system defined by Eq. 2.15, the directional smoothing coefficients are given by

$$\begin{aligned} v_\xi &= \sum_{m=0}^{M-1} v_m|\cos\theta_m| \\ v_\eta &= \sum_{m=0}^{M-1} v_m|\sin\theta_m| \end{aligned} \tag{2.21}$$

where

$$\begin{aligned} v_m &= \left(\sqrt{\frac{max(g_m,g_{33})}{g_{33}}}-1\right)\times f(\alpha_m) \\ f(\alpha_m) &= \left\{\begin{array}{cc} \frac{1}{2} & 0\geq\alpha_m\geq\frac{\pi}{4} \\ \frac{1}{2}\sin 2\alpha_m & \frac{\pi}{4}<\alpha_m<\frac{\pi}{4} \\ 0 & \frac{\pi}{2}\leq\alpha_m \end{array}\right\} \end{aligned} \tag{2.22}$$

with

$$g_m = |r_m - r_o|^2 \tag{2.23}$$

and α_m is the angle between the local normal and $r_m - r_o$. This definition of v_m ensures the resulting smoothing is zero when $g_{33} \geq (g_{11}, g_{22})$ and is not as dissipative as Laplacian smoothing. The definition of $f(\alpha_m)$ ensures that the smoothing is applied only in locally concave regions. The smoothing is terminated either when the maximum error norm is less than a user specified tolerance or the number of smoothing iterations is greater than a user specified number. Before updating the point position, the new position is checked to see if it satisfies the visibility condition. The position is not updated if the test fails. After each smoothing iteration, the second surface mesh node positions are updated to reflect changes in the first surface mesh (figure 2.2). Smoothing does most of its work in concave regions. After smoothing is completed, the second surface mesh is discarded and the process is repeated by considering the first surface mesh as the initial surface mesh until the desired number of layers is achieved.

2.4 Data structures

The data structure plays an important role in implementing any mesh generation algorithm. The software developed is written in C++ using vectors and lists available in the standard template library. The data structure is face based, similar to the radial edge data-structure. Each face has a pointer to elements that share it. Attributes like aspect ratio, normal, angle, and node information are stored for each face. A face can be a triangle, a quadrilateral, or a five-sided polygon. There is no restriction on the number of faces a element can have. The number of nodes can be different from its previous layer, but there will be face to face match between consecutive layers.

CHAPTER III

TOPOLOGICAL OPERATIONS FOR LOCAL QUALITY IMPROVEMENT

This chapter discusses local quality improvement operations that are applied at the first surface mesh. The first point that must be addressed is the general issue of element quality. This topic has been studied extensively and is well understood for simplex elements such as triangles and tetrahedra [32, 30, 33]. Wa and Chen [34] attempted to quantify quality for hexahedral and prismatic elements using a combination of face properties to define a quality metric. Inherent to this approach is the assumption that a volume element is of good quality if its faces are of good quality. Following a somewhat different approach, Owen et al. [19] examined the quality of a pyramidal element by decomposing it into tetrahedral elements and then applying the tetrahedral element quality metric to the resultant elements. Obviously, this approach depends on the procedure followed to decompose the pyramid.

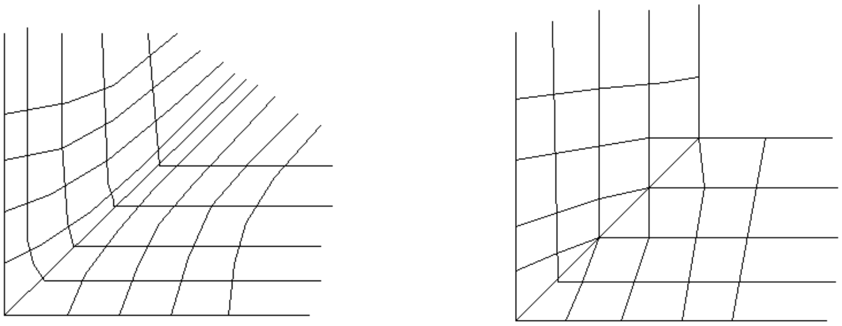
The approach taken here is modeled after the face-based decomposition employed in [34]. This approach is particularly appropriate for general elements since the decomposition of an arbitrary polyhedron into tetrahedral elements may not always be a straightforward process. A high quality triangular face is taken to be one that is isotropic while a high quality quadrilateral face is one that is rectangular. It is not clear what configuration constitutes a quality *n-gon*.

In general, it is difficult to assess the quality of a general volume element since, for non-simplex elements, even notions of convexity must be abandoned. However, a nominal requirement is that the element centroid must be located in the interior of the element. The only time this condition is violated for an element that is not obviously "defective" is for a highly anisotropic extruded element with twisted quadrilateral faces.

3.1 Mesh quality improvement strategy

In essence, the face-based notion of mesh quality reduces the determination of element quality to the determination of the quality of each of the faces that constitute the element. Figure 3.1(a) shows an example of elements near a concave region when extruded from the initial surface mesh. As this mesh is extruded, high aspect ratio elements develop that will produce "slivers" in the tetrahedral mesh. Additionally, the transition between the extruded mesh and the tetrahedral mesh will be poor. The solution to this problem, proposed here and elsewhere [24, 25, 26], is to employ a change in the local element topology. Edge collapse, as shown in figure 3.1(b), stops the generation of poor quality elements and provides the opportunity to extrude further into the domain. Figure 3.2(a) shows elements near convex regions of the geometry. Face refinement, as shown in figure 3.2(b), improves the quality of the mesh by making the aspect ratio of the the elements more uniform. However, since these operations may lead to poor quality on the extruded surface of each layer, quality improvement operations should be performed on the extruded surface as well.

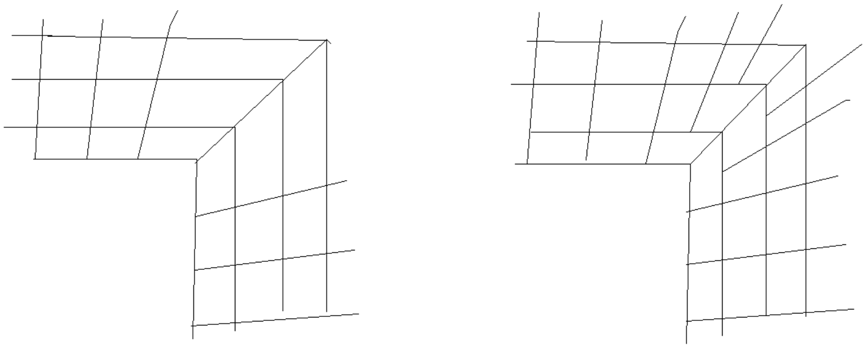
Notice that each of the operations discussed above is applied to the faces of an element, which is consistent with the the notion of element quality proposed above, based on local geometric sensors. The quality improvement process starts with edge collapse followed by face refinement. For complex three-dimensional geometries, these geometric sensors may produce conflicting requirements at a few isolated locations causing faces and edges to be selected for both collapse and refinement. In these cases, edge collapse is given precedence to avoid mesh crossing in concave regions. Elements in each layer of the mesh are made up of faces belonging to the initial surface for the layer, the extruded surface, and the marching faces. Marching faces are those which connect edges in the



(a) Without edge collapse

(b) With edge collapse

Figure 3.1: Edge collapse improves mesh quality near concave regions



(a) Without refinement

(b) With face refinement

Figure 3.2: Face refinement improves mesh quality near convex regions

initial surface mesh to the corresponding edges on the extruded surface mesh. Since the elements are extruded, these faces will be quadrilaterals regardless of the surface mesh topology. Before applying quality improvements, faces in the extruded surface will have the same topology as the initial surface mesh for the layer. The extruded face and marching face topologies may change after applying the quality improvements. It should be noted that these quality improvement operations may be performed independent of the extrusion method used to generate the mesh and are carried out after smoothing the layer and before extruding next layer. These operations may result in a generalized cell topology. The following sections explains the operations in detail. The terms listed below are used in the discussion of the topological operations:

(i) The **Initial/Extruded face aspect ratio** is defined as the ratio of the longest edge to the shortest edge in a face. This is defined for each face in the extruded and initial surface of each layer as shown in the figure 3.3

(ii) The **Marching face aspect ratio (MAR)** is defined by the ratio of the edge length in the marching direction to the edge length on the initial surface. In figure 3.3, the marching face aspect ratio is given by H/B. Marching faces are formed by connecting edges on the initial surface to corresponding edges on the extruded surface.

(iii) The **Reference area (RA)** is an area computed only for the faces in the given initial surface mesh. The same area is treated as reference area for the faces that extrude from the initial surface mesh.

(iv) The **Divergence angle** is defined for each edge as the maximum of angles (θ_1, θ_2) in figure 3.3.

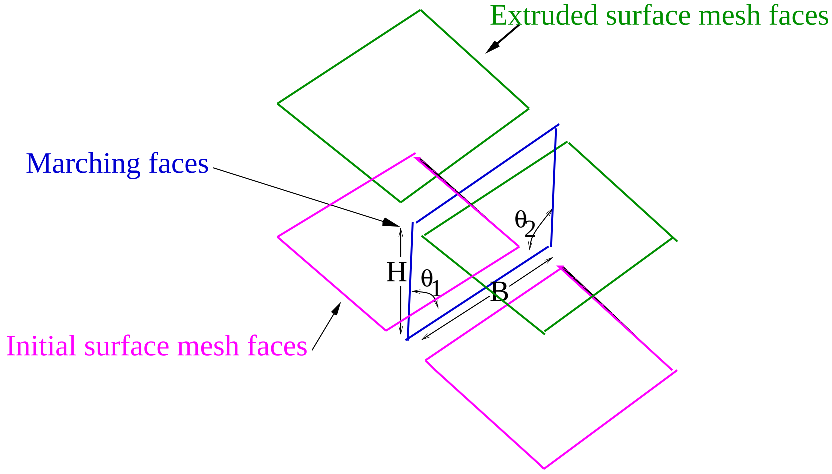


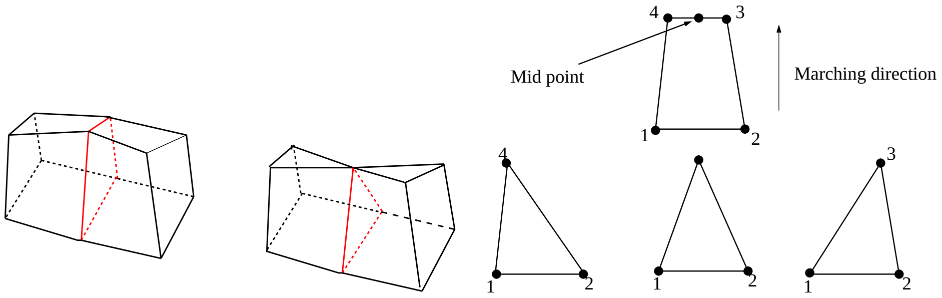
Figure 3.3: Terminology

(v) The **Nodal length scale** is defined as the average length of the edges incident on a node. If N edges are on a node, the nodal length scale is given by

$$\frac{1}{N}\sum_{i=1}^{N} edge\ length_i.$$

3.2 Edge collapse

In [35], where the main aim is to reduce an error metric, the edge collapse operation is extensively used to simplify surface meshes. In this dissertation, the same technique is used to stop the growth of poor quality cells near concave regions and to force the mesh to become more isotropic. Operations like face insertion, face termination [24], and adding degenerate lines in the initial surface mesh [26] are used in the literature. These are dependent on the topology of the initial surface mesh. The advantage of edge collapse is that it is independent of initial surface mesh topology. In the methodology described in this dissertation, the edges are collapsed only on the extruded surface mesh of the layer. The edge collapse operation can reduce a triangular face to an edge or a quadrilateral face to a triangular face. This operation always results in a triangular face



(a) Edge collapse operation-before and after

(b) Three choices to collapse the edge

Figure 3.4: Edge collapse

in the marching direction. Figure 3.4(a) shows the schematic description of the edge collapse operation. There are three ways an edge can be collapsed. The choices are the two ends of an edge and midpoint of an edge as shown in figure 3.4(b). The choice depends on the type of the edge selected. There are four kinds of edges.

(i) Edge not connected to a boundary nodes, i.e., completely internal edge. For this type of edge, the collapse point is chosen such that it maximizes the minimum angle in the valent faces of the two nodes of the edge in the extruded surface mesh.

(ii) Edge where one node is a boundary node. For this type, the boundary node is chosen to maintain boundary integrity.

(iii) Boundary edge where two nodes are boundary nodes and both the nodes have the same boundary condition. For example, if both nodes are marked to lie in same plane, then they have the same boundary conditions. For this type of edge, the collapse point is chosen such that it maximizes the minimum angle in the valent faces of the two nodes of the edge in the extruded surface mesh.

(iv) Boundary edge where two nodes are boundary nodes having different boundary conditions. For example, if the boundary condition is constrained on one node and fixed on the other, then the edge is collapsed to the node having the fixed

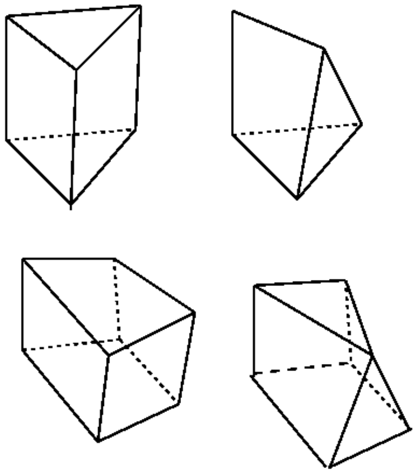


Figure 3.5: Resultant generalized cell topologies when one edge is collapsed in a prism and in a hexahedral cell

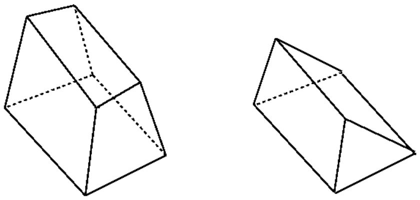


Figure 3.6: Resultant generalized cell topologies when two edges are collapsed in a hexahedral cell

boundary condition. These types of cases are normally encountered at boundaries with sharp corners in the surface mesh.

After edge collapse, the resulting cell topology depends on the number of edges selected for collapse. If the face on the initial surface is a triangle or a quadrilateral and one of its edges is selected for collapse, then the resultant generalized cell topology is as shown in figure 3.5. If two edges are selected in the case of quadrilateral, then the collapse operation shown in figure 3.6 results in a wedge. The edges in the extruded surface mesh to be collapsed are selected if it satisfies one the following criteria:

- **Marching Aspect Ratio:** Each marching face is checked and selected if its MAR value is greater than the specified threshold. The threshold value is specified in such a way that the resultant triangular face in the marching direction is more isotropic. This check is done to identify the "tall and skinny faces" in the marching direction. The ideal value of the threshold is 0.866 which is the length of the bisector of an isotropic triangle. A good range of values is 0.60 to 0.9. This is based on experience. If it is not specified, a default value of 0.70 is taken as a threshold.

- **Reference Area:** The ratio of the area of a face in the extruded surface mesh to the reference area is used to identify converging cells near non-convex regions. For each edge in the extruded surface mesh, an area is computed for both the faces that share the edge. If the ratio of the computed area to the reference area is less than the threshold value for either of the sharing face then the edge is considered for collapse. The threshold is specified as 0.5. This value is based on experience.

The selected edges are assigned priorities according to the MAR and are sorted according to the given priorities such that first edge in the list has the maximum MAR. The list of extruded surface mesh nodes is updated after edge collapse. The edge collapse process is initiated at the first edge in the list. To avoid performing an operation that results in an invalid mesh, all of the edges sharing either of the two nodes of the edge under consideration are marked as non-collapsible edges. The following checks are performed to ensure a valid mesh structure results from edge collapse operation:

- **Folded face check**: This check is done for all the marching faces that share the two nodes of the edge considered for collapse and ensures the centroid lies inside the cell after the collapse operation. The check is performed by taking the dot product of two vectors $\hat{d1} \cdot \hat{d2}$, where $\hat{d1}$ is a vector passing through the cell centers sharing the marching face and $\hat{d2}$ is face normal.

- **Positive face area check**: This check is done to ensure that there are no inverted faces in the extruded surface mesh as a result of edge collapse operation. For

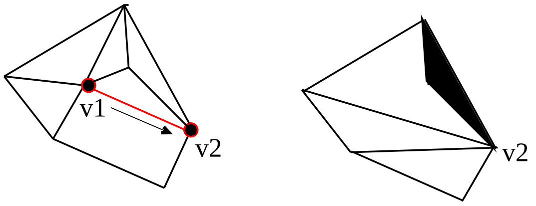


Figure 3.7: Inverted element due to edge collapse operation

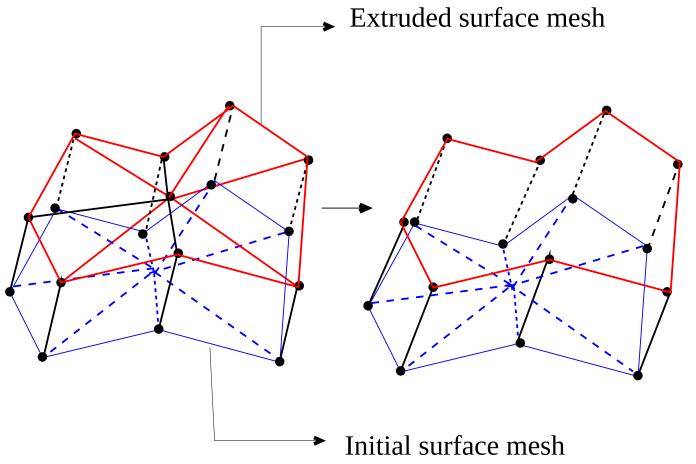


Figure 3.8: Vertex removal operation

example, consider figure 3.7 where an edge made up of (v1,v2) is collapsed to v2. This results in an inverted element.

After each operation the face and cell connectivity is updated to reflect the changes due to edge collapse.

There are a few other topological operations that are similar to edge collapse. Those are vertex removal and face collapse. These operations are not applied due to their limitations. For example, these operations may result in highly non-convex polyhedra or highly skewed marching faces. An example of the vertex removal operation is shown in figure 3.8. In this figure, removal of a vertex forces all its valent faces to be removed. This results in a highly non-convex generalized cell with a polygonal face on the extruded surface mesh.

3.3 Face refinement

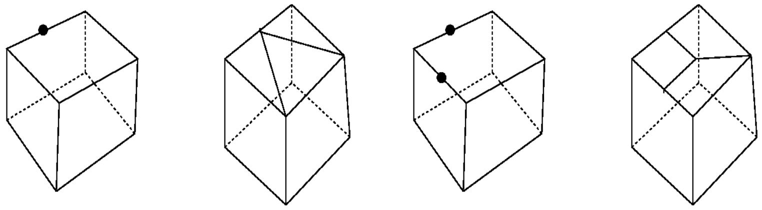
Face refinement is performed to improve the mesh quality near convex regions and to force the mesh to become more isotropic. Faces are refined according to the number of edges marked for refinement. Each marching face is checked and selected for refinement if it satisfies the following criterion:

- **Divergence Angle:** Edges are marked for refinement based on the divergence angle which quantifies the degree of divergence of a mesh face. The divergence angle is calculated for each marching face in the layer. If the angle is greater than the threshold, the edge is marked for refinement. Based on experience, threshold values near 115° are used. The face refinement operation is designed to produce triangles and quadrilaterals on the extruded surface mesh. Figure 3.9 shows face refinement operation according to the number of edges selected for refinement in a hexahedral cell. Similarly, figure 3.10 shows the face refinement operation for a prismatic cell.

The selected edges of the extruded surface are bisected by placing a node at the midpoint and the face topology in extruded surface mesh is updated according to the number of edges selected. The connectivity of the faces in the marching direction is updated from quadrilateral to generalized topology. The selected faces are refined and the cell connectivity is updated. This operation also results in a generalized cell topology.

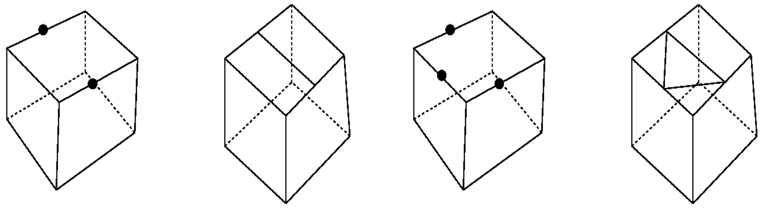
3.4 Extruded surface mesh improvement

The edge collapse operation described above improves the quality along the marching direction. However, this operation does not take into account the quality of the mesh in the extruded surface mesh which may become degraded. Therefore, the following steps are applied locally to the extruded surface mesh to improve the mesh quality. These



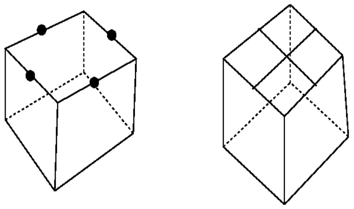
(a) one edge marked

(b) Two consecutive edges marked



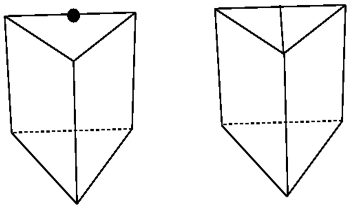
(c) Two non-consecutive edges marked

(d) Three edges marked

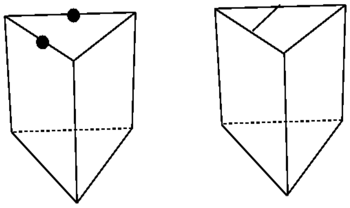


(e) Four edges marked

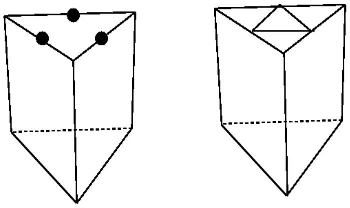
Figure 3.9: Refinement strategies according to the number of selected edges in a hexahedral cell



(a) one edge marked



(b) Two edges marked



(c) Three edges marked

Figure 3.10: Refinement strategies according to the number of selected edges in a prismatic cell

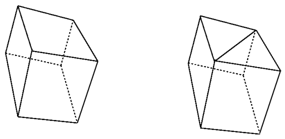


Figure 3.11: Quality improvement by converting a skewed quadrilateral face into two better quality triangular faces

steps are designed in a such a way that the resultant extruded surface mesh will have only triangles and quadrilaterals.

3.4.1 Skewed quadrilaterals

Skewed quadrilateral faces can be converted into two triangles. This operation, shown in figure 3.11, is called splitting. This operation improves the mesh quality in the extruded surface and results in a generalized cell topology. Splitting is also used in the final extruded surface to make it compatible for tetrahedral meshing. This avoids the need of a transition layer between the extruded mesh and tetrahedral mesh.

3.4.2 Skewed triangles

There are two types of skewed triangles. The first type of triangle is like a needle in shape and is shown in figure 3.12(a). This element may result in a poor transition from the near-body to the void-filling mesh. Triangles like these can be eliminated by an edge collapse operation (figure 3.12). The second type of triangle has large obtuse angles (figure 3.13). These triangles may also result in poor transitions from the near-body to the void-filling mesh. The connectivity of these faces is modified by inserting edge “14” as shown in figure 3.14. This operation results in two needle triangles. An edge collapse operation is performed on the newly inserted edge.

(a) Skewed Triangle like “needle” shape

(b) Edge collapse improves the mesh quality locally

Figure 3.12: Quality improvement by edge collapse

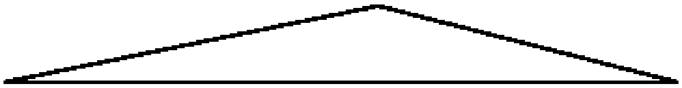


Figure 3.13: Skewed triangle with large obtuse angle

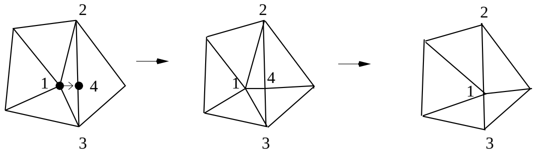


Figure 3.14: Elimination of skewed triangle by inserting an edge and performing an edge collapse

CHAPTER IV

MULTIPLE GEOMETRIES IN CLOSE PROXIMITY

This chapter discusses problems encountered in generating a near-body mesh when there are multiple geometries in close proximity. The first problem encountered occurs when there is insufficient space to grow the near-body mesh. The second problem is smoothness of the transition at the interface of the near-body mesh and the void-filling mesh. The following sections discusses the methodology used to handle these problems in detail.

4.1 Extrusion in the presence of multiple geometries in close proximity

Extruding from different geometries which are in close proximity may lead to mesh crossing as there may not be sufficient space between the geometries to extrude into the domain. There are two methods in the literature that have addressed this problem.

4.1.1 Methodology used in advancing front type schemes

Pirzadeh [13] described an advancing front type method in which an extrusion is terminated locally when the algorithm detects other faces and nodes nearby. Before extrusion, a spring analogy algorithm [13] is used for detecting nearby faces or nodes in each layer. This method may result in premature termination of the near-body mesh in order to provide enough space to fit a void-filling mesh. For example, figure 4.1 shows two bodies, A and B, in close proximity. If these two surface meshes are used to generate

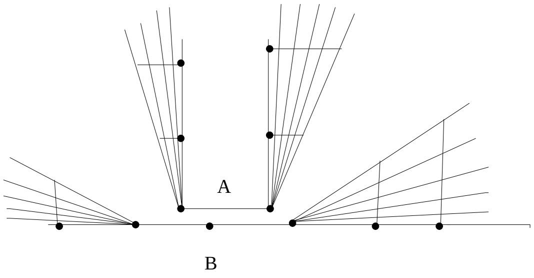


Figure 4.1: Insufficient space to grow near-body mesh

near-body mesh by extrusion, it may result in a zero mesh thickness at a few regions as shown in the figure 4.1.

4.1.2 Methodology used in advancing layer type schemes

Kallinderis [36] described a method used when the near-body mesh is generated using an advancing layer method. In this method, the crossing and/or overlap is avoided by reducing the near-body mesh thickness locally. The normal at each node of the given initial surface mesh is treated as a ray. A ray-triangle intersection test is performed to check if the ray intersects any other faces in the given initial surface mesh. If the ray intersects, then the distance between the node and intersected point is computed. If this distance is less than a predefined tolerance, then the marching distance is reduced accordingly at that node. A procedure to obtain a gradual variation of the marching distances from affected nodes to non-affected nodes is applied. The procedure is performed before starting the near-body mesh generation. The method used in this dissertation is similar to this method with a few modifications.

The following section explains the methodology followed in detail and highlight the differences when compared with two discussed methods.

4.2 Method to handle multiple geometries in close proximity

In this dissertation, the procedure followed to avoid mesh overlap and crossing is to reduce the thickness of the near-body mesh in conjunction with a face refinement operation. This section explains the thickness reduction algorithm. The next section explains the face refinement operation.

If d is the thickness of a near-body mesh, then the minimum thickness of the near-body mesh extruded between any two surface meshes is $2d$. Since a tetrahedral mesh has to be generated between the near-body meshes, it is assumed that a distance of one near-body mesh thickness d is necessary for the tetrahedral mesh. In figure 4.2, if d is the thickness of near-body mesh and D is total distance between two bodies, then, according to the assumption explained above, "D" should satisfy the constraint given by

$$D \geq 3 \times d \ . \tag{4.1}$$

If this condition is not satisfied at any location, then the thickness of the near-body mesh d is reduced to satisfy the constraint. The thickness of the near-body mesh d is given by

$$d = dt \times \sum_{1}^{L} st^{L-1} \ . \tag{4.2}$$

where L is the number of layers, dt is the first point marching distance, i.e., the distance of the first point off the boundary, and st is the stretching factor. The following section describes the procedure used to reduce the near-body mesh thickness automatically to avoid crossing and overlapping.

4.2.1 First step: ray-plane intersection

The first step starts with normal calculations at all the nodes in the given surface mesh. The computed normal at each node is used as a ray and tested to determine if

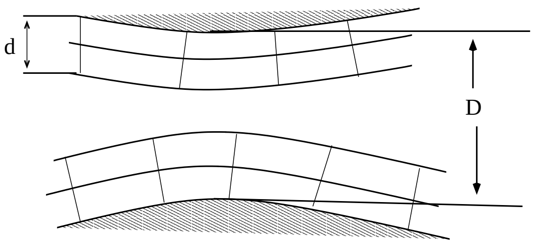


Figure 4.2: Total distance D and near-body mesh thickness d

it intersects any face in the surface mesh. If the ray intersects a face, then the distance D_{new} between the node and the intersected face is computed. If this computed distance satisfies the constraint given in Eq. 4.1, then the next node is considered for the ray-plane intersection test. If it doesn't satisfy the constraint, then the distance D_{new} is used as the new D and the near-body mesh thickness d_{new} at that node is computed using Eq. 4.1. Using this computed near-body mesh thickness d_{new}, the number of layers L and the stretching factor st, the new first point marching distance dt is computed using

$$dt = \frac{d_{new}}{\sum_1^L st^{L-1}}. \tag{4.3}$$

This procedure is done for all the nodes in the surface mesh. As a result, the first point marching distances may not be the same for all nodes in the given surface mesh.

4.2.2 Second step: reduction of initial marching distances

In the step described above, nodes which failed to satisfy the constraint given in Eq. 4.1 are flagged. The initial marching distance dt at the nodes which are neighbors of the flagged nodes are also changed to avoid abrupt changes in the marching layer thickness. The procedure followed is :

- Identify the valent face nodes of the flagged nodes.

- Compute the new first point marching distance at these nodes using Eq.4.4

$$dt_{new} = min(dt_{orig}, C \times dt_{flagged}) \tag{4.4}$$

where dt_{orig} is the original first point marching distance, $dt_{flagged}$ is the reduced first point marching distance of the flagged node. In [36], the constant C is taken to be 1.2. This constant is used to obtain a gradual transition of initial marching distance from flagged to unflagged nodes.

The problem with taking the value of C to be constant is it that assumes the surface mesh is isotropic and has a uniform nodal distribution. For example, consider figure 4.3 in which two bodies “A” and “B” are in close proximity with the surface discretization being non-uniform. Due to the non-uniform discretization, there may be a sudden jump in the marching layer thickness as the constant doesn’t have any knowledge of the surface mesh discretization. This may also lead to mesh crossing if the mesh is extruded further into domain. In order to avoid sudden changes in the marching distances and mesh crossings, properties of the surface mesh discretization are introduced by normalizing the edge lengths. The modified equation is given by

$$dt_{new} = min(dt_{orig}, C \times \frac{L}{L_{max}} \times dt_{flagged}), \tag{4.5}$$

where L is the length of an edge and L_{max} is the maximum edge length in the surface mesh.

- Flag all these nodes

- Repeat the procedure until no more unflagged neighbor nodes are present.

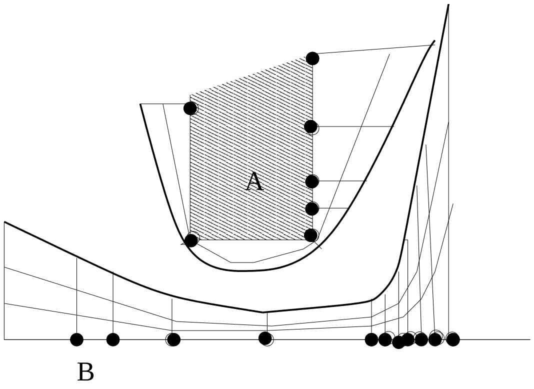


Figure 4.3: Sudden jump in marching layer thickness

4.2.3 Third step: smoothing of initial marching distances

In order to get a smooth variation of the first point marching distances, Laplacian smoothing of the marching step size is performed. Before smoothing, the least first point marching distance (dt_{min}) is determined and used as a boundary condition for smoothing. The Laplacian smoothing is done only for few iteration (20 iterations).

4.2.4 Octree for accelerating ray-plane intersection algorithm

The most expensive component in the process of reducing the marching distance is the ray-plane intersection algorithm. If the ray at each node is tested for intersection with all the faces in the given surface mesh, then the order of the this algorithm is $O(NM)$, where N is the number of nodes and M is the number of faces. As the number of nodes and faces increases, this could be very time consuming and computationally expensive. Therefore, an alternative method is considered. Many algorithms described in [37] are used to accelerate the ray-plane intersections in computer graphics. One such algorithm is a spatial subdivision scheme. There are many spatial subdivision schemes and several well known schemes, i.e, octree, binary, and k-d tree, are discussed in [38].

In this dissertation, the simplest subdivision scheme, an octree, is used for partitioning the space. An octree is a hierarchical data structure. At the top, the root octant represents the box surrounding the entire domain. This root octant is subdivided in to eight octants (Fig. 4.4) and the procedure of subdivision is applied to each resultant octant recursively until a predefined criterion is satisfied. In the present work, the given initial surface mesh is enclosed in an octree and a predefined criterion used to limit the maximum number of faces an octant can enclose. If the number of faces in an octant is more than the limit, the octant is subdivided. This limit is taken as 5 faces so that if the ray intersects the octant, then the number of intersection checks will be at most 5. After enclosing the given initial surface mesh in an octree, ray and octant bounding box overlap test [39] is performed starting from root octant. To perform an overlap test, the length of a ray is taken as $3 \times d$ where d is the near-body mesh thickness. If an octant bounding box doesn't overlap with the ray, then all the faces that are present in the leaves of that octant are not tested for ray-plane intersection. This helps to reduce time spent in the ray-plane intersection test [40].

In [36], the normal from each node is tested for the ray-plane intersection. For example, consider the nodal distribution shown in the figure 4.5 where rays emanating from the nodes of body “A” completely miss the body “B”. Considering only node normals as rays for the ray-plane intersection test may lead to problems. Therefore, along with the node normals, face normals emanating from the face centroid are also considered for ray-plane intersection test. This test is done for all the faces in the surface mesh and it reduces the probability of missing a near-by body completely. If d_n is the near-body mesh thicknesses at a node and d_f is the minimum of the near-body mesh thickness values computed for its valent faces, then the near-body mesh thickness at each node is taken as

$$d_n = min(d_n, d_f). \tag{4.6}$$

By using an octree ($O(NlogM)$), a tremendous speed up is achieved in the ray-plane intersection algorithm when compared with $O(NM)$ algorithm. All these tests are done only once before extrusion occurs.

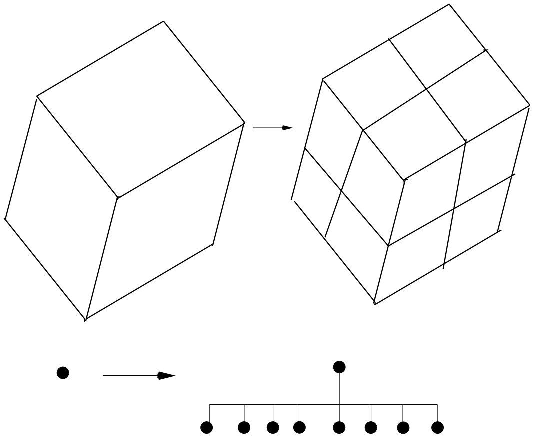


Figure 4.4: Octree data structure

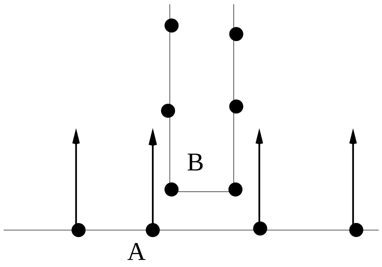


Figure 4.5: Rays from body “A” miss body “B” completely

4.3 Mesh quality at the interface of the near-body & the void-filling mesh

After generating the near-body mesh, the next step is to fill the remaining domain with a void-filling tetrahedral mesh. The quality of the mesh at the interface depends on the smoothness of the transition. For example, in figure 4.6, though there is sufficient space to fit the void-filling mesh, the transition may not be smooth. The extrusion methods in the literature do not give any special consideration to improve the transition at the interface of near-body mesh and void-filling mesh. Therefore these methods may not guarantee a smooth transition at the interface of the extruded mesh and the void-filling mesh.

In order to address this problem, a new global sensor is employed for face refinement. This sensor identifies edges to be refined so that the transition at the interface of near-body mesh and void-filling mesh will be smooth. This procedure is simple, novel, and very easy to implement if an advancing layer scheme is used to generate the extruded mesh. The edges to be refined are identified using the following steps that are performed in each layer of the extrusion:

- First, the smoothness at the interface of near-body and tetrahedral mesh can be measured approximately by the transition ratio at each node of the extruded surface mesh of the last layer. The transition ratio is defined as a ratio of the marching distance of the last layer to nodal length scale (NLS) of the initial surface mesh of the present layer. The curvature is not taken into account as other sensors are used to identify the edges in the regions of curvature variation.

  If the transition ratio is less than the threshold then all the edges sharing the node are considered for next step. The threshold can be user specified and a good range of values is 0.5 to 0.7.

- Second, the starting point of the face refinement operation is determined. As the thickness of the near-body mesh varies, the starting point of the face refinement

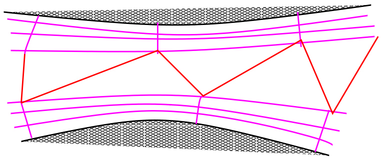


Figure 4.6: Poor transition at the interface of the extruded and the void-filling meshes

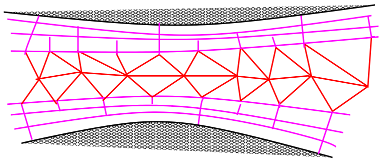


Figure 4.7: Better transition at the interface of the extruded and the void-filling meshes due to global face refinement

also varies. In this step, the refined nodal length scale is used to determine the starting point. The refined nodal length scale is computed by

$$\frac{NLS}{2^{l-l_i}}$$

where NLS is a nodal length scale of the initial surface mesh in the current layer, l is the total number of layers and l_i is present layer number. A new transition ratio is computed using the refined nodal length scale. If the new transition ratio is closer to the threshold and better than the one computed in the first step, then all the edges sharing the node are considered for refinement.

After bisecting the edges that are selected for refinement, the face and cell topology are updated to reflect the results of the face refinement operation as described in chapter 3. By applying this operation the resultant transition from the extruded mesh to the tetrahedral mesh shown in figure 4.7 is much better than the transition shown in figure 4.6. This operation also saves a few points by eliminating the need to refine the given initial surface mesh.

CHAPTER V

RESULTS

A few example meshes are now presented to demonstrate the approach described in the text. The volume mesh generated for each example is output in the face-based COBALT mesh format [41]. VTK [42] is used for visualizing the volume meshes. The void-filling tetrahedral mesh is generated using the in-house software package SolidMesh developed by Marcum [43]. Plots of the face-based quality metrics for the outer most layer of each mesh (which will be used as a starting point to generate the void-filling mesh) are included to demonstrate the effectiveness of the quality improvements. Plots of the extruded surface aspect ratio are included to illustrate the deviation from isotropy. Plots of the marching aspect ratio help quantify the smoothness of the transition between the extruded mesh and the void-filling mesh. Since plots of the extruded surface aspect ratio alone do not provide sufficient information about the skewness of the faces in the extruded surface mesh, plots of the minimum angles for quadrilaterals and triangles are also included. In all the cases, the mesh is extruded until the marching spacing is comparable with the given surface mesh spacing. This is achieved by computing the number of layers as an average of the layers required by smallest and largest edges in the given surface mesh to achieve good transition to tetrahedral mesh. The CFD solutions presented here are included to show that the example meshes presented were useful. These solutions are not intended to represent an extensive validation of generalized meshes.

5.1 Surface of revolution

The first example is a simple surface of revolution. This example is included to depict clearly the effects of additional smoothing near concave regions and of the quality improvement operations. The surface mesh shown in figure 5.1 has 1320 nodes with 1296 quadrilateral faces. A volume mesh consisting of 14520 nodes, 40416 faces, and 12960 hexahedral elements is generated by extruding ten layers without additional smoothing. This results in mesh crossing as shown in figure 5.2. Figures 5.3(a) and 5.3(b) show the cross-section and outer surface of the volume mesh generated by extruding the same number of layers with additional smoothing. In this mesh, the faces emanating from concave regions adjust due to smoothing to avoid crossing. Figure 5.3(a) also indicates that the mesh faces remain mostly orthogonal except in the concave and convex regions of the geometry. Figure 5.3(c) shows the quality plots for the last layer. Due to clustering of the lines emanating from the concave regions, several faces on the extruded surface mesh of the last layer have aspect ratios greater than 10 and a few faces in the marching direction have aspect ratios greater than 2.5. This is more prominently visible on the outer surface of the volume mesh as shown in figure 5.3(b).

The cross section and the outer surface of the volume mesh consisting of 15840 nodes, 44492 faces, 220 prisms, 13866 hexahedra, and 232 generalized elements, generated by extruding the same number of layers with quality improvements are shown in figures 5.4(a) and 5.4(b), respectively. Face refinement is triggered near the sharp corners in the surface mesh which increases the number of elements. As the mesh is extruded further, edge collapse is triggered and eliminates a few faces in each layer. The net result is that the number of faces in the outer surface mesh decreases from 1296 to 1248 while the total number of elements increases from 12960 to 14318. This result is primarily due to the fact that the layer extrusion distance is increased in each layer so that more elements are located near the initial surface mesh where refinement occurs. Quality plots of the last layer are shown in figure 5.4(c). These plots indicate that the maximum extruded surface aspect ratio is less than 2.8 and the maximum marching aspect ratio is less than 1.3. The quality plots indicate improvement in mesh quality according to the face-based quality metrics. Additionally, the mesh can be extruded further without crossing due to the quality improvement operations. The cross section and outer surface of the volume mesh consisting of 21216 nodes, 61688 faces, 848 prisms, 19088 hexahedra, 312 generalized elements obtained after extruding 16 layers are shown in figures 5.5(a) and 5.5(b). These figures show a more uniform mesh on the outer surface. The cross section shows clearly how edge collapse and face refinement work as the mesh is extruded.

5.2 Two spheres in close proximity

The second example geometry is two spheres in close proximity as shown in figure 5.6. This example was chosen to illustrate how the near-body mesh thickness reduction algorithm and face refinement improve mesh quality for two sphere in close proximity. The surface mesh of two spheres consists of 1160 nodes and 2312 triangular faces. The region between the two spheres is less than average edge size of the surface mesh. If a volume mesh is generated by extruding from the surface mesh with out considering the proximity of the second sphere, the resultant near-body mesh may overlap as shown in figure 5.7.

In order to avoid mesh overlapping, a volume mesh is generated by extruding the same number of layers using the automatic reduction of near-body mesh thickness algorithm. New distances of the first point off the boundary are obtained using the algorithm discussed in chapter 4. A shaded contour plot of the new distances of the first point off the boundary as a scalar function for each node is shown in figure 5.8. This contour plot indicates the least marching distance occurs near the gap where the color is blue and it gradually increases to the white color where the marching distance is a maximum. A surface mesh consisting of 198 nodes and 392 triangular faces is considered as the far field and the region between the extruded mesh and the far field is filled with a tetrahedral mesh using SolidMesh [43]. The resultant volume mesh consists of 19211 nodes, 124246 faces, 25432 prisms, and 28501 tetrahedral cells. The cross section of the mesh is shown in figure 5.9(a). Quality plots of the last layer of the extruded mesh are shown in figure 5.9(c). These plots indicate that the marching aspect ratio of all the faces is less than 0.38 which leads to sudden jump in the cell sizes at the transition from near-body mesh to tetrahedral mesh as shown in figure 5.9(b). This jump in cell sizes is expected irrespective of the tetrahedral mesh generator used unless the given initial surface mesh is refined.

A volume mesh consisting of 95252 nodes, 694501 faces, 46423 prisms, 241187 tetrahedra, and 22445 generalized elements is generated by extruding the same number of layers with quality improvements. A cross section and details of the cross section near the gap region are shown in figure 5.10(a) and figure 5.10(b), respectively. Figure 5.10(b) shows the transition from extruded mesh to tetrahedral mesh near the gap region. This figure visually indicates a better transition from extruded mesh to tetrahedral mesh when compared with the mesh generated without quality improvements shown in figure 5.9(b). The number of triangles in the outer surface mesh has increased from 2312 to 26772. The increase in number of triangles is due to the face refinement operation performed to achieve a smooth transition from the extruded mesh to the tetrahedral mesh. Quality plots of the last layer of the extruded mesh are shown in figure 5.10(c). These plots indicate improvement in marching aspect ratio, where most of the faces fall in range of 0.5 to 0.8. The minimum angle in the given initial surface mesh is 35 degrees. To obtain a smooth transition at the interface of near-body mesh and tetrahedral mesh, the face refinement operation is triggered in 7^{th} layer. The resultant extruded surface mesh of one of the spheres visualized from inside the sphere is shown in figure 5.11. The triangles marked are those with included angles less than 35 degrees. These are present due the termination of face refinement in that layer. When the surface mesh is used as the initial surface mesh for the next layer, these marked triangles are refined and result in additional triangles with included angles less than 35 degrees. Figure 5.12 shows a series of extruded surface meshes generated in each layer with the marked triangles by those with included angles less than 35 degrees. These figures indicate the number of marked triangles increased as the mesh is extruded. This effect is shown in the minimum angle quality plot. The surface meshes in figure 5.12 also indicate the progressive nature of face refinement operation which starts from the minimum near-body mesh thickness region. Figure 5.12(c) shows the smooth variation of triangle size with the minimum being at the least marching distance. Therefore, face refinement improves the transition from extruded mesh to tetrahedral mesh, saves many points by eliminating the need to refine the given initial surface mesh to achieve a smooth transition from extruded mesh to a tetrahedral mesh but, in the process, may result in the introduction of poor quality triangles.

5.3 M6 wing

The third example considered here is an ONERA M6 wing geometry [44]. The surface mesh, consisting of 14000 triangles and 7040 nodes, is generated using SolidMesh [43]. A volume mesh consisting of 464640 nodes, 2291535 faces, 910000 prisms is generated by extruding 65 layers without quality improvements. Figures 5.13(a) and 5.13(b) show the outer surface and symmetry plane of the volume mesh. Figure 5.13(c) shows quality plots of the last layer of the volume mesh. The plots indicate a few triangles with minimum angle less than 10° and a maximum extruded surface aspect ratio greater than 9. The mesh faces diverge near the trailing edge and leading edge as the mesh is extruded as can be observed in 5.13(b). This results in skewed triangles. A volume mesh consisting of 532685 nodes, 2621537 faces, 1020239 prisms, and 19185 generalized elements is generated by extruding the same number of layers with quality improvements. Figures 5.14(a) and 5.14(b) show the outer surface and symmetry boundary of the volume mesh. Figure 5.14(c) shows quality plots of the last layer of the volume mesh. The plots indicate an improvement in the minimum angles and a decrease in the maximum extruded surface aspect ratio from 9 to 5. The outer surface mesh in figure 5.14(a) shows the effect of the face refinement operation employed to improve the transition from the extruded mesh to tetrahedral mesh when compared with the surface mesh in figure 5.13(a). Figure 5.14(b) shows the effect of refinement at the trailing edge of the wing. As the geometry is completely convex in nature, a volume mesh consisting of 574262 nodes, 2852521 faces, 1109698 prisms and 29486 generalized elements can be generated by extruding far from the body without using any tetrahedral elements. Figures 5.15(a) and 5.15(b) show a cross section of the extruded mesh and mesh at the symmetry boundary of this volume mesh. Dr. Edward A. Luke [45] computed a transonic viscous simulation using this mesh with the CHEM code [46] and compared the results with experimental data. The following are the conditions used for the simulation. A freestream Mach number of 0.8395, the angle-of-attack of 3.06 degrees and a Reynolds number of 11.72×10^6 based on mean chord length of 0.64607 meters. The results of the solution were compared to experimental pressure measurements taken at seven span stations [45]. The C_P plots (courtesy of Dr. Luke) at three sections of the span are presented in figure 5.16. These plots show good agreement with the experimental data [44]. This represents a preliminary validation effort for meshes containing generalized elements.

5.4 X38 geometry

The fourth example is an X38 geometry (courtesy of Dr. M. G. Remotigue). The initial X38 surface mesh consisting of 38008 triangular faces, and 19006 nodes, shown in figure 5.17, is generated using SolidMesh [43]. A volume mesh consisting of 855270 nodes, 4218888 faces, and 1672352 prisms is generated without quality improvements by extruding 43 layers. Three cross sections and the outer surface of the volume mesh are shown in figures 5.18(a) and 5.18(b), respectively. The clustering of the faces near the concave regions of the mesh can be seen in figure 5.18(b). The faces near convex region become skewed as the mesh diverges as shown in figure 5.19(a). This mesh also crosses near the rudder region of the geometry due to the locally concave nature of the geometry. The quality plots for the last layer, shown in figure 5.19(b), show extruded surface aspect ratios greater than 13 and marching aspect ratios greater than 6. The plots indicate that the outer surface mesh has a few triangular faces with angles less than 5°. This is because the triangular faces near concave/convex regions become more skewed during the extrusion process.

The cross section and outer surface of the volume mesh consisting of 1033030 nodes, 5030406 faces, 1906003 prisms, and 70345 generalized elements generated by extruding the same number of layers with quality improvements are shown in figures 5.20(a) and 5.20(b), respectively. In the outer surface, the number of faces is increased from 38008 to 89536 and the number of nodes is increased from 19006 to 44770 nodes. This increase in number of faces and nodes is due to quality improvement operations. The surface mesh shown in figure 5.21(a) indicates better quality mesh when compared with figure 5.19(a) visually. The quality plots for the last layer of the mesh generated are shown in figure 5.21(b). The quality plots indicate the maximum extruded surface aspect ratio is less than 4.5 and the maximum marching aspect ratio is less than 3.5. The minimum angles for triangular faces are also improved.

A viscous flow simulation is performed for this geometry using Cobalt [47]. A box is taken as a far field boundary. The region between the extruded mesh generated with quality improvements and the far field is filled with isotropic tetrahedral elements using SolidMesh [43]. The mesh in the symmetry plane and a cross section perpendicular to the body axis just aft of the wing leading edge of the volume mesh are shown in figure 5.22. This mesh consists of 1303959 nodes, 8458758 faces, 1906003 prisms, 1736239 tetrahedra, and 70345 generalized elements. These cross sections of the volume mesh indicate a smooth transition from the extruded mesh to the isotropic tetrahedral mesh. The flow conditions are those given by Berry [48] at approximately 110,000 ft altitude. A Mach number of 9.7, and an angle of attack of 20 degrees are used. A Reynolds number of 5.03284×10^4 per inch with a perfect gas is used. The results from the simulation are presented in figure 5.23.

5.5 Tactical aircraft

The fifth example considered here is a tactical aircraft geometry (courtesy of T. Michal, Boeing). The relatively coarse surface mesh with sharp convex and concave regions consists of 6779 nodes and 13554 triangles. The geometry is shown in figure 5.24. This example is included to show that the quality improvement operations are effective independent of geometric complexity. The resulting volume mesh consisting of 13519 nodes, 2046654 faces, and 813240 prisms, is generated by extruding 60 layers. Several cross sections and the outer surface of the volume mesh are shown in figures 5.25(a) and 5.25(b), respectively. The faces near convex and concave regions become skewed as the mesh is extruded as shown in figure 5.25(b). This mesh also crosses at the intersection of the wing and fuselage due to concave nature of the geometry as shown in figure 5.25(a). The quality plots for the last layer shown in figure 5.25(c) indicate a few elements with extruded surface aspect ratio greater than 27, marching aspect ratios greater than 9, and included angles less than 5°.

Three cross sections and the outer surface of the volume mesh, consisting of 612082 nodes, 3036325 faces, 1178266 prisms and 30004 generalized elements generated by extruding the same number of layers with quality improvements, are shown in figures 5.26(a) and 5.26(b), respectively. In the outer surface, the number of faces is increased from 13554 to 15650 and the number of nodes is increased from 6779 to 7827 nodes. This increase in the number of faces and nodes is due to the quality improvement operations. The quality plots for the last layer of the mesh generated are shown in figure 5.26(c). The quality plots indicate the maximum extruded surface aspect ratio is less than 4.0 and the maximum marching aspect ratio is less than 3.0. The minimum angles in the triangular faces are also improved. A sphere is used as the far field boundary. The region between the extruded mesh generated with quality improvements and the far field is filled with isotropic tetrahedral elements using SolidMesh [43]. A cross section in the symmetry plane is shown in figure 5.27. This figure indicates smooth transition from the extruded mesh to isotropic tetrahedral mesh. A viscous simulation is performed on this mesh using Cobalt [47]. The conditions of the simulation are, a Mach number of 0.5, an angle of attack of 20 degrees, and a Reynolds number of 3.36615×10^6 per foot. The pressure plot is shown in figure 5.28.

5.6 Single element injector

The sixth example is a single element injector geometry as shown in figure 5.29. This geometry is included to show the effectiveness of the marching distance reduction algorithm along with quality improvement operations. The coaxial injector has oxygen at the core surrounded by hydrogen. The section that carries hydrogen narrows before the combustion chamber as shown in figure 5.29(d). This region is called the post. Only a quarter of the geometry is taken to reduce the number of nodes and faces. A surface mesh consisting of 24822 nodes and 47912 triangles is generated using SolidMesh [43]. Figure 5.30 shows a close up of the surface mesh near the post region. If the mesh is extruded without any consideration of the geometry in close proximity, mesh crossing may occur near the post region. In order to avoid mesh folding, the marching distance is reduced using the automatic marching reduction algorithm discussed in chapter 4. Figure 5.31 shows a contour plot of the marching distance as a scalar function at each node. The color map in this figure indicates the least marching distance is at the post region and it gradually increases from there.

A volume mesh consisting of 1613430 nodes, 7769128 faces, and 3066368 prisms is generated without quality improvements by extruding 64 layers. The remaining space is filled with a tetrahedral mesh. Figures 5.32(a) and 5.32(b) show cross section and details of the cross section near the post region respectively. These figures indicate the reduction in marching distance helps avoid crossing and the marching distance growth varies smoothly. Figure 5.33 shows more details of the cross section before and after the post region of the geometry. The quality plots for the last layer of the mesh generated are shown in figure 5.34. The quality plots indicate the presences of a few triangles with extruded surface aspect ratio greater than 12 and several faces with marching aspect ratio less than 0.2. Therefore, the transition from the extruded mesh to tetrahedral mesh is not smooth. This can also be visually observed from figure 5.32.

A volume mesh consisting of 2182369 nodes, 9953842 faces, 3585182 prisms and 184326 generalized elements is generated by extruding the same number of layers with quality improvements. The remaining space is filled with a tetrahedral mesh. Figures 5.35(a) and 5.35(b) shows the cross section and details of the cross section near the post region, respectively. Figure 5.36 shows additional details of the cross section before and after the post region of the geometry. This figure indicates a better transition from the extruded mesh to the tetrahedral mesh when compared with the transition shown in figure 5.33. The quality plots for the last layer of the mesh are shown in figure 5.37. The plots indicate the maximum extruded surface aspect ratio is less than 6 and improved marching aspect ratios. These plots indicate a smooth transition from the extruded mesh to isotropic tetrahedral mesh. This can also be visually observed from figure 5.35.

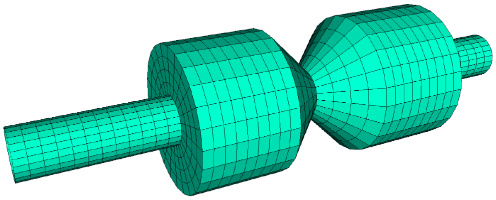


Figure 5.1: Surface of revolution: initial surface mesh

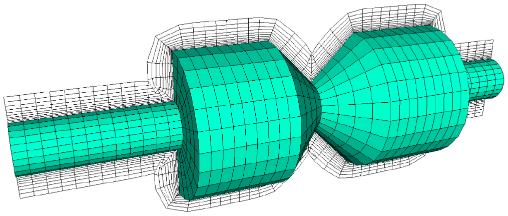
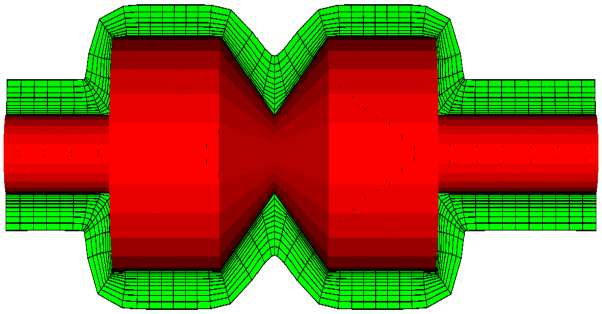
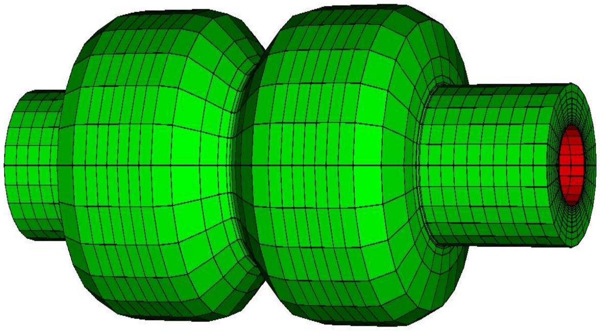


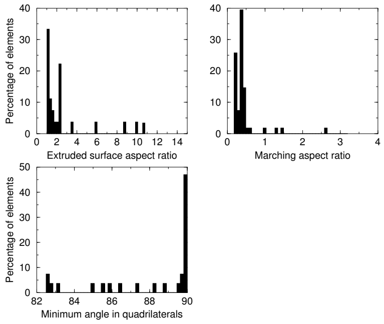
Figure 5.2: Mesh crossing without additional smoothing in concave regions



(a) Cross-section of the volume mesh

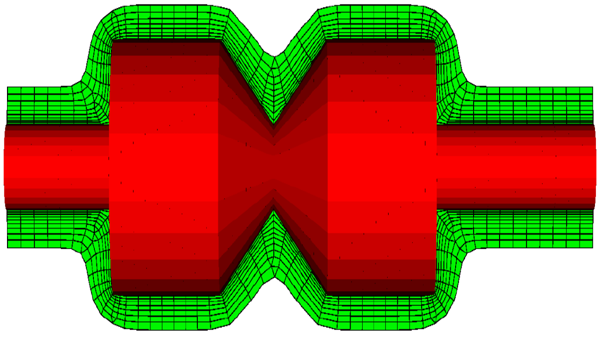


(b) Outer surface mesh of the volume mesh

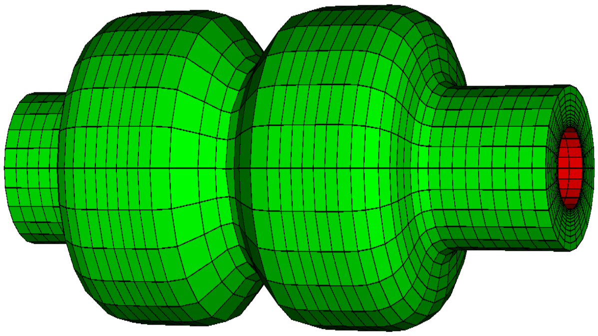


(c) Quality plots of the last layer
of the volume mesh

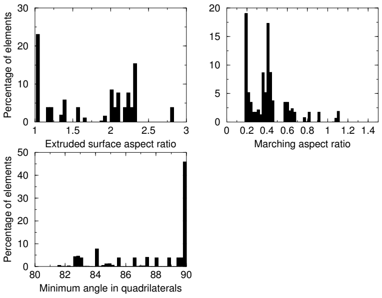
Figure 5.3: Volume mesh for surface of revolution generated by extruding ten layers without quality improvements



(a) Cross-section of the volume mesh

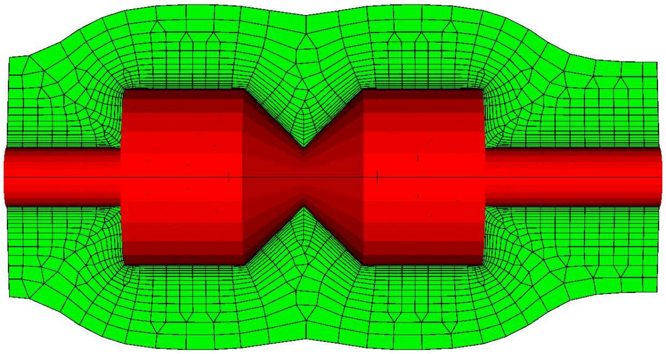


(b) Outer surface mesh of the volume mesh

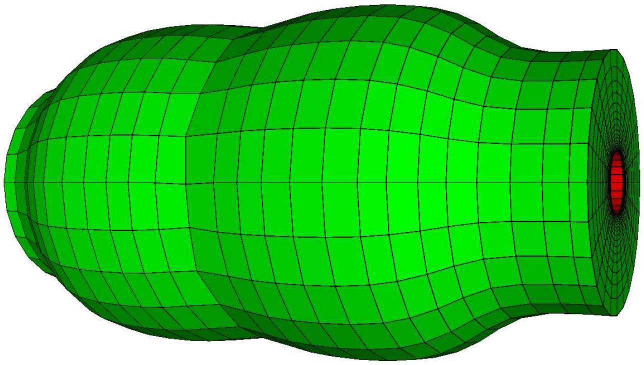


(c) Quality plots of the last layer of the volume mesh

Figure 5.4: Volume mesh for surface of revolution generated by extruding ten layers with quality improvements



(a) Cross section of the volume mesh



(b) Outer surface of the volume mesh

Figure 5.5: Volume mesh generated by extruding far from initial surface mesh.

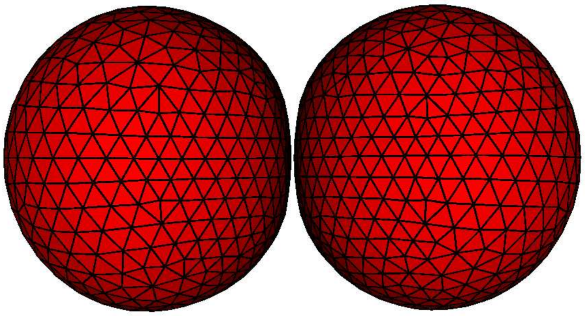


Figure 5.6: Two spheres in close proximity

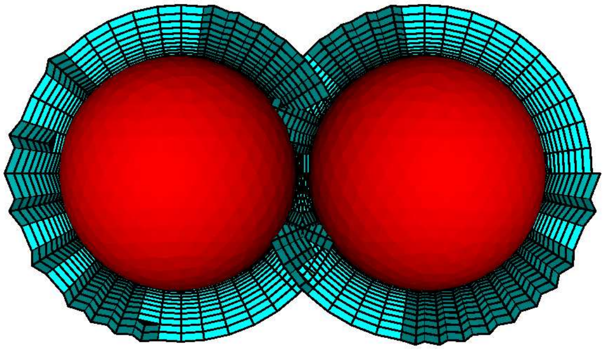


Figure 5.7: Cross section of the volume mesh generated without considering the proximity of the two spheres

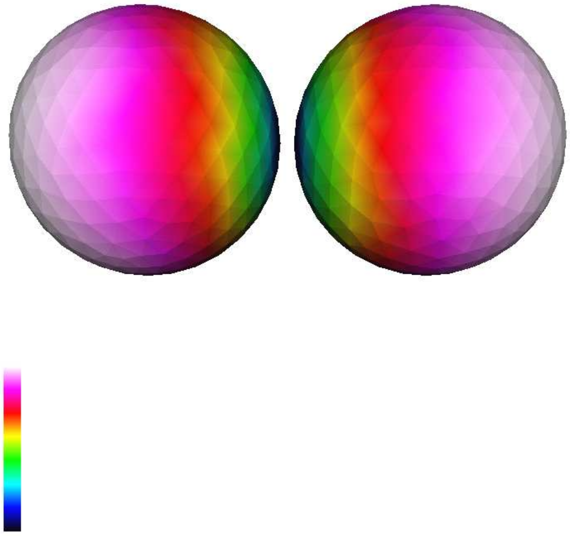
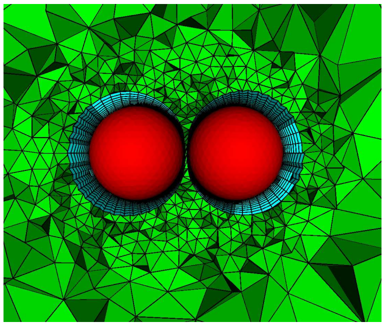
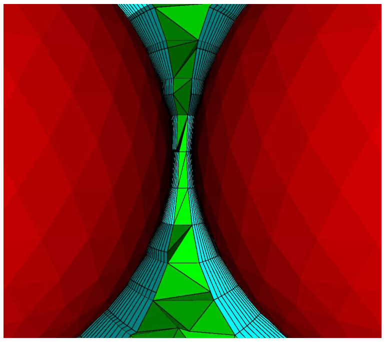


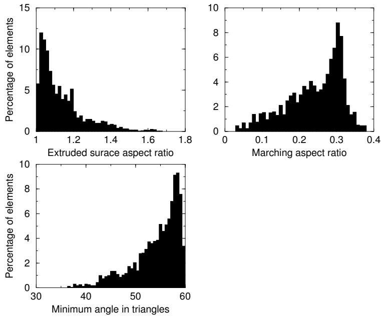
Figure 5.8: Shaded contour plot of marching distances as a scalar function



(a) Cross section of the volume mesh

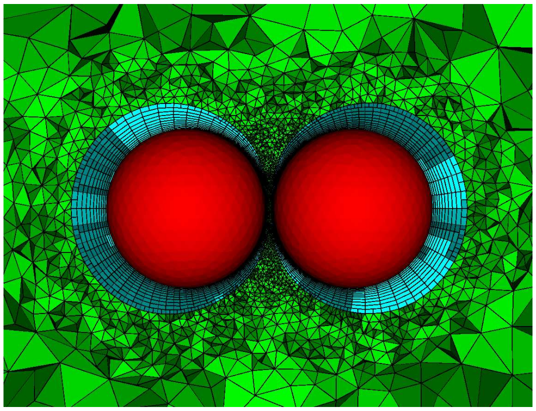


(b) Details of the cross section near gap

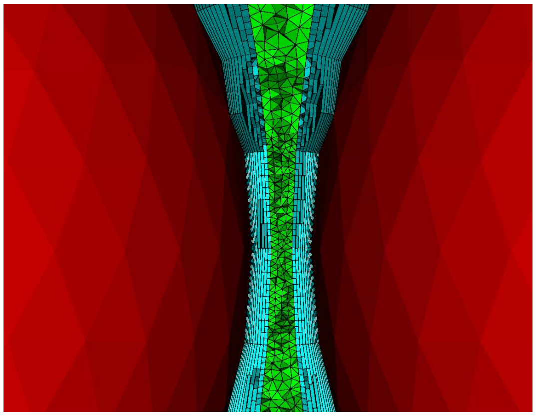


(c) Quality plot of the last layer of the extruded volume mesh

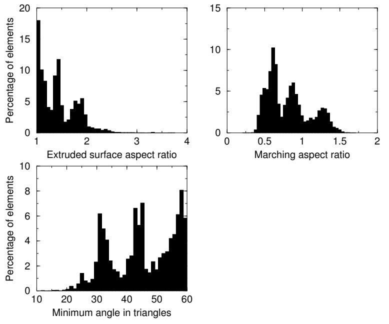
Figure 5.9: Cross section of a volume mesh for the two-sphere geometry



(a) Cross section of the volume mesh

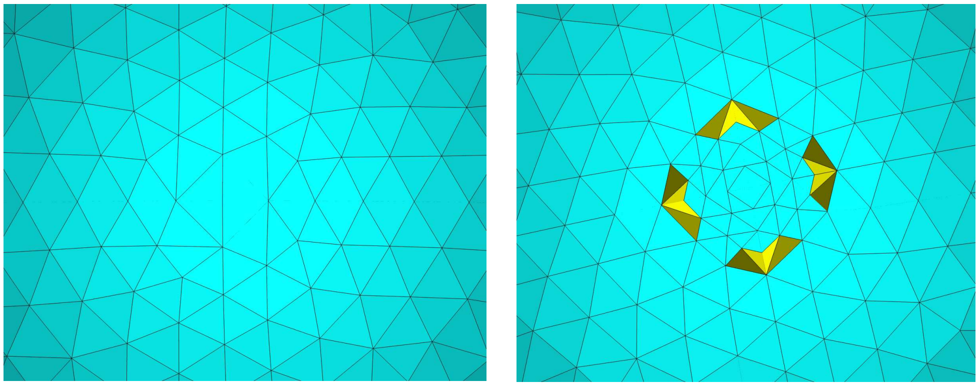


(b) Details of the cross section near gap



(c) Quality plot of the last layer
of the extruded volume mesh

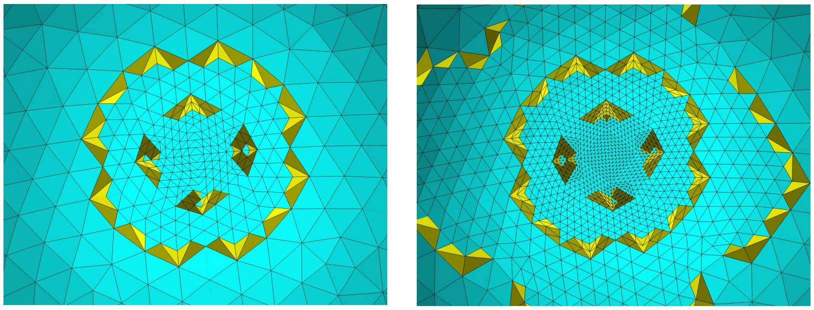
Figure 5.10: Cross section of a volume mesh for the two-sphere geometry



(a) Initial surface mesh

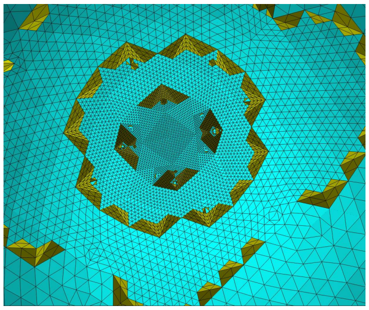
(b) Extruded surface mesh

Figure 5.11: Initial and extruded surface meshes for the 7^{th} layer of the near-body mesh for the two-sphere geometry



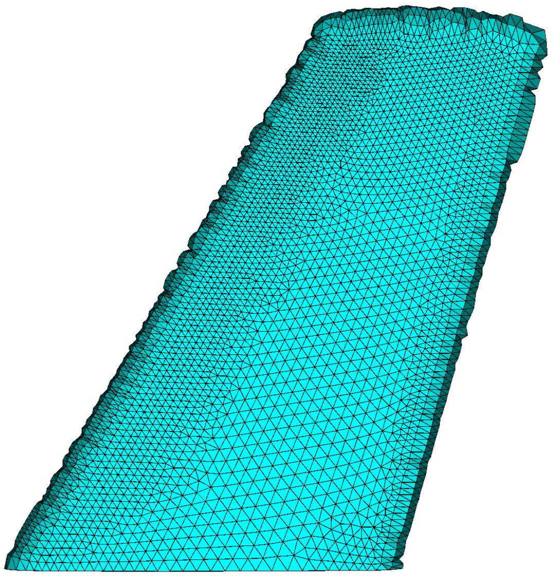
(a) Extruded surface mesh of 8^{th} layer

(b) Extruded surface mesh of 9^{th} layer

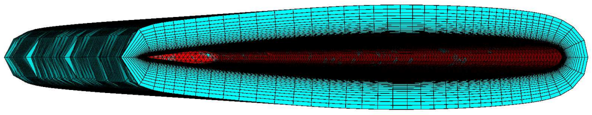


(c) Extruded surface mesh of 10^{th} layer

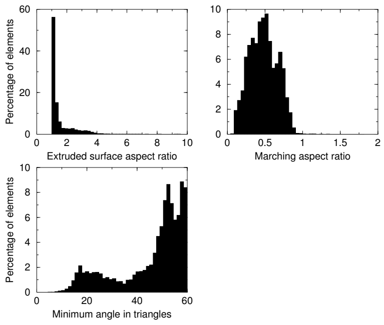
Figure 5.12: Extruded surface meshes of the near-body mesh for the two-sphere geometry



(a) Outer surface of the volume mesh

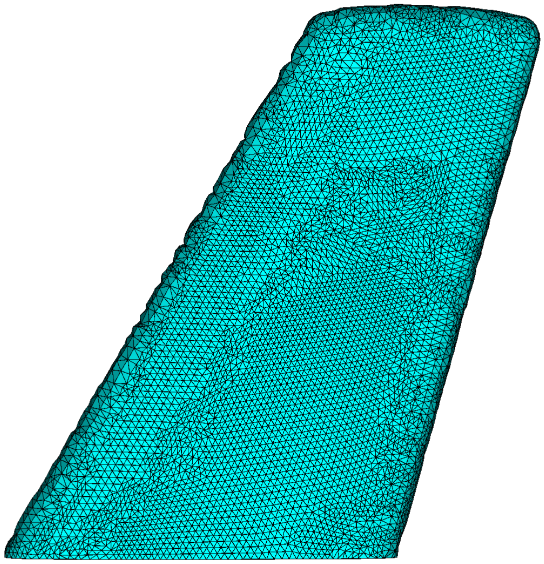


(b) Symmetry boundary of the volume mesh

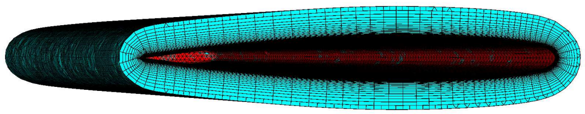


(c) Quality plot of the last layer of the volume mesh

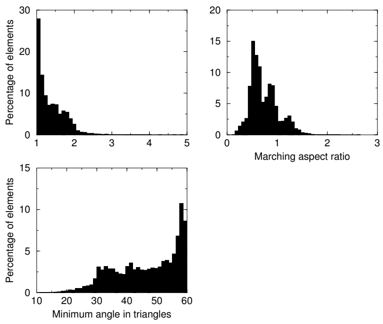
Figure 5.13: Volume mesh for an ONERA M6 geometry generated by extrusion without quality improvements



(a) Outer surface of the volume mesh

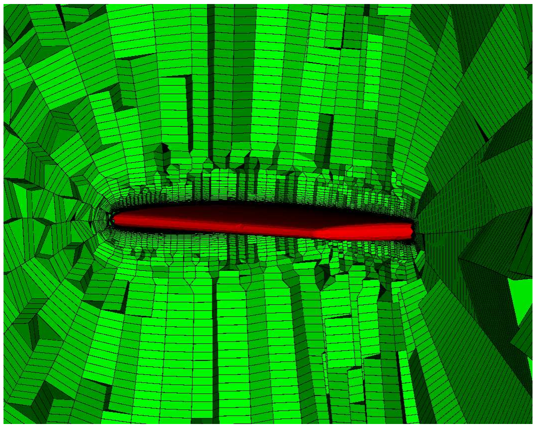


(b) Symmetry boundary of the volume mesh

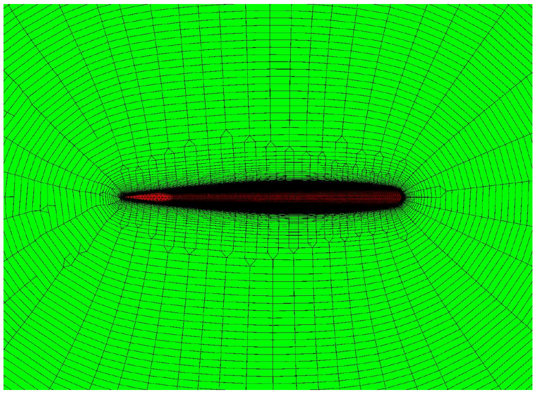


(c) Quality plot of the last layer of the volume mesh

Figure 5.14: Volume mesh for an ONERA M6 geometry generated by extrusion with quality improvements

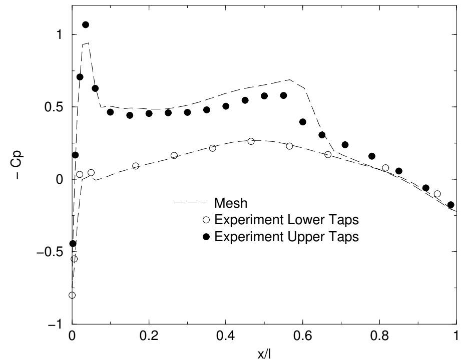


(a) Cross section of the volume mesh

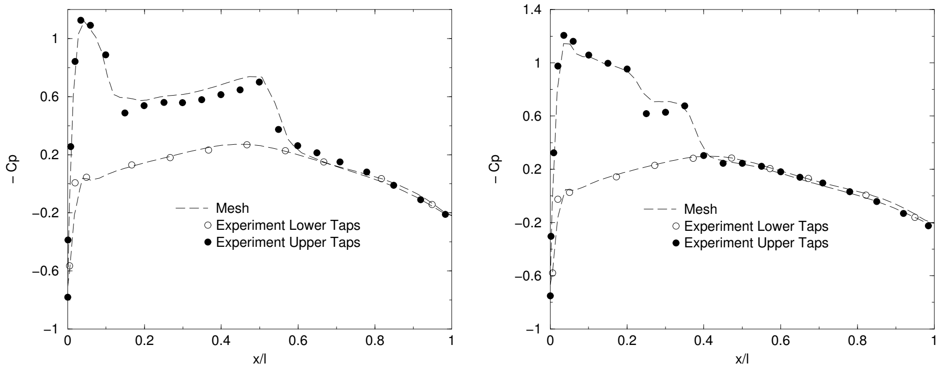


(b) Symmetry boundary of the volume mesh

Figure 5.15: Complete viscous volume mesh for an ONERA M6 wing geometry



(a) ONERA M6 wing, C_p at y/b=0.20



(b) ONERA M6 wing, C_p at y/b=0.44 (c) ONERA M6 wing, C_p at y/b=0.80

Figure 5.16: C_p plots at three span sections of ONERA M6 wing geometry

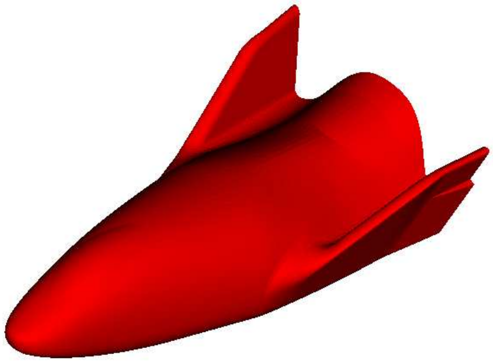
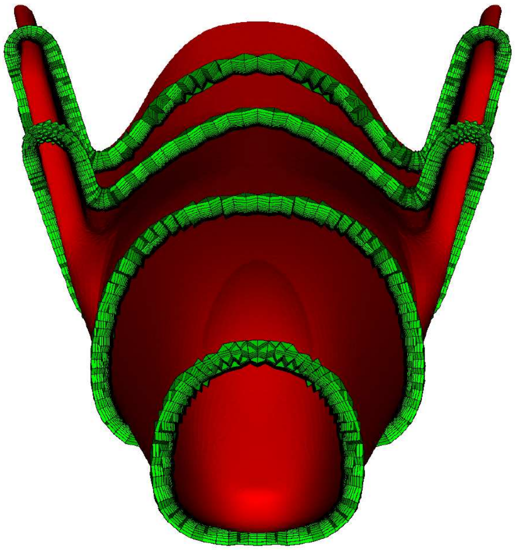
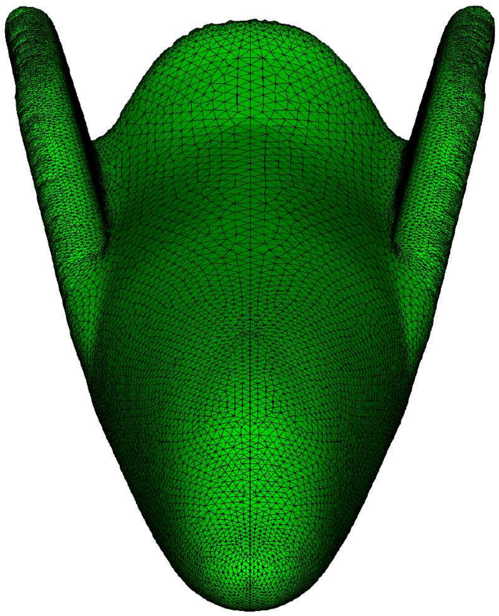


Figure 5.17: X38 geometry

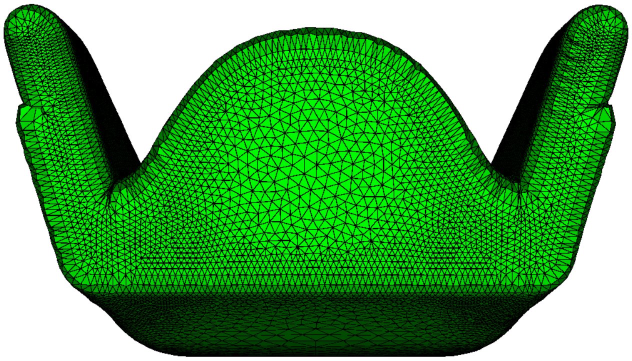


(a) Different cross sections of the volume mesh

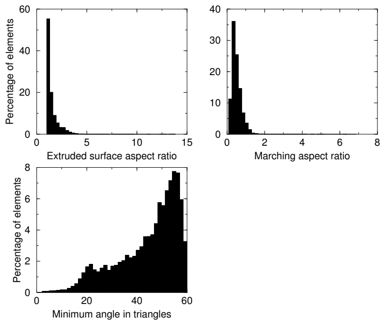


(b) Outer surface of the volume mesh

Figure 5.18: Volume mesh generated without quality improvements

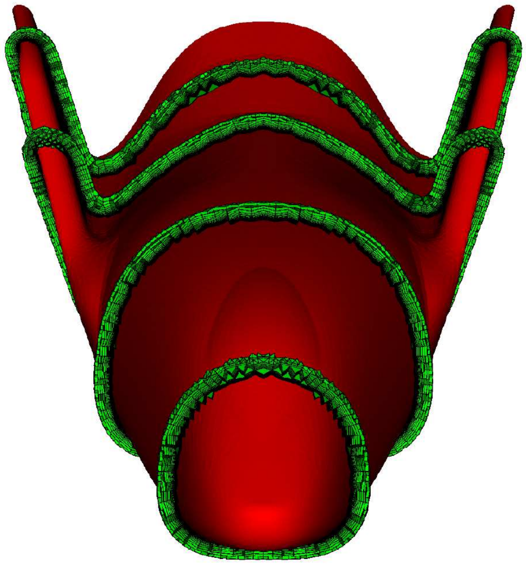


(a) Back end of the volume mesh

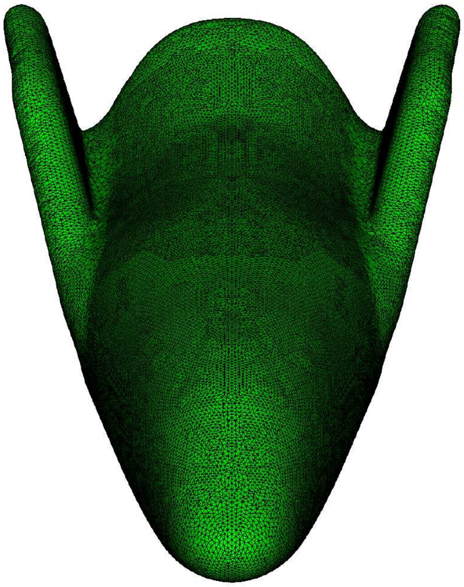


(b) Quality plots of the last layer of the volume mesh

Figure 5.19: Volume mesh generated by extrusion without quality improvements

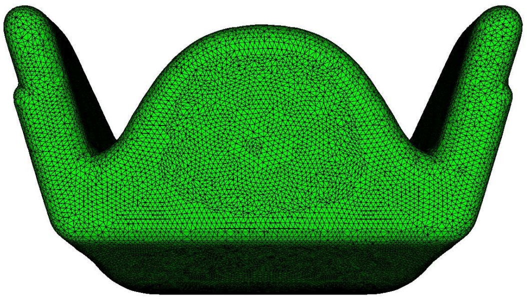


(a) Different cross sections of the volume mesh

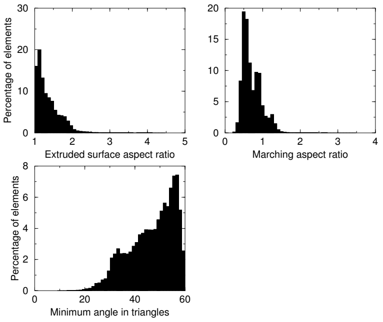


(b) Outer surface of the volume mesh

Figure 5.20: Volume mesh generated with quality improvements

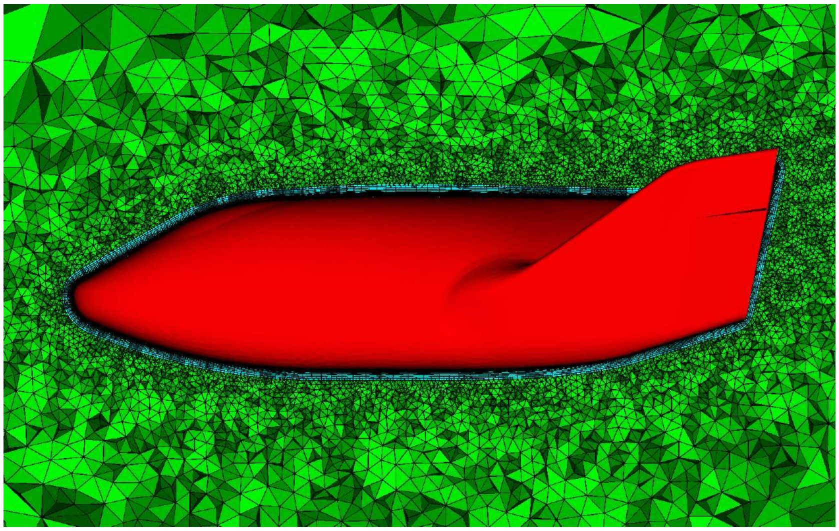


(a) Back end of the volume mesh

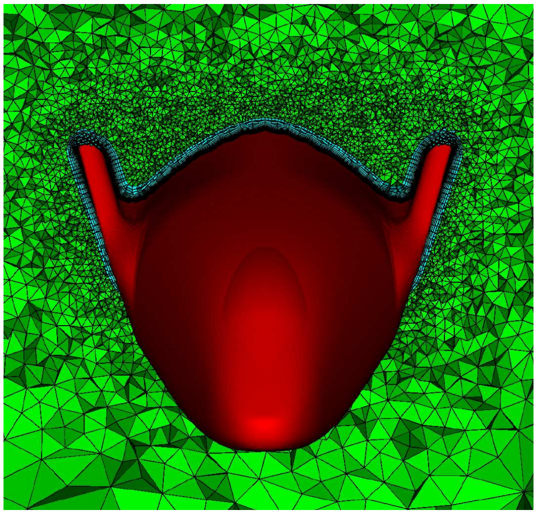


(b) Quality plots of the last layer of the volume mesh

Figure 5.21: Volume mesh generated by extrusion with quality improvements

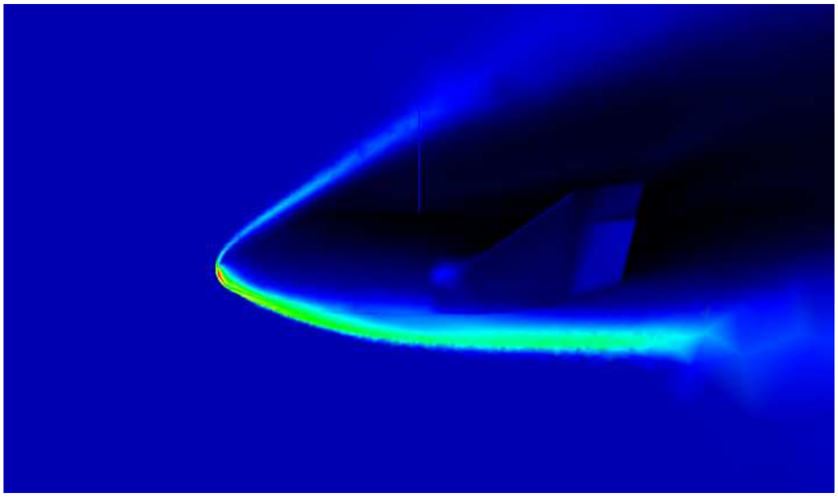


(a) Cross section at symmetry plane

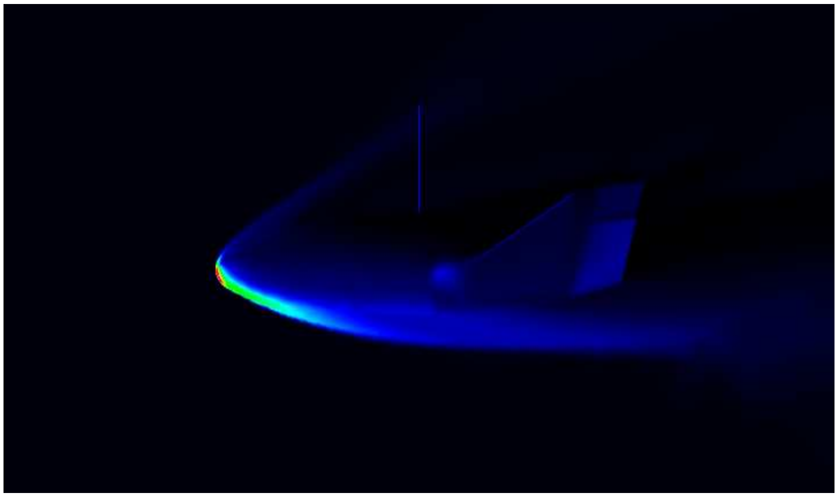


(b) Cross section across wings

Figure 5.22: Complete viscous volume mesh for the X38 geometry



(a) Density plot at symmetric plane



(b) Pressure plot at symmetric plane

Figure 5.23: Viscous solution for the X38 geometry

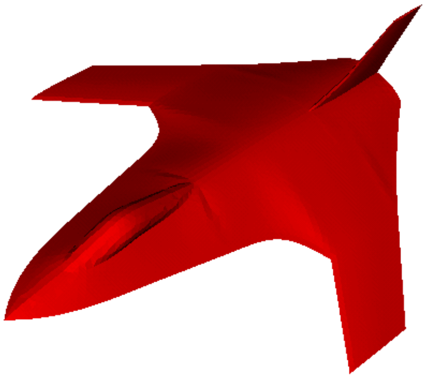
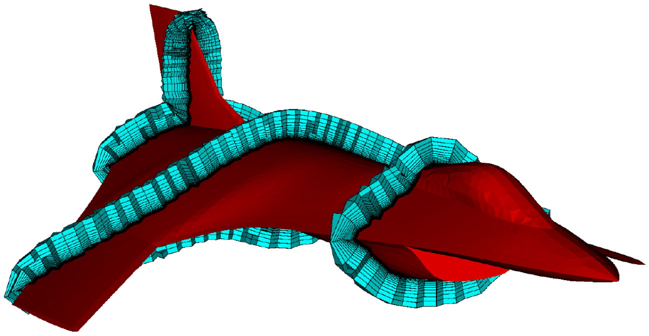
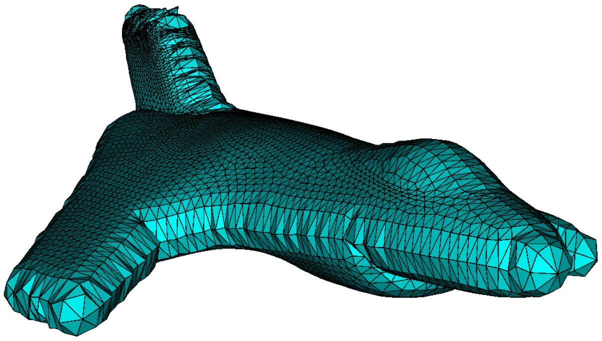


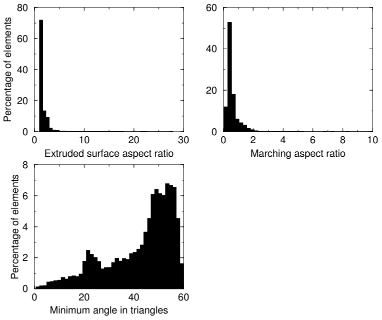
Figure 5.24: Geometry of a swept forward wing fighter



(a) Different cross sections of the volume mesh

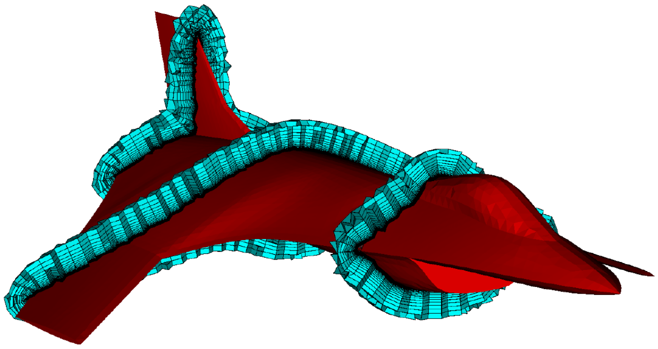


(b) Outer surface mesh of the volume mesh

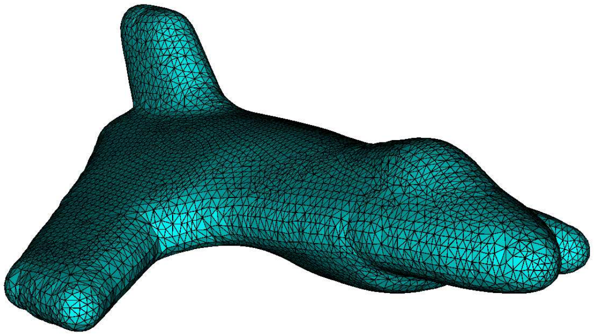


(c) Quality plots of the last layer of the volume mesh

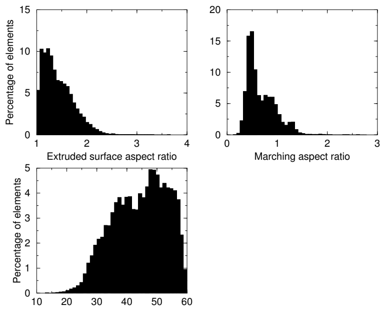
Figure 5.25: Volume mesh for the fighter geometry generated by extrusion without quality improvements



(a) Different cross sections of the volume mesh



(b) Outer surface mesh of the volume mesh



(c) Quality plots of the last layer of the volume mesh

Figure 5.26: Volume mesh for the fighter geometry generated by extrusion with quality improvements

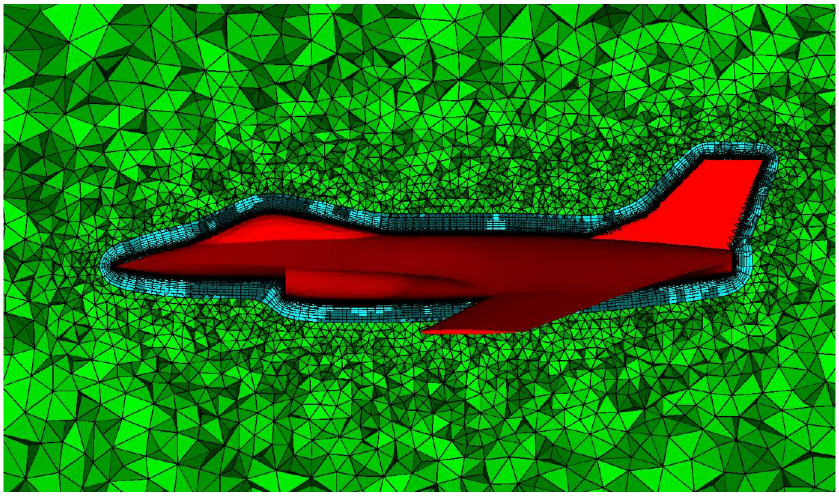


Figure 5.27: Cross section of the viscous volume mesh for the fighter geometry

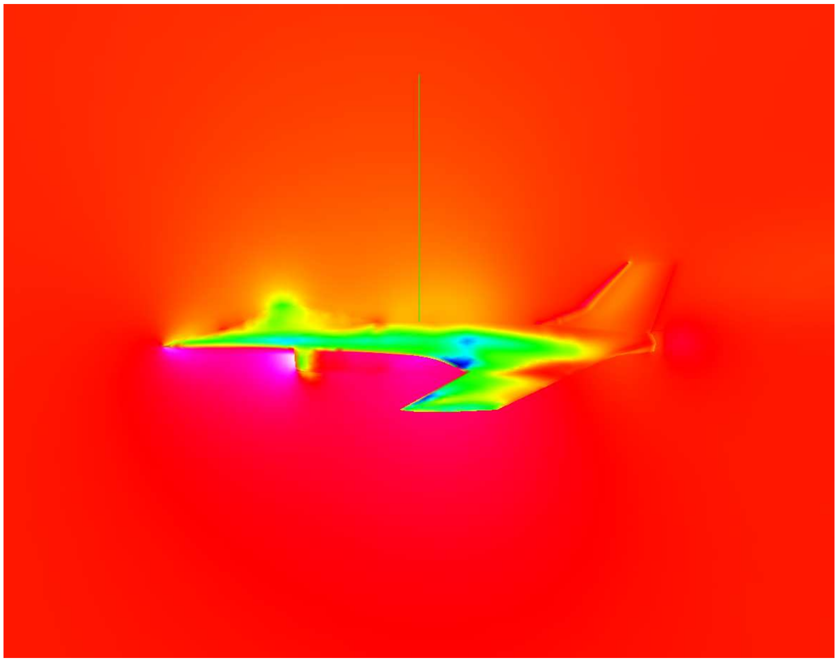
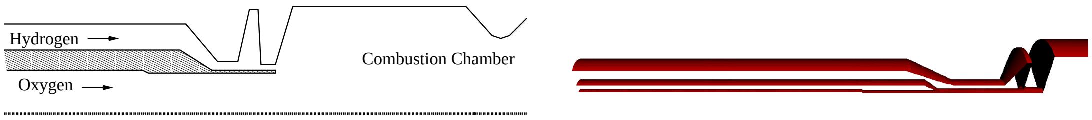


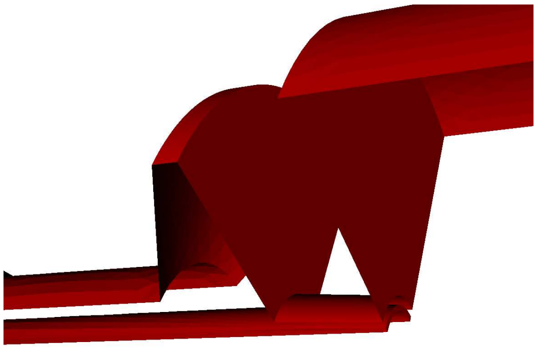
Figure 5.28: Pressure plot of viscous solution for the swept forward wing fighter geometry

(a) Whole geometry



(b) Flow description

(c) Details of the geometry at inflow



(d) Details of the geometry before the combustion chamber

Figure 5.29: Single element injector geometry

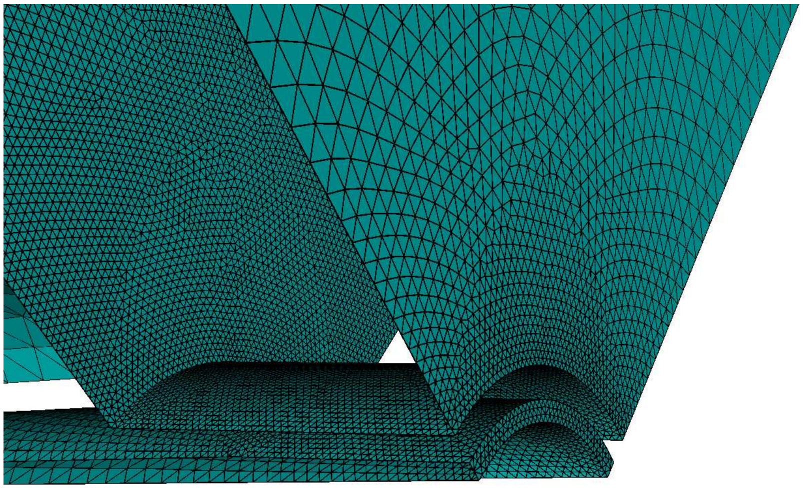


Figure 5.30: Details of the surface mesh at the post

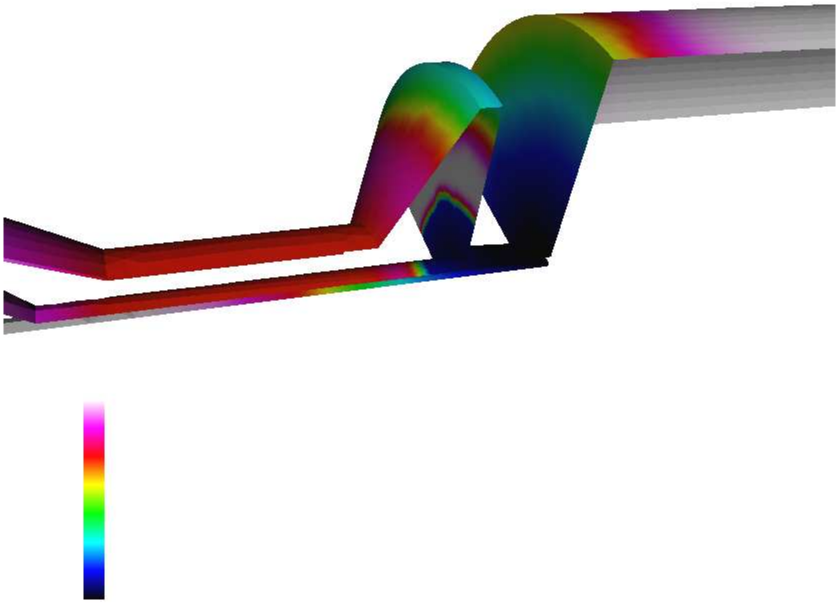
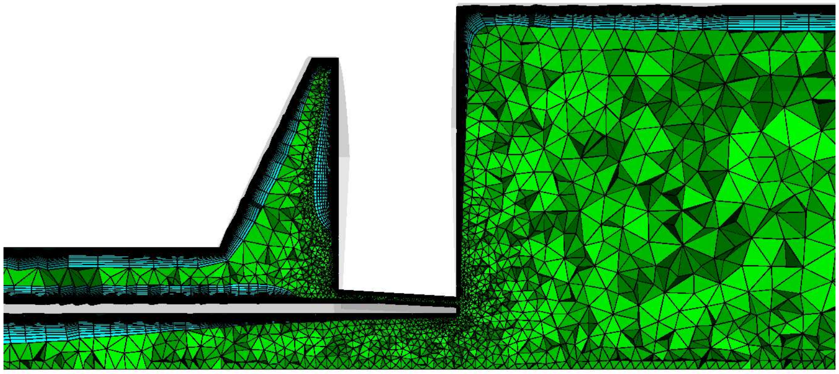
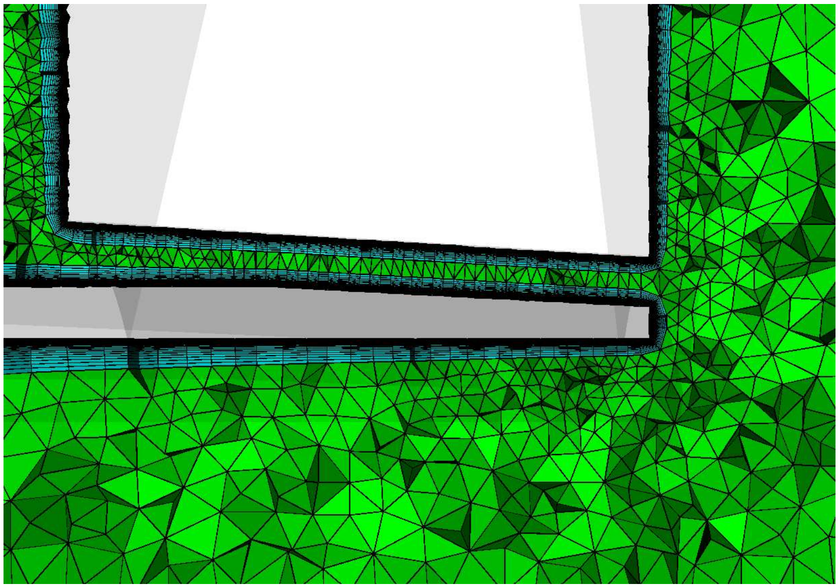


Figure 5.31: Contour plot of marching distance as a scalar function at each point

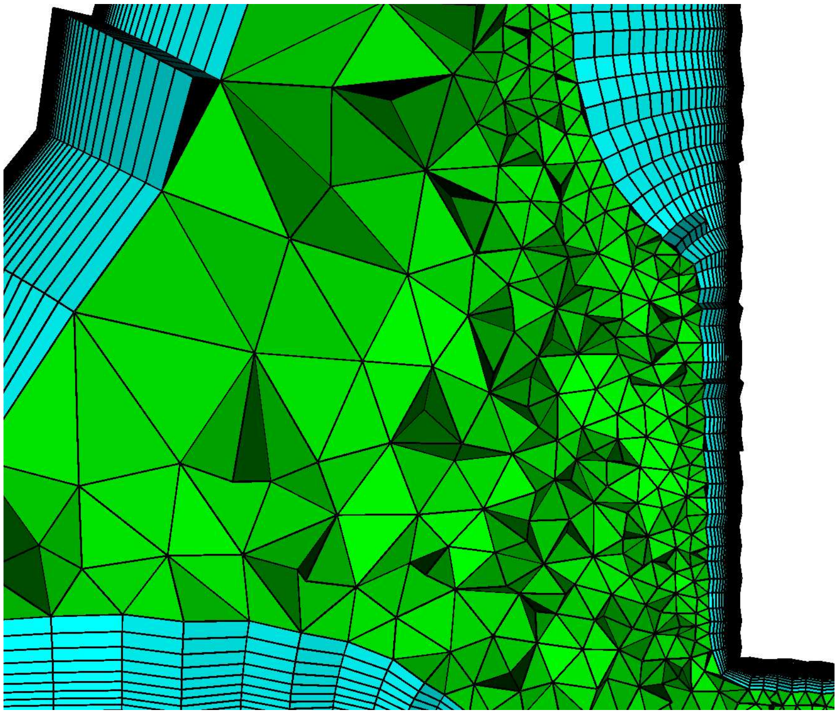


(a) Cross section the volume mesh

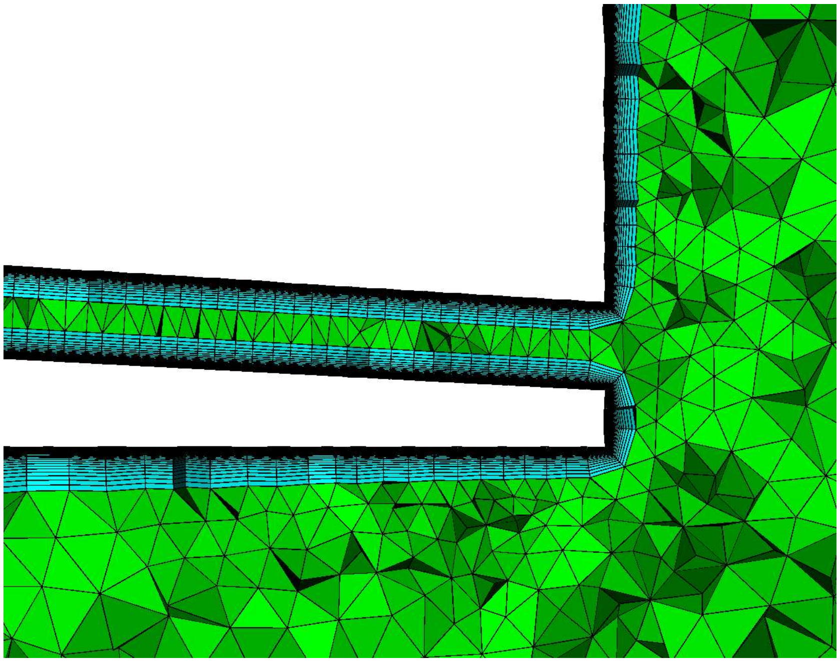


(b) Details of the cross section near the post

Figure 5.32: Volume mesh for single element injector geometry generated by extrusion without quality improvements



(a) Closeup of the cross section before the post



(b) Closeup of the cross section at the post

Figure 5.33: Details of the volume mesh generated by extrusion without quality improvements

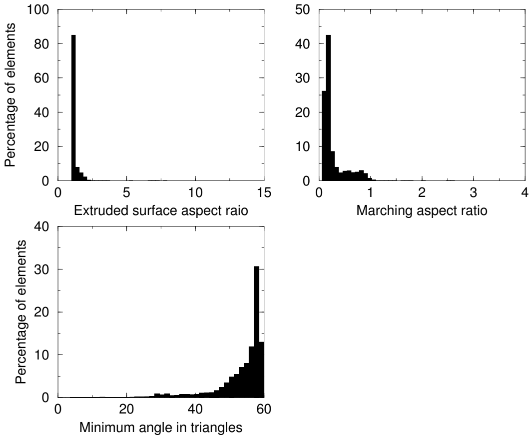
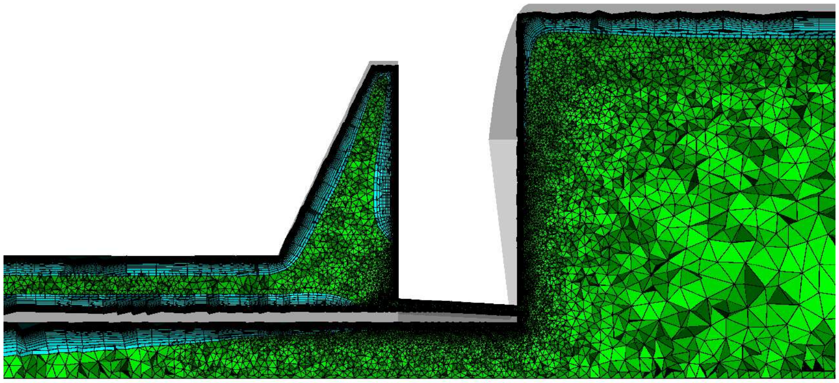
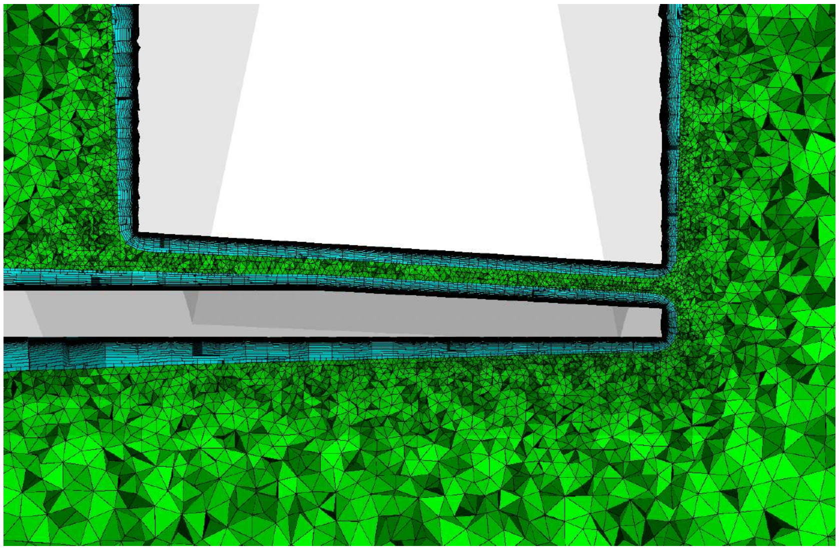


Figure 5.34: Quality plot of the last layer of the volume mesh

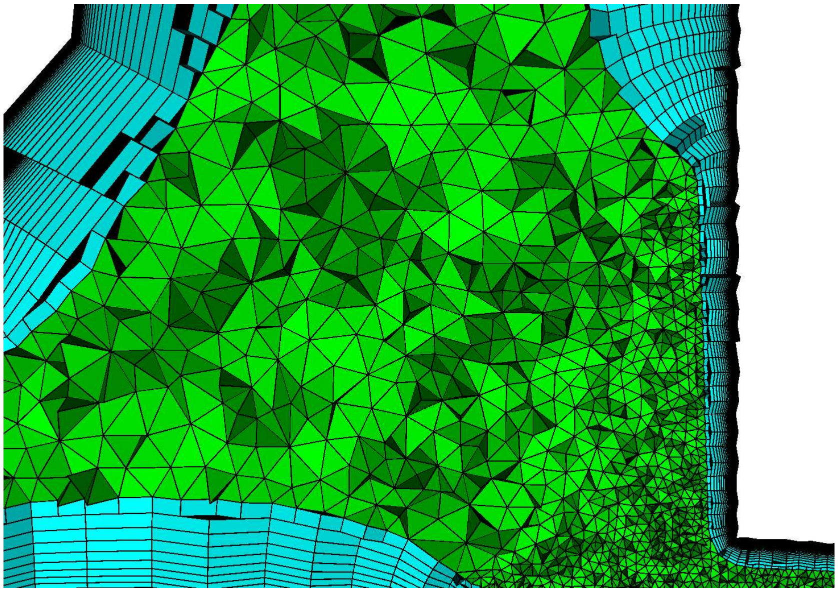


(a) Cross section the volume mesh

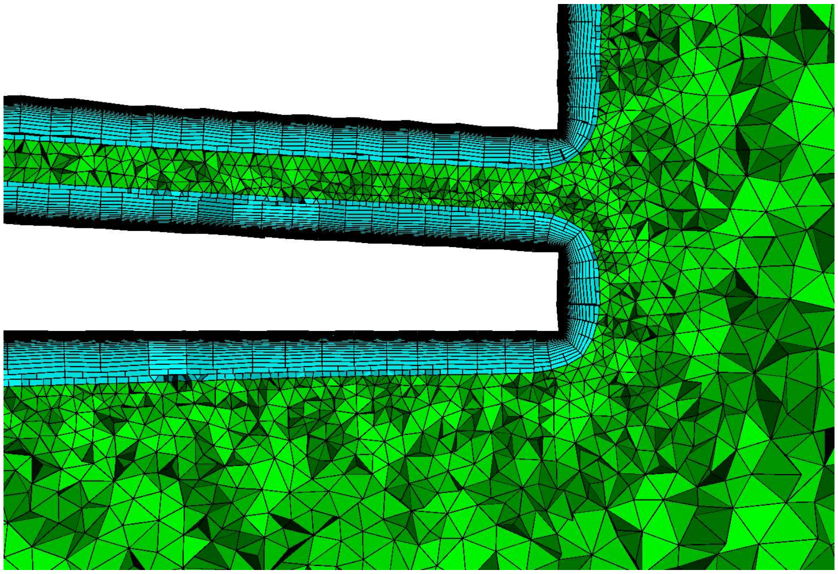


(b) Details of the cross section near the post

Figure 5.35: Volume mesh for single element injector geometry generated by extrusion with quality improvements



(a) Closeup of the cross section before the post



(b) Closeup of the cross section at the post

Figure 5.36: Details of the volume mesh generated by extrusion with quality improvements

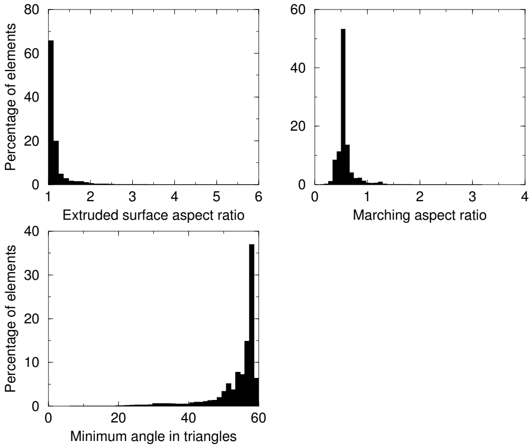


Figure 5.37: Quality plot of the last layer of the volume mesh

CHAPTER VI

CONCLUSIONS

In this dissertation, an advancing layer scheme is developed to generate the extruded near-body mesh from a specified surface mesh of arbitrary topology. Given an initial surface mesh, two layers of a algebraic reference mesh are generated by extruding in the direction of local surface normals. A Poisson equation is used to smooth the reference mesh. The advantage of using the relatively expensive Poisson equation is to exploit its control functions. The control functions help propagate the characteristics of the initial surface mesh into interior of the domain. The process of building and smoothing a reference mesh are repeated until the desired number of layers is reached. As with any extruded mesh, element quality may deteriorate near concave and convex regions of the geometry. This is due to coalescence or divergence of mesh faces. To address this issue, a newly developed quality improvement step is introduced after smoothing and before the extruding next layer. Topological operations including edge collapse, local face refinement, and local reconnection are performed in each layer to drive the mesh toward isotropy. These operations may result in a generalized elements. The topological operations are triggered by local geometric sensors. When extruding from different geometries which are in close proximity, mesh crossing may occur due to a lack of space to grow. To avoid crossing, the marching distance is reduced automatically by using a modified ray-plane intersection algorithm. An octree is used to make the ray-plane intersection algorithm computationally faster. The transition from the extruded mesh to the tetrahedral mesh may not be smooth where the marching distance is reduced to avoid crossing. Therefore, a new global face refinement sensor is used to obtain a smooth transition from the extruded mesh to the tetrahedral mesh.

The results presented here demonstrate the smoothing capability of the new algorithm near concave regions of the geometry. In these results, the mesh faces remains orthogonal except in concave and convex regions of the geometry. Mesh faces emanating out of concave regions adjust accordingly to avoid crossing and vary smoothly. In strongly concave regions, increasing smoothing may avoid crossing. According to the quality plots of the results, the new topological improvement operations improve the transition from the extruded mesh to the tetrahedral mesh when measured using face-based quality metrics. The meshes presented in the results section for multiple geometries in close proximity demonstrated smooth variations of near-body mesh thickness. According to the results, face refinement operation improves the transition and saves points by eliminating the need of refining the given surface mesh. A few computed solutions are presented to illustrate the validity of the approach. A preliminary validation effort is presented for an ONERA M6 wing with a volume mesh containing generalized elements. The results obtained are in agreement with experimental data. Since topologically adaptive generalized elements were successfully employed to improve mesh quality, the results presented here validate the dissertation statement.

Although the efficacy of this approach was demonstrated, the technology is not yet a production tool. Presently, faced-based metrics are used for measuring quality. The issue of appropriate quality of a general volume element remains to be addressed. Extensive validation efforts are required for meshes containing generalized elements. Robustness of the code has to be increased by validating it for more applications. Memory management also has to be improved. Presently the code has no graphical user interface. To make it a useful tool, it should be integrated into commonly used software like SolidMesh [43].

REFERENCES

[1] Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., North Holland, New York, NY, 1985.

[2] Allwright, S. E., "Techniques in Multiblock Domain Decomposition and Surface Grid Generation," *Numerical Grid Generation in Computational Field Simulations*, edited by S. Sengupta, J. F. Thompson, J. Hauser, and P. R. Eiseman, Pineridge Press, Miami, FL, 1988, pp. 559–568.

[3] Dannenhoffer, J. F., "Automatic Blocking for Complex Three-Dimensional Configurations," Tech. rep., May, 1995.

[4] Jiang, M. Y. and Remotigue, M. G., "GUM-B Grid generation Code and Applications," *Numerical Grid Generation in Computational Field Simulations*, edited by M. Cross, B. K. Soni, J. F. Thompson, J. Hauser, and P. R. Eiseman, NSF Engineering Research Center for Computational Field Simulations, Mississippi State, 1996.

[5] "Fluent Inc," "ttp://www.fluent.com" accessed march 2003.

[6] "Gridgen," "http://www.gridgen.com" accessed march 2003.

[7] "Grid Pro," "http://www.gridpro.com" accessed march 2003.

[8] Weatherill, N. P. and Hassan, O., "Efficient Three Dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints," *International Journal for Numerical Methods in Engineering*, Vol. 37, 1994, pp. 2005–2039.

[9] Marcum, D. L. and Weatherill, N. P., "Unstuctured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA J.*, Vol. 33, 1995, pp. 1619–1625.

[10] Lohner, R. and Parikh, P., "Generation of Three Dimensional Unstructured Grids by the Advancing Front Method," *International Journal for Numerical Methods in Fluids*, Vol. 8, 1988, pp. 1135–1149.

[11] Shephard, M. S. and Yerry, M. A., "A modified quadtree approach to finite element mesh generation," *IEEE Computer Graphics and Applications*, Vol. 3, 1983, pp. 25–37.

[12] Shaw, J. A., "Hybrid Grids," *Handbook of Grid Generation*, edited by J. F. Thompson, B. K. Soni, and N. P. Weatherill, 1998.

[13] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.

[14] Marcum, D. L., "Generation of Unstructured Grids for Viscous Flow Applications," *AIAA Paper 95-0212*, 1995.

[15] Garimella, V. R. and Shephard, M. S., "Boundary Layer Meshing for Viscous Flows in Complex Domains," *Proc.* 7th *Int. Meshing Roundtable*, Sandia National Laboratories, 1998, pp. 107–118.

[16] Chappell, J. A., Shaw, J. A., , and Leatham, M., "The Generation of Hybrid Grids Incorporating Prismatic Regions for Viscous Flow Calculations," *Numerical Grid Generation in Computational Field Simulations*, edited by M. Cross, B. K. Soni, J. F. Thompson, J. Hauser, and P. R. Eiseman, NSF Engineering Research Center for Computational Field Simulations, Mississippi State, 1996, pp. 537–546.

[17] Nakahashi, K., Nozaki, O., Kikuchi, K., and Tamura, A., "Navier-Stokes Computations of Two and Three Dimensional Cascade Flows," *Journal of Propulsion and Power*, Vol. 5, May, 1989, pp. 320–326.

[18] Kallinderis, Y. and Ward, S., "Prismatic Grid Generation for 3-D Complex Geometries," *AIAA Journal*, Vol. 31, No. 10, 1993, pp. 1850–1856.

[19] Owen, S., Canann, S., and Saigal, S., "Pyramid Elements for Maintaining Tetrahedra to Hexahedra Conformability," *Trends in Unstructured Mesh Generation AMD*, Vol. 220, 1997, pp. 123–129.

[20] Wang, Z. J., "An Automated Viscous Adaptive Cartesian Grid Generation Method for Complex Geometries," *Numerical Grid Generation in Computational Field Simulations*, edited by M. Cross, B. K. Soni, J. F. Thompson, J. Hauser, and P. R. Eiseman, NSF Engineering Research Center for Computational Field Simulations, Mississippi State, 1998, pp. 577–586.

[21] Ito, Y. and Nakahashi, K., "Unstructured Mesh Generation For Viscous Flow Computations," *Proc.* 11th *Int. Meshing Roundtable*, Sandia National Laboratories, 2002.

[22] Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., and Krysl, P., "Meshless Methods: An Overview and Recent Developments," *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, 1996, pp. 3–47.

[23] Knupp, P., "Winslow Smoothing on Two-Dimensional Unstructured Meshes," *Proc.* 7th *Int. Meshing Roundtable*, Sandia National Laboratories, 1998, pp. 449–457.

[24] Thompson, D. S. and Soni, B. K., "Semistructured Grid Generation in Three Dimensions using a Parabolic Marching Scheme," *AIAA Journal*, Vol. 40, No. 2, 2002, pp. 391–393.

[25] Leatham, M., Stokes, S., Shaw, J. A., Cooper, J., Appa, J., and Blaylock, T. A., "Automatic Mesh Generation For Rapid-Response Navier-Stokes Calculations," *AIAA paper 2000-2247*, 2000.

[26] Cary, A. and Michal, T., "Generalized Prisms for Improved Grid Quality," *AIAA paper 2001-2552*, 2001.

[27] Koomullil, R. P. and Soni, B. K., "Generalized Grid Techniques in Computational Field Simulation," *Numerical Grid Generation in Computational Field Simulations*, edited by M. Cross, B. K. Soni, J. F. Thompson, J. Hauser, and P. R. Eiseman, NSF Engineering Research Center for Computational Field Simulations, Mississippi State, 1998, pp. 521–531.

[28] Soni, B. K., "Elliptic Mesh Generation Systems: Control Functions Revisited," *Appl. Math. Comput.*, Vol. 59, 1993, pp. 151–164.

[29] Noack, R. W. and Anderson, D. A., "Solution-Adaptive Grid Generation using Parabolic Differential Equations," *AIAA Journal*, Vol. 28, 1990, pp. 1016–1023.

[30] Knupp, P., "Matrix Norms & The Condition Number: A General Framework to Improve Mesh Quality Via Node-Movement," *Proc.* 8th *Int. Meshing Roundtable*, Sandia National Laboratories, 1999.

[31] Knupp, P., "Applications of Mesh Smoothing: Copy, Morph, and Sweep on unstructured Quadrilaterals Meshes," *Int. J. Num. Meth. Eng.*, Vol. 45, 1999, pp. 37–45.

[32] Weatherill, N. P., "Unstructured Grids: Procedures and Applications," *Handbook of Grid Generation*, edited by J. F. Thompson, B. K. Soni, and N. P. Weatherill, 1998.

[33] Freitag, L. A., "Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number," *Proc.* 8th *Int. Meshing Roundtable*, Sandia National Laboratories, 1999.

[34] Wa, K. and Chen, Z., "A Simple and Effective Mesh Quality Metric for Hexahedral and Wedge Elements," *Proc.* 9th *Int. Meshing Roundtable*, Sandia National Laboratories, 2000.

[35] Hoppe, H., "Progressive meshes," *ACM SIGGRAPH*, 1996, pp. 99–108.

[36] Khawaja, A., McMorris, H., and Kallinderis, Y., "Hybrid Grids for Viscous Flows around Complex 3-D Geometries Including Multiple Bodies," *AIAA Paper 95-1685*, 1995.

[37] Glassner, S., *An introduction to ray tracing*, Academic Press Limited, 1989.

[38] Berg, M. D., Kreveld, M. V., Overmars, M., and Schwarzkopf, O., *Computational Geometry Algorithms and Applications*, Springer, 1997.

[39] Kay, T. L. and Kajiya, J. T., "Ray Tracing Complex Scenes," *ACM SIGGRAPH Computer Graphics*, Vol. 20, 1986, pp. 269–278.

[40] Moller, T. and Trumbore, B., "Fast, minimum storage ray-triangle intersection," *Journal of graphics tools*, Vol. 2, 1997, pp. 21–28.

[41] "Cobalt$_{60}$ On-Line Documentation, Cobalt$_{60}$,," 1999, http://www.va.afrl.af.mil/vaa/vaac/COBALT/co_docs.html.

[42] *The Visualization Toolkit User's Guide*, Kitware Inc.

[43] "SolidMesh," http://www.erc.msstate.edu/simcenter/docs/solidmesh.

[44] "ONERA M6 Wing," http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html.

[45] Luke, E. A., Tong, X.-L., Wu, J., Tang, L., and Cinnella, P., "A Chemically Reacting Flow Solver for Generalized Grids," *AIAA Journal*, submitted.

[46] Wu, J., Tang, L., Luke, E. A., Tong, X.-L., and Cinnella, P., "Comprehensive Numerical Study of Jet-Flow Impingement over Flat Plates," *Journal of Spacecraft and Rockets*, Vol. 39, 2002, pp. 337–366.

[47] Tomaro, R. F., Strang, W. Z., and Sankar, L. N., "An Implicit Algorithm for Solving Time Dependent Flows on Unstructured Grids," *AIAA paper 1997-0333*, 1997.

[48] Berry, S. A., Weilmuenster, K. J., Alter, S. J., and Merski, N. R., "X-38 Experimental Aeroheating At Mack 10," 35^th AIAA Thermophysis Conference, June 2001, Anaheim, CA.