UNSTRUCTURED MESH GENERATION FOR 3D VISCOUS FLOW

J. Peraire*
Department of Aeronautics and Astronautics,
MIT, Cambridge, Massachusetts

K. Morgan†
Department of Civil Engineering,
University of Wales, Swansea, UK

Abstract

In this paper we review the procedures which are currently available for the generation of unstructured meshes suitable for use in the simulation of high Reynolds number viscous flows. For such simulations, computational efficiency generally dictates the requirement for the use of highly stretched cells in viscous dominated regions. Meshes of this type cannot be directly generated by the classical unstructured mesh generation procedures and, therefore, special techniques need to be devised to deal with these highly anisotropic grids. In this context, this paper will consider hybrid structured/unstructured meshing procedures and also truly unstructured approaches, including a directional mesh refinement algorithm.

1 Introduction

The unstructured mesh approach is now widely accepted as a viable procedure for the simulation of compressible inviscid aerodynamic flows. The procedure allows for the rapid analysis of flows involving complex geometrical configurations and many industrial organizations have access to their own unstructured mesh solution capability. The success of the unstructured mesh approach in the area of inviscid flow simulation has been due, in no small part, to the availability of fully automatic isotropic mesh generation procedures. With such generators, the task of discretising a computational domain of complex shape, defined by a valid boundary representation, becomes a routine and rapid procedure. Inviscid flow simulations can usually be performed effectively on isotropic grids. Exisiting algorithms for generating isotropic grids are rather mature and the quality of the grids produced is generally acceptable for computing accurate Euler solutions.

Given this situation, attention has naturally turned towards the modeling of viscous high Reynolds number flows. A significant volume of work has already been directed towards the development of unstructured mesh solution algorithms capable of simulating viscous flows on general meshes. At the same time, the problem of generating suitable unstructured meshes for such flows has also been receiving attention. This is necessary as current grid generation algorithms for inviscid flows are not well–suited to the process of generating the highly stretched, good quality grids which are essential for high Reynolds number flow simulations.

In this paper, we start by discussing some practical issues in geometry modeling and the interfacing of grid generators with geometry creation tools. Next, we review some algorithms for unstructured grid generation and, in particular we focus on the techniques available for the generation of meshes suitable for high Reynolds number flow simulation. We consider techniques which produce a consistent discretisation of the computational domain, which means that approaches such as the chimera or overlapping cartesian mesh methods, are not addressed. Hybrid structured/unstructured meshing procedures are reviewed, together with truly unstructured algorithms. In particular, we describe the latest developments on a recently proposed directional refinement algorithm [1].

2 Geometry Modeling

When representing a general computational domain, it is useful to distinguish between topological and geometrical entities. The topology of the domain consists of regions, faces, edges and vertices. Regions are closed volumes in three dimensional space, which may be either simply or multiply connected. The boundary of a region is made up of a collection of faces. Each face is bounded by edges, which form one or more closed loops.

*Associate Professor, AIAA Associate Fellow.
†Professor, AIAA Associate Fellow.

Edges connect with each other at vertices. As an additional complexity, "floating" topological entities, such as structural shell elements, modelled as having zero thickness, or thin wires immersed in a three dimensional space, can be present. Faces, edges and vertices must have appropriate geometrical representations. These are referred to as surfaces, curves and points respectively.

This conceptual distinction between topology and geometry facilitates the interface between the mesh generator and existing geometry modeling systems. For industrial applications, the definition of the boundary in a manner which is suitable for grid generation is one of the main bottlenecks. Frequently, geometries generated by CAD systems for manufacturing purposes contain defects, such as gaps and small features that need not be resolved in the flow simulation. These need to be repaired or removed before grid generation can begin. In some circumstances, a flow simulation may require the definition of features not present in the CAD model, such as far field computational boundaries, wake surfaces, etc. A convenient method of dealing with such problems is to work with a unique topology definition, but to allow geometry generated using different systems to co–exist. This approach is particularly useful for repairing existing geometry and adding computational boundaries. In addition, the ability to mix different classes of geometry means that it is unnecessary to convert every geometric definition to a common representation. A convenient means of achieving this is to employ a computational analysis programming interface [2], which allows access to the geometry within several CAD commercial packages. In industrial implementations, where a variety of geometry exchange formats are often in use within the same organization, this approach has been found to increase the robustness of the overall mesh generation process considerably.

3 Generation of Isotropic Grids

The advancing front and Delaunay methods are two approaches for generating unstructured meshes in 2D and 3D domains.

3.1 Advancing Front Method

Advancing front methods were originally developed in 2D in order to connect an a priori defined distribution of points [3]. However, the power of the method became fully apparent when it was combined with an algorithm for point creation and mesh control, using a locally defined variable metric [4, 5], and when it was implemented using efficient data structures [6, 7]. A main advantage of the advancing front method is that the discretization of the boundary is never modified. This means that the surface and volume disctretization processes are completely decoupled. A disadvantage of the method is that there is no rigorous accompanying theory that will imply the success of the generation process for general domains. In practice, however, provided the distortion of the mesh is not too large, reliable implementations can be achieved [8, 9, 10].

3.1.1 Delaunay Approach

Delaunay algorithms [11, 12] provide a very fast and reliable method for performing the tessellation of a given lattice of points in any number of dimensions. For effective grid generation purposes, however, the Delaunay algorithm needs to be supplemented with a suitable process for point creation. One possible option is to select points from cartesian [13] or octree [14] background meshes. This approach leads to very uniform meshes in the interior of the domain, but the quality tends to deteriorate near the boundaries. Other approaches, which combine the point distributions generated by structured grid generators around individual components, have also been employed. More general, however, are the incremental point insertion algorithms in which the location of the new points is determined based on the current grid. Examples of such algorithms include the centroid [15], circumcircle [16], and edge [17], point insertion algorithms. A notable example, which produces very regular meshes, at least in two dimensions, is that presented in [18], where the algorithm used to place the new points resembles the point placement algorithm of the advancing front method. In some respects, this approach combines the two approaches in an optimal manner.

In two dimensions, Delaunay triangulations posess certain regularity attributes which make them very attractive for flow simulation. Unfortunately, these properties are lost in 3D and, for practical applications, the basic Delaunay algorithm needs to be modified to avoid the generation of highly distorted elements. A common approach is to modify the Delaunay approach to use, for instance, a minmax reconnection algorithm [16, 19], in which the element connectivity is locally selected so as to minimize the maximum angle occurring in any tetrahedron.

Unlike the advancing front approach, the Delaunay method does not guarantee the integrity of the computational boundaries. This means that boundary recovery procedures need to be added to ensure that the correct boundary definition is present in the final mesh. Recovery of the boundary may be accomplished by the application of local mesh modification procedures and, in certain circumstances, by the creation of new surface mesh points [20]. An algorithm in which only interior points are generated and which does not require the modification of the surface triangulation is, in principle, possible [21]. In practice, however, the robust implementation of such algorithms for general applications is difficult.

4 Surface Grid Generation

For most volume grid generation procedures, a pre–requisite to the process of discretisation of a 3D computational domain is the discretisation of the boundary representation.

4.1 Extension of 2D techniques

One of the simplest ways of performing a surface discretisation is by the introduction of an intermediate mapping between the three dimensional surface and the two dimensional parametric space. Ideally, this mapping between a 2D region and the 3D surface should conserve metric properties; i.e. angles and ratios. However, this can only be accomplished in the case of developable surfaces, which means that metric properties are only approximately conserved in the general case [22]. In addition, simplified or sub–parametric forms are sometimes employed and these introduce geometrical inaccuracies. After the mapping has been established, well–developed grid generation techniques for the generation of isotropic grids in 2D, such as Delaunay or advancing front methods, can then be used. The approach is very effective and fast, but one of the associated difficulties is that, for complex non–developable surfaces, the mapping may introduce a significant distortion. This is reflected in a degredation of quality in the generated surface mesh. When sub–parametric representations are used, the points generated using this intermediate mapping will not lie on the exact surface definition. This problem needs to be resolved by projecting these points onto the original surface in a post–processing step [9].

4.2 Variable metric techniques

A more sophisticated approach is to utilize a variable metric grid generation technique [23], which allows for the generation of anisotropic grids. In this way, the surface generation can be carried out directly in the 2D parametric domain, and the distortion introduced by the mapping can be factored into the local mesh size and shape specification. Although more elegant, one of the drawbacks of this apporach is that for highly stretched surfaces, the robustness of the algorthim may be reduced.

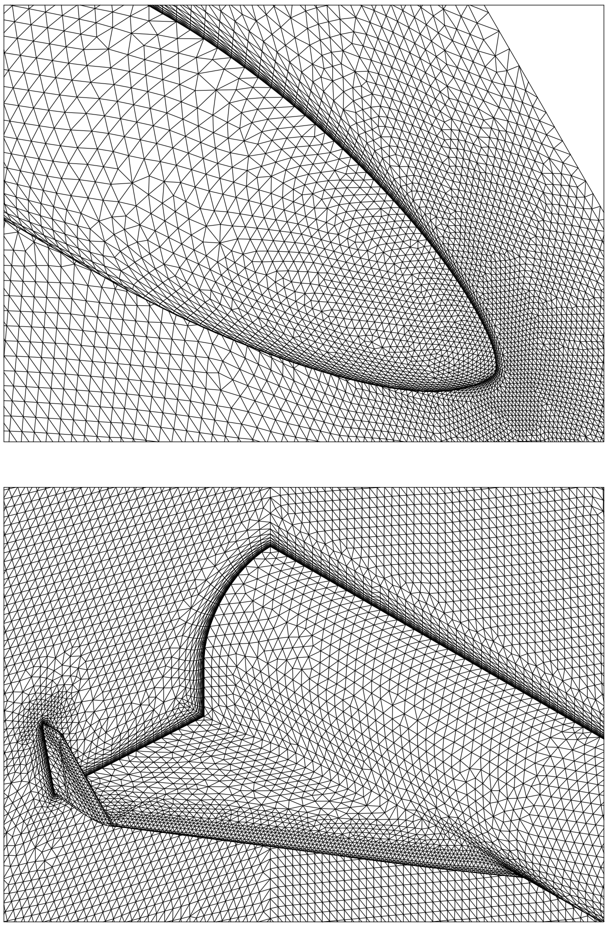

Figure 1: Viscous mesh for a re-entry vehicle generated using a hybrid method with a variable number of layers.

5 Generation of Anisotropic Grids

The unstructured tetrahedral grid generation processes which have been outlined, have proved very successful when isotropic meshes are required. However, they are not well suited to the requirement of producing good quality, highly stretched grids of the type necessary for high Reynolds number viscous flow simulation. For this reason, a number of different approaches, of increasing levels of complexity, which attempt to address this problem, have been proposed.

5.1 Hybrid Approaches

The first attempts at extending the unstructured mesh approach into the area of viscous flow simulation involved a hybrid approach [24, 25]. The basic idea was to associate a structured mesh in the normal direction, consisting of a prescribed number of layers of elements, with each solid surface component. The remainder of the computational domain was then discretised in an isotropic manner, using one of the standard automatic

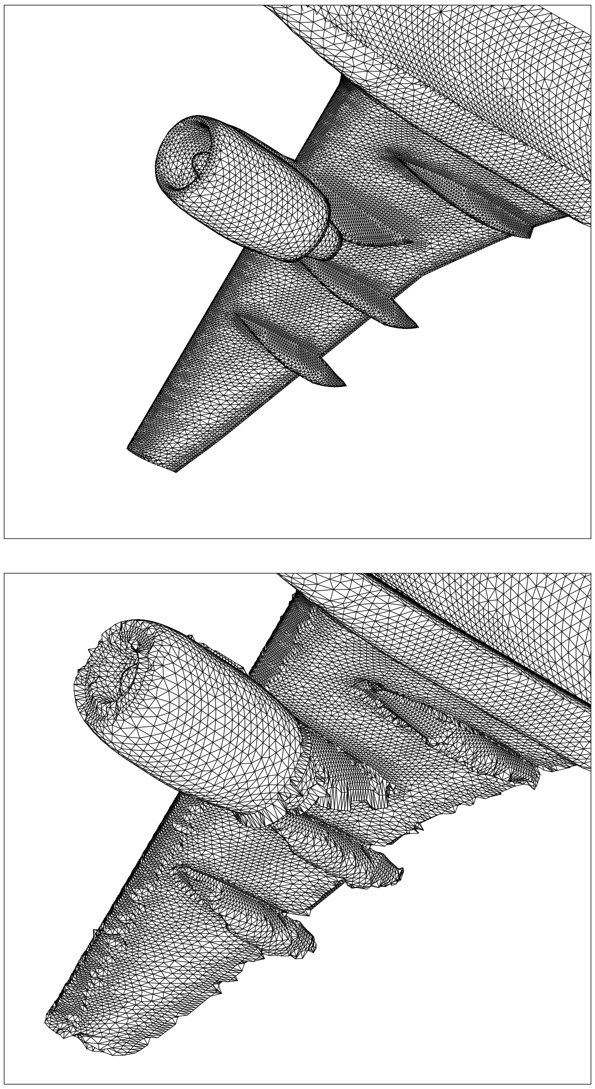

Figure 2: Viscous mesh for a transport configuration generated using the advancing layer method: initial surface and after the generation of eight layers.

meshing procedures. The method proves quite effective for simulations involving relatively simple geometrical shapes, but its usefulness is limited by well–known disadvantages, such as the lack of geometric flexibility, the requirement for excessive user intervention and the generation of low–quality meshes. It is apparent that this approach naturally allows for the use of prismatic cells in the structured region, although structured tetrahedra can also be employed.

More recently, improved approaches have been proposed in which the thickness of the structured layer is varied using grid control functions [26], and which allow for a variable number of structured layers to be generated depending on the geometry and grid size distribution [27]. In some implementations, these modified approaches become very similar to the advancing layer methods of next section.

Figure 1 shows a typical mesh, for a generic re–entry vehicle, generated using the hybrid approach. In this case a maximum of 16 layers of constant, prescribed, thickness have been generated arround the solid surface. The normals to the surface, used to construct the layers, are smoothed to avoid distorted cells, and the layer generation is locally stopped when either the thickness of the layer is comparable to that of the isotropic mesh or a mesh inconsistency is about to occur. After the layer generation process is terminated, the isotropic grid is generated, in this case, using the advancing front method. This approach is seen to generate high quality meshes provided the geometries are relatively simple.

5.2 Advancing Layers Methods

The advancing layers approach was the first attempt at using unstructured mesh generation techniques to aid in the process of constructing highly stretched meshes [28, 29]. The approach can be regarded as a modification to the standard advancing front method. An initial generation front is formed from the faces in the mesh on solid surfaces. For the first layer of cells, the new points are located in a prescribed direction, and at a prescribed distance, from the surface vertices. The prescribed direction is chosen to coincide with a defined normal direction at each surface vertex. The process is repeated for subsequent layers, with recomputation of the normal direction for each layer and local termination of the process when self intersection of the mesh edges is detected or when the generated elements are almost isotropic. This ensures a smooth transition between the isotropic and anisotropic regions of the mesh. This process has been successfully employed to produce highly anisotropic meshes in the vicinity of a variety of complex aerodynamic configurations. Again, a modification to the process can produce mixed prismatic and tetrahedral grids, while for complete flexibility the appearance of pyramidal cells must also be allowed.

Compared to hybrid approaches, advancing layer methods offer a greater degree of flexibility as they easily allow for mulitple normals to be used at singular points [19]. Figure 2 illustrates the use of the advancing layers method applied to a transport configuration [30]. The figure shows the solid surface triangulation together with the front after the generation of eight layers of elements. In is seen that in certain regions, the layer generation process needs to be terminated to prevent self intersection of the front.

5.3 Unstructured methods

These methods attempt to adaptively generate viscous grids without according any special treatment to the boundary layer regions. They are, in principle, truly automatic but, to date, no realistic three dimensional applications have yet been presented. An approach in which highly stretched meshes are obtained adaptively through a combination of mesh refinement and mesh movement steps has been proposed in [31], whereas the method advocated in [32] accomplishes the same objective by using a variable metric technique.

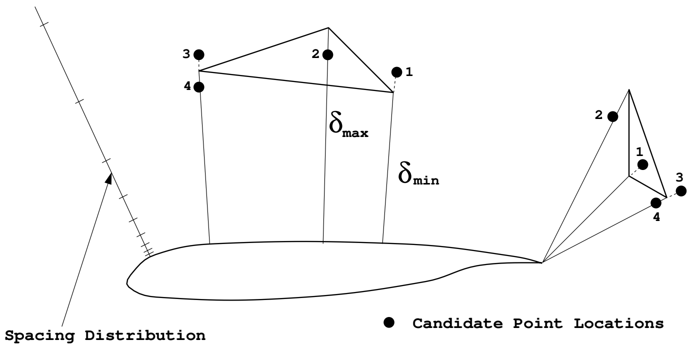

Figure 3: Directional refinement.

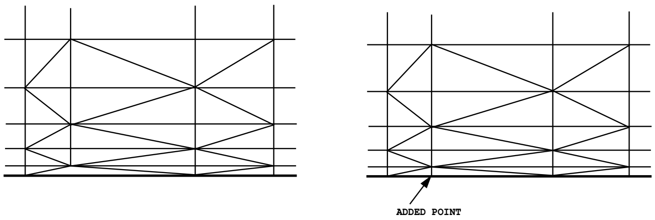

Figure 4: Edge refinement to improve element quality.

6 Directional Refinement

An alternative to the algorithms described in the previous section is the directional refinement algorithm introduced in [1]. The starting point here, is an existing isotropic grid which is recursively subdivided until the final mesh mesh consists of stretched elements which are such that the largest angle in any element is close to ninety degrees. A similar concept was presented in [16] for the two dimensional case. The amount of directional refinement, or stretching, which is required is based upon the distance to a specified curve or surface. Every point in space can be assigned a scalar value δ, which represents the distance between the point and the nearest point on a specified curve or surface. A one dimensional point distribution is specified by the user for every curve or surface which is to be used for the refinement, see figure 3. This represents the target for the distribution of mesh points in the vicinity of the curve or surface in the directionally refined mesh. A geometric point spacing distribution is usually employed, with the additional possibility of having a number of equally spaced points immediately adjacent to the curve or surface.

Each element in the mesh is considered in turn and, if the element size measured with the distance metric (i.e. $\delta_{max} - \delta_{min}$) is larger than the required point spacing distribution, new points are inserted. A list of candidate points, lying on the lines connecting the element vertices and the corresponding nearest point on the curve/surface, is formed. The new points are located at sites determined by the specified one dimensional point distribution. The candidates are ordered as shown in figure 3, with the first and second candidate points corresponding to the vertices with minimum and maximum distance respectively. All other vertices are assigned two additional points. The list of candidates is considered in turn, and the first point in the list that is not too close to an existing point or to a boundary face is inserted. The distance used as a threshold in this check is a fraction of the local one dimensional point spacing.

Several strategies can be followed to implement this directional refinement in three dimensions. One could start from a three dimensional isotropic mesh and refine edges, faces and finally the volume. Alternatively, one can perform the edge refinement immediately after the isotropic edge discretization, the face refinement after the isotropic face discretization and finally the the volume refinement after the volume has been discretized. Both strategies appear to have merits and disadvantages; Starting from a three dimensional grid is arguably more general, but makes the process more expensixe and is harder to implement. On the other hand, the second approach requires a closer coupling between the isotropic grid generation and refinement procedures. We have implemented both approaches and found that the final meshes are comparable. Another possible approach, which can significantly reduce the cost of the refinement process and even improve the quality of the generated meshes, is to combine the directional refinement procedure with an advancing layer algorithm for point placement. In this manner, the grid can be generated very efficiently in regions away from surface intersections or complex configurations, and on the other hand the refinement process can take care of these more complex scenarios. To date, we have not yet implemented such a strategy, although we plan to do so in the near future.

6.1 Edge/Face Refinement

The directional refinement process starts by inserting points on the edges, to match the one dimensional spacing distribution, until all the segments satisfy the required directional refinement requirements.

Next, points are inserted on the faces until the size of all the face triangles satisfy the appropriate refinement criterion. Candidate points need to be on the face, and need to be at a prescribed distance from the curve/surface used for refinement. This is achieved by working on the tangent plane to the surface and then projecting the points onto the surface. After projection, the distance to the face/edge used for refinement may be changed. This problem is solved with a simple iterative algorithm. When the new point is inserted, the triangle containing that point is subdivided into three triangles and then local swapping operations are performed using the minimization of the maximum angle as the criterion for swapping. Both edge and face refinement are carried out before the three dimensional Delaunay triangulation process.

6.2 Minmax Point Insertion

After the boundary triangulation has been recovered, and the volumetric isotropic grid has been generated, additional points are inserted to meet the directional refinement criterion, i.e. for every tetrahedral element whose size, measured by the distance metric, is larger than specified, one point chosen from the list of candidates is inserted, provided that it is not too close to another point or to the boundary. Reconnection at this stage is based on the minmax criterion as described in [1]. The minmax criterion has been found to perform considerably better than the Delaunay criterion in this situation. This process is repeated until no more points can be inserted. It has been found that this step is necessary to ensure that the projected size of all elements is adequate. At this stage, if an element is found to be too large, and can not be broken using the above procedure, a point is inserted at the centre of the largest edge of the element. The largest edge is determined using the distance metric.

Finally, points are added to the edges and faces to improve the element quality. A typical situation where a point would be added to an edge is illustrated in figure 4.

6.3 Implementation

An important implementation issue is the selection of suitable data structures which are flexible enough to allow for the insertion/deletion, and efficient searching, of elements. The Alternate Digital Tree (ADT) data structure [6] is extensively used in the current implementation to store points, tetrahedral elements and triangular boundary facets. The ADT containing the triangles on face used for refinement is used to efficiently calculate the nearest point on the surface to a given point in the domain.

6.4 Examples

Four examples are included which illustrate the described directional refinement capabilities.

Box

The first example is chosen to illustrate some of the basic features of the directional refinement procedure. The geometry considered is shown if figure 5, and is that of a cube with a *floating* planar face. The generated isotropic grid is shown in figure 6. The directional refinement procedure is applied twice: initially with respect to the floating planar face and then with respect to the vertical edge closest to the viwer. Figure 7 shows the discretization after the edge refinement process. An intermediate step in the face refinement process is shown in figure 8, whereas the final refined mesh is shown in figure 9. Finally, figure 10 shows a detail of the refined mesh in the vicinity of the intersection between the face and the edge used for refinement. It is observed that all the directionally refined elements have their largest angle close to ninety degrees.

Turbine Blade

The second example shows the effect of performing directional refinement driven by the distance to a single curve. The geometry in this case was generated using Parasolid [33] and used directly in the meshing process without conversion. A coarse discretization of a generic turbine blade is generated, and the corresponding surface discretization of the hub and blade is shown in figure 11. A directionally refined mesh is then constructed by specifying a geometric distribution of mesh spacing away from the turbine leading edge. The background spacing distribution is kept unchanged. The surface discretisation of the directionally refined mesh is illustrated in figure 12, which shows the enhanced resolution which has been achieved in the vicinity of the leading edge.

Wing

The next example involves the generation of a mesh for the calculation of a viscous flow over a wing. Figure 13 shows the initial discretization of the wing surface. In this case, the figure also shows the discretisation of a planar surface which has been added to the original geometry to represent the wake. This mesh is directionally refined using the requirement that, at any point, the mesh spacing is governed by the distance to the leading and trailing edges and their corresponding one dimensional spacing distributions. The resulting surface mesh is shown in figure 14. When, as in this case, several refinement edges (and/or faces) are used simultaneously, there exists the flexibility for employing different spacing distributions for each edge or face. The distance associated to a given point is then determined as the minimum distance specified by the relevant one dimensional spacing distributions. For the case shown in figure 14, for instance, the spacing distribution for the leading edge is twice as fine as that specified for the trailing edge. Finally, one additional refinement has been carried out, driven by the distance to the wing and wake surfaces. The final surface mesh is shown in figure 15. It is noted than when the refinement procedure is driven by an edge, the tetrahedral elements which are generated have two small dimensions and a larger one. By contrast, when the refinement process is driven by a surface, the elements generated have only one small dimension and two larger ones.

Twin Engine Executive Jet

The final example is a realistic aerospace application and involves the generation of a mesh suitable for a viscous flow simulation over a twin engine executive aircraft. The first mesh which is generated is shown in figure 16. Although they are not shown in this figure, additional surfaces are added to allow for the accurate modeling of the wakes from the wing, the horizontal stabilizers and the engine pylon. To resolve the engine plume, a cylindrical surface, starting at the nacelle trailing edge and extending to about a distance of three quarters of the fuselage length downstream, was also added. A number of refinement operations are performed with the objective of generating a suitable mesh. It has been found to be beneficial to consider surfaces which meet approximately at right angles, such as wing/body or fin/tail, in separate refinement stages. The resulting mesh in this case is found to be approximately the union of the boundary layer meshes which would be produced by considering each of the surfaces alone. This provides enhanced resolution in corner regions, which are likely to be dominated by viscous effects. The mesh produced after 3 directional refinements is shown in figures 17 and 18. The effect of the directional refinement due to the viscous walls and wake surfaces is readily apparent. Although this mesh is probably still too coarse for a realistic calculation, it illustrates the main features and capabilities of the approach.

7 Conclusions

A number of techniques exist which can be effective at generating the highly stretched grids which are required for high Renolds number flow simulation. However, in the presence of demanding geometries, many of these techniques will fail or may require excessive user intervention. Considerable work still needs to be done to identify general purpose automatic algorithms which can generate viscous unstrucured grids reliably. The directional refinement capability is found to be very effective in resolving features such as leading and trailing edges. High mesh resolution is provided in corner regions where viscous effects are likely to dominate the flow.

It can be observed that the refinement produces a highly regular pattern within the mesh near the selected curves/surfaces. This means that tetrahedral cells can be readily grouped into triangular prisms. Although a grouping of this form would not completely eliminate the tetrahedral cells in these regions, it would result in a considerable reduction in the total number of cells in the mesh.

Finally, we believe that combination of the directional refinement process with an advancing point placement algorithm may help in making the method more efficient, and consequently, even more attractive.

Acknowledgements

We wish to express Electronic Data Systems for making the Parasolids system available for this research. The support of R. Haimes, O. Hassan and J. Peiró is also gratefully acknowledged.

References

[1] J. Peraire, K. Morgan, Unstructured mesh generation including directional refinement for aerodynamic flow simulation. *Finite Elements in Analysis and Design*, **25**, 343-356, 1997.

[2] R. Haimes and G.J. Follen, Computational Analysis Programming Interface. *Proceedings of the 6th Interantional Conference on Numerical Grid Generation iin Computational Field Simulation*, July 1998.

[3] S.H. Lo, A new mesh generation scheme for arbitrary planar domains, *Int. J. Num. Meth. Engng.*, **21**, 1403-1426, 1991.

[4] J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, Adaptive remeshing for compressible flow computations, *Journal of Computational Physics*, **72**, 449-466, 1987.

[5] J. Peraire, J. Peiró and K. Morgan, Adaptive remeshing for three–dimensional compressible flow computations, *Journal of Computational Physics*, **103**, 269-285, 1992.

[6] J. Bonet and J. Peraire, An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems, *Int. J. Num. Meth. Engng.*, **31**, 1-17, 1991.

[7] R. Löhner, Some useful data structures for the generation of unstructured grids. *Comm. Appl. Num. Meth.*, **4**, 123-135, 1988.

[8] J. Peiró, J. Peraire and K. Morgan, FELISA system reference manual and user's guide. Volume 1, *University of Wales Swansea Report*, CR/821/94, 1994.

[9] P. Parikh, S. Pirzadeh and R. Löhner, A Package for 3-D Unstructured Grid Generation, Finite Element Flow Solution and Flow Field Visualization, *NASA CR-182090*, 1990.

[10] S. Pirzadeh, Progress towards a user–oriented unstructured viscous grid generator, *AIAA Paper 96–0031*, 1996.

[11] A. Bowyer, Computing Dirichlet tessellations, *Comput. J.*, **24**, 162-166, 1981.

[12] D.F. Watson, Computing the n–dimensional Delaunay tessellation with application to Voronoi polytopes, *Comput. J*, **24**, 167-172, 1981.

[13] T.J. Baker, Developments and trends in three-dimensional mesh generation, *Appl. Num. Math.*, **5**, 275-304, 1989.

[14] M.A. Yerry and M.S. Shepard, Automatic three–dimensional mesh generation by the modified octree technique, *Int. J. Num. Meth. Engng.*, **20**, 1965-1990, 1984.

[15] N.P. Weatherill, A method for generating irregular computational grids in multiply connected planar domains, *Int. J. Num. Meth. Fluids*, **8**, 181-197, 1988.

[16] T. Barth, Aspects of unstructured grids and finite–volume solvers for the Euler and Navier–Stokes equations, *VKI Lecture Series on Computational Fluid Dynamics*, Brussels, 1994-05.

[17] H. Borouchaki, P.L. George, F. Hecht, P. Laug and E. Saltel, Delaunay mesh generation governed by metric specifications. Part I: Algorithms. *Finite Elements in Analysis and Design*, **25**, 61-84, 1997.

[18] S. Rebay, Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer–Watson algorithm, *J. Comp. Phys.*, **106**, 125-138, 1993.

[19] D.L. Marcum, Generation of unstructured grids for viscous flow applications, *AIAA Paper 95–0212*, 1995.

[20] N.P. Weatherill, O. Hassan, and D.L. Marcum, Calculation of steady compressible flow fields with the finite element method, *AIAA Paper 93–0341*, 1993.

[21] P.L. George, F. Hecht and E. Saltel, Automatic mesh generator with specified boundary, *Comp. Meth. Appl. Mech Engng.*, **92**, 269-288, 1991.

[22] W. Cho, Topologically reliable approximation of curves and surfaces, *MIT Ph.D. Thesis, Ocean Engineering Department*, 1997.

[23] J. Peiró, J. Peraire and K. Morgan, The generation of triangular meshes on surfaces, *Applied Surface Modelling*, Chapter **3**, C. Creasy and C. Craggs Eds., 25-33, Ellis Horwood, 1989.

[24] K. Nakahashi, FDM–FEM zonal approach for computations of compressible viscous flows, in *Lecture Notes in Physics* **264**, Springer–Verlag, Berlin, 494-498, 1986.

[25] O. Hassan, K. Morgan and J. Peraire, An adaptive implicit/explicit finite element scheme for compressible viscous high speed flows, *AIAA Paper 89–0363*, 1989.

[26] Y. Kallinderis, A. Khawaja and H. McMorris, Hybrid prismatic/tetrahedral grid generation for complex geometries, *AIAA Paper 95–0211*, 1995.

[27] S.D. Connell and M.E. Braaten, Semi–structured mesh generation for 3D Navier-Stokes calculations, *Mechanical Systems Laboratory, GE R& D Center Report 94CRD15*, 1994.

[28] O. Hassan, E.J. Probert, N.P. Weatherill, M.J. Marchant, K. Morgan and D.L. Marcum, The numerical simulation of viscous transonic flow using unstructured grids, *AIAA Paper 94–2346*, 1994.

[29] S. Pirzadeh, Viscous unstructured three–dimensional grids by the advancing layers method, *AIAA Paper 94–0417*, 1994.

[30] O. Hassan, K. Morgan, E.J. Probert and J. Peraire, Unstructured tetrahedral mesh generation for three dimensional viscous flows, *Int. J. Num. Meth. Engng.*, **39**, 549-567, 1996.

[31] O. Hassan, K. Morgan, J. Peraire, E.J. Probert and R.R. Thareja, Adaptive unstructured mesh methods for steady viscous flow, *AIAA Paper 91–1538CP*, 1991.

[32] H. Borouchaki, P.L. George, and B. Mohammadi, Delaunay mesh generation governed by metric specifications. Part II: Applications. *Finite Elements in Analysis and Design*, **25**, 85-110, 1997.

[33] Parasolid v7 System Manuals, Electronic Data Systems, 1995.

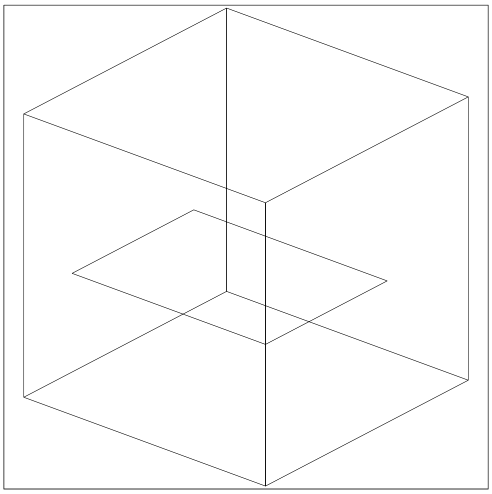

Figure 5: Box: geometry definition.

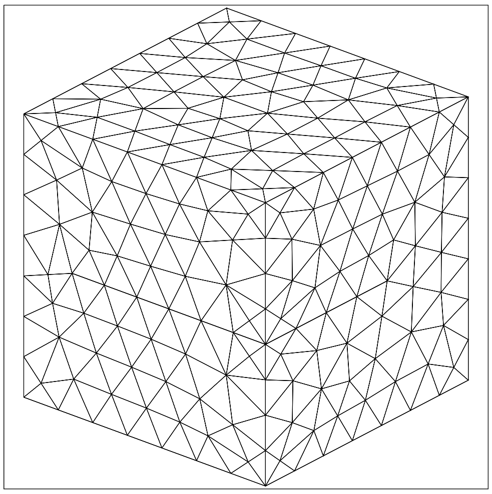

Figure 6: Box: isotropic discretization.

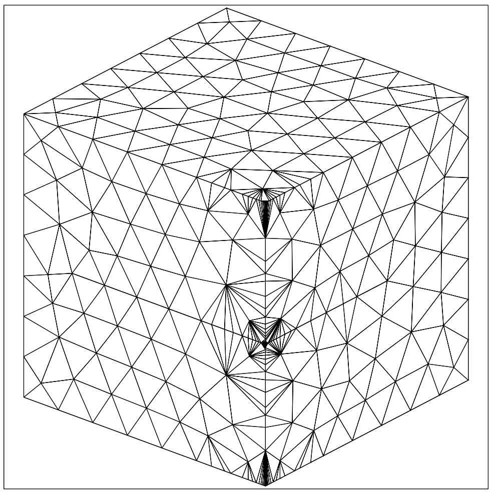

Figure 7: Box: discretization after edge refinement.

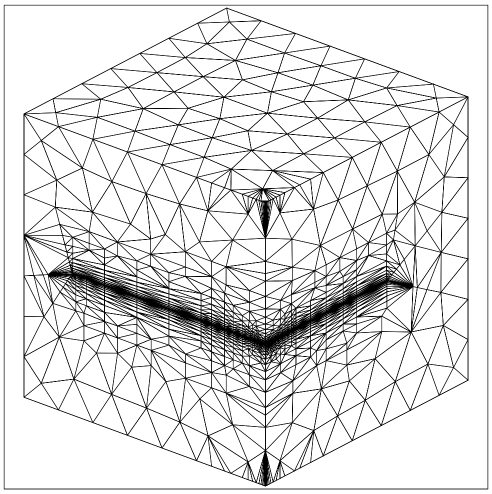

Figure 8: Box: intermediate mesh.

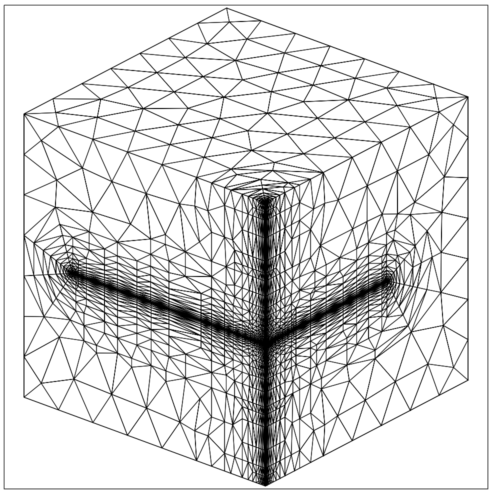

Figure 9: Box: directionally refined mesh.

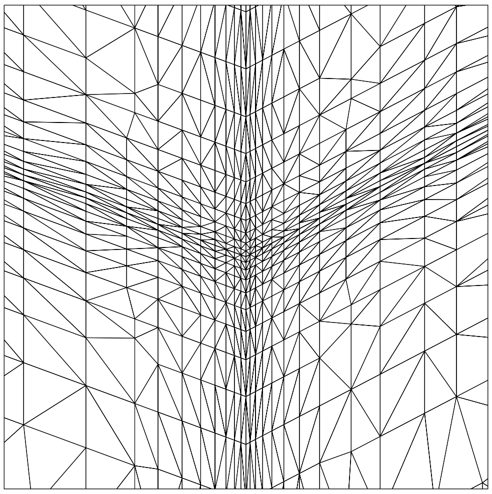

Figure 10: Box: detail of the directionally refined mesh.

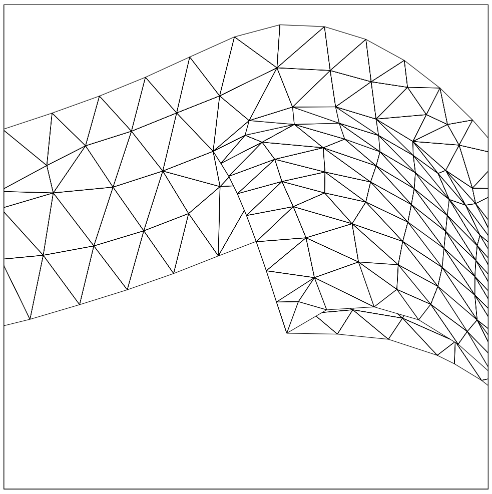

Figure 11: Turbine blade: original unrefined grid

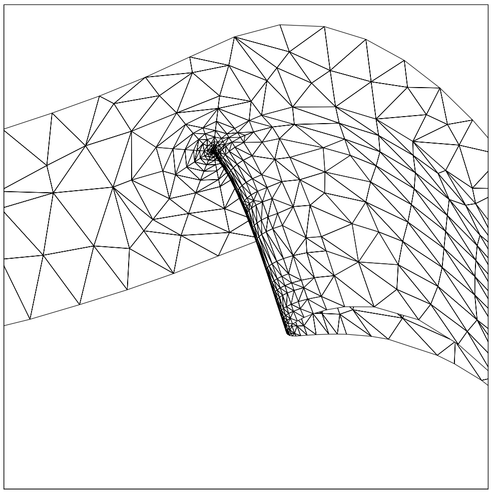

Figure 12: Turbine blade: directional refinement based on distance to blade leading edge

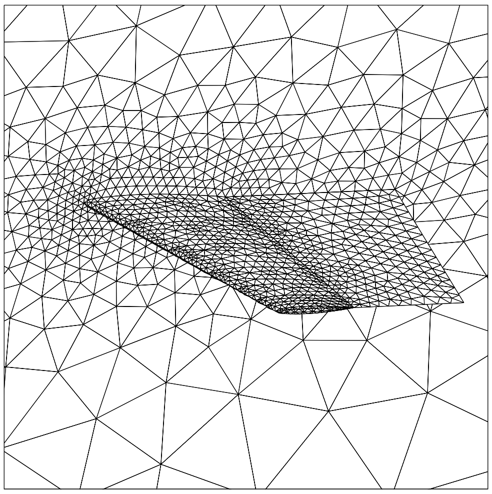

Figure 13: Wing : original unrefined grid

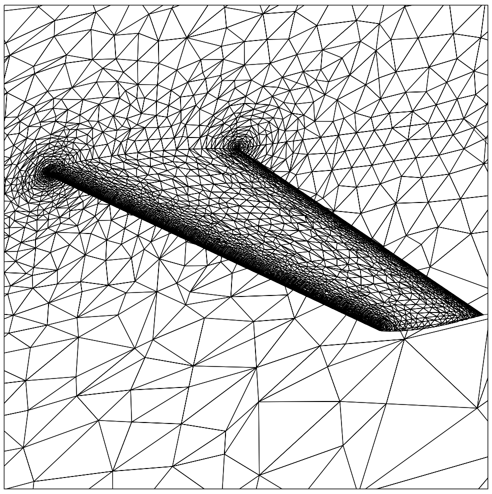

Figure 14: Wing : directional refinement based on distance to leading and trailing edges

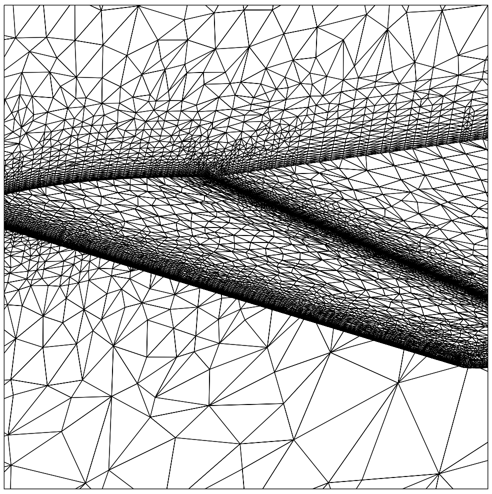

Figure 15: Wing : directional refinement based on distance to wing and wake surfaces

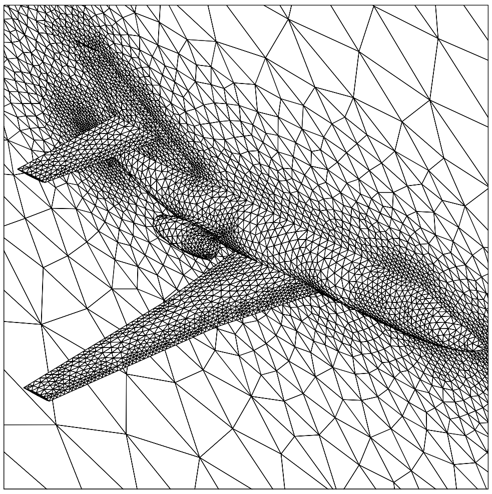

Figure 16: Twin engine jet : original unrefined grid

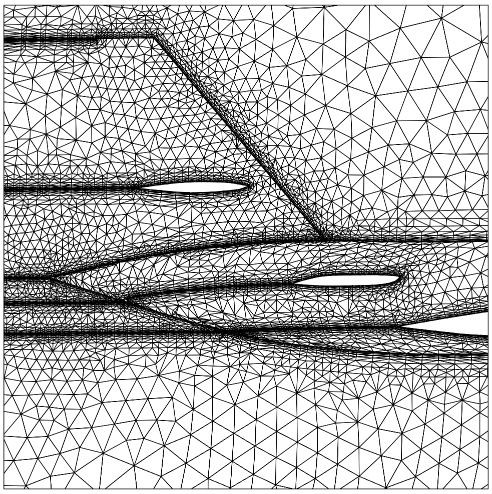

Figure 17: Twin engine jet : directionally refined grid

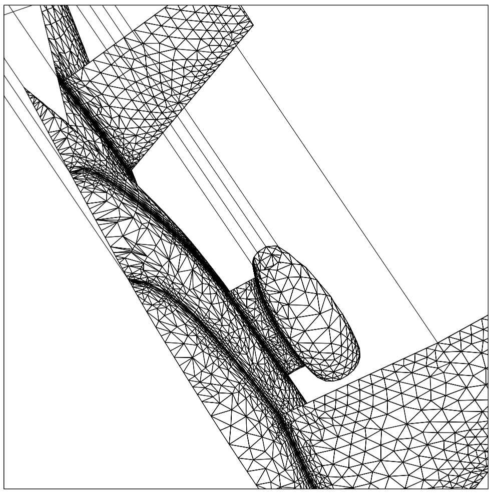

Figure 18: Twin engine jet : directionally refined grid