

Generation of non-isotropic unstructured grids via directional enrichment

Rainald Löhner^{*,†} and Juan Cebra

*Institute for Computational Science and Informatics, M.S. 4C7, George Mason University,
Fairfax, VA 22030-4444, U.S.A.*

SUMMARY

A procedure for the generation of highly stretched grids suitable for Reynolds-averaged Navier–Stokes (RANS) calculations is presented. In a first stage, an isotropic (Euler) mesh is generated. In a second stage, this grid is successively enriched with points in order to achieve highly stretched elements. The element reconnection is carried out using a constrained Delaunay approach. Points are introduced from the regions of lowest stretching towards the regions of highest stretching. The procedure has the advantages of not requiring any type of surface recovery, not requiring extra passes or work to mesh concave ridges/corners, and guarantees a final mesh, an essential requirement for industrial environments. Given that point placement and element quality are highly dependent for the Delaunay procedure, special procedures were developed in order to obtain optimal point placement. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: unstructured grid generation; anisotropic grids; CFD; FEM

1. INTRODUCTION

Many problems of computational mechanics are characterized by a very strong anisotropy in the spatial variation of the fields of interest. A typical example in fluid mechanics is a boundary layer. For laminar flow, the variations in the streamwise direction will be 3–5 orders of magnitude less than normal to it. The same applies to turbulent flows simulated using the Reynolds-averaged Navier–Stokes (RANS) equations with suitable turbulence models. The reliable generation of high-quality grids for RANS simulations has been attempted with varying degrees of success by several authors during the last decade [1–8]. Given that general purpose generation of highly stretched grids has proven difficult, and that computing power is increasing steadily, the immediate question that comes to mind is when RANS simulations will be replaced by large-eddy simulations (LES) or even direct Navier–Stokes (DNS) simulations. Let us assume an optimal mesh for LES simulations.

*Correspondence to: Rainald Löhner, Institute for Computational Science and informatics, M.S. 4C7, George Mason University, Fairfax, VA 22030-4444, U.S.A.

†E-mail: lohner@rossini.gmn.edu

Contract/grant sponsor: AFOSR

This could be an adaptive Cartesian grid that consists of typical (large) Euler cells in the field and very small cells in the boundary layers in order to capture all relevant scales.

Clearly, most points/cells will be located in the boundary layer regions. Denoting by N_p the number of points and by h the characteristic cell size, we have

$$N_p \approx \frac{\text{BLVolume}}{h^3}$$

If we assume, conservatively, a laminar B-747 wing with a chord Reynolds-nr. Of $Re_m = 10^6$, and furthermore assume that the boundary layer obeys the flat plate formula

$$\frac{\delta_{\text{lam}}}{x} = \frac{5.5}{\sqrt{Re_x}}$$

the boundary layer thickness will be approximately $\delta \approx 5 \times 10^{-3}$ m, implying an (isotropic) element size of at most $h \approx 5 \times 10^{-4}$ m. The resulting number of grid-points is then

$$N_p = \frac{N_\delta A_{\text{wing}}}{h^2} = \frac{10 \times 250 \text{ m}^2}{(5 \times 10^{-4} \text{ m})^2} = 10^{10}$$

Current RANS production runs operate with $N_p = 10^7$. From Moore's Law (the doubling of computing power every 18 months), we can foresee LES grids in 15 years. As far as CPU is concerned, the number of timesteps N_t required to advect a particle accurately across the wing is proportional to the number of cells, i.e.

$$N_t = \frac{5 \text{ m}}{10 \text{ m/s}} = O(10^4)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

Assuming the number of floating point operations per point per timestep to be $N_{\text{fpp}} = O(10^3)$, this would result in an operation count of

$$N_{\text{ops}} = O(10^{10})O(10^4)O(10^3) = O(10^{17})$$

Given that the limit of human patience lies somewhere around $O(10^3)$ s, the operation count obtained above implies a CPU performance requirement of $O(10^{15})$ FLOPS. Current production runs can operate at $N_p = 10^{11}$ (100 GFLOPS). Invoking once more Moore's Law, we can foresee LES runs in 15 years. If we perform a sensitivity analysis, we note that the only linear component in these numbers was that of human patience (e.g. 1 h to 1 day). As soon as we increase grid resolution by a factor of 10, we increase the number of points by 10^3 , the number of timesteps by 10, and the total effort by 10^4 , i.e. we have to wait yet another 20 years before we can carry out such a simulation. To make matters worse, the numbers cited above were obtained using the assumption of a laminar boundary layer. Due to the need to capture the laminar sublayer and the higher gradients encountered, fully resolved turbulent flows typically require finer grids than laminar flows. This implies that the numbers cited above should be assumed as very optimistic. Faced with the pressing and immediate need to compute flows where Reynolds-number effects are important, we see that the optimal RANS gridding of geometrically complex domains will remain an important topic of research for the foreseeable future.

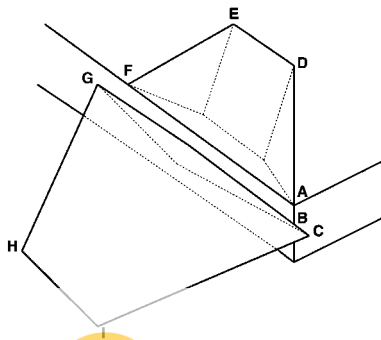


Figure 1. Singular points for generic configuration.

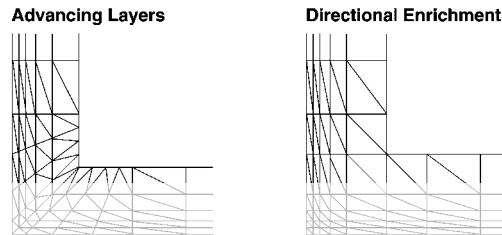


Figure 2. Gridding of convex corners.

2. THE RANS GRIDGING TECHNIQUE

The generation of isotropic unstructured grids has reached a fairly mature state, as evidenced by the many publications that have appeared over the last decade on this subject [9–23] and the widespread use of unstructured grids in industry. The two most widely used techniques are the advancing front technique [10, 11, 16, 19, 22, 23] and the Delaunay triangulation [9, 14, 15, 17, 20, 21, 24]. Hybrid schemes that combine an advancing front point placement with the Delaunay reconnection have also been used successfully [18, 25, 26]. These isotropic mesh generation techniques have been used to generate grids with mildly stretched elements within the context of adaptive remeshing [27–30]. However, they fail when attempting to generate highly stretched elements, a key requirement for Reynolds-averaged Navier–Stokes (RANS) calculations with turbulence models that reach into the sublayer.

A number of specialized schemes have been proposed to remedy this situation [1–7]. The domain to be gridded was divided into isotropic and stretched element regions. In addition, a blending procedure to transition smoothly between these zones was provided. Typically, the stretched mesh region was generated first [2–6]. Although we have used such a scheme [3] for a number of years, we have found several situations in which the requirement of a semi-structured element or point placement close to wetted surfaces is impossible to achieve. Figure 1 shows several such points in the back section of a generic hypersonic flyer. Note in particular that point B is surrounded by at least four different normals, two of which are exactly 180° opposite to each other. Trying to grow a semi-structured mesh with a single normal from this point without producing elements with negative volumes is impossible. Similar problems occur at points A, F and G. These cases clearly point to the requirement of a more general technique, in particular one that allows an arbitrary element connectivity, or number of normals emanating from a point.

A second disadvantage of these ‘advancing layers’ techniques is the fact that in order to preserve the original surface mesh topology as much as possible, corner regions are seldomly gridded optimally. Figure 2 compares the grids produced at a concave corner for an advancing layers procedure and the optimal grid required to capture the spanwise vortices that typically appear there. This observation again points to the requirement of a more general technique, in particular one that allows for the introduction of new surface points while gridding the volume.

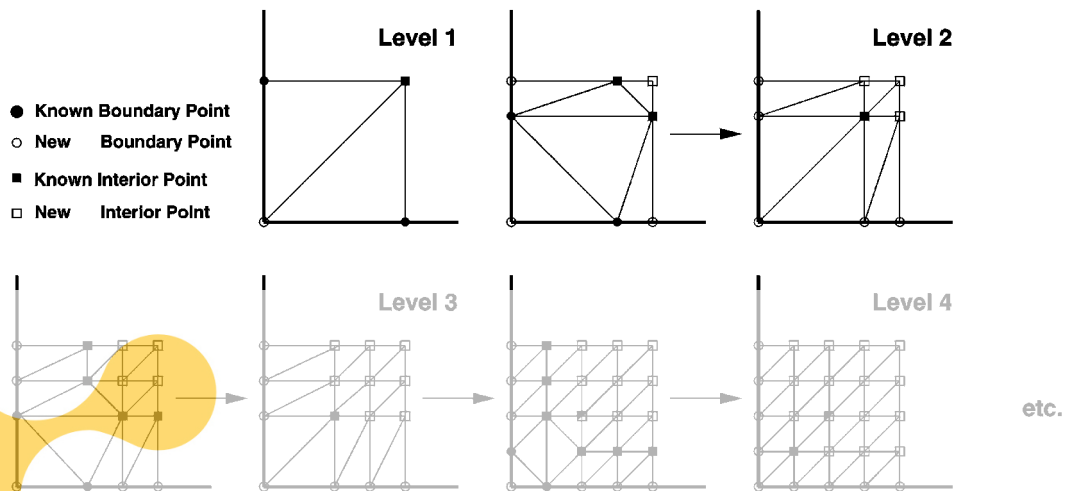


Figure 3. Introduction of points at corners.

The new procedure may be summarized as follows:

1. Generate an isotropic mesh; this can be done with any unstructured grid generator.
2. Remove all points in regions where stretched elements are to be generated.
3. Using a constrained Delaunay technique, introduce points in order to generate highly stretched elements.
4. Introduce the points in ascending levels of stretching, i.e. from the domain interior to the boundary or wake regions.

This procedure has the following advantages:

- (a) Because the procedure already starts with a valid, surface conforming mesh, no surface recovery is required for the Delaunay reconnection, eliminating the most problematic part of this technique.
- (b) Proper meshing of concave ridges/corners is obtained.
- (c) The meshing of concave ridges/corners requires no extra work; this important advantage, which is gained by introducing the points from the domain interior to the boundary or wake regions, is shown in Figure 3.
- (d) Meshing problems due to surface curvature are minimized (this advantage is also gained by introducing the points in ascending levels of stretching).
- (e) A final mesh is guaranteed, an essential requirement for automation.

The disadvantages are the following:

- (i) As with any Delaunay technique, the mesh quality is extremely sensitive to point placement.

The assertion that a valid mesh is always obtained can be made due to the way the algorithm works: at every stage of the point introduction process, the validity of the mesh (neighbour information, topology, element volume, etc.) can be checked. If the introduction of a new point leads to a

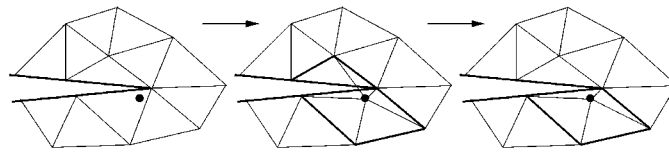


Figure 4. Introduction of field points.

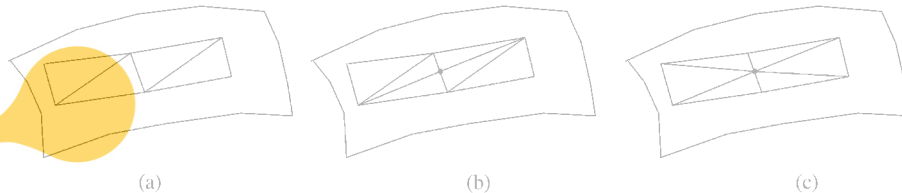


Figure 5. Introduction of surface points: (a) Original surface triangulation; (b) insert point as is and (c) reconnect boundary faces.

non-valid mesh, the point is skipped and stored in a list for later introduction. For the cases shown below, the number of points that could not be introduced was less than 0.001 per cent.

SCIPEDIA

3. INSERTION OF POINTS

Register for free at <https://www.scipedia.com> to download the version without the watermark

The insertion of points is carried out using the constrained Delaunay procedure [31] which may be summarized as follows. Given a new point i at location \mathbf{x}_i :

- (a) find the element(s) \mathbf{x}_i falls into,
- (b) obtain all elements whose circumsphere encompasses \mathbf{x}_i ,
- (c) remove from this list of elements all those that would not form a proper element (based on volume and angles) with \mathbf{x}_i ; this results in a properly constrained convex hull,
- (d) reconnect the outer faces of the convex hull with \mathbf{x}_i to form new elements.

The procedure has been sketched in Figure 4.

For boundary points some additional steps are required. Given a new boundary point i at location \mathbf{x}_i :

- (a) determine if the point is on a boundary edge or face,
- (b) reconnect these elements without regard to the Delaunay criterion,
- (c) find the element(s) \mathbf{x}_i falls into,
- (d) obtain all elements whose circumsphere encompasses \mathbf{x}_i ,
- (e) remove from this list of elements all those that would not form a proper element (volume, angles) with \mathbf{x}_i , this results in a properly constrained convex hull,
- (f) reconnect the boundary faces (see Figure 5),
- (g) reconnect the outer faces of the convex hull with \mathbf{x}_i to form new elements.

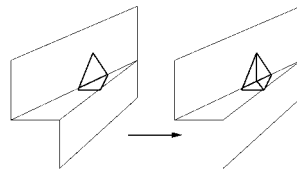


Figure 6. Addition of points along a ridge.

The reconnection of boundary faces is carried out by diagonal swapping. For curved surfaces, it is necessary to apply angle constraints in order not to lose surface resolution/definition or surface patch integrity.

The points are inserted according to layers following the normals emanating from ‘wetted’ surface points, starting from the outermost layer, and moving towards the boundaries or wake centerlines. Points are only introduced if the spacing normal to the wall is below a fraction of the isotropic element size specified by the user at the particular location. This is important for grids with a large variation of element size, and produces a smooth transition from the Euler region into the RANS region.

4. ADDITION OF EXTRA SURFACE POINTS

For complex geometries with narrow surface strips close to concave edges, it is not possible to obtain a good surface mesh unless one introduces further points in these regions. A typical situation is shown in Figure 6.

Additional points are introduced by identifying the corners where potential problems can appear. These corners are typically concave, and are characterized by normals that would introduce points close to another edge. Subsequent reconnection using the constrained Delauney technique would yield elements with very large angles. In order to avoid this, additional points are introduced along the concave edge to mitigate this topological effect.

5. CONSTRUCTION OF NORMALS

The insertion of points to construct highly stretched elements is carried out along normals that may start either on the boundary (boundary layers) or in the field (wakes). The number of normals emanating from a surface point can vary, depending on whether we have a convex or concave surface. Figure 7 shows just a few of a large class of cases that have to be considered.

The point-normals are obtained by comparing the face-normals of the faces surrounding each point. Starting from an arbitrary face surrounding a point, all other face-normals are considered in turn. If the difference of normals, as measured by the scalar product, exceeds a preset tolerance (e.g. 30°), a new point-normal is introduced. In this way, multiple normals emanating from a single point are obtained. For concave ridges or corners (which can be identified by having very large angles between 2 and 3 point-normals and pointing away from the surface) further normals

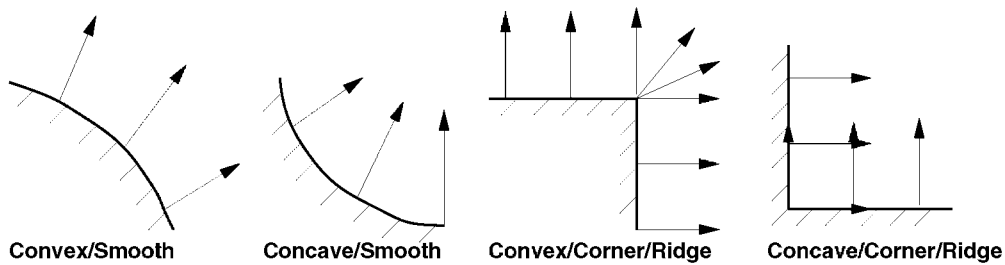


Figure 7. Some possible cases for surface normals.

are introduced with a prescribed angle opening between them. In the examples shown below, this angle opening was set to 30° .

6. REMOVAL OF POINTS BEFORE RANS GRIDDING

Since the quality of grids generated using the Delaunay technique is very sensitive to point placement, it is advisable to remove any (isotropic) points that may interfere with the semi-structured points in the highly stretched regions. The regions where this could happen are identified before starting the RANS meshing. Given the surface and wake patches marked for normal construction and point-introduction, the adjacent elements are marked with an advancing layers procedure, until the distance to the surface or wake center exceeds the distance of the region where stretched elements are required. In most cases, only 2–3 layers of elements are marked for removal. The point removal algorithm may be summarized as follows.

- (1) For each point marked for removal:
 - (a) Obtain all edges touching this point, and the corresponding neighbouring points.
 - (b) Remove the edges that cannot be collapsed due to topological considerations (end-, line-, surface- or volume points).
 - (c) Remove the edges that, if collapsed, would lead to small or negative elements.
 - (d) Remove the edges that, if collapsed, would lead to very small or very large angles.
 - (e) Order the remaining edges according to their length.
 - (f) Remove the marked edges, renumbering the elements.
- (2) Compress and renumber the element and point lists.

The effectiveness of this point removal algorithm may be enhanced by combining it with edge and face swapping and point movement. The final point removal algorithm then takes the form of the following loop:

- (1) DO: For a maximum number of passes:
 - (2) Remove marked points by edge-collapse;
 - (3) IF: Points could be removed:
 - (a) Perform edge and face swapping;

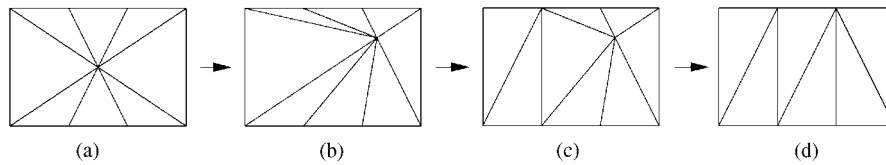


Figure 8. Movement and removal of points.

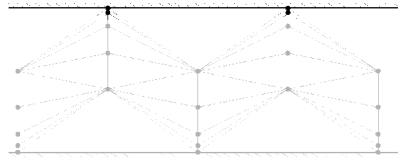


Figure 9. Gridding of gaps.

```
(4) ELSE
```

- (i) Move points that could not be removed;
- (ii) Perform edge and face swapping;

```
(5) ENDIF
```

```
(6) ENDDO
```

The advantage of moving points can be seen from the small example shown in Figure 8. Any edge collapse for the point in the center of the domain would lead to elements with vanishing volumes. Mesh movement in combination with diagonal swapping removes this quandary. This combination of point movement and edge swapping is very efficient, typically leaving only 0.1 per cent of the points marked for removal in the mesh.

7. POINT INTRODUCTION FOR GAPS

In narrow gaps, the introduction of points from two close surfaces covered with RANS grids can lead to very poor mesh quality. Figure 9 shows an example where the resulting mesh is clearly inappropriate for RANS calculations. In order to avoid the introduction of points in such regions, the host element into which the new point falls is checked for proximity to another surface. If any of the points of this element are on the surface and are too close to the new point, the new point is rejected. We remark that the Delauney reconnection procedure requires the determination of the host element, so that this check incurs only a modest amount of CPU.

8. EXAMPLES

The described RANS gridding procedure has been operational for approximately a year, and was used to mesh a series of test cases. Five of these are included here. The isotropic grids required before directional enrichment begins are generated using the advancing front technique [23].

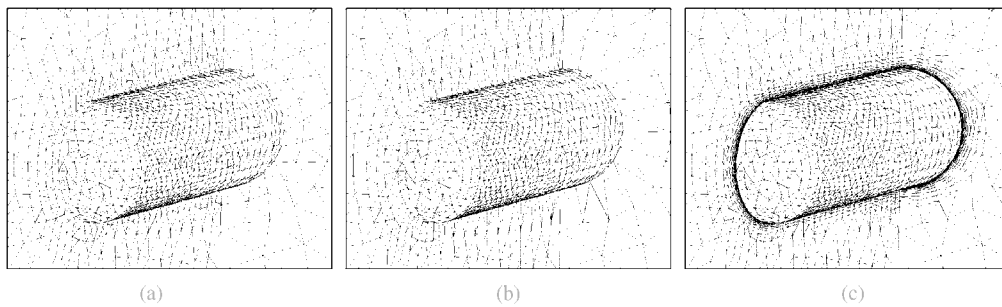


Figure 10. Surface of: (a) isotropic mesh; (b) after element removal; and (c) final mesh.

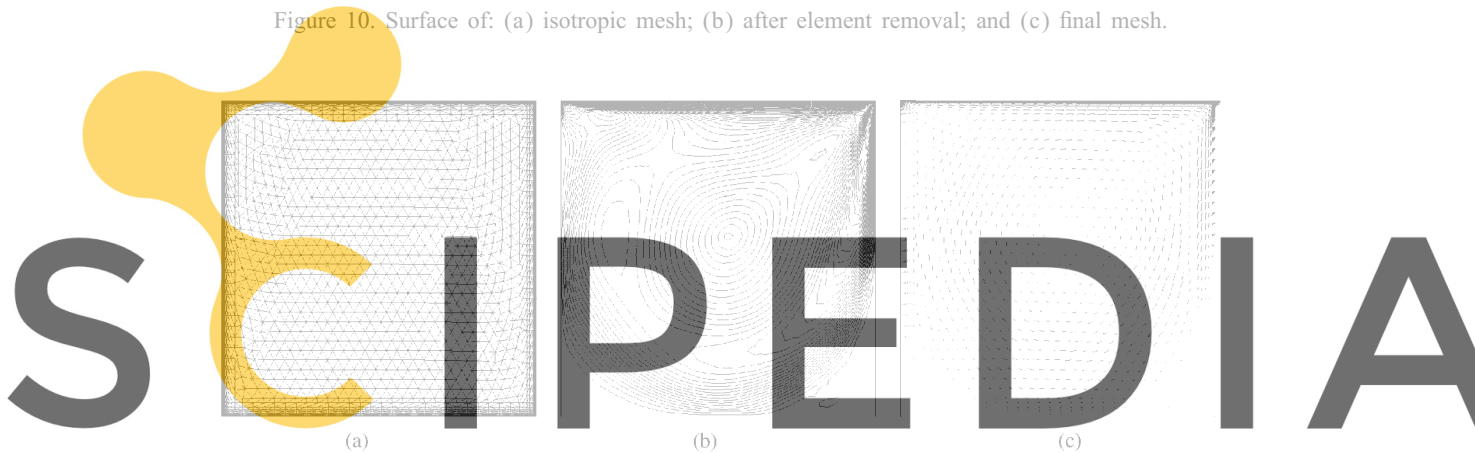


Figure 11. Cavity surface mesh and solution ($Re = 1000$) final mesh.

Register for free at <https://www.scipedia.com> to download the version without the watermark

(a) *Nose-cone*: The surface triangulation of the isotropic mesh is shown in Figure 10(a). The removal of points within the future non-isotropic layers of elements results in the surface mesh shown in Figure 10(b). The final surface mesh is depicted in Figure 10(c). One can see the considerable stretching achieved.

(b) *Cavity*: This is a classic 2-D testcase for incompressible flows. It was run here with a 3-D code, using a parallelepiped of [1.0, 1.0, 0.1], i.e. an extension of 1/10th the side-length in the third dimension. The surface mesh obtained on one of the outside surfaces is shown in Figure 11(a). The results of a simulation for a Reynolds number of $Re = 1000$ are given in Figures 11(b) and 11(c).

(c) *Ship*: This case shows a RANS grid for the generic chemical tanker shown in Figure 12(a). Figures 12(b) and 13(a)–13(c) show surface grids and cross-sections for the generated mesh. The isotropic (Euler) mesh for this case had approximately 1.2 million elements, while the final, anisotropic (RANS) mesh had close to 5 million elements.

(d) *Flyer*: The third configuration considered is that of a generic hypersonic flyer. Figures 14(a) and 14(b) show the surface grids obtained, and Figures 15(a) and 15(b) show a plane cut in the region of the vertical tail and stabilizer. Observe the proper gridding in the corner regions. The isotropic (Euler) mesh for this case had approximately 2 million elements, while the final, anisotropic (RANS) mesh had approximately 6 million elements. On an SGI Origin 2000, using 1 R10000 processor, the isotropic grid generation took approximately 45 min, while the anisotropic

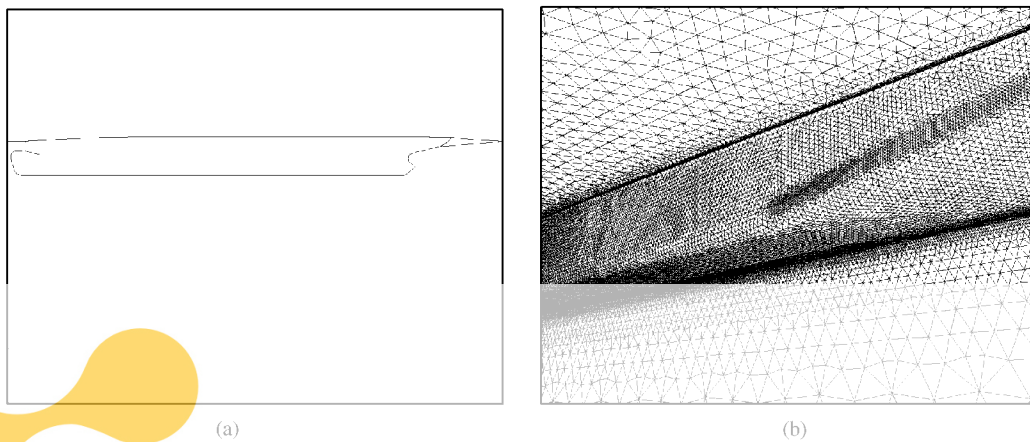


Figure 12. Computational domain and surface mesh along the side.



Figure 13. Transversal cut: (a) stern region; (b) and (c) bow region.

enrichment took approximately 40 min. One can see from these figures that the speed of the proposed RANS gridding technique is acceptable.

(e) *Racecar*: The fourth configuration considered is that of a generic racecar. Figures 16(a) and 16(b) show the surface definition and the overall surface grid, while Figures 17 and 18 focus on particular regions of the mesh (driver, nose of the car, back of the car, gaps, etc.), which had approximately 8.3 million elements. A result for a flow simulation using Luo's implicit LU-SGS-GMRES solver [32] is shown in Figures 19(a) and 19(b) demonstrating the usefulness of the procedure.

9. CONCLUSIONS AND OUTLOOK

A procedure for the generation of highly stretched grids suitable for Reynolds-averaged Navier–Stokes (RANS) calculations has been developed. In a first stage, an isotropic (Euler) mesh is generated. In a second stage, this grid is successively enriched with points in order to achieve highly stretched elements. The element reconnection is carried out using a constrained Delaunay

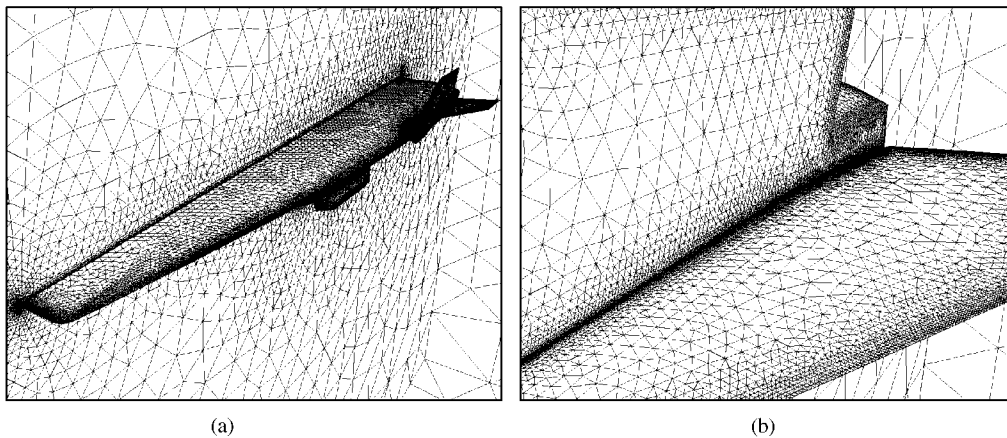


Figure 14. Generic hypersonic flyer: (a) surface grid; and (b) detail.

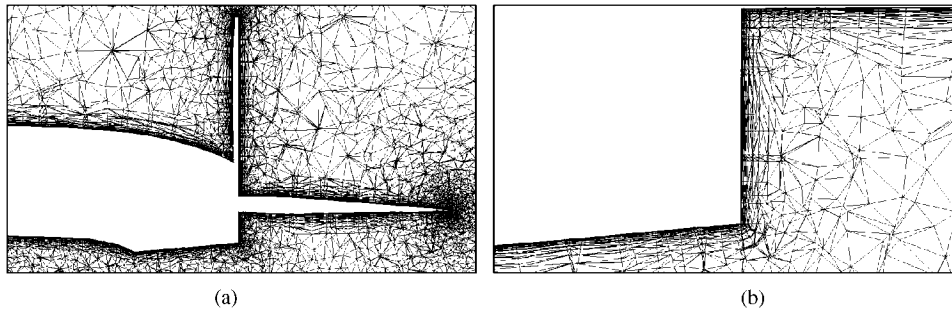


Figure 15. Generic hypersonic flyer: (a) planar cut; and (b) detail.

approach. Points are introduced from the regions of lowest stretching towards the regions of highest stretching. The procedure has the advantages of not requiring any type of surface recovery, not requiring extra passes or work to mesh concave ridges/corners, and guarantees a final mesh, an essential requirement for industrial environments. Given that point placement and element quality are highly dependent for the Delaunay procedure, special procedures are required in order to obtain optimal point placement. Among these, the most important is the removal of points that fall into the RANS zone before enriching the mesh.

Several examples demonstrate the usefulness of the proposed anisotropic gridding technique. Timings from grids generated to date show that the procedure is faster than traditional advancing front techniques for isotropic grids, implying that the generation of anisotropic grids does not place an extra burden on CPU or memory requirements.

Near-future work will centre on:

- (a) improvements in point placement and element control,
- (b) automatic wake placement and topological connection to CAD models, and
- (c) parallelization.

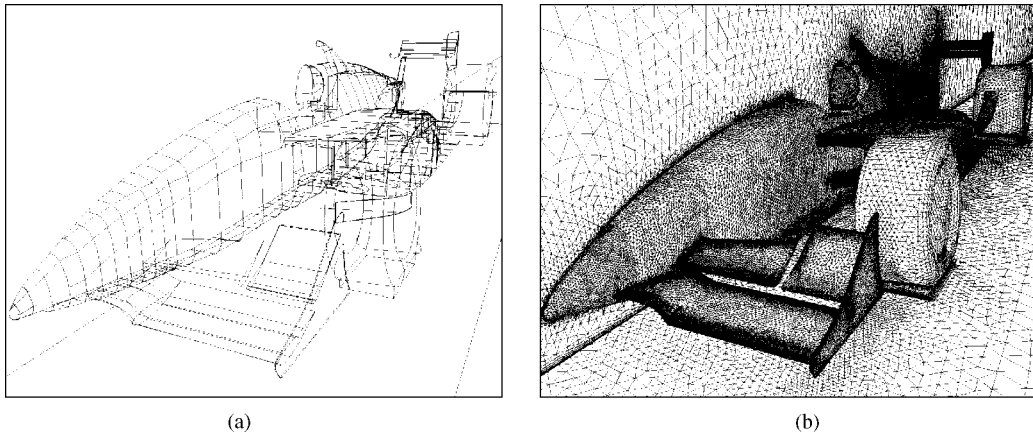


Figure 16. Generic racecar: (a) surface definition; and (b) surface mesh.

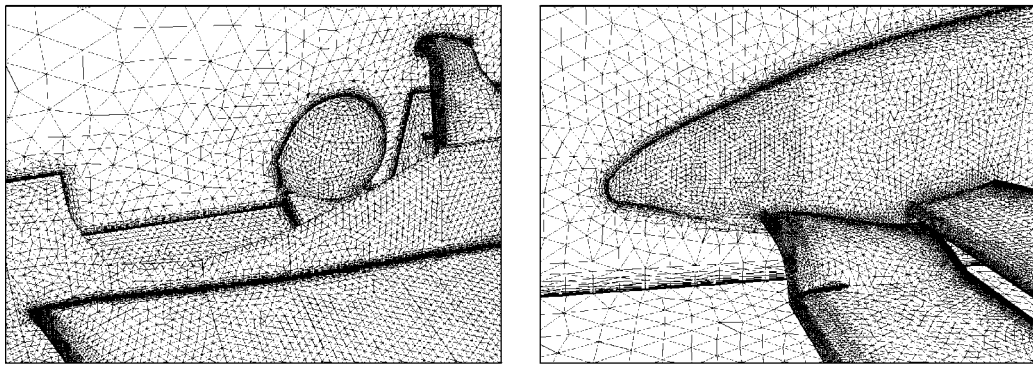


Figure 17. Driver and nose of the car.

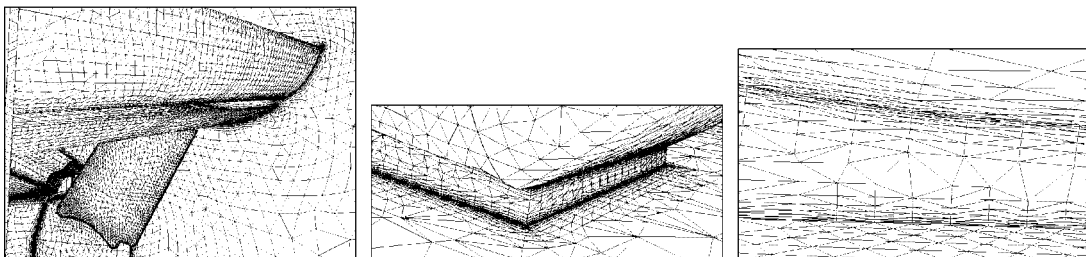


Figure 18. Details of back, back-wheel and car-floor gap.

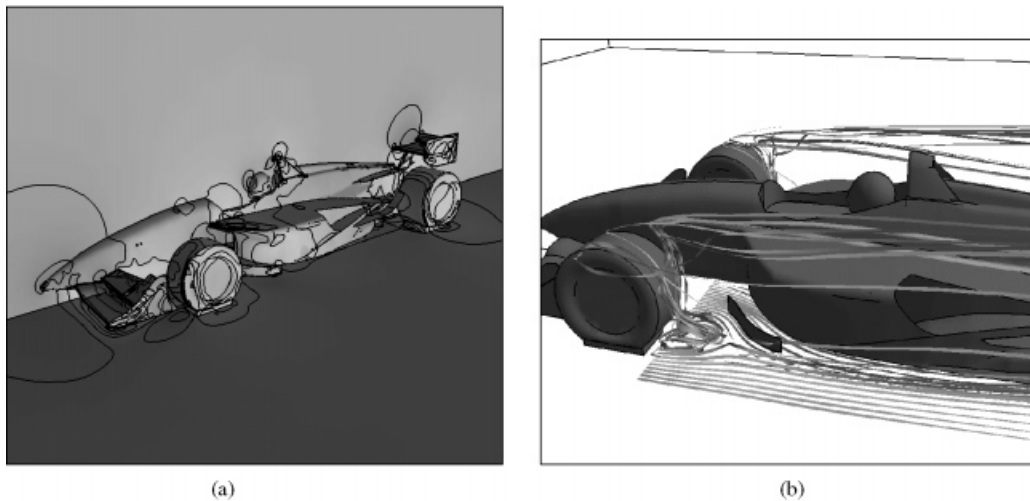


Figure 19. Pressure contours: (a) and (b) streamlines.

Looking further into the future, we envision fully automatic RANS gridders integrated into a multidisciplinary, database-linked framework that is accessible anywhere on demand, simulations with unprecedented detail and realism carried out in fast succession, and first-principles driven virtual reality.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the many insightful discussions with Profs. P. L. George and F. Hecht (INRIA, France), which were instrumental in achieving a fast and robust Delaunay triangulation tool. The data for the generic racecar was provided by Prof. J. Katz (SDSU), and the actual flow solution was obtained by Dr. H. Luo (SAIC).

This work was partially supported by AFOSR, with Dr. Leonidas Sakell as the technical monitor.

REFERENCES

1. Nakahashi K. FDM-FEM zonal approach for viscous flow computations over multiple bodies. *AIAA-87-0604*, 1987.
2. Kallinderis Y, Ward S. Prismatic grid generation with an efficient algebraic method for aircraft configurations. *AIAA-92-2721*, 1992.
3. Löhner R, Matching semi-structured and unstructured grids for Navier–Stokes calculations. *AIAA-93-3348-CP*, 1993.
4. Pirzadeh S. Viscous unstructured three-dimensional grids by the advancing-layers method. *AIAA-94-0417*, 1994.
5. Marcum DL. Generation of unstructured grids for viscous flow applications. *AIAA-95-0212*, 1995.
6. Pirzadeh S. Progress towards a user-oriented unstructured viscous grid generator. *AIAA-96-0031*, 1996.
7. Peraire J, Morgan K. Unstructured mesh generation including directional refinement for aerodynamic flow simulation. *Proceedings of 5th International Conference on Numerical Grid Generation in CFD and Related Fields*, Mississippi, April 1996.
8. Peraire J, Morgan K. Unstructured mesh generation including directional refinement for aerodynamic flow simulation. *AIAA-98-3010*, 1998.
9. Baker TJ. Three-dimensional mesh generation by triangulation of arbitrary point sets. *AIAA-CP-87-1124*, 8th CFD Conference, Hawaii, 1987.
10. Löhner R, Parikh P. Three-dimensional grid generation by the advancing front method. *International Journal for Numerical Methods in Fluids* 1988; **8**:1135–1149.

11. Peraire J, Peiro J, Formaggia L, Morgan K, Zienkiewicz OC. Finite element Euler calculations in three dimensions. *International Journal for Numerical Methods in Engineering* 1988; **26**:2135–2159.
12. George PL, Hecht F, Saltel E. Fully automatic mesh generation for 3D domains of any shape. *Impact of Computing in Science and Engineering* 1990; **2**(3):187–218.
13. Joe B. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design* 1991; **8**:123–142.
14. George PL. *Automatic Mesh Generation*. Wiley: New York, 1991.
15. George PL, Hermeline F. Delaunay's mesh of convex polyhedron in dimension D. Application to arbitrary polyhedra. *International Journal for Numerical Methods in Engineering* 1992; **33**:975–995.
16. Löhner R, Camberos J, Merriam M. Parallel unstructured grid generation. *Computer Methods in Applied Mechanics and Engineering* 1992; **95**:343–357.
17. Weatherill NP. Delaunay triangulation in computational fluid dynamics. *Computational Mathematics and Applications* 1992; **24**(5/6):129–150.
18. Mavriplis DJ. An advancing front Delaunay triangulation algorithm designed for robustness. *AIAA-93-0671*, 1993.
19. Jin H, Tanner RI. Generation of unstructured tetrahedral meshes by the advancing front technique. *International Journal for Numerical Methods in Engineering* 1993; **36**:1805–1823.
20. Weatherill NP, Hassan O. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering* 1994; **37**:2005–2039.
21. Marcum DL, Weatherill NP. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal* 1995; **33**(9):1619–1625.
22. Shostko A, Löhner R. Three-dimensional parallel unstructured grid generation. *International Journal for Numerical Methods in Engineering* 1995; **38**:905–925.
23. Löhner R. Progress in grid generation via the advancing front technique. *Engineering with Computers* 1996; **12**:186–210.
24. Joe B. Delaunay versus max-min solid angle triangulations for three-dimensional mesh generation. *International Journal for Numerical Methods in Engineering* 1991; **31**:987–997.
25. Merriam M. An efficient advancing front algorithm for Delaunay triangulation. *AIAA-91-0792*, 1991.
26. Müller J, Roe PL, Deconinck H. A frontal approach for internal node generation in Delaunay triangulations. *International Journal for Numerical Methods in Engineering* 1993; **17**(2):241–256.
27. Peraire PM, Morgan K, Zienkiewicz OC. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics* 1987; **72**:449–466.
28. Löhner R. An adaptive finite element solver for transient problems with moving bodies. *Computers and Structures* 1988; **30**:303–317.
29. Tilch R. *Ph.D. Thesis*, CERFACS, Toulouse, France, 1991.
30. Habashi WG, Fortin M, Dompierre J, Vallet M-G, Bourgault Y. Anisotropic mesh adaptation: a step towards a mesh independent and user independent CFD. In *Barriers and Challenges in CFD*, Venkatakrishnan V, et al. (eds). Kluwer: Dordrecht, 1998; 99–117.
31. George PL, Hecht F, Saltel E. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering* 1991; **92**:269–288.
32. Luo H, Baum JD, Löhner R. A fast matrix-free implicit method for compressible flows on unstructured grids. *Journal of Computational Physics* 1998; **146**: 664–690.