AIAA JOURNAL
Vol. 39, No. 1, January 2001

Technical Notes

TECHNICAL NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed six manuscript pages and three figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Surface Mesh Generation by Means of Steiner Triangulations

Roque Corral*
Industria de TurboPropulsores, S.A.,
28830 Madrid, Spain
and
Jaime Fernández-Castañeda†
Universidad Politécnica de Madrid, 28003 Madrid, Spain

Introduction

DURING the past few years, a tremendous effort has been done in the development of efficient and accurate algorithms to solve the Navier–Stokes equations about configurations of increasing complexity. Many such attempts have been concentrated in using unstructured grids formed exclusively of triangles or tetrahedra as a result of the flexibility and automation capabilities that such type of elements offer to fill them with an arbitrary computational domain. The discretization of the surface grid is a previous step to the construction of the volumetric grid.

Most of the methods related with surface mesh generation are described as simple extensions of the two-dimensional case[1] and usually underestimate or elude to make explicit reference to the difficulties associated with the quasi-three-dimensional nature of the problem. The direct consequence is that papers specifically devoted to surface mesh generation are scarce or placed outside of the main archival journals. The situation is specially aggravated when dealing with Delaunay-like methods because the basis of most of the documented procedures of surface gridding is the advancing front (AF) approach.[2–4] On the other hand, some authors[2] have reported that surface grid generation requires roughly three times as much CPU time as the standard planar case and is a rather expensive operation.[3] The evidence shows that meshing on an arbitrary surface is a problem substantially more complex than the planar case. If the AF method is used, one of the major difficulties is to find a point on the surface that is equidistant from the two nodes that form the segment of the front under consideration. This problem is easily cast as the one of finding the intersection of a circle and a surface. The computation of this operation is a time-consuming process that can be done either internally or recurring to libraries provided by the solid modeller. To avoid this type of difficulty, some authors have proposed to mesh the surface entirely in the parametric space using a Delaunay method.[5] In this case the algorithm requires the computation of the surface metric from the intrinsic properties of the surface.

This Note presents the experience gained during the development of a surface grid-generation package using Steiner triangulations. First, we will expose the basis of the method for the two-dimensional case. Second, we will go on commenting about the specific problems related with the quasi-three-dimensional nature of the problem and that do not appear on the planar case. Finally we will close the Note raising efficiency and quality issues.

Two-Dimensional Grid-Generation Procedure

At present constrained Delaunay triangulations in two dimensions do not pose notable difficulties. It is the authors' perspective that these types of algorithms are, in principle, simpler and easier to develop than their AF equivalents. However, other types of considerations, much more difficult to estimate, can play an important role in the final decision of choosing between the AF method and Steiner triangulations based in different kinds of connectivities.

Different algorithms have been proposed for the Delaunay tessellation of a given domain; some emphasize constructions based in edge swappings (e.g., Lawson's algorithm), whereas others rely on the incircle property (Bowyer–Watson method). In the present work we have made use of both methods to increase the robustness of the system.

Enforcing the prescribed geometry is usually done in two postprocessing steps. First the edges that define the geometry are required to exist. In the present approach an arbitrary set of edges, not necessarily conforming to a boundary, is forced to exist by executing a sequence of swappings.[6] In a second step all of the triangles placed outside of the domain are flagged, making use of the orientation associated to the closed circuits that define it and of a consistent ordering of the nodes that constitute the triangles.

Interior nodes of the domain are inserted making use of a function that provides a measure of the desired size of the triangles. This element-size function is built just from the information contained in the point distribution of the domain boundaries and of the prescribed curves in its interior. The key idea of the procedure is to use as field-point distribution function the solution of a Laplace equation with Dirichlet boundary conditions.

Mesh quality is improved in two different steps: grid smoothing (that will be described in the next section) and node-degree homogenization. The degree of a node is defined as the number of points that are connected to that node in the grid. The postprocessing step in which nodes of low and high degree are filtered is called node-degree homogenization. This process is useful to break some topological structures that tend to appear during the course of the triangulation and that prevent further improvement of the grid.

We have included in our mesh-generation methodology the capability of removing nodes of degree three, four, and higher than seven, as well as pairs of nodes of degree five. The node homogenization process depends only in the mesh connectivity. After the application of the node homogenization procedure, it is ensured that only nodes of degrees five to seven exist in the grid. (A detailed description of the two-dimensional problem has been previously reported[7] and will not be repeated here.)

Surface Grid-Generation Procedure

The original two-dimensional procedure has been generalized in order to obtain grids on arbitrary surfaces. Here we have closely followed the work reported by Marcum[2] based in an AF point placement strategy and min-max connectivity. In the present version of the algorithm, the AF part has been substituted by a procedure based in adding additional sites to an existing triangulation to improve a

Presented as Paper 98-3013 at the AIAA 29th Fluid Dynamics Conference; Albuquerque, NM, 15–18 June 1998; received 30 November 1998; revision received 26 July 2000; accepted for publication 18 August 2000. Copyright © 2000 by Roque Corral and Jaime Fernández-Castañeda. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Head of Technology and Methods; also Associate Professor, School of Aeronautics. Member AIAA.

†Graduate Research Assistant, School of Aeronautics.

certain measure of grid quality (i.e., a Steiner triangulation). Steiner triangulations have been used to generate planar grids of isotropic and stretched elements[8] and surface triangulations.[1] The main problem on nonplanar surfaces is that it is too expensive to implement ideas that involve real distances (e.g., the incircle test) because, although the concept of a Dirichlet tessellation is well defined in smooth surfaces using the concept of geodesic distance, this is a variational problem whose solution is computationally expensive. To circumvent this difficulty, we have used the edge-swapping algorithm caused by Lawson.[9] The method assumes that a triangulation exists, then all of the interior edges of the mesh are examined. Each of these edges is the diagonal of a quadrilateral formed by the two triangles that share the edge. If the quadrilateral is convex, its diagonal is chosen in such a way that optimizes the selected local measure of quality. It can be shown that if the process is repeated until convergence using the max-min criterion the final mesh is Delaunay. Because we are applying the algorithm in a nonplanar case, we measure the angles in the physical plane. The conjecture is that once the process converges the resulting grid is a Delaunay tessellation. Another alternative would be to solve the problem entirely in the mapped space[5]; in this case a metric based in the intrinsic properties of the surface would have to be created to take into account the three dimensionality of the problem.

Our system uses an approximate physical space grid named MGG (mapped-space generated grid): here we have followed the Marcum's terminology.[2] The aim of this auxiliary grid is to obtain an internal description of the surface, based in a rough surface mesh, that decouples the grid-generation procedure of the CAD system. The true surface definition is only used at the beginning and the end of the process. The basic steps in the overall procedure include the following:

1) Generate a grid entirely in mapped space using the standard planar Delaunay procedure. Obtain the physical space coordinates at the mapped space coordinates using the analytical definition of the surface. This defines the surface grid MGG generated entirely in mapped space.

2) Obtain the physical space coordinates at the MGG coordinates using the CAD definition routine.

3) Generate a first grid in physical space containing just the boundary nodes. Elements placed outside the domain are retained in order to ease search procedures.

4) Triangles not satisfying the element size or curvature criteria are subdivided inserting new nodes in its mapped space centroids.

5) A sequence of swaps is started in order to meet the min-max or max-min criteria. This local reconnection process is repeated until converge.

6) Smooth the node coordinates in physical space.

7) Critical node removal and high-degree node elimination are performed.

8) Steps 4–7 are repeated until convergence.

9) Coordinates in the mapped space are transformed to physical space using the specified geometry definition.

10) Elements outside of the computational domain are eliminated if required.

Sawtooth Prevention Filter

It is well known that the most critical part of a Delaunay method is the initial phase when just a few points have been inserted in the triangulation. In the planar case the potential danger is the failure of the algorithm as a result of round-off errors caused by the nonlocal nature of the operations that take place during this early stage. In the presence of surfaces with moderate to high curvature, another kind of pathological behaviour can appear. We have noticed that under certain circumstances the max-min or min-max criteria can select cell configurations in such a way that the chosen elements lie far away from the actual surface. One of the aforementioned structures is depicted in Fig. 1a. It is clear that this kind of triangle formed during the process of edge swapping should be discarded because they do not constitute a good approximation to the surface. If this cleaning process is not performed on time, this sort of element gives rise on subsequent subdivisions to nodes placed far away from the surface. At this moment the discretized surface is not smooth anymore, and in practice it is observed that it is difficult to recover the nominal surface departing from this situation. The problem is specially aggravated in the presence of grid-smoothing mechanisms that can generate singular grids on the mapped space but geometrically feasible in the physical space (Figs. 1c and 1d). An additional filter has been devised to prevent the appearance of these sawtooth patterns. Every time the possibility of an edge swapping is checked, we verify whether we are dealing with a sawtooth or not. To take that decision, we compare the length of the inspected edge d_0, with the size of the smallest of the four edges of the quadrilateral formed by the two triangles that share the edge $d_{\min}$, if d_0 is smaller than $d_{\min}$, the edge is forced to exist, either preventing or forcing the swapping to occur independently of other type of considerations. The aim of the filter is to avoid the generation of pairs of almost equilateral triangles, fold about its common edge and with the opposite vertices placed very close of each other (Fig. 1a). The filter favors the alternative option (Fig. 1b) and prevents the formation of these undesirable patterns, optimal from the point of view of the standard swapping criteria, but that in very coarse meshes give rise to elements that have nothing to do with the actual surface. When the number of points is increased and the grid starts to roughly describe the surface, the filter gradually loses its role, and almost no corrections are detected in the latest phases of the gridding process.

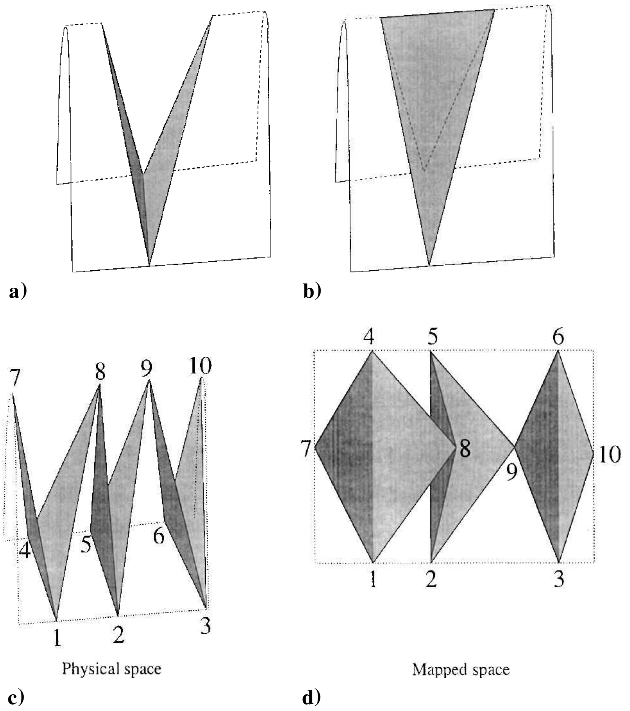

Fig. 1 In the upper part of the figure, two possible configurations for a set of four nodes are sketched. a) This structure is possible within the context of a min-max or max-min swapping criteria because the triangles sharing the edge may be equilateral. b) This structure is forced in our method. c) and d) These structures depict how the problem is aggravated in the presence of smoothing.

The difficulties that might be encountered are illustrated in Fig. 2, where different stages of the generation of a torus surface grid are sketched. The initial mesh (Fig. 2a) is composed by nodes located in the boundaries of the parametric representation of the torus, and the triangles that form the grid are located far away from the actual surface. At this instance special care must be taken to avoid the generation of the sawtooth patterns. In this particular case these would have formed close to the nodes located in the parametric line that defines the minor radius of the torus. In Fig. 2b a series of crests formed by triangles can be observed. The nodes located in the ridge have been forced to be connected by means of this filter; otherwise, the min-max or max-min criteria would have created a set of triangles joined through its bases. Once the surface is roughly

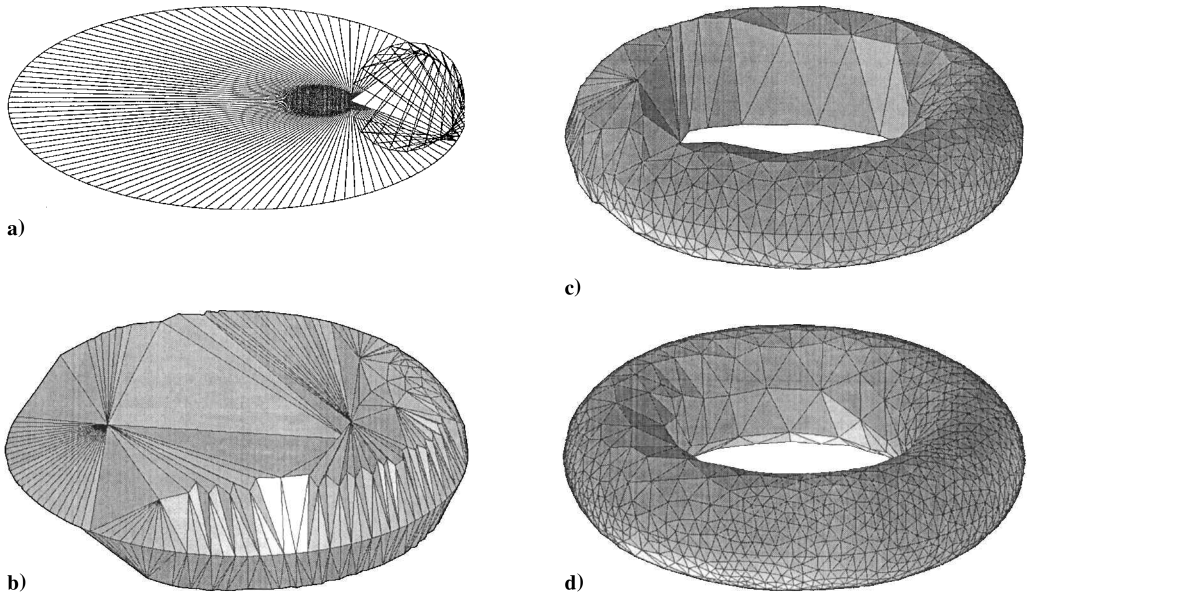

Fig. 2 Sequence of grids during the generation process of a torus surface mesh. a) Corresponds to the initial mesh and contains 280 boundary nodes; b), c), and d) represent next three steps with 275, 594, and 715 more points, respectively.

represented by the surface grid (Fig. 2d), this filter does not modify the basic swapping criteria anymore.

Grid Smoothing

One of the most common procedures to improve mesh quality is the smoothing of the grid coordinates. This process is carried out in the present work by the application of the following Laplacian-like smoothing:

$$\boldsymbol{x}_j = \boldsymbol{x}_j + c\left(\frac{1}{N}\sum_{\text{neigh}} \boldsymbol{x}_{\text{neigh}} - \boldsymbol{x}\right) \tag{1}$$

where $\boldsymbol{x}_j$ is the grid point coordinate vector, $\boldsymbol{x}_{\text{neigh}}$ are the coordinates of the nodes that surround the point j, and N is the number of those nodes. The problem arises when the mesh nodes are located on a surface; in this case the guarantee has to be made that the node movements are contained on the surface $\boldsymbol{x} = \boldsymbol{x}(u, v)$. The best way to fulfill this condition is to perform the smoothing in the mapped space, i.e.,

$$u_j = u_j + c\left(\frac{1}{N}\sum_{\text{neigh}} u_{\text{neigh}} - u_j\right) \tag{2}$$

and in the same manner for v. The problem is that the smoothing has to be carried out in the physical space because the application of the Laplacian operator in the mapped space tends to improve the grid distribution in this plane and to deform the grid in the physical space. To fix this problem, it is better to see the Laplacian filter as an iterative process to reach the equilibrium state of a set of linear springs of identical stiffness, which joins the nodes with the same topology than the mesh (the spring analogy). The surface can then be imagined covered by a network of springs for which the mechanical equilibrium condition states that the net tangential force for each node has to vanish. The projection of the force exerted by one of such springs along the lines $u =$ const can be expressed as

$$F_u = K(\Delta s)_u = K\left(\frac{\mathrm{d}s_u}{\mathrm{d}u}\right)_{v\,=\,\text{const}} \Delta u \tag{3}$$

where K is the stiffness of the spring, Δs_u is the projection of the distance $s_{\text{neigh}} - s_j$ along the line $v =$ const, and $\Delta u = u_{\text{neigh}} - u_j$. Likewise the iteration procedure for of u can be expressed as

$$u_j = u_j + c\left(\frac{1}{N}\sum_{\text{neigh}} \boldsymbol{x}_{\text{neigh}} - \boldsymbol{x}_j\right)_u \left(\frac{\mathrm{d}s_u}{\mathrm{d}u}\right)^{-1}_{j,v\,=\,\text{const}} \tag{4}$$

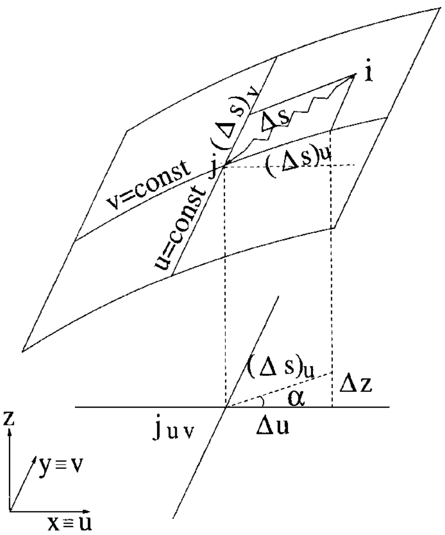

Fig. 3 Geometric derivation of grid-smoothing formula.

where the first bracket is the projection of distance between nodes j and neigh along the lines $v =$ const. The former equation represents the transformation of the length increments along $v =$ const in the corresponding increments in u, that is the updated variable. Figure 3 shows a graphical construction of Eq. (4). To ease the interpretation, the mapped plane (u, v) and the (x, y) plane coincide in this case. The derivative term can be geometrically expressed as

$$\left(\frac{\mathrm{d}s}{\mathrm{d}u}\right)_{j,v\,=\,\text{const}} = \frac{1}{\cos\alpha} \tag{5}$$

A smoothing coefficient c of 0.5 is used for all of the interior points, whereas for the nodes adjacent to the boundaries the coefficient is reduced to 0.25. Typically three smoothing steps are performed.

Refinement Criteria

The mesh-generation procedure is based on the subdivision of the elements of an initial grid by the insertion of new nodes in the centroid of the triangles flagged for refinement. Some criteria have then to be established to select which cells will be subdivided.

The first step is to interpolate the value of the element size function in the centroid of the cell s_c. If s_c is less than one-third of the perimeter of the cell, the element is flagged for refinement. The cell is deleted from the list of cells and three new triangles formed by joining the vertices of the erased cell and its centroid are created.

It is usually expected that the triangulated surface deviates less than a given tolerance from the actual surface. This requirement cannot be directly achieved in a warp surface by using the preceding criterion because this does not have the ability to account for effects associated with the local curvature of the surface. To avoid this limitation, the centroid of cell in the mapped space is calculated, and two auxiliary points are obtained in physical space for this unique node on the parametric plane: one as contained in the current grid and the other as belonging to the MGG grid. If the distance between these two points is greater than the prescribed tolerance, the cell is flagged for refinement.

In surfaces with high curvature, after some insertion steps the threshold imposed by the element size function is always satisfied, and cells are only flagged for refinement by the curvature criterion. When this situation arises, the node degree homogenization techniques erase many of the new points, and the number of insertion steps needed to reach a converged state is large. To alleviate this problem, the procedure used to flag cells by curvature is tailored to mark not only the "suspicious" element but its neighbors as well. The final result is that the overall process is accelerated and a smoother transition between areas of different curvatures is reached.

The same procedure is repeated for each cell, and upon completion of the flagging procedure a sequence of swappings is started in order to improve the quality of the new grid.

Application Examples

During the development stage of the algorithm, we have used a number of elemental surfaces (e.g., paraboloids, cones, tori, etc.) with different topological properties to check the validity of the method. For the sake of conciseness, we will concentrate here in two cases: the first test case is a four-sided surface patch with a uniform node spacing in the mapped space that represents a paraboloid in the physical space. This simple example may give an idea of the asymptotic quality that can be reached by the method. However, the nonuniformity of the boundary nodes and the presence of acute corners in physical coordinates induced by the parameterization can well represent typical difficulties encountered in practical situations. The statistics of the grid built with the Q3D algorithm and the grid that would be obtained if a two-dimensional mesh were generated in the plane, and then the nodes mapped into the surface are shown in Fig. 4. It may be appreciated that the quality of the final grid is high but still below the excellence demonstrated by the AF point placement strategies. The geometry of the second case represents a blade of a low-pressure turbine. It contains the aerodynamic surface, the tip shroud, the platform, and the attachment to the disk and consists of 106 different surfaces. The wire-frame representation of the solid and two details of the tip shroud and the platform are shown in Fig. 5. The blade contains different length scales, and it can be considered as an example of a complex configuration.

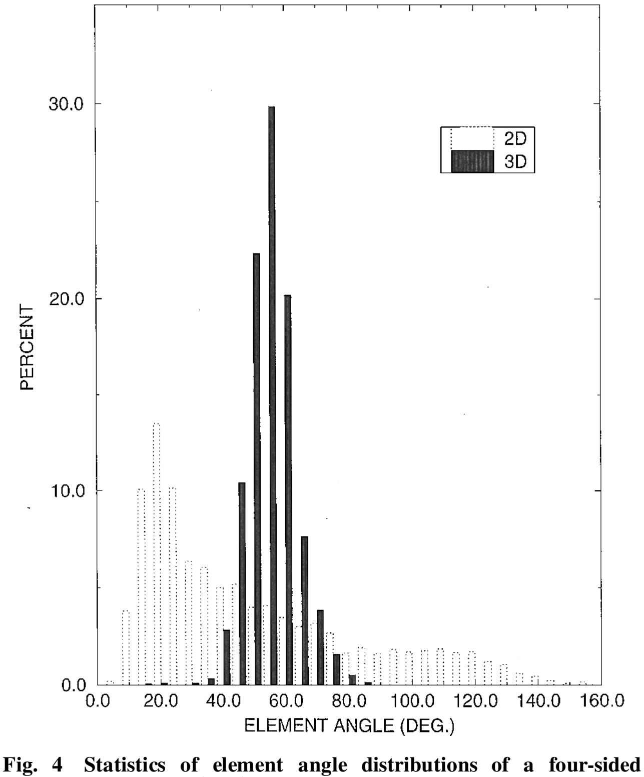

Fig. 4 Statistics of element angle distributions of a four-sided paraboloid patch $z = x^2 + y^2$ with 2195 nodes and 4268 cells for the quasi-three-dimensional grid and the planar grid mapped into the surface.

Table 1 CPU time breakdown for the generation of a mesh of 10^5 elements on a paraboloid[a]

Process	Time, s
Build MGG (1)	0.34
Creation of new points (9)	6.34
Compute physical space coordinates	4.89
Point triangulator (9)	63.34
Watson's algorithm (3)	0.99
Points direct insertion (6)	31.25
Swapping process (28)	30.20
Node degree homogenization (8)	113.72
Case of degree 3 (33)	2.21
Case of degree 4 (50)	8.79
Pairs 5-5 (62)	16.69
Case of degree >7 (10) (151 calls to Watson's algorithm)	82.54
Grid smoothing (8)	2.12
Element size function smoothing (8)	4.95
Total	190.89

[a] The numbers in parentheses indicate the number of calls of the routine. The test case was run on an IBM 39H.

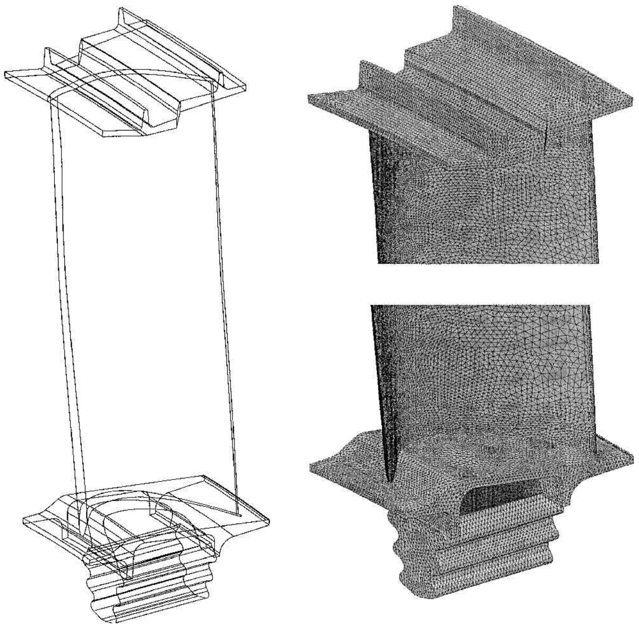

Fig. 5 Wire-frame representation of an LPT blade (left) and close-ups of the mesh in the tip shroud (right top) and the platform (right bottom) regions. The solid model is defined by 102 faces and 304 edges. The surface mesh contains 57,000 nodes and 115,000 cells.

CPU Time Breakdown

To measure the efficiency of the method, a mesh of 10^5 cells was obtained on a paraboloid with the default values of all of the parameters used in the code. The time breakdown is shown in Table 1. With the present procedure a mesh of 10^5 cells requires about 191 s of CPU time. It can be seen that the most expensive part of the algorithm is the elimination of nodes of degree higher than seven. The cause of this inefficiency has not been tracked in detail but must be attributed to a bad practice in the implementation of the algorithm. If the node-degree homogenization techniques are not employed, then the time is reduced to 70 s approximately.

It has been reported[2] that the time necessary to generate a two-dimensional grid with 10^5 elements it is about 17 s on an SGI Indigo[2] R4400. The same author claims that the equivalent method for the generation of a surface grid requires three times as much CPUtime as the two-dimensional planar procedure. Thus the total cost can be estimated in about 1 min of CPU time, which is more or less the same amount obtained if the node-degree homogenization technique is removed from Table 1.

Conclusions

A method for the efficient generation of unstructured grids on arbitrary surfaces based in Steiner triangulations has been presented. The mesh-generation system uses an internal representation of the surface to isolate the grid-generation and CAD systems. The algorithm does not need the tracking of fronts, and therefore its implementation is attractive because of its simplicity. A special filter has been devised to avoid the generation of sawtooth patterns on the surface during the early stages of the gridding process, and this has proved to be an essential part of the method. The quality of the surface and planar grids obtained by the present approach are similar, but still below the attainable by means of an AF point-placement strategy.

Acknowledgments

This project has been funded by the Industria de Turbo Propulsores under Contract ITP-002-96 to the School of Aeronautics of the Universidad Politécnica de Madrid, and its permission to publish this Note is gratefully acknowledged. The authors would like to thank R. Gómez for an early implementation of a constrained version of the Bowyer–Watson algorithm and J. M. Barroso for help and useful comments during the development of this work.

References

[1] Barth, T. J., "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier–Stokes Equations," *Unstructured Grid Methods for Advection Dominated Flows*, R-787, AGARD, 1992, pp. 6-1, 6-61.

[2] Marcum, D. L., "Unstructured Grid Generation Components for Complete Systems," Short Course on Advances in Numerical Grid Generation, Mississippi State Univ., March 1996.

[3] Peraire, J., and Morgan, K., "Viscous Unstructured Mesh Generation Using Directional Refinement," *Proceedings of the V International Conference on Grid Generation in Computational Field Simulation*, 1996, pp. 1151–1163.

[4] McMorris, H., and Kallinderis, Y., "Octree-Advancing Front Method for Generation of Unstructured Surface and Volume Meshes," *AIAA Journal*, Vol. 35, No. 6, 1997, pp. 976–984.

[5] Borouchaki, H., George, P. L., Hecht, F., Laug, P., and Saltel, E., "Delaunay Mesh Generation Governed by Metric Specifications. Part I: Algorithms," *Finite Elements in Analysis and Design*, Vol. 25, No. 1–2, 1997, pp. 61–83.

[6] Sloan, S. W., "A Fast Algorithm for Generating Constrained Delaunay Triangulations," *Computers and Structures*, Vol. 47, No. 3, 1993, pp. 441–450.

[7] Corral, R., and Fernández-Castañeda, J., "Surface Mesh Generation by Means of Steiner Triangulations," AIAA Paper 98-3013, June 1998.

[8] Barth, T. J., "Steiner Triangulation for Isotropic and Stretched Elements," AIAA Paper 95-0213, Jan. 1995.

[9] Lawson, C. L., "Software for C^1 Surface Interpolation," *Mathematical Software III*, edited by J. R. Rice, Academic Press, New York, 1997.

J. Kallinderis
Associate Editor

Continuous-Phase Properties of Homogeneous Particle-Laden Turbulent Flows

J.-H. Chen[*] and G. M. Faeth[†]
University of Michigan,
Ann Arbor, Michigan 48109-2140

Nomenclature

C_D	=	particle drag coefficient
D	=	dissipation factor; Eq. (2)
d_p	=	particle diameter
f_i	=	volume fraction of region i
n''	=	particle number flux
U_p	=	mean streamwise relative velocity of a particle
u, v	=	instantaneous streamwise and cross-stream gas velocity
$\bar{u}, \bar{v}$	=	mean streamwise and cross-stream gas velocity
$\bar{u}', \bar{v}'$	=	rms fluctuating streamwise and cross-stream velocity
ε	=	local rate of dissipation of turbulence kinetic energy
φ	=	average generic property of the overall flow
φ_i	=	average generic property of region i of the flow

Subscripts

i	=	turbulent interwake region
w	=	particle wake region

Introduction

TURBULENCE generation is defined as the direct disturbance of the continuous-phase velocity field by the wakes of dispersed-phase objects in dispersed multiphase flows. Turbulence generation supplements the conventional production of turbulence caused by mean velocity gradients in the continuous phase; it is most important when dispersed-phase objects have large relative velocities (large Reynolds numbers) and relatively large relaxation times compared to characteristic turbulence times. Such conditions are typical of many practical dispersed multiphase flows having significant separated-flow effects, e.g., sprays, particle-laden jets, bubbly jets, rainstorms, etc. Motivated by these observations, the present investigation sought to develop methods to predict the overall continuous-phase properties of homogeneous particle-laden turbulent flows dominated by turbulence generation and to evaluate the predictions based on earlier measurements from this laboratory.[1–5]

Initial observations of turbulence generation in this laboratory considered uniform fluxes of nearly monodisperse spherical particles in still water and air to yield homogeneous and stationary flows where turbulence production was caused entirely by turbulence generation.[1,2] The local rate of dissipation of turbulence kinetic energy ε mainly controls continuous-phase properties in these flows and can be found as the local rate of loss of particle mechanical energy per unit volume, as follows:

$$\varepsilon = \pi n'' d_p^2 C_D U_p^2 | 8 \tag{1}$$

All other properties of these flows, however, are not known and must be related to dissipation rates and particle properties. Finally,

Presented as Paper 2000-0182 at the AIAA 38th Aerospace Sciences Meeting, Reno, NV, 11–14 January 2000; received 8 March 2000; revision received 15 September 2000; accepted for publication 27 September 2000.

[*] Graduate Student Research Assistant, Department of Aerospace Engineering; currently Assistant Professor, Department of Architecture and Marine Engineering, National Cheng-Kung University, Tainan 701, Taiwan, Republic of China.

[†] A.B. Modine Professor, Department of Aerospace Engineering, 3000 François-Xavier Bagnoud Building, 1320 Beal Avenue. Fellow AIAA.