THE GENERATION OF 3D, STRETCHED, VISCOUS UNSTRUCTURED MESHES FOR ARBITRARY DOMAINS

W.N. Dawes,
Cambridge University Engineering Department,
Whittle Laboratory,
Cambridge, UK.

ABSTRACT

This paper presents a procedure which can generate highly stretched, unstructured, viscous meshes for essentially arbitrary three-dimensional configurations. The procedure is based on a combination of the ideas behind Delaunay and moving front methodologies. Examples are presented for a variety of turbomachinery types of flow to demonstrate the potency of the new approach. Recent enhancements to the author's flow solver are also briefly described.

INTRODUCTION

There is little doubt that in the future the efficient simulation of a wide variety of practical, complex geometry flows will take place using CFD based on unstructured mesh systems. There is also little doubt that practical, three dimensional Navier-Stokes simulations cannot proceed unless these unstructured mesh systems support highly stretched, highly anisotropic computational cells.

Practical simulation of high Reynolds number flow, especially three dimensional flow, demands highly stretched mesh structures near solid surfaces; it is simply not viable nor rational to refine nearly equilateral, isotropic mesh down into the laminar sublayer. A flow past a blade at a Reynolds number of one million would need a mesh with a wall-normal mesh spacing of around chord/1000 to avoid the use of wall functions. If the mesh refinement to achieve this takes place in the context of an isotropic mesh then the streamwise mesh spacing would also be of order chord/1000 which implies of order one million nodes for a two-dimensional simulation domain and one billion for three dimensions. The higher Reynolds numbers seen in steam turbines and on aircraft wings would be prohibitively expensive.

For nearly a decade, unstructured meshes have been developed and used for inviscid flow around and through increasingly complex flow domains (see for example the very early efforts of Jameson et al 1986, Peraire et al 1988). However, most mesh generating technology has sought, or been produced by, algorithms which assume, nearly isotropic, equilateral meshes. Three classes of generation methodology have emerged: Delaunay (for example Baker 1987), Moving Front (for example Lohner 1988) and Octree (Sheppard et al 1991). All three were conceived for and are considered to be best suited to producing near isotropic, near-equilateral meshes.

The crux of the Delaunay approach is that the unique triangulation of an arbitrary distribution of points is obtained when the nodes are connected into cells such that the circumcentre (sphere) of each triangular (tetrahedral) cell contains no other points; the weakness of the approach is that the nodes themselves are an input, not a result. In the Moving Front method in two (three) dimensions the initial boundary domain is advanced segment by segment (face by face) into the domain by adding a new node (or using an existing, suitable node) and connecting this to the segment (face) into a new triangle (tetrahedron). The Octree method is based on successive division of cartesian hexahedral cells (or equilateral tetrahedrons) until conforming to body topology and then "snapping" nodes onto the surfaces of the body itself. This method produces a very good mesh in the interior but a very poor mesh near the body surfaces and need not be considered any further here.

By contrast to meshes for inviscid work, unstructured meshes for viscous flows have been produced in an ad hoc way, and out of urgent necessity, by various artifices. These include shredding multi-block meshes containing already highly stretched viscous zones into tetrahedrons (Dawes 1992); embedding local, highly anisotropic, structured mesh zones within an otherwise nearly isotropic unstructured mesh (Hassan et al 1994); and "surface inflation" by marching hexahedral elements out of the surface to
some suitable distance, triangulating them and then combining
them with an isotropic unstructured mesh generated outboard
(Connell et al 1995). Techniques have also been developed in two
dimensions based on locally transforming the coordinate system
into a highly stretched space, generating an isotropic mesh and then
reverting the transformation to produce a stretched mesh in physical
space (see review by Mavriplis 1995) but these methods appear not
to translate at all to three dimensions.

MESH GENERATION STRATEGY

CAD-like geometry definition

To support essentially arbitrary geometry, the input to the new
mesh generator is a CAD-like model. This model is built using sets
of point-strings (poly-lines) bound into surfaces using a simple
topology model. Each surface is then represented by a tensor product
patch (for example a simple Coons patch). The basic setup is
illustrated in Fig. 1. Of the three elements (wireframe, patches and
topology) the topology model is the most significant (see also

Triangulation/tetrahedralisation

To connect the nodes into a valid mesh (and hence place new
nodes into an existing mesh) the Delaunay method was chosen. Of
the three candidate methods described in the introduction only
Delaunay is "guaranteed" (subject to finite word length arithmetic)
to always produce a valid mesh. The choice was made following
numerical experimentation which lead to the observation that
stretched right angle triangles are capable of being Delaunay and
that sets of stretched right angle triangles are also capable of all
being Delaunay and can be constructed provided the node placement
strategy keeps nodes out of the circumspheres of the
target set of stretched right angle triangles. From the point of view
of truncation error it is believed that stretched right angle triangles
are to be preferred to the alternative type of stretched triangle with
two very small and one very large angle.

Delaunay triangulation/tetrahedralisation

The essence of a Delaunay triangulation is that an arbitrary set
of nodes can be connected into triangles (tetrahedrons) such that no
circumcircle (circumsphere) contains any other node. This is
illustrated in Fig.2; here point D lies inside the circumsphere of
ABC so the Delaunay triangulation is ADB and BCD. A Delaunay
triangulation possesses the properties of uniqueness (subject to
round-off error in the computer implementation) and global
"optimality". The "optimality", in two-dimensions anyway,
consists of the so-called "max-min" property, i.e. the maximum
angle in each triangle is minimised by the connectivity.

Fig.1 A simple CAD-like wireframe-patches-topology
model.

Fig.2 The triangulation of four nodes; D lies inside
the circumsphere of ABC so the Delaunay
triangulation is ADB and BCD.

The most common methodology adopted is to insert points
sequentially into an existing, Delaunay mesh. This insertion creates
new, non-Delaunay cells which must then be reconnected until all
cells are Delaunay once again. This reconnection is highly local,
affecting only cells which contain the new node within their
circumcircles (circumspheres).

In two dimensions two basic methods of re-triangulation are
used - "edge swapping" (Lawson 1977) and "insertion polygon" (for
By contrast, in three dimensions only the insertion polyhedron method (i.e., the three-dimensional equivalent of the insertion polygon) seems to have been used (for example Baker 1987). Recently, some authors (for example Marcum et al 1995) have started to adopt restricted sorts of local edge reconnection to modify mesh quality but these ad hoc approaches tend to lead to locally optimum connections and seem not confer either global optimality or uniqueness. Ferguson (1988) seems to have been the first author to recognize that it is possible to devise an edge-swapping approach in three dimensions which is capable of inserting new nodes in a true, locally Delaunay manner; this local "Delaunayness" then guarantees that the overall mesh be globally Delaunay - i.e., "optimum" and unique.

The Ferguson three-dimensional edge swapping algorithm is patterned after Lawson's approach and is motivated by the same objective - to construct the insertion polyhedron iteratively. The method proceeds as follows. First a new point, A, is inserted into an existing tetrahedralisation. If this node is located within a cell, the cell is divided into four; if located on a face, the two neighbouring cells are divided into six; if on an edge, each cell on that edge is divided into two. Second, each cell, JT, containing the new node is paired with each cell JN, neighbour to the face in JT opposite the new node. If the new node falls within the circumsphere of that neighbour cell then the local connections must be broken and re-made to produce a Delaunay connectivity. Depending on the local orientation, three generic sets of edge swaps can be identified. So, third, the edge joining new node A to to the node E in neighbour cell JN opposite the face ABC shared with cell JT is compared with face ABC as illustrated in Fig.4. If line AE passes through face ABC then edge swaps shown in Fig.4a are performed; if AE intersects AB, BC or CA then edge swaps shown in Fig.4b are performed; if AE passes outside ABC but within the circumcircle of ABC then edge swaps shown in Fig.4c are performed. Fourth, this process is continued for all cells containing new node A until, in effect, the whole insertion polyhedron has been built. The overwhelming advantage of this approach, in comparison with the usual insertion polyhedron method, is the ease with which it can be coded (in finite word length arithmetic) to be robust and the straightforward way it allows boundary integrity to be maintained.

The present approach

The present mesh generation strategy proceeds in three stages: a CAD-like geometry definition stage, triangulation of the surface mesh and tetrahedralisation of the volume mesh. Within each stage the point placement and point-to-point connectivity are viewed as coupled and mutually supporting operations.

Stage 1. Build a CAD-like model of the geometry as described earlier.

Stage 2. In the second stage the domain surface is triangulated, patch by patch, using an edge-swapping Delaunay algorithm and then mapped onto the full 3D shape via the geometry model. This proceeds, in detail, via the following steps (illustrated with the simple tube-in-tube geometry of Fig.1):
Fig. 4 The three basic classes of edge swapping allowed by the Ferguson algorithm.
**Step 1: Boundary Triangulation**

The nodes defining the edge of each surface patch are triangulated by iterative point insertion into a triangulation produced initially by enclosing the entire domain within one large dummy triangle (which will be stripped out at the end) - Fig. 5.

**Step 2: Boundary Recovery**

Boundary integrity must be guaranteed and is enforced in this step, if found necessary, by inserting appropriate new nodes in an iterative manner to repair broken edges.

**Step 3: Boundary Constraint**

Hereafter the integrity of the boundary is maintained by forming a list of triangles inside and outside the domain. This is accomplished by a simple Gaussian surface integral over the domain with respect to dummy sources placed in each triangle in turn (Longley, 1995). From now on the Delaunay triangulation is "constrained" in the sense that edges now known to lie on the domain edge are not permitted to be swapped (a key advantage of an edge-based algorithm).

**Step 4: Protective "Halo" Layer**

Now a protective "halo" of nodes is placed within the domain, on normals from each edge node, and using the Delaunay neighbourhoods of the triangulated boundary segments to establish appropriate edge-normal distances; these halo nodes are added to the triangulation - Fig. 6. The introduction of this concept, tied closely to the background triangulation of the boundary nodes, is a key novelty introduced in the present approach. The point of this halo, and establishing it in this manner, is to provide a zone within which viscous nodes can be added protected from any interference from other nodes (especially those nodes on other portions of the domain boundary in close proximity) and also to handle corners on the domain edge - where the use of these halo nodes guides the mesh towards desirable local isotropy.

**Step 5: Viscous Layers**

Then viscous nodes are grown from the edge nodes, at a specified rate and from a specified initial spacing, along the normals but only within the halo; these viscous nodes are added to the triangulation - Fig. 7.

**Step 6: Interior**

Finally the remainder of the current surface is covered with nodes using the iterative point placement strategy of Holmes et al (1988) whereby nodes are added to the circumcentres of any triangle which fails a quality test and/or a size test. The quality test is based on the ratio of the triangle area to the area of its circumcircle. The size test is based on growing triangles towards a target size which is interpolated out from the boundaries (avoiding the need for the more usual background mesh construction) - Fig. 8.

**Step 7: Smoothing and Mapping**

Each surface is triangulated in local curvilinear space, then smoothed using a simple Laplace filter and then mapped onto the fully three-dimensional geometry via the CAD-like geometry model - Fig. 9.

**Stage 3.** Finally, the interior of the domain is filled with tetrahedrons using essentially the same sequence of operations as described above and again using an edge-swapping Delaunay algorithm. The main differences with respect to the surface meshing algorithm are as follows. The first is that in the "in-filling" stage new nodes are placed at centroids since this is much more controllable in three-dimensions than circumcentre placement. The second, and more interesting, difference in three dimensions is the presence and control of "sliver" cells. A sliver cell is one with vanishing volume formed from four nearly coplanar nodes. No equivalent exists in two dimensions. These sliver cells form perfectly properly within an entirely Delaunay mesh connectivity.
and quite easily since they have a very compact circumsphere. Despite being "optimal" in the sense of Delaunay, they are not helpful for CFD simulation and must be removed. The strategy adopted in the present work was to insert each new node using the Delaunay test to drive the edge swapping until the edge swapping was either completed or stalled. Then the swapping criterion was changed so that the two cells in confrontation, cells JT & JN in Fig.4, swap if either (but not both) are slivers and in such a way as to remove the sliver. This combination of a globally "optimum" criteria, Delaunay, with a local post-processing criteria, designed to remove slivers, has proved very effective.

Fig.9 Step 7: smooth the mesh and map onto the CAD model.

Further examples
The approach was illustrated above with the deceptively simple case of a tube-in-tube geometry. In this section, some more examples are given, starting with a gas turbine blade in cascade. Fig.10 shows the surface mesh; note the highly stretched, viscous mesh near the blade surfaces. In practice it would also be desirable to cluster mesh (adaptively) along the wake.

The second example is that of a compressor cascade with a stagnation pressure probe sited downstream of the trailing edge. Fig.11 shows perspective views of the surface mesh for the geometry. To help illustrate the generation of the volume mesh, Fig.12 shows stages in the development of the mesh in midspan via slices showing the mesh after the addition of (a) the halo nodes, (b) the viscous layer nodes and (c) after completion. The same sequence is repeated in Fig.13 showing zoomed views of a constant span slice near the casing and cutting not only the blade but also the probe itself.

Fig.10 Hub and perspective mesh for a gas turbine blade showing highly stretched elements near the blade viscous surfaces.
Statistics

To conclude the description of the mesh generation, some mesh quality statistics have been extracted from the compressor-plus-probe example. Fig. 14 shows volume-population statistics plotted as population vs. \( \log(\text{VOL}/\text{VOL}_{\text{max}}) \) where \( \text{VOL}_{\text{max}} \) is the maximum volume in the domain. Fig. 15 shows skew-population statistics. Here, skew is defined as the logarithm of the ratio of the smallest cell vertex-to-opposite-face distance divided by the smallest edge length in that cell. This measure is chosen since it can distinguish between genuinely stretched cells and slivers. No attempt has yet been made to optimise the generation procedure (in particular there are many quality and integrity tests and much slow writing to disk of diagnostic messages and statistics). Nevertheless, around 2000 tetrahedrons can be generated per minute on a modest, 90 MHz Pentium workstation, running in 64 bit precision.

FLOW SOLVER

The simulations displayed in this paper were obtained using the author's 3D, unstructured mesh, solution-adaptive Navier-Stokes solver (Dawes 1994) The basic approach need only be summarised here along with some of the more significant recent developments. The equations solved are the fully 3D, unsteady, compressible, Reynolds-averaged, Navier-Stokes equations expressed in strong conservation form. The full stress tensor is retained, including the full viscous energy equation. Turbulence is modelled via k-\( \epsilon \) transport equations (Patel et al 1985) together with appropriate low Reynolds number terms (Lam et al 1981) to handle the approach to the solid surfaces. The application to mixed rotating - stationary
The seven equations of motion are discretized in finite volume form on each tetrahedral control volume of an unstructured mesh. The primary variables are assumed to have a piecewise linear variation over cell faces between the vertices so that the flux sum for a given cell is evaluated to second order accuracy in space. The derivative terms in the viscous stresses are piecewise constant over the cell (since the primary variables are piecewise linear) and are computed by simple application of the Gauss divergence theorem. Using all the cells surrounding each individual node as a control volume then allows the evaluation of the viscous stress terms at that node.

Artificial diffusion is added to control shock capture and solution decoupling. The currently used smoothing operator, however, is rather different in form to that used in earlier publications and can be thought of as a third order implementation (Tatsumi et al 1994) of Denton’s well known first order flux distribution formula (Denton 1983). The basis of this method is to consider the flux associated with edge ‘i’ linking nodes i and j to consist of the arithmetic average of fluxes evaluated using nodal values of the state vector plus a smoothing flux, \( d_i \):

\[
F_i = \frac{1}{2} (F_{i-1} + F_{i+1}) + d_i
\]

In Denton’s approach, changes to the state vector associated with convection are sent downwind and changes associated with the
pressure terms are sent upwind. This effective smoothing flux is equivalent to the edge-wise first difference:

\[ \phi^{(1)} = \text{mag}(u) [\Delta p_A \Delta p_B \Delta p_B^T] \text{sign}(u) [\Delta p_A] \]

This means that the algorithm is essentially first order accurate in the transient, but Denton adds correction factors which effectively remove these smoothing terms once convergence is reached, leading to a second order accurate steady state. By contrast, Tatsumi et al. (1994) suggested using a third difference of this basic first difference smoothing flux as a background smoother but switching it back to the first order form near shock waves using a pressure switch in the conventional way. Accordingly, the smoothing fluxes adopted in the present version of the flow solver are:

\[ d_j = c^{(2)} \phi^{(1)} - c^{(4)} \sum d_j^{(1)} - d_j^{(1)} \]

where \( c^{(2)} = AVA p_j \), \( c^{(4)} = \text{min}(0, 0.5 (AVA - AVA p_j)) \) with \( p_j \) the pressure switch. This is combined with the use of locally edge-wise projected areas which acts to scale the smoothing appropriately on highly anisotropic meshes.

The net flux imbalance into each cell is used to update the flow variables via a four step Runge-Kutta time marching algorithm with residual smoothing (Jameson et al. 1987). Standard fractional step coefficients of 1/4, 1/3, 1/2 and 1 are used, conferring second order time accuracy (fourth order for linear equations). However, a novelty in the present work is that during each fractional step the energy equation is updated first and an intermediate value of pressure, \( p \), inferred using a low Mach number approximation:

\[ \rho E^{k+1} = \rho E^n + \Delta t \sum \rho u \rho_{w0} \cdot A \]
\[ p^* = p^k + (\gamma-1) (\rho E^{k+1} - \rho E^n) \]

This semi-implicit pressure is then used in the momentum equation fractional steps. At the end of the fractional step, static pressure is then re-evaluated from the most recent state vector and re-synchronised. This inexpensive strategy improves the low Mach number performance and convergence rate of the algorithm and permits larger stable time steps.

For calculations with a steady state solution as the objective the algorithm is run with spatially varying time steps (ie. a constant CFL number imposed everywhere) and correspondingly with a spatially constant residual smoothing coefficient selected. Clearly, unsteady simulations must use a spatially constant time step which will invariably be limited by the very fine mesh in the viscous dominated regions near wetted surfaces. To alleviate this limit and make numerical simulation practical, a spatially varying residual smoothing is used, as suggested and validated by Jorgenson et al. (1989), which has the effect that in the inviscid part of the flow no residual smoothing is used and the algorithm reduces to a simple explicit time accurate integration. In the viscous dominated regions near wetted surfaces the allowable time step is increased substantially beyond the local explicit limit; provided the residual smoothing coefficient is not too large and carefully controlled then the formal second order temporal accuracy is not disrupted.

At inflow boundaries the total pressure, total temperature, turbulent kinetic energy and dissipation rate and two flow angles are specified and the derivative of static pressure in the streamwise direction set to zero; at the outflow boundaries the static pressure is specified and the other variables extrapolated from the interior. At present the inflow boundary is not treated in a truly non-reflecting manner; however, the outflow uses a simple one-dimensional non-reflecting treatment. On solid surfaces zero normal fluxes of mass, momentum and energy are imposed. The wall shear stress is computed either from the laminar sublayer or log law equations depending on whether the local value of the wall coordinate \( Y^+ \) is less than or greater than 10 respectively.

**SAMPLE FLOW CALCULATIONS**

**Tube-in-tube**

This first example is flow through the tube-in-tube geometry at a Reynolds number of 40. The cylinder forming the inner tube is treated as a viscous body but the outer tube is set to be an inviscid boundary. Fig. 16 shows predicted velocity vectors in two slices in orthogonal mid-sections of the geometry. The expected, closed, steady, reversed flow zone forms downstream of the cylinder, with an extent in good agreement with earlier predictions (Dawes 1994).
However, the flowfield in fact can be seen to be three dimensional with a pronounced, but steady, cellular structure downstream of the cylinder, transverse to the primary flow direction. This three dimensionality is emphasised by the Mach number distributions shown in Fig. 17. The origin of the three dimensionality is the interaction between the two distinct curvatures in the tube-tube sidewall intersections which generates a pair of streamwise vortices which remain near either sidewalls.

**Compressor cascade with a downstream pressure probe**

The second example is that of a compressor cascade with a stagnation pressure probe sited downstream of the trailing edge. This reflects an often expressed concern over the use of intrusive measurement techniques, particularly in high Mach number flow. Fig. 18 shows predicted Mach number distributions on a constant span slice plane passing through the body of the probe. The probe can be seen to lie in the main blade wake and the new wake associated with the probe itself is also clearly visible. The Reynolds number based on the probe diameter is around 1500 but in the present work, where the intent is to demonstrate a mesh generating procedure, no attempt was made to locally refine the mesh to resolve the probe vortex shedding. To try and assess any impact of the probe on the blade surface, Fig. 19 shows predicted surface pressure distributions on the compressor cascade and endwall (and on the probe itself). In fact, little perturbation to the blade surface pressure distribution is visible, although some perturbation can be seen on the endwall. The “measured” stagnation pressure has not yet been compared with that which exists in the flow in the absence of the probe. Future work will address this and also traverse the probe into the separated zone towards the aft of the suction surface and systematically assess both probe interference and probe accuracy.
CONCLUDING REMARKS

• This paper presents a procedure which can generate highly stretched, viscous meshes for essentially arbitrary three dimensional configurations.
• It is shown that it is feasible to use the Delaunay approach to generate highly anisotropic meshes, provided care is taken to generate the mesh in a constrained sequence with viscous layer nodes suitably protected within a halo derived from an initial triangulation of domain boundaries.
• Employing an edge-swapping implementation of Delaunay in both two and three dimensions produces a fast and very robust algorithm with natural control of boundary integrity.
• Examples are presented for a variety of turbomachinery types of geometry to demonstrate the potency of the new approach; sample flow simulations validate the mesh.
• The artificial viscosity used in CFD and, hence, accuracy would be helped if the node connectivity was more homogeneous (i.e., more similar edge counts associated with each node). Perhaps the basic Delaunay approach might be modified such that connectivity should itself be adaptively determined by the evolving solution; we are working in this area now.
• Related work has been completed aimed at placing nodes on the domain surfaces with some reference to the underlying physical curvature of that surface.

ACKNOWLEDGEMENT

The author would like to thank Conor Fitzsimons for pointing out the existence of Niall Ferguson's work.

REFERENCES

Baker TJ "Three dimensional mesh generation by triangulation of arbitrary point sets" AIAA Paper 87-1124, 1987
Connell SD and Braaten ME "Semistructured mesh generation for three-dimensional Navier-Stokes calculations" AIAA Journal, vol. 33, no. 6, pp 1017-1024, 1995

Longley JP Private communication, 1995