Advanced Unstructured Grid Generation for Complex Aerodynamic Applications

Shahyar Z. Pirzadeh [1]

NASA Langley Research Center, Hampton, Virginia, 23681, USA

A new approach for distribution of grid points on the surface and in the volume has been developed. In addition to the *point* and *line* sources of prior work, the new approach utilizes *surface* and *volume* sources for automatic curvature-based grid sizing and convenient point distribution in the volume. A new exponential growth function produces smoother and more efficient grids and provides superior control over distribution of grid points in the field. All types of sources support anisotropic grid stretching which not only improves the grid economy but also provides more accurate solutions for certain aerodynamic applications. The new approach does not require a three-dimensional background grid as in the previous methods. Instead, it makes use of an efficient *bounding-box* auxiliary medium for storing grid parameters defined by *surface* sources. The new approach is less memory-intensive and more efficient computationally. The grids generated with the new method either eliminate the need for adaptive grid refinement for certain class of problems or provide high quality initial grids that would enhance the performance of many adaptation methods.

Nomenclature

a = source strength

d = radial distance between a target point and the axis of a cylindrical *volume* source

h/l = ratio of vertical distance from aircraft over the body length

I,J,K = index numbers of a target point position in relation to a bounding-box

[1] Senior Research Engineer, Configuration Aerodynamics Branch, NASA Langley Research Center, Mail Stop 499.

ℓ	=	axial distance between a target point and end 1 of a cylindrical *volume* source
L	=	source edge length
n	=	geometric growth exponent
N	=	total number of sources
$\lvert\mathbf{r}\rvert$	=	Euclidean distance between a target point and a source
$\mathrm{R_i}$, $\mathrm{R_o}$	=	inner and outer radii of a cylindrical *volume* source
u, v	=	parametric coordinates
x, y, z	=	Cartesian coordinates
α	=	generic notation representative of grid stretching parameters; also flow incidence
$\overline{\alpha}$	=	weighted average value
ϕ	=	growth function
η	=	hybrid growth exponent
κ	=	maximum principal curvature
θ	=	angular resolution
ξ	=	index defining projection zone of a bounding-box
Δs_0	=	primary grid spacing at the source
ΔS_0	=	secondary (stretched) grid spacing at the source
Δs	=	primary grid spacing at the target point
$\Delta\bar{s}$	=	characteristic length (average mesh spacing)
Γ, Ω	=	geometric rates of growth

Introduction

The unstructured grid methodology is currently at a stage of maturity that allows discretization of complex, three-dimensional (3D), real-world configurations with relative ease and reasonable amount of time and effort. Thanks to many advances by a number of researchers in the science/art of grid generation, this crucial step no longer represents an obstacle for the routine use of Computational Fluid Dynamics (CFD) in the context of large-scale (industrial) applications.

However, despite considerable progress towards facilitating the grid generation process itself, some aspects of the generated grids still lack the desired characteristics for producing accurate CFD solutions and, thus, require further attention. Among the challenges is generation of good quality grids (in a practical and effective manner) for capturing complex flow physics characterized by discontinuities, extensive regions of flow separation behind blunt bodies, and vortices. Suitable grids for solving these complex flow problems require well-resolved, anisotropic grids in the field far away from the geometry.

For many years, solution-adaptive grid techniques have been hailed as the ultimate solution to difficult CFD problems. Although the assertion is plausible, and the potential of adaptive grid methods is indisputable, there are some issues that have limited the capabilities (and thus the success) of these methods in practice. Among the concerns is the adequacy of the initial grid to guide the adaptation process in the "right" direction. In addition, the quality of the refined grids directly affects the viability and accuracy of the final solution. A misconception among many CFD practitioners is that an adapted solution is always a good solution; whereas, experience has shown that an adapted solution can be invalid (converged to an incorrect solution) if it originates from a poor quality initial grid.

One of the crucial aspects of any grid generator is its convenience and ability to control point distribution in the field. While automation of grid spacing is desirable, many are based on resolving surface features[1-3] and are limited in what can be controlled in the field. The flexibility of any method that allows the user to intervene and freely prescribe the desired field-grid characteristics, including multi-directional anisotropic stretching, at arbitrary locations is equally important. Many conventional methods are based on manual description of grid length scales on the model geometry or at the nodes of a secondary mesh commonly referred to as a *background grid*. These methods benefit from a high degree of flexibility but are labor-intensive and may lack the desired accuracy. The success of these methods depends on the experience and attentiveness of the practitioner in tailoring the mesh according to specific requirements of the problem at hand. Obviously, prescription of hundreds of parameters by hand is prone to errors that affect the grid robustness.

The present work is an attempt to mitigate the problem of grid management by maintaining a balance between automation and flexibility for human control. The new approach builds on prior works based on the "source" concept[4] and an automatic curvature-based mesh sizing technique[3] reported earlier. The method exploits the benefits of both while alleviating some of the shortcomings of each. In addition, new capabilities are introduced for better control and automation.

The process of grid distribution with sources, presented in Ref. 4, is analogous to propagation of heat from source elements in a conducting medium. The process results in a smooth and natural dissipation of length scales from the sources throughout the domain. This approach has enjoyed much success and popularity owing to its flexibility as well as the quality of the generated grid. It also provides excellent control for applying multi-directional anisotropy that enhances the grid economy (grid counts) without compromising the effective grid resolution. Although the *point* and *line* sources of Ref. 4 provide a higher level of grid quality and control compared to the conventional means, they still lack the desired automation as well as the control of grid distribution in an extended region in the volume. Manual prescription of a large number of *point* and *line* sources to manage grid distribution in the bulk of a 3D domain is a tedious and time-consuming task and often fails to produce the desired result.

In Ref. 3, an alternative method was developed by which mesh sizing was determined based on the topology of the underlying surface geometry. In this method, the grid length scales are automatically derived from the local surface curvatures and are distributed among the nodes of a recursively refined *octree* background grid enclosing the entire domain. The grid parameters are then interpolated from the background grid during the unstructured grid generation. Although this approach substantially enhanced the automation of grid generation over the method of sources, it still suffered from the lack of control over grid distribution in the volume and also in regions on the surface where the local surface curvature alone could not provide the desired grid resolution. In addition, construction of the *octree* background grid was "expensive" in terms of computational time and storing the entire secondary grid (often with a resolution comparable to the unstructured grid itself) in the computer memory. Furthermore, the technique did not provide the much-needed option of anisotropic grid stretching for reducing grid counts - a vital capability for generating large Navier-Stokes (NS) grids. Although it was shown in Ref. 3 that grids could be generated using the two methods of curvature-based and sources in combination, the quality and control of the grid distribution could not be fully guaranteed because of the different (and inconsistent) techniques employed by the two methods for propagation of length scales in the field.

This paper describes an advanced mesh sizing technique for the generation of high quality unstructured grids for solving complex CFD problems. The new approach is geared towards combining the two techniques described above synergistically in order to exploit the advantages of both while removing the shortcomings of each as much as possible. By combining the two methods, we attempt to keep a balance between automation and controllability for an effective and convenient grid management. The method is implemented in a new version of the unstructured grid generator VGRID[5] developed at the NASA Langley Research Center.

Methodology

The new approach employs source elements as a means for defining and distributing grid length-scales in the computational domain similar to that in Ref. 4. In addition to *point* and *line* sources of the prior work, two more source types, *surface* and *volume*, are introduced in the present approach. Figure 1 shows a schematic of different source types currently implemented in VGRID. These are the simplest geometric shapes (with the exception of arbitrary surface sources) for which various operations such as projection and interpolation can be performed analytically and quickly. Other geometric entities such as curves and arbitrary volumetric shapes can also be implemented with the expense of more extensive computations.

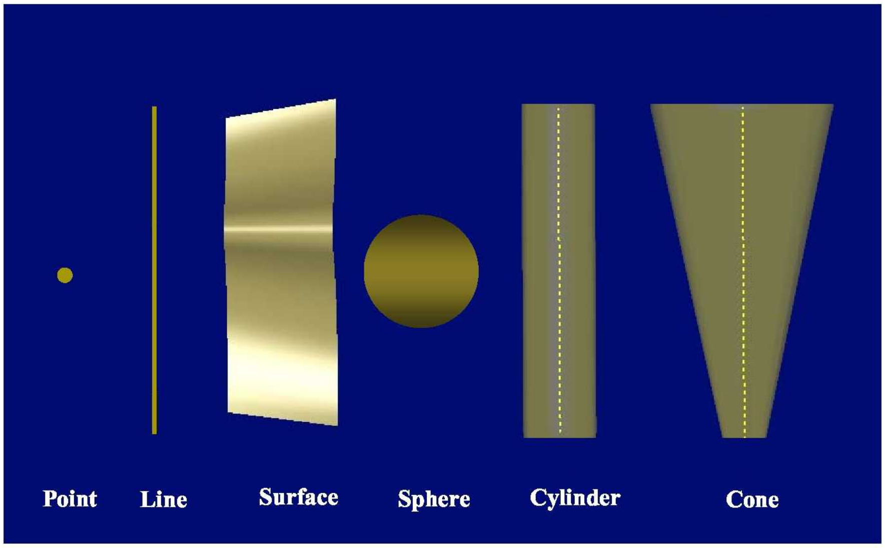


Figure 1. Different types of source elements for defining mesh parameters.

Source Propagation

In the context of prior work in Ref. 4, the *point* and *line* source elements behave as source terms in a governing elliptic partial differential equation for dispersing grid spacing parameters in the field. The solution scheme employed in that work involves interpolation of spacing parameters from *line* sources using the method of line integration. The numerical solution of the resulting Poisson equation, in effect, boils down to a weighted averaging scheme using a uniform Cartesian background grid. The inherent diffusive characteristic of the scheme tends to automatically average out any inconsistencies among sources and provides a smooth variation of mesh length-scales throughout the domain. Although mesh sizing through sources is convenient and flexible, the averaging nature of the scheme employed in Ref. 4 makes the control of mesh distribution difficult. For example, a small change in the grid spacing or source strength at one location may affect the grid resolution in other areas.

Alternatively, a different approach was adopted in Ref. 3 by which the mesh length scales propagate from the sources in the field using the geometric growth function

$$\Delta s_{|\mathbf{r}|} = \Delta s_0 (1+\Gamma)^n \tag{1}$$

where the exponent n corresponds with the shortest Euclidean distance $|\mathbf{r}|$ between a target point in the field and the source. The exponent is determined by equating $|\mathbf{r}|$ to the sum of the first (n-1) terms of the geometric series given by Eq. (1), i.e.,

$$n = \frac{\ln\left(1+\dfrac{|\mathbf{r}|}{\Delta s_0}\Gamma\right)}{\ln(1+\Gamma)}. \tag{2}$$

The typical value for the geometric growth factor Γ is 0.1 that corresponds to a 10 percent rate of growth of the mesh spacing.

A combination of the *point/line* source technique of Ref. 4 and the *growth function* of Ref. 3 has produced mixed success due to incompatibility of the two methods. In the present approach, a modified version of the above growth function is applied to all types of sources uniformly. Two modifications have been implemented to the function in this work. To induce an exponential rate of growth (as opposed to geometric), the exponent n in Eq. (1) is extended by an additional term defined as

$$\phi = 1+\frac{|\mathbf{r}|}{\Delta \bar{s}}\Omega. \tag{3}$$

The exponential growth factor Ω varies between 0.0 and 0.2, and the characteristic length $\Delta \bar{s}$ in Eq. (3) is set to an average value of the mesh spacing extrema.

The second enhancement to the growth function is the implementation of strength (intensity) factors for individual sources similar to those used in Ref. 4. In combination with the global geometric growth rate Γ, the adjustable source strengths provide better control over distribution of length-scales from each source. Finally, the modified growth function used in the present approach is given by

$$\Delta s_{|\mathbf{r}|} = \Delta s_o \left(1 + \frac{\Gamma}{a}\right)^{\eta} \tag{4}$$

where

$$\eta = \left(1 + \frac{|\mathbf{r}|}{\Delta \overline{s}}\Omega\right) \frac{\ln\left(1 + \frac{|\mathbf{r}|\,\Gamma}{\Delta s_o a}\right)}{\ln\left(1 + \frac{\Gamma}{a}\right)}. \tag{5}$$

The nominal value for the source strength (*a*) is 1. A value above 1 makes the source stronger and propagates its spacing attributes farther out in the field. Values smaller than 1 make the source weaker compared to other sources.

Unlike the prior methods that make use of a Cartesian or *octree* background grid to store mesh length-scales in the field, the present approach requires no such auxiliary grids. The primary length-scales are directly evaluated from the sources using Eq. (4), and the value (Δs) at a target point is set to the minimum of mesh length-scales computed for all sources.

For generating anisotropic stretched grids, a secondary or stretched spacing parameter (ΔS_0) and the corresponding stretching direction are also required. The stretching parameters are computed at the target points using a weighted averaging of the values associated with each source, i.e.,

$$\overline{\alpha} = \sum_{i=1}^{N} \frac{\alpha_i}{|\mathbf{r}_i|} \Big/ \sum_{i=1}^{N} \frac{1}{|\mathbf{r}_i|} \tag{6}$$

where $\overline{\alpha}$ represents the stretching parameters (secondary spacing or the direction vector) at the target point, and α_i denotes the corresponding value for the i^{th} source at a distance $|\mathbf{r}_i|>0$ from the point.

The new growth function provides more efficient and smoother distribution of length-scales in the field as well as flexibility for better control of grids using individual source strengths and the global growth factors. Unlike the method in Ref. 4, no *ad hoc* adjustment of source strengths is required, and the method provides the precise grid spacing that the user prescribes at the source location. The new universal method guarantees the uniformity of the length scales computed from different source types as opposed to the inconsistency of the hybrid approach taken in Ref. 3.

Point/Line Sources

The *point* and *line* sources are the simplest geometric entities. Grid parameters (primary and secondary length-scales as well as the stretching direction) are prescribed and stored at the *point* or the two ends of a *line* source. The prescribed mesh parameters vary linearly along a *line* source between the two ends, and the stretching direction is defined along the source. The shortest distance $|\mathbf{r}|$ is calculated by a simple projection of the target point onto the source in 3D (Fig. 2), and the mesh parameters are evaluated using Eqs. (4) and (6). The *point* and *line* sources are useful for controlling mesh characteristics in small regions locally such as corners and edges of a geometry.

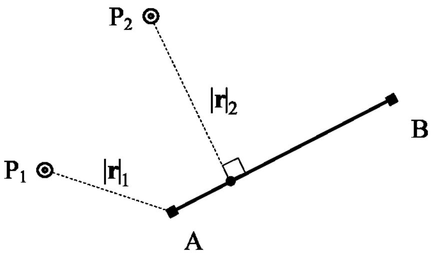


Figure 2. Projection of target points $\mathbf{P}_i$ to *line* source $\mathbf{L}_{\mathbf{A\text{-}B}}$.

Surface Sources

Surface sources are defined in terms of *Non-Uniform Rational B-Spline* (NURBS) representations and are usually made of the same underlying surfaces that define the geometry for surface mesh generation. However, arbitrary surfaces can also be defined in any form and shape and used as sources at any location in the field. To define grid parameters on the sources, the surfaces are first discretized using a simple 2D structured mesh (Fig. 3), and the grid parameters are distributed and stored at the nodes of this mesh. The shortest distance $|\mathbf{r}|$ is determined by projection of the target point onto the surface using the NURBS library of Ref. 6, and the mesh parameters are evaluated using Eqs. (4) and (6).

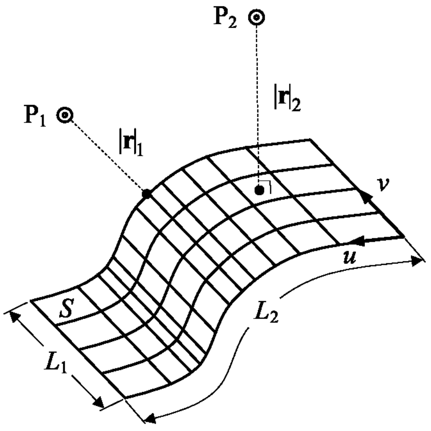


Figure 3. Projection of target points $\mathbf{P}_i$ to the *surface* source *S*.

The grid parameters on the surface of a source are determined in this work based on:

a) Local curvature of the surface source (finer grid resolution at high-curvature areas and vice versa).

b) User-prescribed values.

c) Geometric limiting factors such as minimum edge length (L) of the source.

A blend of these options can also be used to produce the desired grid resolution. For example, if the option (a) alone does not adequately resolve the grid at a location of low curvature, other options can be activated in combination.

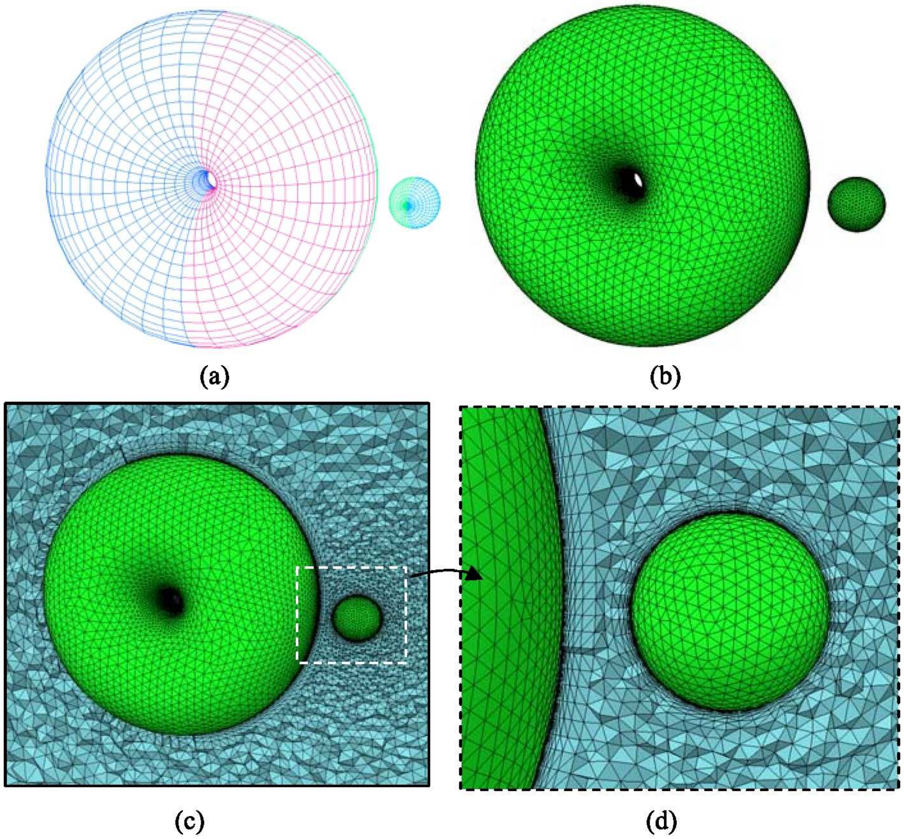
For the curvature option, the primary mesh length-scale is automatically computed on the surface using the following relationship between the local length-scale (Δs_0), the corresponding maximum principal curvature (κ) computed locally, and a prescribed angular resolution (θ)

$$\Delta s_o = 2 sin\left(\frac{\theta}{2}\right) \Big/ \kappa \tag{7}$$

The angular resolution is independent of the model scale, and a nominal value of 15 degrees works well for most applications. Lower angles result in finer grids and vice versa. The maximum principal curvature is computed at a point on the surface using the NURBS library of Ref. 6. After the primary length-scales are computed on all nodes of the 2D mesh, the values are smoothed out throughout the surface by propagating the minimum values in the u and v directions using a growth function similar to Eq. (1). A separate growth factor (Γ_s) controls the rate of growth of the grid resolution on the surface from regions of high curvature (finer grid) to other areas.

In the second option (b), the user prescribes the mesh parameters at the four corners of a *surface* source. The quantities in the interior of the surface are computed using bi-linear interpolation of the values at the corners. This option is useful for resolving the grid on a flat surface or where the surface curvature is low. In the last option (c), grid length-scales are automatically determined based on the minimum edge length of a *surface* source such that a predetermined number of grid elements fit across the source. This option is useful for resolving the grid on narrow surfaces such as the blunt trailing edge of a wing.

The *surface* sources, especially with option (a), provide excellent grid quality and distribution of points on the surface with minimum amount of manual work. Figure 4 shows an example grid in which the grid resolution is automatically computed based on the surface curvature using six *surface* sources. As indicated, more points are added at locations of high curvature such as the small sphere as well as the hole in the middle of the torus and vice versa. Also, note that the close proximity of the smaller object has affected the grid resolution on the larger object with a lower curvature. As a result, grid is finer in the gap and on the torus with a smooth transition both on the surface and in the volume. In this example, only the underlying NURBS surfaces defining the geometry are used as sources. No additional information other than the growth rates and the minimum/maximum allowable grid size is required.



(a)

(b)

(c)

(d)

Figure 4. Sample grid generated using six *surface* sources with automatic curvature-based grid sizing: (a) sources defined in terms of NURBS, (b) surface mesh, (c) surface/volume mesh, and (d) close up view of surface and volume grid between the two objects.

An important feature of the surface source concept in the present work is its capability of supporting anisotropic grid stretching. The method employed in Ref. 3 lacked this feature, and grid stretching was accomplished by introducing additional *line* sources in combination with the curvature-based grid sizing. In this work, stretching is implemented by prescribing a secondary length-scale (ΔS_0) in the parametric *u* or *v* direction and computing the stretching parameters in the field using Eq. (6). The anisotropic grid stretching is extremely useful for generating efficient Reynolds-Averaged Navier-Stokes (RANS) grids on complex configurations. The feature reduces the number of grid elements substantially without compromising the effective resolution of the grid in the essential directions. Figure 5 illustrates a sample grid generated using a *surface* source with automatic computation of the primary grid spacing. The lower NURBS surface with a ripple acts not only as an underlying surface definition but also as a source for computing the primary length scales. As indicated, the grid is automatically refined in the *u* direction to resolve the curvature of the ripple but not in the other direction resulting in high-aspect-ratio triangles.

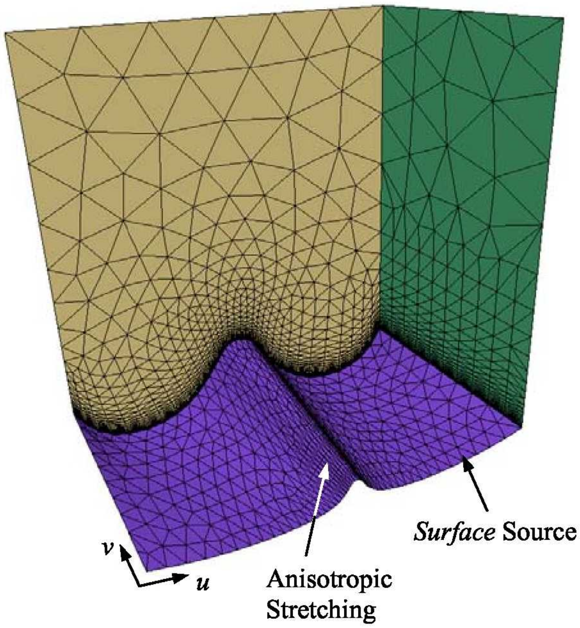


Figure 5. Sample RANS grid with automatic curvature-based grid sizing and anisotropic stretching generated using a *surface* source.

Another difference between the present curvature-based approach and that reported in the previous work^3 is in the mechanism that transmits the grid length-scales from the surface into the field. The method in Ref. 3 relies on an *octree* background grid for this purpose. Although the *octree* mesh provides an excellent tool for the distribution of length-scales throughout the field, it suffers from an efficiency drawback that limits the applicability of the method for generating very large grids. The construction of an octree was shown in Ref. 3 to be relatively fast thanks to implementation of some efficient algorithms and data structures. However, the computer memory requirement for storing data on a large background grid poses an obstacle to the routine application of the method for solving real world problems. The requirement is especially restrictive for complex configurations with extensive variations in the surface curvature resulting in a very fine grid resolution. The size of the background grid for such applications can become comparable to that of the final unstructured grid. The generation of a massive grid with this method would require a large computer to store both grids in the memory. In the present method, no 3D background grid is needed for transmitting data from surface into the volume; therefore, the memory requirement is much smaller.

While the elimination of background grid improves the memory problem, it obviously reduces the speed of the computation. Direct query of surface sources for every mesh point in the field requires projection of the point onto the underlying CAD surfaces. The projection process involves extensive numerical search operations which are highly CPU-intensive. To reduce both the memory requirement and the computational costs, a new approach has been implemented in the present work as described in the following section.

Bounding-Box Approach

The idea behind this approach is to map the grid information stored on a *surface* source to an intermediate medium defined by a simple geometric entity (such as a Cartesian box or any other analytical surface) that encloses the source completely (Fig. 6a). The surface of this simple shape acts as a new auxiliary *surface* source for which the projection can be performed analytically and, thus, considerably faster. The new technique, in effect, reduces the problem of a large 3D background grid to a number of much smaller 2D problems.

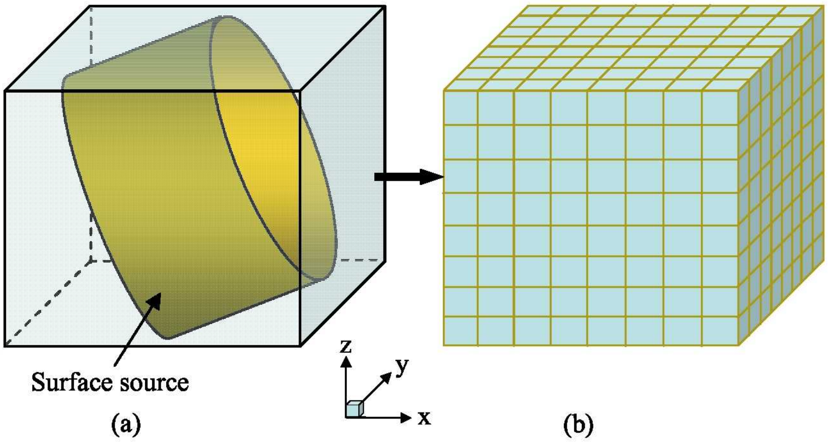


Figure 6. Bounding-box auxiliary source: (a) *surface* source and its bounding box, (b) box faces discretized with two-dimensional Cartesian grids.

The following steps describe the process of query of *surface* sources using the bounding-box concept:

1) Define a Cartesian bounding-box for each *surface* source based on the min/max dimensions of the source (x_{min}, x_{max}, etc.)

2) Discretize the six faces of each bounding box using a uniform 2D Cartesian grid (Fig. 6b). The present approach uses a 20X20 discretization on each face.

3) Extrapolate spacing parameters from each source to its corresponding bounding-box mesh using the growth function given by Eqs. (4) and (6).

4) During the unstructured grid generation, evaluate the spacing parameters at a target point from:

a) a *surface* source itself if the point is inside the corresponding bounding box, and/or

b) the faces of all bounding-boxes that exclude the target point.

The query process is considerably faster with this approach because, regardless of the number of *surface* sources, the expensive CAD projection is always limited to, at most, one source when the point is inside the corresponding box (assuming there is no overlap between the bounding boxes). Query of an auxiliary (box) source is greatly simplified, as the critical part of the process is reduced to a simple analytic projection of the target point to a readily identifiable segment of the Cartesian box.

A 3D space that contains a box is divided into twenty seven projection zones around and inside the box. The projection zone in which a target point P (with coordinates x,y,z) resides is determined by the relation

$$\xi = I + J + K \tag{8}$$

Where

$$I = \begin{cases} 100, & \text{for} \quad x \le x_{min} \\ 200, & \text{for} \quad x_{min} < x < x_{max} \\ 300, & \text{for} \quad x \ge x_{max} \end{cases} \qquad J = \begin{cases} 10, & \text{for} \quad y \le y_{min} \\ 20, & \text{for} \quad y_{min} < y < y_{max} \\ 30, & \text{for} \quad \text{ŹŹŹŹŹŹŹ}\, y_{max}y \end{cases} \qquad K = \begin{cases} 1, & \text{for} \quad \text{ŹŹŹŹŹŹŹŹŹ}_{min} \\ 2, & \text{for} \quad z_{min} < z < z_{max}, \\ 3, & \text{for} \quad \text{ŹŹŹŹŹŹŹŹŹ}_{max} \end{cases} \tag{9}$$

Equation (8) gives a unique index for each of the twenty-seven projection zones. Once the zone ξ of a target point is identified, the point is simply projected to a corner (C), edge (E), or face (F) of the bounding box according to the format outlined in Table 1 and the notations defined in Fig. 7.

Table1. Twenty Seven Projection Zones and Corresponding Elements of a Bounding-Box

Zone ξ	Projects to	Zone ξ	Projects to
111	C1	223	F6
112	E9	231	E4
113	C5	232	F4
121	E1	233	E8
122	F1	311	C2
123	E5	312	E10
131	C3	313	C6
132	E11	321	E2
133	C7	322	F2
211	E3	323	E6
212	F3	331	C4
213	E5	332	E12
221	F5	333	C8
222	Source		

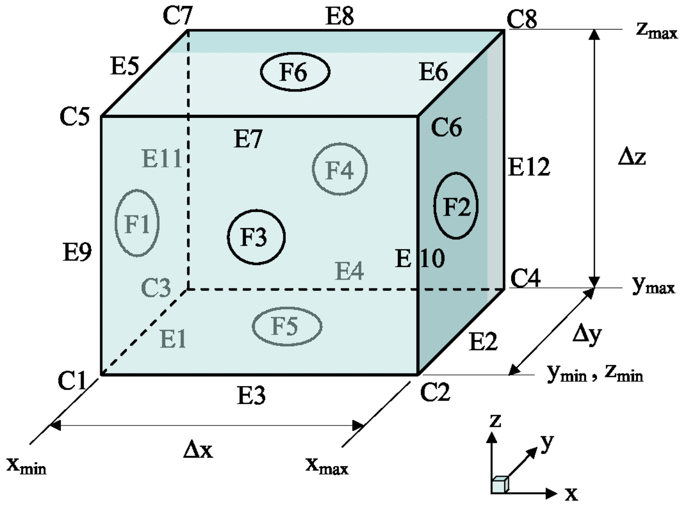


Figure 7. Corner, edge, and face labels of a bounding-box.

The procedure described above is extremely fast, as it requires no time-consuming search operations. Note that only a small fraction of the points in a volume grid are adjacent to the geometry and, thus, the majority falls outside of all *surface* source bounding-boxes. Therefore, the expensive CAD projection (even for one source at a time) is completely eliminated for most part of the grid generation. The method provides excellent grid quality with a fraction of computational time (compared to the direct query of sources) and with substantially less memory requirement in comparison to methods utilizing a 3D background grid.

Volume Sources

Although *point*, *line*, and *surface* sources offer great versatility for defining mesh resolutions on the surface, they are not as much effective for controlling grid density in an extended region in the volume away from the geometry. Such grid requirements often arise from applications involving, for example, massively separated flows in the wake of a bluff body where a uniformly refined mesh is a required. Earlier efforts to generate suitable grids for such applications with conventional sources required manual insertion/description of hundreds of line or even surface sources without much success. Alternatively, adaptive methods attempt to refine the mesh in the field by point insertion and/or grid movement. An issue with this type of refinement is grid distortion as the mesh density is increased through local post-processing operations rather than by the grid generator itself in a natural way. Furthermore, the lack of adequate grid resolution in the initial mesh may result in refinements in the wrong regions of the grid or may produce no sufficient refinement at all.

Volume sources provide excellent control over grid distribution anywhere in the field. With these sources, the prescribed grid resolution remains constant (or varies as specified by the user) inside the source and grows smoothly outside using Eqs. (4) and (6). Figure 8 illustrates an application of volume sources for resolving the grid in the wake of a circular cylinder that is positioned in the flow field with its axis normal to the free stream. This classic problem of real fluid flow is characterized by extensive flow separation behind the cylinder resulting in a complex 3D pulsating vortex structure shed into the wake. Experience has shown that accurate numerical simulation of this geometrically simple but computationally challenging flow problem requires precise specification of the grid characteristics in the wake region. This problem serves as a good example to signify the benefits of *volume* sources in producing the exact grid properties that the user desires to enforce anywhere in the field.

Volume sources are defined in terms of simple geometric shapes such as spheres, right circular cylinders, or cones with inner radius R_i and outer radius R_o. A *volume* source can be solid (R_i=0) where the grid spacing inside the source remains constant or varies linearly from one end of cylinder/cone to the other (like those shown in Fig. 8). For a hollow source (R_i>0), grid spacing remains constant between the two radii and grows both inside and outside

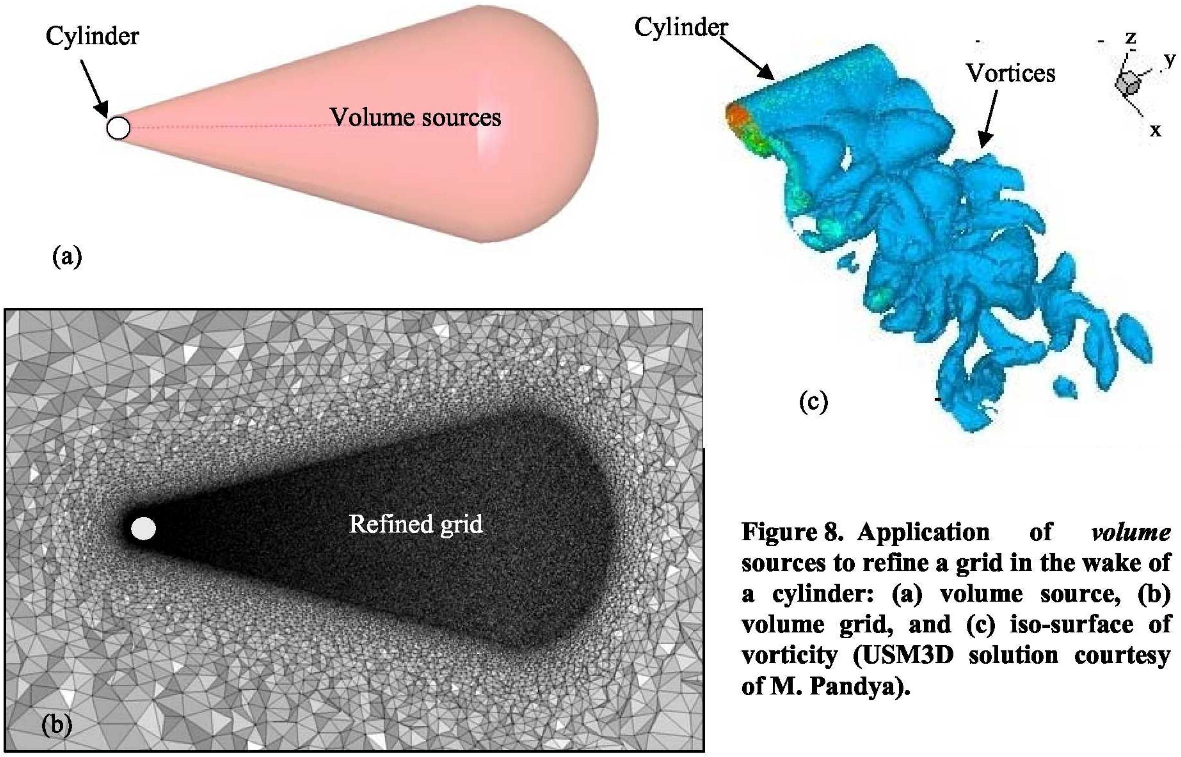


Figure 8. Application of *volume* sources to refine a grid in the wake of a cylinder: (a) volume source, (b) volume grid, and (c) iso-surface of vorticity (USM3D solution courtesy of M. Pandya).

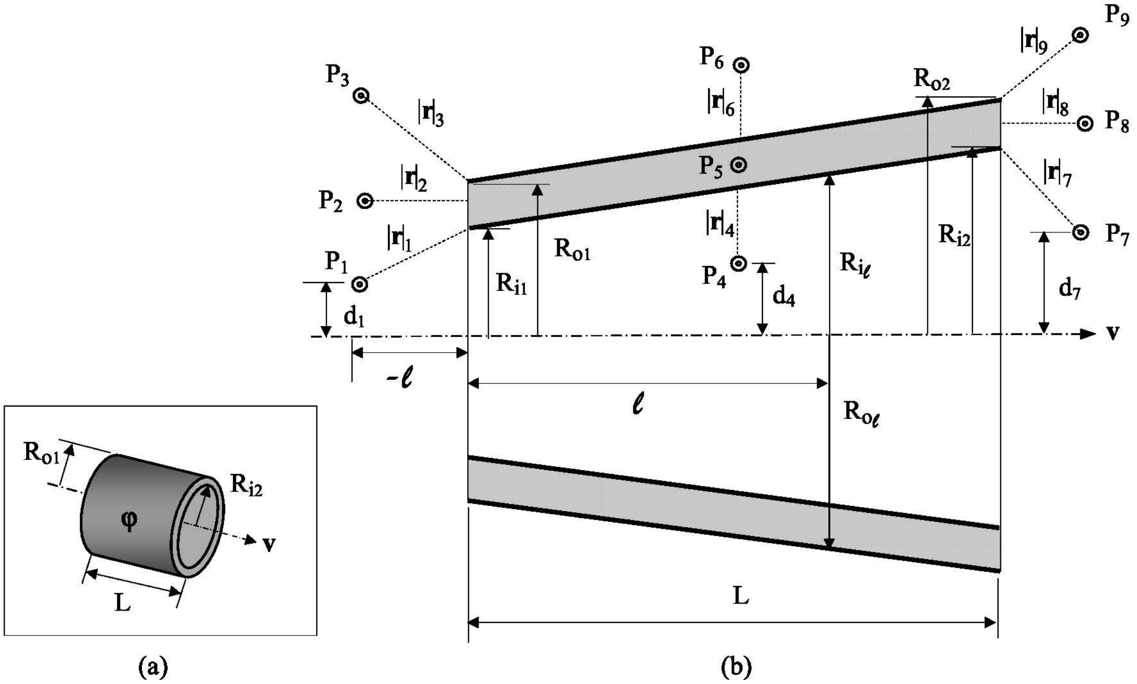


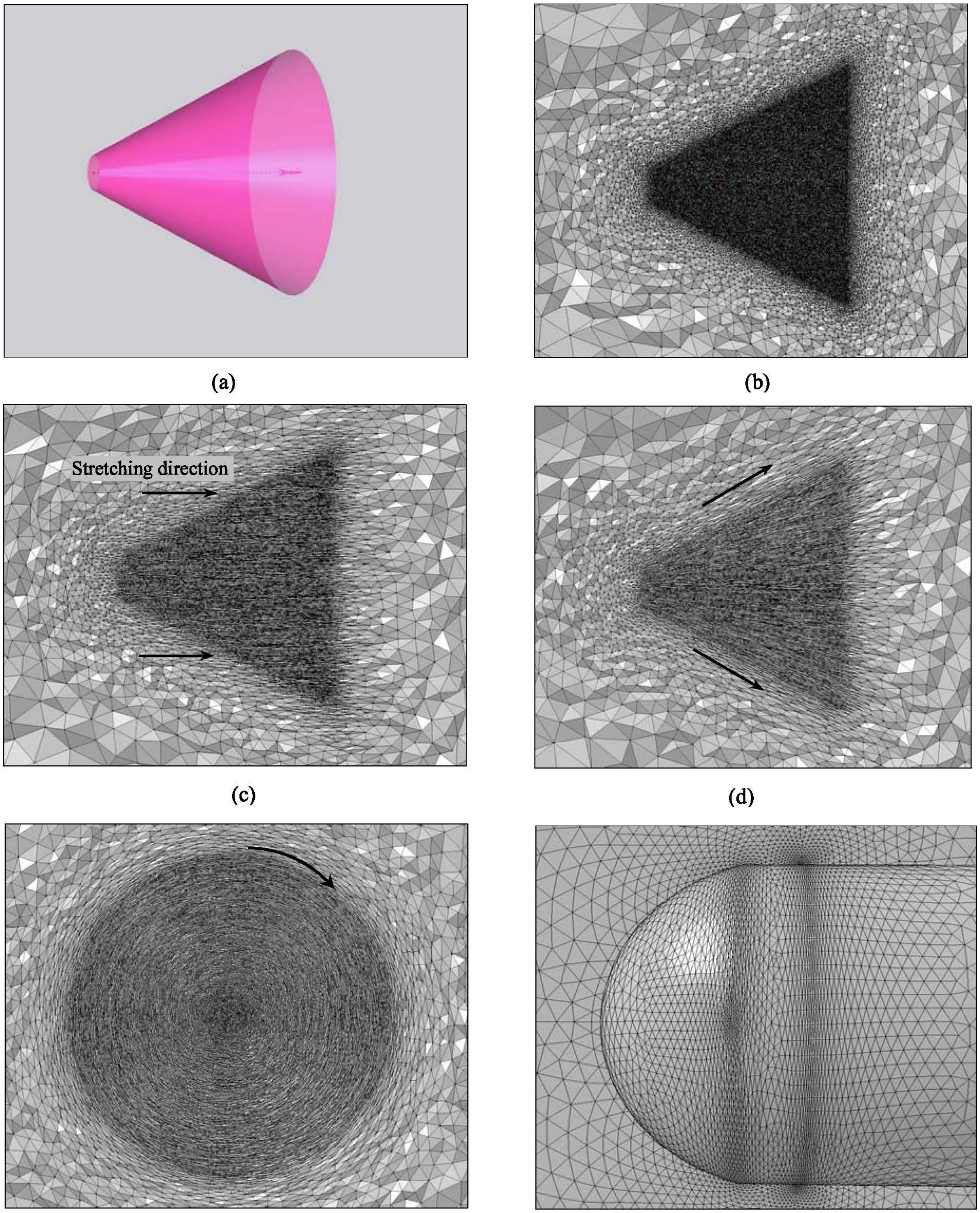
Figure 9. Projection of target points P_i in nine different projection zones of *volume* source φ: (a) overall view of a *volume* source in 3D, (b) cross-section of a conical source.

of the source using the same growth functions applied to other source types. Note that a *volume* source degenerates to a *surface* source for R_i set to R_o. The projection of a target point in the field to a *volume* source is simple and efficient. As illustrated in Fig. 9, a point in 3D space lies in one of the nine projection zones of a hollow cylinder. The minimum distance from a target point to the source can be determined analytically regardless of orientation of the source axis (**v**) in space. For a target point positioned anywhere inside the source (point P_5), the minimum distance $|\mathbf{r}|$ is equal to zero, and the primary length-scale in Eq. (4) reduces to the prescribed value for the source (Δs_0).

Anisotropic grid stretching has also been implemented for volume sources resulting in an even larger effect on reducing the number of grid elements in nonessential directions. Grids can be stretched with *volume* sources in the axial, tangential to the wall, or circumferential direction. Figure 10 illustrates examples of grid stretching using volume sources with different stretching directions. The grids in Figs. 10(b)-10(e) have been generated using a conical source, whereas the one in Fig. 10(f) has been generated with two narrow cylindrical sources to produce circumferential stretching on the surface and in the field. Note that the circumferential stretching is at its maximum level at the outer radius of the source and gradually reduces in magnitude to zero at the center as shown in Fig. 10(e). Although the extent of linear stretching (axial/tangential) is theoretically unlimited with a *volume* source, the amount of circumferential stretching is constrained by the source radius. The upper bound of the secondary (stretched) length-scale for circumferential stretching is a function of the primary spacing and the outer radius of the source given by

$$\Delta S_o \le \sqrt{\Delta s_o\left(\mathrm{R}_\mathrm{o} - \frac{1}{4}\Delta s_o\right)}. \tag{10}$$

Anisotropic stretching is a powerful tool for generating efficient grids for large problems. The capability is particularly important for generating RANS grids on complex configurations for which the size of a mesh can become prohibitively large. Anisotropic stretching reduces the number of grid elements in directions of low flow gradients without affecting the grid resolution in other essential directions. As an example of the power of grid stretching in reducing the grid size, the isotropic mesh shown in Fig. 10 (b) has over 13.5 million cells, whereas the stretched grid in Fig. 10(d) with the same primary spacing contains only 1.6 million cells. Grid stretching is also essential for producing accurate solutions for certain class of flow problems as will be shown in another example later in the paper.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 10. Examples of anisotropic grid stretching in different directions with *volume* sources: (a) a conical source, (b) cross-section of volume grid with no stretching, (c) volume grid with axial stretching, (d) volume grid with tangential stretching, (e) volume grid cut with circumferential stretching, and (f) surface grid with circumferential stretching using two narrow cylindrical sources.

Sample Results

Two sample applications are presented in this section to demonstrate the new grid generation capabilities for computing real-world, complex, aerodynamic problems. The first application features geometric complexities posing difficult grid generation issues, and the second example represents a challenging CFD problem requiring special grid properties for producing accurate results. In addition, an unconventional (non-aerodynamic) example grid is presented to demonstrate the power of curvature-based sourcing for automatic grid point distribution.

Apollo Launch Abort Vehicle (LAV)

The vehicle configuration consists of a front cylindrical body with a nose cone, four abort motors, a structural truss system, and a crew module as shown in Figure 11. The geometry features complexities such as truss beams in very close proximity and tight angles at the end junctures that make generation of RANS grids extremely difficult even with unstructured grid methodology. Precise control of grid spacing at these difficult locations is crucial for growing thin-layered viscous grids of good quality without introducing complications during the generation process. In addition, a uniformly fine grid is required to resolve the flow details around the entire truss system, center body, and the jet exhaust of the abort engines. Similar to the cylinder example shown earlier, the blunt base of the crew module creates a large flow separation region. Proper resolution of the grid in the wake is essential for predicting the aerodynamic forces and moments accurately.

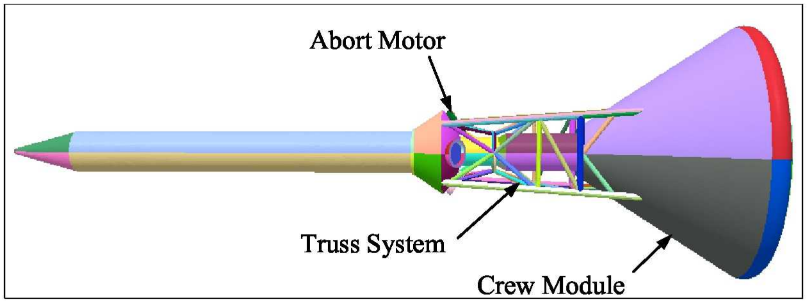


Figure 11. CAD model definition for the Apollo Launch Abort Vehicle (LAV).

Conventional sourcing of such a complex configuration would require hundreds of *line*, *point*, or even *surface* sources to define proper grid spacing in the critical regions. With the current approach, only nine *volume* sources are prescribed and easily inserted around the entire geometry as illustrated in Figure 12. Seven sources are used to define the spacing functions on the geometry itself, including only one to cover the entire complex area around the truss system (highlighted in the magenta color), and two to refine the grid in the wake region. Anisotropic stretching

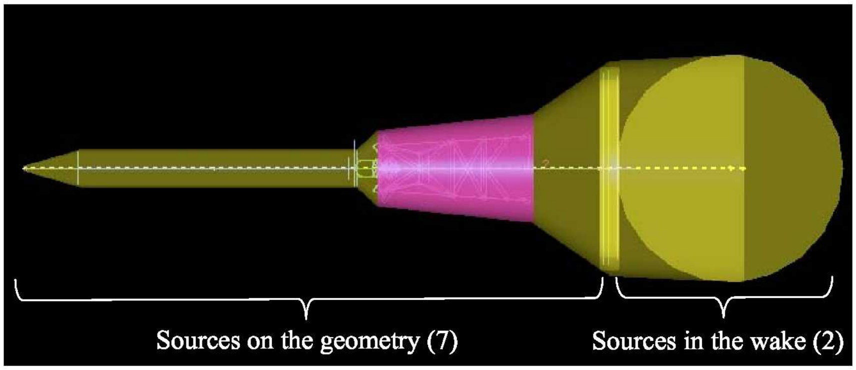
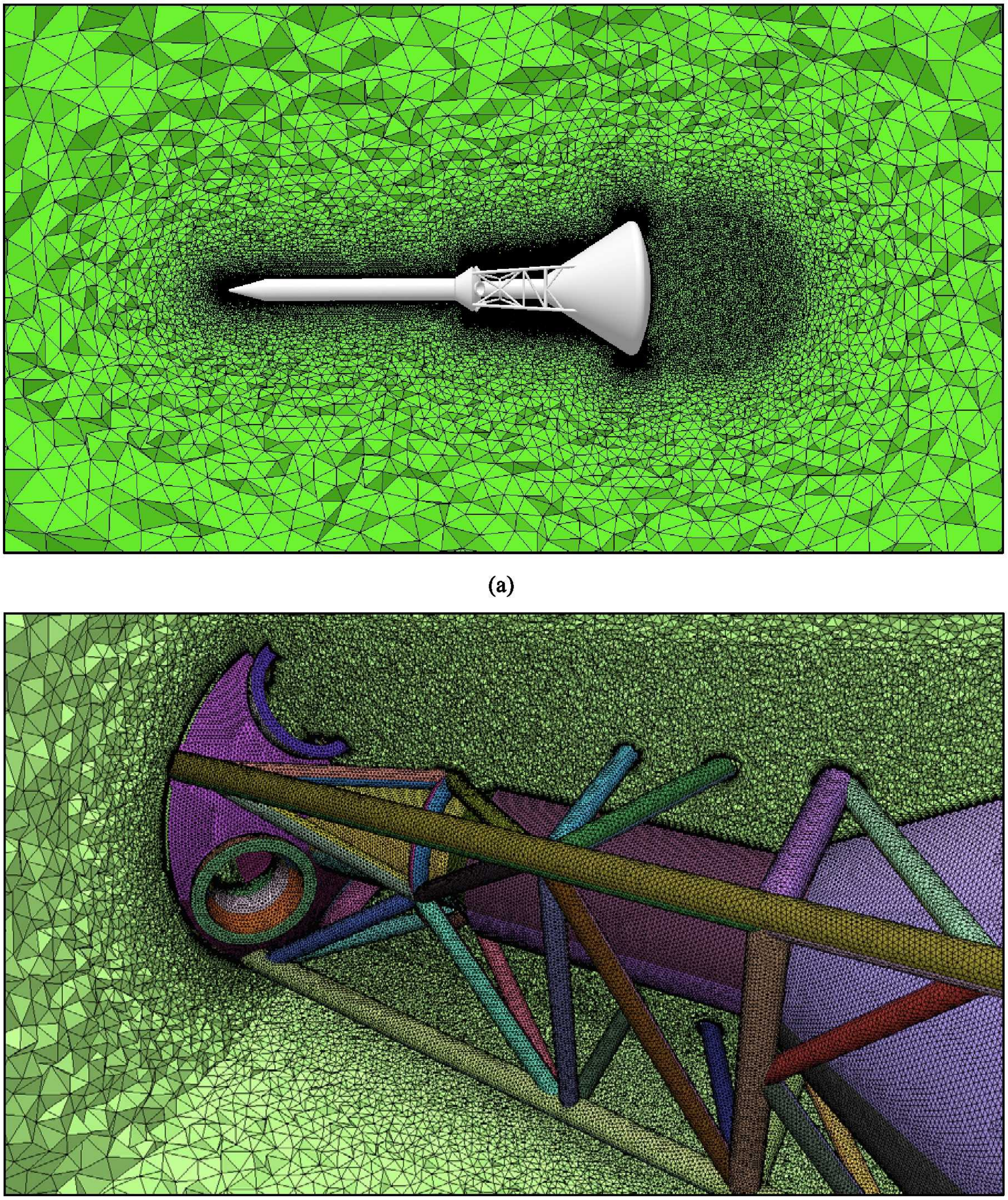


Figure 12. Application of *volume* sources for the LAV configuration.

is applied along the front cylindrical section (in the axial direction) and around the back rim of the crew module (in the circumferential direction) to reduce the grid counts.

The generated RANS grid is illustrated in Figure 13. As evident, the grid is adequately resolved in the areas of interest with a smooth distribution of points both on the surface and in the volume. The efficiency and convenience that the *volume* sources provide are unparalleled for controlling grid distribution in difficult areas such as that shown in Fig. 13(b) as compared to the conventional methods. This grid contains 8,216,471 nodes and 48,355,970 tetrahedrons. A breakdown of the grid counts and the generation CPU time is presented in Table 2. A MacBook Pro laptop with 2Ghz Intel Core Duo processor and 2GB of RAM was used to generate this grid.

Extensive RANS computations of the LAV configuration have been performed at the NASA Langley Research Center as part of the NASA Crew Exploration Vehicle (CEV) Aerosciences Project. Although the focus of this paper is on the grid generation aspect of the ongoing work, a sample flow result is presented in Figure 14 for completeness and to demonstrate the viability of the generated grids for producing good quality solutions. These solutions have been generated using the unstructured grid solver USM3D[7] on a number of different grids generated with the present method.



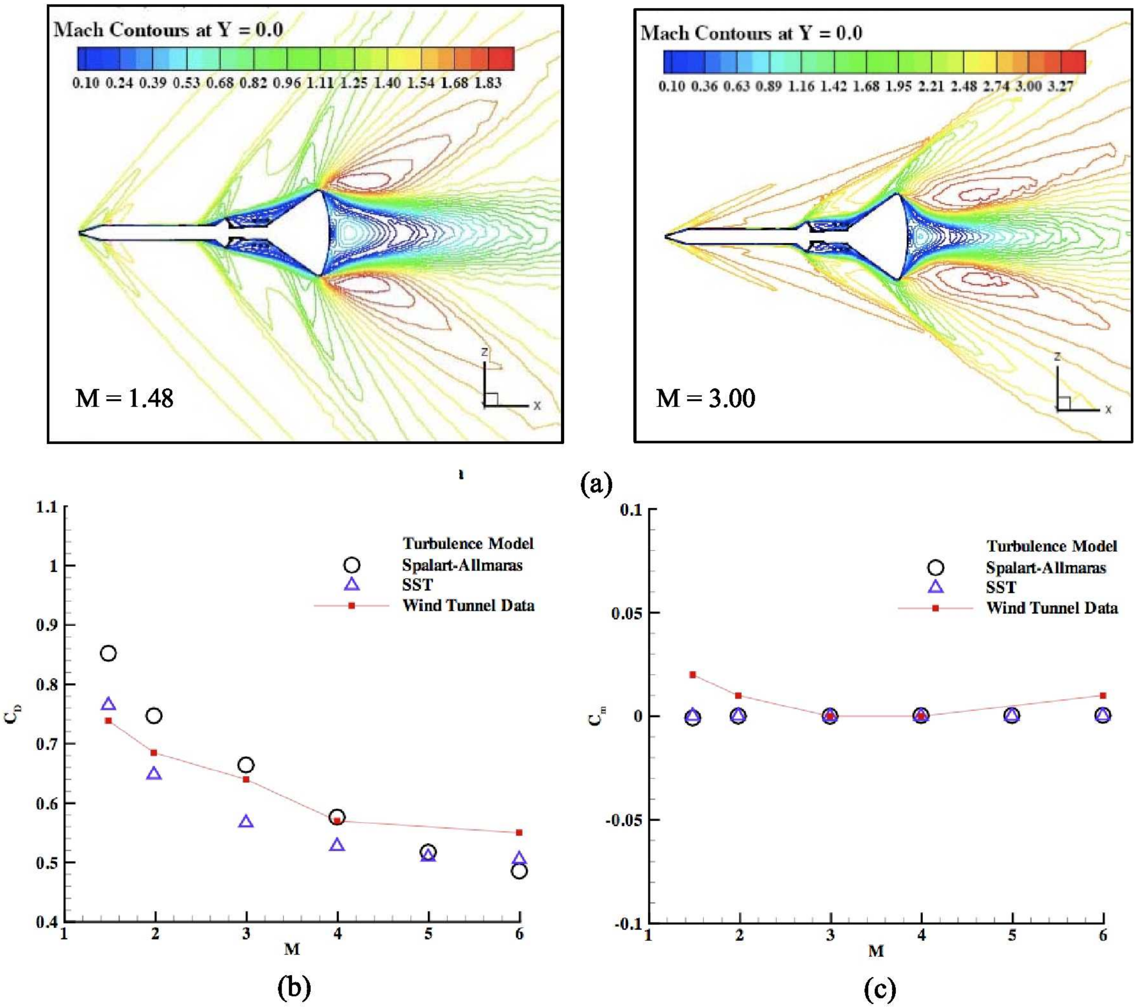
(a)

(b)

Figure 13. Tetrahedral grid on the Apollo LAV configuration: (a) geometry and the volume grid, (b) close-up view of surface and volume grid cuts around the truss system.

Table 2. Grid Statistics for the LAV Test Case

Grid Segments	Number of Nodes	Number of Triangles	Number of Tetrahedra	Generation Time (m)
Surface Grid	190,032	380,216	----	14.0
Volume Grid (viscous part)	4,216,778	----	25,279,941	8.5
Volume Grid (inviscid part)	3,809,661	----	23,076,029	65.5
Total	8,216,471	380,216	48,355,970	88.0



(a)

(b)

(c)

Figure 14. Comparison of CFD and wind tunnel data for the Apollo LAV Configuration at α = 0°: (a) sample Mach contours, (b) drag coefficient, (c) pitching moment coefficient. (USM3D solutions courtesy of Neal T. Frink and Naomi McMillin.)

Sonic Boom Prediction

One of the challenges facing CFD application is simulation of the sonic boom problem. Aircraft flying at supersonic speeds disseminate pressure waves that extend a long distance in the flow field. Accurate prediction of the wave strengths far away from the model poses a challenge for the computational simulation of the problem. Among the difficulties is generation of suitable grids, particularly in 3D. Typical CFD grids usually grow off the surface too early causing the numerical solution to dissipate and lose the wave signals within a short distance from the geometry. For a successful simulation, a mesh should adequately resolve the waves several body lengths away from the model. The control of grid resolution far in the field is generally a challenge, especially with unstructured grids.

There is a large volume of reports in the literature addressing the problem of sonic boom computation. Among them, many rely on adaptive grid refinement techniques. While adaptive methods provide a powerful tool for improving the accuracy and efficiency of CFD solutions, their level of success usually hinges on the adequacy of the initial grid upon which the adaptation process embarks. The resolution of typical initial grids is often insufficient around the geometry or quickly grows in the field within a short distance from the surface. As a result, the initial solutions obtained on these coarse grids often miss the essential flow features and cause the subsequent adapted grids/solutions fail to converge to the desired level of accuracy.

Experience has shown that in addition to proper mesh density, other grid properties such as anisotropy and orientation of cell faces in relation to the wave angles can also play an important role in the accuracy of the predicted wave strengths. Previous attempts by the author and other investigators to resolve the waves by refining isotropic unstructured grids have produced limited success for solving the sonic boom problem. While the importance of grid alignment and shock-fitting techniques for resolving the pressure waves has been known (and exploited using structured grids) for many years, their implementation for unstructured grids has been a problem in itself.

In the present work, the new *volume* sources have been employed to produce a better control over grid distribution in an extended region in the field. Furthermore, the stretching feature of *volume* sources has provided the ability to produce very high-aspect-ratio, anisotropic grids that accommodate better alignment of stretched tetrahedral cells with the wave angles.

To demonstrate the utility of the new technology for computing sonic boom problems, a generic wing-body configuration referred to as the Segmented Leading Edge (SLE) model has been employed. The model (shown in Figure 15) has been tested in the NASA Langley Unitary Plan Supersonic Wind Tunnel (UPWT) as well as the NASA Glenn 10×10 Wind Tunnel.

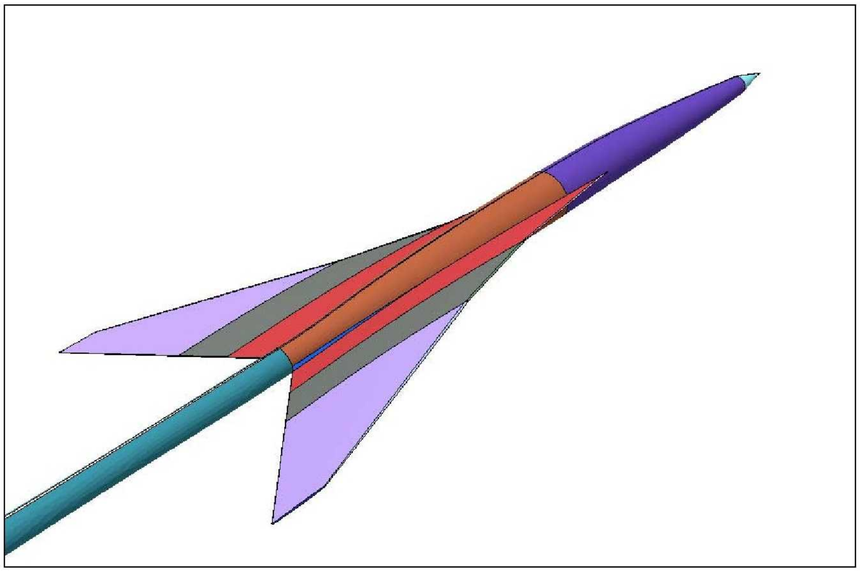


Figure 15. CAD model definition for the Segmented Leading Edge (SLE) configuration.

Figure 16 illustrates *volume* sources used to prescribe grid resolution around the aircraft and in a region beneath the geometry extended 10 body lengths downward. Note that for this study, prediction of wave signatures only below the aircraft is of interest. The cylindrical sources are positioned at an angle of 30 degrees with respect to the freestream direction (Mach angle corresponding to a freestream Mach number of 2). The radii of these sources are large enough to cover the entire profile of the aircraft and to resolve the pressure waves emanating from different sections of the aircraft downward. The grid on the surface of geometry is mostly isotropic except at the wing leading edge. However, high level of grid stretching with a maximum cell aspect ratio of about 80 is prescribed for the *volume* sources to stretch the tetrahedral cells along the waves. Such stretching ratios are well within the acceptable limits of most tetrahedral-based Navier-Stokes flow solvers. In general, VGRID insures grid quality by regenerating any inviscid cell having a negative volume or a minimum included angle between edges of the tetrahedra less than a user-prescribed value, which is typically 0.01 to 0.03 degrees.

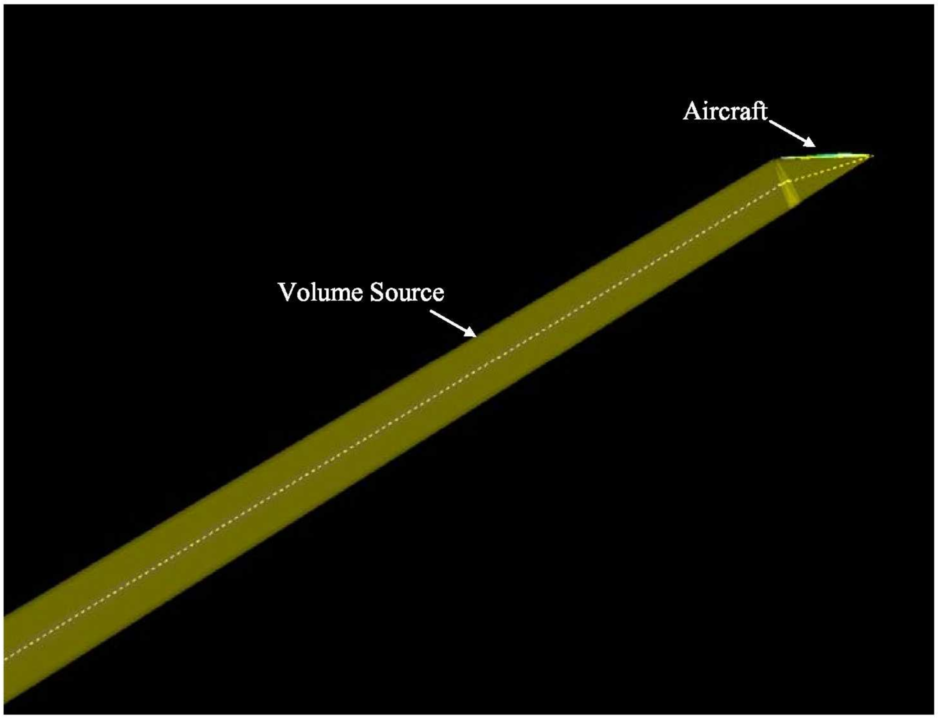
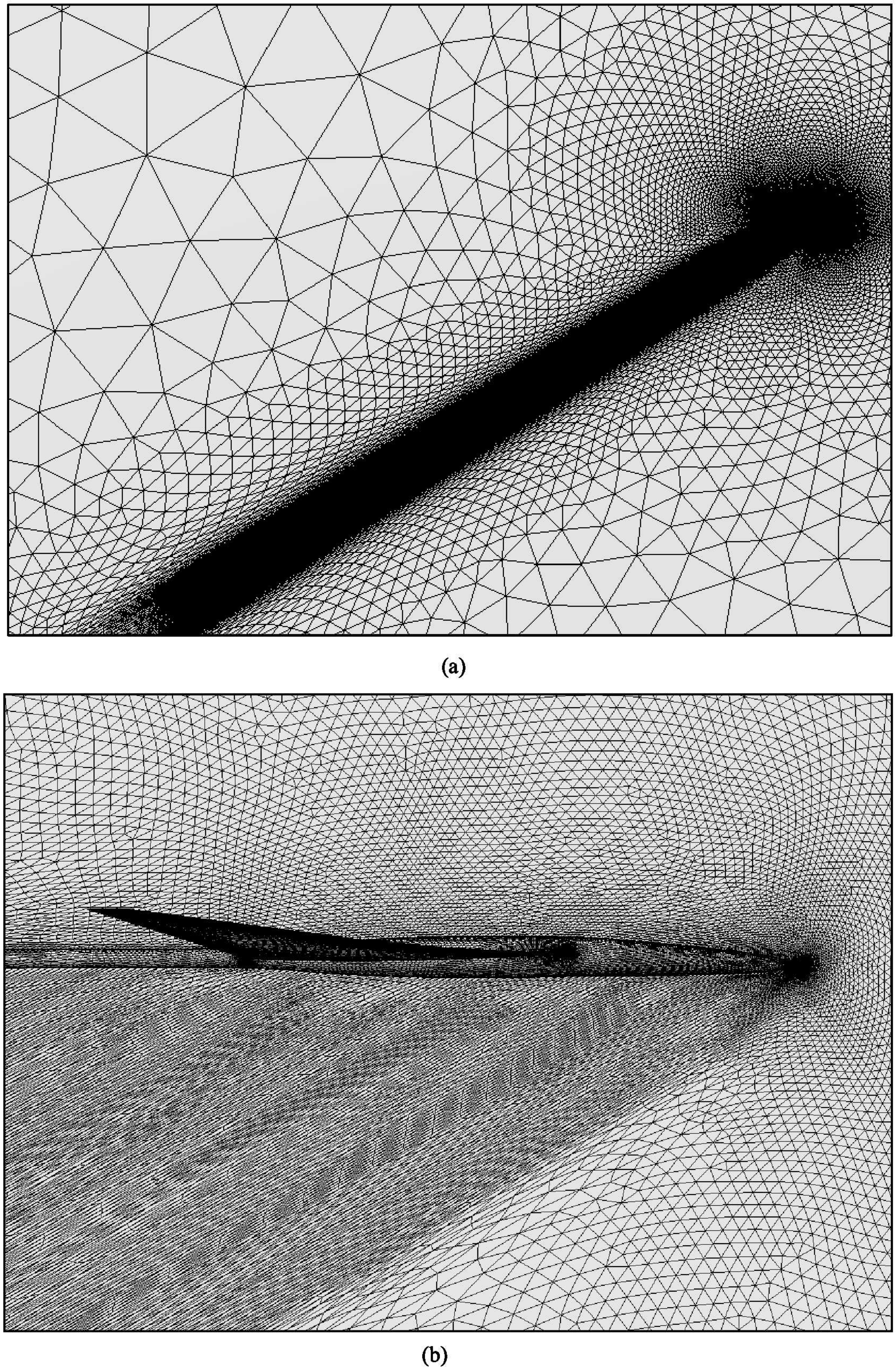


Figure 16. Volume sources for defining grid properties on the SLE configuration.

Figure 17 depicts the triangulation on the symmetry plane, which is indicative of the volume grid distribution in the field. As illustrated in Fig. 17(a), the grid is densely clustered normal to the Mach angle with a resolution that remains constant along the waves extended far in the field. Figure 17(b) shows a close-up view of the grid on the symmetry plane revealing the high-aspect-ratio grid elements stretched in the direction of the waves. Despite its super fine resolution in a large proportion, this grid contains only 8.7 million tetrahedrons and 1.5 million nodes thanks to the power of grid stretching in reducing the cell count.

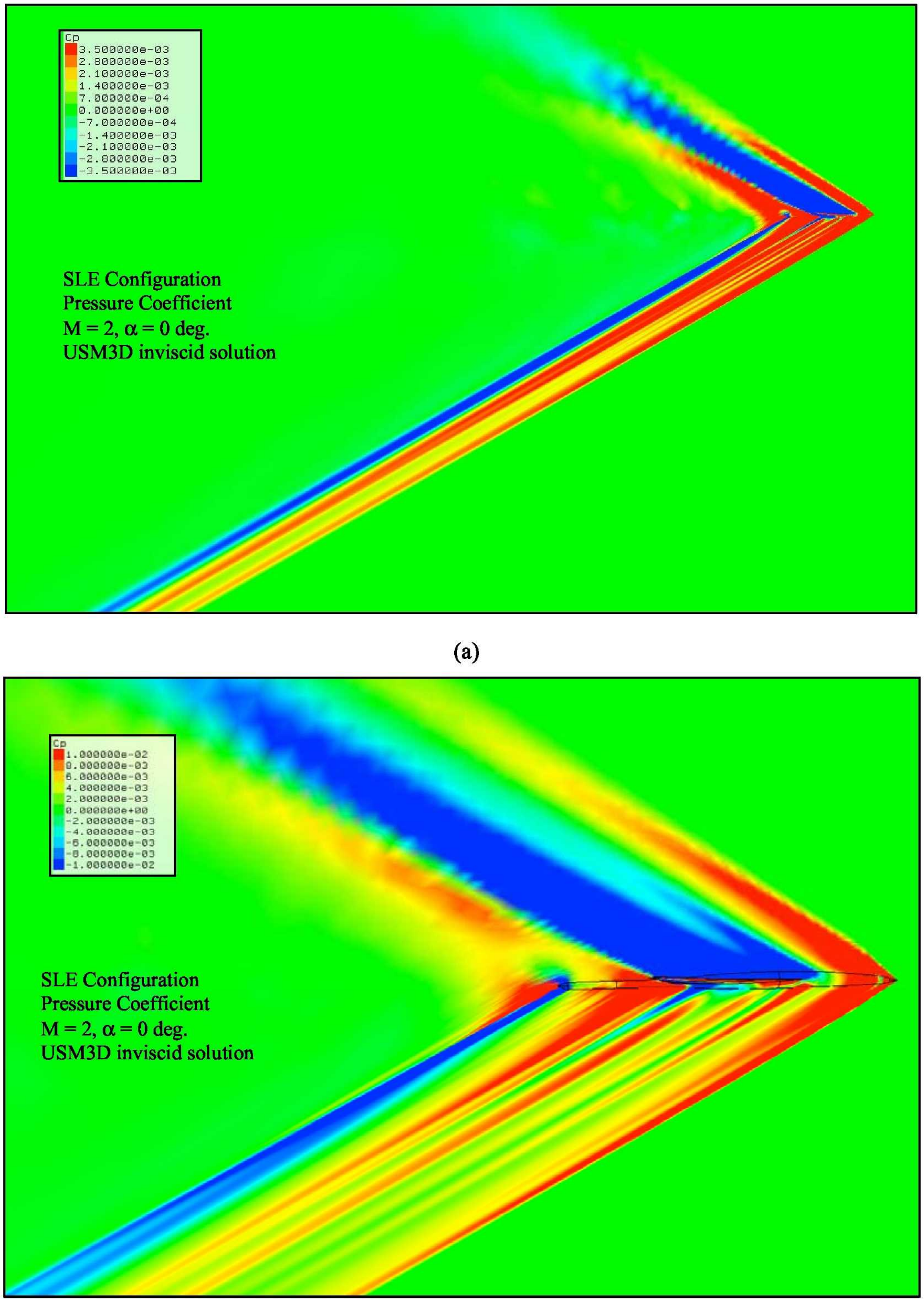
An inviscid flow solution was obtained on this grid using USM3D at an incidence angle of 0 degree and a freestream Mach number of 2. Figure 18 shows pressure coefficient contours on the symmetry plane at the far and near fields. As indicated, the pressure waves remain well defined even at 10 body lengths below the geometry, whereas those in the upper section (where the grid is not adequately resolved) dissipate quickly. An earlier experiment with the same configuration had demonstrated that while an isotropic grid refinement in the regions of interest improved the solution, it had not been sufficient to produce the quality results obtained with the present anisotropic grid refinement. A comparison of the CFD and experimental pressure distributions at 2.5 body lengths (*h/l*) below the aircraft is presented in Figure 19. This solution is considerably better than any prior solutions obtained with the same solver but using different grids, some even generated with adaptive refinement techniques



(a)

(b)

Figure 17. Surface mesh including triangulation of the symmetry plane for the SLE configuration: (a) far field, (b) close-up view near the geometry.



(a)

(b)

Figure 18. Pressure distribution on the SLE configuration at $M_\infty = 2$ and $\alpha = 0°$: (a) far field, (b) close-up view near the geometry.

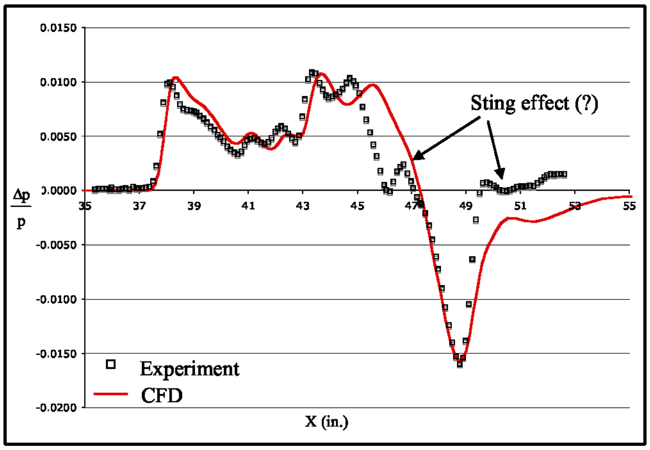


Figure 19. Comparison of streamwise pressure distributions between CFD and Wind tunnel data for the SLE configuration at *h/l* = 2.5, M_∞ =2, and α = 0°.

containing much larger number of grid elements. As shown in Figure 19, most pressure peaks are predicted well except in two areas that are marked with arrows. These discrepancies are believed to be due to differences in the shape of the sting and its connection to the fuselage between the wind tunnel and the present CFD models.

Recently, researchers at NASA Langley Research Center have conducted more comprehensive computational studies of the sonic boom problem with different methods, some using the present *volume* source technology. Interested readers are referred to References 8-10 for a thorough discussion of the problem and more results.

Human Body

The final example is a mesh on an unconventional configuration to demonstrate the capabilities of the curvature-based *surface* sources for automatic grid point distribution. The geometry is that of a human body defined in terms of 64 NURBS surfaces as shown in Fig. 20. The model features details with small and localized curvatures suitable for testing the new *surface* source and the bounding-box techniques. The problem makes use of all 64 NURBS surfaces as sources, and does not employ any other type of source for distributing grid points on the surface or in the field.

Figure 21 illustrates an isotropic triangular surface mesh generated on the geometry. As clearly shown in this figure, all geometric features are reflected on the mesh, and grid points are clustered according to the topology of the underlying surfaces automatically. For example, the mesh is appropriately refined at locations of higher curvature such as the facial features, fingers, and even small variations in the body curvature throughout the surface. The grid is smooth and of excellent quality.

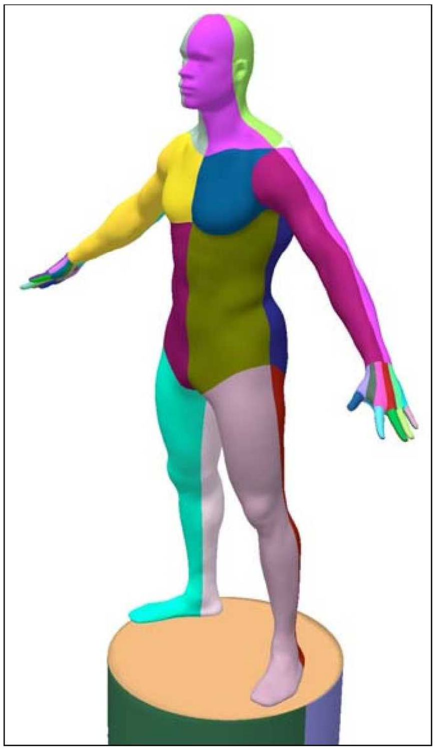


Figure 20. Human Body configuration defined in terms of 64 NURBS surfaces.

The control of grid distribution for this example has been remarkably convenient because the process is highly automatic yet flexible as the grid spacing and its rate of variation on the surface are readily controllable using a few global parameters. The application of *point*, *line*, or *volume* sources, instead of *surface* sources, would have involved manual insertion of a large number of sources and prescription of spacing parameters for every individual source; a process that is highly labor-intensive and time-consuming. Furthermore, the use of conventional sources would have required adjustment of source parameters through several trial-and-error iterations before obtaining the desired result. Nevertheless, the use of conventional sources (regardless of the number and manual adjustments of sources) is insufficient to produce the level of quality that the present mesh exhibits. Generation of the present grid using surface sources required no *ad hoc* adjustment of the grid parameters and no trial-and-error iterations were performed. The mesh shown in Fig. 21 was generated in the first try. This important aspect of the approach alleviates the adverse effect of human error/inexperience for generating difficult grids and also removes the user variance factor, so similar grids generated by different practitioners will be compatible and more standard.

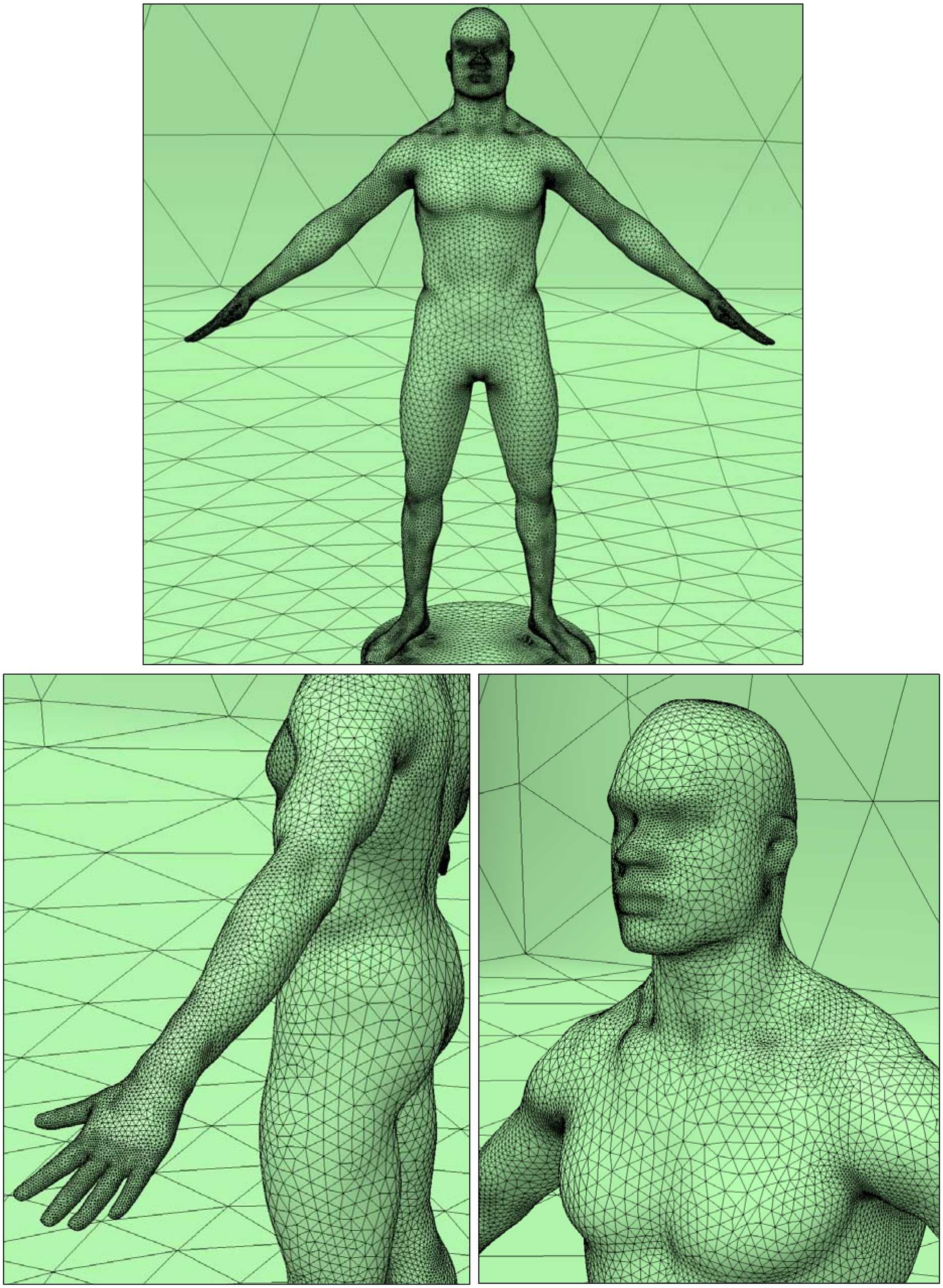


Figure 21. Triangulation of the Human-Body configuration using curvature-based mesh sizing.

It should be emphasized that generation of a grid using a large number of *surface* sources is impractical without the use of a background grid (such as that employed in Ref. 3) or the bounding-box technique implemented in the present work. Direct query of *surface* sources during grid generation is highly inefficient and computationally expensive. The efficiency of a *surface* source is directly proportional to the resolution of the underlying NURBS surface.

Table 3 compares the CPU times and the grid statistics for two RANS grids generated on the Human-Body configuration: one with the bounding-box technique and another using direct source query.

Table 3. Grid Statistics and Generation Times for the Human-Body Grid

Grid Segments	**Without Bounding-Box**		**With Bounding-Box**	
	Triangles/tetrahedra	Time (m)	Triangles/tetrahedra	Time (m)
Surface mesh	98,814	221.54	94,306	21.57
Volume mesh (viscous)	7,507,098	135.35	7,169,067	14.38
Volume mesh (inviscid)	2,779,059	1340.22	2,151,909	34.09
Total	10,286,157 (tet)	**1697.11**	9,320,976 (tet)	**70.04**

The numbers in this table indicate that the effect of the bounding-box on reducing the generation time for different grid components is enormous. Without the bounding-box approach, the total grid generation time is about 28 hours using direct source query. Applying the bounding-box technique reduces the generation time to less than 1.2 hours - a speedup factor of 24. Note that the viscous portion of the grid contains about 70 percent of the cells, yet it requires considerably less amount of generation time than the other grid components. The reason is that the Advancing-Layers method does not rely on source query for constructing viscous layers as much as the Advancing-Front technique does for generating the surface mesh and the inviscid portion of the grid. Although the bounding-box approach has improved the efficiency of the method substantially, the *surface* sources are still more CPU-intensive than other types of sources. However, saving of the labor time, which is otherwise required for setting up the conventional sources, fully compensates for the higher CPU time spent for query of *surface* sources. Furthermore, storing additional data on the bounding box faces in this example requires a modest increase in the computer memory of only 6.8 Mbytes for 64 *surface* sources, which is insignificant compared to the memory requirement of a full 3D background grid. These grids were generated using a Mac Pro desktop with 2 Dual-Core, 3Ghz Intel Xeon processors and 16GB of memory.

Concluding Remarks

A new sourcing technique for distribution of mesh length-scales has been devised and implemented in the unstructured grid generation code VGRID. In addition to conventional *point* and *line* sources, the new method utilizes *surface* and *volume* sources for automatic curvature-based grid sizing and also convenient point distribution in the volume. The new method does not require a 3D background grid for storing grid parameters and, thus, is more efficient and less memory-intensive. To increase the efficiency of the surface source query, a new approach of bounding-box is implemented. The bounding-box technique, in effect, reduces the problem of storing data on 3D background grids to a 2D problem in which only six faces of a bounding box are used as a storage device. A new exponential growth function produces smoother and more efficient grids and provides superior control over distribution of grid points in the field as compared to the previous method of source propagation. All types of sources support anisotropic grid stretching, which improves the grid economy considerably and provides more accurate solutions for certain aerodynamic applications such as sonic boom prediction.

The sample applications presented in this paper demonstrate the capability of the method for generating good quality grids suitable for computing complex aerodynamic problems. Each source type offers its own unique strengths for controlling grid distribution suitable for a specific problem at hand. For example, *volume* sources are best suited for the LAV and the Sonic Boom applications presented in this paper, whereas *surface* sources are more effective for handling problems such as the Human Body example. The strength of the present approach of sourcing is that all types of sources can be combined and simultaneously utilized in a problem featuring different geometric and/or flow physics requirements. Although the sourcing method alone is not a substitute for adaptive grid techniques, it provides better initial grids that can enhance the performance of many adaptation methods. Work is currently under way to further improve the efficiency of the *surface* sources and to extend the implementation of additional *volume* source shapes.

Acknowledgments

This work is supported by the NASA Fundamental Aeronautics Program, Subsonic Fixed Wing Project. The author would like to thank Richard A. Wahls (Project Scientist) for his encouragement and support during the course of this study, Jamshid Samareh (Vehicle Analysis Branch) for providing his NURBS library used for surface interrogation in this work, Ed Parlette (ViGYAN Inc.) and Craig Hunter (Configuration Aerodynamics Branch) for modifying the grid utility code GridTool for setting up *volume* and *surface* sources, Neal Frink (Configuration Aerodynamics Branch) and Mohagna Pandya (Analytical Services & Materials) for providing the USM3D solutions on the LAV and Circular Cylinder configurations, and Norma Farr (Analytical Services & Materials) for providing the SLE and Human-Body geometry definitions.

References

[1] Quadros, W.R., Shimada, K., and Owen, S. J., "3D Discrete Skeleton Generation by Wave Propagation on PR-Octree for Finite Element Mesh Sizing," ACM Symposium on Solid Modeling and Applications, Editors G. Elber, N. Patrikalakis, 2004.

[2] Zhu, J., Blacker, T., and Smith, R., "Background Overlay Grid Size Functions," Proceedings of 11th International Meshing Roundtable, pp. 65-74, 2002.

[3] Kania, L. A. and Pirzadeh, S. Z., "A Geometrically-derived Background Function for Automated Unstructured Mesh Generation," AIAA paper 2005-5240.

[4] Pirzadeh, S. Z., "Structured Background Grids for Generation of Unstructured Grids by Advancing-Front Method," *AIAA Journal*, Vol. 31, No. 2, pp. 257-265, 1993.

[5] Pirzadeh, S. Z., "Unstructured Viscous Grid Generation by the Advancing-Layers Method," *AIAA Journal*, Vol. 32, No. 8, pp. 1735-1737, 1994.

[6] Samareh, J. A., "Aerodynamic Shape Optimization Based on Free-form Deformation," AIAA paper 2004-4630.

[7] Frink, N. T., "Tetrahedral Unstructured Navier-Stokes Method for Turbulent Flows", *AIAA Journal*, Vol. 36, No. 11, pp. 1975-1982, 1998.

[8] Campbell, R. L., Carter, M. B., Deere, K. A., and Waithe, K. A., "Efficient Unstructured Grid Adaptation Methods for Sonic Boom Prediction," AIAA paper 2008-7327.

[9] Carter, M. B. and Deere, K. A., "A Grid Sourcing and Adaptation Study Using Unstructured Grids for Supersonic Boom Prediction ," AIAA paper 2008-6595.

[10] Park M. A. and Darmofal, D. L. "Output-Adaptive Tetrahedral Cut-Cell Validation for Sonic Boom Prediction," AIAA paper 2008-6594.