Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Three-dimensional elliptic grid generation with fully automatic boundary constraints

Upender K. Kaul *

NASA Ames Research Center, M/S 258-5, Moffett Field, CA 94035, United States

ARTICLE INFO

Article history:
Received 10 July 2009
Received in revised form 15 April 2010
Accepted 16 April 2010
Available online 29 April 2010

Keywords:
Enhanced elliptic grid generation
Three-dimensional boundary constraints
Decay functions
Grid clustering
Orthogonal grids
Single-zone grids
Mars Science Laboratory (MSL)
Canopy
Aeroshell
Inflatable Aerodynamic Decelerator (IAD)

ABSTRACT

A new procedure for generating smooth uniformly clustered three-dimensional structured elliptic grids is presented here which formulates three-dimensional boundary constraints by extending the two-dimensional counterpart[1] presented by the author earlier. This fully automatic procedure obviates the need for manual specification of decay parameters over the six bounding surfaces of a given volume grid. The procedure has been demonstrated here for the Mars Science Laboratory (MSL) geometries such as aeroshell and canopy, as well as the Inflatable Aerodynamic Decelerator (IAD) geometry and a 3D analytically defined geometry. The new procedure also enables generation of single-block grids for such geometries because the automatic boundary constraints permit the decay parameters to evolve as part of the solution to the elliptic grid system of equations. These decay parameters are no longer just constants, as specified in the conventional approach, but functions of generalized coordinate variables over a given bounding surface. Since these decay functions vary over a given boundary, orthogonal grids around any arbitrary simply-connected boundary can be clustered automatically without having to break up the boundaries and the corresponding interior or exterior domains into various blocks for grid generation. The new boundary constraints are not limited to the simply-connected regions only, but can also be formulated around multiply-connected and isolated regions in the interior. The proposed method is superior to other methods of grid generation such as algebraic and hyperbolic techniques in that the grids obtained here are C^2 continuous, whereas simple elliptic smoothing of algebraic or hyperbolic grids to enforce C^2 continuity destroys the grid clustering near the boundaries.

1. Introduction

A smooth and orthogonal grid around arbitrary geometries is invariably generated using grid generation techniques based on the solution of partial differential equations. One such powerful technique is based on the solution of elliptic partial differential equations [1–9]. Elliptic grid generation methods are generally used to create C^2 smooth grids on which accurate numerical solutions [10] to a given physical problem are obtained. This grid generation procedure involves the numerical solution of inhomogeneous elliptic partial differential equations [1–9]. The inclusion of inhomogeneous terms in these equations allows a grid to satisfy clustering and orthogonality properties in the vicinity of specific surfaces in three dimensions and in the vicinity of specific lines in two dimensions. Although the prediction of quantities such as pressure in a given CFD simulation can be reasonably obtained by other grid generation methods also, elliptic grids yield more accurate predictions

* Tel.: +1 650 604 2973; fax: +1 650 604 4036.
E-mail addresses: ukaul@mail.arc.nasa.gov, Upender.Kaul@nasa.gov.

[1] US patent 7231329.

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2010.04.028

Nomenclature

x,y,z	Cartesian coordinates
ξ,η,ζ	generalized curvilinear coordinates
$J = J((x,y,z)/(\xi,\eta,\zeta))$	Jacobian of transformation
$b_i(\eta,\zeta), d_i(\xi,\zeta), f_i(\xi,\eta)$	respective decay parameter functions for the ξ,η,ζ boundaries, calculated automatically
i	index for a particular boundary component

of higher order quantities such as skin friction and heat transfer, as compared with other grid generation techniques [10]. This is because the elliptic grids satisfy the C^2 continuity requirement that permits accurate prediction of these higher order quantities. For example, the need to determine the heat transfer accurately in a CFD simulation over a re-entry space vehicle such as the space shuttle is critical in ensuring the health of the vehicle during re-entry. For this reason, elliptic grids are being generated, for example, inside the cavities over the shuttle's outer surface created by impacts or other damage resulting in the loss or damage of a given tile. Although grids created by algebraic or hyperbolic techniques can be smoothed by elliptic smoothing to ensure the C^2 continuity of the grid, the near-boundary clustering of the grid gets destroyed in this process, resulting in poor heat transfer prediction.

Many other elliptic grid generation methods have been reported in the literature [11–19]. For example, in the method of Azarenok [11], use of elliptic functional is made to enforce grid orthogonality and clustering close to the boundary.

In the elliptic grid generation techniques presented earlier [3–5], the free parameters used in the inhomogeneous term that control the clustering and the orthogonality of the grid near boundaries were specified externally. This requirement was alleviated partially by a later study [6], where the near-boundary orthogonality was satisfied automatically, without the need for external specification of the associated free parameters. But, the other parameters, called the decay parameters, that control the rate of clustering near a boundary, still needed to be prescribed. The author in his 2D study [8], proposed an automatic procedure that obviated the need for the external specification of these decay parameters. In this 2D study [8], these decay parameters were no longer the prescribed constants, but functions of a single generalized coordinate variable, say, η, over a given generalized coordinate boundary, say ζ = constant. Thus the elliptic grid generation procedure was fully automatic, without any external parameter specification, and the rate of clustering over a given boundary was adaptable, from point to point. In the present study, this methodology [8] has been extended to three dimensions so that the orthogonality and the rate of clustering in, e.g. ξ direction are free to adapt from point to point over a given (η,ζ) surface.

2. Formulation

Two-dimensional form of the inhomogeneous elliptic partial differential equations (pdes) for grid generation used by Thompson et al. [3–5] contained four explicit parameters that need to be prescribed by the user. Later, Steger and Sorenson [6] prescribed a semi-automatic scheme that reduced the requirement for explicit prescription of these parameters to the two parameters, called decay parameters. In a subsequent study [8], the author further enhanced this methodology to fully automate the elliptic grid generation process that completely eliminated the need for the explicit user prescription of decay parameters.

Additionally, in the enhanced fully automated methodology of Kaul [8], the decay parameters are no longer a specified set of constants chosen manually for the four boundaries for two-dimensional applications, but four decay functions, each a function of one independent coordinate variable over a given boundary, which are calculated as part of the solution process. This feature makes it possible to cluster a grid normal to any arbitrarily shaped boundary.

Below, a three-dimensional analog of the earlier two-dimensional methodology [3–5] is given. Then, the development of the semi-automatic [6] and the enhanced fully automated [8] methodologies is briefly discussed. Finally, the extension of the fully automated methodology [8] for two-dimensions is extended to three-dimensional applications.

Three-dimensional governing equations for elliptic grid generation are expressed as:

$$\xi_{xx} + \xi_{yy} + \xi_{zz} = P(\xi,\eta,\zeta) = -a_i \cdot \mathrm{sgn}(\xi - \xi_i)\exp\{-b_i|\xi - \xi_i|\}, \tag{1}$$

$$\eta_{xx} + \eta_{yy} + \eta_{zz} = Q(\xi,\eta,\zeta) = -c_i \cdot \mathrm{sgn}(\eta - \eta_i)\exp\{-d_i|\eta - \eta_i|\}, \tag{2}$$

$$\zeta_{xx} + \zeta_{yy} + \zeta_{zz} = R(\xi,\eta,\zeta) = -e_i \cdot \mathrm{sgn}(\zeta - \zeta_i)\exp\{-f_i|\zeta - \zeta_i|\}, \tag{3}$$

where ξ, η and ζ are generalized curvilinear coordinates, x, y and z are Cartesian coordinates, and $P(\xi,\eta,\zeta)$, $Q(\xi,\eta,\zeta)$ and $R(\xi,\eta,\zeta)$, are inhomogeneous terms; a_i, b_i, c_i, d_i, e_i and f_i are manually selected constants, and the subscript "i" refers to a particular boundary component associated with the problem.

A simplified two-dimensional form of Eqs. (1)–(3) is written as [6]

$$\xi_{xx} + \xi_{yy} = -a_i^{\cdot}\mathrm{sgn}(\eta - \eta_i)\exp\{-d_i|\eta - \eta_i|\}, \tag{4}$$

$$\eta_{xx} + \eta_{yy} = -c_i^{\cdot}\mathrm{sgn}(\eta - \eta_i)\exp\{-d_i|\eta - \eta_i|\}. \tag{5}$$

Eqs. (4) and (5) were used [6] to semi-automatically generate the two-dimensional grids with appropriate clustering and orthogonality at the walls. The functions, a and c, are obtained by satisfying the orthogonality constraint and a prescribed spacing requirement at a given η boundary, as shown [6]. But, the decay parameter, d_i, was prescribed manually, for the boundary, η_i.

As mentioned above, a fully automatic boundary procedure was proposed [8] to eliminate the need for manual selection of the decay parameters for each boundary. Also, these decay functions, no longer constants, but functions of the independent coordinate variables along a given boundary, were calculated as part of the solution process. This fully automatic boundary procedure has been used successfully for various two-dimensional complex geometries [8].

In this paper, extension of this 2D automatic procedure to three-dimensional applications is presented. Geometries chosen here are those for the MSL aeroshell and canopy, as well as the geometry for a tension cone IAD. These geometries are being used to study planetary re-entry flows of Entry, Descent and Landing (EDL) systems. The present procedure makes it possible to generate single-block grids for these geometries. Another 3D geometry, analytically defined, is also used to demonstrate the present procedure.

Using the fully automated approach [8], for a given boundary, ζ_i ($\zeta > \zeta_i$), for example, Eqs. (1)–(3) are modified here in the context of Eqs. (4) and (5) and are written in the following form:

$$\xi_{xx} + \xi_{yy} + \xi_{zz} = p_3(\xi, \eta, \zeta), \tag{6}$$

where, $p_3(\xi,\eta,\zeta) = -a_{3,i}(\xi,\eta)\mathrm{sgn}(\zeta-\zeta_i)\exp\{-f_i(\xi,\eta)|\zeta-\zeta_i|\} \approx \left(-a_{3,i}(\xi,\eta) + a_{3,i}(\xi,\eta)f_i(\xi,\eta)|\zeta-\zeta_i|\right)\mathrm{sgn}(\zeta-\zeta_i)$

$$\eta_{xx} + \eta_{yy} + \eta_{zz} = q_3(\xi, \eta, \zeta), \tag{7}$$

where, $q_3(\xi,\eta,\zeta) = -c_{3,i}(\xi,\eta)\mathrm{sgn}(\zeta-\zeta_i)\exp\{-f_i(\xi,\eta)|\zeta-\zeta_i|\} \approx \left(-c_{3,i}(\xi,\eta) + c_{3,i}(\xi,\eta)f_i(\xi,\eta)|\zeta-\zeta_i|\right)\mathrm{sgn}(\zeta-\zeta_i)$

$$\zeta_{xx} + \zeta_{yy} + \zeta_{zz} = r_3(\xi, \eta, \zeta), \tag{8}$$

where,

$$r_3(\xi,\eta,\zeta) = -e_{3,i}(\xi,\eta)\mathrm{sgn}(\zeta-\zeta_i)\exp\{-f_i(\xi,\eta)|\zeta-\zeta_i|\}, \approx \left(-e_{3,i}(\xi,\eta) + e_{3,i}(\xi,\eta)f_i(\xi,\eta)|\zeta-\zeta_i|\right)\mathrm{sgn}(\zeta-\zeta_i)$$

and, where $a_{n,i} = a_{n,i}(\xi,\eta)$, $c_{n,i} = c_{n,i}(\xi,\eta)$ and $e_{n,i} = e_{n,i}(\xi,\eta)$, and where $n = 3$ corresponds to the ζ boundary, under consideration.

Similar expressions hold for the inhomogeneous terms for the ξ ($n = 1$) and η ($n = 2$) boundaries.

It is worth noting here that, for example, the expression for p_3 in Eq. (6) is guaranteed to yield the maximum and minimum values at the corresponding ζ_i boundaries. As an example, for $\zeta > \zeta_i$, p_3 acts as a source term and for $\zeta < \zeta_i$, p_3 acts as a sink term, thus ensuring the minimum and maximum values at the corresponding boundaries respectively.

The positive decay parameters, b_i, d_i and f_i, for the corresponding boundaries, ξ, η and ζ, respectively, are expressed as parameter functions, $b_i(\eta,\zeta)$, $d_i(\xi,\zeta)$ and $f_i(\xi,\eta)$ in the present approach, and the corresponding terms, $a_{2,i}(\xi,\zeta)$, $c_{2,i}(\xi,\zeta)$, $e_{2,i}(\xi,\zeta)$ and $a_{1,i}(\eta,\zeta)$, $c_{1,i}(\eta,\zeta)$, $e_{1,i}(\eta,\zeta)$, hold for η and ξ boundaries respectively.

Without loss of generality, one can consider the neighborhood of a given ζ-boundary segment i, $\zeta - \zeta_i \geqslant 0$. It can be shown that in a selected region on one side of this boundary segment, where $f_i(\xi,\eta)(\zeta-\zeta_i) \ll 1$, the governing equations and the inhomogeneous terms have the limiting forms similar to those given in [8].

Treatment of a boundary segment, $(\zeta - \zeta_i) < 0$, is analogous. Similar pdes hold for regions close to ξ-boundary and η-boundary segments.

Rewriting, the limiting governing equations near a ζ boundary become [8]

$$\begin{aligned}
&\xi_{xx} + \xi_{yy} + \xi_{zz} - a_{3,i}(\xi,\eta)f_i(\xi,\eta)|\zeta-\zeta_i|\mathrm{sgn}(\zeta-\zeta_i) = -a_{3,i}(\xi,\eta)\mathrm{sgn}(\zeta-\zeta_i),\\
&\eta_{xx} + \eta_{yy} + \eta_{zz} - c_{3,i}(\xi,\eta)f_i(\xi,\eta)|\zeta-\zeta_i|\mathrm{sgn}(\zeta-\zeta_i) = -c_{3,i}(\xi,\eta)\mathrm{sgn}(\zeta-\zeta_i),\\
&\zeta_{xx} + \zeta_{yy} + \zeta_{zz} - e_{3,i}(\xi,\eta)f_i(\xi,\eta)|\zeta-\zeta_i|\mathrm{sgn}(\zeta-\zeta_i) = -e_{3,i}(\xi,\eta)\mathrm{sgn}(\zeta-\zeta_i).
\end{aligned}$$

It can be easily seen that the pdes above represent a self-adjoint operator of the form:

$$L(\theta) = \mathrm{div}(k\ \mathrm{grad}(\theta)) - q\theta$$

and, therefore, boundary constraints, valid in the neighborhood of each of the six boundary segments, are incorporated by applying the Green's theorem in three dimensions. For example, for the ζ boundary, the constraint will be given by

$$\int_S (\partial\theta/\partial n)\,d\sigma = \int_V \{-e_{3,i}(\xi,\eta) + e_{3,i}(\xi,\eta)f_i(\xi,\eta)|\theta|\}\mathrm{sgn}(\theta)\,d\tau, \tag{9}$$

where $\theta = \zeta - \zeta_i$, $d\sigma$ is a differential area element, $d\tau$ is a differential volume element, k is identically equal to 1 and n refers to a direction that is locally normal to a bounding surface S representing a totality of six surfaces including the boundary segments of interest and V is a volume enclosed by S. This integral-type boundary constraint can be used to calculate the decay parameter analog, $e_{3,i}(\xi,\eta)f_i(\xi,\eta)$. Similar constraints can be used to calculate the decay parameter analogs, $a_{1,i}(\eta,\zeta)b_i(\eta,\zeta)$ and $c_{2,i}(\eta,\zeta)d_i(\xi,\zeta)$. The essence of the present technique lies in the integral constraint (9), which is based on the Green's Theorem as applied to self-adjoint operators shown above. The integral constraint (9) allows the functions b, d and f to be calculated automatically as part of the solution. Thus, the present technique is fully automatic as opposed to the "semi-automatic" technique [6]. While it is true that all the parameters are automatically calculated as part of the solution, the control over the rate of stretching and the boundary grid spacing are maintained by the present method, according to the given grid distribution, specified as an initial condition for the method. It is the initial prescription of the starting grid that drives the decay parameters to attain values consistent with the initially prescribed spacing and stretching rate. When Neumann boundary conditions are prescribed along a given boundary, the boundary points on that boundary are then free to slide. But, the normal grid spacing on a given boundary is fixed by the initial prescription.

The novelty of the present method is actually twofold: (i) the solutions have been obtained automatically without the user prescribing the decay parameter functions, *b*, *d* and *f*, which is the major advantage of the present technique over the classical approach, and (ii) the nonlinear inhomogeneous term has been linearized that ensures convergence to the smooth solution. The novelties as mentioned in (i) and (ii) above are not mutually exclusive since the linearization of the homogeneous terms makes it possible to express the governing equations in the self-adjoint operator form, as shown above.

As pointed out by Kaul [8], the physical analogy in 2D does not extend to 3D, since the 2D physical analogy of the limiting form of governing equations shown above holds for heat transfer over long thin fins (US patent 7231329). In the context of grid clustering, the reader is referred to a discussion of the classical approach [5] based on differential geometry.

When expressed in terms of the generalized coordinate, ζ, the boundary constraint given in Eq. (9) is transformed as follows:

$$\int_S I\,d\sigma = \int_S (\partial\theta/\partial n)\,d\sigma = \int_S (\partial\zeta/\partial n)\,d\sigma. \tag{10}$$

The integral $\int_S I\,d\sigma$ in Eq. (10) can be written as an algebraic sum of six integrals, evaluated over the indicated boundary segments:

$$\int_S I\,d\sigma = \int_{\xi\,\max} I\,d\sigma + \int_{\eta\,\max} I\,d\sigma + \int_{\zeta\,\max} I\,d\sigma - \int_{\xi\,\min} I\,d\sigma - \int_{\eta\,\min} I\,d\sigma - \int_{\zeta\,\min} I\,d\sigma, \tag{11}$$

where the surface configurations $\xi\max$, $\xi\min$, etc. represent the corresponding boundary segments that together make up the surface *S*. For the first and fourth integral pair, the second and fifth integral pair, and the third and sixth integral pair in Eq. (11), the following respective relations are derived:

$$\int_\xi I\,d\sigma = \int_\xi (1/J\sqrt{\alpha_{11}})\alpha_{13}\left[\left(x_\eta^2+y_\eta^2+z_\eta^2\right)\left(x_\zeta^2+y_\zeta^2+z_\zeta^2\right)\right]^{1/2} d\zeta\,d\eta, \tag{12}$$

$$\int_\eta I\,d\sigma = \int_\eta (1/J\sqrt{\alpha_{22}})\alpha_{23}\left[\left(x_\xi^2+y_\xi^2+z_\xi^2\right)\left(x_\zeta^2+y_\zeta^2+z_\zeta^2\right)\right]^{1/2} d\xi\,d\zeta, \tag{13}$$

$$\int_\zeta I\,d\sigma = \int_\zeta (1/J)\left[\alpha_{33}\left(x_\eta^2+y_\eta^2+z_\eta^2\right)\left(x_\xi^2+y_\xi^2+z_\xi^2\right)\right]^{1/2} d\eta\,d\xi, \tag{14}$$

$$\begin{aligned}
\alpha_{11} &= J^2\left(\xi_x^2+\xi_y^2+\xi_z^2\right),\\
\alpha_{22} &= J^2\left(\eta_x^2+\eta_y^2+\eta_z^2\right),\\
\alpha_{33} &= J^2\left(\zeta_x^2+\zeta_y^2+\zeta_z^2\right),\\
\alpha_{12} &= J^2\left(\xi_x\eta_x+\xi_y\eta_y+\xi_z\eta_z\right),\\
\alpha_{13} &= J^2(\xi_x\zeta_x+\xi_y\zeta_y+\xi_z\zeta_z),\\
\alpha_{23} &= J^2\left(\eta_x\zeta_x+\eta_y\zeta_y+\eta_z\zeta_z\right),
\end{aligned}$$

where $J = J((x,y,z)/(\xi,\eta,\zeta))$ is a Jacobian of the transformation $(x,y,z) \to (\xi,\eta,\zeta)$.

The Eqs. (12)–(14), for the three integral pairs, are valid for the corresponding ξ,η,ζ surfaces defining the bounding box for determination of clustering near the given ζ = constant surface. The functions, *a*, *c* and *e* are automatically obtained by satisfying the orthogonality and prescribed spacing requirements at a given ζ boundary, just as *a* and *c* are obtained for the 2D case in Ref. [6] for a given η boundary (see Eqs. (4) and (5) above). Then, the constraint given by Eq. (9), in conjunction with Eqs. (12)–(14), automatically gives the function *f*, for the given ζ boundary. In the classical approach, the user defined parameter *d*, e.g. for the 2D case, is thought of as the decay parameter, but as seen by the heat transfer analogy in 2D [8], it is the products of *a* and *f*, *c* and *f*, *e* and *f*, in the 3D case, that define the decay rate from the given ζ_i boundary and hence these products are referred to as the decay parameter analogs. The boundary grid spacing normal to all the six boundaries is controlled by the grid spacings, prescribed initially.

Again, as pointed out earlier, these decay parameters are no longer user specified constants, as in the conventional methods, but functions of two independent coordinate variables over a given surface normal to which the clustering is required. Another notable point is that since the inhomogeneous terms are now linearized, the method is robust and relatively insensitive to the grid distribution as in the conventional method.

Eqs. (12)–(14) can be used to express the boundary constraints in the computational space (generalized coordinate variables). The following governing equations in computational space are solved, subject to the boundary constraints derived above:

$$\alpha_{11}x_{i,\xi\xi} + \alpha_{22}x_{i,\eta\eta} + \alpha_{33}x_{i,\zeta\zeta} + 2\{\alpha_{12}x_{i,\xi\eta} + \alpha_{13}x_{i,\xi\zeta} + \alpha_{23}x_{i,\eta\zeta}\} = -J^2\{p_3x_{i,\xi} + q_3x_{i,\eta} + r_3x_{i,\zeta}\},$$
$$x_i = x, y \text{ or } z.$$

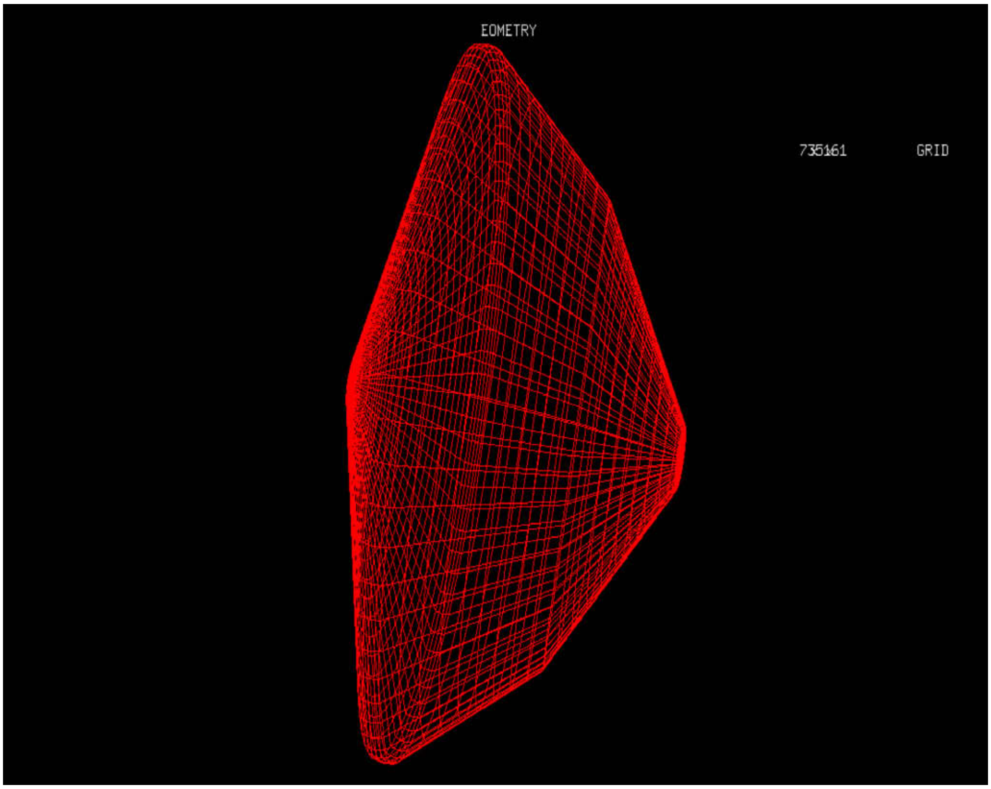


Fig. 1. Grid over the MSL aeroshell geometry.

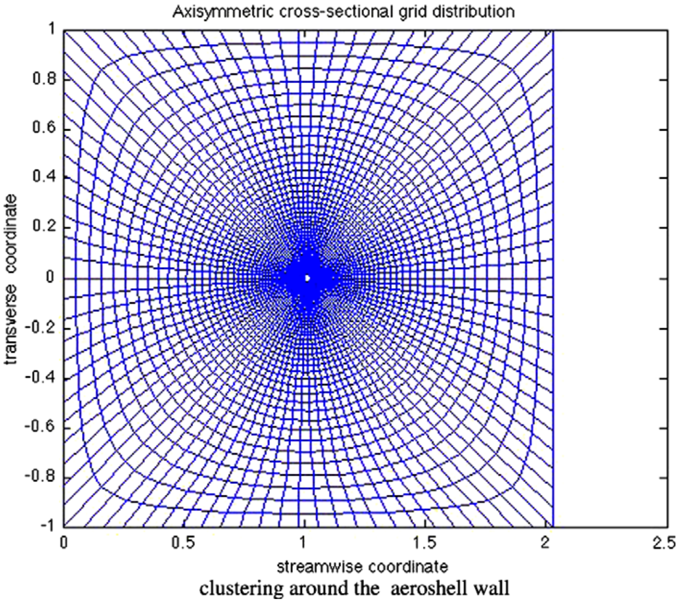


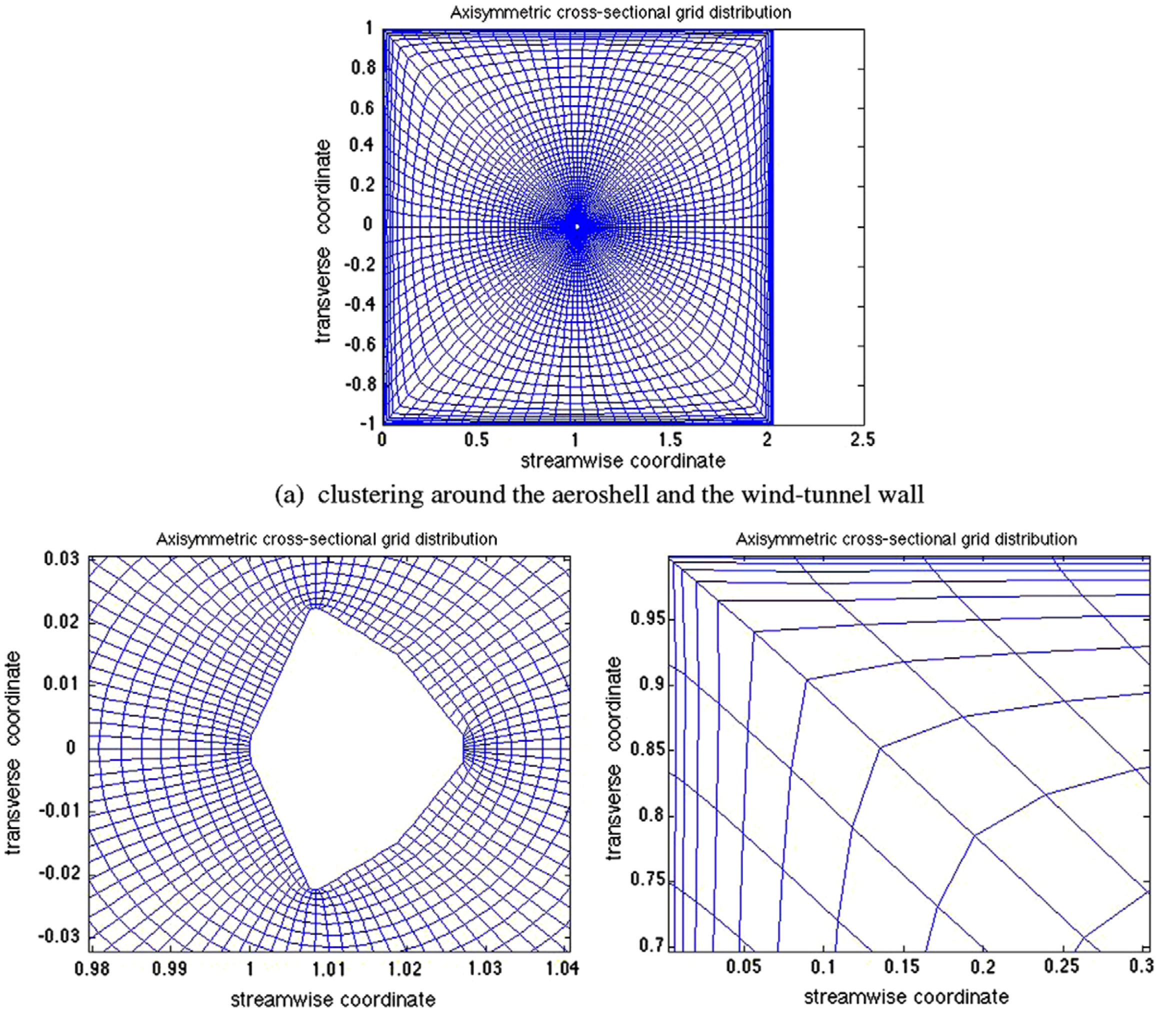
Fig. 2. Cross-section of the elliptic grid around the MSL aeroshell; outer boundary is far-field.

A decay parameter analog, such as $e_{3,i}(\xi,\eta)f_i(\xi,\eta)$, may vary with one or more of the generalized coordinates, (ξ,η), rather than being constant, and this variation is determined as part of the solution of the elliptic grid problem, rather than being prescribed initially by the user. This grid solution can be determined for either a static grid or a dynamically changing grid. Hence, dynamically changing grids can be generated automatically without the user intervention.

The preceding analysis has focused on the neighborhoods of the grid boundary segments. As noted in the preceding, in an interior region, far from the grid boundary segments, the defining partial differential equations become homogeneous, and an orthogonal and uniform grid cell distribution is obtained which smoothly transitions from the interior to the boundaries.

3. Numerical approach and boundary conditions

The numerical approach used to solve the elliptic problem is a successive line relaxation method in which, as the initially specified clustering rate near the boundary is increased, the relaxation parameter needs to be decreased. This experiment is very simple and fast. Once the functional relationship between the relaxation parameters and the clustering rate is determined, the technique is relatively insensitive to any other effects including those of the normal grid spacing at the boundaries. In the author's experience, as long as the initial grid is reasonably prescribed in a non-degenerate manner, the present procedure invariably leads to an acceptable solution with the orthogonality and, especially, clustering requirements fulfilled. This is because the nonlinear inhomogeneous terms are linearized in the present case which ensure a relatively smooth convergence to the desired solution.



(a) clustering around the aeroshell and the wind-tunnel wall

(b) enlarged close to the aeroshell

(c) enlarged close to the tunnel wall

Fig. 3. Cross-section of the elliptic grid around the MSL aeroshell and the wind-tunnel wall.

In the numerical scheme, all the first order and second order spatial derivative terms are central-differenced and hence are second order accurate, including those used in the boundary constraints. The tridiagonal system of discretized equations obtained thus is inverted using the Thomas algorithm in a given coordinate direction. For further details of the numerical method used, the reader is referred [6,8].

There is a choice between the Dirichlet and Neumann type boundary conditions. Typically, in a given direction, where grid points are allowed to slide, Neumann boundary conditions are used, and Dirichlet boundary conditions are used in the other direction(s).

4. Results

Some key elements of the enhanced elliptic grid generation methodology are demonstrated through a few selected computational grid generation examples. The geometries for planetary EDL systems, such as an aeroshell and a canopy for the Mars Science Laboratory (MSL), as well as the tension cone IAD, are considered here. In subsequent computational fluid dynamics studies, a direct comparison with experimental data will be made for these and other geometries. There are currently some parallel experimental studies [20–22] being carried out on similar geometries. For the MSL and the IAD geometries, single-block grids are generated with the present grid generation procedure which are uniformly clustered at the body. Also, as mentioned earlier, the present methodology is demonstrated for an analytically defined 3D geometry. With the conventional elliptic grid generation methodology, as mentioned above, decay parameters are prescribed by the user as constants over a given boundary. If the boundary slope is discontinuous, then this prescription fails to generate uniformly clustered grids around sharp corners and high curvature regions. To avoid this problem encountered in the conventional approach, grids are typically decomposed into multiple blocks and the grids are generated separately for these blocks.

In the present methodology, decay parameters are automatically calculated as decay functions, as part of the solution. Thus, a uniformly clustered grid is generated over an arbitrarily shaped boundary as a single-block grid, as long as the boundary is simply-connected. The grid over the MSL aeroshell geometry, shown in Fig. 1, is generated as a single-block 51 × 72 × 61 grid; 72 points in the meridional direction, 51 points in the streamwise direction and 61 points in the direction normal to the body. An axisymmetric cross-section, 51 × 61, of the volume grid around the aeroshell is shown in Fig. 2. This grid is shown to be clustered at the aeroshell wall. Away from the aeroshell, the grid uniformly stretches to the far-field boundary. No clustering requirement is enforced at the far-field boundary. A note of caution should be made here regarding realizing strict orthogonality of the grid lines at the surface. Theoretically, it is a matter of carrying the iterations indefinitely to make the grid orthogonal in the limit. However, in practice, strict orthogonality will not be achieved, since that is a function of how stretched or clustered the grid is at the boundaries. The higher the rate of clustering at the boundary, the higher the degree of orthogonality at the boundary. This has to do with the assumption made in the linearization of the inhomogeneous terms, where, as the clustering rate increases, the decay parameter decreases, and hence the linearization approximation of the exponent term becomes more accurate. In this context, for example, Visbal and Knight [16] have observed that the strictly orthogonal grids can be generated with a partial control of the mesh spacing and nearly orthogonal grid with strict control of the mesh spacing. In the author's experience, with the Neumann boundary conditions, the grids generated near the boundaries are strictly orthogonal at those boundaries, but with Dirichlet boundary conditions they are not always strictly orthogonal.

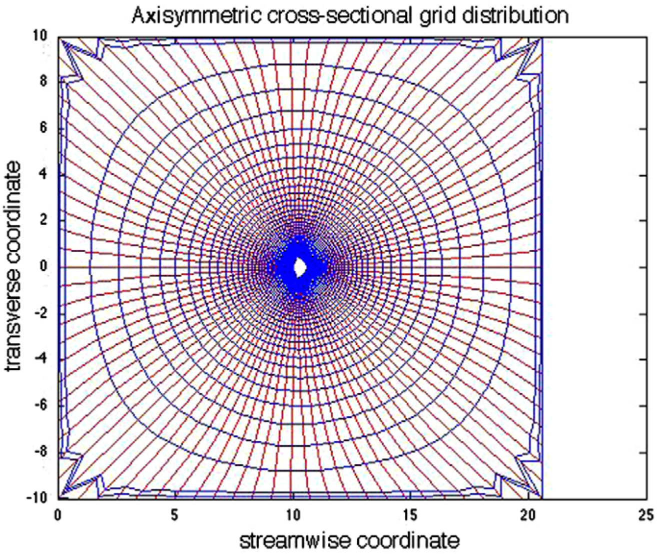
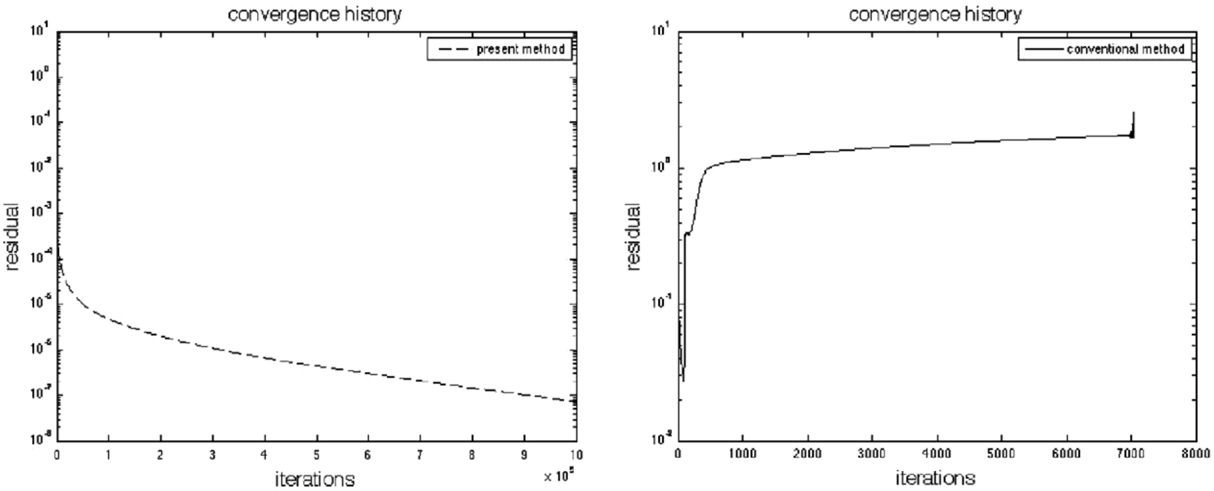


Fig. 4. A representative aeroshell grid calculated with the standard technique (user-specified decay parameter constants).

Fig. 3(a) and (b) shows the elliptic grid around the aeroshell bounded by a tunnel wall. Therefore, the grid is clustered at both the aeroshell and tunnel walls. Again, as noted earlier, the degree of orthogonality at the tunnel walls is not strictly enforced.

A representative calculation is made with the standard technique [5,6] with arbitrarily chosen values for the decay parameter. In this case, the grid clustering fails at the tunnel walls and actually leads to an erroneous solution at the corners. This is shown in Fig. 4. With the decay parameter constants thus specified by the user, in this case, the clustering is not uniform and fails to resolve the corners. The clear advantage of the present technique lies in the automatic calculation of the decay parameter functions that enables uniform clustering around any simply-connected domain with discontinuous slopes.

Convergence history for the present method and that for the standard method are shown in Fig. 5(a) and (b). A smooth convergence is exhibited by the present fully automatic method in Fig. 5(a). The final grid corresponding to this calculation is shown in Fig. 3, as discussed above. Fig. 5(b) shows the corresponding convergence history for the standard method [5,6] where iterations eventually diverge, with the Jacobians becoming negative. Corresponding grid, just before the calculations blow up, is shown in Fig. 4 above.



(a) present method

(b) standard method

Fig. 5. Convergence history.

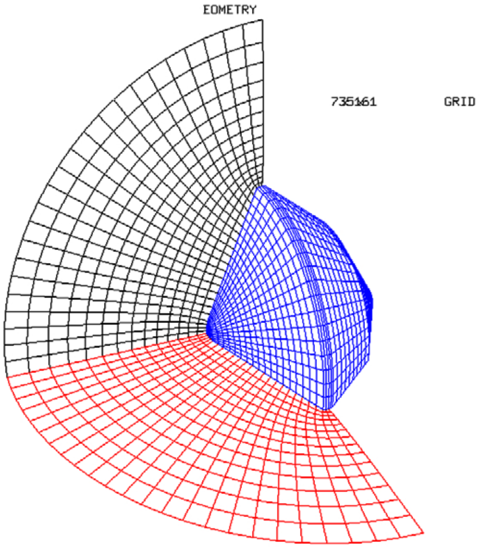


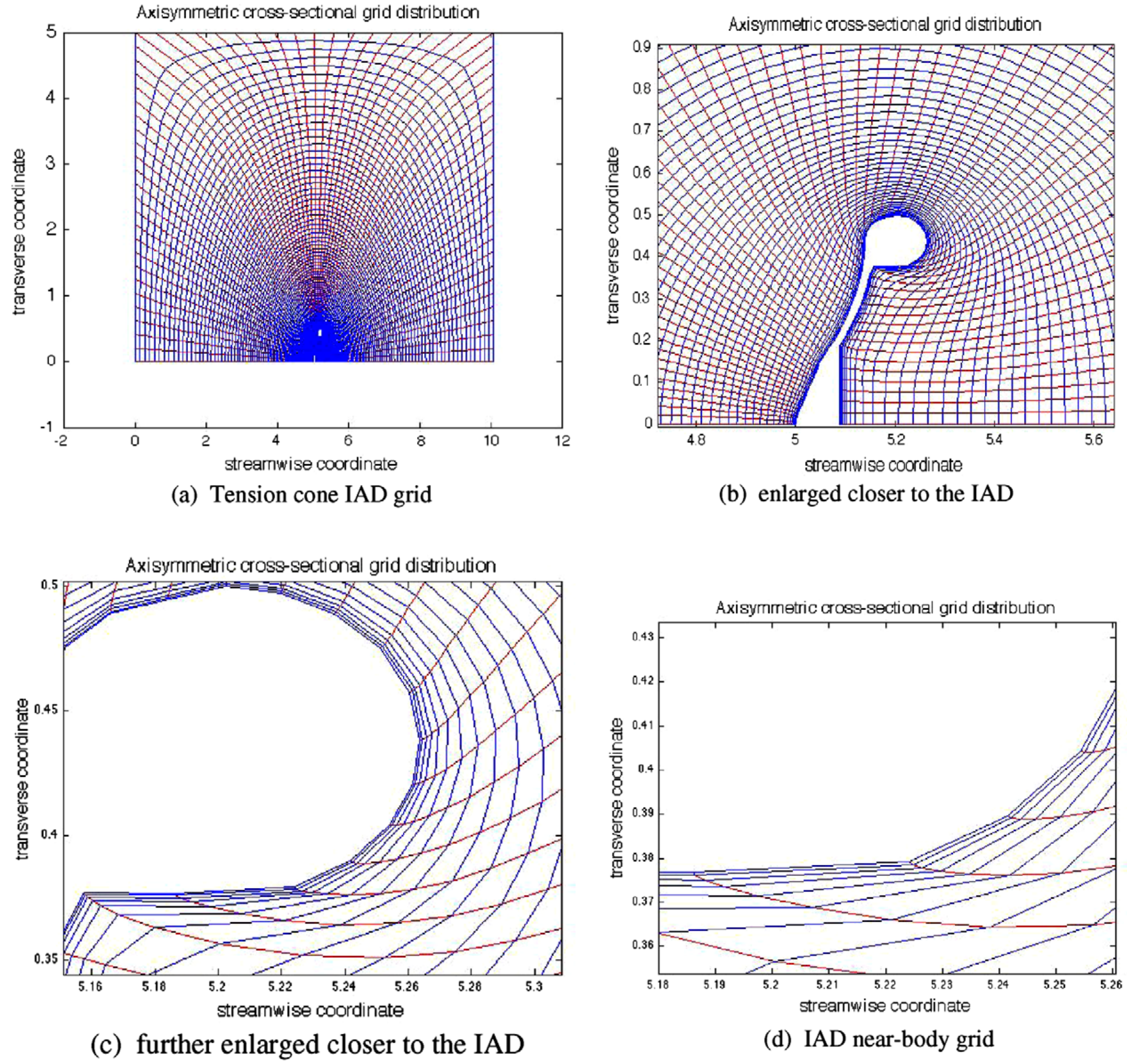
Fig. 6. A three-dimensional slice of the grid in the vicinity of the aeroshell.

A three-dimensional slice through the volume grid obtained with the present method including the aeroshell surface is shown in Fig. 6.

Fig. 7(a)–(d) shows a single-block 91 × 72 × 67 grid for the tension cone IAD; Fig. 7(b)–(d) show progressively enlarged views of the IAD grid near the body.

One of the key elements of the new fully automatic elliptic grid generation procedure is that since the decay function varies over a given body surface, it can automatically resolve the curvature of the surface in accordance with the clustering requirement. It is clearly not possible to prescribe such a decay function manually that is required with the conventional elliptic grid generation schemes.

As shown in Fig. 7(c) and (d), although the grid is not orthogonal near the boundary, the nondegeneracy of the grid is ensured here indirectly in the sense that only the non-degenerate solutions will converge in the present method. An example of nondegeneracy is that Jacobians become negative. But, in a two-dimensional problem, the boundary constraints will physically yield the correct solution to the long thin fin heat transfer problem near a boundary, and hence a non-degenerate solution to the grid problem is guaranteed in the 2D case. The author has not yet identified any known physical principle on which the 3D grid problem is based, and hence nondegeneracy is not known to be guaranteed. But, in practice, a converged solution always yields a non-degenerate solution with the present method, as long as the initial grid is not degenerate. For a



(a) Tension cone IAD grid

(b) enlarged closer to the IAD

(c) further enlarged closer to the IAD

(d) IAD near-body grid

Fig. 7. Cross-section of the elliptic grid around the IAD; outer boundary is far-field.

detailed discussion on this topic, the reader is also referred to [23–26]. Nondegeneracy conditions for a hexahedral cell and grid have been discussed in detail by Ushakova [23–25] and Azarenok [26].

As an example, a low speed turbulent flow calculation at a Reynolds number of 2×10^7 was carried out with the $k - \varepsilon$ turbulence solver [27–29], and the corresponding pressure contours over the aeroshell are shown in Fig. 8. The flow solver used in the present study was originally used in an earlier study [30].

Another grid calculation was carried out for the MSL disk-band-gap canopy, and the grid over the canopy geometry is shown in Fig. 9. Again, the grid shown in Fig. 9 represents a single-block grid. The grid at the "vent" and the gaps is created as if it was a solid surface. The presence of the vent and the gaps is simulated through the CFD boundary condition prescription. There is an axis of singularity running through the center of the vent all the way from the upstream boundary to the downstream boundary. Pressure contours over the canopy in low speed laminar flow are shown in Fig. 10.

Next, the IAD grid was used to calculate supersonic laminar flow at Mach number of 2 and Reynolds number of 5×10^5, using a recently developed explicit high-order flow solver, EDLFLOW-F [31], which is fourth order accurate in time and sixth order accurate in space. The temporal integration is achieved using Runge–Kutta method and the spatial integration is

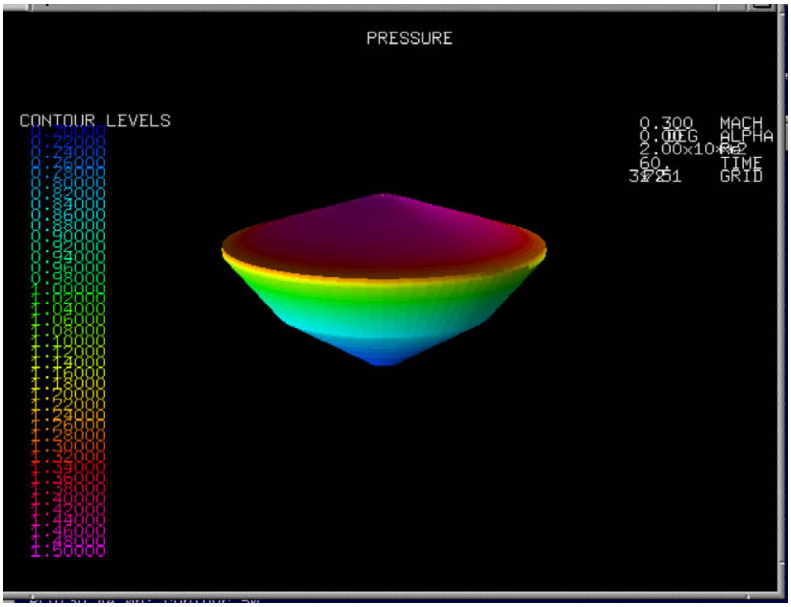


Fig. 8. Axisymmetric pressure contours about the aeroshell in low speed turbulent flow, Re = 2×10^7.

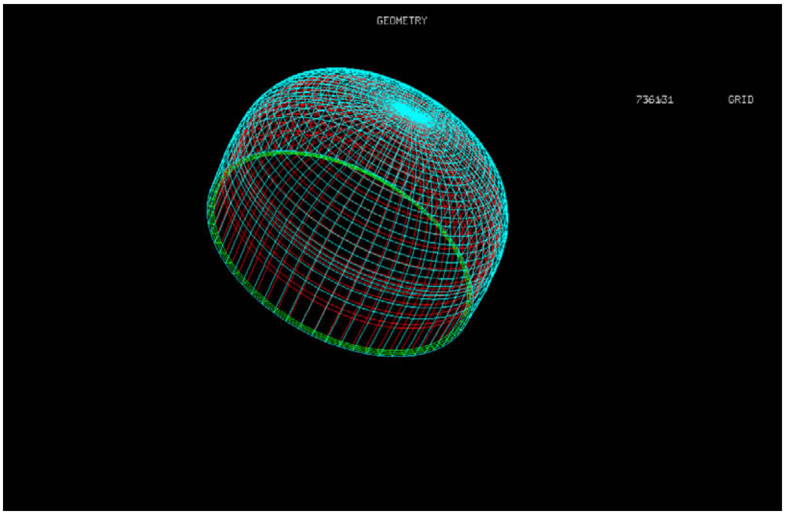


Fig. 9. A continuous single-block grid over the MSL disk-band-gap canopy geometry.

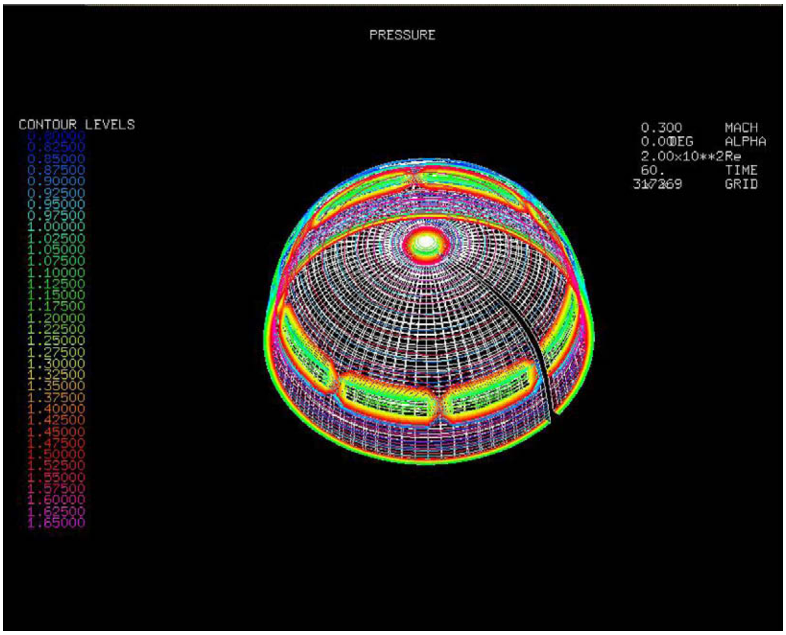


Fig. 10. Pressure contours over the MSL canopy in low speed laminar flow.

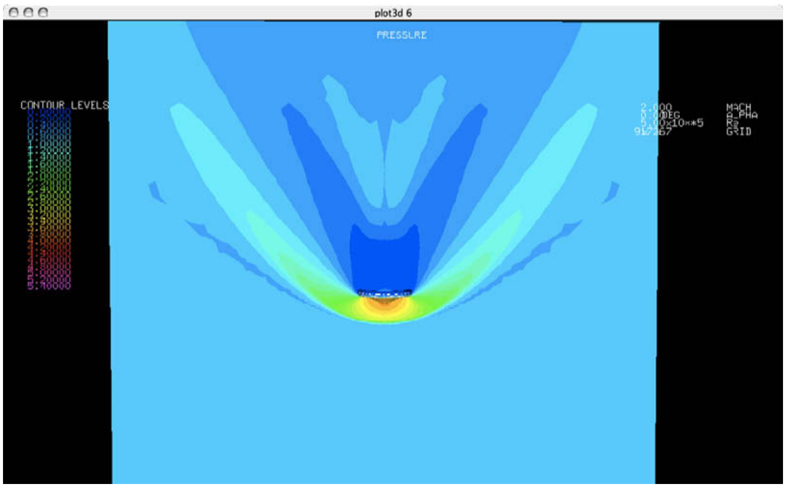


Fig. 11. Pressure contours over the Tension Cone IAD for free stream $M = 2$ and $\mathrm{Re} = 5 \times 10^5$ (laminar flow).

achieved using compact differencing scheme [32]. The solver, EDLFLOW-F, has been developed at NASA Ames recently to study highly dynamic planetary Entry, Descent and Landing (EDL) flows. The shock capturing is based on hyperviscosity approach [33]. An eighth order linear spatial filtering [34] is used to filter out the high frequency components of the solution, and the geometric conservation law and conservative metrics [35] are used to preserve free stream flow in the far-field. The CFL stability is enhanced by adding a sixth order dissipation designed for EDLFLOW-F [31], and the metrics are also differenced with the same high-order stencil as the flow variables.

The tension cone IAD results in terms of steady state Mach number contours are shown in Fig. 11, where a crisp capturing of the bow shock is shown. As shown in Fig. 11, conservative metrics [35] result in the capture of free stream flow upstream of the bow shock smoothly. All the other regions of the flow, subsonic pocket between the bow shock and the IAD, sonic lines, the expansion region and the perfect symmetric wake are captured with high fidelity. As mentioned before, a clustered single-block 91 × 72 × 67 elliptic grid was used for this calculation, as shown in Fig. 7. The grid is fine enough to capture the shock accurately. It may be noted here that the high fidelity of the CFD solution [31] shown in Fig. 11 refers to the high-order temporal-spatial scheme used for the Navier–Stokes equations.

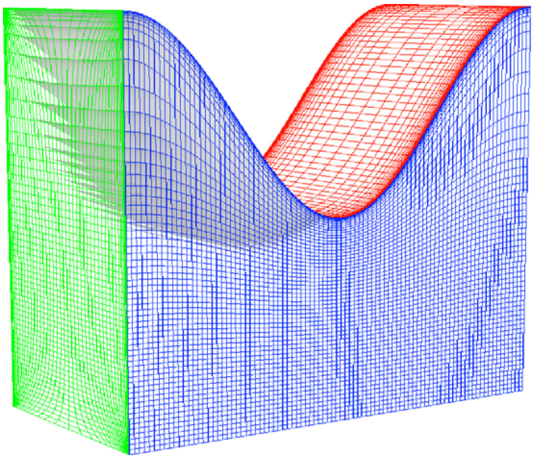


Fig. 12. A single-block $51 \times 91 \times 61$ grid for the convex geometry.

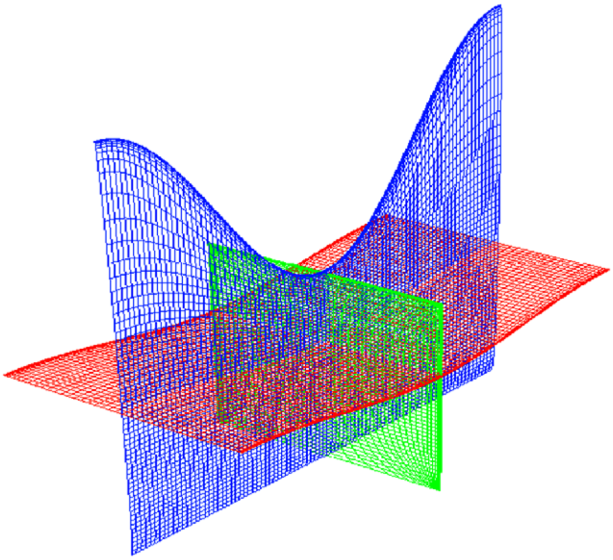


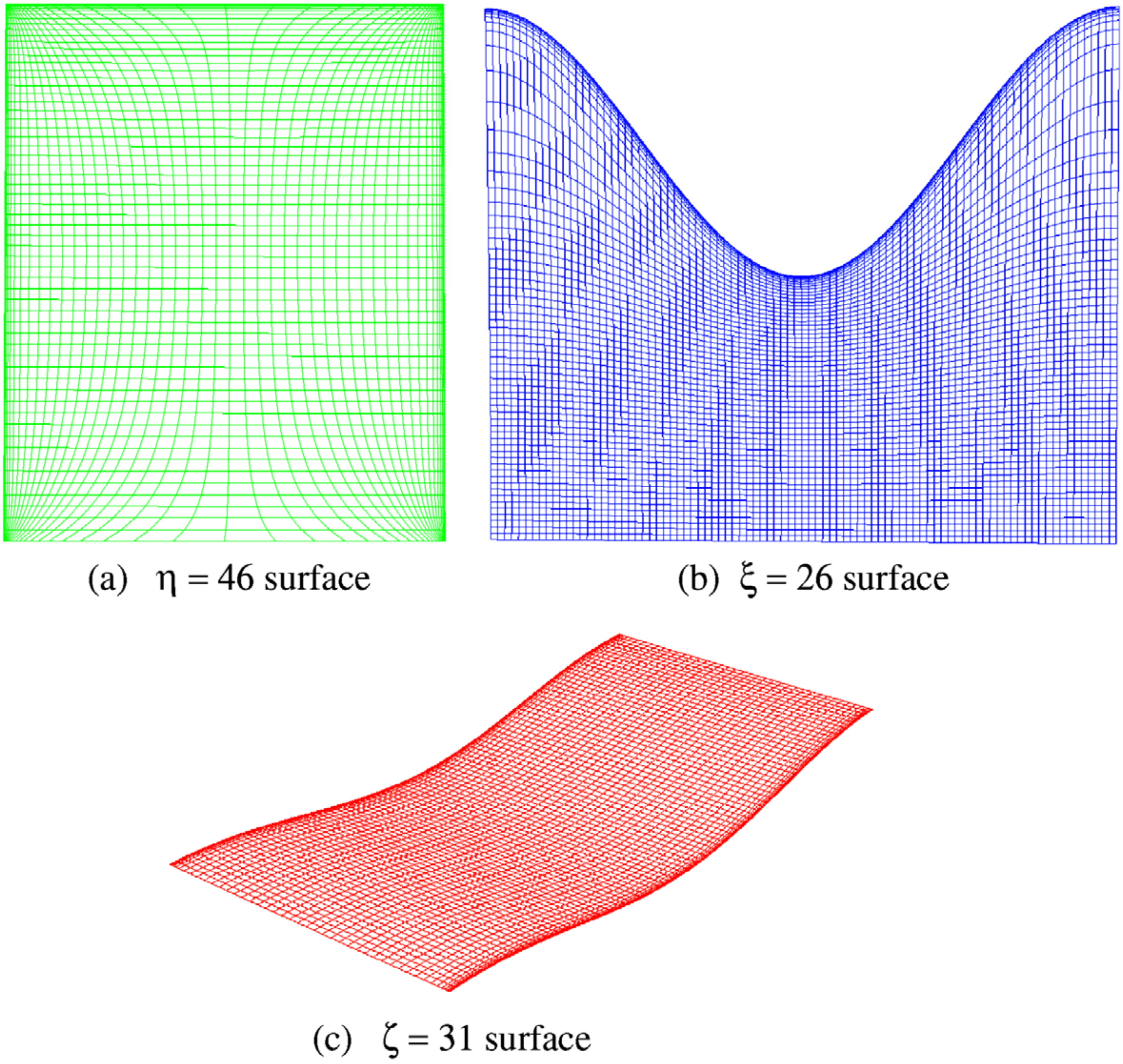
Fig. 13. Three orthogonal surfaces ($\xi = 26, \eta = 46, \zeta = 31$) slicing through the middle of the geometry.

Finally, an example of internal grid generation was considered. The geometry for this application is analytically defined and is shown in Fig. 12. The top $z(\zeta)$ boundary is convex and the 3D geometry is given by the following equation:

$$Z = 0.75 + 0.25\sin(\pi(0.5 - 2y)); \quad -1 \leqslant y < 0; \quad 0 \leqslant x \leqslant 0.5.$$

This geometry represents a similar challenge for the elliptic grid generation method as the aeroshell in a wind-tunnel since clustering occurs in two orthogonal directions, $x(\xi)$ and $z(\zeta)$, and the boundary constraints need to be satisfied simultaneously at the x–z corners. Fig. 13 shows three orthogonal surfaces ($\xi = 26$, $\eta = 46$, $\zeta = 31$) slicing somewhere through the middle of this geometry. Neumann boundary conditions are used in the ξ and η directions. Boundary constraints are used in the ξ directions well as in the ζ direction. It is worth noting that in the ξ direction, both the constraints and the Neumann boundary conditions are prescribed. Dirichlet boundary conditions are enforced at the ζ boundaries. A single-block $51 \times 91 \times 61$ grid was used for this geometry.

Detailed views of these orthogonal surface grids are shown individually in Fig. 14(a)–(c). Fig. 14(a) shows the $\xi - \zeta$ grid, Fig. 14(b) shows the $\eta - \zeta$ grid, and Fig. 14(c) shows the $\eta - \xi$ grid. The grids satisfy the smoothness requirement all over the domain. The clustering requirements are met at the $\zeta = 61$, $\xi = 1$ and $\xi = 51$ boundaries. As discussed before, the orthogonality property is satisfied approximately. But, orthogonality, being a function of how far the solution is driven to convergence and the initial rate of clustering, as discussed before, can be enforced reasonably satisfactorily for all practical purposes.


(a) $\eta = 46$ surface

(b) $\xi = 26$ surface

(c) $\zeta = 31$ surface

Fig. 14. Three surface grids corresponding to the orthogonal surfaces slicing through the middle of the convex geometry.

5. Conclusions

A fully automated three-dimensional elliptic grid generation method has been developed and demonstrated for the three-dimensional MSL aeroshell and canopy geometries as well as the tension cone IAD. The present method has also been demonstrated for a three-dimensional convex geometry defined analytically. Clustered body-orthogonal grids for the MSL aeroshell in external flow, aeroshell in a wind-tunnel, the canopy, the IAD and the convex geometry were generated automatically without any user intervention to enforce the clustering properties. Representative low speed and supersonic flows were simulated, using these grids. This makes the new fully automatic three-dimensional patented enhanced elliptic grid generator, EEGG3D, a powerful tool for generation of volume grids around complex geometries, especially deforming grids about deploying geometries such as MSL canopies and the Inflatable Aerodynamic Decelerators (IAD).

Acknowledgments

This work was supported by the Supersonics Project of the NASA Fundamental Aeronautics Program. The author thanks Ed Schairer of NASA Ames Research Center for providing the analytical geometry definition for the MSL aeroshell and the canopy and Ian Clark of Georgia Institute of Technology for providing the geometry definition for the Inflatable Aerodynamic Decelerator. The author would also like to thank the reviewers for their helpful comments and suggestions.

References

[1] W.P. Crowley, University of California, Lawrence Radiation Laboratory, Internal Report, Livermore, California, 1962.
[2] A. Winslow, Numerical solution of the quasi-linear Poisson equation, J. Comput. Phys. 1 (1966) 149–172.
[3] J.F. Thompson, F.C. Thames, C.W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate systems for fields containing any number of arbitrary two-dimensional bodies, J. Comput. Phys. 15 (1974) 299–319.
[4] J.F. Thompson, F.C. Thames, C.W. Mastin, TOMCAT – a code for numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number of arbitrary two-dimensional bodies, J. Comput. Phys. 24 (1977) 274–302.
[5] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, Numerical Grid Generation, Elsevier, New York, 1985.
[6] J.L. Steger, R.L. Sorenson, Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations, J. Comput. Phys. 33 (1979) 405–410.
[7] S.J. Alter, F.M. Cheatwood, Elliptic Volume Grid Generation for Viscous CFD Parametric Design Studies, AIAA Paper 96-1999.
[8] U.K. Kaul, New boundary constraints for elliptic systems used in grid generation problems, J. Comput. Phys. 189 (2003) 476–492.
[9] U.K. Kaul, Fully Automated Single-Zone Elliptic Grid Generation for Mars Science Laboratory (MSL) Aeroshell and Canopy Geometries, NASA-TM-2008-215358, 2008.
[10] U.K. Kaul, D.S. Chaussee, A comparative study of the parabolized Navier–Stokes (PNS) code using various grid generation techniques, Int. J. Comput. Fluids 13 (4) (1985) 421–441.
[11] B. Azarenok, A variational hexahedral grid generator with control metric, J. Comput. Phys. 218 (2) (2006) 720–747.
[12] G. Hansen, A. Zardecki, D. Greening, R. Bos, A finite element method for three-dimensional unstructured grid smoothing, J. Comput. Phys. 202 (1) (2005) 281–297.
[13] V. Villamizar, O. Rojas, J. Mabey, Generation of curvilinear coordinates on multiply connected regions with boundary singularities, J. Comput. Phys. 223 (2) (2007) 571–588.
[14] S. Bourdet, Analyse physique d'écoulements compressibles instationnaires autour de structures portantes dans le contexte d'interaction fluide-structure, Institut National Polytechnique de Toulouse, 2005.
[15] Y.N. Jeng, W.J. Kuo, Two-dimensional elliptic grid solver using boundary grid control and curvature correction, AIAA J. 38 (2) (2000) 217–224.
[16] M. Visbal, D. Knight, Generation of orthogonal and nearly orthogonal coordinates with grid control near boundaries, AIAA J. 20 (3) (1982) 305–306.
[17] G. Ryskin, L.G. Leal, Orthogonal mapping, J. Comput. Phys. 50 (1983) 71–100.
[18] V. Akcelik, B. Jaramaz, O. Ghattas, Nearly orthogonal two-dimensional grid generation with aspect ratio control, J. Comput. Phys. 171 (2001) 805–821.
[19] L. Eca, 2D orthogonal grid generation with boundary point distribution control, J. Comput. Phys. 125 (1996) 440–453.
[20] I.G. Clark, A.L. Hutchings, C.L. Tanner, R.D. Braun, Supersonic inflatable aerodynamic decelerators for use on future robotic missions to Mars, J. Spacecraft Rockets 46 (2) (2009) 340–352.
[21] J.H. Bell, Transonic/Supersonic Wind Tunnel Testing of the NASA Orion Command Module, AIAA Paper 2007-1006.
[22] J.C. Ross, Aerodynamic Testing in Support of Orion Spacecraft Development, AIAA Paper 2007-1004.
[23] O.V. Ushakova, Conditions of nondegeneracy of three-dimensional cells. A formula of a volume of cells, SIAM J. Sci. Comput. 23 (4) (2001) 1274–1290.
[24] O.V. Ushakova, Nondegeneracy conditions for different types of grids, in: O.V. Ushakova (Ed.), Advances in Grid Generation, Nova Science Publishers, New York, 2005 (Chapter 9).
[25] O.V. Ushakova, Classification of hexahedral cells, Comput. Math. Math. Phys. 48 (8) (2008) 1327–1348.
[26] B.N. Azarenok, A method of constructing adaptive hexahedral moving grids, J. Comput. Phys. 226 (1) (2007) 1102–1121.
[27] U.K. Kaul, An Implicit Finite Difference Code for a Two-Equation Turbulence Model for 3D Flows, NASA TM-85241, 1985.
[28] U.K. Kaul, A fully implicit numerical scheme for the solution of transport equations in fluid dynamics, in: C. Taylor, J.H. Chin, G.M. Homsy (Eds.), Numerical Methods in Laminar and Turbulent Flow, vol. 2, Pineridge Press, Swansea, U.K., 1991, pp. 1724–1733.
[29] U.K. Kaul, SUPKEM: A Fully Implicit Hyperbolic/Parabolic Partial Differential Equation Solver for the Integration of Unsteady Three-Dimensional Turbulence Kinetic Energy and Dissipation Rate Equations, NASA Invention Disclosure, NASA-ARC-16260-1, 2008.
[30] U.K. Kaul, R.P. Shreeve, A fully viscous Navier–Stokes formulation in generalized curvilinear coordinates for prediction of internal turbine blade-cooling flows, AIAA J. Thermophys. Heat Transfer 10 (4) (1996) 621–626.
[31] U.K. Kaul, EDLFLOW-F: A Next-Generation High-Order High-Fidelity All-Speed Time-Accurate Flow Solver for Simulating Fluid Flows, NASA Invention Disclosure, NASA-ARC-16349-1, 2009.
[32] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.
[33] A.W. Cook, W.H. Cabot, Hyperviscosity for shock–turbulence interactions, J. Comput. Phys. 203 (2005) 379–385.
[34] D.V. Gaitonde, J.S. Shang, J.L. Young, Practical aspects of higher-order numerical schemes for wave propagation phenomena, Int. J. Numer. Methods Eng. 45 (1999) 1849–1869.
[35] P.D. Thomas, C.K. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA J. 17 (10) (1979) 1030–1037.