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

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. The new procedure also enables generation of single-zone 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 zones for grid generation. The new boundary constraints are not limited to the boundaries only, but can also be formulated around isolated pockets 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.

---

*Associate Fellow, AIAA

\(^1\)U.S. patent 7231329
I. 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 integrated 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 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 higher order quantities such as skin friction and heat transfer. 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.

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 2-D study\(^8\), proposed an automatic procedure that obviated the need for the external specification of these decay parameters. In this 2-D study\(^8\), these decay parameters were no longer the prescribed constants, but functions of a single generalized coordinate variable, say, \(\eta\), over a given generalized coordinate boundary, say \(\zeta = \text{constant}\). 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., \(\xi\) direction are free to adapt from point to point over a given \((\eta, \zeta)\) surface.

II. 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, 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, 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 is given. Then, the development of the semi-automatic and the enhanced fully automated methodologies is briefly discussed. Finally, the extension of the fully automated methodology for two-dimensions is extended to three-dimensional applications.

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

\[
\begin{align*}
\xi_{xx} + \xi_{yy} + \xi_{zz} &= P(\xi, \eta, \zeta) = -a_{i} \text{sgn}(\xi - \xi_{i}) \exp(-b_{i}|\xi - \xi_{i}|), \\
\eta_{xx} + \eta_{yy} + \eta_{zz} &= Q(\xi, \eta, \zeta) = -c_{i} \text{sgn}(\eta - \eta_{i}) \exp(-d_{i}|\eta - \eta_{i}|), \\
\zeta_{xx} + \zeta_{yy} + \zeta_{zz} &= R(\xi, \eta, \zeta) = -e_{i} \text{sgn}(\zeta - \zeta_{i}) \exp(-f_{i}|\zeta - \zeta_{i}|),
\end{align*}
\]

where \(\xi, \eta\) and \(\zeta\) 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 Equations (1-3) is written as

\[
\begin{align*}
\xi_{xx} + \xi_{yy} &= -a_{i} \text{sgn}(\eta - \eta_{i}) \exp(-d_{i}|\eta - \eta_{i}|), \\
\eta_{xx} + \eta_{yy} &= -c_{i} \text{sgn}(\eta - \eta_{i}) \exp(-d_{i}|\eta - \eta_{i}|),
\end{align*}
\]

Equations (4) and (5) were used to semi-automatically generate the two-dimensional grids with appropriate clustering and orthogonality at the walls. But, the decay parameter, \(d_{i}\), was prescribed manually, for a given boundary, \(\eta_{i}\).

As mentioned above, a fully automatic boundary procedure was proposed 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.

In this paper, extension of this 2-D 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-zone grids for these geometries.

Using the fully automated approach, for a given boundary, \( \zeta_i (\zeta \neq \zeta_i) \), for example, Equations (1), (2) and (3) are modified here in the context of equations (4) and (5) and are written in the following form.

\[ \xi_{xx} + \xi_{yy} + \xi_{zz} = p_3(\xi, \eta, \zeta) \quad (6) \]

where, \( p_3(\xi, \eta, \zeta) = -a_{3,1}(\xi, \eta) \sgn(\zeta - \zeta_i) \exp\{- f_i(\xi, \eta)|\zeta - \zeta_i|\} \)
\[ \approx (-a_{3,1}(\xi, \eta) + a_{3,1}(\xi, \eta) f_i(\xi, \eta) (\zeta - \zeta_i)) \sgn(\zeta - \zeta_i) \]
\[ \eta_{xx} + \eta_{yy} + \eta_{zz} = q_3(\xi, \eta, \zeta), \quad (7) \]

where, \( q_3(\xi, \eta, \zeta) = -c_{3,1}(\xi, \eta) \sgn(\zeta - \zeta_i) \exp\{- f_i(\xi, \eta)|\zeta - \zeta_i|\} \)
\[ \approx (-c_{3,1}(\xi, \eta) + c_{3,1}(\xi, \eta) f_i(\xi, \eta) (\zeta - \zeta_i)) \sgn(\zeta - \zeta_i) \]
\[ \zeta_{xx} + \zeta_{yy} + \zeta_{zz} = r_3(\xi, \eta, \zeta), \quad (8) \]

where, \( r_3(\xi, \eta, \zeta) = -e_{3,1}(\xi, \eta) \sgn(\zeta - \zeta_i) \exp\{- f_i(\xi, \eta)|\zeta - \zeta_i|\} \)
\[ \approx (-e_{3,1}(\xi, \eta) + e_{3,1}(\xi, \eta) f_i(\xi, \eta) (\zeta - \zeta_i)) \sgn(\zeta - \zeta_i) \]
and, where \( a_{k, i} = a_{k, i}(\xi, \eta), \ c_{k, i} = c_{k, i}(\xi, \eta), \) and \( e_{k, i} = e_{k, i}(\xi, \eta) \) and where \( k = 3 \) corresponds to the \( \zeta \) boundary, under consideration.

Similar expressions hold for the inhomogeneous terms for the \( \xi \) (k=1) and \( \eta \) (k=2) boundaries.

The positive decay parameters, \( b_i, d_i, \) and \( f_i, \) for the corresponding boundaries, \( \xi, \eta \) and \( \zeta \), 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, \( a_{2,1}(\xi, \zeta), \ c_{2,1}(\xi, \zeta), \ e_{2,1}(\xi, \zeta) \) and \( a_{1,1}(\eta, \zeta), \ c_{1,1}(\eta, \zeta), \ e_{1,1}(\eta, \zeta) \), hold for \( \eta \) and \( \xi \) boundaries respectively.

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

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

Rewriting, the limiting governing equations near a \( \zeta \) boundary become:

\[
\begin{align*}
\xi_{xx} + \xi_{yy} + \xi_{zz} - a_{3,i}(\xi, \eta) f(\xi, \eta) (\xi - \zeta) &= -a_{3,i}(\xi, \eta) \\
\eta_{xx} + \eta_{yy} + \eta_{zz} - c_{3,i}(\xi, \eta) f(\xi, \eta) (\eta - \zeta) &= -c_{3,i}(\xi, \eta) \\
\zeta_{xx} + \zeta_{yy} + \zeta_{zz} - e_{3,i}(\xi, \eta) f(\xi, \eta) (\zeta - \zeta) &= -e_{3,i}(\xi, \eta)
\end{align*}
\]

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

\[
L(\theta) = \text{div}(k \text{grad}(\theta)) - q\theta
\]

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 \( \xi \) boundary, the constraint will be given by

\[
\int_S (\partial\theta/\partial n) \, d\sigma = \int_V \{ -e_{3,i}(\xi, \eta) \, \text{sgn}(\zeta - \zeta_i) + e_{3,i}(\xi, \eta) f(\xi, \eta) \theta \} \, d\tau , \quad (9)
\]

where \( \theta = (\xi, \eta, \zeta) \), \( d\sigma \) is a differential area element, \( d\tau \) is a differential volume element, \( n \) 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, \( e_{3,i}(\xi, \eta) f(\xi, \eta) \).

Similar constraints can be used to calculate the decay parameter analogs, \( a_{1,i}(\eta, \zeta)b_i(\eta, \zeta) \), and \( c_{2,i}(\eta, \zeta)d_i(\zeta, \zeta) \). When expressed in terms of the generalized coordinate, \( \zeta \), the boundary constraint given in Eq. (9) is transformed as follows.

\[
\int_S I \, d\sigma = \int_S (\partial\theta/\partial n) \, d\sigma = \int_S (\partial\zeta/\partial n) \, d\sigma \quad (10)
\]

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:

\[
\int_S I \, d\sigma = \int_{\xi_{\text{max}}} I \, d\sigma + \int_{\eta_{\text{max}}} I \, d\sigma + \int_{\zeta_{\text{max}}} I \, d\sigma \\
- \int_{\xi_{\text{min}}} I \, d\sigma - \int_{\eta_{\text{min}}} I \, d\sigma - \int_{\zeta_{\text{min}}} I \, d\sigma \quad (11)
\]

where the surface configurations \( \xi_{\text{max}}, \xi_{\text{min}}, \) 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.

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

\[
\int_\eta I \, d\sigma = \int_\eta \left(1/\sqrt{a_{22}}\right) \alpha_{23} \left[(x_\xi^2 + y_\xi^2 + z_\xi^2) (x_\eta^2 + y_\eta^2 + z_\eta^2) \right]^{1/2} \, d\xi \, d\zeta . \quad (13)
\]
\[ \int I d\sigma = \int (1/J) [\alpha_{33} (x^2 + y^2 + z^2) (x^2 + y^2 + z^2)]^{1/2} d\eta d\xi, \] (14)

\[ \begin{align*}
\alpha_{11} &= J^2 (\xi_x^2 + \xi_y^2 + \xi_z^2), \\
\alpha_{22} &= J^2 (\eta_x^2 + \eta_y^2 + \eta_z^2), \\
\alpha_{33} &= J^2 (\zeta_x^2 + \zeta_y^2 + \zeta_z^2), \\
\alpha_{12} &= J^2 (\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z), \\
\alpha_{13} &= J^2 (\xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z), \\
\alpha_{23} &= J^2 (\eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z),
\end{align*} \]

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

Equations (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.

\[ \begin{align*}
\alpha_{11} x_{i,\xi} &+ \alpha_{22} x_{i,\eta} + \alpha_{33} x_{i,\zeta} + 2 \{ \alpha_{12} x_{i,\xi\eta} + \alpha_{13} x_{i,\xi\zeta} + \alpha_{23} x_{i,\eta\zeta} \} = \\
&= -J^{-2} \{ p_3 x_{i,\xi} + q_3 x_{i,\eta} + r_3 x_{i,\zeta} \}, \\
x_i &= x, y \text{ or } z.
\end{align*} \]

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, \((\xi,\eta)\), 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.

### III. 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 studies, a direct comparison with experimental data will be made for these geometries. There are currently some parallel experimental studies\textsuperscript{11,12} being carried out on similar geometries. For the MSL and the IAD geometries, single zone grids are generated with the present grid generation procedure which are uniformly clustered at the body. 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 zones and the grids are generated separately for these zones.

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 zone 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-zone 51x72x61 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, 51x61, of the volume grid around the aeroshell is shown in Fig. 2(a,b). This grid is shown to be clustered at the aeroshell wall. Away from the aeroshell, the grid uniformly
(a) clustering around the aeroshell wall

(b) enlarged close to the aeroshell

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

stretches to the far-field boundary. No clustering requirement is enforced at the far-field boundary. As shown, the grid generated is orthogonal and uniformly clustered at the aeroshell surface.
Fig. 3(a,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. A three-dimensional slice through the volume grid including the aeroshell surface is shown in Fig. 4.
Fig. 4 A three-dimensional slice of the grid in the vicinity of the aeroshell

Fig. 5(a,b) shows a single-zone 91x72x67 grid for the tension cone IAD; Fig. 5(b) shows an enlarged view 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.

Fig. 5 Cross-section of the elliptic grid around the IAD; outer boundary is far-field
As an example, a low speed turbulent flow calculation at a Reynolds number of $2 \times 10^7$ was carried out with the $k-\varepsilon$ turbulence solver\textsuperscript{13-15}, and the corresponding pressure contours over the aeroshell are shown in Fig. 6. The flow solver used in the present study was originally used in an earlier study\textsuperscript{16}.

Another grid calculation was carried out for the MSL disk-band-gap canopy, and the grid over the canopy geometry is shown in Fig. 7. Again, the grid shown in Fig. 7 represents a single-zone grid. Pressure contours over the canopy in low speed laminar flow are shown in Fig. 8.

Finally, the IAD grid was used to calculate supersonic laminar flow at Mach number of 2 and Reynolds number of $5 \times 10^5$, using a recently developed explicit high order flow solver, EDLFLOW-F\textsuperscript{17}, which is 4th order accurate in time and 6th order accurate in space. The temporal integration is achieved using Runge-Kutta method and spatial integration is achieved using compact differencing scheme\textsuperscript{18}. 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\textsuperscript{19}. An 8th order linear spatial filtering\textsuperscript{20} is used to filter out the high frequency components of the solution, and the geometric conservation law and conservative metrics\textsuperscript{21} are used to preserve free stream flow in the far field. The CFL stability is enhanced by adding a 6th order dissipation designed for EDLFLOW-F\textsuperscript{17}, and the metrics are also differenced with the same high order stencil as the flow variables.
Fig. 7  A continuous single-zone grid over the MSL disk-band-gap canopy geometry

Fig. 8  Pressure contours over the MSL canopy in low speed laminar flow
The tension cone IAD results in terms of steady state Mach number contours are shown in Fig. 9, where a crisp capturing of the bow shock is shown. As shown in Fig. 9, conservative metrics\textsuperscript{21} 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-zone 91x72x67 elliptic grid was used for this calculation, as shown in Fig. 5.

![Mach number contours over the Tension Cone IAD](image.jpg)

**Fig. 9** Mach number contours over the Tension Cone IAD for free stream $M=2$ and $Re=5 \times 10^5$ (laminar flow)

**IV. 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. Clustered body-orthogonal grids for the MSL aeroshell in external flow, aeroshell in a wind-tunnel, the canopy and the IAD 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 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).

**V. Acknowledgement**

This work was supported by the Supersonics Project of the NASA Fundamental Aeronautics Program. The author would like to thank Ed Schairer of NASA Ames Research Center for providing the analytical geometry definition for the MSL aeroshell and the canopy.
VI. References

1. Crowley, W. P., University of California Lawrence Radiation Laboratory Internal Report, 1962, Livermore, California


