



AIAA 97–1987

Complex Volume Grid Generation Through the Use of Grid Reusability

Stephen J. Alter
*Lockheed Martin Engineering & Sciences,
Hampton, Virginia 23681*

13th AIAA Computational Fluid Dynamics Conference
June 29–July 2, 1997/Snowmass, CO

Complex Volume Grid Generation Through the Use of Grid Reusability

Stephen J. Alter*

Lockheed Martin Engineering & Sciences, Hampton, Virginia 23681

This paper presents a set of surface and volume grid generation techniques which reuse existing surface and volume grids. These methods use combinations of data manipulations to reduce grid generation time, improve grid characteristics, and increase the capabilities of existing domain discretization software. The manipulation techniques utilize physical and computational domains to produce basis function on which to operate and modify grid character and smooth grids using Trans-Finite Interpolation, a vector interpolation method and parametric re-mapping technique. With these new techniques, inviscid grids can be converted to viscous grids, multiple zone grid adaption can be performed to improve CFD solver efficiency, and topological changes to improve modeling of flow fields can be done simply and quickly. Examples of these capabilities are illustrated as applied to various configurations.

Nomenclature

I, ξ	streamwise computational direction measured from nose to tail of body
J, η	circumferential computational direction measured from top to bottom of body
K, ζ	computational direction normal to body surface
$\Delta S_{i,j}$	distance between points (i, j, k) and $(i, j - 1, k)$
X, Y, Z	Cartesian coordinates

Introduction

MUCH emphasis in the Computational Fluid Dynamics (CFD) arena is placed on the identification of viscous effects on evolving designs of complex configurations across the speed range. The analysis of viscous flow fields is a necessity in the detailed analysis phase of vehicle design. The necessity arises from the ability of CFD to predict flow field characteristics in regimes that are not attainable through experimental techniques, as well as the verification of experimental results.¹

The use of viscous computations in the various phases of vehicle design have been hampered by the time required to obtain volume grids that can accurately capture boundary layer effects and entire flow domains. The reduction of excessive time required to generate volume grids for viscous computations (i.e. viscous volume grids), has been a primary thrust of researchers for nearly a decade.^{2,3} Techniques used in recent years have placed emphasis on the development of grid adaption to capture viscous related gradients,⁴⁻⁶ as opposed to initial development of viscous volume grids.

Generation of structured volume grids to perform

viscous computations is difficult if not impossible, using state-of-the-art domain discretization tools. The difficulty arises from the need to tightly pack grid points near the wall of a configuration to accurately capture and model the boundary layer.⁷ There are several popular methods available to generate volume grids with specified grid point spacings and orthogonal grid-line incidence, including:

- Elliptic partial differential equation (PDE) solvers;⁸⁻¹¹
- Hyperbolic PDE solvers;¹²
- Parabolic PDE solvers;¹³ and
- Algebraic solvers;¹⁴

Each of these schemes has its own set of advantages and disadvantages. A common challenge to each of the methods is the efficient control of grid spacing and grid-line incidence at the domain boundaries. The hyperbolic methods offer the best control of point spacings and orthogonality, but do not provide for the control of outer boundary shape and can frequently produce less than favorable results at the outer boundary.¹⁵ Elliptic solvers can provide grid control as well as fast generation of inviscid volume grids, but the equations become stiff for highly clustered grids. To retain the control provided by elliptic solvers and be able to generate viscous grids, conversion of inviscid grids to viscous grids with appropriate point spacings and orthogonality, could be performed.

Current methods of generating volume grids for CFD use the solution of algebraic and partial differential equations. These methods are usually contained within graphical user interfaces, to ease the determination of control variables as they affect the overall grid being built. One widely used approach utilizes various algebraic techniques to generate an initial surface grid. The surface grid is usually smoothed through the

*Senior Aeronautical Engineer, Senior Member AIAA

Copyright © 1997 Stephen J. Alter. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

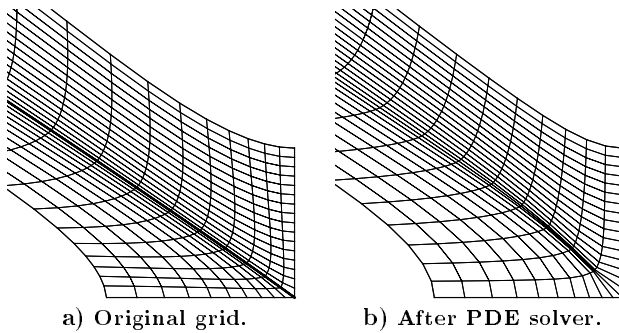


Fig. 1 Points forced out of a concave region.

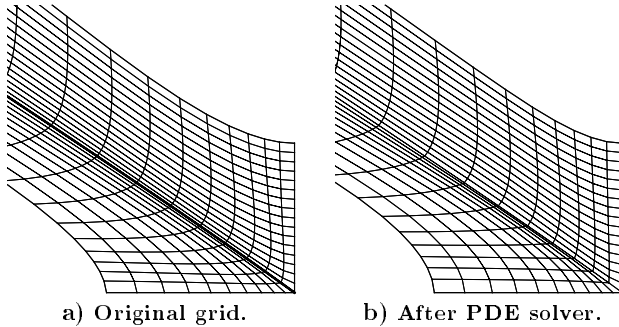


Fig. 2 Discontinuous cell spacings at a surface interface.

solution of a non-linear Poisson's equation, to obtain desirable grid line incidence character at a boundary, such as orthogonality or slope continuity at a matching interface.

When the solution of an entire block face produces undesirable grid point stretching or poor grid line incidence angles, the face can be subdivided into multiple regions, termed subfaces. Then, the algebraic and PDE equations can be used to further refine the grid by modifying specific regions of the block face. This method of using face decompositions to generate high quality grids in the two dimensional computational domain is common and very powerful, as very complex faces can be generated in this manner. However, this process is very time consuming and can produce undesirable results. The most common problem is grid points being forced out of concave regions and non-uniform grid point spacings between two subfaces, as shown in Fig. 1 and Fig. 2 respectively.

Volume grid generation is more complicated and the two dimensional approaches to solving point spacing, grid-line incidence and smoothness are not necessarily applicable. This paper presents a set of algebraic approaches to do volume grid refinements and offer grid reusability through sub-blocking techniques. This reduces the time to generate the defining faces of a three-dimensional (3D) block, as well as offers a technique to generate viscous volume grids from inviscid domain discretizations and grid adaption. These techniques use standard transfinite interpolation coupled

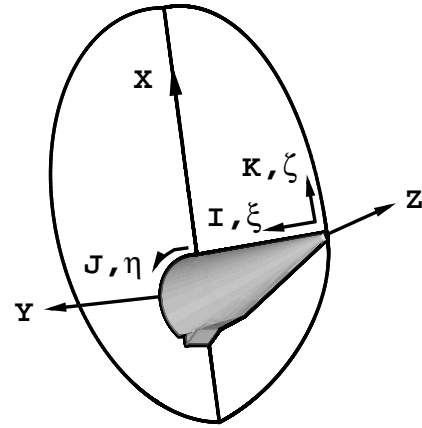


Fig. 3 Grid-point and computational coordinate orientations.

with copying and redistributing grid points along fixed computational directions through the use of parametric splines. These processes use simple assumptions to develop complex volume grid manipulations to do grid smoothing and ensure placement of grid point densities in regions of strong flow gradients. Manipulations of two and three dimensional grids will be shown as examples of each technique. All of the techniques discussed in this paper can be found in the Volume Grid Manipulation Language¹⁶ (VGM) tool. Each of the manipulations performed in this paper were done using an SGI R4400 CPU with 256Mb of memory, unless otherwise noted.

Computational Orientation

The coordinate system used in this paper is illustrated in Fig. 3. As shown, the I (or ξ)-coordinate increases from the nose to tail, the J (or η)-coordinate increases from the top to the bottom of the vehicle, and the K (or ζ)-coordinate increases from the geometry surface to the outer boundary. Throughout the explanation of techniques in this paper, this configuration will be used. It represents a sphere, cone, cut-flare with a windside bodyflap. The only complex face that will be addressed in the development of the inviscid grid will be the exit-plane at the maximum I location.

Domain Discretization Process

To generate an external volume grid about a vehicle, the wall or surface of the configuration must first be generated. This is usually accomplished using a Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) system that can model the surface upon which the grid is built as well as the control grid characteristics, such as point spacing and grid-line intersections. Then the rest of the flow domain about the vehicle is discretized by constructing a 3D domain that would encompass the flow field. In doing so, 6 boundary faces of a computational cube containing the 3D flow field are defined.

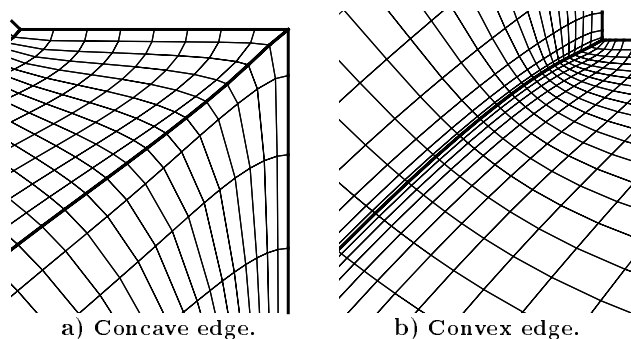


Fig. 4 Problems associated with concave and convex regions.

For 3D elliptic solvers, these faces are typically initialized with algebraic grid generators, and smoothed by solving Poisson's equation with source terms to control grid line incidence, intersections and point spacing.

Surface grids used to define the domain of a volume grid should contain no crossed grid lines, point-to-point stretchings of less than 1.5 and minimized grid line curvature to provide the best grid for CFD simulations.¹⁵ The generation process for simple 2D faces, such as the X-Z plane in Fig. 3, only requires the identification of boundary conditions at face edges to develop a usable surface grid. As the complexity of the surface increases, the 2D face can be discretized into sub-domains⁷ (i.e. subfaces), where each subface is elliptically smoothed with dependence on the adjacent subfaces.

The use of PDE solvers to develop block face or surface grids can result in quality characteristics that are not suitable for CFD purposes. One characteristic is non-uniform point spacings at an intersection of two subfaces. Illustrated in Fig. 2 is the result of using a slope continuity boundary condition at the interface of two subfaces, for the elliptic solver. Although the grid lines are slope continuous, the cell spacings are not. A second poor grid characteristic is the tendency of elliptic solvers to pull points away from concavities and push points onto convexities, thereby reducing and increasing grid resolution, respectively. The increasing of grid resolution on convexities is permissible in the realm of CFD but convexities also pose the problem of wrapping a grid around a 270 degree corner. Wrapping onto a convexity can produce poor cell spacings traversing the corner as shown in Fig. 4. Reducing grid density in concavities may not provide adequate resolution of flow field gradients and can produce an improper modeling of such a region.

To improve the grid-point spacings in concavities and on convexities, the regions can be re-subfaced to encompass the poor grid densities and point spacing discontinuity and re-solved. For example, to repack grid points into a concavity, two subfaces are constructed such that each share an edge that traverses from the corner of the concavity to somewhere on the

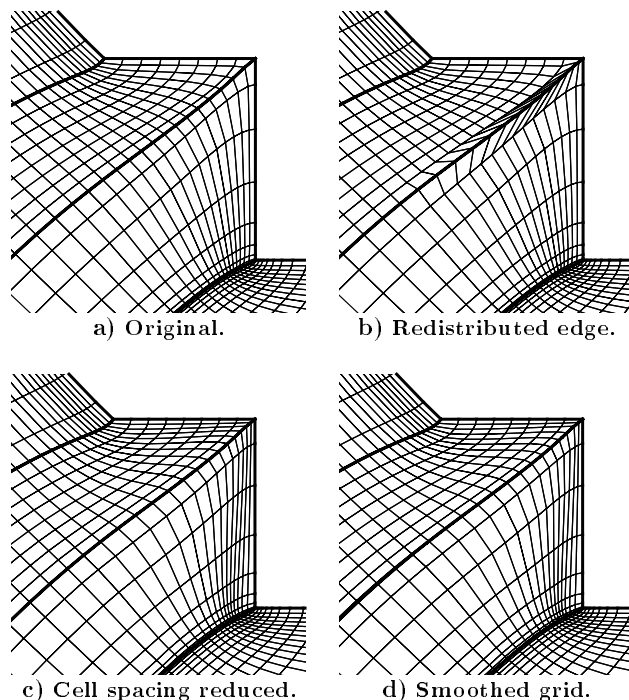


Fig. 5 Results of repacking grid points into a concave region.

interior of the surface grid, as shown in Fig. 5(a). Then, this edge is redistributed to place more points near the corner as shown in Fig. 5(b), and each subface is elliptically solved with slope continuity across the shared edge illustrated in Fig. 5(d). The solution usually produced grid-point spacing discontinuities at the shared edge, shown in Fig. 5(c). Then the region has to be re-subfaced to encompass the poor spacings and re-solved. Again, this may cause the points used to repack region, to pull away from concavity. Obtaining a suitable grid in this type of area often requires a compromise on grid point spacing and grid-point density. Generation of a usable surface grid in this region typically requires a minimum of four sets of two, redefined subfaces and can consume as much as 4 hours of user time on an SGI R4400 CPU. If a 3D block face has multiple concavities and convexities, the time required to generate the surface grid can be extensive. When three-dimensional solving is done, these subfacing and resubfacing techniques do not exist in a solver, which limits the effectiveness of the PDE grid solution as applied to CFD simulations.

Improved Process

Instead of utilizing the subface decomposition procedure previously described, several new techniques afforded by the VGM code will be used to alleviate the problems associated with elliptic smoothing of 2D faces in regions of concavities and convexities. The first method utilizes the computational domain of the 2D face by using the arclength parameter as a distribution function for manipulating the location of grid

points. This is done by computing the arclength function in a computational direction using equation 1:

$$\Delta S_{i,j} = \sqrt{\Delta x_{i,j}^2 + \Delta y_{i,j}^2 + \Delta z_{i,j}^2} \quad (1)$$

where,

$$\Delta x_{i,j} = x_{i,j} - x_{i,j-1}$$

$$\Delta y_{i,j} = y_{i,j} - y_{i,j-1}$$

$$\Delta z_{i,j} = z_{i,j} - z_{i,j-1}$$

if the direction of the arclength is J. The initial function, when mapped to the physical domain, locates the grid points in their original positions but creates a bridge between the coordinates. This will be referred to as the basis function. By changing the arclength distribution function, the grid point locations along the basis function will be changed. The arclength parameter is one dimensional, which offers a simple link to the three dimensional physical domain. By grouping a series of arclengths to form a region (i.e. zone) the new distributions used for the arclengths in a computational direction can be created with a single function or multiple dependent functions.

The poor grid-point spacing variances traversing the perimeter direction of the concavity, illustrated in Fig. 2, can be alleviated by simply redistributing a region that encompasses the poor spacings in the direction of such spacings produces the best results. To do this redistribution, basis functions are constructed from the grid lines to be smoothed, by generating an arclength parameter in the J-direction, as shown in Fig. 6. The arclength parameter S is monotonic in the direction computed, so interpolation along this line will not produce double values. By isolating the region to be smoothed, and computing the existing cell sizes at the ends of the region in the direction of the redistribution, Vinokur's¹⁷ function can be used to smooth the grid along the basis functions in the selected region. In this case, a single redistribution function is used to smooth a grid.

To correct the problem of grid point extraction from concave regions, as shown in Fig. 1, a combination of redistribution of the grid lines based on computational coordinates and interpolation of distribution functions from regions on each side of the redistribution, in the direction away from the concave region usually works best. To do this redistribution, the grid line that emanates from the corner and the adjacent grid lines are redistributed by using Vinokur's method with a specified cell size at the origin and the current cell size at a benign point on the interior, based on computational coordinates. By redistributing based on computational coordinates the cell sizes specified are percentages of the current values, which implies that a cell size of 1.0 is the current cell size. In this case, the eleventh point on the interior was chosen as

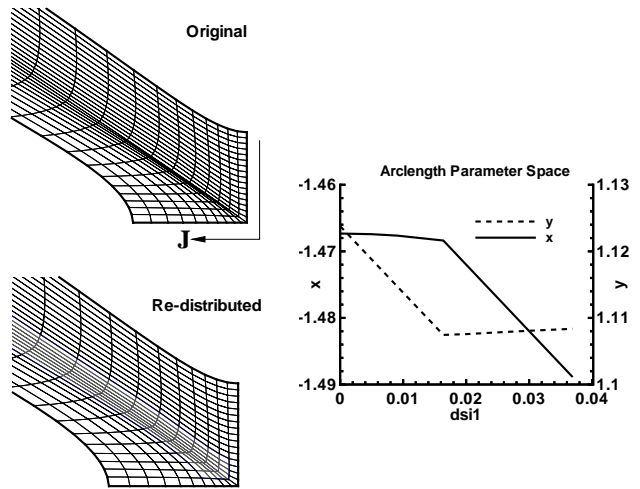


Fig. 6 Cell spacing discontinuities removed via Vinokur redistribution along arclength parameter in I-direction.

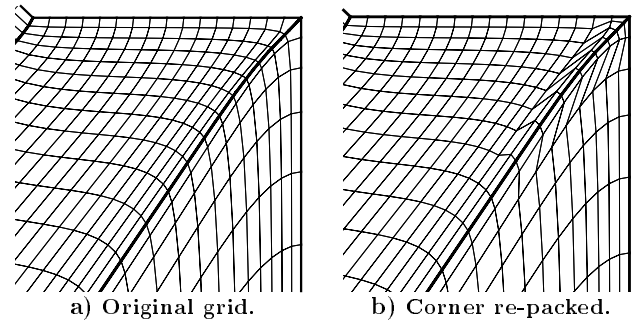


Fig. 7 Center grid lines at corner redistributed to pack points into concavity.

the stopping point for the redistribution. The resulting grid is shown in Fig. 7. Next, the grid lines on both sides of the redistributed grid line have to be manipulated to smooth out the kink produced by the redistribution. To do this smoothing, a region on both sides of the redistributed grid line is chosen such that a blending region can be made to transition the manipulated grid to the non-manipulated regions. To do this, a region of 11 points wide in the cross-sectional direction is identified, as shown in Fig. 8(a). The normalized arclength parameter for the identified region in the direction of the original redistribution is computed using equation 1 and 2:

$$\Delta S_{i,j} = \frac{\Delta S_{i,j}}{\sum_{i=1}^{I_{\max}} \Delta S_{i,j}} \quad (2)$$

where the direction of the arclength is I (see Fig. 8(b)). To smooth the grid, the beginning and ending arclength parameter functions are blended onto the region using an elliptic blending function¹⁸ The elliptic blend between the interface with the undisturbed grid

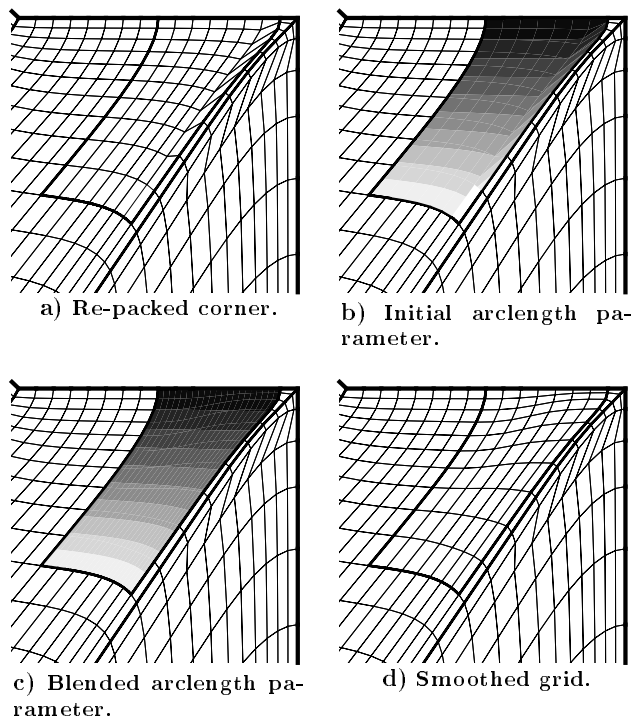


Fig. 8 Smoothed top region grid after packing points into the concavity.

and the redistributed grid line, is chosen because the resulting normalized arclength parameter will not deviate significantly near the end points. Utilizing the elliptic blending function produces a new normalized arclength parameter field shown in Fig. 8(c). Using this new blended arclength parameter as the basis function to redistribute the grid lines in the identified region to be smoothed, the grid becomes slope continuous at the beginning and nearly slope continuous at the end, as illustrated in Fig. 8(d). The slope continuity results from the continuity of the elliptic function used to blend the normalized arclength parameter. Performing the same type of redistribution on the other side of the originally redistributed grid line that emanated from the corner results in higher quality characteristic grid lines, as shown in Fig. 9. This process of grid smoothing will be referred to as parametric remapping, because the arclength parameter is modified from its original value to generate multiple redistribution functions along the basis functions for smoothing a grid. This process is robust, but does require the grid lines being redistributed to be of good quality to compute the basis functions upon which the grid points are moved.

The parametric remapping process is not limited to blending an arclength parameter from beginning to end of a region in the cross-direction. Multiple regions can be connected together and a cubic spline or Neville's method can be used to compute the new redistribution functions for their respective basis functions. For example, the above problem of the concave

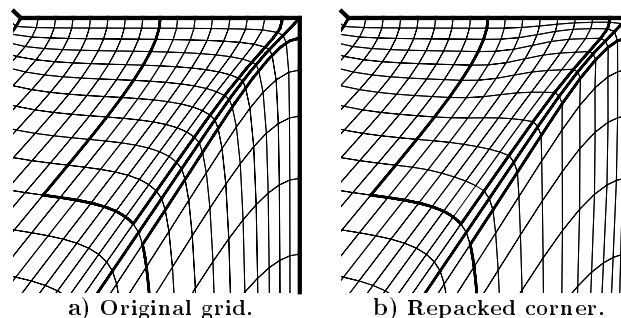


Fig. 9 Smoothed full region grid after packing points into the concavity.

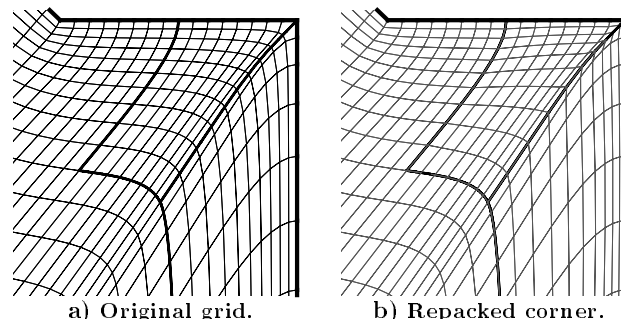


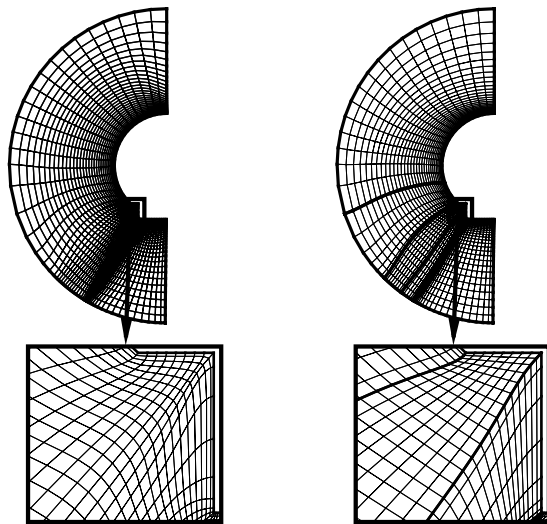
Fig. 10 Smoothed grid using multiple regions after packing points into the concavity.

region could have been smoothed by concatenating the fore and aft regions of the grid line emanating from the concave corner. The blending in the cross-direction could have been done with a cubic spline resulting in the smoothed replaced grid shown in Fig. 9, with the redistribution function values shown in Fig. 10.

Use of single or multiple redistribution functions along basis functions as described can be a very powerful tool in the smoothing of grids produced by other codes. Each set of functions can be coupled or single valued depending on the type of redistribution warranted. Maintaining the grid line upon which the redistribution is performed ensures recovery from poorly chosen redistribution functions and can promote better control over grid smoothing. These methods, when applied to the concave problem take about 5 minutes to execute on the SGI machine, thereby reducing the correction time of grid problems by nearly a factor of 50. By reusing the poorly generated grid as opposed to regenerating a new grid, the time to generate a 3D block face can be significantly reduced. Therefore, use of parametric remapping to do grid smoothing and correct grid problems can lead to the development of inviscid volume grids more efficiently.

Grid Smoothing with TFI

Besides manipulating arclength functions to smooth a surface or volume grid to correct problems created by the use of PDE solvers, the use of algebraic grid generation techniques can be apropos. The only require-



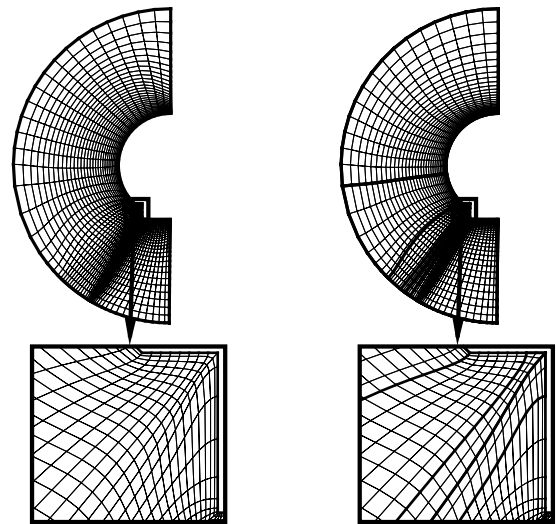
a) Original grid.

b) TFI regenerated.

Fig. 11 Poorly chosen subface domains for performing TFI to smooth a grid.

ment is that for Trans-Finite Interpolation¹⁹ (TFI) to work properly, the defining bounds of a region to be regenerated must be well posed.¹⁹ For example, the defining edges or faces, of a surface or volume, respectively can not have sharp changes in grid line character. This is illustrated in Fig. 11. If the identified region on the left is regenerated using TFI, the resulting grid on the right has poor grid quality. The poor grid quality is evident by the loss of orthogonality at the interior boundary in the concave region. In the same grid, if the identified region on the left in Fig. 12 is regenerated, the higher quality grid results because the orthogonal grid lines are used as the defining boundaries for the algebraic grid generation. The choosing of subfaces or subzones is extremely important when using TFI for smoothing purposes, but the technique can be used effectively. The choice of subface domains is important because the TFI method is very sensitive to the shape and distribution of points on the defining boundaries. If the defining boundaries are improperly chosen, the resulting grid will not be of any higher quality. In contrast, if the chosen subface boundaries are well posed, the interior grid can be of high quality.

To recover from selecting poorly identified regions for using the TFI technique, grid line smoothing is still possible. To smooth these types of grid lines, shown in Fig. 11(a), simply interpolate in the arclength parameter domain for those points that are producing poor grid character. This is done by first computing the basis function in the direction of the smoothing. Then vectors are created by fixing points on both sides of the region to be smoothed, shown on the left in Fig. 13. The derivatives at these fixed points are computed and used in Hermite interpolation in the arclength param-



a) Original grid.

b) TFI regenerated.

Fig. 12 Appropriately chosen subface domains for performing TFI to smooth a grid.

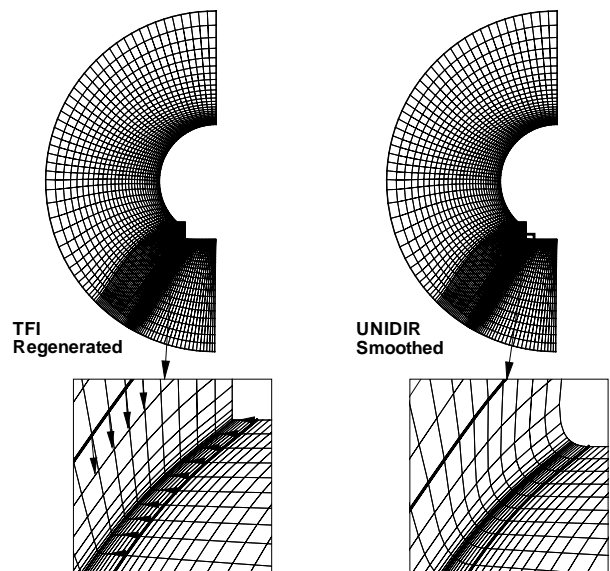


Fig. 13 UNI-DIRectionally smoothed grid across poorly chosen subface domain from performing TFI.

eter domain to ultimately smooth the grid lines, as illustrated on the right Fig. 13. This process will be referred to as UNIDIR or UNI-DIRectional smoothing. This smoothing process is not robust because the end points of a redistribution region at the cross-directional limits may not need smoothing. To account for this, a Dirichlet type boundary condition can be applied to the ends of the region by blending from no manipulation to full manipulation with weighting factors computed based on computational location, shown in Fig. 14. In Fig. 14, the cross-directional end points of the region are not smoothed but the interior grid is.

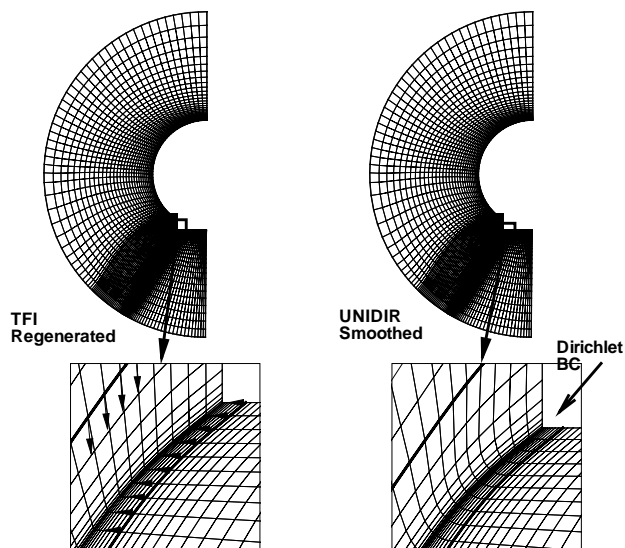


Fig. 14 UNI-DIRectionally smoothed grid across poorly chosen subface domain with Dirichlet boundary condition.

Use of TFI for grid smoothing, coupled with the UNIDIR process can easily remove poor quality grid line character. Employing the Dirichlet type boundary condition with the UNIDIR process can also provide a stand alone grid smoothing tool. Sometimes a PDE solver can produce highly kinked grid lines around convex corners that traverse 270 degrees, as shown in Fig. 14. Using the UNIDIR process, and maintaining the existing cell spacings, this kink can be easily eliminated. Whether TFI is coupled with the UNIDIR process, or used alone, each of these techniques offers powerful tools to remove poor grid line character and improve the overall quality of a surface and volume grid. In addition, these techniques are fast and efficient, thereby offering a faster alternative to using an elliptic solver and subface decompositions to correct poor grid quality.

Application of VGM Techniques

To explain how to use these powerful manipulations in accordance with one another, three examples will be illustrated. The first grid that will be created by the VGM techniques is a viscid grid from an inviscid grid for the example configuration. Utilizing the parametric remapping and TFI/UNIDIR techniques, the 3D block faced domain was generated for the example configuration. The volume grid was subsequently generated using the 3DMAGGS⁸ code. The initial volume grid exhibited regions of poor grid quality around the bodyflap leading edge regions, shown in Fig. 15. These poorly generated regions were regenerated using both classical rerunning of the PDE solver with various source term formulations, and using VGM to smooth the regions. The total time required by the PDE solver to obtain a smooth and usable volume grid for the poorly generated regions as well as the rest of

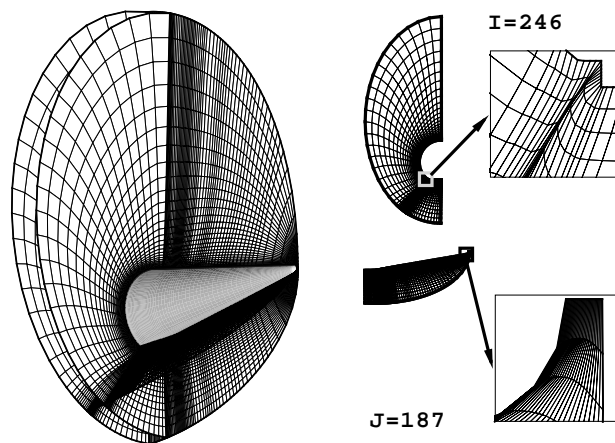


Fig. 15 Poor grid characteristic regions on example configuration.

the volume grid was 25 CRAY-YMP CPU hours, as compared to the 1 hour of manipulations using VGM. The latter only modifies the pertinent regions while the PDE solver operates on the entire volume grid. The repaired grid using the VGM techniques is shown in Fig. 16(b).

To generate the final viscous volume grid, the grid is subdivided into two equal regions in the K-dimension. The region that contains the wall is redistributed in the computational domain with 10% cell sizes at the wall and current cell sizes at the interface to the outer block. The redistribution is done twice, each time increasing the number of points by 100%, while the outer block remains fixed. The effects of redistribution in the computational domain have two significant results. First, if the grid lines have orthogonal character initially, the degree of orthogonality will be maintained or increased, as shown in Fig. 16(b) by the sparse overlay of the inviscid grid onto the viscous grid. This in turn increases the resolution of the boundary layer region as well as improves the grid quality. Second, by redistributing grid lines based on computational coordinates, the distribution function in this case is identical everywhere which results in a grid with similar grid line character in the non-redistribution directions; hence, good grid characteristics are either retained or enhanced due to improved near wall orthogonality. In summary, utilizing inviscid grids provides a functional skeleton on which the viscous grid is based, offering good grid characteristics and ease of generation.

A second application of the VGM technique is grid adaption of an axi-symmetric grid about a wind tunnel test configuration for the Mars-Pathfinder spacecraft.²⁰ In this example, the volume grid is represented by a 2D surface. The computed flow field using the original grid is shown at the top of Fig. 17. It is clearly evident that there are two main flow structures in the wake region that must be modeled with a higher degree of grid point clustering. The outer bow shock and the internal shear layer require better resolution to cap-

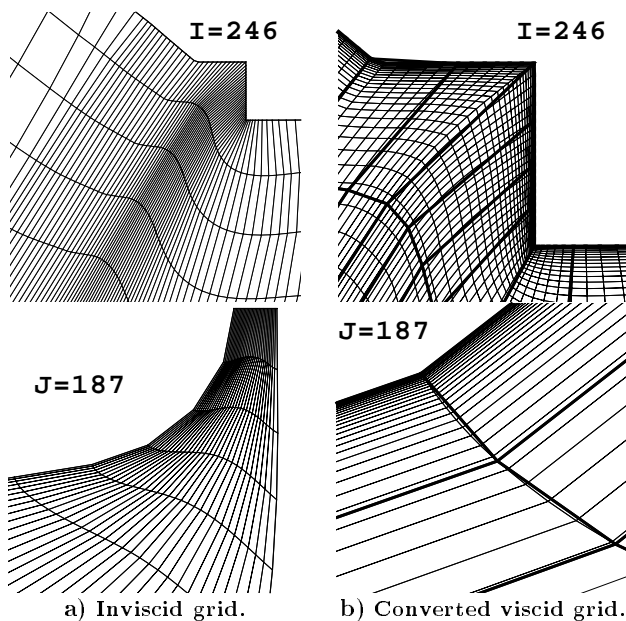
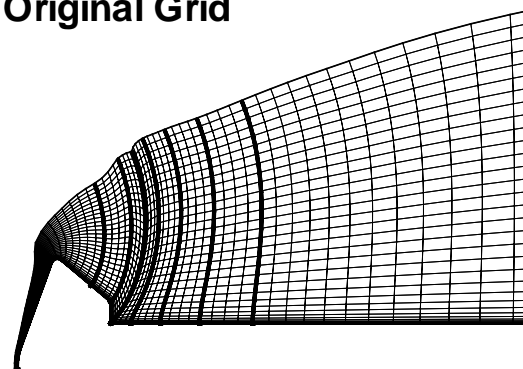


Fig. 16 Representative planes from an inviscid grid converted from an inviscid template.

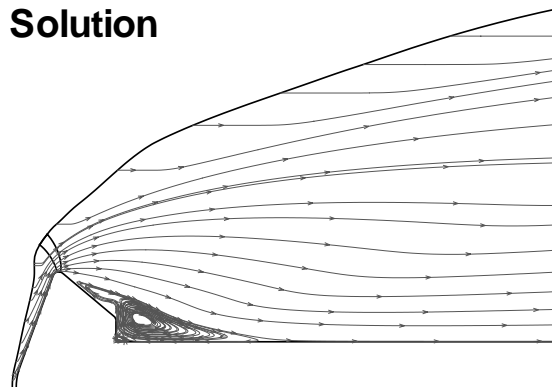
ture the important features. To adapt the grid for both features, equally spaced cross-sections are redistributed from the maximum diameter of the vehicle to the end of the wake, such that all wall corners are preserved. For this example, each grid line is redistributed to place 60 cells between the wall and the outer edge of the shear layer, 24 cells from the edge of the shear layer to the outer bow shock and 4 more cells from the edge of the bow shock to the outer boundary, for a total of 69 points (68 cells). Then the basis functions in the direction of the body to shock are computed for all interior grid lines between the redistributed equally spaced grid lines. By performing multiple remappings along these basis functions and UNIDIR smoothing at the interface to the forebody, an adapted grid results as shown at the bottom of Fig. 17. This entire process, from writing the VGM script to adapting the grid took 15 minutes. By comparison, the use of a PDE solver to do the same clustering and subsequent elliptic solving took 90 minutes and numerous subfaces.

Although all of the manipulations illustrated to this point have been in two dimensions, these manipulations are possible in three dimensions. For example, the X33 concept proposed by Lockheed Martin was initially evaluated for high speed regimes using a multiple block decomposition with ordered subset matches at block interfaces and large discontinuities in cell spacings. Further analysis was required in the transonic regime, but the grid from the hypersonic regime was not suitable due to the significant discontinuities in cell spacings, the number of blocks, and the non-point to point block interfaces. To generate a volume grid for transonic regime computations, the hypersonic grid was converted to a single block domain with cell to

Original Grid



Solution



Adapted Grid

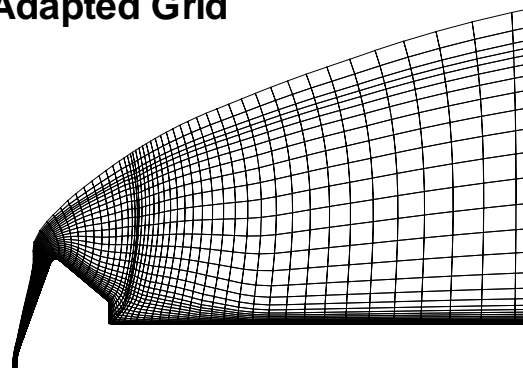


Fig. 17 Adapted flow field based on multiple flow structures.

cell scales of less than 1.5. This conversion was done through a series of single function remappings on various regions of the volume grid, the results of which can be seen in Fig. 18. Notice that on the left, the darker grid lines identify the various blocks that are converted into a single block on the right. This conversion from the hypersonic grid to the transonic grid took approximately 2 hours of manipulations. Use of a PDE solver to do the same conversion would have required a regeneration of the entire volume grid from scratch, consuming at least 100 hours of CPU time (the time it took to generate the initial hypersonic grid).

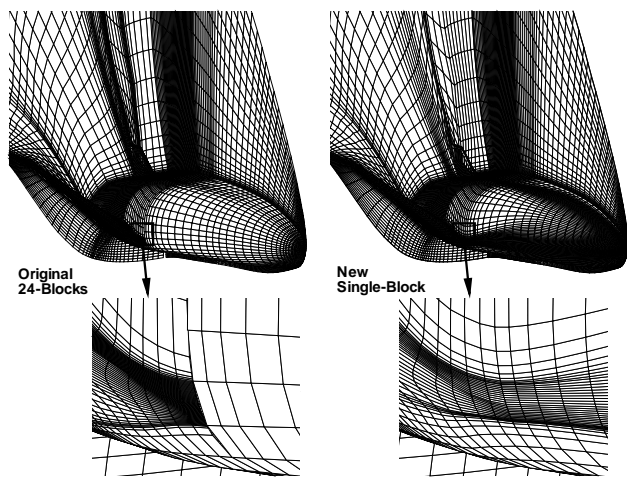


Fig. 18 X33 converted from a hypersonic to transonic grid.

Conclusion

A set of techniques for manipulating existing surface and volume grids to improve grid quality characteristics, reduce grid generation time and increase grid reusability have been presented. The parametric remapping techniques offer flexibility in the use of good directionally dependent grid lines to form the basis for simple but powerful manipulations for grid smoothing. The TFI and UNIDIR methods provide efficient manipulating capabilities that are not usually available in the solution of PDE's for grid generation purposes. Each technique, when used in conjunction with one another, forms a robust tool that can offer faster turnaround times to eliminate poor quality grid characteristics and provide reusability of existing grids. By reusing existing surface and volume grids, the process of obtaining CFD simulations about complex configurations can be significantly augmented and the time to get a final solution can be reduced.

Acknowledgment

This work was performed under NASA Langley Research Center contract NAS1-96014 and NAS1-20255. The author wishes to express his appreciation to Dr. David Schuster for his editorial comments, William Kleb for his contributions to the typesetting of this document and K. James Weilmuenster for permitting this work to be done.

References

- ¹Alter, S. J. and Cheatwood, F. M., "Elliptic Volume Grid Generation for Viscous Computations in Parametric Design Studies," AIAA Paper 96-1999, June 1996.
- ²Korpus, R., "Improved Methods for the Orthogonal Control of Highly Clustered Elliptically Generated Grids," *Software Systems for Surface Modeling and Grid Generation*, edited by R. E. Smith, Vol. CP-3143, NASA, 1992.
- ³Vatsa, V. N., Sanetrik, M. D., and Parlette, E. B., "Block-Structured Grids for Complex Aerodynamic Configurations: Current Status," *Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamic (CFD) Solutions*, edited by Y. K. Choo, Vol. CP-3143, NASA, 1995, pp. 163-177.

- ⁴Gnoffo, P. A., "An Upwind-Biased Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows," NASA Technical Paper 2953, February 1990.
- ⁵Davies, C. B. and Venkatapathy, E., "A Simplified Self-Adaptive Grid Code, SAGE," NASA TM-102198, October 1989.
- ⁶Benson, R. A., *Development of a Time-Accurate Solution Algorithm Coupled to a Dynamic Solution-Adaptive Grid Algorithm with Applications to Generic Inlet/Diffuser Configurations*, Ph.D. thesis, North Carolina State University, 1994.
- ⁷Alter, S. J. and Weilmuenster, K. J., "Single Block Three-Dimensional Volume Grids About Complex Aerodynamic Vehicles," NASA TM-108986, May 1993.
- ⁸Alter, S. J. and Weilmuenster, K. J., "The Three-Dimensional Multi-block Advanced Grid Generation System (3DMAGGS)," NASA TM-108985, April 1993.
- ⁹Sorenson, R. L. and Alter, S. J., "3DGRAPE/AL: The Ames/Langley Technology Upgrade," No. CP-3291, May 1995, pp. 447-462.
- ¹⁰J. P. Steinbrenner, Chawner, J. R., and Fouts, C. L., "The GRIDGEN 3D Multiple Block Grid Generation System," Wright Research and Development Center Report WRDC-TR-90-3022, October 1989.
- ¹¹Thompson, J. F., "A Composite Grid Generation Code for General 3D Regions-the EAGLE Code," *AIAA Journal*, Vol. 26, No. 3, March 1988, pp. 1-10.
- ¹²Chan, W. M., Chiu, I., and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM-108791, October 1993.
- ¹³Noack, R. W. and Anderson, D. A., "Solution Adaptive Grid Generation Using Parabolic Partial Differential Equations," AIAA paper 88-0315, January 1988.
- ¹⁴Soni, B. K., "A User's Guide to GENIE," Mississippi state university, MS, November 1991.
- ¹⁵Olynick, D. D. R., "Importance of 3-D Grid Resolution and Structure for Calculating Reentry Heating Environments," AIAA Paper 96-1857, June 1996.
- ¹⁶Alter, S. J., "The Volume Grid Manipulator (VGM): A Grid Reusability Tool," NASA CR to be published by, March 1997.
- ¹⁷Vinokur, M., "On One-Dimensional Stretching Functions for Finite-Difference Calculations," NASA CR-3313, 1993.
- ¹⁸Alter, S. J. and Cheatwood, F. M., "Volume Grid Expansion Techniques for Computational Fluid Dynamic Algorithms," AIAA Paper 96-0029, January 1996.
- ¹⁹Soni, B. K., "Two- and Three-Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics," AIAA Paper 85-1526, 1985.
- ²⁰Wereinski, P. F., "Mars Aerocapture Analysis for the MESUR/MARS-Pathfinder Aeroshell in Low L/D Configurations," AIAA Paper 95-3495, June 1995.