1995

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

ADAPTIVE GRID METHODS FOR RLV ENVIRONMENT ASSESSMENT AND NOZZLE ANALYSIS

Prepared By:	Hugh J. Thornburg, Ph.D.
Academic Rank:	Research Engineer I
Institute and Department:	Mississippi State University National Science Foundation Engineering Research Center
NASA/MSFC:	
Laboratory:	Structures and Dynamics
Division:	Fluid Dynamics
Branch:	Computational Fluid Dynamics
MSFC Colleague(s):	Paul McConnaughey, Ph.D. T.S. Wang, Ph.D. Joe Ruf

Adaptive Grid Method for RLV Environment Assessment and Nozzle Analysis

by

Hugh Thornburg
NSF Engineering Research Center for
Computational Field Simulation
Mississippi State University
Mississippi State, MS 39762

INTRODUCTION

Rapid access to highly accurate data about complex configurations is needed for multi–disciplinary optimization and design. In order to efficiently meet these requirements a closer coupling between the analysis algorithms and the discretization process is needed. In some cases, such as free surface, temporally varying geometries, and fluid structure interaction, the need is unavoidable. In other cases the need is to rapidly generate and modify high quality grids. Techniques such as unstructured and/or solution–adaptive methods can be used to speed the grid generation process and to automatically cluster mesh points in regions of interest. Global features of the flow can be significantly affected by isolated regions of inadequately resolved flow. These regions may not exhibit high gradients and can be difficult to detect. Thus excessive resolution in certain regions does not necessarily increase the accuracy of the overall solution.

Several approaches have been employed for both structured and unstructured grid adaption. The most widely used involve grid point redistribution, local grid point enrichment/derefinement or local modification of the actual flow solver. However, the success of any one of these methods ultimately depends on the feature detection algorithm used to determine solution domain regions which require a fine mesh for their accurate representation. Typically, weight functions are constructed to mimic the local truncation error and may require substantial user input. Most problems of engineering interest involve multi–block grids and widely disparate length scales. Hence, it is desirable that the adaptive grid feature detection algorithm be developed to recognize flow structures of different type as well as differing intensity, and adequately address scaling and normalization across blocks. These weight functions can then be used to construct blending functions for algebraic redistribution, interpolation functions for unstructured grid generation, forcing functions to attract/repel points in an elliptic system, or to trigger local refinement, based upon application of an equidistribution principle. The popularity of solution–adaptive techniques is growing in tandem with unstructured methods. The difficultly of precisely controlling mesh densities and orientations with current unstructured grid generation systems has driven the use of solution–adaptive meshing Use of derivatives of density or pressure are widely used for construction of such weight functions, and have been proven very successful for inviscid flows with shocks[2,7,11]. However, less success has been realized for flowfields with viscous layers, vortices or shocks of disparate strength. It is difficult to maintain the appropriate mesh point spacing in the various regions which require a fine spacing for adequate resolution. Mesh points often migrate from important regions due to refinement of dominant features. An example of this is the well know tendency of adaptive methods to increase the resolution of shocks in the flowfield around airfoils, but in the incorrect location due to inadequate resolution of the stagnation region. This problem has been the motivation for this research.

In this research a NURBS representation is employed to define block surfaces for boundary point redistribution. The features described have been implemented into Adapt2D/3D. An adaptive grid system capable of automatically resolving complex flows with shock waves, expansion waves, shear layers and complex vortex–vortex and vortex–surface interactions. An adaptive grid approach seems well suited for such problems in which the spatial distribution of length scales is not known a priori.

APPROACH TO ADAPTION

The elliptic generation system:

$$\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\hat{r}_{\xi^i\xi^j} + \sum_{k=1}^{3} g^{kk}P_k\hat{r}_{\xi^k} = 0 \qquad (1)$$

where r : Position vector,
g^{ij} : Contravariant metric tensor
ξ^i : Curvilinear coordinate, and
P_k : Control function.

is widely used for grid generation [1]. Control of the distribution and characteristics of a grid system can be achieved by varying the values of the control functions P_k in Equation 1 [1]. The application of the one dimensional form of Equation 1 combined with equidistribution of the weight function results in the definition of a set of control functions for three dimensions.

$$P_i = \frac{(W_i)_{\xi^i}}{W_i} \quad (i = 1,2,3) \qquad (2)$$

These control functions were generalized by Eiseman [2] as:

$$P_i = \sum_{j=1}^{3}\frac{g^{ij}}{g^{ii}}\frac{(W_i)_{\xi^i}}{W_i} \quad (i = 1,2,3) \qquad (3)$$

In order to conserve some of the geometrical characteristics of the original grid the definition of the control functions is extended as:

$$P_i = \left(P_{initialgeometry}\right) + c_i(P_{wt}) \quad (i = 1,2,3) \qquad (4)$$

where $P_{initial\ geometry}$: Control function based on initial geometry
P_{wt} : Control function based on current solution
c_i : Constant weight factor.

These control functions are evaluated based on the current grid at the adaption step. This can be formulated as:

$$P_i^{(n)} = P_i^{(n-1)} + c_i(P_{wt})^{(n-1)} \quad (i = 1,2,3) \qquad (5)$$

where

$$P_i^{(1)} = P_i^{(0)} + c_i(P_{wt})^{(0)} \quad (i = 1,2,3) \qquad (6)$$

A flow solution is first obtained with an initial grid. Then the control functions P_i are evaluated in accordance with Equations 2 and 5, which is based on a combination of the geometry of the current grid and the weight functions associated with the current flow solution[11].

Evaluation of the forcing functions corresponding to the grid input into the adaptation program has proven to be troublesome. Direct solution of Equation 1 for the forcing functions using the input grid coordinates via Cramer's rule or IMSL libraries was not successful. For some grids with very high aspect ratio cells and/or very rapid changes in cell size, the forcing functions became very large. The use of any differencing scheme other than the one used to evaluate the metrics, such as the hybrid upwind scheme[8], would result in very large mesh point movements. An alternative technique for evaluating the forcing functions based on derivatives of the metrics was implemented[3].

$$P_i = \frac{1}{2}\frac{(g_{11})_{\xi^i\xi^i}}{g_{11}} + \frac{1}{2}\frac{(g_{22})_{\xi^i\xi^i}}{g_{22}} + \frac{1}{2}\frac{(g_{33})_{\xi^i\xi^i}}{g_{33}} \quad (i = 1,2,3) \qquad (7)$$

This technique has proven to be somewhat more robust, but research efforts are continuing in this area.

WEIGHT FUNCTIONS

Application of the equidistribution law results in grid spacing inversely proportional to the weight function, and hence, the weight function determines the grid point distribution. Ideally, the weight would be the local truncation error ensuring a uniform distribution of error. However, evaluation of the higher–order derivatives contained in the truncation error from the available discrete data is progressively less accurate as the order increases and is subject to noise. Determination of this function is one of the most challenging areas of adaptive grid generation. Lower–order derivatives must be non–zero in regions of wide variation of higher–order derivatives, and are proportional to the rate of variation. Therefore, lower–order derivatives are often used to construct a weight function as a proxy for the truncation error. Construction of these weight functions often requires the user to specify which derivatives and in what proportion they are to be used. This can be a time consuming process. Also, due to the disparate strength of flow features, important features can be lost in the noise of dominant features. The weight functions developed by Soni and Yang [7] and Thornburg and Soni [8] are examples of such efforts. The weight function of Thornburg and Soni [8] has the attractive feature of requiring no user specified inputs. Relative derivatives are used to detect features of varying intensity, so that weaker, but important structures such as vortices are accurately reflected in the weight function. In addition, each conservative flow variable is scaled independently. One–sided differences are used at boundaries, and no–slip boundaries require special treatment since the velocity is zero. This case is handled in the same manner as zero velocity regions in the field. A small value, epsilon in equation 8, is added to all normalizing quantities. In the present work this weight function has been modified using the Boolean sum construction method of Soni [7]. Also, several enhancements of an implementation nature have been employed. For example epsilon has been placed outside the absolute value operator. This eliminated the possibility of spurious gradients in the weight function in regions where epsilon was nearly equal and opposite in sign to the local normalizing flow variable. Also, the normalizing derivatives have been set to an initial or minimum value of ten percent of the freestream quantities. This alleviates problems encountered in flows without significant features to trigger adaption in one or more coordinate directions. Otherwise a few percent variation would be normalized to the same level as a shock or other strong feature. The current weight function is as follows:

$$W = \frac{W^1}{max(W^1, W^2, W^3)} \oplus \frac{W^2}{max(W^1, W^2, W^3)} \oplus \frac{W^3}{max(W^1, W^2, W^3)}$$

Where, $k=1,2,3,$ and

$$W^k = 1 + \frac{\left| \varrho_{\xi^k} \right|}{|\ \varrho\ | + \varepsilon} \oplus \frac{\left| (\varrho u)_{\xi^k} \right|}{|\ (\varrho u)\ | + \varepsilon} \oplus \frac{\left| (\varrho v)_{\xi^k} \right|}{|\ (\varrho v)\ | + \varepsilon} \oplus \frac{\left| (\varrho w)_{\xi^k} \right|}{|\ (\varrho w)\ | + \varepsilon}$$

$$\oplus \frac{\left| \varrho_{\xi^k \xi^k} \right|}{|\ \varrho\ | + \varepsilon} \oplus \frac{\left| (\varrho u)_{\xi^k \xi^k} \right|}{|\ (\varrho u)\ | + \varepsilon} \oplus \frac{\left| (\varrho v)_{\xi^k \xi^k} \right|}{|\ (\varrho v)\ | + \varepsilon} \oplus \frac{\left| (\varrho w)_{\xi^k \xi^k} \right|}{|\ (\varrho w)\ | + \varepsilon} \quad (8)$$

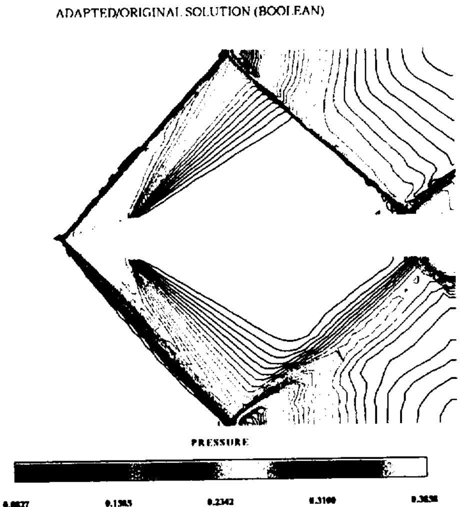

Figure 3. Comparison of Solutions Using Adapted Grid.

References

1. Thompson, J.F., Warsi, Z.U.A. and Mastin, C.W., Numerical Grid Generation: Foundations [illegible]

2. Eiseman, P.R., "Alternating Direction Adaptive Grid Generation," AIAA Paper 83–1937, 1983.

3. Soni, B.K., "Structured Grid Generation in Computational Fluid Dynamics," Vichnevetsky, R., Knight, D., and Richter, G. (eds.), Advances in Computer Methods for Partial Differential Equations VII, Rutgers University, pp 689–695, June 1992.

4. Buning, P.G. and Steger, J.L., "Graphics and Flow Visualization in Computational Fluid Dynamics," AIAA Paper 85–1507–CP, Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, 1985.

5. NASA LeRC and USAF AEDC, "NPARC 1.0 User Notes," June 1993.

6. Ghia, K.N., Ghia, U., Shin, C.T. and Reddy, D.R., "Multigrid Simulation of Asymptotic Curved–Duct Flows Using a Semi–Implicit Numerical Technique," Computers in Flow Prediction and Fluid Dynamics Experiments, ASME Publication, New York 1981.