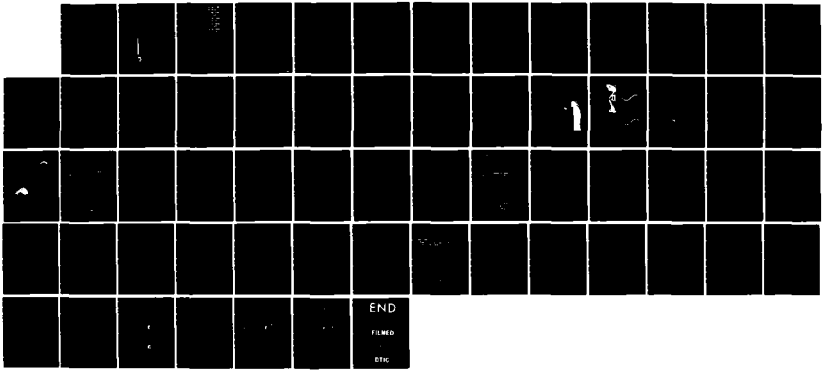
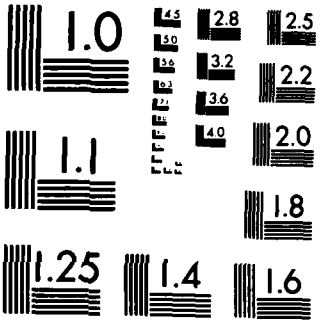
AD-A151 175 APPLICATION OF ADAPTIVE GRIDS IN SOLVING THE PARTIAL DIFFERENTIAL EQUATIO..(U) IOWA STATE UNIV AMES ENGINEERING RESEARCH INST D A ANDERSON ET AL. SEP 84 1/1

UNCLASSIFIED ISU-ERI-AMES-85412 AFOSR-TR-85-0186 F/G 20/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ENGINEERING
RESEARCH
INSTITUTE

IOWA STATE
UNIVERSITY
AMES, IOWA

Final Report

May 1, 1983 - July 31, 1984

APPLICATION OF ADAPTIVE GRIDS IN SOLVING THE PARTIAL DIFFERENTIAL EQUATIONS GOVERNING FLUID FLOW

Dale A. Anderson, Principal Investigator
Richard G. Hindman, Major Staff

Department of Aerospace Engineering
and Computational Fluid Dynamics Institute

September 1984

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

ENGINEERING RESEARCH

ENGINEERING RESEARCH
ENGINEERING RESEARCH
ENGINEERING RESEARCH
ENGINEERING RESEARCH

Final Report

May 1, 1983 - July 31, 1984

APPLICATION OF ADAPTIVE GRIDS IN SOLVING THE PARTIAL DIFFERENTIAL EQUATIONS GOVERNING FLUID FLOW

Dale A. Anderson, Principal Investigator
Richard G. Hindman, Major Staff

Department of Aerospace Engineering
and Computational Fluid Dynamics Institute

September 1984

Prepared for
AFOSR Grant-83-0167

ISU-ERI-Ames-85412
CFD-10
Project-1650

COMPUTATIONAL FLUID DYNAMICS INSTITUTE
COLLEGE OF ENGINEERING
IOWA STATE UNIVERSITY, AMES, IOWA 50011

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 85-0186
6a. NAME OF PERFORMING ORGANIZATION IOWA STATE UNIVERSITY 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFOSR/NA
6c. ADDRESS (City, State and ZIP Code) ENGINEERING RESEARCH INSTITUTE AMES, IA 50011	7b. ADDRESS (City, State and ZIP Code) Bolling AFB, DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AIR FORCE OFFICE OF SCIENTIFIC RESEARCH 8b. OFFICE SYMBOL (If applicable) AFOSR/NA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-83-0167
8c. ADDRESS (City, State and ZIP Code) BOLLING AFB, DC 20332	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2307; TASK NO. A1; WORK UNIT NO.
11. TITLE (Include Security Classification) APPLICATION OF ADAPTIVE GRIDS IN SOLVING THE PARTIAL DIFFERENTIAL EQUATIONS GOVERNING FLUID FLOW (UNCLASSIFIED)	
12. PERSONAL AUTHOR(S) ANDERSON, DALE A HINDMAN, RICHARD G	
13a. TYPE OF REPORT FINAL	13b. TIME COVERED FROM 01MAY83 TO 31JUL84
14. DATE OF REPORT (Yr., Mo., Day) 1984, September	15. PAGE COUNT 59
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) GRID GENERATION; FINITE DIFFERENCE METHODS; ADAPTIVE GRIDS; MULTIDIMENSIONAL;
19. ABSTRACT (Continue on reverse if necessary and identify by block number) A brief review of the goals and progress of the research on adaptive grid generation is presented. The principal results of the research are given by four papers supported by Grant AFOSR-83-0167 which comprise the appendix: "Adaptive Grid Methods for Partial Differential Equations" from Advances on Grid Generation, and AIAA papers 84-1608, 84-1610, and 84-1668.	
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr James D Wilson	22b. TELEPHONE NUMBER (Include Area Code) 202/767-4935 22c. OFFICE SYMBOL AFOSR/NA

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ABSTRACT

Work performed under Grant AFOSR-83-0167 is summarized in this report. A review of the original goals is presented, new directions and ideas are included, and significant accomplishments are listed. This report covers research completed under funding intended for the first year of a three-year program. The original grant period was interrupted at the end of the first year because the principal investigator transferred from Iowa State University to the University of Texas at Arlington. Papers published with support from this program are listed and included as an appendix to this report.

CONTENTS

1. INTRODUCTION

The application of finite-difference methods in numerically solving partial differential equations governing fluid flow has become increasingly commonplace over the past two decades. Early work was concentrated on solving simple linearized equations for very simple geometric configurations. As computers became more sophisticated, algorithms were improved and applications to more complex problems were attempted. Today, most companies and government agencies with interests in fluid dynamics have in-house capabilities for solving flow problems using numerical techniques.

Even though complicated problems can be solved, there are still a number of pacing areas that are crucial in the computational fluid dynamics field. One of these areas, mesh generation, is perhaps one of the most important topics needing further development if continued progress is to be made. In fact, coordinate system selection and grid generation are probably the most important topics requiring study if continued advances in digital simulation of fluid flow around flight vehicles are to be made.

While a number of problems in grid generation for different geometries can be identified, the placement of grid points in order to adequately resolve a flow and provide a reduction of global error in a numerical calculation is of major interest. Since the solution of a particular flow is not known a priori, the grid points cannot be placed in the best positions before the calculation is complete. Consequently, it is advantageous to adjust the grid point locations in such a way as to provide the best solution as the computation progresses. This idea

of adaptive grids was the subject of AFOSR Grant-83-0167, "Application of Adaptive Grids in Solving the Partial Differential Equations Governing Fluid Flow."

The original proposal for work under this program was based upon applying the equidistribution concept for generating grids to two- and three-dimensional problems. The technique applied by Rai and Anderson[1-2] was selected as a candidate method. This method was used in early experiments in two-dimensions for small amounts of grid adjustment with good success.

Early in the program, several example calculations were completed for high values of grid adaption with the Rai and Anderson scheme. For large adaption rates, severe grid skewness was encountered. The one-dimensional method employed by Dwyer et al.[3] was also extended to two dimensions. Similar examples were tested with this scheme and severe grid skewness also resulted for large adaption rates. Faced with severe grid distortion in the multidimensional case, a new way of creating an adaptive mesh was needed. Not only must the technique be capable of providing the desired grid adaption but some positive method of grid skewness control is necessary. It is now accepted by researchers that any method for generating an adaptive grid without an active skewness control will ultimately fail in two or three dimensions.

Faced with this mesh control problem, it was necessary to review the fundamental concept of equidistribution of a weight function over an area. This concept was retained as the main idea in the generation of an adaptive grid. However, it was clear that the relationship between the weight function and the mesh skewness must be developed. Research

on this issue progressed until an explicit formulation relating the angle between mesh lines to the weight function was developed. This also lead to a simple way of generating an orthogonal grid for two dimensions.

2. SIGNIFICANT MILESTONES ACHIEVED AND STATUS OF RESEARCH UNDER AFOSR GRANT-83-0167

2.1 Milestones

During the present program, a number of important contributions were made in the understanding of adaptive grids.

1. In an invited review paper presented to ASME,[4] a number of methods for creating two-dimensional mesh systems were shown to be virtually the same. These schemes are based upon the concept of equidistribution of a weight function over an area.

2. At the same time it was shown that all schemes for generating an adaptive mesh without an active skewness control will fail.

3. Recently, the direct relationship between the skewness of mesh lines and the weight function was demonstrated.[5] This shows how the grid distortion can be controlled while still providing an adaptive grid.

2.2 Status

At the time of the termination of this program, the concept of constructing a two-dimensional grid based on equidistribution was well established. In addition, the coupling between grid adaption, skewness, and weight function was well understood. A series of numerical experiments is needed to demonstrate that grid adaption and control can both be attained for two-dimensional grids. This demonstration would pave the way for practical application to the equations governing fluid flow.

In reviewing the goals and proposed methodology for attaining the goals of this program, few changes were required. The necessary changes in approach were a result of knowledge gained as research results were obtained. Continued research on adaptive grids using the ideas developed on this program should provide a practical method that can be applied to useful problems.

3. REFERENCES

1. Rai, M. M. and Anderson, D. A., "Grid Evolution in Time Asymptotic Problems," Journal of Computational Physics, Vol. 43, No. 2, October 1981, pp. 327-344.

2. Rai, M. M. and Anderson, D. A., "The Use of Adaptive Grids in Conjunction with Shock Capturing Methods," AIAA Paper 81-1012, June 1981.

3. Dwyer, H. A., Smooke, M. D., and Kee, R. J., "Adaptive Gridding for Finite-Difference Solutions to Heat and Mass Transfer Problems," Numerical Grid Generation, ed. J. F. Thompson. New York: Els Science Publishing Co., Inc., 1982, pp. 339-350.

4. Anderson, D. A., "Adaptive Grid Methods for Partial Differential Equations," in Advances in Grid Generation, presented at Applied Mechanics, Bioengineering, and Fluids Engineering Conference (Houston, Texas,) ed. K. Ghia and U. Ghia. New York: ASME, 1983, pp. 1-15.

5. Anderson, D. A. and Rajendran, N., "Two Approaches Toward Generating Orthogonal Adaptive Grids," AIAA Paper 84-1610, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, Colorado, June 25-27, 1984.

4. APPENDIX

Papers published with funding made available under Grant AFOSR-83-0167.

A. Anderson, D. A., "Adaptive Grid Methods for Partial Differential Equations," in Advances in Grid Generation, presented at Applied Mechanics, Bioengineering, and Fluids Engineering Conference (Houston, Texas,) ed. K. Ghia and U. Ghia. New York: ASME, 1983, pp. 1-15.

B. Anderson, D. A. and Rajendran, N., "Two Approaches Toward Generating Orthogonal Adaptive Grids," AIAA Paper 84-1610, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, Colorado, June 25-27, 1984.

C. Holcomb, J. E. and Hindman, R. G., "Development of a Dynamically Adaptive Grid Method for Multidimensional Problems," AIAA Paper 84-1668, June 1984.

D. Steinbrenner, J. P., Tassa, Y., and Anderson, D. A., "An Adaptive Grid Scheme Applied to Two-Dimensional Airfoil Problems," AIAA Paper 84-1608, June 1984.

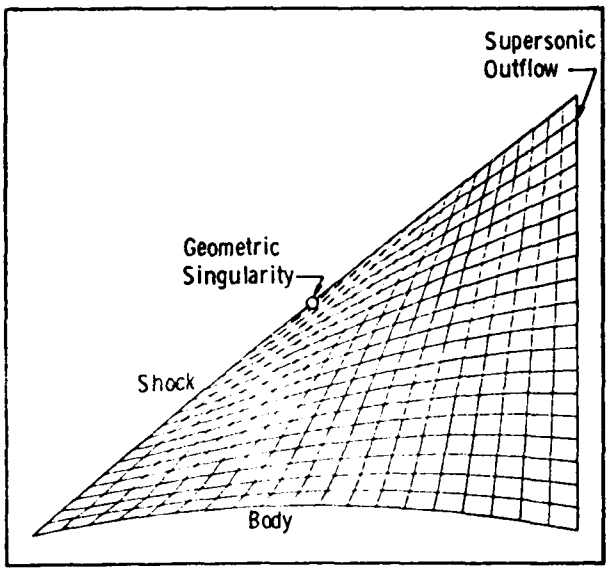


Fig. 13 Grid structure for supersonic flow over an ogive

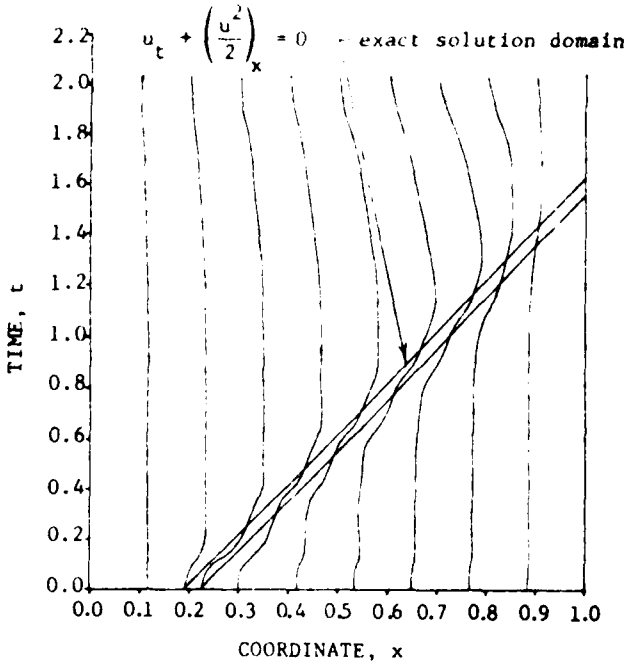


Fig. 14 Time history of adaptive grid for a moving discontinutiy

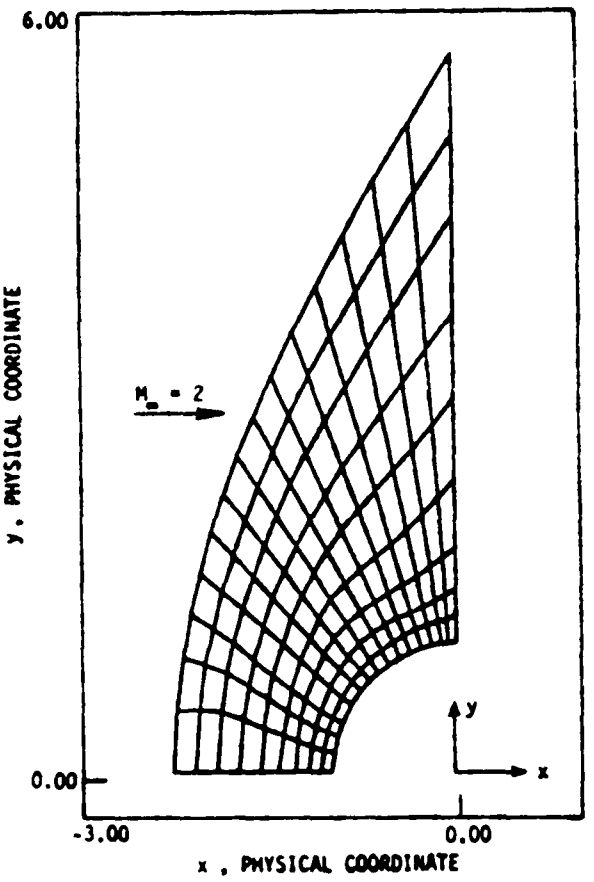


Fig. 15 Converged adaptive grid for supersonic blunt-body problem

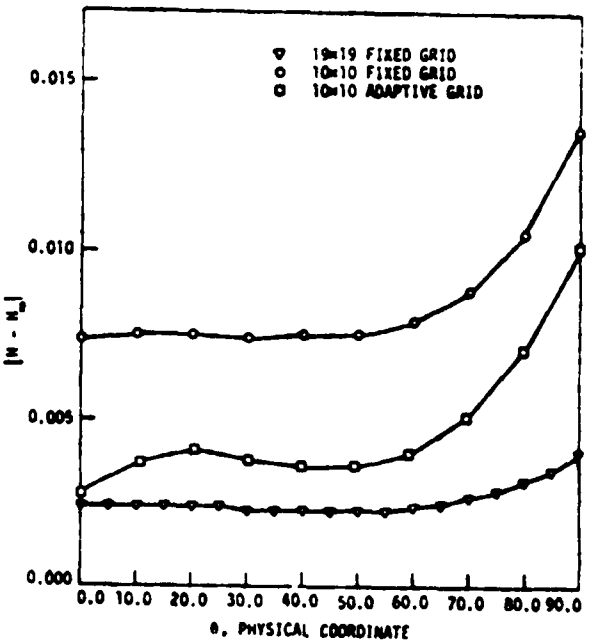


Fig. 16 Total enthalpy comparison for the supersonic blunt-body problem

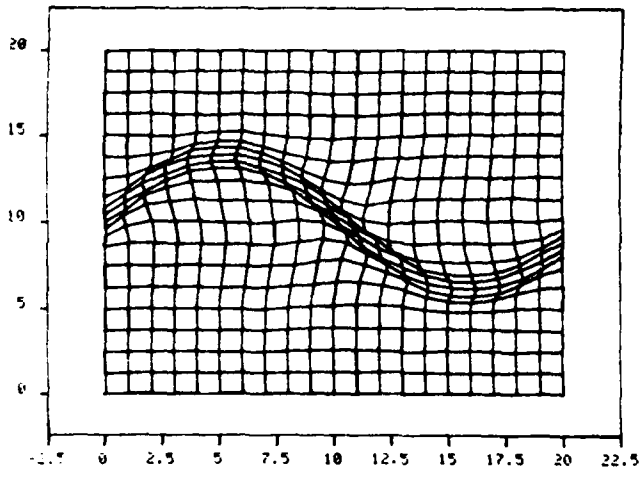


Fig. 9 Grid adaption for sinusoidal function distribution, a = 0, b = 10

$$u(x,y) = 0 \qquad 0 \le y \le 9. + 4\sin\left(\frac{2\pi x}{21}\right)$$

$$u(x,y) = 0.5\left[y - 9. - 4\sin\left(\frac{2\pi x}{21}\right)\right],$$

$$9. + 4\sin\left(\frac{2\pi x}{21}\right) \le y \le 11. + 4\sin\left(\frac{2\pi x}{21}\right)$$

$$u(x,y) = 1.0 \qquad 11. + 4\sin\frac{2\pi x}{21} \le y \le 20.$$

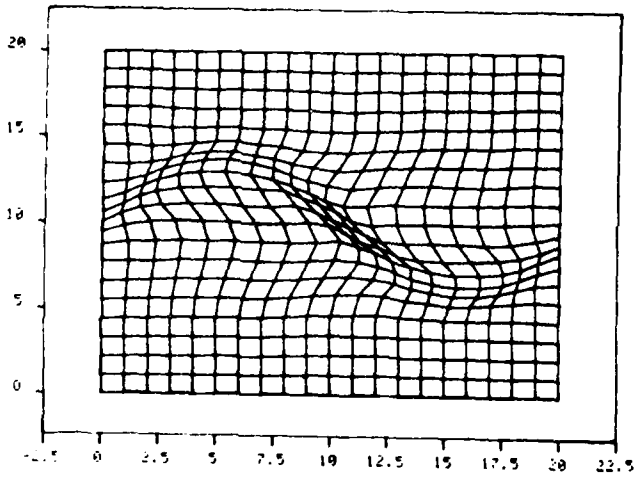


Fig. 10 Grid adaption for sinusoidal function distribution, a = 2, b = 3

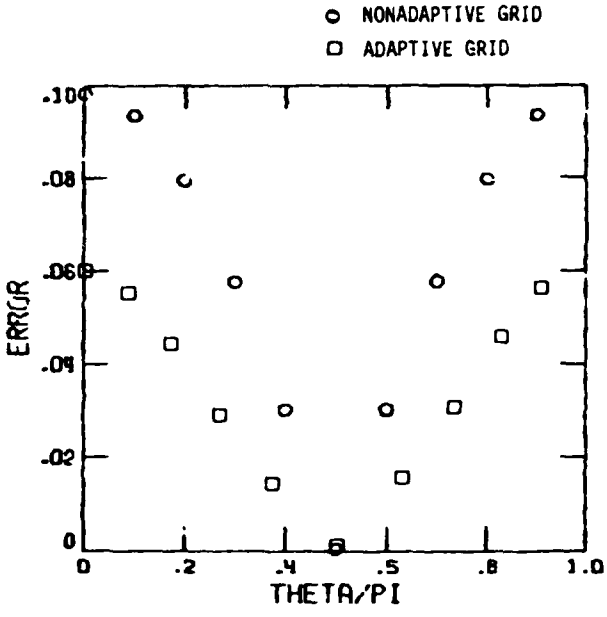


Fig. 11 Comparison of error in surface potential for a cylinder

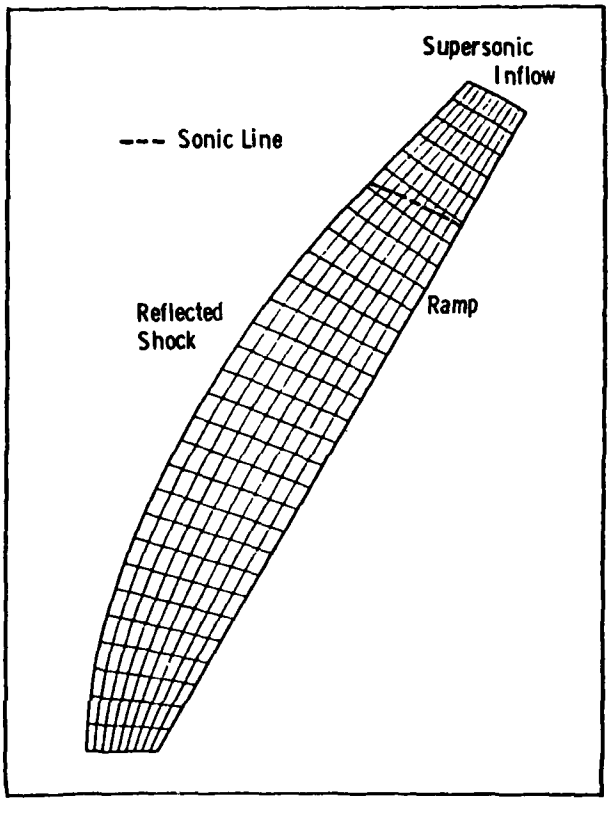
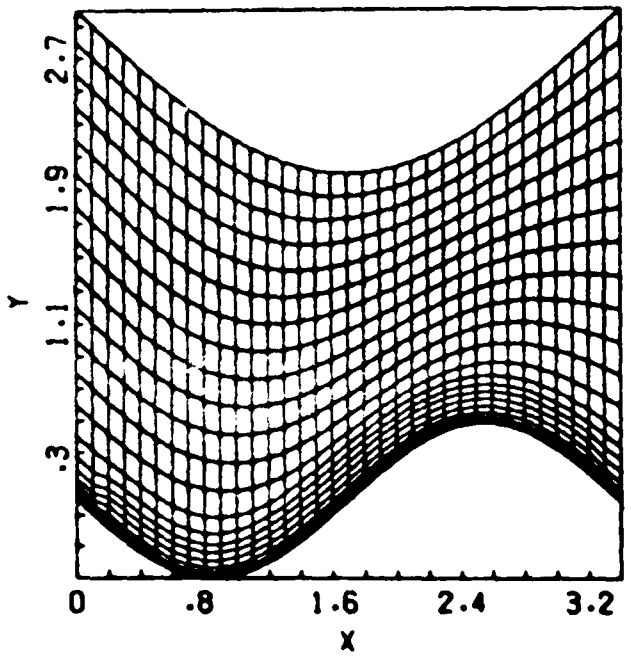


Fig. 12 Grid structure for Mach reflection of a shock from a ramp



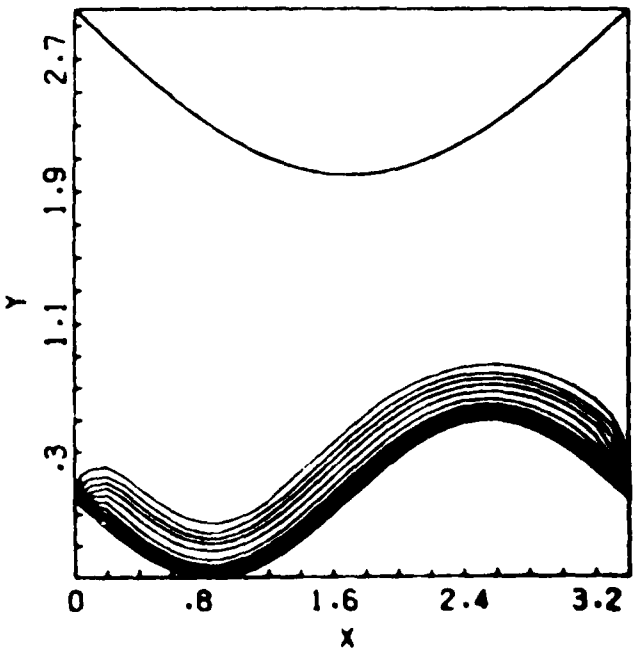


Fig. 6 Intermediate grid structure and isotherms

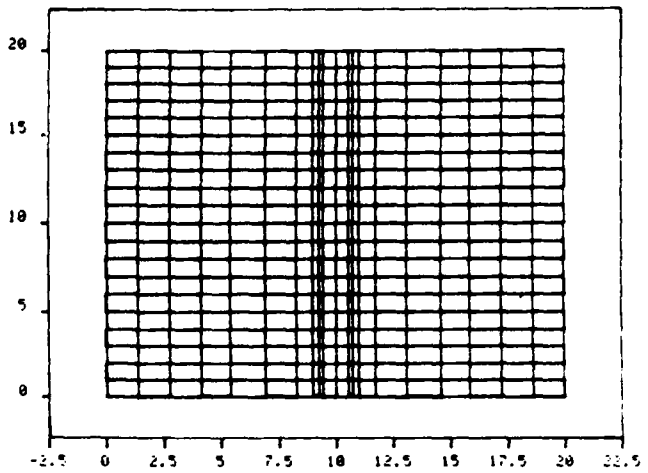


Fig. 6 One-dimensional grid adaption, a = 10, b = 0

$u(x,y) = 1.0 \qquad 0 \leq x \leq 9$

$u(x,y) = 0.5(11. - x) \qquad 9 \leq x \leq 11$

$u(x,y) = 0.0 \qquad 11 \leq x \leq 20$

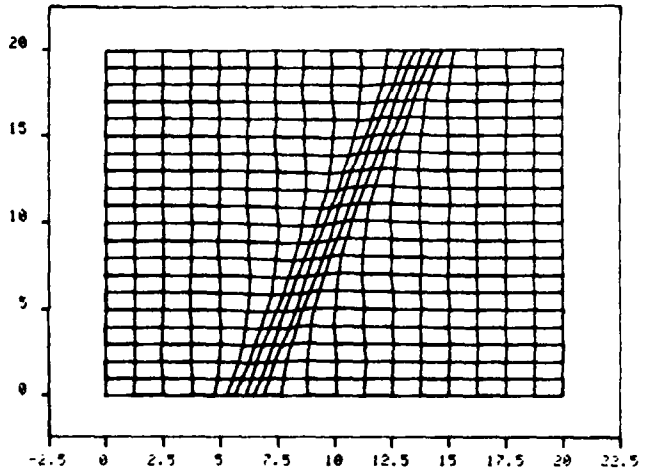


Fig. 7 Grid adaption with two degrees of freedom, a = 10, b = 0

$u(x,y) = 1.0 \qquad 0 \leq x \leq 0.4(y + 12.5)$

$u(x,y) = 0.5(7. - x) + 0.2y, \quad 0.4(y + 12.5) \leq x \leq 0.4(y + 17.5)$

$u(x,y) = 0.0 \qquad 0.4(y + 17.5) \leq x \leq 20$

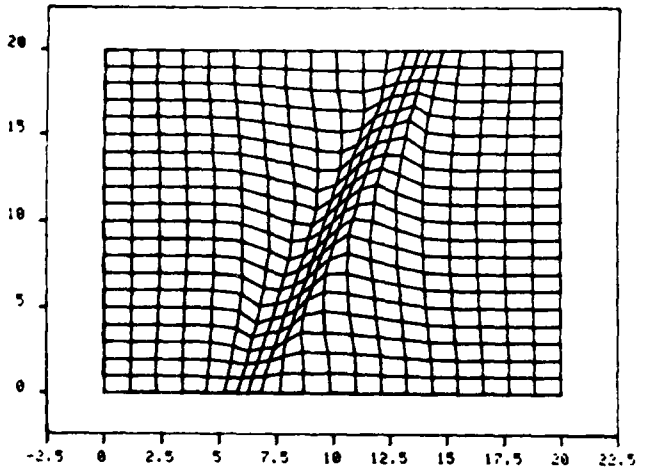


Fig. 8 Grid adaption with two degrees of freedom, a = 4, b = 4

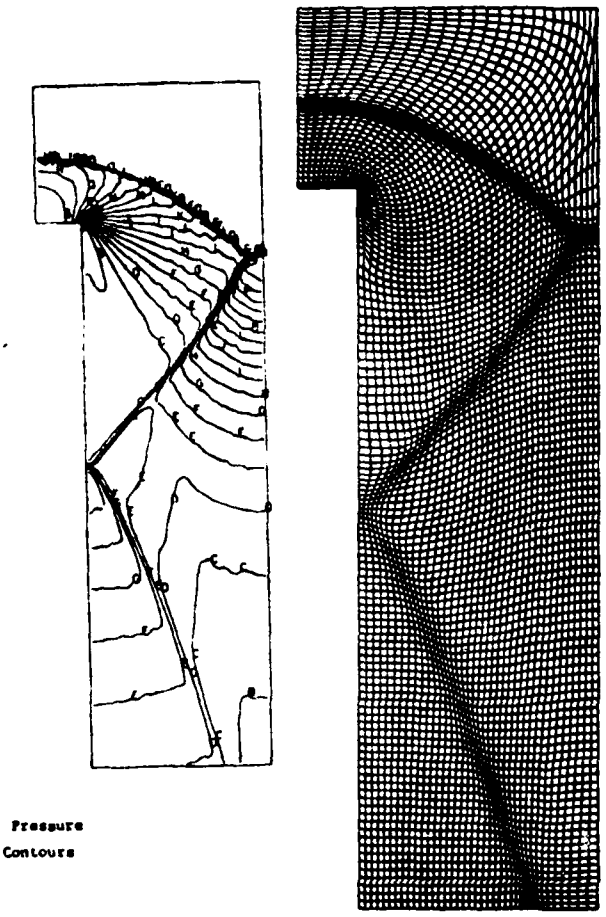


Fig. 3 Supersonic flow over forward step
Mach reflection at the wall boundary

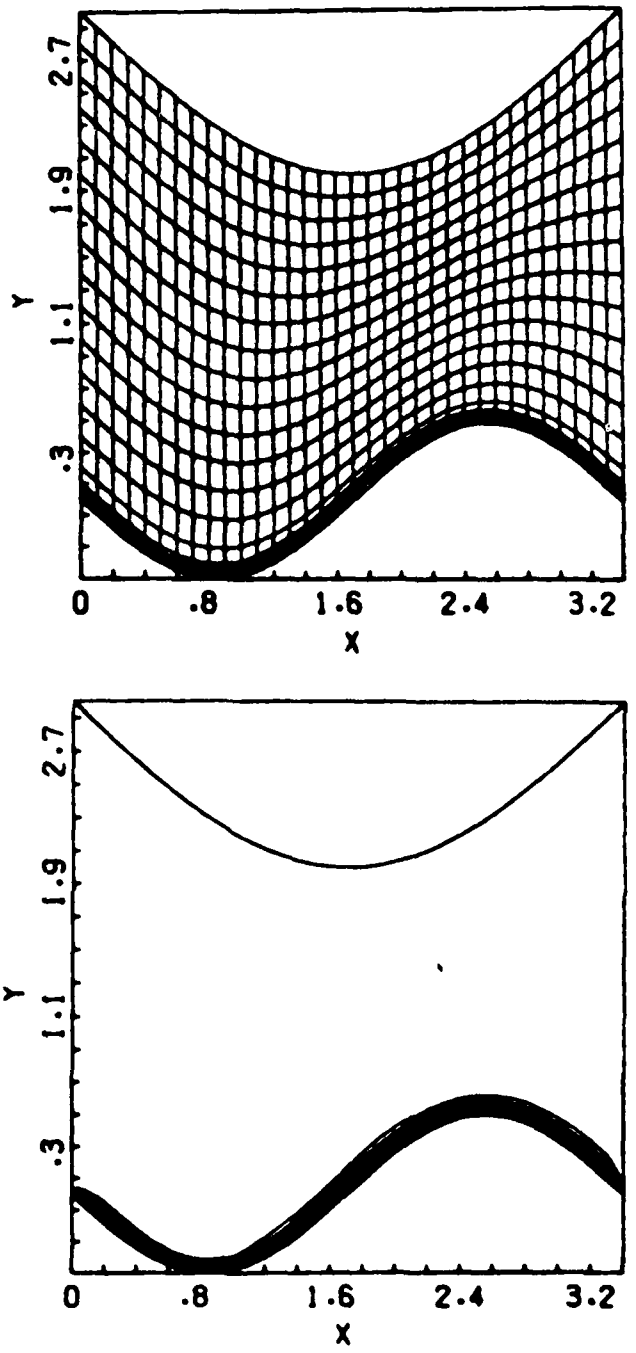


Fig. 4 Early time grid structure and isotherms

16. Klopfer, G.H. and McRae, D.S., "The Nonlinear Modified Equation Approach to Analyzing Finite-Difference Schemes, AIAA Paper 81-1029, presented at AIAA Computational Fluid Dynamics Conference, Palo Alto, California, June 1981.

17. Mastin, W. and Thompson, J., "Adaptive Grids Generated by Elliptic Systems," AIAA Paper 83-0451, presented at AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, January 1983.

18. Miller, K., "Moving Finite Elements. II," SIAM Journal of Numerical Analysis, Vol. 18, 1981, pp. 1033-1057.

19. Miller, K. and Miller, R.N., "Moving Finite Elements. I," SIAM Journal of Numerical Analysis, Vol. 18, 1981, pp. 1019-1032.

20. Rai, M.M. and Anderson, D.A., "Application of Adaptive Grids to Fluid-Flow Problems with Asymptotic Solutions," AIAA Journal, Vol. 20, April 1982, pp. 496-502.

21. Rai, M.M. and Anderson, D.A., "Grid Evolution in Time Asymptotic Problems," Journal of Computational Physics, 1981, pp. 327-344.

22. Rai, M.M. and Anderson, D., "The Use of Adaptive Grids in Conjunction with Shock-Capturing Methods," AIAA Paper 81-1012, presented at the AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, California, June 1981.

23. Saltzman, J. and Brackbill, J., "Applications and Generalizations of Variational Methods for Generating Adaptive Meshes," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 865-884.

24. Thompson, J.F., "A Survey of Grid Generation Techniques in Computational Fluid Dynamics," AIAA Paper 83-0447, presented at AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, January 1983.

25. Thompson, J.F., "General Curvilinear Coordinate Systems," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 1-30.

26. Thompson, J.F., "Elliptic Grid Generation," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 79-105.

27. Thompson, J.F., Warsi, Z.U.A., and Mastin, C.W., "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review," Journal of Computational Physics, Vol. 47, 1982, pp. 1-108.

28. White, A.B., Jr., "On the Numerical Solution of Initial/Boundary-Value Problems in One Space Dimension," SIAM Journal of Numerical Analysis, Vol. 19, 1982, pp. 683-697.

29. Winslow, A., "Numerical Solution of the Quasi-linear Poisson Equation," Journal of Computational Physics, Vol. 1, 1966, pp. 149-172.

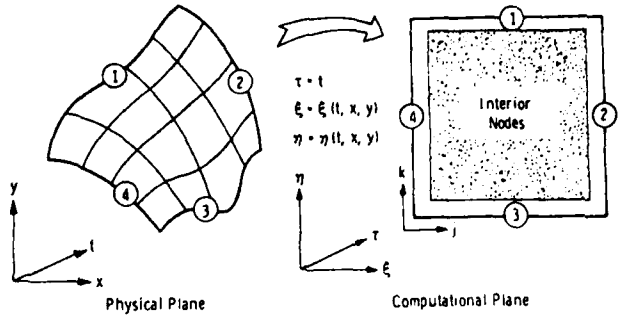


Fig. 1 Physical and computational domains

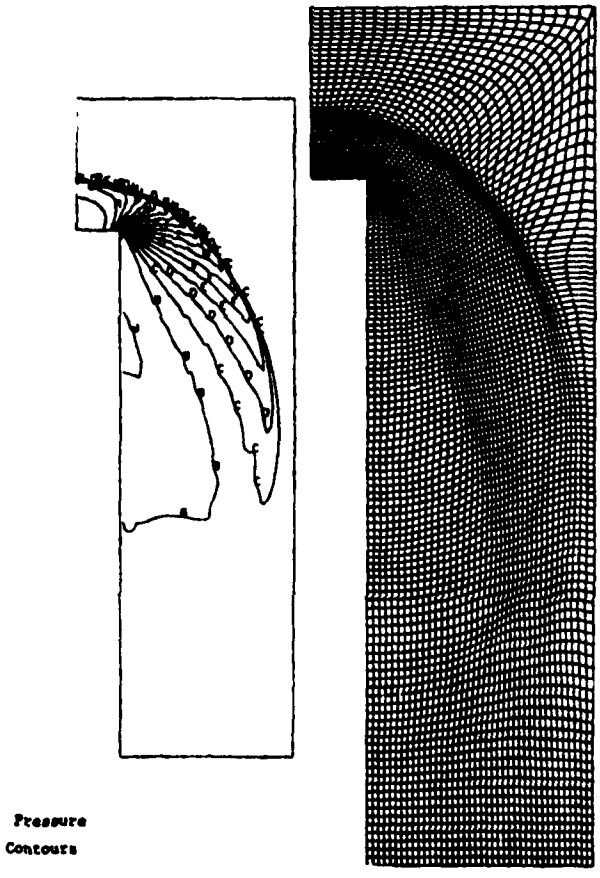


Fig. 2 Supersonic flow over forward step

of grid smoothness is invoked, an approximate expression for the error terms can be derived in terms of the coefficient matrices of the original system and the errors incurred in forming the first derivatives of the dependent variables. These error expressions are weighted averages of all the contributions of the governing equations for the fluid flow problem.

AREAS OF CONTINUED AND FUTURE RESEARCH

A number of problem areas need considerable work in order to create adaptive grid schemes which are readily applicable to a wide class of problems. These areas include the following:

1. Better estimates of numerical error are necessary for use in those adaptive methods which attempt to reduce the error in the solution.

2. Additional effort must be expended in gaining an understanding of how points are allocated in different regions for two- and three-dimensional problems for a given grid adaption scheme.

3. New methods for controlling point motion while still providing sufficient adaption should be developed for the various schemes.

4. Application of existing adaptive methods to three-dimensional problems should be encouraged because the largest gains in computational efficiency will be realized in these cases.

Research in adaptive grid schemes has really received emphasis during approximately the past five years. Fresh and innovative ideas are needed for construction of better approaches to the problems in this area. Researchers should be encouraged to thoroughly explore any idea that may have an impact on the development of new, successful techniques.

ACKNOWLEDGEMENTS

This work was partially sponsored by funds made available through U.S. Army Contract DAAK11-82-R-0123 and NASA Grant NGT 016-002-801.

Figures 4, 5, and 14 through 20 from cited references are reprinted with permission of the American Institute of Aeronautics and Astronautics.

Figures 2 and 3 are reprinted by permission of the publisher from the cited article in Numerical Grid Generation, Joe F. Thompson (ed.), North Holland 1982, copyright 1982 by Elsevier Science Publishing Co., Inc.

REFERENCES

1. Ablow, C.M., "Equidistant Mesh for Gas Dynamic Calculations," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 859-863.

2. Acharya, S. and Patankar, S.V., "Use of An Adaptive Grid for Parabolic Flows," AIAA Paper 82-1015, presented at the AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri, June 1982.

3. Anderson, D.A., "Solution Adaptive Grids for Partial Differential Equations," ARO Report 82-3, Proceedings of the 1982 Army Numerical Analysis and Computers Conference, Vicksburg, Mississippi, February 1982, pp. 575-591.

4. Anderson, D.A. and Rai, M.M., "A New Approach to Solution Adaptive Grids," Proceedings of the AIAA/ASME Symposium on Computers in Flow Predictions and Fluid Dynamics Experiments, November 1981, p. 95.

5. Anderson, D.A. and Rai, M.M., "The Use of Solution Adaptive Grids in Solving Partial Differential Equations," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 317-338.

6. Brackbill, J.U., "Coordinate System Control: Adaptive Meshes," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 277-294.

7. Brackbill, J. and Saltzman, J., "Adaptive Zoning for Singular Problems in Two Dimensions," Journal of Computational Physics, Vol. 46, June 1982, pp. 342-368.

8. Dwyer, H.A., "Grid Adaption for Problems with Separation, Cell Reynolds Number, Shock-Boundary Layer Interaction, and Accuracy," AIAA Paper 83-0449, presented at AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, January 1983.

9. Dwyer, H., Kee, R., and Sanders, B., "Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," AIAA Journal, Vol. 18, October 1980, pp. 1205-1212.

10. Dwyer, H.A., Smooke, M.D., and Kee, R.J., "Adaptive Gridding for Finite Difference Solutions to Heat and Mass Transfer Problems," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 339-356.

11. Gelinas, R.J. and Doss, S.K., "The Moving Finite Element Method: Applications to General Partial Differential Equations with Multiple Large Gradients," Journal of Computational Physics, Vol. 40, 1981, pp. 202-249.

12. Gnoffo, P.A., "A Vectorized, Finite-Volume, Adaptive-Grid Algorithm for Navier-Stokes Calculations," Numerical Grid Generation, Joe F. Thompson, editor, Elsevier Science Publishing Co., Inc., 1982, pp. 819-835.

13. Gnoffo, P.A., "A Vectorized, Finite-Volume, Adaptive Grid Algorithm Applied to Planetary Entry Problems," AIAA Paper 82-1018, presented at AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri, June 1982.

14. Hindman, R.G., Kutler, P., and Anderson, D., "Two-Dimensional Unsteady Euler-Equation Solver for Arbitrarily Shaped Flow Regions," AIAA Journal, Vol. 19, April 1981, pp. 424-431.

15. Hindman, R.G. and Spencer, J., "A New Approach to Truly Adaptive Grid Generation," AIAA Paper 83-0450, presented at AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, January 1983.

GRID POINT CONTROL AND GRID ADAPTION

One of the most difficult problems encountered in using adaptive grids is that of grid point control. This problem is not severe in one-dimensional adaption but is much more acute in multidimensional applications. In the variational formulation of Ref. 7, grid control was a consideration in the construction of the method. The problem remains in selecting the values of λ that are used to provide the contributions due to smoothness, orthogonality, and adaption. In order to provide sufficient adaption, high grid skewness may be encountered. Computed results show that a direct trade-off exists among the various terms in the grid generator. Once again, the problem of selecting certain parameters before the solution is obtained occurs.

In Dwyer's (8) recent paper, he describes a technique of predetermining the percentage of mesh points assigned to grid adaption. In this paper, a one-dimensional adaption technique was used similar to that in Eq. (29). The weight function used was of the form

$$w_1 = 1 + b\left|\frac{\partial f}{\partial s}\right| \tag{60}$$

where b is a constant and f is the function which is monitored and used for adaption. The ratio, R_1, is formed by computing the relative contribution of the grid adaption to the computational coordinate and is given by

$$R_1 = b\,\frac{\int_0^{S_{max}} \left|\frac{\partial f}{\partial s}\right| ds}{\int_0^{S_{max}} \left[1 + b\left|\frac{\partial f}{\partial s}\right|\right] ds} \tag{61}$$

If the ratio R_1 is held at a fixed value during a calculation and the value of b is determined from Eq. (61), the relative weight placed on adaptivity in the mesh remains fixed. This mechanism provides some control on the assignment of mesh points to different areas of interest in the problem. A similar division can be used in conjunction with the inverse adaption scheme of Eq. (24).

Grid point control for other schemes such as that of Rai and Anderson is achieved by logic built into the adaption algorithm as opposed to the governing equations. For example the constant P in the one-dimensional grid speed equation [Eq. (57)] is continuously adjusted during the numerical computations to prevent the maximum grid speed from exceeding a predetermined value. An additional constraint is placed upon the grid speeds when the grid points are closer than a specified value. To prevent excessive stretching, the grid speeds at those points are attenuated by the factor

$$x_t = x_{t_{cal}}\, e^{-1/x_\xi^2}$$

where $x_{t_{cal}}$ is the grid speed computed from the grid speed equation. This prevents grid speeds from being excessively large in regions where point density is high and provides good control of the grid point motion.

Another question closely associated with grid point control is that of attempting to define appropriate functions to use in the adaption process. For example, what is the best choice of the weighting function in the variational schemes and what is the best choice for e in Eq. (57)? This question is not easily answered even when a single scalar equation is used.

In viscous problems, resolution of viscous regions is probably best accomplished by keeping the cell Reynolds number less than one. This provides a well defined viscous layer where first-order upwind schemes will not produce large artificial diffusion and an oscillation free solution will be obtained when higher-order schemes are used. In principle, any function which would permit clustering, such as gradient information should provide adequate adaption. Numerous researchers have used second derivative information. In Ref. 8 a nonlinear combination of first and second derivatives has been used while second derivatives have also been used in Refs. 5 and 21. White (28) has used curvature of the solution for the clustering function and has shown good results for a scalar equation. It should be noted that a second derivative clustering function can be viewed as a first approximation to the curvature.

For each grid adaption scheme and each problem, the choice of a grid adaption function may be different. For example, the choice of gradient alone for the weight function w_1 in Eq. (22) would not be satisfactory. Methods based upon this expression would produce grids with no mesh points in uniform regions. On the other hand, gradient may work well in the two-dimensional application of Eq. (18) because measures of grid smoothness and orthogonality are also included.

As previously noted, the lowest-order error term of the modified partial differential equation was used to drive the grid in Refs. 20, 21, and 22. This error term involves a third derivative assuming a second-order method is used. Numerical evaluation of the third derivative (or any higher-order derivative) is generally very noisy. As a result, data used to evaluate derivatives is smoothed to prevent feedback into the grid motion. If the assumption of a very smooth grid is made, the error can frequently be approximated with lower-order derivatives.

When systems of equations are solved or physical problems are studied where the resolution of more than one physical event is necessary, additional difficulties are encountered. In Ref. 8, the problem of flame propagation about a spherical particle in the presence of a low Reynolds number Stokes flow was studied. In this case, resolution of both the viscous regions and the laminar flame zone is desired. The grid adaption functions should include contributions which would resolve both the flame front and the viscous region. In this case, the problem was solved on two grids and the solution was determined by interpolating between them. The two grids showing details of each region are shown in Figs. 19 and 20.

A one-dimensional inviscid shock tube problem was studied in Ref. 16. In this example, a reduction of the truncation error in the modified differential equation was desired. An adaptive grid was used and the grid driving function (w_1) was selected to be a linear combination of the error terms from each of the governing equations. A more analytical approach to the derivation of the function used to drive the adaptive grid was taken in Ref. 5. The Euler equations in multidimensional space were under study and expressions for the error terms in Eq. (58) were needed. When the assumption

a rectangle in computational space. Since the grid is three sided in physical space, a geometric singularity is shown on the shock wave. It should be remembered that the grid point locations for this scheme are computed by integrating the grid speeds instead of solving the steady grid equation.

Hindman and Spencer (15) have continued this approach and have considered the one-dimensional grid equation

$$x_{\xi\xi} + x_{\xi}^{3} P = 0 \tag{54}$$

Again, the grid speeds are established by differentiating this expression with respect to τ. The P function was selected (at least for one case) so that Eq. (51) was consistent with Eq. (42). The form for P becomes

$$P = \frac{a g_{\xi}}{1 + \eta} \frac{1}{x_{\xi}^{2}} \tag{55}$$

Figure 14 shows the time history of the grid motion when the inviscid Burgers' equation is solved using a single discontinuity for initial conditions. MacCormack's scheme was used to integrate both the equations of motion and the grid speeds. The main result of interest in this figure is that the grid tends to relax as time increases. This can be avoided by solving the steady grid equation after a predetermined number of steps and interpolating the solution on the new grid.

Rai and Anderson (20,21,22) and Anderson and Rai (3,4,5) have constructed an adaptive grid scheme using a different approach from those discussed above. This method also determines the grid speed and the grid point locations are established by integration. A much simpler grid speed expression than that used in Ref. 14 was developed. The method is based upon an attraction model. It is assumed that the best grid for a given problem is one where the solution error at every point reaches a constant value. In creating this grid, more points are needed in regions where the error is larger than the average and fewer points are needed in regions where the local error is smaller than the average value. This leads directly to the idea of associating a capacity to induce velocity with the local error at each point. For two points, A and B, the grid speed induced at point B due to an error at point A is written

$$V_{BA} = K \frac{|e|_A - |e|_{av}}{r_{BA}^{n}} \tag{56}$$

where e represents the local error, av indicates average value over all points, r_{BA} is the distance from A to B in computational space, K is a proportionality constant, and n is a power which controls the attenuation of the attraction with distance. For a one-dimensional problem, the grid speed in physical space becomes

$$x_{\tau_j} = \frac{K}{\xi_{x_j}} \left[\sum_{i=j+1}^{N} \frac{(|e|_i - |e|_{av})}{r_{i,j}^{n}} - \sum_{i=1}^{j-1} \frac{(|e|_i - |e|_{av})}{r_{i,j}^{n}} \right] \tag{57}$$

Grid point positions produced by integrating Eq. (57) must not cross. In this scheme, grid points do not cross for sufficiently small time steps because the local value of the error will be less than the average value if points are very close together. This creates a sign switch on the grid speed (repulsion). As points approach each other the term ξ_x becomes very large which makes further movement of the grid points very difficult. The constant, K, in Eq. (57) is adjusted to scale the grid speed to a predetermined maximum value during the calculations.

The driving function used to establish grid point motion using this scheme ($|e| - |e|_{av}$) can be based on error or the gradient of any flow variable or any other function which provides the desired attraction. A number of error measures were used to drive the grids in Refs. 5 and 21. In general, the form of the error was established by using an approximation to the lowest-order error term of the modified partial differential equation. Thompson (24) suggests that the grid solution obtained using Eq. (57) is equivalent to solving a variational problem by an iterative approach. In this case, the variation of $|e| - |e|_{av}$ is minimized over the field. This corresponds to the weight function w_1 in Eq. (22).

In order to provide a smooth grid at the boundaries, a reflection at the boundaries in the computational domain was used. For example, at the right-hand boundary, $\xi = 1$ and $\xi_t = 0$. However, if this condition is explicitly used, sometimes the grid will not be smooth near this boundary. If a series of points are reflected and have an error measure assigned to them in such a way that $\left(\xi_t\right)_{\xi=1} = 0$ when computed from the grid equation, the boundary region will have a very smooth grid. The influence of this reflection is in the value of η_t along the boundary. This value and the resulting point locations are very sensitive to boundary point treatment.

Figure 15 shows the converged adaptive grid generated for the supersonic blunt body problem in two dimensions. Figure 16 shows the difference between the total enthalpy at the cylinder surface as computed from the numerical solution and the free stream value. Since this is an inviscid calculation, this is a measure of the solution accuracy. Error in total enthalpy is significantly reduced when an adaptive grid is used.

A technique for aligning a grid with a high gradient region is presented in Refs. 5 and 22. This technique was developed for use with shock capturing methods. It is well known that shock aligned coordinate systems permit much better computation of shocks with these methods because the flux terms are then continuous across the shocks. Shock alignment is accomplished by creating a grid speed in a two-dimensional problem along only one of the coordinates. This grid speed is proportional to the product of the gradient of some property of the solution (density, pressure) along both the ξ and η directions. The result is an effective rotation of the coordinate line segments in such a way as to line up with level surfaces of h. Again, only movement in one coordinate direction is necessary in a two-dimensional problem. Figure 17 shows the converged grid for an oblique shock and the pressure distribution for both shock capturing on a uniform grid and an aligning grid is shown in Fig. 18. The quality of the solution is dramatically improved when the aligning grid is used.

where it is assumed that

$$|q_\xi|\Delta\xi - 2g > 0$$

In regions where this is not true, any positive value of a satisfies the condition on grid crossing.

In Ref. 17 an adaptive grid generation scheme is proposed which for one-dimensional problems is of the form

$$x_{\xi\xi} + AT_\xi x_\xi^2 = 0 \tag{44}$$

where A is a positive constant and T is some nonnegative function of the solution. In the context of Eq. (22), this can be interpreted as a scheme which is similar to either Dwyer's or Gnoffo's method with the weight function defined as

$$w_1 = e^{\int_0^\xi AT_\xi x_\xi d\xi} \tag{45}$$

The two-dimensional formulation of Eq. (44) is of the form

$$\nabla^2\xi = AS_\xi$$
$$\nabla^2\eta = BT_\eta \tag{46}$$

The equations which are solved in the computational domain are

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2[AS_\xi x_\xi + BT_\eta x_\eta] = 0$$
$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2[AS_\xi y_\xi + BT_\eta y_\eta] = 0 \tag{47}$$

where the coefficients α, β, and γ are functions of the metrics, and S and T are the grid adaption functions. As in the one-dimensional case, an estimate of the upper limit for A and B can be established by requiring that grid lines be noncrossing. These estimates are

$$A < 2/(\gamma|S_\xi|\Delta\xi)$$
$$B < 2/(\alpha|T_\eta|\Delta\eta) \tag{48}$$

The potential flow about a cylinder was computed in Ref. 17 as an example using the adaptive grid scheme in Eq. (46). Figure 11 shows the error between the exact and computed solutions for the surface potential for a nonadaptive conformal grid and an adaptive grid. In general, an adaptive grid provides solutions which exhibit lower error. In this case the S and T functions were defined as

$$S = (\phi - x)_{\xi\xi}$$
$$T = (\phi - x)_{\eta\eta}$$

and ϕ is the disturbance potential.

All of the methods discussed thus far have relied upon solving a steady grid equation to determine the grid point location. The grid speed is determined by using a backward difference using the computed mesh point location and the previously known positions. Hindman et al. (14) developed a technique of computing the grid speed directly from the grid generator. The idea is to evaluate the time derivative of the steady grid generation equation and solve this equation for the grid speeds. In principle, this idea is applicable to the methods for creating an adaptive grid presented above.

If the Thompson scheme is used as a starting point, the equations which determine the mapping relating physical and computational space are

$$\nabla^2\xi = P$$
$$\nabla^2\eta = Q \tag{49}$$

In the computational domain these equations become

$$G[x] = 0$$
$$G[y] = 0 \tag{50}$$

where the operator G is defined by

$$G = \alpha\frac{\partial^2}{\partial\xi^2} - 2\beta\frac{\partial^2}{\partial\xi\partial\eta} + \frac{\partial^2}{\partial\eta^2} + J^2\left[P\frac{\partial}{\partial\xi} + Q\frac{\partial}{\partial\eta}\right] \tag{51}$$

and α, β, and γ are the usual functions of the metrics. Solution of Eq. (46) provides the grid point locations at any time. The grid point speeds of boundary points are usually obtained from shock relations or other expressions which must be satisfied. The interior grid speeds (x_τ, y_τ) could be obtained from backward differences as noted previously, but a better approach may be to differentiate Eqs. (50) with respect to τ. This yields a system of equations of the form

$$[M]\vec{Z} = \vec{r} \tag{52}$$

where

$$\vec{Z} = (x_\tau, y_\tau)^T$$

and

$$\vec{r} = -J^2(P_\tau x_\xi + Q_\tau x_\eta, P_\tau y_\xi + Q_\tau y_\eta)^T \tag{53}$$

The solution of Eq. (52) yields the interior point grid speed necessary to advance the governing equations of the fluid flow problem. If P and Q are selected to be functions of the solution of the flow equations, the grid adjusts adaptively through the time derivatives of these terms. The boundary grid point speeds influence the interior solution through the boundary conditions on the system of equations.

In Ref. 14, the P and Q functions were set equal to zero providing grid and grid speed solutions which did not adapt to internal flow changes but only to the boundary motion. Figure 12 shows the geometry of the grid produced for solving the problem of a planar shock wave passing over an inclined ramp. Figure 13 shows the grid used to solve for the inviscid supersonic flow over an ogive. This grid in physical space is mapped into

$$\zeta_s = \frac{y_\eta}{\left[x_\eta^2+y_\eta^2\right]}\left[\frac{(x_\xi^2+y_\xi^2)(x_\eta^2+y_\eta^2) - (x_\xi x_\eta + y_\xi y_\eta)}{\sqrt{x_\xi^2+y_\xi^2}}\right] \quad (32)$$

Replacing the derivative with respect to s in Eq. (31) results in the expression

$$\left[(x_\eta^2+y_\eta^2)\frac{\partial}{\partial\xi} - (x_\xi x_\eta + y_\xi y_\eta)\frac{\partial}{\partial\eta}\right]\left[\frac{\xi_s}{w_1}\right] = 0 \quad (33)$$

This expression can be expanded and becomes a second-order differential equation in x and y. The companion relationship along constant ξ surfaces can be derived in the same manner.

The inverse relationship corresponding to Eq. (29) may be written

$$s = s_{max}\frac{\int_0^\xi \frac{1}{w_1}\,d\xi}{\int_0^{\xi_{max}} \frac{1}{w_1}\,d\xi} \quad (34)$$

The differential equation satisfied by S corresponding to Eq. (30) is

$$\frac{\partial}{\partial\xi}(w_1 S_\xi) = 0 \quad (35)$$

or

$$S_{\xi\xi} + \frac{S_\xi w_{1\xi}}{w_1} = 0 \quad (36)$$

In this case, the partial differential equation which must be satisfied is easily recovered using the fact that

$$S_\xi = \sqrt{x_\xi^2+y_\xi^2} \quad (37)$$

The governing expression becomes

$$x_\xi x_{\xi\xi} + y_\xi y_{\xi\xi} + (x_\xi^2+y_\xi^2)\frac{w_{1\xi}}{w_1} = 0 \quad (38)$$

While the author has not carried out the algebraic manipulations (lazy) required in Eq. (33), it is not apparent that the result would be equivalent to Eq. (38).

It should be noted that Eq. (38) is the expression which is valid along an η = constant surface. The companion expression valid along a constant ξ surface may be written directly as

$$x_\eta x_{\eta\eta} + y_\eta y_{\eta\eta} + (x_\eta^2+y_\eta^2)\frac{w_{2\eta}}{w_2} = 0 \quad (39)$$

It is of interest to consider the grid adaption scheme resulting from the simultaneous solution of Eqs. (38) and (39) and some examples were studied. The weight functions w_1 and w_2 were selected to be of the form

$$w_1 = 1 + a|u_\xi| \quad (40a)$$

$$w_2 = 1 + b|u_\eta| \quad (40b)$$

and the grid adaption was implemented for a specified function u(x,y) on a grid which is 21 x 21 in dimension.

Figure 6 shows the results for specifying u(x,y) to be a function of x only. As expected, the grid adjusts only in the x direction and no changes occur along the y coordinate. The clustering is achieved using u(x,y) = 1 for $x \leq 9$ and u(x,y) = 0 for $x \geq 11$. The function, u(x,y), was selected to decrease linearly from one to zero from x = 9 to x = 11. The two bands of clustering which appear in Fig. 6 are due to averaging used on the grid point calculation. The averaging gives the effect of using a second derivative for clustering in this case. If no averaging is used, the grid clusters between x = 9 and x = 11 where the derivative $\partial u/\partial x$ attains a maximum.

Figure 7 shows the grid resulting from applying the adaption routine for a u(x,y) which varies linearly from one to zero but this variation occurs about the symmetric straight line

$$y = 2.5x - 15$$

passing through the domain. The value of u is taken to be either zero or one outside this small two-unit wide region. In Fig. 7, b = 0 and the attraction is only along the η = constant surfaces. In this case, we see that the adaption in the η direction also creates a contraction along the ξ = constant lines. This is a result of the coupling between the two directions. A case which includes adaption in both directions is shown in Fig. 8. Additional cases with the corresponding u(x,y) are shown in Figs. 9 and 10. The function, u(x,y), was set equal to one above and zero below a sinusoidal surface which shows up very well in these grid plots. This function again decreased from one to zero over a two Δy interval in the y direction. The grid distortion problems inherent with this scheme are apparent in these results. Notice that the grid becomes distorted all along the dividing surface of u(x,y) but is much worse where the sine wave slope is roughly at equal angles to the x and y axes. At points in this neighborhood, clustering along both directions creates severe strain of the area elements and leads to the distortion shown in Figs. 9 and 10.

The adaption process works well for these cases, but it is difficult to estimate the proper values of a and b unless some numerical experiments are performed. It should be noted that the definition of the transformation implies that arbitrarily large values of a and b can be used. Due to the discretization of the differential equation, the clustering constants a and b can not be arbitrarily large. Mastin and Thompson (17) have shown that for a one-dimensional problem

$$|x_{\xi\xi}| < 2|x_\xi|/\Delta\xi \quad (41)$$

in order to prevent the grid lines from crossing. For the differential equation

$$x_{\xi\xi} + \frac{ag_\xi x_\xi}{1+ag} = 0 \quad (42)$$

this leads to a bound for the constant, a, which may be written

$$a \leq \frac{2}{|g_\xi|\Delta\xi - 2g} \quad (43)$$

Many applications require grid adaption in only one dimension. For this reason, it is worthwhile to consider minimizing the functional, I_v, defined in Eq. (14) specialized to one dimension.

$$I_v = \int w x_\xi d\xi \tag{19}$$

The Euler-Lagrange equation may be written

$$\left(\frac{\partial}{\partial \xi} - \frac{d}{dx}\frac{\partial}{\partial \xi_x}\right)\left(\frac{w(x)}{\xi_x}\right) = 0 \tag{20}$$

In this case, a first integral may be directly written as

$$w x_\xi^2 = \text{constant} = C_1 \tag{21}$$

or

$$\sqrt{w}\, x_\xi = w_1 x_\xi = C \tag{22}$$

This expression states that the product of the mesh spacing and the weight function, w_1, should remain constant in physical space. Equation (22) may be integrated to obtain the expression for either the physical coordinate in terms of the computational coordinate or vice versa. Let $x = 0$ when $\xi = 0$ and $x = x_{max}$ when $\xi = \xi_{max}$. If Eq. (22) is integrated and the computational coordinate is determined,

$$\xi = \xi_{max} \frac{\int_0^x w_1 dx}{\int_0^{x_{max}} w_1 dx} \tag{23}$$

and if the physical coordinate is evaluated one obtains the equation

$$x = x_{max} \frac{\int_0^\xi \frac{1}{w_1} d\xi}{\int_0^{\xi_{max}} \frac{1}{w_1} d\xi} \tag{24}$$

Equation (23) is exactly the expression used by Dwyer (8) and Dwyer et al. (9,10). Many of the applications of this law for grid adjustment have been in combustion and heat or mass transfer and the results have been very good. In Dwyer's formulation, the weighting function w_1 was selected to be a linear combination of derivatives of temperature or some other pertinent variable of interest, i.e.

$$w_1 = 1 + ag = 1 + a\left[\left|\frac{\partial T}{\partial x}\right| + b\left|\frac{\partial^2 T}{\partial x^2}\right|\right] \tag{25}$$

The results of applying Eq. (23) with the weighting function of Eq. (25) with $b = 0$ from Ref. 10 is shown in Figs. 4 and 5. In this problem, the equation for unsteady heat conduction was solved on the domain shown with the temperature set equal to zero everywhere. At $t = 0$, the temperature was raised impulsively to a constant value on the lower boundary and held at a fixed value. Figure 4 shows the isotherms and the grid at an early time showing the high temperature gradients near the lower boundary. As time increases, the heat flow into the domain can be observed in both the isotherms and the grid geometry of Fig. 5.

Equation (24) is the integral form of the discrete adaptive grid scheme proposed by Gnoffo (12,13). Gnoffo has used this adaptive grid generator while solving the Navier-Stokes equations with a weight function based upon monitoring either Mach number or velocity gradient, i.e.

$$w_1 = 1 + ag \tag{26}$$

where

$$y = \left|\frac{dM}{dx}\right| \tag{27}$$

or

$$g = \left|\frac{dV}{dx}\right| \tag{28}$$

Calculations of the flow over the Viking Aeroshell vehicle using this scheme are presented in Ref. 13. Good results were obtained in the cases reported and sufficient grid adaption was used to adequately resolve the viscous layer.

The adaptive grid calculations cited from Refs. 10 and 13 are for adaption along either a constant ξ line or a constant η line. These one-dimensional calculations can be achieved using either Eq. (23) or Eq. (24) if x is replaced by S where S is arc length along either constant ξ or constant η lines. In addition, S can only depend upon one computational coordinate. For instance, if S represents arc length along a constant η line, then S is assumed to only be a function of ξ. In this case, the Dwyer method and Gnoffo's approach are identical. However, if arc length along ξ or η surfaces in physical space is considered to depend upon both coordinates, these methods are not the same.

In particular, suppose the analog of Eq. (23) along an η = constant surface is written

$$\xi = \xi_{max} \frac{\int_0^S w_1 ds}{\int_0^{S_{max}} w_1 ds} \tag{29}$$

If a derivative is taken with respect to S, this expression becomes

$$\frac{\xi_s}{w_1} = \frac{\xi_{max}}{\int_0^{S_{max}} w_1 ds} \tag{30}$$

where the right-hand side only depends upon the arc measure along the constant ξ surfaces. Differentiating again yields

$$\frac{\partial}{\partial s}\left(\frac{\xi_s}{w_1}\right) = 0$$

or

$$\xi_{ss} = -\frac{\xi_s w_{1s}}{w_1} \tag{31}$$

The identity relating ξ_s and the usual coordinate metrics is

ADAPTIVE GRID SCHEMES

Brackbill and Saltzman (6,7) and Saltzman and Brackbill (23) have extended Winslow's method (29) for generating a computational mesh to include grid adaption. In this approach, a variational technique is used to minimize a linear combination of a measure of grid smoothness, orthogonality, and volume (area) variation. The smoothness is measured by integrating the change of the computational coordinates over the physical domain and for two-dimensions may be written

$$I_s = \int_D [(\nabla\xi)^2 + (\nabla\eta)^2]dV \tag{7}$$

where dV represents differential volume in physical space and D denotes the physical domain. The smoothest mapping between the physical and computational domains is obtained by minimizing I_s alone. The result is that Laplace's equation for the computational coordinates, ξ and η, must be solved to determine the transformation.

The transformation which minimizes I_s is obtained by solving the Euler-Lagrange equations. Equation (7) is first written as an integral in computational space using the identities

$$\xi_x = y_\eta/J\ ,\quad \xi_y = -x_\eta/J\ ,\quad \eta_x = -y_\xi/J\ ,\quad \eta_y = x_\xi/J \tag{8}$$

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{9}$$

Equation (7) then becomes

$$I_s = \int_{D_1} \frac{(x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2)d\xi d\eta}{J} \tag{10}$$

where D_1 indicates integration over the computational domain. The Euler-Lagrange equations for this variational problem are

$$\left(\frac{\partial}{\partial x} - \frac{\partial}{\partial\xi}\frac{\partial}{\partial x_\xi} - \frac{\partial}{\partial\eta}\frac{\partial}{\partial x_\eta}\right)\left(\frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J}\right) = 0 \tag{11}$$

and

$$\left(\frac{\partial}{\partial y} - \frac{\partial}{\partial\xi}\frac{\partial}{\partial y_\xi} - \frac{\partial}{\partial\eta}\frac{\partial}{\partial y_\eta}\right)\left(\frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{J}\right) = 0 \tag{12}$$

If the indicated differentiations are carried out in Eqs. (11) and (12), the familiar form of the LaPlace grid generator in the computational domain is obtained.

In addition to a smoothness requirement, control of mesh skewness is obtained if a measure of grid orthogonality is included. This orthogonality measure is provided by the integral

$$I_o = \int_D (\nabla\xi \cdot \nabla\eta)^2 J^3 dv \tag{13}$$

or in computational space

$$I_o = \iint_{D_1} (x_\xi x_\eta + y_\xi y_\eta)^2 d\xi d\eta \tag{14}$$

The grid adaption is provided by minimizing a weighted average of the volume variation over the field and an appropriate integral is

$$I_v = \int_D wJdV \tag{15}$$

or

$$I_v = \int_{D_1} wJ^2 dV \tag{16}$$

where w is the weighting function which produces the grid adaption. Clearly, an adaptive system is obtained if the product of a positive weight function and the cell area is held at a fixed value over the physical domain.

In order to incorporate smoothness, orthogonality and adaption in the grid generator, the linear combination of the integrals given in Eqs. (10), (14), and (16) is written as

$$I_t = I_s + \lambda_v I_v + \lambda_o I_o \tag{17}$$

If the Euler-Lagrange equations for minimizing Eq. (17) are derived, they are of the form

$$b_1 x_{\xi\xi} + b_2 x_{\xi\eta} + b_3 x_{\eta\eta} + a_1 y_{\xi\xi} + a_2 y_{\xi\eta} + a_3 y_{\eta\eta} = -\frac{J^2}{2w}\frac{\partial w}{\partial x} \tag{18a}$$

$$a_1 x_{\xi\xi} + a_2 x_{\xi\eta} + a_3 x_{\eta\eta} + c_1 y_{\xi\xi} + c_2 y_{\xi\eta} + c_3 y_{\eta\eta} = -\frac{J^2}{2w}\frac{\partial w}{\partial y} \tag{18b}$$

where the a_i, b_i, and c_i are functions of the weighting coefficients, λ_v and λ_o, and the metric coefficients of the transformation. The values of a_i, b_i, and c_i are given in Ref. 7 and are not repeated here. The mesh which is generated when Eqs. (18) are solved can be varied by adjusting the weighting coefficients. For example, a large value of λ_v will provide more adaption in the grid with less emphasis on orthogonality and smoothness.

Figures 2 and 3 show the pressure contours and the adaptive grid for the inviscid supersonic flow over a forward facing step. The similarity between the pressure contours and the mesh is quite good. In this case, the weight function for grid adaption was taken to be the square of the magnitude of the pressure gradient divided by the pressure. If one is interested in tracking a shock, this is a reasonable choice. Strictly speaking, the shock wave in an inviscid flow is a discontinuity and even with grid adaption, the shock mathematically still should occur between two grid points. Of course, the shock is given a pseudo-viscous profile through the introduction of artificial viscosity either by smoothing the data or by the form of the algorithm. In this case a flux-corrected transport scheme was used. This provides a smooth shock profile in the solution.

The extension of the variational approach to three-dimensional problems is straightforward. Additional terms must be added to include the third dimension but conceptionally the approach is the same. Saltzman and Brackbill (23) have presented several examples of three-dimensional adaptive grid calculations and include mesh generation for a wing-fuselage junction.

another good reason for using an adaptive grid. Of course, the ideas of resolving regions of rapid change in dependent variable and reducing error are not mutually exclusive. The largest numerical errors are usually found in regions where the solution is changing most rapidly.

The idea of an adaptive grid implies that the solution of a partial differential equation is being computed using some sort of iterative or marching technique For hyperbolic or parabolic problems, the solution is computed by advancing in space or time and adjusting the grid as the solution progresses. For elliptic problems, a relaxation procedure provides intermediate results which are used to adjust the grid point positions. A simple linear equation can be used to illustrate some of the considerations that must be made in employing a solution adaptive grid.

When a partial differential equation is transformed from physical to computational space, the metrics of the transformation and the grid speeds appear in the equation. These additional terms must be evaluated in order to solve the differential equation written in computational coordinates. As an example, the first-order wave equation may be written

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = 0 \tag{1}$$

where u(t,x) is the unknown dependent variable and c is a constant wave speed. For this one-dimensional example, let the transformation relating the physical and computational domains be written

$$\begin{aligned} \tau &= t \\ \xi &= \xi(x,y) \end{aligned} \tag{2}$$

where τ and ξ are coordinates in the computational domain. Transforming the wave equation into computational coordinates yields

$$\frac{\partial u}{\partial \tau} + (\xi_t + c\xi_x)\,\frac{\partial u}{\partial \xi} = 0 \tag{3}$$

The terms ξ_x and ξ_t may be replaced by the expressions

$$\xi_t = -x_\tau / x_\xi \tag{4}$$

and

$$\xi_x = 1/x_\xi \tag{5}$$

The original expression may then be written

$$\frac{\partial u}{\partial \tau} - \frac{(x_\tau - c)}{x_\xi}\,\frac{\partial u}{\partial \xi} = 0 \tag{6}$$

When a solution of Eq. (6) is computed using a fixed grid, the metric coefficient, x_ξ is determined from the grid geometry which is established at the beginning of the calculations. This term does not change so long as a fixed mesh is used. The grid speed in physical space, x_τ is zero in this case. When an adaptive grid is used, the grid speed is nonzero because the grid changes as the calculation proceeds and the metric coefficient also changes each time the grid is altered.

The metric represents the ratio of arc lengths in the physical and computational planes and the grid speed provides the dynamic coupling of the moving grid with the evolving solution of the differential equation. Any method for constructing an adaptive grid must provide a means of estimating these terms since they explicitly appear in the transformed differential equation. Exceptions to this can be cited. In problems where a time-asymptotic solution is computed, the grid can be fixed and the grid speed term set equal to zero. After a predetermined number of iterations, the computational mesh is adjusted and the iteration process is resumed. This regridding procedure is equivalent to solving a sequence of initial boundary value problems. Other exceptions include cases where the solution changes by an extremely small amount over a single iteration and the grid speed is set equal to zero in the governing differential equation. However, both the grid speed and metric term should be included in the general case.

Adaptive grid methods can be divided into two categories. In the first category, some law relating the grid points in the physical and computational domain is used to establish new physical grid point locations (x's) at the end of each time step. The grid speed term is estimated for the next integration step by using a simple backward difference. The second class of schemes relies on directly establishing the grid speed by some rule. The grid speed is integrated along with the differential equation and the new grid point positions are established. From this information, the metrics are computed by evaluating the ratio of arc lengths in the physical and computational domains.

There are advantages and disadvantages to both approaches for generating an adaptive grid. Methods which directly generate the new coordinates through a defined mapping are conceptually easy to apply. Since the grid point locations are established through the use of a steady grid equation, the grid speeds are most easily determined by using a backward difference. This difference is usually first order in time and more accurate dynamic coupling of the grid motion and the partial differential may be desirable. Some grid point location schemes initially developed for one-dimensional applications are difficult to extend to two- or three-dimensional problems. Techniques which directly determine grid speed from some grid speed law are easily used in multidimensional applications because grid point location is determined by a simple integration. The major disadvantage is that physical laws which relate grid speed and grid adaption may be difficult to formulate and the success of the method depends upon the ingenuity used in constructing this law. For these schemes, point control is also a problem.

A number of adaptive grid generation schemes in both classes have been developed. These methods have been used with success on a variety of problems. In the following sections, a number of the most successful schemes from both categories are reviewed. In this discussion, the problem of constructing an adaptive grid is viewed as one of allocation. How should a fixed number of grid points be distributed to improve the quality of a numerical solution? This distribution of mesh points is influenced by both motion of the boundaries and solution changes on the interior of the physical domain. The main focus of interest in this paper is on grid point motion caused by solution changes on the interior.

1983 ASME Applied Mechanics,
Bioengineering and Fluids
Engineering Conference
June 20-22 Houston, TX

ADAPTIVE GRID METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

Dale A. Anderson
Department of Aerospace Engineering and Computational Fluid Dynamics Institute
Iowa State University
Ames, Iowa

ABSTRACT

A number of techniques of constructing adaptive mesh generators for use in solving partial differential equations are reviewed in this paper. Techniques reviewed include methods based on steady grid generation schemes and those which are explicitly designed to determine grid speeds in a time-dependent or space-marching problem. Results for candidate methods are included and suggestions for areas of future research are suggested.

INTRODUCTION

The numerical solution of the partial differential equations governing both internal and external flows has reached a high state of development during the past fifteen years. Numerical methods for solving the different types of equations have been available for much longer. However, the ability to treat complex geometries common to most physical problems has only recently been acquired. In fact, the pacing item in advancing numerical procedures for solving fluid flow and heat transfer problems has been the development of general techniques for numerically constructing mesh systems which are boundary conforming.

The construction of a suitable grid is the first task that must be completed when the numerical solution of a system of partial differential equations is desired. Once the grid is generated, the system of equations is discretized and the resulting system of algebraic equations is solved. This solution yields the values of the dependent variables at each of the mesh points. The solution of the governing equations is completed in a computational domain which is selected to be rectangular shaped for simplicity. The physical and computational domains are related through a mapping as schematically illustrated in Fig. 1 for a two-dimensional case. The problem of numerical grid generation is concerned with techniques for establishing this relationship between physical and computational space. Thompson (24) and Thompson et al. (27) have presented a comprehensive review of the state of the art in numerical grid generation.

In solving partial differential equations using numerical methods, the selection of the locations of the mesh points is important in establishing the quality of the solution. These grid point positions are generally determined initially and remain fixed throughout the calculation. In order to determine the best grid point locations, an a priori knowledge of the solution of the physical problem is desirable. Unfortunately, this knowledge is unavailable and only the general features of the solution may be initially understood. For example, flow over a body must be computed with a grid employing a sufficient number of points in the viscous regions to resolve the salient features of the flow. In this case, a grid may be constructed using a compression mapping in order to provide a large number of points in the viscous layer near the body. High mesh densities are desirable in regions where large gradients exist. Since the exact location and size of these regions is initially unknown, the construction of a suitable grid in the general case is difficult and some means of incorporating information from the solution in locating the grid points is needed.

The concept of a solution adaptive grid is appealing for a number of reasons. In many problems, multiple length scales appear and a grid which resolves a physical process scaled to one significant length can't resolve events which occur on a scale less than the size of the smallest cell or mesh increment. A typical example is again provided by the flow of a viscous fluid over a body. The inner flow near the body in the boundary layer will be resolved in sufficient detail to be of use only if very small grid spacing is used. If mesh spacing is used with the minimum size determined by that required to resolve the outer inviscid flow, the detail of the boundary layer is completely lost. Heat transfer and skin friction data obtained from such a calculation are completely meaningless. With the use of an adaptive grid, the physical behavior of the fluid in both regions can be adequately established using the same set of governing equations. The different length scales in the problem are accommodated by a variable mesh size. In a sense, this approach is analogous to classical methods which require a solution of the inner and outer flow with appropriate matching conditions. Two sets of governing equations must be solved while numerically, a single set of governing equations is solved, but the grid position problem must also be treated.

When a partial differential equation is discretized, errors are present in the computed solution. If the mesh points are adjusted during the calculation to reduce some measure of the local or global error, the quality of the solution will be improved. This is

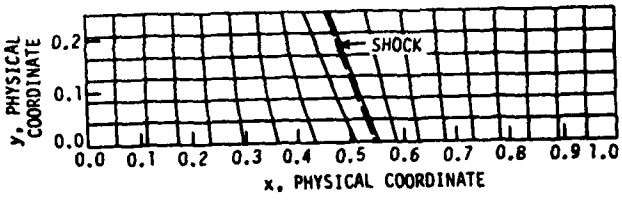


Fig. 17 Shock aligned grid for a two-dimensional plane shock problem

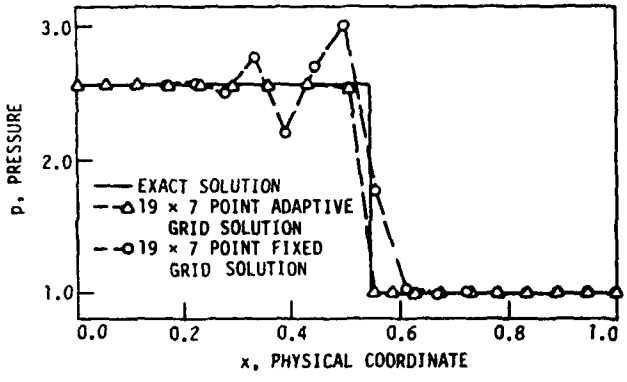
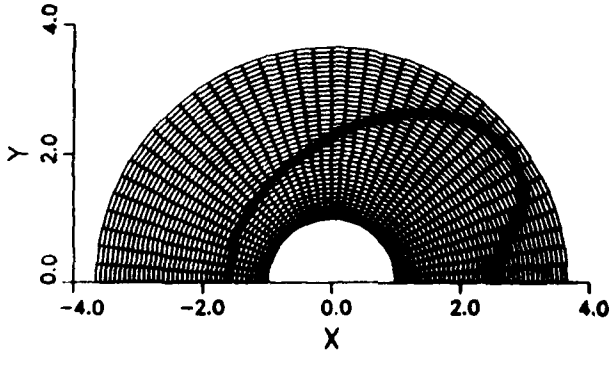


Fig. 18 Comparison of computed pressure solutions



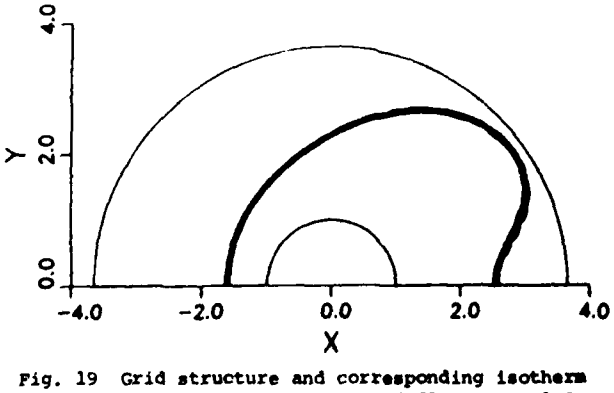
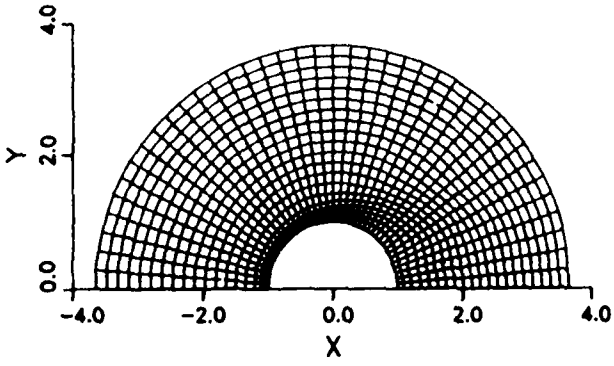


Fig. 19 Grid structure and corresponding isotherm distribution in separated flow, R_e = 100



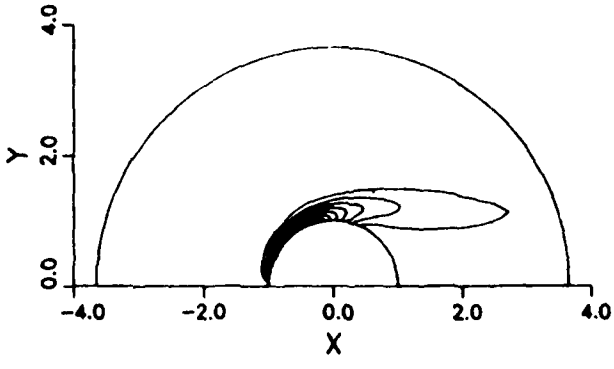


Fig. 20 Grid structure and corresponding vorticity distribution for separated flow, R_e = 100

AIAA'84

AIAA-84-1610

Two Approaches Toward Generating Orthogonal Adaptive Grids

D. A. Anderson and N. Rajendran, Iowa State University, Ames, Iowa

AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference

June 25-27, 1984/Snowmass, Colorado

TWO APPROACHES TOWARD GENERATING ORTHOGONAL ADAPTIVE GRIDS

Dale A. Anderson* and N. Rajendran**
Iowa State University
Ames, Iowa 50011

ABSTRACT

Methods for constructing adaptive grids in more than one dimension have been developed. These methods are usually based upon the idea of equidistribution of a weight function over a grid. Unfortunately, for large grid adaption rates, severe skewing occurs in the mesh. Two techniques for generating an orthogonal adaptive grid are developed and results of applying both schemes to some simple functional examples are presented for the two-dimensional case. Extension to three dimensions is discussed and advantages and disadvantages of the methods are identified.

INTRODUCTION

Grid generation has always been a problem of major concern in the numerical solution of partial differential equations. During the past ten years, satisfactory methods for generating body-fitted mesh systems have evolved and have been used with great success on a variety of problems. More recently, a great amount of interest has centered on the development of dynamically adaptive mesh systems which evolve with the solution of the PDE. Adaptive grid schemes are attractive and are desirable for a number of reasons. These reasons have been discussed in detail by a number of authors.[1,2]

Adaptive schemes in one dimension have been developed and applied by many including Gnoffo,[3] Dwyer et al.,[4] and Rai and Anderson.[5] Basically, these one-dimensional schemes all rely upon equidistribution of a weight function over a mesh, i.e.,

$$wx_{\xi} = C \qquad (1)$$

where w is some positive weighting function, x_{ξ} is the metric of the transformation from physical to computational space, and C is a constant. Using this law, points can be distributed to satisfy any requirement built into the weight function. This expression [Eq. (1)] was solved for either x or ξ by Dwyer and Gnoffo using a direct integration. Rai and Anderson used an iterative residual approach to obtain the same result.

The extension of the equidistribution idea to more than one dimension is desirable since most gains from dynamically adaptive grids will undoubtedly be in multidimensional applications. A logical place to start the extension to two dimensions is to construct a two-coordinate, independent scheme using the direct integral of the equidistribution law given in Eq. (1). Anderson[2] has reported such an extension and has applied this to simple functions to study the adaption process.

The main difficulty with this approach is that high grid skewness occurs even for moderate grid adaption. In other multidimensional studies, Rai and Anderson applied their scheme to a number of examples. The grid adaption employed in these examples was not sufficiently large to induce skewness problems.

The problem of controlling grid distortion in constructing adaptive grids for multidimensional applications must be addressed. Methods formulated without a direct means of grid skewness control are not viable in applications where dense point clustering is desired. The grid distortion problem is avoided if grid orthogonality is enforced when a mesh is generated. In this paper, two schemes for constructing an adaptive, orthogonal mesh are presented. While these methods are still in the exploratory/development stage, the preliminary results are promising.

PROBLEM REVIEW

In order to understand the difficulty of constructing a multidimensional adaptive grid, it is necessary to review the equidistribution concept and show some typical results. The most easily understood multidimensional scheme employs independent grid point adaption along the constant computational coordinate surfaces in physical space.

Let (ξ,η) represent the computational coordinates and (x,y) be coordinates in the physical domain. If S represents arc length along a constant η surface in physical space, a simple equidistribution law controlling point motion along this surface may be written.

$$S_{\xi} w_1 = c_1(\eta) \qquad (2)$$

where w_1 is a positive weight function which depends upon the solution of the PDE system under consideration and c_1 is a function of the computational coordinate, η. This equation can be solved to obtain the integral for arc length

---

*Professor, Department of Aerospace Engineering. Director, Computational Fluid Dynamics Institute. Member AIAA.

**Research assistant, Department of Aerospace Engineering. Member AIAA.

$$S/S_{max} = \frac{\int_0^{\xi} \frac{1}{w_1}\, d\xi}{\int_0^{\xi_{max}} \frac{1}{w_1}\, d\xi} \qquad (3)$$

If N represents arc length along the ξ equal constant surface, the companion equation with weight function w_2 is

$$N/N_{max} = \frac{\int_0^{\eta} \frac{1}{w_2}\, d\eta}{\int_0^{\eta_{max}} \frac{1}{w_2}\, d\eta} \qquad (4)$$

The physical coordinates (x,y) can be recovered from Eqs. (3) and (4). These coordinates can also be computed directly from differential equations. Since

$$S_\xi = [x_\xi^2 + y_\xi^2]^{1/2} \qquad (5)$$

and

$$N_\eta = [x_\eta^2 + y_\eta^2]^{1/2} \qquad (6)$$

governing differential equations for (x,y) can be obtained from the equidistribution laws along ξ and η equal constant surfaces. For example, differentiating Eq. (2) and employing Eq. (5) yields the expression

$$x_\xi x_{\xi\xi} + y_\xi y_{\xi\xi} + (x_\xi^2 + y_\xi^2)\frac{\partial}{\partial\xi}(\ell n w_1) = 0 \qquad (7)$$

The companion expression along a constant ξ surface may be written (see Ref. 2).

$$x_\eta x_{\eta\eta} + y_\eta y_{\eta\eta} + (x_\eta^2 + y_\eta^2)\frac{\partial}{\partial\eta}(\ell n w_2) = 0 \qquad (8)$$

Typical results of applying this independent equidistribution concept along constant ξ and η surfaces are shown in Figs. 1 and 2. The weight functions were of the form

$$w_1 = 1 + A\left|\frac{\partial u}{\partial \xi}\right|$$

$$w_2 = 1 + B\left|\frac{\partial u}{\partial \eta}\right| \qquad (9)$$

where A and B are constants which determine the magnitude of the adaption desired. The shock-like discontinuity in Fig. 1 is typical of many functions where rapid changes occur along one primary coordinate. The choice of a sinusoidal shaped surface in Fig. 2 provides functional changes in both directions and leads to adaption along both families of coordinate surfaces. At lower values of the adaption constants, A and B, distortion is relatively low. However, it is clear that significant distortion occurs in both cases shown. One would not expect to obtain good results if a differential equation was solved with a numerical method using these grids. Some means of controlling the grid skewness must be incorporated in the mesh generator.

Method OT1

The grid skewness at each point in a mesh is easily evaluated by computing the angle of intersection between constant ξ and η surfaces. Consider the intersection of ξ and η equal constant lines in physical space (see Fig. 3). If $\hat{i}_\xi$ is the unit vector along the η equal constant curve and $\hat{i}_\eta$ is the unit vector along the ξ equal constant curve, the cross product of these unit vectors may be written

$$|\hat{i}_\xi \times \hat{i}_\eta| = \sin\theta \qquad (10)$$

where

$$\hat{i}_\xi = \frac{ix_\xi + jy_\xi}{\sqrt{x_\xi^2 + y_\xi^2}} \qquad (11a)$$

and

$$\hat{i}_\eta = \frac{ix_\eta + jy_\eta}{\sqrt{x_\eta^2 + y_\eta^2}} \qquad (11b)$$

Performing the indicated operations, Eq. (10) may be written

$$J = S_\xi N_\eta \sin\theta \qquad (12)$$

where θ is the intersection angle between ξ and η equal constant curves and J is the Jacobian of the transformation

$$J = x_\xi y_\eta - x_\eta y_\xi \qquad (13)$$

The intersection angle is easily monitored by computing $\sin\theta$ through Eq. (12). In fact, the angle θ is directly influenced by the choice of weight functions. If the equidistribution laws for S_ξ and N_η are substituted into Eq. (12), the result is

$$c_1 c_2 / w_1 w_2 = J/\sin\theta \qquad (14)$$

Assume that grid adaption along one coordinate (S) is all that is necessary. Since the arc elements along the S direction are calculated independently, the arc is given by Eq. (3). Suppose that grid distortion is controlled by selecting the intersection angle, θ in Eq. (14). However if θ is specified, the constant c_2 and the weight function w_2 cannot be independently chosen. Since c_1/w_1 and θ are given,

$$N_\eta = c_2/w_2 = \frac{J(w_1/c_1)}{\sin\theta} \qquad (15)$$

and

$$N = \int_0^{\eta} \frac{J(w_1/c_1)}{\sin\theta}\, d\eta \qquad (16)$$

The constant c_1 is of the form

$$c_1 = f(\eta) = \frac{S_{max}(\eta)}{\int_0^{\xi_{max}} \frac{1}{w_1}\, d\xi} \qquad (17)$$

It is interesting to note that the length scale provided through c_2 does not appear in Eq. (16). This shows that the value of N computed at a given point is not scaled as in Eq. (4). The absence of a normalizing length scale in Eq. (16) indicates that the system of PDEs governing this scheme is hyperbolic. The solution of such a problem must be computed in the computational domain by specifying initial data at $\eta = 0$ (N = 0), and marching the solution outward to η_{max}. The outer boundary in physical space corresponding to η_{max} must float and is determined as part of the solution.

Figure 4 shows a solution for the same shock-like function employed in Fig. 1 in the physical domain. In this case, the angle θ, has been selected to be 90 degrees so an orthogonal mesh is created. The grid is 21 × 21 and the weight function w_1 is given in Eq. (9). Again the outer boundary is free and the solution determines the final shape. Both the orthogonality and the floating outer boundary are apparent in these results.

Figure 5 shows the orthogonal grid generated using the sinusoidal function of Fig. 2. The resulting grid shows no evidence of skewing although the distorted outer boundary is again apparent. In almost all calculations involving dynamically adapting mesh systems, the grid point speeds, locations, or the forcing functions are smoothed. In computing adaptive grids with OT1, it was observed that adding smoothing relaxed the orthogonality condition. Thus, no smoothing was employed in computing the results for method OT1.

The results for the orthogonal calculations shown are very good. In applications where adaption is one dimension is desirable and a free outer boundary is not a problem, this is a viable approach for generating an adaptive grid. Can this scheme be extended to three dimensions?

In three dimensions, the applicable equidistribuion laws would be of the form

$$S_\xi = c_1/w_1 \qquad (18a)$$

$$N_\eta = c_2/w_2 \qquad (18b)$$

$$M_\zeta = c_3/w_3 \qquad (18c)$$

where S, N, and M are arc lengths along the computational coordinate surfaces. The angle between the ξ and η directions can again be controlled by noting that

$$|\hat{i}_\xi \times \hat{i}_\eta| = \sin\theta \qquad (19)$$

or

$$|(ix_\xi + jy_\xi + kz_\xi) \times (ix_\eta + jy_\eta + kz_\eta)|$$

$$= S_\xi N_\eta \sin\theta \qquad (20)$$

The angle between the normal to the plane formed by the unit vectors $\hat{i}_\xi$ and i_η and the unit vector $\hat{i}_\zeta$ can be monitored by forming the box product

$$(\hat{i}_\xi \times \hat{i}_\eta) \cdot \hat{i}_\zeta = \sin\phi \qquad (21)$$

This expression reduces to

$$S_\xi N_\eta M_\zeta = J/\sin\theta \qquad (22)$$

If it is assumed that the adaption is in the ξ direction, the quantity S_ξ is prescribed by the equidistribution law. Equations (20) and (22) provide the additional expressions for the quantities N_η and M_ζ. This formulation is similar to the two-dimensional case. While the linearization and classification of the system for three dimensions has not been done, it is reasonable to expect that the grid would be computed on an open domain. The governing equations are probably hyperbolic due to the similarity to the two-dimensional case. An orthogonal grid can be generated with adaption in one direction if ϕ and θ are both taken to be 90 degrees.

Method OT2

The method presented above is most attractive for generating adaptive grids for those problems where grid adaption is necessary in only one direction. However, it seems more appropriate to employ an equidistribution law based on cell area or volume rather than arc length when problems in more than one dimension are considered. In a recent paper, Anderson[6] has introduced such an idea where the equidistribution law is

$$Jw = A_p(t)/A_c \qquad (23)$$

In this expression, w is a positive measure of the solution and is the weight function, J is the Jacobian, $A_p(t)$ is the physical domain integral

$$A_p(t) = \iint w\,dx\,dy \qquad (24)$$

and A_c is the area of the computational domain. This equidistribution law is incorporated in an area continuity equation

$$\frac{\partial x_\tau}{\partial x} + \frac{\partial y_\tau}{\partial y} = \frac{\partial}{\partial \tau}\left[\ell n\left(\frac{A_p(t)}{w}\right)\right] \qquad (25)$$

where (x_τ, y_τ) are the grid speeds. If an orthogonal grid is desired, the time derivative of the orthogonal condition

$$\partial/\partial\tau\ [x_\xi x_\eta + y_\xi y_\eta] = 0 \qquad (26)$$

completes the set for the unknowns (x_τ, y_τ). If the initial grid is orthogonal, the final grid will be orthogonal. After the grid speeds have been determined, the grid point locations are obtained by integration with respect for τ.

The system of Eqs. (25) and (26) is hyperbolic and the grid speeds are determined by marching the solution outward away from the initial data surface at $\eta = 0$. Boundary conditions can be enforced at $\xi = 0$ and $\xi = \xi_{max}$. The outer boundary is free to float as determined by the integration of the grid speeds. Notice that this system is weakly elliptic through the source term of Eq. (25). For the results shown in Figs. 6 and 7, the outer boundary was held at a fixed position and the last grid line computed in the hyperbolic marching scheme was at $\eta_{max} - \Delta\eta$. The source term was computed on a domain with fixed boundaries. The results show that this grid is also adaptive and orthogonal. In this case the weight function was selected to be of the form

$$w = 1 + |\nabla_{\xi,\eta} u|A \qquad (27)$$

The results in Figs. 6 and 7 for the shock and sine function are computed with grid adaption based upon an area equidistribution. While some similarities exist between the results obtained using OT1 and OT2, the grids produced do show some differences. One of the problems noted when employing OT2 is that grid adaption is always accomplished at the expense of available cell area at a greater value of η. Since the grid equations are hyperbolic, the area equidistribution law always necessitates the borrowing of area at large η. Consequently, the adjustment of the mesh is slow since the source term of Eq. (25) is the only means for providing an upstream influence.

For two-dimensional problems, the grid produced by either method OT1 or OT2 are satisfactory for the cases considered. The extension of OT2 to three dimensions can be accomplished in a straightforward manner. The orthogonality condition, Eq. (26), must be altered to include the term $z_\xi z_\eta$. In addition, another condition is necessary to provide the proper cell orientation. This relationship is supplied by

$$\partial/\partial\tau\,(J - S_\xi N_\eta M_\zeta) = 0 \qquad (28)$$

This last expression provides another PDE for the grid speeds in three dimensions.

It is interesting to note that one can derive the relationship between the area weight function, w, and the one-dimensional functions w_1 and w_2. For the orthogonal grids considered, the two-dimensional result is

$$S_\xi N_\eta = J = c_1 c_2 / w_1 w_2 = A_p(t)/wA_c \qquad (29)$$

Since $c_1 w_1$ is selected and c_2/w_2 is determined by orthogonality, the corresponding w for the area equidistribution case can be directly computed from Eq. (29).

SUMMARY AND CONCLUSIONS

Two schemes for producing adaptive, orthogonal grids have been presented. The first is based upon one-dimensional equidistribution and provides adaption along only one coordinate. The other coordinate location is determined by the orthogonality constraint. The second method employs the concept of equidistribution over an area or volume to generate a single PDE for the grid speeds or point locations. Additional expressions are obtained from the orthogonality conditions.

Both schemes produce systems of hyperbolic partial differential equations. This is expected since even in the general, nonadaptive case, orthogonal grids cannot be obtained on a closed domain when Dirichlet boundary conditions are used in solving the governing PDEs.

Method OT1 can be implemented by solving for arc lengths along constant ξ and η surfaces and then computing the corresponding values of x and y. However, an alternative approach is to compute x and y directly from the governing PDEs.

Method OT2 was formulated using grid speeds. However, a steady formulation may also be used. With this approach, the governing linearized PDEs must be solved by marching outward away from an initial data surface. In this case, the x and y coordinates are obtained instead of the grid speeds. The steady formulation of method OT2 is exactly the adaptive counterpart of the grid generation scheme presented by Steger and Sorenson.[7]

Both schemes provide reasonable results for the simple test problems illustrated here. Additional work is needed in evaluating the applicability of these ideas to actual flow problems. Studies coupling the orthogonal generators with the flow equations will commence in the near future.

ACKNOWLEDGEMENTS

This work was supported with funds made available under U.S. Army contract DAAK11-82-R-0123 and under Air Force Office of Scientific Research grant AFOSR-83-0167.

REFERENCES

1. Dwyer, H.A., "Grid Adaption for Problems with Separation, Cell Reynolds Number, Shock-Boundary Layer Interaction, and Accuracy," AIAA Paper 83-0449, presented at AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, 1983.

2. Anderson, D.A. "Adaptive Grid Methods for Partial Differential Equations," In Advances in Grid Generation, ed. K. Ghia and U. Ghia, presented at Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, Texas. New York: ASME, 1983, pp. 1-15.

3. Gnoffo, P.A., "A Vectorized, Finite-Volume, Adaptive-Grid Algorithm for Navier-Stokes Calculations," In Numerical Grid Generation, Proceedings of a Symposium on the Numerical Generation of Curvilinear Coordinate Systems and their Use in the Numerical Solution of Partial Differential Equations,, ed. J.F. Thompson. New York: Elsevier/North Holland, 1982, pp. 819-835.

4. Dwyer, H., Kee, R., and Sanders, B., "Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," AIAA Journal, Vol. 18, 1980, pp. 1205-1212.

5. Rai, M.M. and Anderson, D.A., "Grid Evolution in Time Asymptotic Problems," Journal of Computational Physics, Vol. 43, 1981, pp. 327-344.

6. Anderson, D.A., "Application of Adaptive Grids to Transient Problems," In Adaptive Computational Methods for Partial Differential Equations, ed. I. Babuska, J. Chandra, and J. Flaherty. Philadelphia: SIAM, 1983, pp. 208-223.

7. Steger, J.L. and Sorenson, R.L., "Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates," Numerical Grid Generation Techniques, NASA CP 2166, 1980, pp. 463-478.

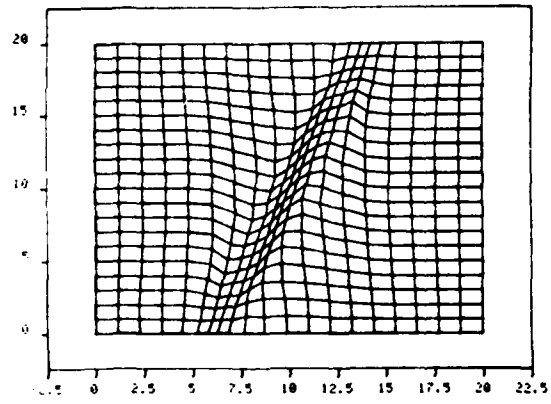


Figure 1. Grid adaption with two degrees of freedom, A = 4, B = 4

$$u(x,y) = 1.0 \qquad 0 \le x \le 0.4(y + 12.5)$$

$$u(x,y) = 0.5(7. - x) + 0.2y, \quad 0.4(y + 12.5) \le x \le 0.4(y + 17.5)$$

$$u(x,y) = 0.0 \qquad 0.4(y + 17.5) \le x \le 20$$

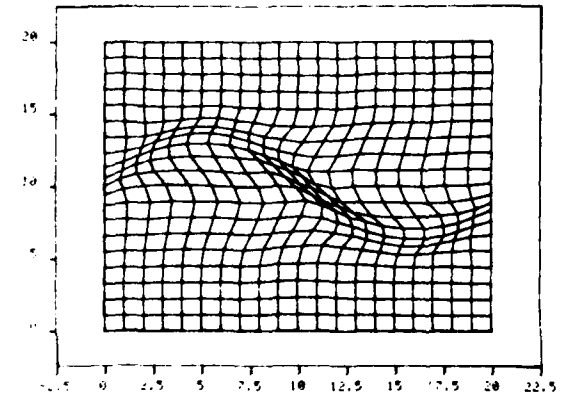


Figure 2. Grid adaption for sinusoidal function distribution, A = 2, B = 3

$$u(x,y) = 0 \qquad 0 \le y \le 9. + 4\sin\left(\frac{2\pi x}{21}\right)$$

$$u(x,y) = 0.5\left|y - 9. - 4\sin\left(\frac{2\pi x}{21}\right)\right|, \quad 9. + 4\sin\left(\frac{2\pi x}{21}\right) \le y \le 11. + 4\sin\left(\frac{2\pi x}{21}\right)$$

$$u(x,y) = 1.0 \qquad 11. + 4\sin\frac{2\pi x}{21} \le y \le 20.$$

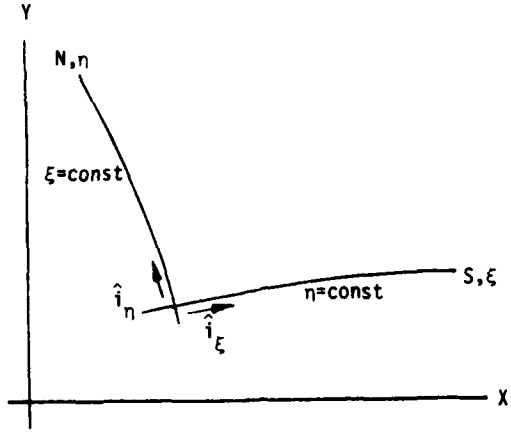


Figure 3. Grid line intersection

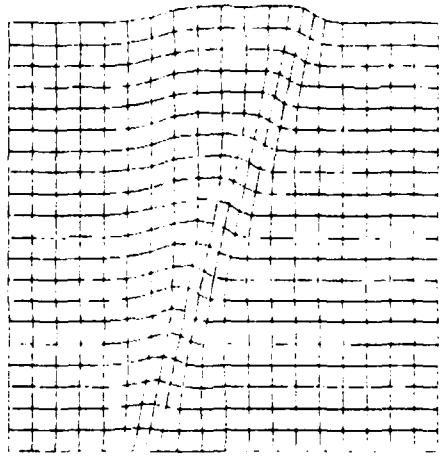


Figure 4. Orthogonal, one-dimensional adaptive grid for shock-like function (see Fig. 1), A = 3.0

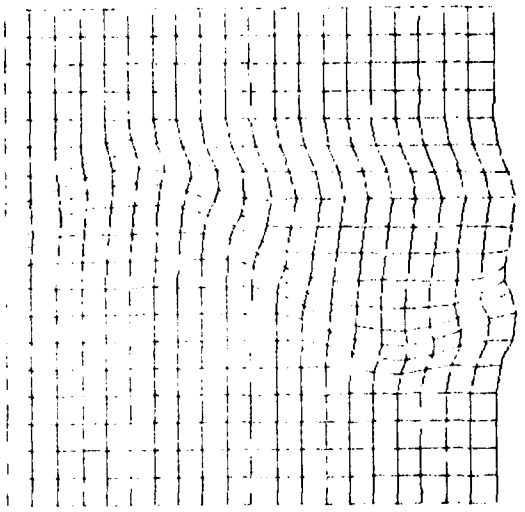


Figure 5. Orthogonal, one-dimensional adaptive grid for sinusoidal function (see Fig. 2), A = 3.0

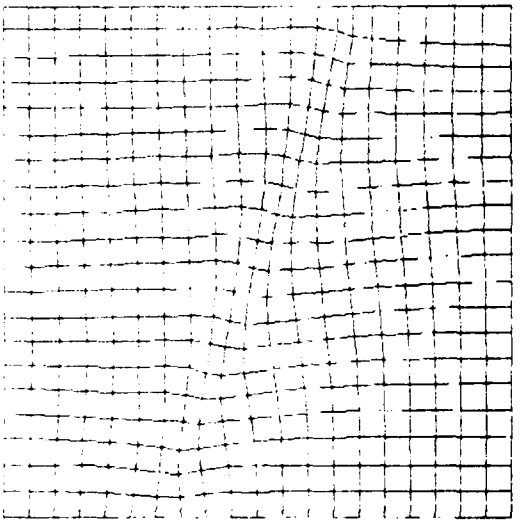


Figure 6. Orthogonal, two-dimensional adaptive grid for shock-like function (see Fig. 1), A = 1.0

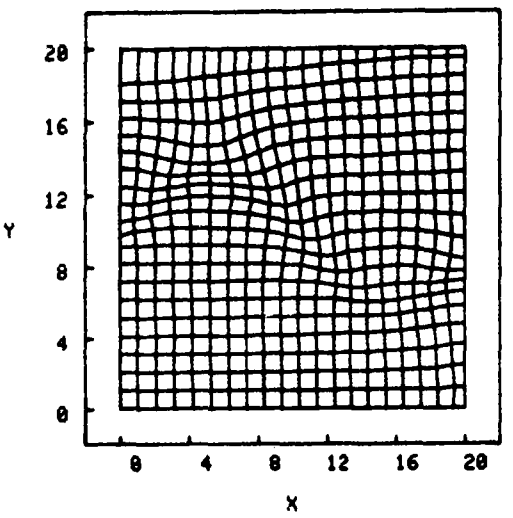


Figure 7. Orthogonal, two-dimensional adaptive grid for sinusoidal function (see Fig. 2), A = 1.0

AIAA'84

AIAA-84-1668

Development of a Dynamically Adaptive Grid Method for Multidimensional Problems

J. E. Holcomb and R. G. Hindman, Iowa State University, Ames, Iowa

AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference

June 25-27, 1984/Snowmass, Colorado

DEVELOPMENT OF A DYNAMICALLY ADAPTIVE GRID METHOD FOR MULTIDIMENSIONAL PROBLEMS

J. Eric Holcomb* and Richard G. Hindman**
Iowa State University
Ames, Iowa 50011

ABSTRACT

An approach to solution adaptive grid generation for use with finite difference techniques, previously demonstrated on model problems in one space dimension, has been extended to multidimensional problems. The method is based on the popular elliptic steady grid generators, but is "dynamically" adaptive in the sense that a grid is maintained at all times satisfying the steady grid law driven by a solution-dependent source term.

Testing has been carried out on Burgers' equation in one and two space dimensions. Results appear encouraging both for inviscid wave propagation cases and viscous boundary layer cases, suggesting that application to practical flow problems is now possible. In the course of the work, obstacles relating to grid correction, smoothing of the solution, and elliptic equation solvers have been largely overcome. Concern remains, however, about grid skewness, boundary layer resolution and the need for implicit integration methods. Also, the method in 3-D is expected to be very demanding of computer resources.

NOMENCLATURE

a	= clustering constant
c,d	= wave speeds
e	= numerical error
f,g	= flux vectors in governing equation
F	= function describing surface (F=0)
G	= grid speeds $\vec{r}_\tau = (x_\tau, y_\tau)$
J	= Jacobian of transformation
k	= the quantity $J(P\vec{r}_\xi + Q\vec{r}_\eta)$
P,Q	= grid clustering (forcing) functions
$\vec{r}$	= position vector = (x,y)
R	= residual of steady grid equation
s	= distance along surface
t	= time
u	= model dependent flow variable
w	= positive measure of solution gradient
(x,y)	= physical coordinates
α,β,γ	= coefficients arising from transformation of Laplacian
(ξ,η)	= computational coordinates
λ	= damping constant
μ	= viscosity coefficient
τ	= computational time
ω	= smoothing constant
∇	= backward difference or gradient operator
Δ	= forward difference or increment (as in $\Delta\xi$, $\Delta\tau$)
∇^2	= Laplacian operator in x,y domain
$\nabla^2_{\xi\eta}$	= Laplacian operator in ξ,η domain

Matrices

[A]	= coefficient matrix in grid speed equation
[B]	= coefficient matrix in grid speed equation
[Pu]	= Jacobian matrix of derivatives of P with respect to u
[Qu]	= Jacobian matrix of derivatives of Q with respect to u
[S]	= Smoothing operator expressed in matrix form

Subscripts

i,j	= row and column indices
k,l	= summation indices
x,y,z	= partial differentiation
ξ,η,τ	= partial differentiation

Superscripts

~	= smoothed quantities
-	= simplified forms

INTRODUCTION

Methods for generating fixed finite difference grids around two-dimensional airfoils and other geometries have evolved to the point where such grids are routinely employed. Often these grids are generated by an elliptic partial differential equation relating the physical and the computational domains[1] (see Chapter 10). The need for developing a class of solution-adaptive methods may arise from: (1) boundary motion in unsteady flow problems; (2) moving shock problems; (3) time-marching to a steady state, where regions requiring high resolution are not known in advance; (4) space-marching problems (e.g. parabolized Navier-Stokes), where a grid must be generated in each transverse plane moving downstream.

It is natural to suppose that the steady (static) grid generators might be extended to the dynamic case. This can be accomplished by differentiating the elliptic p.d.e. with respect to time to yield the so-called grid speed equation describing point motion. Grid speeds are then integrated in time simultaneously with the solution to the governing equation at the new locations. Such an approach was tested by Hindman, Kutler, and Anderson[2] on an Euler-equation solver in two dimensions for the case of arbitrary boundaries but no interior grid clustering, and more recently by Hindman and Spencer[3] on Burgers' equation in one dimension with a source term for clustering.

---

*Research assistant, Department of Aerospace Engineering. Student member AIAA.

**Assistant professor, Computational Fluid Dynamics Institute and Department of Aerospace Engineering. Member AIAA.

The present work is a refinement of Hindman's 1-D method, followed by an extension to multidimensional problems. It should be stressed that the main focus of this study has been on method development rather than application, hence only simple geometries have been considered. As will be discussed later, application to practical 1-D and 2-D problems now appears possible.

Readers who are interested in a more complete introduction to adaptive grid methods, or who wish to study alternate approaches, should consult the recent survey papers by Anderson.[4-7]

METHOD

In order to efficiently summarize the method, a table of equations is provided (Table 1). Please refer to this table while reviewing the following comments. Also refer to the block diagram of the "system" formed by the grid and the model flow variable (Fig. 1).

Domain

The generalized transformation mapping the physical space into the computational space is as given by Eq. set (1). The purpose of the transformation is to allow a uniform rectangular grid to be used for computation, while the physical grid conforms to boundary shapes and is clustered where high resolution is needed. As a result of the mapping, derivatives must be transformed according to Eq. set (2).

Steady Grid Law

A specific form of the above mentioned transformation is obtained by solving a Poisson equation (Eq. 3a), an idea developed by Winslow and Thompson[8-10] for the time-invariant case. To solve for the (x,y) point locations, the role of dependent and independent variables must be interchanged, yielding in 2-D a coupled set of two nonlinear elliptic partial differential equations (Eq. 3b). The forcing functions P(x,y,u) and Q(x,y,u) are carefully chosen to provide adequate clustering without grid crossing or overlap.

Clustering Function

A rationale for choosing P(x,u) in 1-D is provided by the integral grid law cited (Eq. 4), where the function w(u) is a positive measure of solution gradient. Differentiation twice with respect to ξ reveals that this form is equivalent to the P-function selected (Eq. 5). It is clear that x will always be a monotonically increasing function of ξ, hence grid crossing cannot occur (except due to numerical effects). The clustering constant controls the amount of adaption, from none ($a = 0$) up to a large amount ($a \rightarrow \infty$). The possibility of using a different clustering function will be discussed later. In 2-D, P and Q are obtained by a logical extension of the 1-D form.

Simplified forms of the grid equations (Eqs. 6, 7b) resulting from the cancellation of (x_ξ) factors were used in 1-D. As a result, the equations are linearized and may be solved directly. In 2-D, P and Q can still be split into a factor involving only u and one involving only x and y.

Grid Speed Equation

This equation (Eq. 7a,b) describing the rates of point motion (x_τ, y_τ) is derived by differentiating the steady grid law with respect to time (τ). Grid speeds would be zero (with no boundary motion) except for the fact that changes in the solution cause the clustering functions to vary. Thus it is very important that accurate representations for P_τ and Q_τ be obtained. In the present work, this is done by taking analytical derivatives (Table 3). Usually P and Q depend only on x,y, and u at the center point of the finite difference molecule and its immediate neighbors, which simplifies the calculation of P_τ and Q_τ. Even so, substantial amounts of computation and storage are required, suggesting that it may be better to obtain these quantities by backward differencing in time as part of an iterative process for solving the grid speed equation.

Governing Equation

The p.d.e. governing the physical process is cast into conservation law form (Eq. 8a), then transformed to computational space, including terms due to grid point motion (Eq. 8b). For Burgers' equation (Eqs. 8c,d) the flux vectors f and g take the scalar form $f = g = u^2/2$. Linearized expressions $f = cu$ and $g = du$ were used for simplicity in 2-D. In the case of the Navier-Stokes or Euler equations, u, f, and g form a set of vector quantities, but only one element of u (say the density) need be selected to drive the grid. All flux derivative terms for the inviscid problem were evaluated by upwind differencing, analogous to the flux splitting class of methods applied to the Euler equations.

A special problem is presented by the viscous Burgers' equation in that upwind differencing on the convective terms generates excessive dissipation and hence large errors in the steady-state solution. This may be overcome by using a centrally-differenced scheme, however grid size is then restriced to satisfy a mesh Reynolds number constraint. The third-order upwind correction proposed by Leonard[11] was found to be successful in one dimension. (Also see results section.)

ALGORITHM

The computer program written for this study executes the steps that follow with the explanation tailored for the 2-D case. The physical domain in 2-D is the unit square.

1. Establish input data: (a) initial distribution of u as a function of (x,y) or (ξ,η); (b) clustering constant, smoothing constant, and grid damping constant (see below for explanation of smoothing and damping); (c) time step limits and wave speeds if applicable; (d) iteration limits, tolerances, and over-relaxation factors; (e) number of steps to be computed and/or steady-state convergence criterion; (f) initialization of (x,y) to uniform grid, and x_τ and y_τ to zero.

2. Solve steady grid generator for initial grid. A Gauss-Seidel point iterative process is employed, updating x,y,P,Q and related quantities during each sweep through the grid. Over-relaxation is generally possible and speeds convergence considerably. Boundaries of the 2-D domain are treated by applying the 1-D steady grid law. The forcing functions $\bar{P}$ and $\bar{Q}$ may be calculated in advance if u is initially specified as a function of (ξ,η), otherwise u(x,y) must be corrected by interpolation or other means, and $\bar{P}$ and $\bar{Q}$ recalculated as (x,y) change at a point.

Begin loop

3. Calculate the transformation metrics (x_ξ,x_η,y_ξ,y_η), the coefficients α, β, γ, and the Jacobian J. Also calculate the coefficient matrices A and B in the grid speed equation (Table 2) and the residuals R^x and R^y of the steady grid generator.

4. Solve the grid speed equation: (a) calculate the six nonzero P_u and Q_u quantities at each point; begin loop (b) calculate u_τ, which depends upon (x_τ,y_τ), the flux vectors, and the viscous terms if present; (c) update the boundary grid speeds by a line relaxation scheme acting on the 1-D grid speed equation; (d) update the interior grid speeds by an alternating direction line relaxation scheme, whereby sweeps in the ξ-direction update the x-equation and sweeps in the η-direction update the y-equation; (e) exit loop if converged.

Notes: Under-relaxation must be used to achieve convergence, especially for large clustering constants. It is important to include the dependence of u_τ on (x_τ,y_τ) implicitly in the relaxation scheme wherever possible.

5. Obtain new flow solution and grid locations: (a) re-calculate u_τ for converged grid speeds; (b) establish allowable time step size (limited by an inviscid CFL-type condition and by a viscous term condition); (c) perform first-order explicit integration to get new values for x,y, and u; (d) update the functions P and Q at the new time level.

Exit loop if requested number of time steps have been completed, or if solution has reached a steady state.

SPECIAL CONCERNS

Although the algorithm is quite straightforward, a few special points need to be addressed:

Smoothing

There are at least three reasons why smoothing of the solution, u, might be necessary: (a) to suppress oscillations ("wiggles") associated with certain methods of integrating the governing equation; (b) to avoid grid crossing that may occur due to numerical effects when the forcing functions become large; (c) to allow the derivatives P_τ and Q_τ to be computed analytically, even near a discontinuity in u. For the present algorithm only the last two reasons apply, since integration schemes are selected to avoid oscillations in u. Therefore, smoothing is needed only for the purpose of grid calculations, an important distinction because the undesirable effects of artificial smoothing-induced diffusion and dispersion are avoided when integrating the governing equation.

To understand the nature of problem (b), consider the simplified grid equation, $x_{\xi\xi} + x_\xi \bar{P}(u) = 0$. It is easily seen that discretization by central differencing on $x_{\xi\xi}$ and x_ξ will cause grid crossing if abs(P) > 2, even though the exact mathematical solution does not exhibit this property. However, by applying a smoothing operator, one can always prevent the forcing function from becoming too large and also insure that sufficient smoothness exist to compute P_τ and Q_τ.

Table 4 presents 1-D and 2-D versions of the smoothing operator. Since both $[P_u]$ and [S] are tridiagonal in 1-D, the grid speed equation (containing P_τ) becomes pentadiagonal, requiring 2.5 times as much computation to solve as the standard Thomas algorithm for tridiagonal systems. In 2-D, the equations are solved iteratively and one need only smooth u_τ before updating the grid speeds on each step. For the viscous Burgers' equation, no smoothing is required if the starting solution is smooth. In practice, such an initial condition may be obtained by solving $\nabla^2_{\xi\eta} u = 0$, provided that one is not interested in transient behaviors associated with other possible initial conditions.

Grid Correction

The grid speed equation may be written in the form $dR/d\tau = 0$, where R is the residual of the steady grid generator. Such a form is neutrally stable, that is errors in integration will neither be amplified nor damped out. An obvious fix is to add a damping term, $dR/d\tau + \lambda R = 0$, where the damping constant λ may be chosen as large as $1/\Delta\tau$ for stability with an explicit integration scheme. The resulting grid correction method is both effective and much simpler than previous methods (involving a new solution to the steady grid equation while holding u fixed in either physical or computational space).

Boundary Conditions

Dirichlet boundary data can be specified by fixing (x,y) at points along a ξ=const or η=const boundary. Unfortunately, this does not allow for

grid adaptation on the boundary. A better idea is to use the 1-D grid method to treat the boundaries. Easy application of the 1-D method is possible for straight boundaries (as considered here); however, for curved boundaries, a more general form must be used, $s_{\xi\xi} + s_{\xi}\bar{P}(u) = 0$, where s is the distance along the surface. In the grid speed equation the general boundary condition is $G \cdot \nabla F = -\partial F/\partial t$, where G is the vector (x_τ, y_τ) of grid speeds, and $F(x,y,t) = 0$ describes a moving surface in 2-D. A correction step will probably be necessary for curved boundaries since the condition stated is first-order.

Differencing

As pointed out above, exclusive use of central differencing in solving the grid equations may result in numerical difficulties. A novel idea to replace the usual "arithmetic mean difference," $x_\xi = (\nabla x + \Delta x)/2\Delta\xi$ with a "geometric mean difference," $x_\xi = \sqrt{\nabla x \Delta x}/\Delta\xi$ was tested and found to yield excellent results in 1-D (including the 1-D wave propagation results presented). However, the idea was discarded for general use because extension to higher dimensions is difficult. One problem is that ∇x and Δx may be of opposite sign. Also, the geometric difference operator is nonlinear, and the computation time required to take square roots becomes significant in 2-D.

RESULTS AND DISCUSSION

One-Dimensional Inviscid

Adaptive grid solutions to $u_t + uu_x = 0$ were computed for the initial condition of a 1-0 discontinuity on an 11 point grid covering the interval [0,1]. The left boundary was fixed at u = 1, while the right boundary required no special treatment due to the use of upwind differencing. Results are presented in the form of a grid time history plot (Figs. 2a,b). It can be seen that the grid tracked the wave quite well and maintained a reasonable degree of clustering. Wave speed errors (compared to the exact speed of 1/2) were calculated and found not to exceed 2%. Problems with dissipation in u were not experienced, partly since the nonlinear nature of Burgers' equation causes profiles having a negative slope to steepen.

One-Dimensional Viscous

Steady-state solutions to $u_t + uu_x = \mu u_{xx}$ were computed on a 21 point grid for the boundary conditions u(0) = 1 and u(1) = 0. The initial distribution of u was a ramp function given by u(x) = 1-x. Results for viscosity coefficients of $\mu = 0.05$ and $\mu = 0.10$ are presented in Tables 5 a-f and compared with the exact small viscosity solution, $u(x) = \tanh((1-x)/2\mu)$. Although the adaptive grid method does provide better boundary layer resolution and accuracy than a fixed uniform grid method, there is still room for improvement. Convergence required between 100 and 500 time steps, pointing out the severe stability restrictions for explicit methods applied to viscous problems.

As a final point, a brief experiment with the method of Leonard to difference the u_x term was conducted, yielding excellent results (Table 5g,h) which seem to verify the claimed third-order spatial accuracy of the method. Otherwise, central differences on the convective terms were used in both the 1-D and the 2-D viscous cases.

Two-Dimensional Inviscid

Solutions to $u_t + cu_x + du_y = 0$ were computed with wave speeds c = d = 1, an initial condition of a 1 - 1/2 - 0 discontinuity along the main diagonal, and a 16x16 grid on the unit square. Inflow boundaries were treated by simulating an infinite domain with a diagonal wave, while the outflow boundaries again required no special treatment. Grid plots, including lines of constant u, are presented for successive time steps (Figs. 3a-d) for a test run with a clustering constant of a = 10. Good tracking of the wave was obtained, and wave speed errors were small. Clustering at the discontinuity produced skewed grid cells that may or may not be desirable in more general problems. Dissipation in the solution u did occur to a significant extent as time progressed, causing the clustering of the grid to diminish. Numerically, some problems were experienced when the wave intersected the corners, forcing the use of a small first time step.

Two-Dimensional Viscous

Steady-state solutions to $u_t + u_x + u_y = \mu(u_{xx}+u_{yy})$ were computed on a 16x16 grid with a viscosity coefficient of $\mu = 0.10$. The boundary conditions were $u(x,0) = u_b(x)/u_b(0)$, $u(0,y) = u_b(y)/u_b(0)$, $u(x,1) = u(1,y) = 0$, where the function u_b is defined by $u_b(s) = 1 - \exp[(s-1)/\mu]$. The initial distribution of u was obtained as indicated in the preceding "Smoothing" section. An exact time-invariant solution, $u(x,y) = u_b(x)u_b(y)/[u_b(0)]^2$, is presented by Anderson and Rai.[12] The computed results show maximum deviations from this solution of 0.019 (uniform square grid), 0.016 (a = 10), and 0.014 (a = 20). Resolution of the boundary layer was judged to be adequate but not ideal. Convergence required up to 400 time steps.

Clustering

A major concern in adaptive grid work is that of obtaining adequate resolution of the physical domain with as few total grid points as possible. The clustering function used in this study is derived from an equidistribution law based on the solution gradient u_ξ, and should provide reason-

able control over grid point locations. Nonetheless, it has already been demonstrated that discretization effects and smoothing play a crucial role in determining the grid. Also, results from the boundary layer cases show that achieving sufficient resolution of viscous flow profiles is not easy. For these viscous problems, it is likely that the solution gradient in physical space, $u_x = u_\xi / x_\xi$, should be used to drive the grid rather than u_ξ. Such a modification increases the complexity of the functions P, Q and their derivatives P_τ and Q_τ. In any event, it seems that experimenting with alternate grid driving functions and grid generation concepts is desirable.

Computational Considerations

In one dimension the present method is efficient in terms of both computation time and storage, even for large grids. Also, direct solution of the simplified grid and grid speed equations is possible in 1-D.

On the other hand, stepping up to two dimensions greatly increases time and storage requirements. Direct solutions are no longer practical, so iterative methods must be employed. The number of iterative cycles required to converge the grid speed equation may range from only one, if the flow solution is not changing much from step to step, up to 30 or 40 (on a 16x16 grid) starting from $x_\tau = y_\tau = 0$ at all points. Over 40 2-D arrays need to be stored in order to avoid recalculating various quantities. Since explicit integration in 2-D is severely limited by small time steps, it appears certain that implicit schemes will have to be developed for adaptive grid work, as they already have been for fixed-grid PNS calculations.

There appears to be little theoretical difficulty in extending the present adaptive grid formulation to a third dimension, but this might only be practical now for moderate grid size problems run on a machine in the supercomputer class. Although adaptive grids will always be more complicated to use than fixed grids, the extra work is often justified by improved solution accuracy for a given number of grid points (hence perhaps an overall savings to achieve the same accuracy level).

As previously mentioned, it appears that the present method has advanced to the point where application to solving a set of vector equations is now possible. For inviscid problems, an explicit predictor/corrector method[7] or a split flux differencing method might be successfully employed to integrate the governing equations. In the viscous case, a class of non-iterative, implicit, approximate factorization methods already exist for solving the PNS equations in generalized coordinates[1] (see Chapter 8). However, interaction between the grid and flow dynamics will force a new look at these implicit schemes for adaptive grid calculations.

FUTURE WORK

Future work should center around the following subjects already discussed: (a) implicit integration methods; (b) curved or time-varying boundaries; (c) application to the Euler, Parabolized Navier-Stokes, or full Navier-Stokes equations; (d) further investigation of clustering functions.

ACKNOWLEDGEMENT

The authors gratefully acknowledge support of this work by the National Aeronautics and Space Administration under training grant NGT 016-002-801, by the Air Force Office of Scientific Research under grant AFOSR-83-0167, and by the Aerospace Engineering Department and Engineering Research Institute at Iowa State University.

REFERENCES

1. Anderson, D.A., Tannehill, J.C., and Pletcher, R.H. Computational Fluid Mechanics and Heat Transfer. New York: McGraw-Hill, 1984.

2. Hindman, R.G., Kutler, P., and Anderson, D. "Two-Dimensional Unsteady Euler-Equation Solver for Arbitrarily Shaped Flow Regions." AIAA Journal, 19(1981), pp. 424-431.

3. Hindman, R.G. and Spencer, J. "A New Approach to Truly Adaptive Grid Generation." AIAA Paper 83-0450, Jan. 1983.

4. Anderson, D.A. "Solution-Adaptive Grids for Partial Differential Equations." In ARO Report 82-3: Proceedings of the 1982 Army Numerical Analysis and Computers Conference, Vicksburg, Mississippi, 1982, pp. 575-591.

5. Anderson, D.A. "Application of Adaptive Grids to Transient Problems." In Adaptive Computational Methods for Partial Differential Equations, ed. I. Babuska, J. Chandra, and J. Flaherty. Philadelphia: SIAM, 1983, pp. 208-223.

6. Anderson, D.A. "Adaptive Grid Methods for Partial Differential Equations." In Advances in Grid Generation, ed. K. Ghia and U. Ghia, Presented at Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, Texas. New York: ASME, 1983, pp. 1-15.

7. Anderson, D.A. "Adaptive Mesh Schemes Based on Grid Speeds." AIAA Paper 83-1931, July 1983.

8. Winslow, A. "Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh." J. of Computational Physics, 1(1966), pp. 149-172.

9. Thompson, J.F., Thames, F.C., and Mastin, C.W. "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate Systems for Fields Containing Any Arbitrary Number of Two-Dimensional Bodies." J. Computational Physics, 15(1974), pp. 299-319.

The code employed herein has been extensively applied to different flow problems, including two-dimensional shock wave boundary layer interaction over a flat plate[9], transonic turbulent afterbody flow[9], turbulent and inviscid transonic flow over airfoils[13], and others[14,15]. Turbulence models in the examples above are algebraic, since completely satisfactory multi-equation models for flows with large separated regions are not yet available[13]. The variety of flow problems solved with this code prove its reliability, making it a viable scheme to use with this adaptive grid routine.

INITIAL GRID GENERATION

Since an equidistribution adaptive grid routine is not a grid generation routine as well, an initial starting grid must be created by some other means before a solution is run. With adaption available in only one coordinate in the proposed scheme, the optimal starting mesh for this routine would be one which has a sufficient point distribution in the remaining computational direction.

Thomas and Middlecoff[16] have introduced a method of grid generation based on earlier techniques which allows for a priori grid point clustering in at least one computational direction. The technique follows the well-known method of Thompson, Thames and Mastin[17], whereby an elliptic system of two equations of the form

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= P(\xi,\eta) \\ \eta_{xx} + \eta_{yy} &= Q(\xi,\eta) \end{aligned} \qquad (19)$$

is solved. By interchanging the roles of the dependent (ξ,η) and independent (x,y) coordinates in this equation, a quasilinear elliptic system of equations is obtained, which is then solved by finite differences. Thomas and Middlecoff have obtained analytical expressions for the weighting functions P and Q which will cluster points on the interior of the grid to the same degree as the specified point distribution on the grid boundaries.

The test grid described earlier (Figure 2) as well as all subsequent grids were generated by a solver based on this technique. In Figure 2, relatively high values for s_ξ were specified on the grid's inner boundary (η=1), at the leading edge (ξ=51,) and at both sides of the trailing edge (ξ=21 and ξ=81). The influence from the inner boundary is seen in the point distributions for η=constant curves further from the body. More important, though, is the point distribution in the η direction, which does not change with adaption. The high grid clustering in the η direction near the body seen in Figure 2 is necessary to resolve the large solution gradients existing near the body. With a sufficient clustering in the η-direction, the adaptive grid solver is then used to adaptively cluster points in the ξ-direction.

NUMERICAL RESULTS

NACA0012 Airfoil Results

The majority of the numerical results of this study were obtained for transonic flow past an NACA0012 airfoil. The initial grid employed was again the C-type 2-D test grid of Figure 2, and on this grid, the Euler equations were solved using Tassa's[15] Navier-Stokes code. The Euler equations were selected rather than the full Navier-Stokes equations to reduce the starting grid dimensional requirements, thus permitting the adaption routine to be tested on a less complicated mesh. With the flow conditions selected, M_∞ = 0.75 and α = 2 degrees, a shock wave forms near the upper-surface midchord of the airfoil.

A steady-state solution was run using a variable time step integration procedure on the fixed grid. The numerical solution was found to converge after only a few hundred time steps, resulting in the density contour field of Figure 3a, which gives no indication of a shock wave, due to the sparsity of grid points (Δx=0.6) in the anticipated shock region.

A new solution was then formed, starting the flow impulsively from free stream conditions on the initial grid, and then passing control to the adaptive grid routine after every 20 time integration steps, again using a variable time integration procedure. For this case, minimum grid spacing along each η=constant curve was set at 0.003 chords (Δs_{min}=0.003), the initial weighting constant A (equation (10)) was set at 10, and both functions f_2 and f_3 were turned off, allowing for adaption based purely on density gradients. Figures 8a and 9a depict the converged grid and density field for this test case. On the newly adapted grid, the numerical density field reveals a shock wave. The point density in the adapted grid has increased near the shock region, but still is not exceptionally high. As a result, the shock remains somewhat smeared, albeit over a smaller region than with the initial grid.

To investigate the convergence of this adaptive grid algorithm, it was interesting to follow, among other variables, the current value of the constant A, the actual minimum grid spacing in the adaptive coordinate direction, and the maximum distance any point moved between adaption sweeps, Δx_{max}. The adaptive grid was considered converged when the first two of these three parameters approached a constant value, and the last parameter went to zero. The convergence history of the adaptive grid test case above is shown in Figure 7. These particular values of the adaptive variables (A, B, Δs_{min}, etc.) are seen to provide an adaptive grid which converges quickly and smoothly. Note that the converged minimum grid spacing $\min_{\xi,\eta}(s(\xi+1)-s(\xi))$ is equal to 0.003, precisely the value of Δs_{min} chosen at the onset of the problem. Note also that Δx_{max} reaches a peak value after three adaptive sweeps (60 integration steps), and not immediately, as one might expect. This is due to the grid relaxation technique defined by equation (18), which prohibits large grid point movement in the early numerical development of the solution.

In order to increase the shock region resolution even further, it was logical to reduce the Δs_{min} selected in the adaption routine, thus increasing the final A weighting constant and likewise increasing point density in the areas of high fluid density gradients. Figures 8b and 9b show the converged adapted grid and density contour distribution corresponding to a Δs_{min} equal to

daptive Algorithm

ne numerical algorithm used to obtain the adaptive grid point distribution governed by equation 9b) is reasonably straightforward, essentially onsisting of two rather large programming loops, ne within another. The outer loop controls which = constant surface is being adapted, increasing rom $\eta=1$ to $\eta=\eta_{max}$. The inner loop is iterative, nd has a convergence criterion which must be satsfied for the current η = constant curve before he η index is incremented on the outer loop. The nner convergence loop works as follows: Equation 9b) is integrated numerically to obtain a new oint distribution $\xi(s)$. With this vector and ith $\rho(s)$, $x(s)$ and $y(s)$, an interpolation scheme s applied to calculate the newly adapted vectors namely $s_{new}(\xi)$, $\rho_{new}(\xi)$, $x_{new}(\xi)$ and $y_{new}(\xi)$. he new weighting function $F(s)$ (equation (10)) is alculated from the new vectors, and equation (9b) s once again integrated. This process is continued until the L_2 norm of the Δs vector, defined as

$$L_2 = \sum_{\xi=\xi_1}^{\xi_2} \left(s_{new}(\xi) - s_{old}(\xi)\right)^2 \qquad (16)$$

falls below a specified tolerance. When the convergence criterion is met, control is passed to the outer loop, and the convergence loop is applied to the next η = constant curve.

At this point, several comments about the algorithm are in order.

1. Due to inherent truncation errors, finite difference approximations of the streamwise density gradients needed in function f_1 are not sufficiently smooth. To eliminate this problem, it is necessary to apply several sweeps of explicit second order smoothing of the form

$$\rho_{ij}^{k+1} = [1 + ALP\ \nabla^2]\rho_{ij}^{k} \qquad (17)$$

to the local density field before the grid is adapted, where $0 \le ALP \le 0.25$ from stability considerations.

2. By adjusting the value of the constant A in equation (10) after each call to the adaption routine, it is possible to specify a time-asymptotic minimum distance between adjacent grid points on a given s-curve, called Δs_{min}. This constraint is needed to keep points from clustering too closely across a shock, which causes numerical problems, as explained later.

3. In many cases, particularly in the early development of a flowfield from an impulsive start, numerical difficulties may arise if the grid points are moved too drastically in one adaptive sweep. It is useful to under-relax the calculated point distribution $s(\xi)$ according to

$$s(\xi) = s_{old}(\xi) + RELMIN\left\{s_{new}(\xi) - s_{old}(\xi)\right\} \qquad (18)$$

where $s(\xi)$ is the final arc-length function, $s_{old}(\xi)$ is the current s function, and $s_{new}(\xi)$ is the final s function obtained from the inner convergence loop for a given η = constant line. By choosing a small value for RELMIN initially, and by increasing it gradually to 1 after several adaptive sweeps, changes in the grid point between adaptions are sufficiently small to prevent solution instabilities.

4. It is important also to mention that in the present formulation of the adaptive algorithm, temporal metric terms (grid speeds) in the transformed equations governing the flow are set equal to zero. As a result, once the grid is updated through adaption, the corresponding solution vector (e.g, (ρ,u,v,e)) must also be updated. This is done through interpolation from the current grid and solution vector. Since the grid speeds are neglected, it is difficult to determine if a time accurate solution can be obtained with this adaptive scheme. In light of this, only steady-state solutions are examined in this paper.

This algorithm is designed to supplement an existing aerodynamic solver, ideally linked to the main program as a single subroutine. Alterations needed to implement the adaptive routine affected only two per cent of the total programming lines in the Navier-Stokes/Euler code described next, and it is not anticipated to be much higher for most other codes.

AERODYNAMIC SOLVER

All numerical flowfield results in this study were obtained from a finite difference code developed by Tassa[9], which solves the unsteady 2-dimensional Reynolds averaged Navier-Stokes equations written in conservation form on a general non-orthogonal curvilinear coordinate system[10]. Flow variables and physical directions are non-dimensionalized so that the four governing P.D.E.s and the equation of state $P = \rho RT$, included for closure of the system, become normalized. This allows the characteristic parameters of the flow, such as Reynolds number, to be varied independently.

The resulting parabolic system of equations is solved numerically through a modified form of the Briley-McDonald Alternating Direction Implicit scheme[11]. Whereas the Briley and McDonald dual time level scheme represents all but the energy equation in conservation form, Tassa's modified three time level scheme writes even the energy equation in conservation form. Non-linear terms in these equations are linearized by using Taylor series expansions at the known time level. By representing the dependent variables ρ, u, v and e as the sum of values at the known time level and an incremental value, a linear matrix equation is obtained in terms of the unknown incremental values. The Douglas-Gunn[12] procedure for generating ADI schemes is then applied to the new system of equations, splitting the matrix equation into a system of two one-dimensional matrix operators. After discretizing the spatial operators using second-order formulas, the incremental solution vector is found by block elimination techniques. Tassa and Schuster[13] have found it necessary to add artificial dissipation near regions of severe pressure gradients such as shocks, to suppress high frequency components. In addition, fourth-order dissipation is added to the dependent variables in the Euler equations in order to reduce the overshoot of pressure across the shock.

$$f_1(s) = \left|\frac{\partial \rho}{\partial s}\right| = \left|\frac{\partial \rho}{\partial \xi}\,\xi_s\right| = \left|\frac{\partial \rho}{\partial \xi}\frac{1}{\sqrt{x_\xi^2 + y_\xi^2}}\right| \quad (11)$$

The density gradient is selected with the case of transonic flow in mind. Across a shock, density changes rapidly in a physical sense, and for the Euler equations, discontinuously in a mathematical sense. If the streamwise direction of flow and the s-curves (η = constant loci) are nearly aligned in space, then $\partial\rho/\partial s$ will be relatively high in the shock region, forcing the grid points to move towards the shock location. For the test solution previously mentioned, this is indeed the case, as evidenced by the adaptive grid of Figure 3b. Note that along each η = constant line, the areas of higher grid point clustering are at the airfoil leading edge and in the shock region. This is as expected, since it is in exactly these regions where a large streamwise density gradient is observed (Figure 3a).

In order to preserve the shape of the s-curves while using adaption, it is important to retain at least a minimum number of grid points in the regions of high arc curvature, particularly along the boundaries of the physical domain, where altering the boundary will likewise alter the problem under consideration. For this reason, $f_2(s)$ is defined as

$$f_2(s) = B\left|\frac{x_\xi y_{\xi\xi} - y_\xi x_{\xi\xi}}{(x_\xi^2 + y_\xi^2)^{3/2}}\right| = B|K| \quad (12)$$

where K is the mathematical definition of the curvature of s (parameterized by ξ), and B is again a positive constant, used to control the degree of clustering based on grid curvature. Observe that if the spatial density gradients are negligible and if both functions $f_1(s)$ and $f_2(s)$ are set equal to zero, then the right hand side of equation (10) approaches unity. With this weighting function, grid points are spaced at equal increments in the s-direction along each η = constant surface, as shown in Figure 4. Compared with the test grid of Figure 2, the grid in Figure 4 has poor resolution at the leading edge, and hence, the true airfoil shape is not sufficiently defined. Furthermore, since the shape of each η = constant curve is algebraically defined by curves fit through the grid points, few points in regions of high arc curvature might eventually alter the shape of the airfoil after several adaptive sweeps. Fortunately, by clustering points in these high curvature regions, this problem can be avoided. Figure 5 depicts the same test grid with points redistributed according to function f_2 for a nominal value of the constant B. On this grid, the shape of the η = constant curves near the leading edge are preserved.

Another noticeable difference between Figures 2 and 4 is in the grid spacing near the boundaries of the adaptive domain. The elliptic properties of the original grid generator insure that grid spacing changes smoothly throughout the mesh. However, since the adaptive grid algorithm redistributes points within the adaptive domain with no regard to the points outside the domain, there is no guarantee that grid spacing is continuous across the boundary. Figure 4 illustrates this fact. Along the ξ_1 and ξ_2 lines, where grid spacing in the ξ direction varies rapidly, the accuracy of the finite difference equations may be inadequate. For example, at any point along the arc, a finite difference equation with second-order truncation error on a uniformly spaced grid will decay to first order accuracy on a non-uniform grid whenever the grid spacing on both sides of the point differ greatly- that is, when $\left|\frac{\Delta s^+}{\Delta s^-}\right| << 1$ or $\left|\frac{\Delta s^+}{\Delta s^-}\right| >> 1$.

To remedy this spacing problem, a third weighting function is introduced, defined as

$$f_3(s) = D\left[\xi_s^*(s=s_{max})\,e^{-g\left(\frac{s}{s_{max}}\right)} + \xi_s^*(s=0)\,e^{-g\left(\frac{s_{max}-s}{s_{max}}\right)}\right] \quad (13)$$

$$D \geq 0,\ g=50$$

where the asterisk refers to the value of ξ_s evaluated outside of the adaptive boundary. This function is chosen since, for a large enough value of g (=50),

$$f_3(s_{max}) \approx D[\xi_s^*(s_{max})],\ f_3(0) \approx D[\xi_s^*(0)]$$

and

$$f_3(s) \approx 0 \text{ for } 0 << s << s_{max} \quad (14a,b)$$

Differentiating equation (9b) whith respect to s yields

$$F(s) = \xi_s \frac{\int_0^{s_{max}} F(s)\,ds}{(\xi_2 - \xi_1)} = \xi_s(C_1) \quad (15)$$

Provided that both grid curvature and density gradients are negligible at the adaptive domain boundaries, it can be shown that by choosing the constant $D = C_1/A$, grid spacing (ξ_s) will be continuous across the boundary of $D_{adaptive}$. As a result, the order of the solution truncation error near the boundary should not decrease (still $o(\Delta x^2)$) appreciably.

Function $f_3(s)$ is used to create the adaptive grids in Figures 6a and 6b. The first of these grids has a weighting function equal to $f_3(s)$ alone, producing a grid with equal spacing along each s-curve everywhere except near the boundary of $D_{adaptive}$. Both functions f_2 and f_3 are used to create Figure 6b, which has clustering near the leading edge as well as equal grid spacing through the boundary. Indeed, it is a combination of all three weighting functions that will produce the most desirable adaptive grid.

Consider first a positive weighting function w=w(s), associated with some partial differential equation, chosen so that w increases as the grid point density (ξ_s) needed to approximate the solution to the partial differential equation to some fixed error also increases. Saltzman[5] has shown, through a variational approach, that by minimizing the integral

$$I(s) = \int_0^{s_{max}} W(s)\ s_\xi\ d\xi \qquad (5)$$

the error due to solution approximation is also minimized. The Euler-Lagrange equation corresponding to this integral is now

$$\left(\frac{\partial}{\partial \xi} - \frac{\partial}{\partial s}\frac{\partial}{\partial \xi_s}\right)(W\ s_\xi) = 0 \qquad (6)$$

which reduces to the condition that

$$W(\xi)^{\frac{1}{2}}\ s_\xi = \text{constant}, \qquad (7a)$$

or, with s as the independent variable,

$$W(s)^{\frac{1}{2}}\ /\ \xi_s = \text{constant} \qquad (7b)$$

since s is assumed to be a function of ξ only. Replacing $\sqrt{w(\xi)}$ with $F(\xi)$ for the sake of convenience, and assuming that the $\Delta\xi$ between adjacent grid points along each s-curve is equal to one, equation (7a) can be approximated by forward-differences as $F(\xi)\Delta s$=constant, where $\Delta s = s(\xi+1)-s(\xi)$. This states that the product of the weighting function and the grid spacing is equally distributed along the s-curve. Anderson et al.[6] have for this reason named adaptive grid methods governed by equation (7) equidistribution schemes. Now, the boundary conditions for (7a) and (7b) along each s-curve are

$$s(\xi_1) = 0 \qquad (8a)$$
$$s(\xi_2) = s_{max}$$

and

$$\xi(0) = \xi_1 \qquad (8b)$$
$$\xi(s_{max}) = \xi_2$$

The corresponding solutions to equations (7a) and (7b) are then found to be

$$\left(\frac{s}{s_{max}}\right) = \frac{\int_{\xi_1}^{\xi} 1/F(\xi)\ d\xi}{\int_{\xi_1}^{\xi_2} 1/F(\xi)\ d\xi} \qquad (9a)$$

and

$$\left(\frac{\xi-\xi_1}{\xi_2-\xi_1}\right) = \frac{\int_0^s F(s)\ ds}{\int_0^{s_{max}} F(s)\ ds} \qquad (9b)$$

respectively.

For a general weighting function F(s), neither of equations (9) can be numerically integrated directly. The integrands on the right hand side of each equation are functions of the grid point distributions $s(\xi)$ and $\xi(s)$, which appear also on the left hand side of each equation. To solve the equations, then, an indirect method, such as an iterative updating procedure, must be employed. Although equation (9a) may appear to be a better choice than (9b) for numerical integration, on the basis of iterative convergence speeds this was not the case. Rather, the number of iterations needed for convergence of (9b) was as much as an order of magnitude less than the number needed for convergence of (9a), and because of this, equation (9b) was used exclusively in this work. This difference in convergence speeds was due in part to the form of the weighting function F(s) used, described below.

Weighting Function

The weighting function chosen for this study is of the form

$$F(s) = 1 + A(f_1(s) + f_2(s) + f_3(s)) \qquad (10)$$

where A is a positive constant and $f_1(s)$, $f_2(s)$, and $f_3(s)$ are each non-negative functions. Including the constant, 1, in F(s) allows A to control the degree of grid clustering, and insures that F(s) will not approach zero ($\xi_s \to 0$), which is not feasible.

To illustrate the utility of each of these three functions f_1, f_2 and f_3, a test grid was generated, the inner detail of which is presented in Figure 2. The inner boundary of this C-type grid (η = 1 surface) is an NACA0012 airfoil, with 61 points wrapped counter-clockwise around the surface, and 42 more points, ranging from $1 \le \xi \le 21$ and $81 \le \xi \le 101$, distributed downstream of the trailing edge. The adaptive domain, $D_{adaptive}$, is bounded by two η=constant lines emanating from the trailing edge (ξ_1=21,ξ_2=81). The adaptive coordinate chosen is ξ, meaning that $s(\xi)$ functions will be redistributed along each of the 21 nearly concentric η=constant curves. The dimensions of the adaptive and total domains are then 61 x 21 and 101 x 21, respectively. On this grid, a converged 2-D conservative variable Euler equation solution was generated for a Mach number M_∞=0.75 and angle of attack α =2.0 degrees. These flow conditions are known to produce a shock just upstream of the upper surface midchord on a grid with sufficient point density in the shock region. On this test grid, however, the shock region point density is sparse, and the shock is smeared across several grid points, as shown in Figure 3a, which pictures curves of constant fluid density. The methods used to generate both the initial grid and the initial solution are described later in this paper.

The most important term in the weighting function is $f_1(s)$, defined as the first partial derivative of fluid density with respect to arclength s. Numerically, this derivative is easily calculated by noting that

minimizing the integral, additional P.D.E.s must be solved. Nevertheless, the fact that these techniques are based on a firm mathematical foundation should eventually make them more popular than ad hoc procedures.

One particular type of variational adaptive grid, previously employed by both Dwyer[6] and Gnoffo[7], is particularly attractive if adaption is needed in only one computational coordinate. This technique is referred to as an equidistribution adaptive grid scheme, and is the type of grid scheme extensively studied in this work. The scheme, which is formulated and explained in some detail in the next section, is an ideal technique for use with transonic airfoil problems, where grid point adaption is usually only needed in the streamwise direction of flow.

ADAPTIVE GRID SCHEME

Mathematical Formulation

When finite-difference techniques are used to obtain the solutions to the partial differential equations governing fluid flows, a finite number of points in physical space must first be selected. These points comprise the grid, or mesh, at which the solution to the discretized versions of the P.D.E.s (the finite difference equations) are to be calculated. In two dimensions, the physical location of each grid point can be defined by its two Cartesian coordinates, (x,y). Mesh points can alternatively be defined by two coordinates ξ and η, chosen so that the grid points in the x-y physical plane become equally spaced and fixed in time in the ξ-η computational plane. For simplicity, ξ and η are integer valued, usually set equal to 1. The computational coordinates of each point are then associated with a storage location in a two-dimensional array. The representation in either the computational or physical plane uniquely defines a given mesh point.

Now, the computational plane and the physical plane are mathematically related through the vector-valued mapping

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{Bmatrix} \tag{1a}$$

which is schematically represented in Figure 1. Provided that this transformation is both one-to-one and onto, the inverse mapping also exists, and is defined as

$$\begin{Bmatrix} \xi \\ \eta \end{Bmatrix} = \begin{Bmatrix} \xi(x,y) \\ \eta(x,y) \end{Bmatrix} \tag{1b}$$

In differential notation, this transformation can be written as

$$\begin{Bmatrix} dx \\ dy \end{Bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{Bmatrix} d\xi \\ d\eta \end{Bmatrix} \tag{2a}$$

or inversely as

$$\begin{Bmatrix} d\xi \\ d\eta \end{Bmatrix} = \frac{1}{J} \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix} \begin{Bmatrix} dx \\ dy \end{Bmatrix} \tag{2b}$$

where $J = x_\xi y_\eta - x_\eta y_\xi$, the Jacobian of the mapping. The major advantage of employing computational coordinates is that through equations (2a) and (2b), the P.D.E.s governing the flow can be transformed so the independent variables are now ξ and η. This reduces the complexity of the finite-difference equations, since the computational grid on which they are solved is both equally spaced and non-moving.

The complete grid region in the physical plane will be known as the total domain, D_{total}, and is mathematically defined as

$$D_{total} = \left\{ \begin{matrix} x = x(\xi,\eta) \\ y = y(\xi,\eta) \end{matrix} \;\middle|\; \begin{matrix} 1 \le \xi \le \xi_{max} \\ 1 \le \eta \le \eta_{max} \end{matrix} \right\} \tag{3}$$

The ranges on the computational coordinates are chosen for convenience only. Either past experience or intuition may determine that inadequacies in grid point spacing prevent the discretized P.D.E.s from approximating the governing equations of motion to a desired degree. In general, only a subregion of the total domain will have an inadequate grid point distribution, so only the grid points in this region, called $D_{adaptive}$, and defined as

$$D_{adaptive} = \left\{ \begin{matrix} x = x(\xi,\eta) \\ y = y(\xi,\eta) \end{matrix} \;\middle|\; \begin{matrix} \xi_1 \le \xi \le \xi_2 \\ 1 \le \eta \le \eta_{max} \end{matrix} \right\} \tag{4}$$

will need to be redistributed. $D_{adaptive}$, as the name suggests, is then the domain in which the adaptive transformation will be applied. Schematically, $D_{adaptive}$ is the shaded area of Figure 1. Note that since $D_{adaptive} \subseteq D_{total}$, the cases where either $\xi_1=1$ or $\xi_2=\xi_{max}$ are certainly allowable.

Now let $s(\xi)$ be the arclength along an η=constant surface in $D_{adaptive}$ such that $s(\xi=\xi_1)=0$ and $s(\xi=\xi_2)=s_{max}$, and observe that $s(\xi)<s(\xi+1)$ for all $\{\xi | \xi_1<\xi<\xi_2\}$. Calculation of a new $s(\xi)$ function for each η=constant surface will result in a redistribution of the points along the s-curve. Since the points are free to move only along η=constant surfaces and not along ξ=constant surfaces, the selection of the adaptive coordinate (ξ in this case) is not a trivial matter, but is dependent on the type of flow being modeled. Fortunately, in many cases, particularly those with well-defined regions of large solution gradients, the correct adapting coordinate is immediately apparent. The problem now becomes one of determining the new $s(\xi)$ function for each η=constant curve.

AN ADAPTIVE GRID SCHEME APPLIED TO TWO-DIMENSIONAL AIRFOIL PROBLEMS

John P. Steinbrenner*
Iowa State University, Ames, Ia.

Yehuda Tassa**
Lockheed-Georgia Co., Marietta, Ga.

Dale A. Anderson†
Iowa State University, Ames, Ia.

ABSTRACT

A dynamically adaptive grid scheme based on equidistribution in one computational coordinate is applied for the first time to inviscid transonic flow numerically solved on C-type airfoil grids. Steady-state solutions are obtained for NACA0012 and RAE2822 airfoils using both fixed and solution adaptive grids, and results for both grids are compared with previous numerical and experimental data. The adaptive grid algorithm is seen to resolve details of the flow field near the upper-surface midchord shock not seen in the fixed grid solution, thus eliminating the need for a priori grid point clustering in the region of the anticipated shock. In addition, problems inherent to schemes of this type are discussed, and suggestions for further study are also made.

INTRODUCTION

The generation of computational meshes for use in solving discretized systems of partial differential equations (P.D.E.s) is presently a subject of intense research. In most cases, a grid is generated by either algebraic, complex variable, or differential equation methods before any numerical solutions are calculated, with the resultant mesh being used for all subsequent computations. A major pitfall of this accepted technique lies in the fact that the mesh point distribution often proves to be inadequate for approximating the problem under consideration.

When the P.D.E.s governing certain fluid flows are discretized and solved on an insufficient mesh, it is possible that certain high gradient phenomena of the flow field will not be captured, due to the sparsity of grid points in the high gradient regions. For example, boundary layer, free shear layer, and captured shock regions (flows with multiple length scales) all require locally high grid resolution for the solution to be approximated to a given degree of accuracy. This problem might be alleviated by using a grid with high resolution throughout, but the added computer storage and time demands make this impractical. Alternatively, points could be clustered only in the anticipated regions of large solution gradients. This, however, requires an a priori knowledge of the flow field, which many times is not available.

A better suggestion is to use a dynamically adaptive grid, i.e., one which continually adjusts the point distribution within the mesh as the flow solution is advanced in time. An ideal adaptive mesh scheme would be one which readjusts the grid points so that the local truncation errors of the discretized equations of motion are reduced to a minimum, constant value throughout the mesh. Such a scheme would require analytical or approximate expressions for the local truncation error. Unfortunately, except for the most basic equations of fluid motion, expressions for the truncation error of the associated finite difference equations are extremely difficult or impossible to calculate. Therefore, when truncation errors are not available, grid point adaption should be driven by some other mathematical or physical relation which will still improve the overall accuracy of the finite difference equations of motion. For most adaptive grid schemes, this is indeed the case, although the means used to reach this end differ greatly.

Anderson[1] has separated existing grid techniques into two distinct categories. In the first of these two categories, a mathematical law defining the speed of the grid points is postulated, and the new grid point locations are obtained by integrating in time. Although calculating grid point locations from the grid speeds is straightforward, formulating grid speed laws based on sound physical reasoning is not. In fact, a fair amount of ingenuity is often necessary. For example, in a technique developed by Rai and Anderson[2], grid speeds are calculated from an attraction model. In this model every two grid points induce on each other a small velocity which is dependent both on the distance between the two points and the other point's deviation from the average solution error. The grid speed at each point is then set equal to the sum of all of the velocities induced by every other point in the grid. A survey of adaptive grid schemes based on grid speed laws is given in reference 3.

In the second of these two categories, grid points are moved to new locations through specified mathematical mappings, and then the grid speeds needed in the equations of motion are calculated from backward differences. Among the techniques falling into this class are variational methods, developed extensively by Brackbill[4] and Saltzman[5]. In variational methods, an integral containing a measure of some grid parameter, such as orthogonality or smoothness, is minimized. One disadvantage of variational methods is that by

---

*Research assistant, Department of Aerospace Engineering. Student member AIAA.

**Staff scientist. Associate fellow AIAA.

†Professor, Department of Aerospace Engineering and Computational Fluid Dynamics Institute. Member AIAA.

AIAA-84-1608

An Adaptive Grid Scheme Applied to Two-dimensional Airfoil Problems

J. P. Steinbrenner and D. A. Anderson, Iowa State University, Ames, Ia.; and Y. Tassa, Lockheed-Georgia Co., Marietta, Ga.

AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference

June 25-27, 1984/Snowmass, Colorado

Table 5. One-dimensional viscous Burgers' equation. Numerical results for boundary layer.

(a) $\mu = 0.05$, $a = 0.0$
step = 200, time = 5.00

N	x	u	error
.			
.			
.			
17	0.8000	0.9756	0.0116
18	0.8500	0.9286	0.0234
19	0.9000	0.8000	0.0384
20	0.9500	0.5000	0.0379
21	1.0000	0.0000	0.0000

Integrated error = 0.00601

(b) $\mu = 0.05$, $a = 10.0$
step = 300, time = 3.39

N	x	u	error
17	0.8576	0.9060	0.0156
18	0.9027	0.7686	0.0188
19	0.9396	0.5562	0.0164
20	0.9710	0.2911	0.0094
21	1.0000	0.0000	0.0000

Integrated error = 0.00329

(c) $\mu = 0.05$, $a = 20.0$
step = 400, time = 3.72

N	x	u	error
17	0.8815	0.8443	0.0153
18	0.9186	0.6871	0.0154
19	0.9489	0.4832	0.0122
20	0.9753	0.2490	0.0067
21	1.0000	0.0000	0.0000

Integrated error = 0.00289

(d) $\mu = 0.10$, $a = 0.0$
step = 600, time = 3.75

N	x	u	error
15	0.7000	0.9109	0.0057
16	0.7500	0.8557	0.0074
17	0.8000	0.7706	0.0090
18	0.8500	0.6548	0.0096
19	0.9000	0.5106	0.0085
20	0.9500	0.3500	0.0051
21	1.0000	0.0000	0.0000

Integrated error = 0.00296

(e) $\mu = 0.10$, $a = 10.0$
step = 400, time = 3.43

N	x	u	error
15	0.7388	0.8687	0.0056
16	0.7887	0.7906	0.0064
17	0.8360	0.6816	0.0066
18	0.8803	0.5419	0.0060
19	0.9219	0.3765	0.0045
20	0.9614	0.1929	0.0024
21	1.0000	0.0000	0.0000

Integrated error = 0.00224

(f) $\mu = 0.10$, $a = 20.0$
step = 500 time = 3.53

N	x	u	error
15	0.7624	0.8354	0.0054
16	0.8098	0.7460	0.0058
17	0.8532	0.6310	0.0056
18	0.8931	0.4936	0.0048
19	0.9302	0.3389	0.0035
20	0.9656	0.1724	0.0018
21	1.0000	0.0000	0.0000

Integrated error = 0.00204

(g) Leonard 3rd-order differencing
$\mu = 0.05$, $a = 5.0$
step = 100, time = 3.21

N	x	u	error
8	0.7810	0.9789	0.0037
9	0.8826	0.8394	0.0137
10	0.9501	0.4867	0.0251
11	1.0000	0.0000	0.0000

Integrated error = 0.00312

(h) Leonard 3rd-order differencing
$\mu = 0.05$, $a = 5.0$
step = 200, time = 2.66

N	x	u	error
16	0.7846	0.9728	-0.0006
17	0.8364	0.9269	-0.0001
18	0.8858	0.8177	0.0025
19	0.9292	0.6157	0.0061
20	0.9663	0.3306	0.0054
21	1.0000	0.0000	0.0000

Integrated error = 0.00047

Table 3. Finite difference calculation of P_τ and Q_τ

Note: Examples given for calculation of P_τ. Expressions for Q_τ are similar.

General form

$$P = P(x,y,u)$$

$$P_{\tau_{i,j}} = (\partial P_{i,j}/\partial u_{k,l})u_{\tau_{k,l}} + (\partial P_{i,j}/\partial x_{k,l})x_{\tau_{k,l}} + (\partial P_{i,j}/\partial y_{k,l})y_{\tau_{k,l}}$$

where summation on the indices k,l is implied.

Specific form

$$P(x,y,u) = \bar{P}(u)/\gamma(x,y)$$

$$P_\tau = -\bar{P}\gamma_\tau/\gamma^2 + \bar{P}_\tau/\gamma$$

$$\gamma_\tau = \partial/\partial_\tau(x_\xi^2 + y_\xi^2) = 2(x_\xi x_{\tau_\xi} + y_\xi y_{\tau_\xi})$$

$$\bar{P}_{\tau_{i,j}} = (\partial\bar{P}_{i,j}/\partial u_{k,l})u_{\tau_{k,l}} \quad \text{as above.}$$

For $\bar{P} = (2au_\xi u_{\xi\xi})/(1 + au_\xi^2)$ with central differencing,

$$(\partial\bar{P}_{i,j}/\partial u_{i-1,j}) = p(2\nabla u_{i,j} + \bar{P}_{i,j}u_{\xi_{i,j}})$$

$$(\partial\bar{P}_{i,j}/\partial u_{i,j}) = -4pu_{\xi_{i,j}}$$

$$(\partial\bar{P}_{i,j}/\partial u_{i+1,j}) = p(2\Lambda u_{i,j} - \bar{P}_{i,j}u_{\xi_{i,j}})$$

where $p = a/(1 + au_{\xi_{i,j}}^2)$

Table 4. Smoothing operator

$$\tilde{u} = [S]u$$

$$\tilde{P} = \bar{P}(\tilde{u}) \qquad \tilde{Q} = \bar{Q}(\tilde{u})$$

$$\bar{P} = [\bar{P}_{\tilde{u}}][S]u_\tau \qquad \bar{Q}_\tau = [\bar{Q}_{\tilde{u}}][S]u_\tau$$

[S] is derived from a finite difference representation of

$$\tilde{u} = u + \frac{\omega}{4}u_{\xi\xi} \qquad \text{(1-D)}$$

$$\tilde{u} = u + \frac{\omega}{8}(u_{\xi\xi} + u_{\eta\eta}) \qquad \text{(2-D)}$$

In 1-D, [S] is a two-dimensional array. [S] is always constant.

Table 1. (Cont.)

1-D equations	2-D equations
8. Governing equation	
(a) General conservative form	
$u_t + f_x$ = viscous terms	$u_t + f_x + g_y$ = viscous terms
$u_\tau + \xi_t u_\xi + \xi_x f_\xi$ = viscous terms	$u_\tau + \xi_t u_\xi + \eta_t u_\eta + \xi_x f_\xi + \eta_x f_\eta + \xi_y g_\xi + \eta_y g_\eta$ = viscous terms
(b) Burgers' equation	
$u_t + uu_x = \mu u_{xx}$	$u_t + cu_x + du_y = \mu\nabla^2 u$
$u_\tau + (\xi_t + u\xi_x)u_\xi = \mu\nabla^2 u$	$u_\tau + (\xi_t + c\xi_x + d\xi_y)u_\xi + (\eta_t + c\eta_x + d\eta_y)u_\eta = \mu\nabla^2 u$

See equation set (2) for ∇^2. Note that $\nabla^2\xi = P$ and $\nabla^2\eta = Q$.

Table 2. Coefficient matrices [A] and [B] in 2-D grid speed equation

$[A] = [\vec{a}_1, \vec{a}_2]$ $\quad [B] = [\vec{b}_1, \vec{b}_2]$

Let $\vec{r} = (x,y)^T$ and $\vec{k} = J(P\vec{r}_\xi + Q\vec{r}_\eta)$

$\vec{a}_1 = 2(-x_\eta \vec{r}_{\xi\eta} + x_\xi \vec{r}_{\eta\eta} + y_\eta \vec{k})$	$\vec{b}_1 = 2(x_\eta \vec{r}_{\xi\xi} - x_\xi \vec{r}_{\xi\eta} - y_\xi \vec{k})$
$\vec{a}_2 = 2(-y_\eta \vec{r}_{\xi\eta} + y_\xi \vec{r}_{\eta\eta} - x_\eta \vec{k})$	$\vec{b}_2 = 2(y_\eta \vec{r}_{\xi\xi} - y_\xi \vec{r}_{\xi\eta} + x_\xi \vec{k})$

Table 1. (Cont.)

1-D equations	2-D equations

3. Steady grid equation

(a) $\nabla^2\xi = P(\xi,u)$

(b) $x_{\xi\xi} + x_\xi^3 P(x,u) = 0$

(a) $\nabla^2\xi = P(\xi,\eta,u)$; $\nabla^2\eta = Q(\xi,\eta,u)$

(b) $\alpha\vec{r}_{\xi\xi} - 2\beta\vec{r}_{\xi\eta} + \gamma\vec{r}_{\eta\eta} + J^2(P\vec{r}_\xi + Q\vec{r}_\eta) = 0$

where:

$\vec{r} = (x,y)^T$ $\quad \beta = x_\xi x_\eta + y_\xi y_\eta$

$\alpha = x_\eta^2 + y_\eta^2$ $\quad \gamma = x_\xi^2 + y_\xi^2$

4. Integral grid law

$$\xi = \frac{\int_0^x [1 + aw(s)]ds}{\int_0^1 [1 + aw(s)]ds}\,\xi_{max}$$

5. Clustering function

$$P(x,u) = \frac{aw_\xi}{x_\xi^2(1 + aw)}$$

where:

$w(u) = u_\xi^2$

$$P(x,y,u) = \frac{aw_{1\xi}}{\gamma(1 + aw_1)}$$

$$Q(x,y,u) = \frac{aw_{2\xi}}{\alpha(1 + aw_2)}$$

where:

$w_1(u) = u_\xi^2$ $\quad w_2(u) = u_\eta^2$

α,γ defined in (3).

6. Simplified grid equation/clustering function

$P(x,u) = \bar{P}(u)/x_\xi^2$

$x_{\xi\xi} + x_\xi\bar{P}(u) = 0$

$P(x,y,u) = \bar{P}(u)/\gamma(x,y)$

$Q(x,y,u) = \bar{Q}(u)/\alpha(x,y)$

Substitute into (3b).

7. Grid speed equation

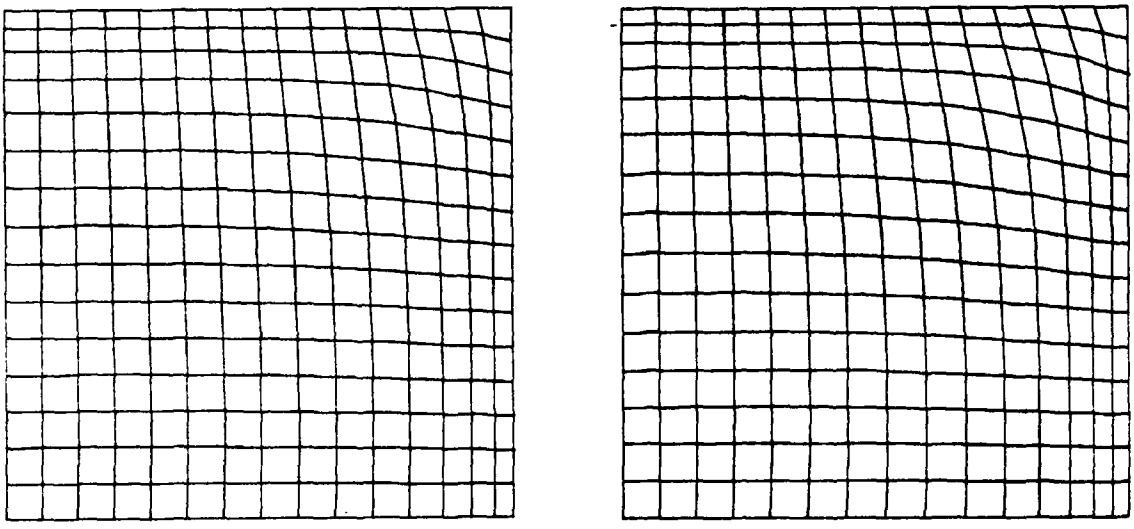
(a) $(x_\tau)_{\xi\xi} + 3x_\xi^2 P(x_\tau)_\xi + x_\xi^3 P_\tau = 0$

(b) $(x_\tau)_{\xi\xi} + \bar{P}(x_\tau)_\xi + x_\xi\bar{P}_\tau = 0$

(a) $\alpha(\vec{r}_\tau)_{\xi\xi} - 2\beta(\vec{r}_\tau)_{\xi\eta} + \gamma(\vec{r}_\tau)_{\eta\eta} + [A](\vec{r}_\tau)_\xi + [B](\vec{r}_\tau)_\eta + J^2(P_\tau\vec{r}_\xi + Q_\tau\vec{r}_\eta) = 0$

See [A] and [B] in Table 2.

See P_τ, Q_τ in Table 3.



(a) (b)

Fig. 4 Two-dimensional viscous Burgers' equation converged grid plot.
(a) a = 10.0, steps = 300, time = 1.90
(b) a = 20.0, steps = 400, time = 1.98

Table 1. Summary of equations

1-D equations	2-D equations
1. Coordinate transformation	
$\xi = \xi(x,t)$	$\xi = \xi(x,y,t)$
$\tau = t$	$\eta = \eta(x,y,t)$
	$\tau = t$
2. Transformation of derivatives	
$\begin{bmatrix} \partial/\partial t \\ \partial/\partial x \end{bmatrix} = \begin{bmatrix} 1 & \xi_t \\ 0 & \xi_x \end{bmatrix} \begin{bmatrix} \partial/\partial \tau \\ \partial/\partial \xi \end{bmatrix}$	$\begin{bmatrix} \partial/\partial t \\ \partial/\partial x \\ \partial/\partial y \end{bmatrix} = \begin{bmatrix} 1 & \xi_t & \eta_t \\ 0 & \xi_x & \eta_x \\ 0 & \xi_y & \eta_y \end{bmatrix} \begin{bmatrix} \partial/\partial \tau \\ \partial/\partial \xi \\ \partial/\partial \eta \end{bmatrix}$
$\nabla^2 = \xi_x^2 \frac{\partial^2}{\partial \xi^2} + (\nabla^2 \xi) \frac{\partial}{\partial \xi}$	$\nabla^2 = (\xi_x^2 + \xi_y^2)\frac{\partial^2}{\partial \xi^2} + 2(\xi_x \eta_x + \xi_y \eta_y)\frac{\partial^2}{\partial \xi \partial \eta}$
	$+ (\eta_x^2 + \eta_y^2)\frac{\partial^2}{\partial \eta^2} + (\nabla^2 \xi)\frac{\partial}{\partial \xi} + (\nabla^2 \eta)\frac{\partial}{\partial \eta}$
where:	
$\xi_t = -x_\tau \xi_x$	$\xi_t = -x_\tau \xi_x - y_\tau \xi_y$; $\eta_t = -x_\tau \eta_x - y_\tau \eta_y$
$\xi_x = 1/x_\xi$	$\xi_x = \frac{1}{J} y_\eta$; $\xi_y = -\frac{1}{J} x_\eta$
	$\eta_x = -\frac{1}{J} y_\xi$; $\eta_y = \frac{1}{J} x_\xi$
	$J = x_\xi y_\eta - x_\eta y_\xi$

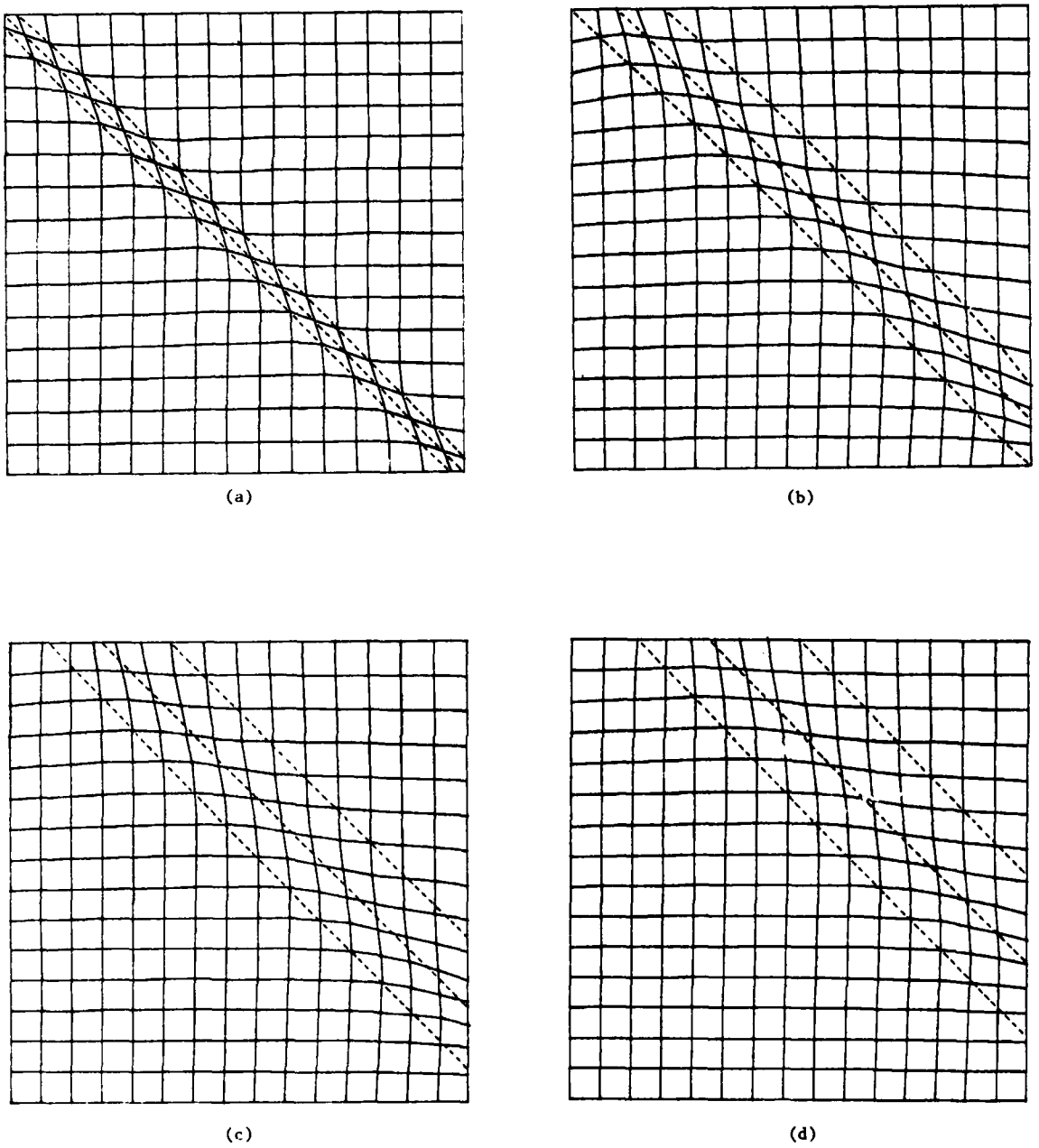


Fig. 3 Two-dimensional inviscid Burgers' equation grid plot ($a = 10.0$, $\omega = 1.0$, $\lambda = 20.0$).
(a) steps = 0, time = 0.00
(b) steps = 5, time = 0.05
(c) steps = 10, time = 0.10
(d) steps = 15, time = 0.15
Note: Dashed lines define boundary of wave and center of wave.

10. Mastin, C.W. and Thompson, J.F. "Adaptive Grids Generated by Elliptic Systems." AIAA Paper 83-0451, Jan. 1983.

11. Leonard, B.P. "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation." Comput. Methods Appl. Mech. Eng., 19(1979), pp. 59-98.

12. Anderson, D.A. and Rai, M.M. "The Use of Solution Adaptive Grids in Solving Partial Differential Equations." In Numerical Grid Generation, ed. J. F. Thompson. New York: Elsevier/North Holland, 1982, pp. 317-338.

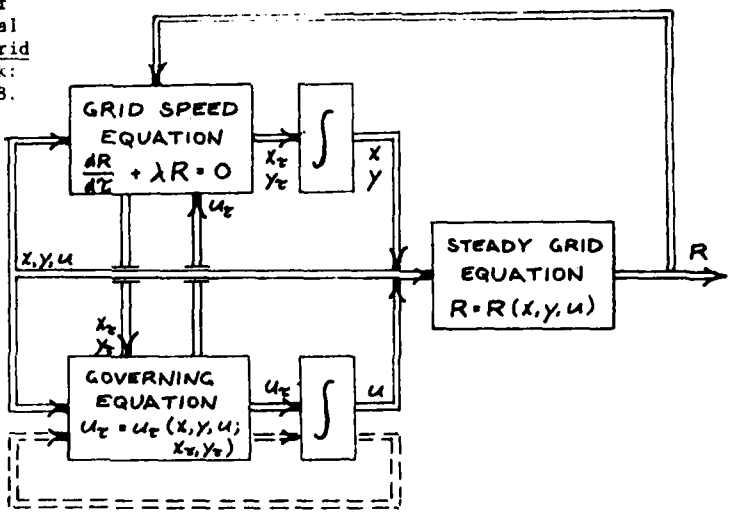
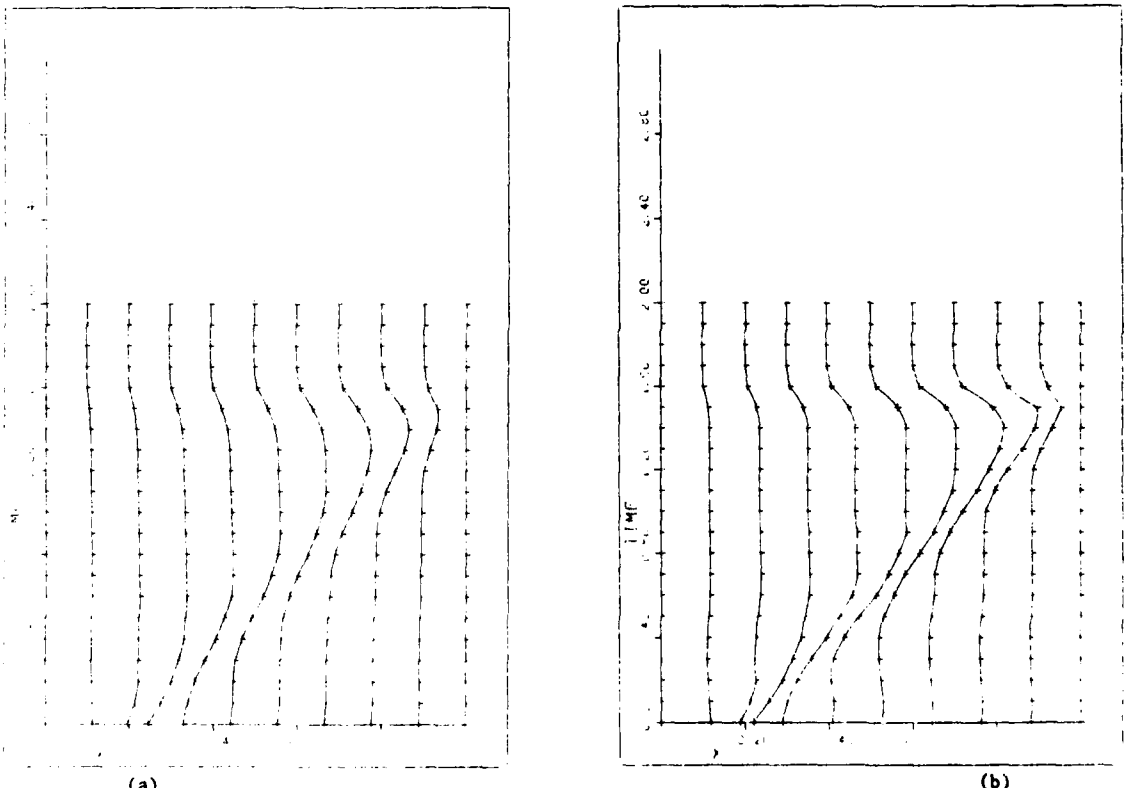


Fig. 1 Block diagram of "system" formed by grid and fluid dynamics.



(a) (b)

Fig. 2 One-dimensional inviscid Burgers' equation grid time history plot.
(a) a = 5.0, ω = 1.0, λ = 0
(b) a = 10.0, ω = 1.0, λ = 0

0.0025. As expected, the shock region in these figures is better resolved than in the previous figures, corresponding to a Δs_{min} of 0.0030. One striking difference between these two cases is near the trailing edge wake area. The adaptive boundary derivative matching term f_3 is included in the weighting function F(s) in Figures 8a and 9a, but is not included in Figures 8b and 9b. With f_3 turned on (Figure 8b), the large density solution errors (and presumably all other solution errors) present in the trailing edge zone of Figure 8a appear to reduce to the level of the original fixed grid solution (Figure 3a). A disadvantage of function f_3 can be seen in the adapted grid of Figure 8b, however. The addition of f_3 into F(s) seems to shear the cells upstream of the trailing edge (i.e., near the boundaries of the adaptive domain), more prominently on the lower surface of the airfoil. Perhaps this can be corrected by using a modified form of the function f_3.

In an attempt to validate both the location and strength of the shocks predicted on the adapted grids of Figure 8, the resulting Cp-curves from these grids were compared with data recently generated by Coakley[18]. Coakley has applied an implicit second-order upwind scheme to the Euler equations for identical flow conditions past the NACA0012 airfoil. The C-type grid used in that study was of nearly the same dimensions as the test grid of this study, although grid point clustering near the shock region was slightly higher on Coakley's grid.

Due to the differences of the finite difference structures employed in each method, it was anticipated that an implicit upwind scheme would better resolve a shock than an ADI-type scheme on similar grids. The curves in Figure 10a show this to be the case. As expected, a well-defined shock is observed in Coakley's data, but not in the data from the original grid. The Cp-curves corresponding to the adaptive grids of Figure 8, however, more closely resemble the data from the upwind scheme. In fact, in Figure 10c, (Δs_{min} = 0.0025), the Cp-curves match remarkably well, particularly in shock locations, which differ by as little as two per cent. There is also good Cp agreement in Figure 10c downstream of the shock and on the lower airfoil surface.

The most disconcerting region of the adaptive grid Cp curve spans from the leading edge to the shock on the upper surface. The discrepancy in Cp values indicates that the adaptive grid Euler flow above the airfoil does not accelerate to the Mach number realized with the upwind scheme. Assuming that this problem was due to truncation errors induced by insufficient point clustering at the leading edge, an additional adaptive grid run was made, with Δs_{min} = 0.0025, and with function f_2. Adaption to grid curvature was included to bring more points to the leading edge. As seen in Figure 10d, with higher leading edge clustering, the Cp curve more closely matches Coakley's data, although there is an overshoot in pressure just before the shock on the newly adapted grid. Note also the difference in the trailing edge Cp curves of Figure 10c and 10d. This is attributable to function f_3, which is used only in Figure 10c. Compared with the fixed grid solution of Figure 10a, however, the adaptive grid solutions of Figures 10b-10d more clearly resolve the upper surface midchord shock wave.

RAE2822 Airfoil Results

To further validate the adaption grid scheme, it is desirable to compare numerical adaptive grid results with empirical data. Consequently, numerical solutions were obtained for flow conditions equivalent to those presented in an empirical study by Cook et al.[19]. In that experiment, extensive boundary layer, wake and pressure measurements were made for transonic flow past an RAE2822 airfoil. The particular case selected for numerical comparison corresponded to turbulent steady flow, with a Mach number M_∞ =.73, an angle of attack α=3.19° and a Reynolds number equal to 6.52 million. Experimentally, the boundary layer was tripped at a distance of 0.03 chord lengths downstream of the leading edge. Under these conditions, the boundary layer did not separate from the airfoil, and a weak midchord shock formed on the airfoil upper surface. Considering these facts and the high Reynolds number of the flow, it was reasonable to assume that the flow could be modeled by an inviscid approximation.

For this reason, numerical solutions were again generated from Tassa's Navier-Stokes code in the Euler mode. Except for the angle of attack, which was reduced to α=2.57° from a wall interference correction formula suggested by Cook et al., and except for the Reynolds number, flow conditions used were identical to those of the experiment. The inner detail of the initial grid used for these results is presented in Figure 11a. Only the steady-state solution was of interest, so the solution was advanced from impulsive free stream conditions using variable time steps, and was seen to converge after several hundred integrations. The density contours for this case, presented in Figure 12a, give no indication of any shock formation. With hopes of defining a shock, the flow field was solved again from an impulsive start, this time with the adaptive grid solver employed after every 20 iterations. The minimum spacing constraint was set at Δs_{min} =0.005, and both functions f_2 and f_3 were turned off. The converged adaptive grid for this case is shown in Figure 11b. The large Δs_{min} constraint selected here prohibits grid point clustering to the degree seen in previous NACA0012 cases. Nevertheless, aside from the leading edge region, the highest point density appears to be just downstream of the upper surface midchord region. As before, this high clustering region corresponds to a shock wave, seen in Figure 12b.

The numerical and experimental results of this flow are presented together in Figure 13. The first of these figures compares empirical surface pressures with numerical surface pressures obtained on the original grid. Outside the first 20 per cent of the airfoil, the empirical and numerical curves do not compare very well. Figure 13b compares the experimental data with the solution obtained on the adapted grid of Figure 12. The Cp curves compare very well along the entire lower surface of the airfoil. In addition, the locations of the shock are in very near agreement. As

is often observed, the shock location in the Euler solution is downstream of the experimental location, although the distance here is rather small, less than 10 per cent of one chord length.

Once again, the largest discrepancy in the Cp curves is on the airfoil upper surface upstream of the shock. No attempt was made to run another case with induced clustering at the leading edge as was done for the NACA0012 airfoil in the last section. One additional source of error may have been in the effective angle of attack used computationally. It is conceivable that better results would be realized if the lift coefficients C_L rather than the angles of attack are matched in the numerical and experimental comparisons. Despite this one region of poor Cp correlation, the adaptive grid routine has indeed clustered points sufficiently to resolve the shock wave seen in the experimental data but not seen with the fixed grid.

DISCUSSION

The practicality of a 1-dimensional adaptive grid scheme applied to a 2-dimensional transonic airfoil problem has been demonstrated in this work. Advantages of a 1-D scheme are numerous. Besides being easy to formulate and easy to understand, this algorithm can be attached to an existent code with relatively few problems. Additionally, the added computational effort needed for the scheme is minor. One run through the adaption subroutine took less CPU time than one half of one time integration step. With the adaptive routine utilized after every 20 to 30 integration steps, the increase in CPU time was only about two per cent. Furthermore, solution convergence rates with and without adaption were nearly the same when a solution was run with a fixed Δt time step.

Unfortunately, there also disadvantages of a 1-D adaption scheme on a 2-D problem. The foremost problem is that in the equidistribution formulation, the final grid point spacing along one η = constant curve is almost totally independent of the adaptive distribution of every other curve. As a result, grid intersections are rarely orthogonal, and sometimes become so skewed that the solution diverges. Grid skewness is evident in Figure 8b, and to some extent, in Figure 8a. Near the airfoil leading edge, where arc curvature is at its highest, skewness often became a problem. With the skewness came large solution errors, which induced even more skewness, due to the form of the selected weighting function F(s). It was almost always necessary to use conservative (large) values of Δs_{min} to insure a convergent adaptive grid. Because of this inherent skewness problem, a technique which enforces orthogonality at the grid intersection would be highly desirable.

Finally, as mentioned earlier, grid speeds in the transformed equations of motion were set equal to zero in this study, and as a result, the solution vector was interpolated after each adaptive sweep. Neglecting grid speeds prevented time-accurate solutions from being obtained, however. If the grid speeds were indeed calculated from backward differences and included in the equations of motion, time accurate convergence rate studies could be made, and the true effect of the adaptive scheme on total computational time could be determined.

ACKNOWLEDGEMENTS

This work was jointly supported under the Iowa State University/Lockheed-Georgia Flight Sciences Program and by the Air Force Office of Scientific Research under grant AFOSR-83-0167.

REFERENCES

1. D. A. Anderson. "Adaptive Grid Methods for Partial Differential Equations." Presented at the 1983 ASME Applied Mechanics, Bioengineering and Fluids Engineering Conference, Houston, Texas, June, 1983.

2. M. M. Rai and D. A. Anderson. "Application of Adaptive Grids to Fluid Flow Problems with Asymptotic Solutions." AIAA 81-0114. Presented at the AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, January, 1981.

3. D. A. Anderson. "Adaptive Mesh Schemes Based on Grid Speeds." AIAA 83-1931. Presented at the AIAA 6th Computational Fluid Dynamics Conference, Danvers, Massachusetts, July, 1983.

4. J. U. Brackbill. "Coordinate System Control: Adaptive Meshes." Symposium on the Numerical Generation of Curvilinear Coordinate Systems and use in the Numerical Solution of Partial Differential Equations, Nashville, Tennessee, 1982.

5. J. Saltzman. "A Variational Method for Generating Multidimensional Adaptive Grids." DOE/ER/03077-174. Courant Mathematics and Computing Laboratory, New York University, February, 1982.

6. H. A. Dwyer. "Grid Adaption for Problems with Separation, Cell Reynolds Number, Shock-Boundary Layer Interaction, and Accuracy." AIAA 83-0449. Presented at the AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, January, 1983.

7. P. A. Gnoffo. "A Vectorized, Finite-Volume Adaptive Grid Algorithm Applied to Planetary Entry Problems." AIAA 82-1018. Presented at the AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri, January, 1981.

8. D. A. Anderson, J. C. Tannehill and R. H. Pletcher. Computational Fluid Mechanics and Heat Transfer. Hemisphere Publishing Company, New York, New York, 1984.

9. Y. Tassa. "An Implicit Method for Solving the Navier-Stokes Equations with Application to Shock Boundary Layer Interaction." Lockheed Georgia Report LG79RR001, 1979.

10. J. L. Steger. "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils." AIAA 77-665. Presented at the AIAA 10th Fluid and Plasmadynamics Conference, Albuquerque, New Mexico, June, 1977.

11. W. R. Briley and H. McDonald. "Solutions to the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method." Journal of Computational Physics, 24 (August, 1977), 372-397.

12. J. Douglas and J. E. Gunn. "A General Formulation of Alternating Direction Methods." Numerische Math., 6 (1967), 428.

13. Y. Tassa and D. M. Schuster. "Navier-Stokes Solutions of Shock Boundary Layer Interaction at Transonic Speeds." Second Symposium on Numerical and Physical Aspects of Aerodynamics Flows, Long Beach, California, January, 1983.

14. D. M. Schuster and Y. Tassa. "Computed Static Stall of a Supercritical Airfoil." Third International Symposium on Flow Visualization, University of Michigan, Ann Arbor, Michigan, September, 1983.

15. Y. Tassa and N. L. Sankar. "Dynamic Stall of an Oscillating Airfoil in Turbulent Flows Using Time Dependent Navier-Stokes Solver." IUTAM Symposium on Unsteady Turbulent Shear Flow, Toulouse, France, May, 1981.

16. P. D. Thomas and J. F. Middlecoff. "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations." AIAA Journal, 18 (1980), 652-656.

17. J. F. Thompson, F. C. Thames and C. M. Mastin. "Automatic Numerical Generation of Body Fitted Curvilinear Coordinate Systems for Fields Containing any Number of Arbitrary Two-Dimensional Bodies." Journal of Computational Physics, 15 (1974), 299-319.

18. T. J. Coakley. "Implicit Upwind Methods for the Compressible Navier-Stokes Equations." AIAA 83-1958. Presented at the AIAA 6th Computational Fluid Dynamics Conference, Danvers, Massachusetts, July, 1981.

19. P. H. Cook, M. A. McDonald and M. C. P. Firmin. "Aerofoil 2822- Pressure Distributions, and Boundary Layer and Wake Measurements." AGARD Advisory Report No. 138, May, 1979.

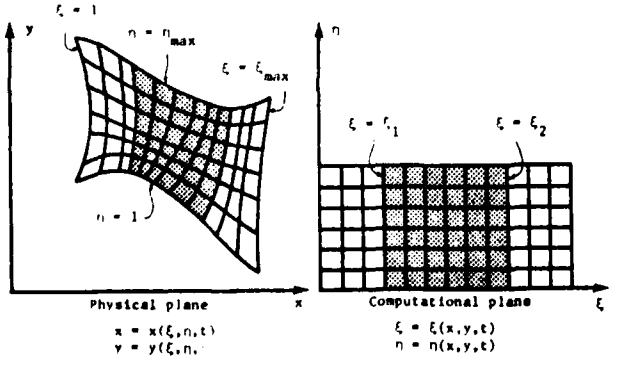


Fig. 1 Physical and computational domains

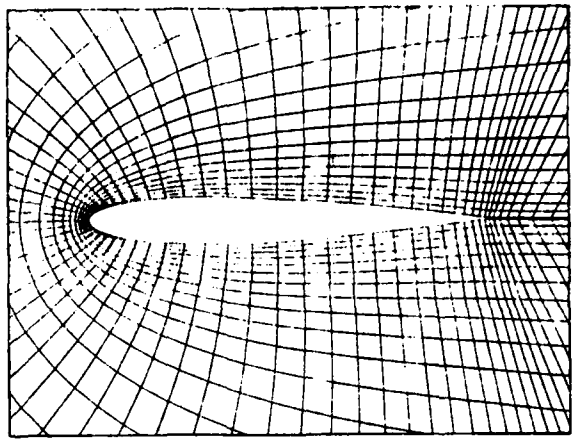


Fig. 2 Test grid ($-0.2 \leq x \leq 1.2$)

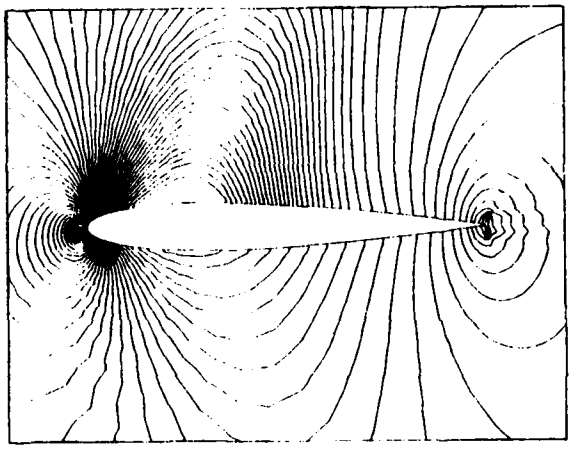


Fig. 3a Density contours of converged solution

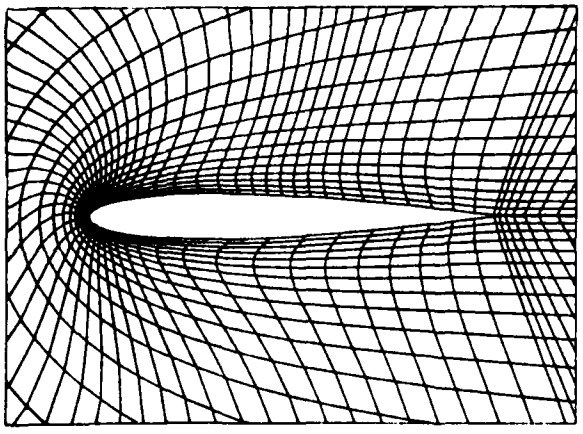


Fig. 3b Test grid adapted to density gradient (f_1)

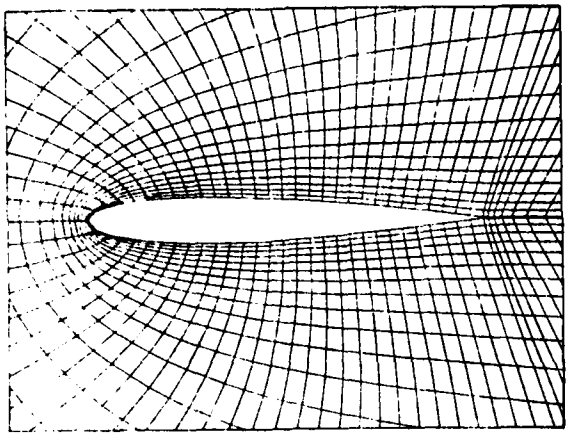


Fig. 4 Adapted test grid with F(s)=1.

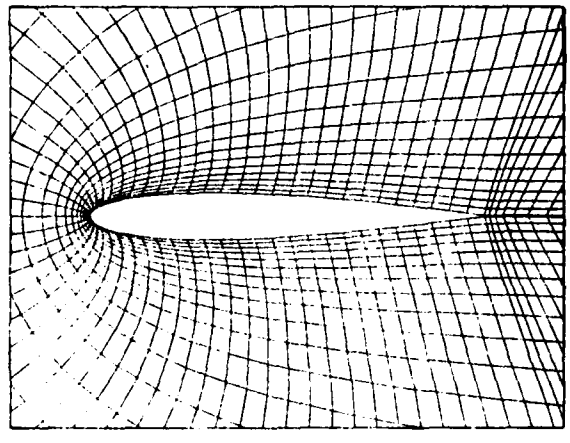


Fig. 5 Adaption to $f_2(s)$ (B=0.1)

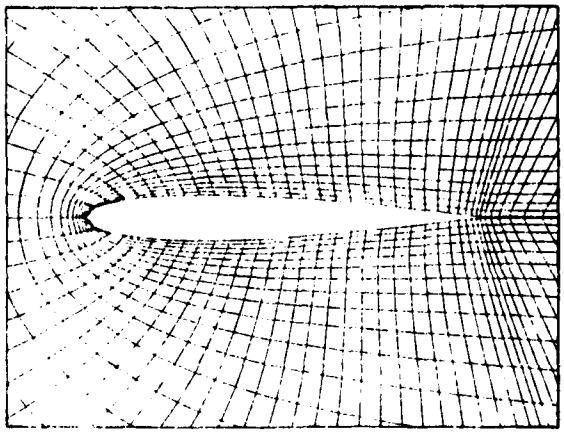


Fig. 6a Adaption to $f_3(s)$

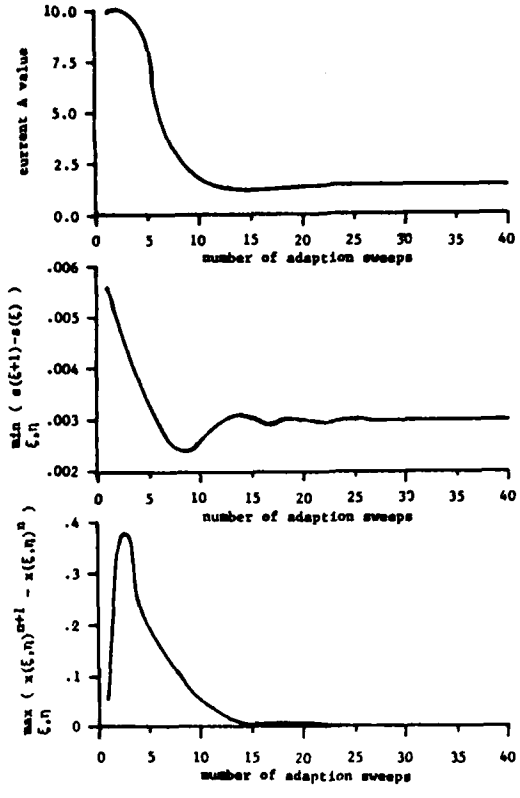


Fig. 7 Adaptive grid convergence (Δs_{min} = 0.003)

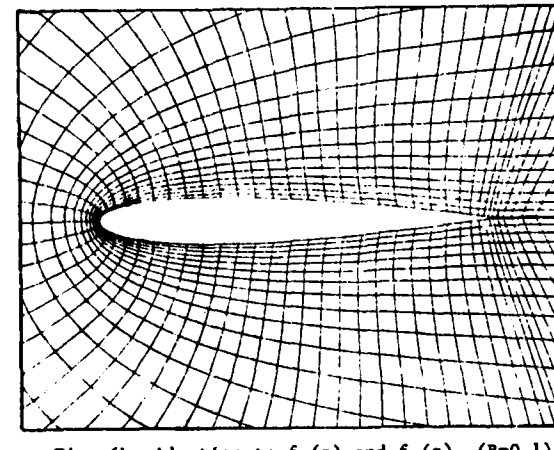
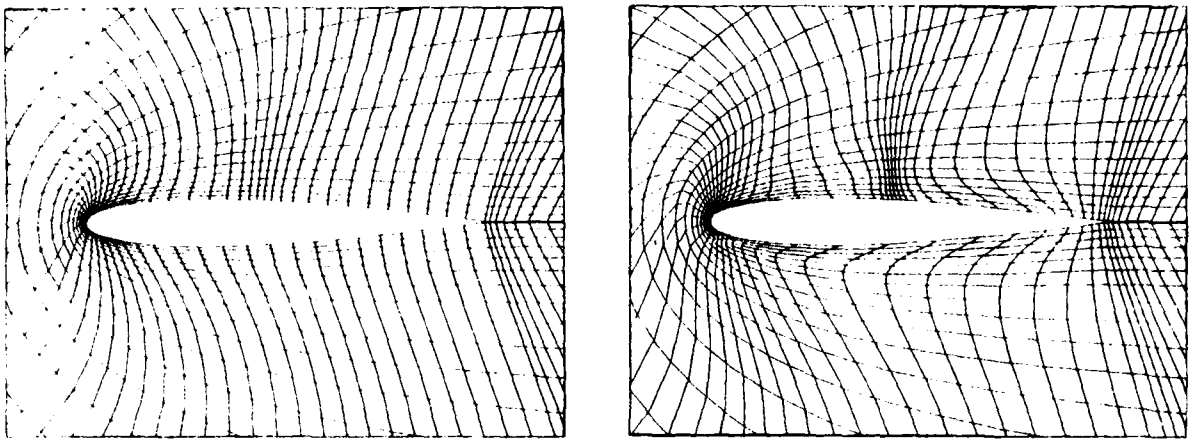
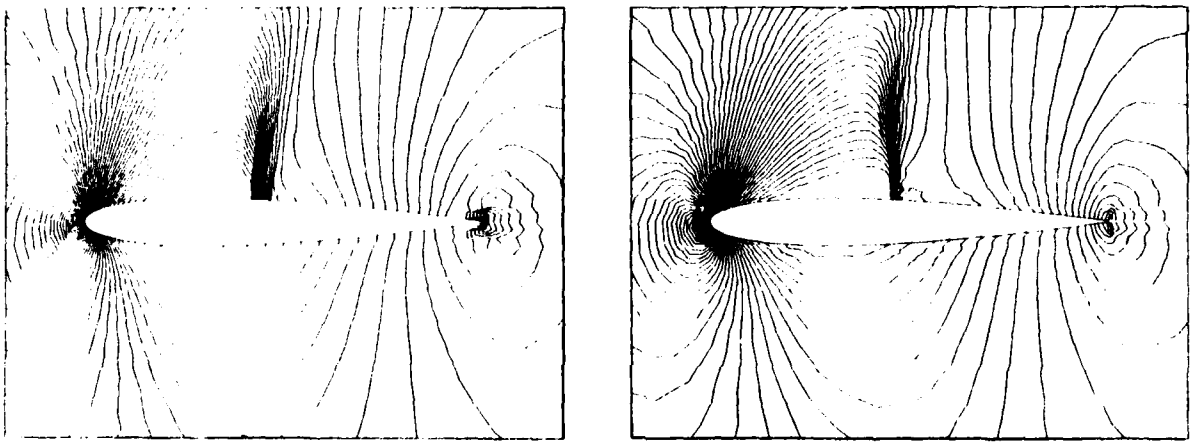


Fig. 6b Adaption to $f_3(s)$ and $f_2(s)$ (B=0.1)



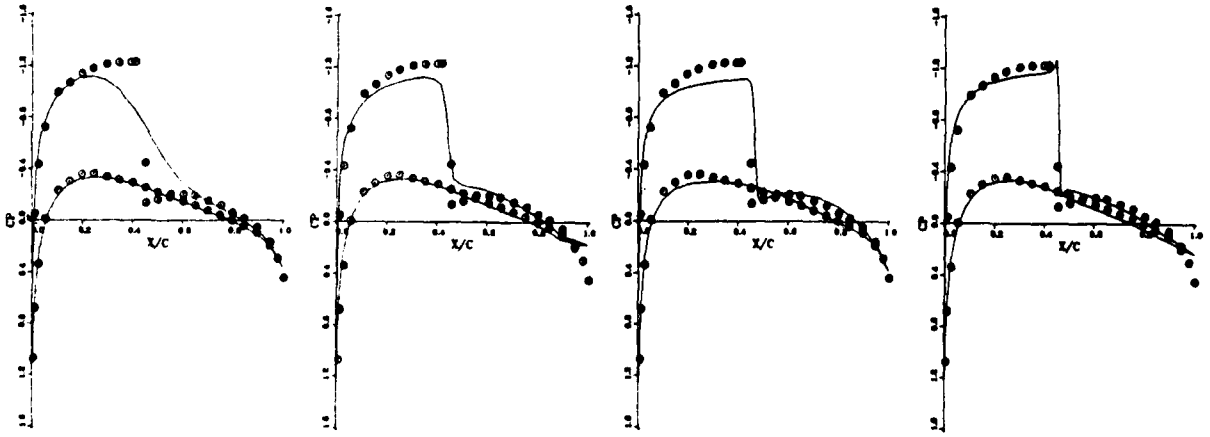
a. $\Delta s_{min} = 0.003$ b. $\Delta s_{min} = 0.0025$

Fig. 8 NACA0012 airfoil adaptive grids



a. $\Delta s_{min} = 0.003$ b. $\Delta s_{min} = 0.0025$

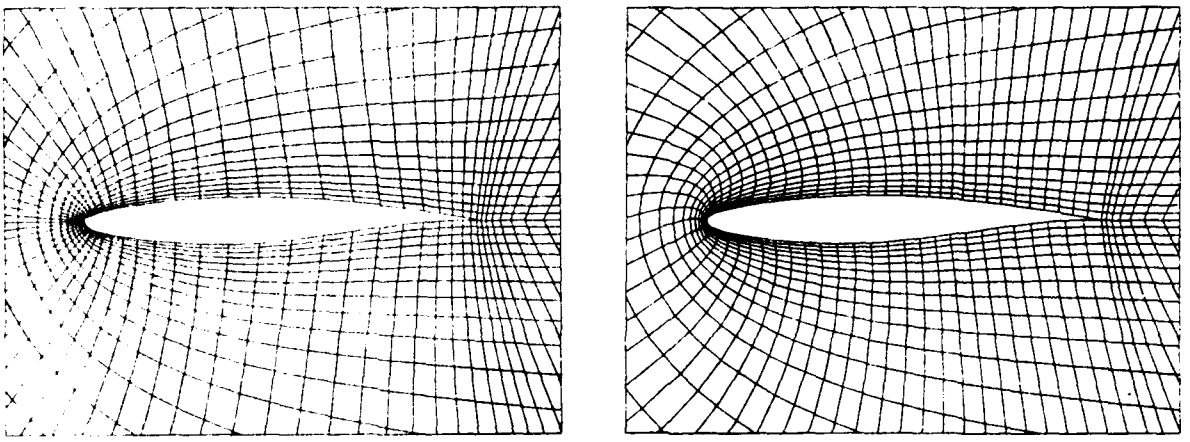
Fig. 9 NACA0012 airfoil density contours



a. Test grid b. $\Delta s_{min} = 0.003$ c. $\Delta s_{min} = 0.0025$ d. $\Delta s_{min} = 0.0025$, $\beta = 0.08$

(———— adaptive grid, ooooo Coakley)

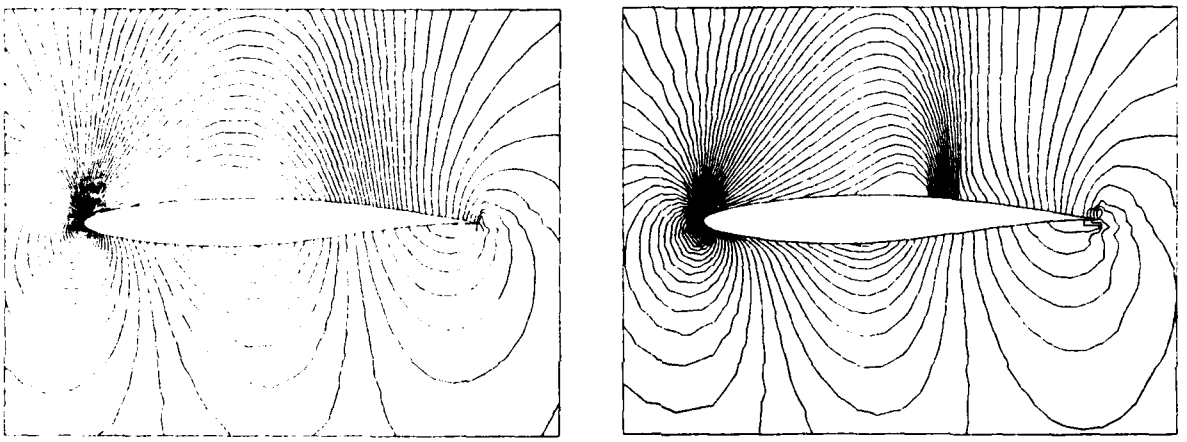
Fig. 10 NACA0012 airfoil Cp-distributions



a. Original grid

b. $\Delta s_{min} = 0.005$

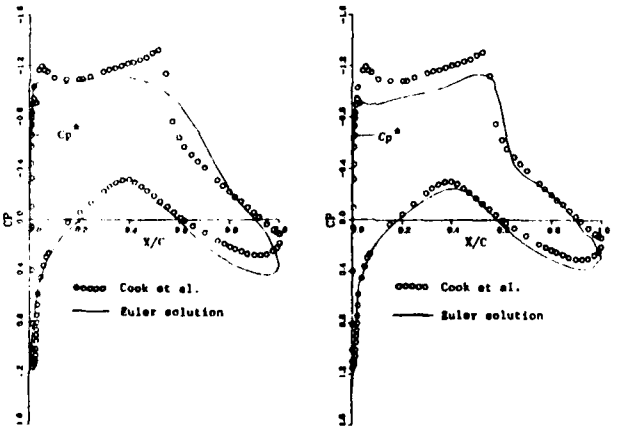
Fig. 11 RAE2822 airfoil grids



a. Original grid

b. $\Delta s_{min} = 0.005$

Fig. 12 RAE2822 airfoil density contours



a. Original grid

b. $\Delta s_{min} = 0.005$

Fig. 13 RAE2822 airfoil Cp-distributions

END

FILMED

4-85

DTIC