

AD-A253 723



1

WL/MN-TR-91-83

Analysis and Control of Grid Quality in Computational Simulation

**J. F. Thompson
C. W. Mastin
B. Gatlin**

**Engineering Research Center
Mississippi State University
Mississippi State MS 39762**

DTIC
ELECTE
JUL 30 1992
S A D

JUNE 1992

FINAL REPORT FOR PERIOD MARCH 1989 - SEPTEMBER 1991

92-20650



Approved for public release; distribution is unlimited.

WRIGHT LABORATORY, ARMAMENT DIRECTORATE

Air Force Systems Command ■ United States Air Force ■ Eglin Air Force Base

92 7 2 004

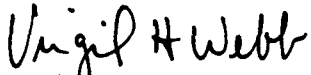
NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise as in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service (NTIS), where it will be available to the general public, including foreign nationals.

FOR THE COMMANDER


VIRGIL H. WEBB, Col, USAF
Chief, Weapon Flight Mechanics Division

Even though this report may contain special release rights held by the controlling office, please do not request copies from the Wright Laboratory, Armament Directorate. If you qualify as a recipient, release approval will be obtained from the originating activity by DTIC. Address your request for additional copies to:

Defense Technical Information Center
Cameron Station
Alexandria VA 22304-6145

If your address has changed, if you wish to be removed from our mailing list, or if your organization no longer employs the addressee, please notify WL/MNAA, Eglin AFB FL 32542-5000, to help us maintain a current mailing list.

Do not return copies of this report unless contractual obligations or notice on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 1992	3. REPORT TYPE AND DATES COVERED Final Mar 89 - Sep 91		
4. TITLE AND SUBTITLE Analysis and Control of Grid Quality in Computational Simulation		5. FUNDING NUMBERS C: F08635-89-C-0209 PE: 62602F PR: 2567 TA: 03 WU: 26		
6. AUTHOR(S) J. F. Thompson, C. W. Mastin, and B. Gatlin				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Engineering Research Center Mississippi State University Mississippi State MS 39762		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Wright Laboratory, Armament Directorate Weapon Flight Mechanics Division Aerodynamics Branch (WL/MNAA) Eglin AFB FL 32542-5000		10. SPONSORING/MONITORING AGENCY REPORT NUMBER WL/MN-TR-91-83		
11. SUPPLEMENTARY NOTES Availability of this report is specified on verso of front cover.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE A		
13. ABSTRACT (Maximum 200 words) The objective of this project was to determine and formulate true measures of grid quality that can be used to evaluate numerically generated grids. Empirical studies were conducted on the relationships between grid metric quantities and actual numerical solution error for physical field problems for which experimental data was available. Correlations of the actual solution error were made for numerical solutions of partial differential equations for which analytical solutions were available. Emphasis was placed on equations that model some aspect of some physical problems. In these studies, detailed localized comparisons were made throughout the field. This is particularly important since certain truncation error relations may be important only in regions of high gradient. Correlations of actual solution error were made for physical field problems. The EAGLE grid and flow codes were used as test beds in this task. General graphical tools for grid analysis and procedures for automatic adjustment of the grid to improve the quality were developed. These tools and procedures were incorporated in the EAGLE code.				
14. SUBJECT TERMS Numerical Grid Generation Grid Quality Error Estimation		15. NUMBER OF PAGES 89		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

PREFACE

This program was conducted by Mississippi State University, Mississippi State MS 39762, under Contract F08635-89-C-0209, with the Air Force Wright Laboratory, Armament Directorate, Eglin Air Force Base FL 32542-5000. Dr. Lawrence E. Lijewski, WL/MNAA, managed the program for the Armament Directorate. The Principal Investigator was Dr Joe F. Thompson. Dr C. Wayne Mastin and Dr Boyd Gatlin were Associate Investigators, and Phu Luong was the Graduate Assistant. The program was conducted during the period from March 1989 through September 1991.

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

DTIC QUALITY INSPECTED 2

TABLE OF CONTENTS

I. INTRODUCTION	1
II. GRID REFINEMENT	5
III. ADAPTIVE GRID GENERATION	8
Variational approach	8
Control function approach	9
IV. GRID QUALITY MEASURES AND SOLUTION ERROR	12
Skew angle	12
Aspect ratio	12
Laplacian	13
Arc length	13
Error estimations	14
V. RESULTS AND DISCUSSIONS	17
Static adaptation results	17
Static adaptation to quality measures	18
Static adaptation to flow solution	18
Multiple adaptation results	20
Multiple adaptation for the airfoil RAE 2822	20
Multiple adaptation for a double wedge.....	21
Multiple adaptation for a wind tunnel.....	24
Grid refinement	25

VI. CONCLUSIONS	27
REFERENCES	28
APPENDICES	
A) EAGLE ADAPTIVE COMMANDS.....	31
B) EAGLE ADAPTIVE NAMELISTS	33
C) INCORPORATION INTO FLOW CODES.....	42
D) FLOW CODE ENHANCEMENTS.....	46

I. INTRODUCTION

This project has involved several investigators over the duration of the contract and has touched on several fields of research. The overriding theme of the research has been on grid quality and the improvement of numerical simulation through control of error. This has been achieved by making major modifications to the EAGLE grid code and the EAGLE flow code that is referred to as the MISSE code. For the most part, these modifications are transparent to the user.

In the earlier form of the adaptive EAGLE system [1,2], the coupling of the adaptive grid system with a computational fluid dynamics (CFD) code required the encapsulation of both the entire grid code and the CFD flow code into separate subroutines and the construction of a driver to call each. This was inefficient in that it included some unnecessary parts of the grid code and required significant modification, and perhaps restructuring, of the CFD code. In particular, the flow code arrays and/or the grid code arrays had to be modified to be compatible in structure.

In the effort described here, the elliptic grid generation procedure in the EAGLE grid code has been separated from the main code into a subroutine, and a new subroutine that evaluates several grid quality measures at each grid point has been added. The elliptic grid routine can now be called either by a CFD code to generate a new adaptive grid based on flow variables and quality measures through multiple adaptation or by the EAGLE main code to generate a grid based on quality measure variables through static adaptation. Arrays of flow variables can be read into the EAGLE grid code for use in static adaptation as well.

These major changes in the EAGLE adaptive grid system make it easier to convert any CFD code that operates on a block-structured grid (or single-block grid) into a multiple adaptive code. The conversion procedure is accomplished by adding the elliptic grid generation subroutine, and certain other subroutines from the EAGLE grid system that are involved in the elliptic grid generation process, to the flow code. The CFD code may then call the elliptic grid generation routine at each time step when a new grid is desired. The CFD code passes its current solution to this EAGLE routine via a scratch file, and the EAGLE routine returns the new grid to the CFD code also via the scratch file. One restriction is that the initial grid must be generated by the EAGLE system or be processed through that system, which provides the necessary parameters and structural information to be read from files by the adaptive EAGLE routines. This structure eliminates the need for compatibility between CFD and grid arrays.

In the present work, the control function approach is used as the basic mechanism for the adaptive grid generation. The static and multiple adaptive grid generation techniques are investigated by formulating the control functions in terms of either grid quality measures, the flow solution, or both.

Previous work [1,2] allowed the grid to only adapt to the gradient of a variable. The work described here has extended this adaptive mechanism to also allow adaptation to the curvature of a variable or to the variable itself. The system provides for different weight functions in each coordinate direction. In addition, the mechanism now includes the ability to calculate the weight functions as weighted averages of weight functions from several flow variables, and/or quality measures. This allows the adaptation to take into account the effect of many of the flow variables instead of just one. The construction of the weighted average of flow variables and quality measures and the choice of adaptation to gradient, curvature, or variable are all controlled in each coordinate direction through input parameters.

The quality measures now available in the EAGLE grid system are skewness, aspect ratio, arc length, and smoothness of the grid. These grid quality measures, and the resulting control and weight function values, can be output for graphical contouring.

In addition to the above adaptation using the elliptic grid equations, a grid refinement capability is introduced. When a grid is constructed, the MISSE flow solver may, in each block, compute a solution on the input grid or compute a solution on a refined or coarsened grid. This grid refinement or coarsening may be done in any or all three coordinate directions. Thus there exists considerable generality in refinement strategies for a multiblock grid. However, the complete generality of a true local grid refinement scheme that would lead to a more complicated data structure does not exist.

An error estimate is a desirable, though often unattainable, component in an adaptive grid procedure. Two techniques have been developed to deal with error estimation. One method uses the grid refinement capability previously described. Under certain conditions, the error in a numerical solution can be estimated by comparing fine and coarse grid solutions. Another method of error estimation has been included. The method is based on the premise that the truncation error in a difference approximation can be estimated by comparing coarse and fine grid difference operators evaluated on the same numerical solution.

Section II contains a discussion of grid refinement and coarsening. It also includes a description of the interpolation schemes that are used. The basic idea and the governing equations that lead to adaptive grid generation are presented in Section III. Section IV discusses the development of several grid quality measures and the truncation error estimation techniques for use in the construction of adaptive grid generation. A discussion of the results of static adaptation of several grids is presented in Section V. Multiple adaptation performed with the adaptive MISSE Euler flow code [3] for several configurations are also discussed

in that section. Finally, conclusions and recommendations are made in Section VI. The use of new EAGLE commands and namelists along with several examples are given in Appendices A and B. The procedure for the incorporation of adaptive EAGLE into flow codes is given in Appendix C. Other improvements and enhancements that have been made on the flow code are given in Appendix D.

II. GRID REFINEMENT

The purpose of an adaptive grid is to reduce the error in a numerical computation. An adaptive grid can be constructed by either locally refining and/or coarsening a grid or by moving the existing grid points. A local grid refinement procedure can be easily incorporated into a grid or flow code provided the refinement strategy is applied to complete blocks.

The grid refinement procedure involves a sequence of one-dimensional interpolations. Suppose a grid is defined by a curvilinear coordinate system with physical variables x, y, z and computational variables $\xi^i, i = 1, 2, 3$. By splining the existing physical coordinate values, any number of new points can be inserted by evaluating the spline function at new values for the curvilinear coordinates. The new points will lie on a spline curve. It should be noted that new boundary grid points may also be inserted by the refinement procedure. These points may not lie on the actual boundary surface that was used to construct the original grid. However, one should try to preserve the characteristics of the physical boundary as far as possible. For this reason the choice of the spline function is important, and the Akima spline has been used in this work to avoid the oscillations that may occur with other cubic spline methods. In the process of developing an efficient grid for a particular problem, it may be decided that some of the grid blocks have too many points and it may be desirable to coarsen a grid in certain coordinate directions. This can also be done as a sequence of one-dimensional operations. For example, along the grid lines in a single coordinate direction, every other grid point can be deleted. This can then be repeated along the remaining grid lines in the other directions. In order to minimize the modifications to the MISSE flow code, the cases that have been implemented involve

only halving and doubling the spacing in the computational region. However, repeated refinements could be done on successive restarts of a numerical solution.

One of the objectives in developing the grid refinement procedure is to allow for selective refinement and coarsening of a grid as a solution develops. While this has not been automated, the user has the option of examining the solution and restarting with a finer or coarser grid in selected blocks. This, in particular, allows for an initial solution to be started on a coarse grid and then a final solution to be computed on a refined grid. In order to accomplish this, there must be the capability of interpolating between coarse and fine grids. That capability has been incorporated into the flow code. Since the option exists for refinement in any or all coordinated directions, the interpolation is also done in a sequence of one-dimensional operations. A simple linear interpolation in computational space is used.

Consider a function q_k , $k = 1, \dots, k_{max}$ defined at the cell centers of a one-dimensional grid. On a grid refinement, the new cell centers are located at points that will be indexed as $k + \frac{1}{4}$ and $k + \frac{3}{4}$. The function values at these points are computed as

$$q_{k+\frac{1}{4}} = \frac{3}{4}q_k + \frac{1}{4}q_{k+1},$$

$$q_{k+\frac{3}{4}} = \frac{1}{4}q_k + \frac{3}{4}q_{k+1}.$$

With a cell-centered solution algorithm, as is considered here, it is also necessary to use interpolation to transfer values to a coarser grid. Reversing the above process, if a coarse grid solution value q_k is needed, it can be computed from fine grid solution values by the formula

$$q_k = \frac{1}{2}(q_{k-\frac{1}{4}} + q_{k+\frac{1}{4}}).$$

The interpolation procedure is repeated in each coordinated direction in which the grid has been refined or coarsened.

The process of grid refinement and coarsening may destroy the continuity of grid lines at block boundaries. Therefore, a similar interpolation scheme has been introduced to handle the block boundary interface.

III. ADAPTIVE GRID GENERATION

Variational approach

Among the various adaptive grid generation techniques, the minimization of the integral of some grid property over the computational domain is known as the variational approach. The resulting Euler variational equations from the calculus of variations will constitute the grid generation system. The choice of what property is to be minimized depends on what is expected from the grid. For example, Brackbill and Saltzman [4] constructed this adaptive grid by minimizing a weighted combination of integrals that emphasize smoothness, orthogonality, and concentration of the grid. A similar approach considering smoothness, a measure of the grid cell area, and the orthogonality of the grid lines can be found in Roache and Steinberg [5]. Several other grid properties that might be considered, such as square of cell volume and inverse cell volume, can be found in Warsi and Thompson [6]. By solving a large-scale nonlinear minimization problem using a conjugate gradient method, the orthogonality of the interior and boundary angles has been controlled, as described by Castillo [7].

Due to the complexity of the resulting Euler variational equations, they are difficult to solve and solution algorithms may not converge. Kennon and Dulikravich [8] and Carcaillet, et al. [9] have developed algorithms for the direct solution of the variational problem. In these two reports, different discrete problems were formulated as unconstrained optimization problems and then solved by a conjugate gradient iterative method. A survey of the types of integrals that may be included in a variational problem, and the geometric properties that each integral imposes upon the grid, can be found in Soni and Mastin [10].

Control function approach

The control function approach to adaptation is developed by noting that the one-dimensional form of the elliptic grid generation system can be written as

$$x_{\xi\xi} + Px_{\xi} = 0, \quad (3.1)$$

and the differentiated form of the equidistribution principle, $Wx_{\xi} = \text{constant}$,

$$Wx_{\xi\xi} + W_{\xi}x_{\xi} = 0, \quad (3.2)$$

where P is the function to control the coordinate line spacing, and W is the weight function.

From (3.1) and (3.2), the control function can be defined in terms of the weight function and its derivative as

$$P = \frac{W_{\xi}}{W}. \quad (3.3)$$

This equation can be extended in a general three-dimensional form as

$$P_i = \frac{W_{\xi^i}}{W}. \quad (3.4)$$

This approach was suggested by Anderson [11,12] and has been applied with success for two-dimensional configurations by Johnson and Thompson [13,14] and for three-dimensional configurations by Kim and Thompson [1] and by Tu and Thompson [2].

The complete generalization of (3.4) was proposed by Eiseman [15] as

$$P_i = \sum_{j=1}^3 \frac{g^{ij}}{g^{ii}} \frac{(W_i)_{\xi^j}}{W_i}, \quad (3.5)$$

where W_i is the weight function chosen for the ξ^i direction. This definition of the control functions provides a convenient means to specify three separate control functions, with one in each coordinate direction.

In order to preserve the geometrical characteristics of the existing grid, it is practical to construct the control functions in such a manner that the control functions defined by (3.5) are added to the initial set of control functions obtained from the geometry, i.e.,

$$P_i = (P_i)_g + C_i(P_i)_w, \quad i = 1, 2, 3, \quad (3.6)$$

where

$(P_i)_g$: control function based on geometry

$(P_i)_w$: control function based on weight function

(C_i) : weight coefficient to be specified

In these equations the weight function W can be computed by different formulas for different adaptive mechanisms.

Adaptation to

$$\text{Variable} : W = 1 + |V|$$

$$\text{Gradient} : W = 1 + |\nabla V| \quad (3.7)$$

$$\text{Curvature} : W = (1 + \beta |K|) \sqrt{1 + \alpha |\nabla V|^2}$$

where V can be either a flow solution variable or a grid quality measure. Here β, α are in the interval $[0, 1]$ and

$$K = \frac{\nabla^2 V}{(1 + |\nabla V|^2)^{3/2}} \quad (3.8)$$

is the curvature of the variable V .

Using these definitions of the control functions, the elliptic generation system given by

$$\nabla^2 \xi^i = g^{ii} P_i, \quad i = 1, 2, 3, \quad (3.9)$$

or transformed to the computational region as

$$\sum_{i=1}^3 \sum_{j=1}^3 g^{ij} \bar{r}_{\xi^i \xi^j} + \sum_{k=1}^3 g^{kk} P_k \bar{r}_{\xi^k} = 0, \quad (3.10)$$

that becomes an adaptive grid generation system. This system (3.10) is then solved iteratively in adaptive EAGLE by the point successive overrelaxation (SOR) method to generate the adaptive grid.

IV. GRID QUALITY MEASURES AND SOLUTION ERROR

The objective of this part of the investigation is to develop a means of evaluating grids through the computation of certain grid properties that are related to grid quality and to develop techniques for estimating the truncation error.

Following Kerlick and Klopfer [17] and Gatlin [18], the grid quality measures are taken as skew angle, aspect ratio, grid Laplacian, and arc length. Techniques for estimating the truncation error due to the work of Mastin [19] are also included. At each grid point in a general three-dimensional grid, each property can have three values associated with the three directions. The approach taken under this investigation is to treat each surface of constant ξ^i separately for ease in graphical interpretation.

Skew angle

The minimum skew angle between intersecting grid lines is one of the most important measurable grid properties. This angle can be expressed in terms of the covariant metric elements as

$$\theta_{ij} = \cos^{-1} \left\{ \frac{g_{ij}}{\sqrt{g_{ii}g_{jj}}} \right\}. \quad (4.1)$$

Since $g_{12} = g_{21}$, $g_{13} = g_{31}$, and $g_{23} = g_{32}$, the three skew angles associated with each grid point in a three-dimensional grid are θ_{12} , θ_{23} , and θ_{31} . Figure 1 illustrates the skew angle at a grid point on a surface of constant ξ^3 .

Aspect ratio

Since aspect ratio is the ratio of the lengths of the sides of a grid cell, it can be defined in two different ways. For example, on a surface of constant ξ^3 , this

ratio can be expressed in terms of metric elements g_{ii} and g_{jj} as

$$AR_{ij} = \sqrt{\frac{g_{ii}}{g_{jj}}} \quad (4.2a)$$

or

$$AR_{ji} = \sqrt{\frac{g_{jj}}{g_{ii}}} \quad (4.2b)$$

Large changes in aspect ratio of grids from one part of the field to another may inhibit the convergence of viscous flow solutions to a steady state. The arc lengths involved in the geometric representation of AR_{ij} and AR_{ji} on a surface of constant ξ^3 are illustrated in Figure 2.

Laplacian

A useful measure of the smoothness of a grid is the Laplacian of the curvilinear system, $\nabla^2 \xi^i$, $i = 1, 2, 3$, which is simply the rate of change of grid point density in the grid. For a perfectly uniform grid, the grid Laplacian would vanish everywhere, but exceedingly large values may arise in highly stretched grids. The mathematical representation of the grid Laplacian is defined in terms of the contravariant metric elements g^{ij} , the contravariant base vectors $\vec{a}^i$, and the position vector $\vec{r}$ as

$$\nabla^2 \xi^l = - \sum_{i=1}^3 \sum_{j=1}^3 g^{ij} \vec{a}^i \cdot \vec{r}_{\xi^i \xi^j} \quad l = 1, 2, 3. \quad (4.3)$$

Figures 3.1 and 3.2 illustrate the behavior of the grid Laplacian on a surface of constant ξ^3 .

Arc length

Another important measure of the grid quality is the local rate at which grid spacing changes. On a coordinate surface of constant ξ^3 , and along a coordinate line of constant ξ^2 , the grid spacing can be defined as

$$d_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2]^{\frac{1}{2}}. \quad (4.4a)$$

The normalized rate at which grid spacing changes (ARCL) is then

$$(ARCL)_i = \frac{d_i - d_{i-1}}{\frac{1}{2}(d_i + d_{i-1})}. \quad (4.4b)$$

Figure 4 illustrates the geometric representation of $(ARCL)_i$.

Error estimation

Finite element schemes have one advantage over finite difference and finite volume schemes in that rigorous error estimates can be computed in many cases. Although truncation error estimates can be derived for nearly all numerical methods, they frequently involve higher order derivatives of the solution that cannot be estimated with any accuracy using the numerical solution. A different approach is taken here. Based on truncation error estimates that are computed in the multigrid methods, the truncation error is estimated by applying a coarse and a fine grid difference approximation to the numerical solution. Another, more rigorous, approach to estimating error is to compare coarse and fine grid solutions. Such a procedure has long been used in the numerical solution of ordinary differential equations and is referred to as Richardson extrapolation. This approach can now be carried out using the grid refinement procedures, which have been included in the MISSE flow code. Both of these techniques should be considered as heuristic error estimates, which are intended only to give an order of magnitude estimates of the truncation error and the solution error.

The derivation that follows can be applied to the Euler equations for compressible flow and other systems of conservation laws of the form

$$u_t + f_x + g_y + h_z = 0. \quad (4.5a)$$

The transformation of this system to an arbitrary curvilinear coordinate system is

$$U_t + (F_{\xi^1} + G_{\xi^2} + H_{\xi^3}) = 0, \quad (4.5b)$$

where $\sqrt{g}$ is the Jacobian of the transformation and

$$F = \sqrt{g}(\xi_x^1 f + \xi_y^1 g + \xi_z^1 h)$$

$$G = \sqrt{g}(\xi_x^2 f + \xi_y^2 g + \xi_z^2 h)$$

$$H = \sqrt{g}(\xi_x^3 f + \xi_y^3 g + \xi_z^3 h).$$

Let h be the spacing of the fine grid and nh be the spacing of the coarse grid. Let $\mathcal{L}_h$ be the difference approximation operator of order p on the fine grid and $\mathcal{L}_{nh}$ be the same difference approximation operator on the coarse grid. Then the finite difference approximation of the PDE can be represented on the fine grid as

$$u_t + f_x + g_y + h_z = \mathcal{L}_h(F, G, H) + T(h)^p \quad (4.5c)$$

and on the coarse grid as

$$u_t + f_x + g_y + h_z = \mathcal{L}_{nh}(F, G, H) + T(nh)^p, \quad (4.5d)$$

where n is an integer. From (4.5c) and (4.5d), the estimate of the truncation error on the fine grid can be computed as

$$T(h)^p = \frac{\mathcal{L}_{nh} - \mathcal{L}_h}{(1 - n^p)}. \quad (4.6)$$

Richardson extrapolation can be used to compute the error in the numerical solution. Even though numerical solutions must be computed on both fine and coarse grids, the error estimates that result do not have the large peaks at solution singularities that can be encountered with the truncation error described above. Thus the solution error estimates may sometimes be more useful in the construction of adaptive grids.

Assume that there are two numerical solutions of order p accuracy for (4.5b) that have been computed on a fine grid and on a coarse grid, with grid spacing h and nh , respectively, in each coordinate direction. Assuming that the same

p th order method is used in both cases, the relation between the two numerical solutions and the actual solution u of the PDE can be established as

$$u = U_h + R(h)^p \quad (4.7a)$$

and

$$u = U_{nh} + R(nh)^p. \quad (4.7b)$$

From these equations, an extrapolated value of u can be computed as

$$u = \frac{n^p U_h - U_{nh}}{(n^p - 1)}. \quad (4.7c)$$

Thus the estimate of the error in the numerical solution computed on the h grid is:

$$u - U_h = \frac{U_h - U_{nh}}{(n^p - 1)}. \quad (4.7d)$$

V. RESULTS AND DISCUSSION

The adaptive grid generation system based on the control function approach, as described in the previous sections, has been used to generate static and multiple adaptive grids for several geometries. Results are presented in this section. The static adaptive grids were obtained by adapting the initial grids to either grid quality measure variables or to existing flow solution variables. The multiple adaptive procedure was tested on several different configurations with the adaptive MISSE Euler flow code for transonic and supersonic flow cases.

Static adaptation results

Some examples of the grid quality adaptation are shown in Figures 5 through 13, for adaptation to various quality measures. Contour plots of the quality measures are given for comparison with those for the initial elliptic grid shown in Figure 5. (In all these plots, the colors range from blue through green to red as the value of the quantity represented increases.) In these contour plots the Laplacian is a smoothness measure, with blue being the smoothest. In the skew angle plots, red is the most orthogonal. In the aspect ratio plots, green is the closest to unity, while blue and red indicate large ratios in the two directions. Approximately the same number of total adaptive iterations were run in each case. The value given for the variable ITMAX on each figure is the number of inner iterations between reevaluations of the quality and application of further adaptation. The value for variable ITMAXA is the total number of adaptations. The value for AFIXP = NO means the control functions are updated based on the geometry of the previous grid rather than the initial grid, at each adaptation.

Static adaptation to quality measures

Comparison of Figure 6 with Figure 5 shows that adaptation to the skewness is effective in reducing the skewness in one region, while increasing the skewness in other regions of the grid. A small improvement in aspect ratio occurs, but the smoothness of the grid is decreased. Figure 7 shows the difference of the average skew angle between the initial and adaptive grids.

Comparison of Figure 8 with Figure 5 indicates that adaptation to aspect ratio does improve both aspect ratio and smoothness of the grid; the skewness is increased, however. Figure 9 shows the large change in average skew angle for this case. Comparison of Figure 10 with Figure 5 shows that adaptation to smoothness improves the skewness and aspect ratio of the grid effectively, but the adaptive grid is not as smooth as the initial elliptic grid. The average skew angle of the initial and adaptive grids for this case are plotted in Figure 11.

Figures 7, 12, and 13 show the beneficial effect of including adaptation to aspect ratio, arc length, and smoothness with adaptation to skewness: the skewness is reduced more by the combination than with skewness adaptation alone. A little improvement occurs in aspect ratio; the smoothness of the grid does, however, decrease.

Results from these examples show that the adaptation to the combination of all grid quality measures or to each individually can improve some grid properties while damaging others. For example, the adaptation to the Laplacian of this particular grid can reduce the skewness, but the resulting adaptive grid is not as smooth as the initial grid. The choice of the adaptive variable for the adaptation very much depends on what property of the grid needs to be improved and the configurations of the grids.

Static adaptation to flow solution

Another new feature of the adaptive EAGLE grid code, in addition to the static adaptation to the quality measure, is the adaptation to an existing flow

solution. Examples demonstrating this new feature are shown in Figures 14 through 16. A value given for ITMAXA = 3 on each plot indicates the total number of adaptations, and AFIXP = NO indicates the control functions were computed based on the geometry of the previous grid. The density variable from the Euler solution (MISSE Euler code) with a Mach number of 2 on a double wedge (81 by 20) grid is considered in this case. Figure 14 shows the color contour plot of the density obtained from a 200-time-step solution on the initial grid. This grid was then adapted statically to the gradient of the density in one direction with a weight coefficient C_1 of 1.5 and to the curvature of the density in the other direction with a weight coefficient C_2 of 0.5, α of 1, and β of 0.5, (AWT=GRAD,CURV, and RHO=1,1). The color plots of the control functions and weight functions in each direction are also shown in this figure.

Figure 15 shows another case of the grid being adapted statically to the curvature of the density in one direction with a C_1 of 0.5, α of 1, and β of 0.5, and to the gradient of the density in the other direction with a C_2 of 1.5. Figure 16 shows the initial grid and several adaptive grids which result from different adaptive mechanisms and parameters.

There are several significant advantages from these new features of the adaptive EAGLE grid code that cannot be found in the earlier form [1,2]. For example, grid quality measures along with control and weight functions in each direction of the resulting generated grid can be output for graphical representation. This makes it easier for EAGLE users to properly examine each property of the resulting generated grid. Moreover, these grid quality measures can be used in addition to the flow solution in the weighted average during the multiple adaptation process. The capability of static adaptation to an existing flow solution of the adaptive EAGLE grid code is sometimes very helpful in the multiple adaptive process. Since it is not clear what weight functions and weight coefficients or what flow variable or how many adaptive iterations will produce a reasonable

grid for different type of flows, the users can use the static adaptive mode of the EAGLE grid code to determine these parameters. The determination of these parameters prior to submitting a multiple adaptive run of the adaptive flow code can save the users a considerable amount of time.

Multiple adaptation results

Results of multiple adaptation performed with the adaptive MISSE Euler flow code are shown in Figures 17 through 34. In all these plots, NIT is the total number of time steps, INT indicates the number of time steps at which the first adaptation is performed, NCL is the number of time steps between each adaptation, and MAXINT indicates the number of time steps at which the last adaptation is performed. Values of weight functions AWT_1 , AWT_2 ; weight coefficients C_1, C_2 ; adaptive variables density RHO_1 , RHO_2 ; pressure $PRES_1$, $PRES_2$ are given for ξ^1 and ξ^2 directions, respectively. For example, $AWT=GRAD, CURV$, $C_1 = .5$, $C_2 = .3$, $RHO=1,0$, $PRES=0,1$, and $\alpha = 1, \beta = 1$ can be interpreted as the adaptation to the density gradient in ξ^1 direction with a C_1 of .5 and to the curvature of the pressure in ξ^2 direction with a C_2 of .3 and with coefficients of gradient and curvature being $\alpha = 1$ and $\beta = 1$ respectively.

Multiple adaptation for the airfoil RAE 2822

The first case considered here is a transonic airfoil (RAE 2822) with 3.22 degree angle of attack and Mach number of 0.728. Figure 17(c) shows the density contours obtained from 500 time steps on the initial grid, (81 by 41) while the density contours obtained from the same number of time steps on the adaptive grid are in 17(d). The adaptation is to the gradient of the combination density and pressure in both directions ($AWT=GRAD,GRAD$, $RHO=1,1$, $PRES=1,1$) with a C_1 of .75, and a C_2 of .2. The total number of adaptations is 5.

Figure 18 shows the pressure contours on the initial and the adaptive grids. The pressure coefficients of the upper and lower surfaces of this airfoil are plotted

in Figure 19 along with the experimental data. From this figure the adaptive grid gives a better prediction of the location of the shock and a much steeper shock than the initial grid. A record of the CPU time on an IRIS 4D/440VGX machine shows that the total CPU time for the initial grid without adaptation was 1499.93 CPU seconds and for the adaptive grid was 1518.63 CPU seconds, an 1.2 percent increase.

Multiple adaptation for a double wedge

The major goal of multiple adaptation is to move the grid points to capture major features of the flow field as the flow develops. In supersonic flow, these features are shocks and expansions. Flow over a double wedge and into a wind tunnel are considered in the present investigation.

Results obtained from a supersonic flow at a Mach number of 2 over fine (121 by 41) and coarse (81 by 31) double wedge grids are shown in Figures 20 through 32. Figure 20 shows the density contours obtained from 300 time steps on the initial and adaptive grids (121 by 41). The grid was adapted to the density gradient in the flow direction ($RHO = 1,0$) with a C_1 of .7 and to the pressure gradient in the normal direction with a C_2 of .5. A total of four adaptations was used for this case, and the variable $AFIXP = NO$.

Figure 21 shows the pressure contours on the initial and adaptive grids (121 by 41). Figure 22 shows the pressure coefficients of the lower wall and a convergence history of the two solutions are shown in Figure 23. In Figure 23, the high peaks at each adaptation are due to the use of the previous solutions on the new adapted grid. From these figures, clearly the adaptive grid gives a much better representation of the shock regions as well as the expansion regions. Shocks are much sharper for the solution obtained on the adaptive grid. A record of the CPU time on an IRIS 4D/440VGX machine shows that the total CPU time for the initial grid (121 by 41) without adaptation was 1481.51 CPU seconds and for the adaptive grid (121 by 41) was 1599.02 CPU seconds, an 8

percent increase. With this time record, it is worthy to use multiple adaptation to achieve a better representation of the solution. Contour plots of the density for the initial fine grid (121 by 41), initial coarse grid (81 by 31), and adaptive grid (81 by 31) are shown in Figure 24. In this figure, solutions on the right panel were obtained from the corresponding grids on the left panel. The coarse grid was adapted to the combination of density and pressure in ξ^1 direction with weight coefficient C_1 of .5 and to the gradient of this combination in ξ^2 direction with weight coefficient C_2 of .5, (AWT=VAR,GRAD, RHO=1,1, PRES=1,1). The total number of adaptations is four with AFIXP=NO. Figure 25 shows the pressure contours obtained on each grid. Pressure coefficients of the lower walls obtained from the fine and coarse grids are plotted in Figure 26. Figure 27 shows the pressure coefficients of the lower walls of the fine and adaptive coarse grids.

Different adaptive mechanisms were applied to the coarse grid in the multiple adaptation process are shown in Figures 28 through 33. Figure 28(b) shows the density contours obtained on the adaptive grid of Figure 28(a). The initial grid was adapted to the curvature of the combination of density and pressure in both directions (AWT=CURV,CURV, RHO=1,1, PRES=1,1). Total number of adaptations is four with a C_1 of .7 and a C_2 of .7. The coefficients of the gradient and curvature are $\alpha = 1$ and $\beta = .5$, respectively, and AFIXP=YES. Figure 30 shows the pressure coefficient of the lower wall for this case.

Figure 28(d) shows the density contours obtained on the adaptive grid of Figure 28(c). The adaptive mechanism for this case was pressure gradient in both directions with a C_1 of .7, a C_2 of .7, and total number of adaptations is four (AWT=GRAD,GRAD, RHO=0.0, PRES=1,1). The pressure coefficient on the lower wall obtained from this adaptation is shown in Figure 31.

The initial grid, adapted to the gradient of the combination of density and pressure in ξ^2 direction only, is shown in Figure 28(e). Total number of adaptations is five with a C_1 of 0., a C_2 of .9, and AFIXP = NO. Density contours

obtained from this adaptive grid are shown in Figure 28(f), and the pressure coefficient on the lower wall is plotted in Figure 32. Adaptive grids and pressure contours of these adaptive mechanisms are shown in the left and right panel of Figure 29, respectively.

From these figures, the representation of the shocks on the adapted coarse grid is much sharper and closer to the fine grid solution than the nonadaptive coarse grid. The total CPU time for obtaining 300 time steps solution for the adaptive grid was approximately 800 seconds for each adaptive mechanism, nearly a 50 percent time saving compared to that of the fine grid.

The adaptation to the combination of density and pressure in ξ^1 direction and to the gradient of this combination in ξ^2 direction of Figure 24 gives a smoother pressure coefficient behind the shock than the adaptation to the gradient of pressure alone of Figures 28(c) and (d). The adaptation to the curvature of Figures 28(a) and (b) gives a better result, however, with a little over prediction of the pressure coefficient right behind the shock. The adaptation to the gradient of the combination of the density and pressure in ξ^2 direction only of Figures 28(e) and (f) gives the closest solution to the fine grid solution. The convergence history of solutions obtained from this adaptive grid and non-adaptive fine and coarse grids are plotted in Figure 33.

From these results, clearly multiple adaptive grids produce a better representation of the shock regions as well as the expansion regions than that of the same nonadaptive grid. Among these adaptive mechanisms, the use of the weighted average of weight functions computed from several flow variables gives better results than the use of a single variable. Another advantage that should be mentioned here is the controlling of the direction in which adaptation is applied. As shown above, the adaptation in only one direction (ξ^2) gives the closest solution to the fine grid solution. Moreover, the grid in this adaptive mechanism is not disturbed as much as the adaptation in both directions. The minimum skew

angle for this case is higher compared to those of adaptation in both directions. Of course, this is true only for a certain number of adaptations and a particular value of weight coefficients.

Multiple adaptation for a wind tunnel

Results from the supersonic flow at a Mach number of 2 in a wind tunnel are shown in Figures 34 through 35. These results were also obtained in 300 time steps. Figure 34(a) is the initial grid, (c) is the adaptive grid adapted to the error estimation in both directions, and (e) is the adaptive grid adapted to gradient of the combination of density and pressure in both directions. The number of adaptations is five for both cases, with a C_1 of 1 and a C_2 of 1 for adaptation to error estimation, and a C_1 of 0.6 and a C_2 of 0.55 for the adaptation to gradient of the combination. Figure 34(b), (d), and (f) are the pressure contours on initial and adaptive grids. Figure 35 shows the contour plot of the error estimation on each grid. Shocks are much sharper for solutions obtained on the adaptive grids than the nonadaptive grids for this configuration in supersonic flow as well.

Results from these examples show that multiple adaptive grids performed well in capturing major features of the flow field in supersonic flow for these particular configurations. The adaptations to the combination of the grid quality measures such as skewness of the grid and the flow solution for these particular grids not only make the grid more skewed but also resulted in poor resolution of the major features of the flow field. On the other hand, the adaptation to the error estimation and the use of the weighted average in weight functions computed from several flow variables does, in fact, improve the solutions.

The computation of the weight functions and the choice of the adaptive solution variable are independent from one direction to another, which enable the users to have more freedom in choosing suitable adaptive mechanism for each kind of flow. For example, in the case of boundary layers and shocks occurring in the same flow field, the users may choose to adapt the grid to the velocity magnitude

gradient in the normal direction to capture the boundary layer regions and to the pressure gradient in the flow direction to capture the shocks.

Grid refinement

Block grid refinement can be used to improve the quality of numerical solutions. In the following examples a single block grid is divided into subblocks to demonstrate this concept. It should be noted that the subdivision and refinement strategy is done manually for these examples. This could be done automatically for simple configurations like the examples considered here. However, additional research is still needed before the process can be automated for more general problems.

The first example is the solution about an airfoil. The solution is for a Mach number of 0.8 and a 1.25 degree angle of attack. The regions of interest for this problem are the stagnation point at the leading edge, a strong shock on the upper surface, and a weak shock on the lower surface. A solution on a 81 by 41 grid is plotted in Figure 36. The truncation error estimate for the solution vector is plotted in Figure 37. The error estimate is effected in this example by difficulties near the airfoil surface caused by characteristic boundary conditions. The grid is then subblocked and refined in both directions in the region around the shock on the upper surface and near a possible shock on the lower surface. The solution computed on this locally refined grid shows a marked difference in the location and strengths of the shocks, as seen in Figure 38. This solution is compared with the refined single block 161 by 81 solution appearing in Figure 39. The two solutions are nearly identical, indicating that the local block refined solution is as accurate as the fine grid solution. There is no noticeable effect of the interpolation at the interfaces. The coarse and fine grid solutions in Figures 36 and 39 were used to compute and estimate the solution error, using Richardson extrapolation. That estimate is plotted in Figure 40.

The next application of block grid refinement is for flow about a three-dimensional cylindrical store. The solution is computed with a Mach number of 1.2 and a zero angle of attack. The first solution is plotted in Figure 41. The shock structure, which should form near the front and rear of the body, is hardly distinguishable due to the coarseness of the grid. These features can be brought out by refining the grid, as seen in Figure 42. The truncation error computed from the coarse grid solution is also plotted in Figure 43. The largest values occur near the front and rear stagnation points and also along the axis of symmetry where the Jacobian vanishes.

VI. CONCLUSIONS

The widely used EAGLE grid generation system has been extended and enhanced so that it can be readily coupled with existing PDE solvers that operate on structured grids to provide a flexible adaptive grid capability. The adaptive EAGLE grid code now can be used for generating not only algebraic grids and elliptic grids but static adaptive grids as well. In the static adaptation, the grid can be adapted to an existing PDE solution or to grid quality measures or to a combination of both. The test cases show that some grid properties can be improved by the static adaptation to grid quality measures.

In this study, the weight functions can be formulated as weighted average of weight functions from several flow variables or several quality measures or the combination of both. Different weight functions and adaptive variables can be applied in each direction. These operations are controlled through the input parameters in static as well as multiple adaptation mode.

There are several successful incorporations of the adaptive EAGLE subroutines into flow codes, including the MISSE Euler solver. Several configurations are considered for each of these adaptive flow codes for the investigation of the new weight functions formulations and grid quality measures in the multiple adaptation. Results obtained from the adaptive MISSE Euler flow code show considerable success as measured by improvements in shock resolution on coarse grids in the compressible flows.

Several enhancements to the MISSE code itself will aid in the estimation and control of error. These include the capability of block refinement and coarsening and the computation of a truncation error estimate.

REFERENCES

- 1) H. J. Kim and J. F. Thompson, "Three Dimensional Adaptive Grid Generation on a Composite Block Grid," *AIAA 88-0311, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, 1988.*
- 2) J. Tu and J. F. Thompson, "Three Dimensional Solution-Adaptive Grid Generation on Composite Configurations," *AIAA 90-0329, AIAA 28th Aerospace Sciences Meeting, Reno, Nevada, 1990.*
- 3) D. L. Whitfield, "Implicit Upwind Finite Volume Scheme for the 3-D Euler Equations," *Mississippi State University, MSSU-EIRS-ASE-85-1, September 1985.*
- 4) J. U. Brackbill and J. S. Saltzman, "Applications and Generalizations of Variational Method for Generating Adaptive Mesh," *Numerical Grid Generation, J. F. Thompson, Ed., pp. 865-878, North-Holland, 1982.*
- 5) P. J. Roache and S. Steinberg, "A New Approach to Grid Generation Using a Variational Formulation," *AIAA 85-1527, AIAA 7th Computational Fluid Dynamics, Conference, Cincinnati, Ohio. 1985.*
- 6) Z. U. A. Warsi and J. F. Thompson, "Application of Variational Methods in the Fixed and Adaptive Grid Generation," *Computers and Mathematics with Applications, Vol. 19, pp. 31-41, 1990.*
- 7) J. E. Castillo, "A Direct Variational Grid Generation Method: Orthogonal Control," *Numerical Grid Generation in Computational Fluid Mechanics, Sengupta et. al., Ed., pp. 247-256, Pineridge Press, 1988.*

- 8) S. R. Kennon and G. S. Dulikravich, "A Posteriori Optimization of Computational Grids," AIAA 85-0483, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, 1985.
- 9) R. Carcaillet, "Optimization of 3-D Computational Grids and Generation of Flow Adaptive Computational Grids," AIAA 86-0156, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, 1986.
- 10) B. K. Soni and C. W. Mastin, "Variational Methods for Grid Optimization," Submitted for publication, *Journal of Applied Mathematics and Computers*, 1990.
- 11) D. A. Anderson, "Equidistribution Schemes, Poisson Generators, and Adaptive Grids," *Applied Mathematics and Computation*, Vol. 24, pp. 211-227, 1987.
- 12) D. A. Anderson, "Generating Adaptive Grids with Conventional Grid Scheme," AIAA 86-0427, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, 1986.
- 13) B. F. Johnson and J. F. Thompson, "Discussion of a Depth-Dependent Adaptive Grid Generator for Use in Computational Hydraulics," *Numerical Grid Generation in Computational Fluid Mechanics*, pp. 629-640, J. Hauser and C. Taylor, Ed., Pineridge Press, 1986.
- 14) J. F. Thompson, Unpublished Research, Coastal Engineering Research Center, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi, 1986.
- 15) P. R. Eiseman, "Adaptive Grid Generation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, pp. 321-376, 1987.
- 16) J. F. Thompson, "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of PDE," *Applied Numerical Mathematics*, Vol. 1, pp. 3-27, 1985.

- 17) G. D. Kerlick and G. H. Klopfer, "Assessing the Quality of Curvilinear Coordinate Meshes by Decomposing the Jacobian Matrix," *Numerical Grid Generation*, J. F. Thompson, Ed., pp. 787-796, North-Holland, 1982.
- 18) B. Gatlin, et.al., "Extensions to the EAGLE Grid Code for Quality Control and Efficiency," AIAA 90-0148, AIAA 29th Aerospace Sciences Meeting, Reno, Nevada, 1991.
- 19) C. W. Mastin, "Error Estimates and Adaptive Grids for the Numerical Solution of Conservation Laws," *Proceedings of the First International Conference on Computational Physics*, pp.73-76, Boulder, Colorado, June 1990.

APPENDIX A

EAGLE ADAPTIVE COMMANDS

NAME

RESTART generates grid from restart file.

SYNOPSIS

\$ 'RESTART', FILNAM = _____ , VARIN = _____ \$

DESCRIPTION

The function of this command is to read in the restart file from the previous run. This file is unformatted and its name has been set to 'rsfile'.

PARAMETER

FILNAM = 'rsfile'

VARIN = 'NO' indicates that during the previous run the adaptive variables array has not been saved on the restart file.

VARIN = 'YES' indicates that during the previous run the adaptive variables array has been saved on the restart file, and adaptation is to be done on the current run based on this adaptive variables array.

EXAMPLE

\$ 'BLOCK' , SIZE=35,21,1\$

\$ 'BLOCK' , SIZE=80,41,1\$

\$ 'RESTART' , FILNAM = 'rsfile' , VARIN = 'NO' \$

This grid runstream begins with the command 'BLOCK', which indicates that the previous runstream generated a two-block grid. The first block has dimensions $35 \times 21 \times 1$, and the second block has dimension $80 \times 41 \times 1$. The order and the size of each block in a restart runstream must be consistent with that of the previous one. The adaptive variables array is not contained in this 'rsfile', as indicated by VARIN = 'NO'.

See also:

RESTART = 'YES'

Bugs:

Errors may occur if the namelists VAROUT and VARIN are not specified properly.

Notes:

The word RESTART here is used in two cases: the namelist name RESTART and the command RESTART. The same word has two different functions.

NAME

VFILE reads in the flow solution variables file.

SYNOPSIS

\$ 'VFILE', FILNAM = _____ , FORM = _____ \$

DESCRIPTION

The function of this command is to read in the flow solution variables file to perform static adaptation.

PARAMETER

FILNAM is the name of the solution file.

FORM = 'LIST' indicates the form of this file is formatted, and it has the same format as the Q file of PLOT3D.

EXAMPLE

\$ 'VFILE' , FILNAM = 'soln.fmt' , FORM = 'LIST' \$

Bugs:

To avoid errors in this case, this line should always be right before the CUT command, or otherwise before the command END of input.

APPENDIX B

EAGLE ADAPTIVE NAMELISTS

PARAMETER

ITMAXA is the number of adaptive iterations.

PARAMETER

ADAPT = 'NO' or 'NONE' are default values; elliptic grid is produced.

ADAPT = 'YES' produces the adaptive grid.

PARAMETER

AWT = 'VAR', 'VAR', 'VAR' is the adaptation to variable with the weight function as

$$W = 1 + |V|,$$

where V is either a flow variable or a quality measure variable.

AWT = 'GRAD', 'GRAD', 'GRAD' is the adaptation to the gradient of the variable with the weight function as

$$W = 1 + |\nabla V|,$$

where V is either a flow variable or a quality measure variable.

AWT = 'CURV', 'CURV', 'CURV' is the adaptation to the curvature of the variable with the weight function as

$$W = (1 + \beta |K|) \sqrt{1 + \alpha |\nabla V|^2},$$

where

$$K = \frac{\nabla^2 V}{(1 + |\nabla V|^2)^{3/2}}.$$

V is either a flow variable or a quality measure variable.

Notes:

One may specify $AWT = 'VAR', 'GRAD', 'CURV'$, which means the grid is adapted to the variable in the 1 direction, to the gradient in the 2 direction and to the curvature in the 3 direction. Any combination of this is also valid.

PARAMETER

CW: Weight coefficients, default values are 1.0, 1.0, 1.0.

PARAMETER

ALPHA: Coefficient of the gradient, in the range from 0 to 1, default to 1.0.

PARAMETER

BETA: Curvature coefficient, in the range from 0 to 1, default to 1.0.

Notes:

The following set of parameters, ASKEW through VORR, represent the geometric and solution variables for the adaptive process. The default value is 0.0. A value of 1.0 indicates the variable is used in the calculation of the weight function.

PARAMETER

ASKEW: Skew angle.

PARAMETER

AASPE: Aspect ratio.

PARAMETER

AARCL: Arc length.

PARAMETER

APLAC: Laplacian.

Notes:

If all of these namelists are not 0.0, then the weight function is computed as a weighted average of the individual weight functions.

PARAMETER

RHO: Density.

PARAMETER

RHOU: X-momentum.

PARAMETER

RHOV: Y-momentum.

PARAMETER

RHOW: Z-momentum.

PARAMETER

RHOE: Energy.

PARAMETER

VOMA: Velocity magnitude.

PARAMETER

VORR: Vorticity magnitude.

PARAMETER

VARIN: Defaulted value is 'NO', VARIN = 'YES' indicates the restart file 'rsfile' contains adaptive variables array from the previous run.

PARAMETER

VAROUT: Defaulted value is 'NO', VAROUT = 'YES' indicates that in the current run adaptive variables array will be saved on the restart file 'rsfile'.

PARAMETER

RESTART: Defaulted value is 'NO', RESTART = 'YES' means a restart file, namely 'rsfile', will be generated at the end of the current run.

PARAMETER

AFIXP: Defaulted value is 'YES', AFIXP = 'NO' means the control function is updated at every adaptive iteration.

PARAMETER

INTCYL: Defaulted value is 0 and represents the number of time steps at which the adaptation is performed.

PARAMETER

NUMCYL: Defaulted value is 999 and represents the interval of time step between adaptations.

PARAMETER

MAXINT: Defaulted value is 9999 and indicates the number of time step at which the last adaptation is performed.

PARAMETER

QUALITY: Defaulted value is 'NO'.

QUALITY = 'MEASURE' means the grid is being adapted to the quality measure variables alone.

QUALITY = 'YES' means the grid is being adapted to the combination of quality measure variables and flow solution variables.

EXAMPLE 1

This example shows a typical line of the grid runstream for the case where the grid is being adapted statically to one of the quality measure variables.

```
$ 'INITIAL', BLEND=2*'ARC',ALL='YES',CHECK='NO',  
    CONTYP='RADIUS',ITMAX=11,CONFAC=0.1,  
    ITMAXA=5,  
    AWT='VAR','VAR','VAR',  
    CW=1.0,1.0,1.0,  
    ASKEW=1.,  
    AASPE=0.,  
    AARCL=0.,  
    APLAC=0.,  
    AFIXP='NO',  
    ADAPT='YES' $
```

Here the grid is adapted to the skew angle variable, which is indicated by ASKEW=1.0 and AWT='VAR','VAR','VAR'. The weight coefficients are taken to be 1.0 in all directions by CW=1.0,1.0,1.0. The control functions are updated at each adaptive iteration by setting AFIXP='NO', and the total number of adaptations is ITMAXA=5. Adaptation is enabled by ADAPT = 'YES'.

EXAMPLE 2

This example shows some typical lines of the grid runstream for the case where the grid is being adapted statically to one of the flow solution variables.

```
$ 'INITIAL', BLEND=2*'ARC',ALL='YES',CHECK='NO',
      CONTYP='RADIUS',ITMAX=11,CONFAC=0.1,
      ITMAXA=3,
      AWT='VAR','GRAD','CURV',
      CW=0.2,0.5,0.3,
      RHO=1.,0.,0.,
      RHOU=0.,
      RHOV=0.,
      RHOW=0.,
      RHOE=0.,0.,0.,
      AFIXP='YES',
      ADAPT='YES' $
```

C THE CONSTRUCTION OF THE GRID

```
$ 'VFILE' , FILNAM = 'soln.fmt' , FORM = 'LIST' $
$ 'END' $
$ 'ERROR' $
$ 'END' $
```

Here the grid is adapted to the density in the 1 direction, to the gradient of the density in the 2 direction, and the curvature of the density in the 3 direction of the flow solution that is indicated by RHO=1 and AWT='VAR','GRAD','CURV'. The weight coefficients are CW=0.2,0.5,0.3. The control functions are being adapted from the original geometric forms in this case since AFIXP='YES', and the total number of adaptation is ITMAXA=3. Adaptation is enabled by ADAPT = 'YES'.

EXAMPLE 3

This example demonstrates the use of the namelist RESTART to generate a restart file in the adaptive mode.

```
$ 'INITIAL', BLEND=2*'ARC',ALL='YES',CHECK='NO',  
    CONTYP='RADIUS',ITMAX=11,CONFAC=0.1,  
    ITMAXA=3,  
    AWT='VAR','GRAD','CURV',  
    CW=0.2,0.5,0.3,  
    RHO=1.,0.,0.,  
    RHOU=0.,  
    RHOV=0.,  
    RHOW=0.,  
    RHOE=0.,0.,0.,  
    AFIXP='YES',  
    ADAPT='YES',  
    RESTART='YES',  
    VAROUT='YES' $
```

Here VAROUT = 'YES' means the adaptive variable has been saved in the restart file.

EXAMPLE 4

This example shows some typical lines of the grid runstream corresponding to that of Example 3 through the use of the RESTART command in the adaptive mode.

```
$ 'INITIAL', BLEND=2*'ARC',ALL='YES',CHECK='NO',
      CONTYP='RADIUS',ITMAX=11,CONFAC=0.1,
      ITMAXA=3,
      AWT='VAR','GRAD','CURV',
      CW=0.2,0.5,0.3,
      RHO=1.,0.,0.,
      RHOU=0.,
      RHOV=0.,
      RHOW=0.,
      RHOE=0.,0.,0.,
      AFIXP='YES',
      ADAPT='YES',
$ 'BLOCK' , SIZE=35,21,1$
$ 'BLOCK' , SIZE=80,41,1$
$ 'RESTART' , FILNAM = 'rsfile' , VARIN = 'YES' $
```

Here VARIN = 'YES' indicates the adaptive variable has been read in for use in the adaptation.

EXAMPLE 5

This example shows some typical lines of the grid runstream corresponding to the use of the weighted average in the adaptation to the flow variables.

```
$ 'INITIAL', BLEND=2*'ARC',ALL='YES',CHECK='NO',  
    CONTYP='RADIUS',ITMAX=11,CONFAC=0.1,  
    ITMAXA=3,  
    AWT='VAR','GRAD','CURV',  
    CW=0.2,0.5,0.3,  
    RHO=1.,1.,1.,  
    RHOU=0.,  
    RHOV=0.,  
    RHOW=0.,  
    RHOE=1.,1.,1.,  
    PRES=1.,1.,1.,  
    AFIXP='YES',  
    ADAPT='YES',
```

Here the specification of $RHO = 1,1,1$, $RHOE = 1,1,1$, and $PRES = 1,1,1$ means the adaptive variable is computed as the combination of density, energy, and pressure in each direction. Since $AWT = 'VAR', 'GRAD', 'CURV'$, the grid is being adapted to this combination in the 1 direction, to the gradient of this combination in the 2 direction, and to the curvature of this combination in the 3 direction.

APPENDIX C

INCORPORATION INTO FLOW CODES

The procedure for coupling of the Adaptive EAGLE grid code into any CFD flow code can be described as follows:

- 1) First, define all variables involved in the adaptive grid generation such as AWT, RESIN, QUALITY, AFIXP, FACTOR, CW, RHO, RHOU, RHOV, RHOW, RHOE, PRES, VOMA, ITMAX, ITMAXA, INTCYL, NUMCYL, KFILE, ALPHA, BETA, ASKEW, AASPE, AARCL, APLAC, MAXINT, ISTAT, etc. at the declaration setion of the main flow code. Definition of these parameters can be found in Appendix B.
- 2) Set up a namelist to read in all of those variables defined in 1.
- 3) Set up two "if" statements inside the time step loop of the main flow code, one to check at what time step the last adaptation is performed and the other to check how often the adaptation is performed.

Inside these "if" statements

- 4) Call subroutine SETVAR.
- 5) Call subroutine ELLGEN.
- 6) Call subroutine RAGRID.

The functions of these subroutines are described below:

ROUTINE	CALLED BY	FUNCTION
SETVAR	Main flow code	Computes the adaptive variable based on the current flow solution. After computing the adaptive variable this subroutine then writes out these values into scratch file KFILE.
WRTVAR	SETVAR	Writes out the adaptive variable into scratch file KFILE.

RAGRID	Main flow code	Reads in the adaptive variable from scratch file KFILE.
ELLGEN	Main flow code	Generates a new adapted grid based on flow variables in the Dynamic Adaptation process.

There are several subroutines that are called by ELLGEN and their functions are described below:

ROUTINE	CALLED BY	FUNCTION
CCDRA & CCDWA	SSD	When the namelist KSTORE = 'CORE' & ADAPT = 'YES', these subroutines are called and have the same function as CCDR & CCDW in the original case.
JACBCK	ELLGEN	Checks for a twisted grid.
QUAL2D & QUAL3D	ELLGEN	Computes the quality measures of the grid.
REDVAR	ELLGEN	Reads in the adaptive variable that has been written out on scratch file KFILE for use in the adaptive process.
SSDRA & SSDWA	SSD	When the namelist KSTORE = 'CORE' & ADAPT = 'YES', these subroutines are called and have the same function as SSDR & SSDW in the original case.
REDVAR	ELLGEN	Called when the static adaptation to the available flow variables is performed. It reads in the flow variables file.
REDRES	ELLGEN	Reads in the restart file 'rsfile'.

SETIMP	ELLGEN	Sets control function values on boundaries.
SETIMR	ELLGEN	Sets coordinates values at Neumann, image, and reflective points.
SETIMV	ELLGEN	Sets the adaptive variable values at image points equal to the current values at the corresponding object points, sets weight values at Neumann and reflective boundaries and special points, and also extrapolates to other boundary points.
SETIMW	ELLGEN	Has the same function as SETIMV but with the weight values instead.
SMOOTHW	ELLGEN	After the weight values are set on the field and boundaries, this subroutine smooths these weight values.
STOREP	ELLGEN	When KSTORE = 'FILE' & ADAPT = 'YES', it is called to store the original control function values in array for later use.
STOREPW	ELLGEN	This subroutine has the same function as STOREP for KSTORE = 'CORE' & ADAPT = 'YES' .
WEIT2D & WEIT3D	ELLGEN	Computes the weight function values inside the field.
WEITCON	ELLGEN	Performs the linear combination of the original control function values that have been saved by STOREP or STOREPW and the weight function values.
WAGRID	ELLGEN	Writes out the new grid into scratch file KFILE.

EXAMPLE

A simple FORTRAN code below demonstrates the incorporation of the Adaptive EAGLE grid into CFD flow codes.

Begin of Adaptive Section #1

(DECLARATION OF ALL PARAMETERS)

End of Adaptive Section #1

DO 10 NTIME=1,NTMAX (The time step loop of main flow code)

Begin of Adaptive Section #2

IF(INTCYL .LE. MAXINT) THEN (Check for the last adaptation)

IF(NTIME .EQ. INTCYL) THEN (Check for the first adaptation)

SETVAR subroutine computes the adaptive variable based on the current flow solutions.

IF(QUALITY .NE. 'MEASURE') CALL SETVAR

ELLGEN subroutine performs the adaptive grid that bases on the old grid and the variables of this INTCYL time step.

CALL ELLGEN

RESIN = 'NO'

RAGRID subroutine reads in the new grid that comes out from ELLGEN.

CALL RAGRID

INTCYL = INTCYL + NUMCYL

IF(INTCYL .GT. NTMAX) INTCYL = 0

END IF

END IF

End of Adaptive Section #2

10 CONTINUE

APPENDIX D

FLOW CODE ENHANCEMENTS

An option has been included that allows the user to select a refinement or coarsening of the grid in any of the three coordinate directions in any block. The grid selection is made by the value assigned to the character array REGRID in the namelist FINPUT. The following choices exist for the array elements.

PARAMETER

REGRID(IC,IB) = 'REFINE'. The grid in block number IB is to be refined in coordinate direction IC.

REGRID(IC,IB) = 'COARSE'. The grid in block number IB is to be coarsened in coordinate direction IC.

REGRID(IC,IB) = 'NO'. Input grid dimensions are used in coordinate direction IC of block number IB. (Default)

For example, REGRID(2,3) = 'REFINE' would instruct the program to refine block number 3 in the coordinate direction 2; that is, refine in the J direction if the coordinate directions are denoted I, J, and K.

The user has the option of creating a restart file after the final cycle of the flow solver has been completed. When a restart file is to be written by the flow code, the solution may be either output on the actual computational grid as it exists after any grid refinement (or coarsening) or interpolated back onto the original grid that was read into the flow code. This choice is determined by the value of the character variable DUMP in namelist FINPUT as follows.

PARAMETER

DUMP = 'NEWGRID'. The solution is output with the grid that was used in the computations.

DUMP = 'OLDGRID'. The solution is output with the grid that was read into the flow code. (Default)

DUMP = 'NO' means no restart file is created.

Selected surfaces or blocks of grid points and solution values can be written out in plot3d formatted files. The character variable PLOT3D in the namelist FINPUT is used to indicate whether plot3d data is to be read by the program.

PARAMETER

PLOT3D = 'YES'. PLOT3D data is included in the input file.

PLOT3D = 'NO' means there is no PLOT3D data in the input file. (Default)

The block number and the array limits for each block of plot3d grid and solution is input into variables in the namelist PLOT3D. The input is terminated by setting the block number to zero.

EXAMPLE

\$ 'PLOT3D', BLKA = 1, STARTA = 1,1,1, ENDA = 41,41,1 \$

\$ 'PLOT3D', BLKA = 1, STARTA = 41,41,1, ENDA = 81,41,1 \$

\$ 'PLOT3D', BLKA = 0 \$

This data would create a two-block plot3d data set from a single surface grid in the MISSE program.

The program can compute a truncation error estimate and print out the values of the estimate by block using various norms. The estimate can also be output in the form of a plot3d solution file. The character variable ERROR in the namelist FINPUT has the following options.

PARAMETER

ERROR = 'YES'. A truncation error estimate is computed and included in the output.

ERROR = 'NO'. A truncation error is not computed. (Default)

File names can now be included in the input file for the MISSE code. This option is indicated in the value of the character variable FNAMES in the namelist FINPUT. The file names are input through the namelist FILES.

PARAMETER

FNAMES = 'YES' means the namelist FILES has values read from the input file.

FNAMES = 'NO' means the default file names are used. (Default)

The actual default values in the MISSE code are used in the following example of input for the namelist FILES.

EXAMPLE

```
$ 'FILES', GRID = 'MISSE.GRD', REST = 'MISSE.RES', P3DGRD =  
'PLOT3D.GRD', P3DSOL = 'PLOT3D.SOL', P3DERR = 'PLOT3D.ERR' $
```

The file GRID is the list formatted grid file from the EAGLE grid code. It may (OUTER = 'YES') or may not (OUTER = 'NO') contain a surrounding layer of points for each block. The file REST is the restart file. The remaining files are plot3d formatted files that are used to display the solution and/or error estimates in various regions. They are only used if the PLOT3D = 'YES' option is exercised.

A program called COMP has been written to compare numerical solutions. While it is not a part of the MISSE code, it is intended to be used with MISSE restart files. The program was written to compare coarse and fine grid solutions, but in fact can be used to compare any two solutions defined on or interpolated onto the same grid. The input consists of two MISSE input files and two MISSE restart files for two separate runs of the flow code. Along with the input from the two flow codes, there are two namelist input lists, CINPUT and CPLOT, which inputs information about files and plotting. A sample namelist input file is as follows:

```
$ 'CINPUT', P3D = 'YES', REST1 = 'MISSE1.RES', REST2 = 'MISSE2.RES',  
P3DGRD = 'COMP.GRD', P3DERR = 'COMP.ERR', DATA1 = 'MISSE1.DAT',  
DATA2 = 'MISSE2.DAT' $  
$ 'CPLOT', BLKA = 1, STARTA = 1,1,1, ENDA = 81,41,1 $  
$ 'CPLOT', BLKA = 0 $
```

The input variables are similar to the MISSE input and interpreted in the obvious way. For example P3D = 'YES' indicates that plot3d output is written and the other variables in CINPUT are the data and restart files from the MISSE solutions. The variables in the namelist CPLOT are the same as were used in the namelist PLOT in the MISSE code. The output consists of a listing of the parameter values used in computing the two solutions and an indication of when there is a difference in any parameter in the two runs of the MISSE code. As with the flow solver, there is the option for plot3d formatted output of the difference in certain flow variables and the complete solution vector.

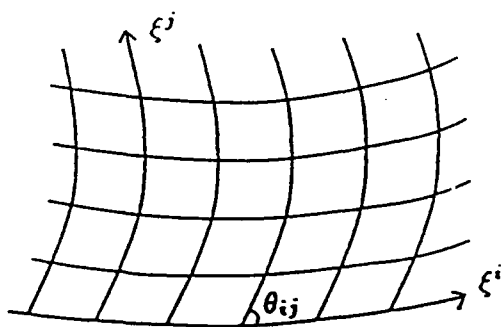


Figure 1. Skew angle .

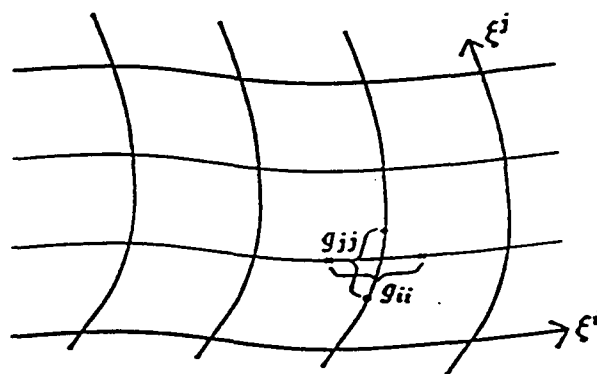


Figure 2. Aspect ratio .

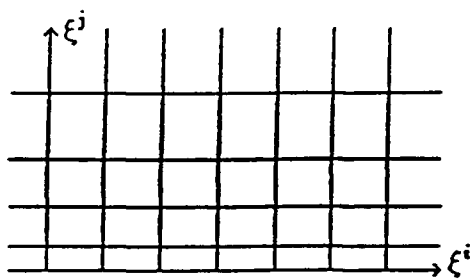
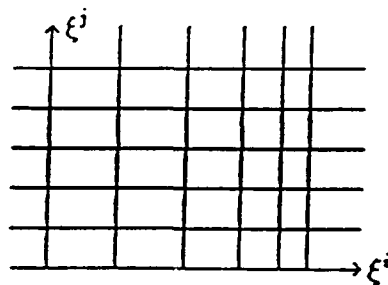
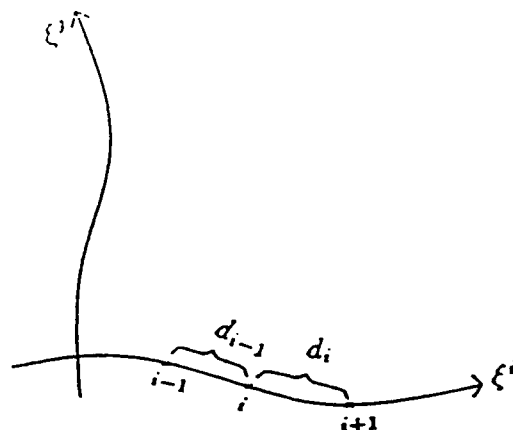
Figure 3.1. Laplacian with
 $\nabla^2 \xi^i = 0, \nabla^2 \xi^j > 0$.Figure 3.2. Laplacian with
 $\nabla^2 \xi^j = 0, \nabla^2 \xi^i < 0$.

Figure 4. Arc length .

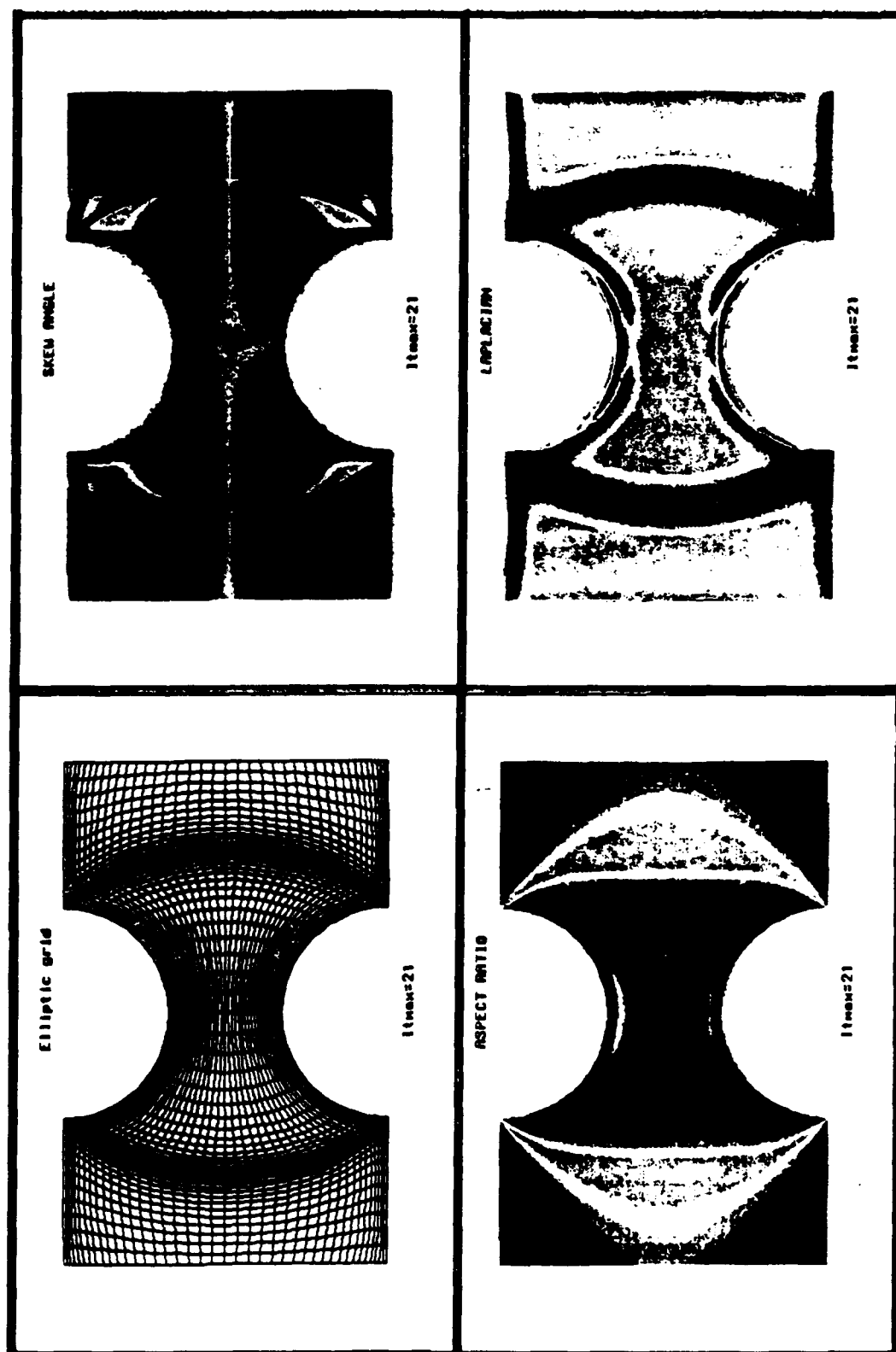


Figure 5. Contour plots of skew angle, aspect ratio, Laplacian of the initial grid.

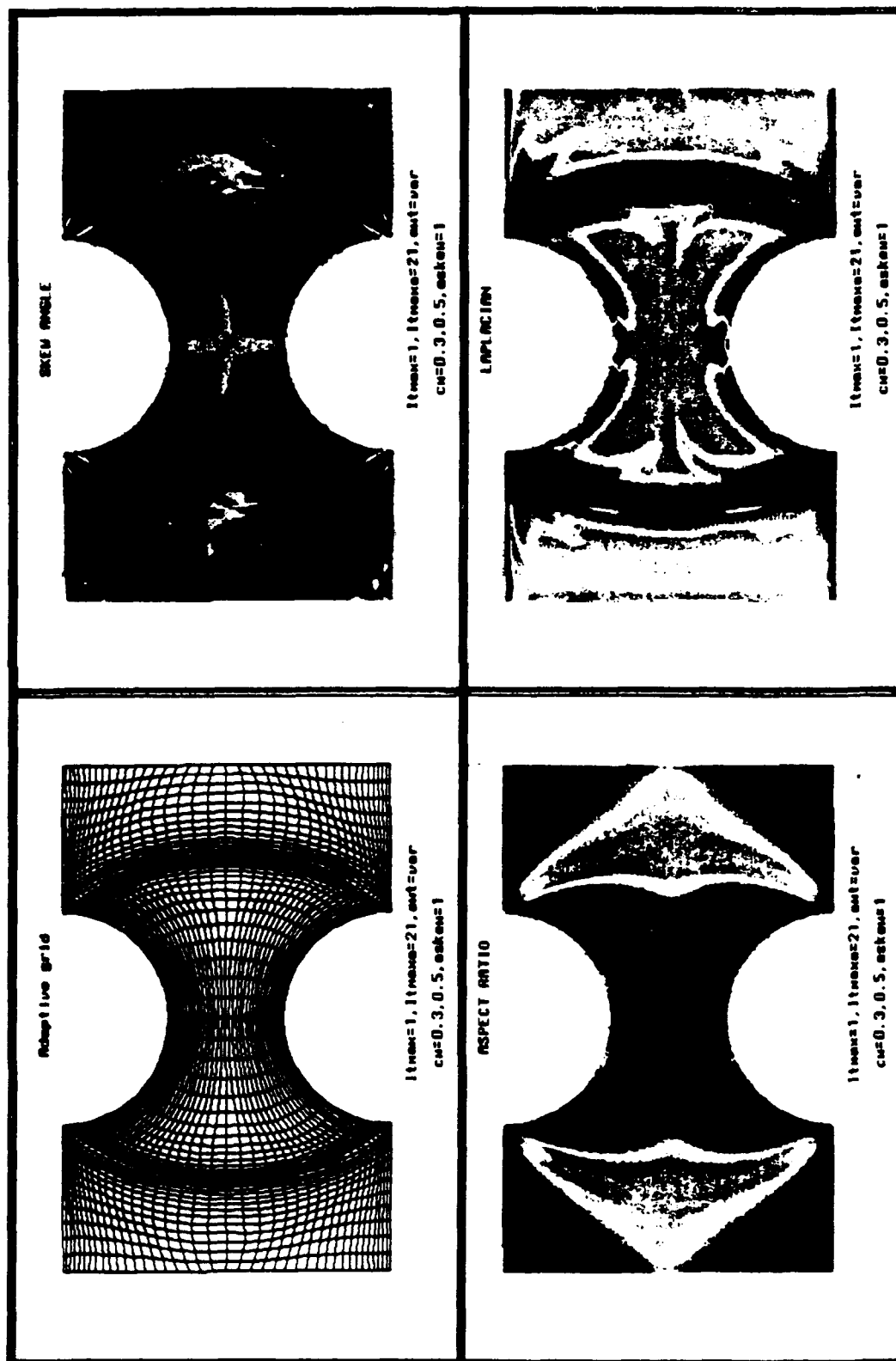


Figure 6. Contour plots of the adaptive grid, adaptation to skewness.

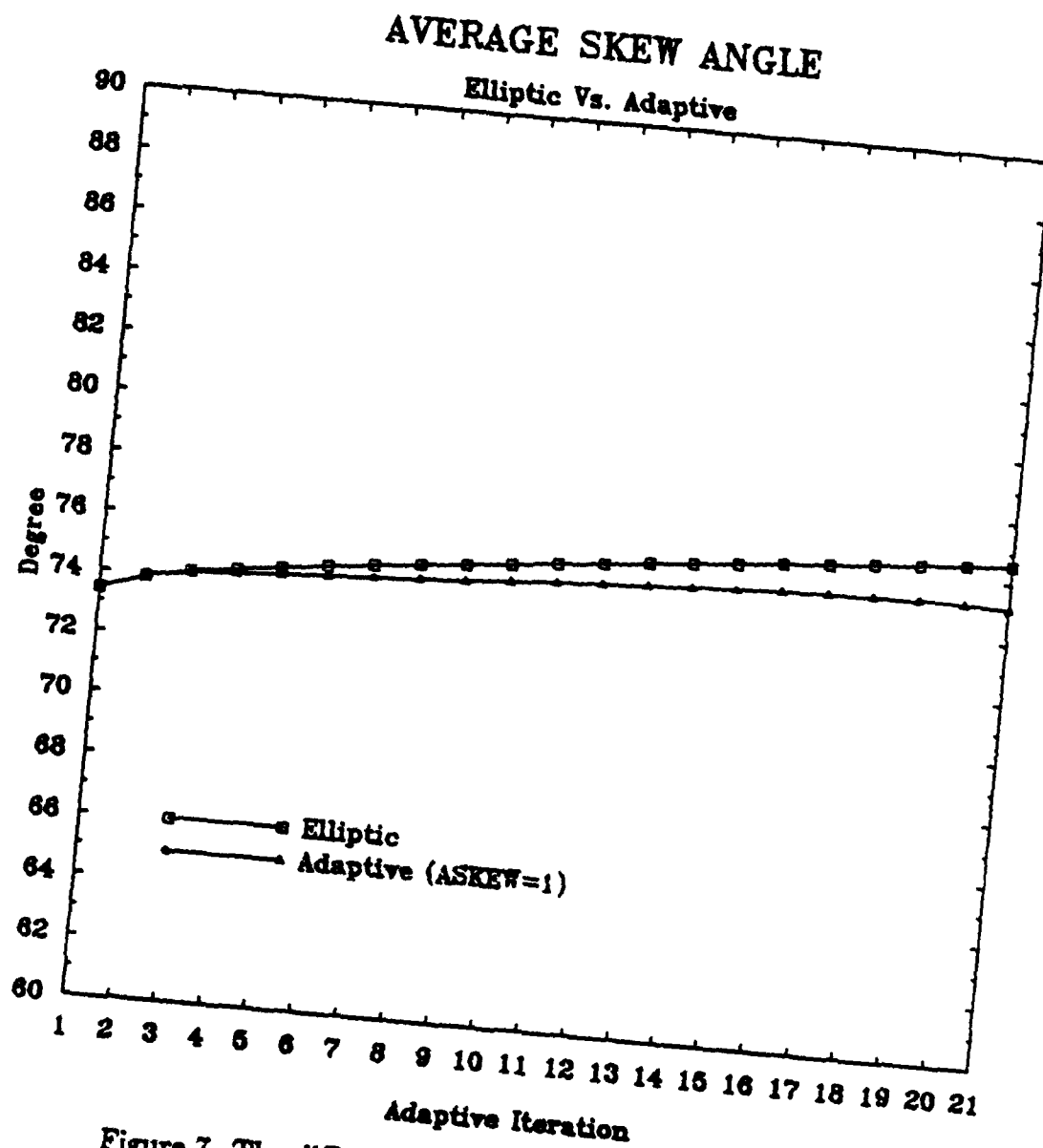


Figure 7. The difference of average skew angle between initial and adaptive grids.

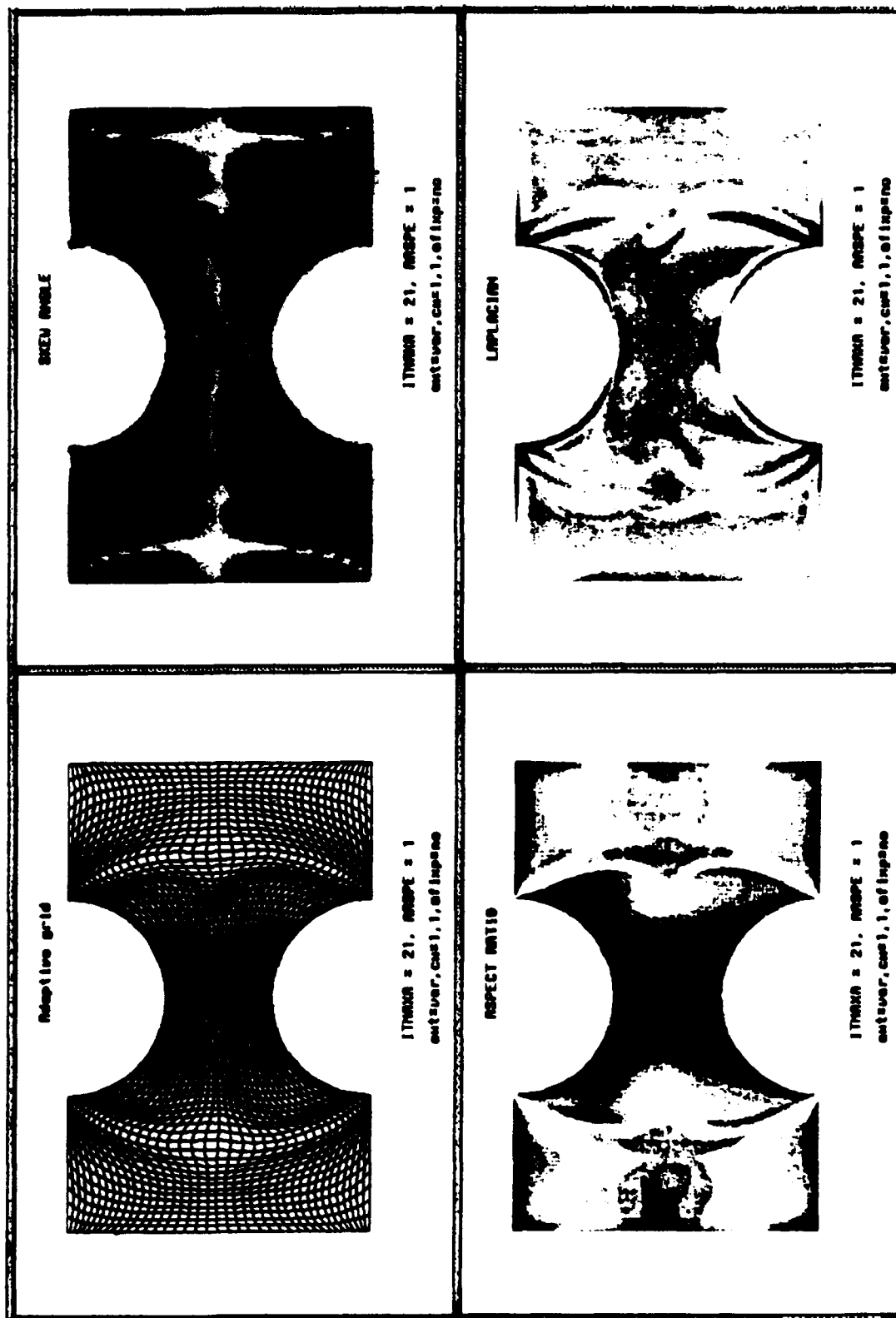


Figure 8. Contour plots of the adaptive grid, adaptation to aspect ratio.

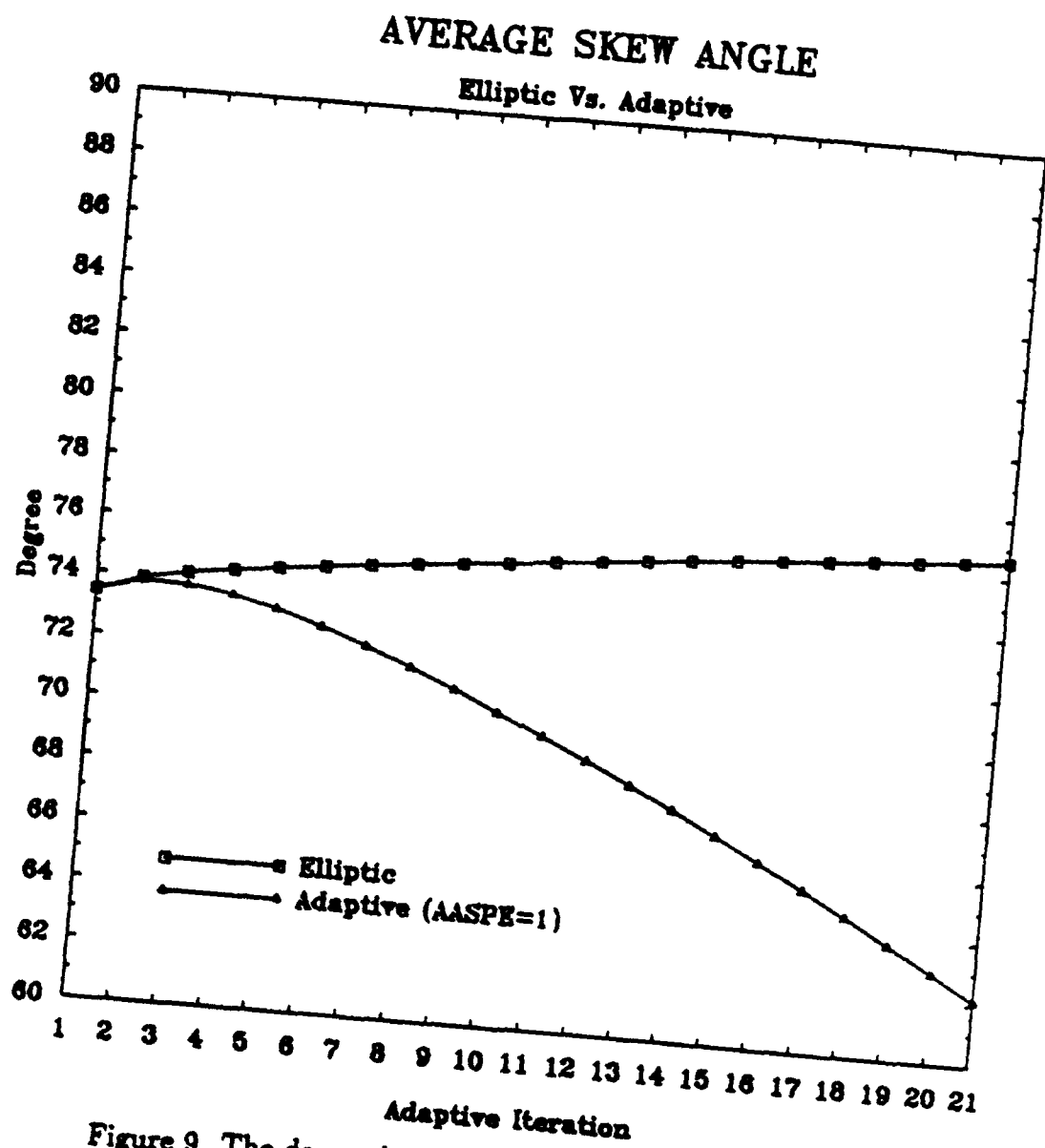


Figure 9. The decreasing of average skew angle in the adaptation to aspect ratio.

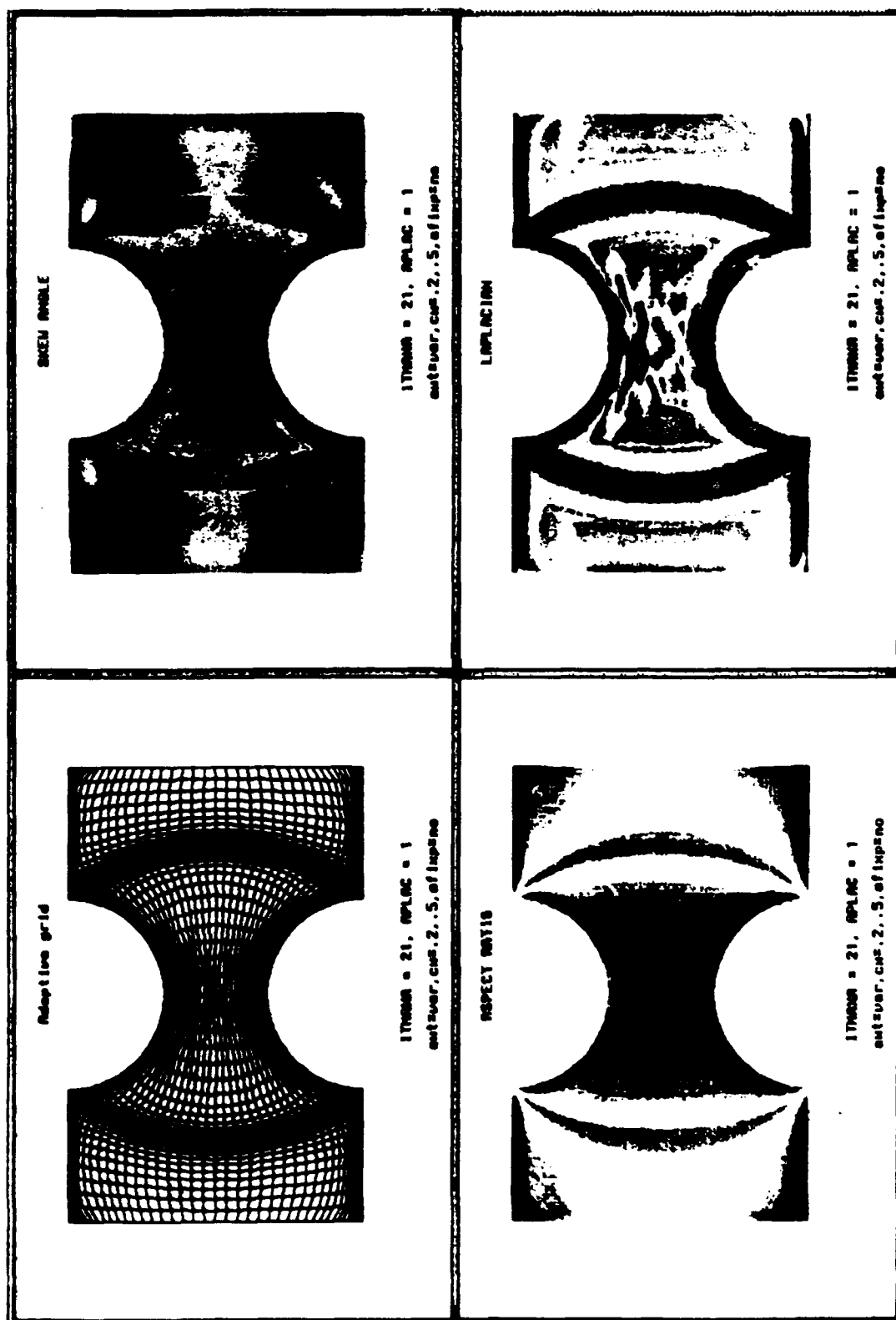


Figure 10. Contour plots of the adaptive grid, adaptation to Laplacian of the grid.

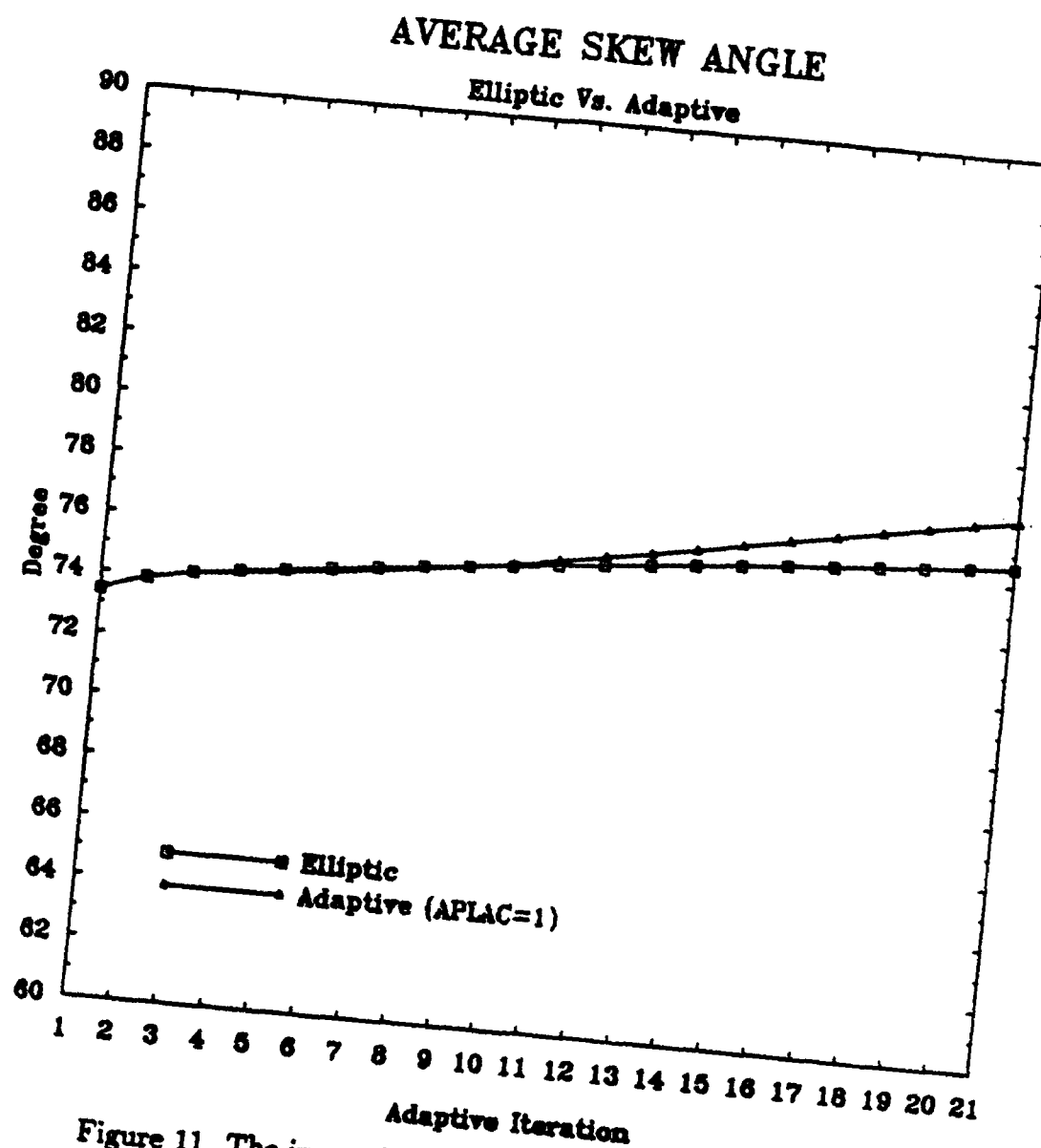


Figure 11. The increasing of average skew angle in the adaptation to Laplacian of the grid.

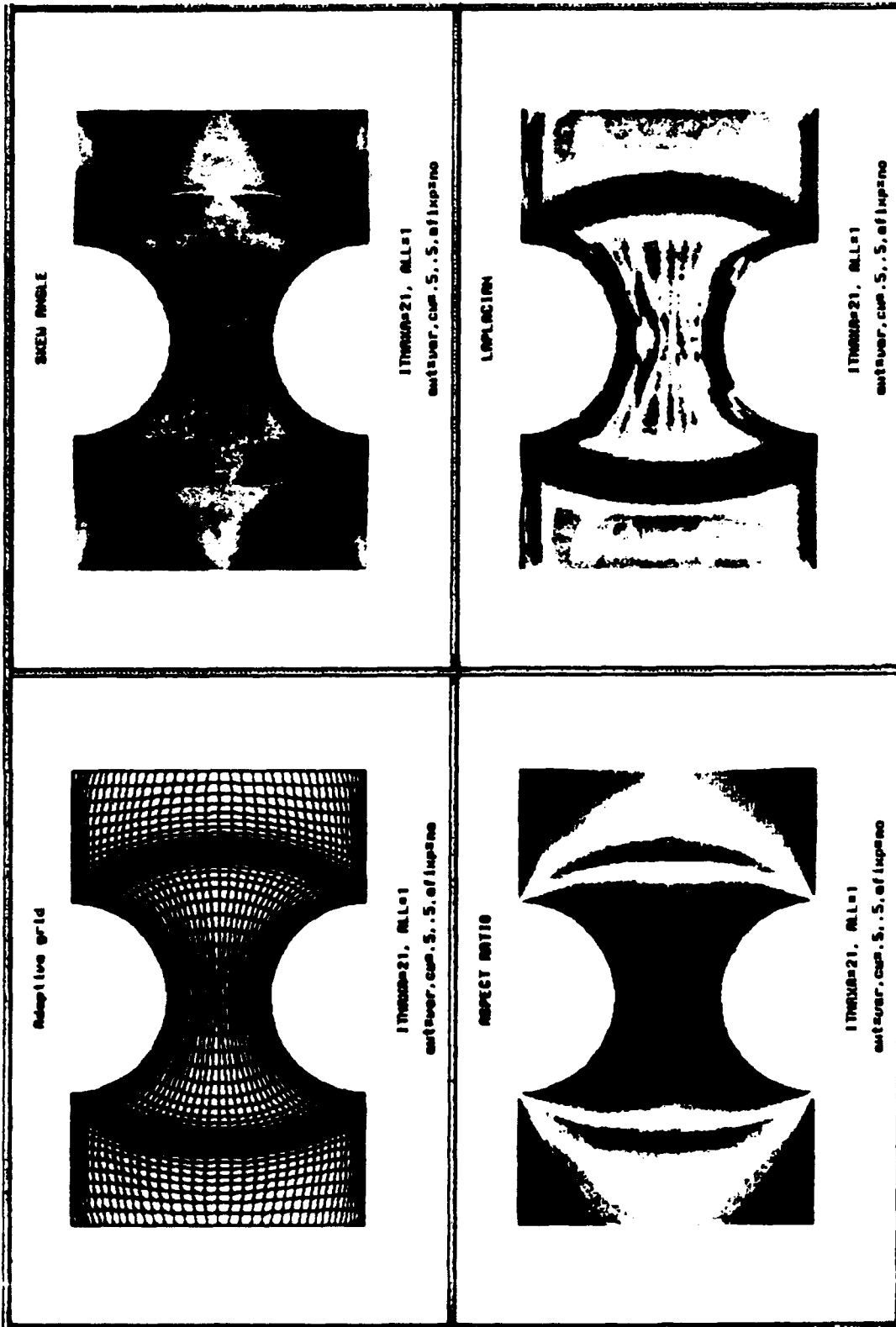


Figure 12. Contour plots of the adaptive grid, adaptation to skewness, aspect ratio, arc length, and Laplacian of the grid.

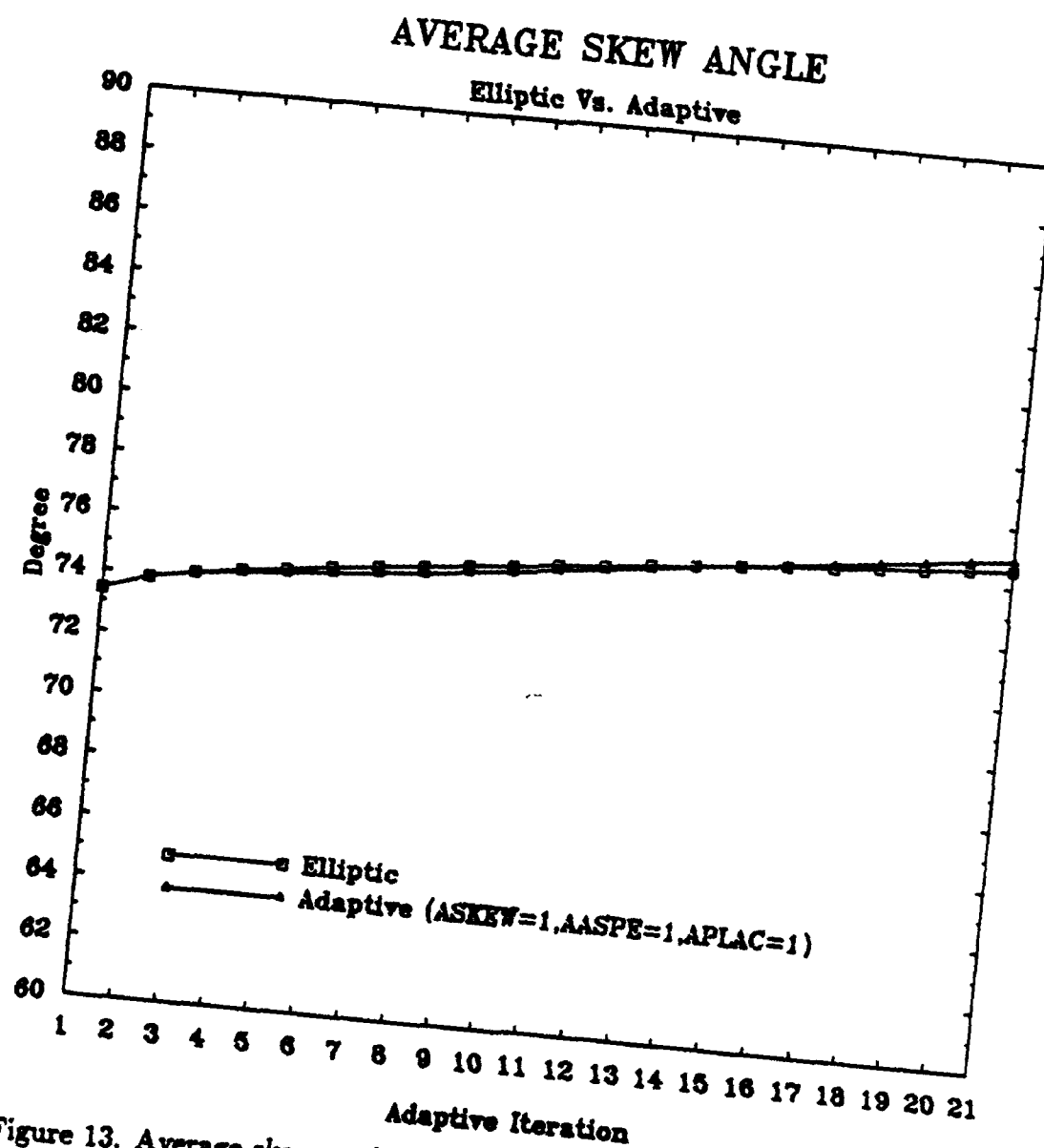
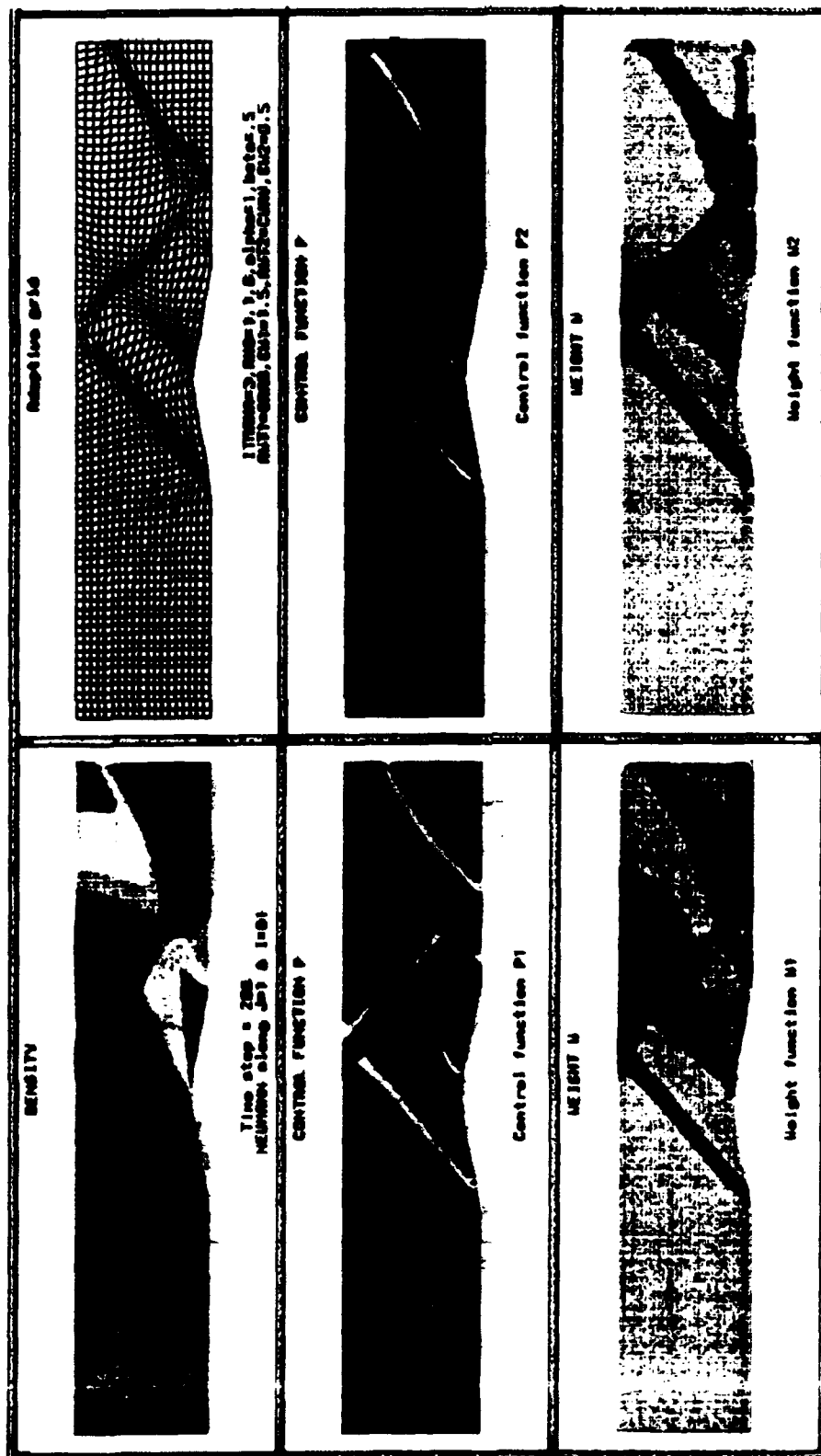
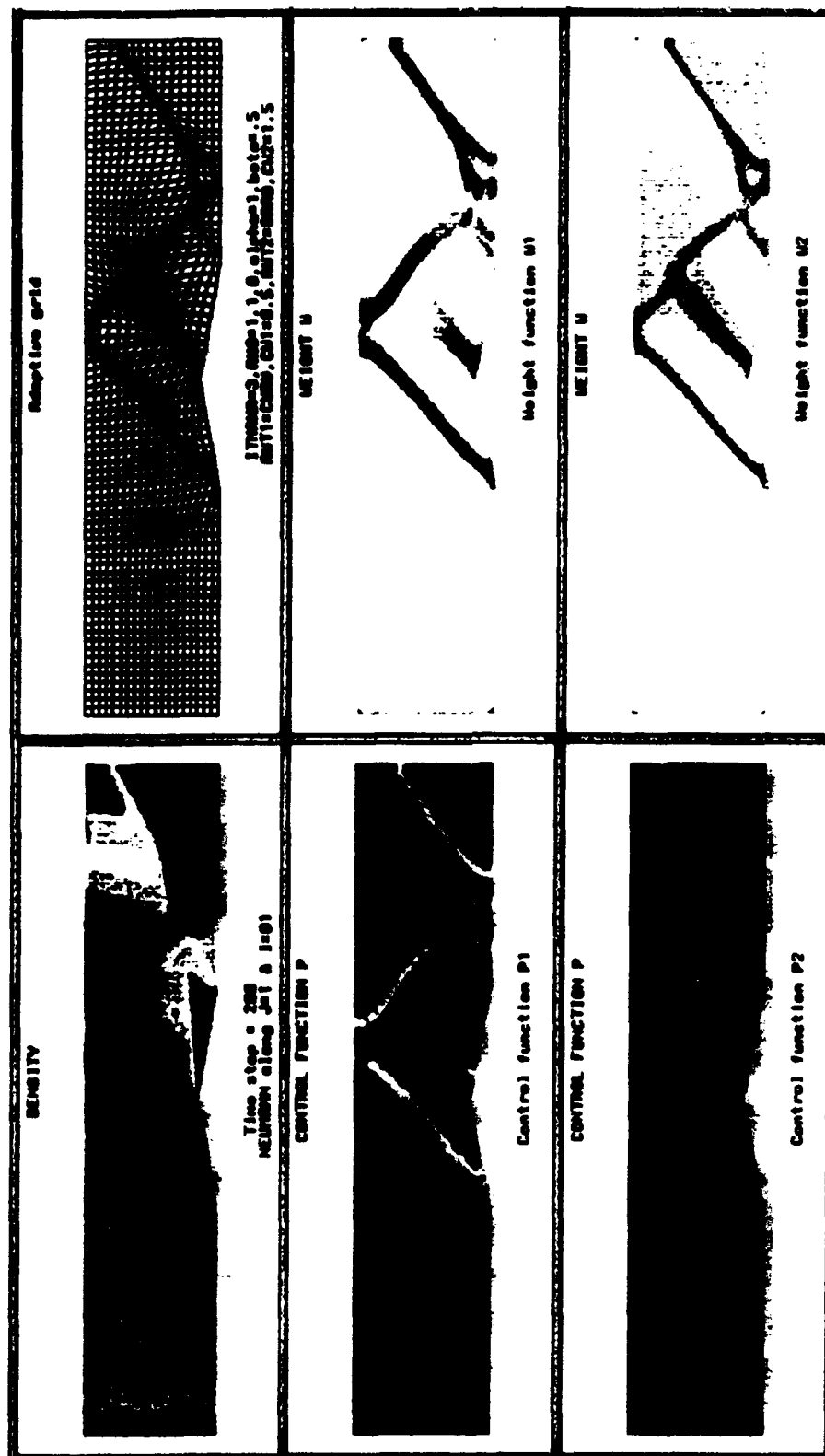


Figure 13. Average skew angle in the adaptation to all grid quality measures.





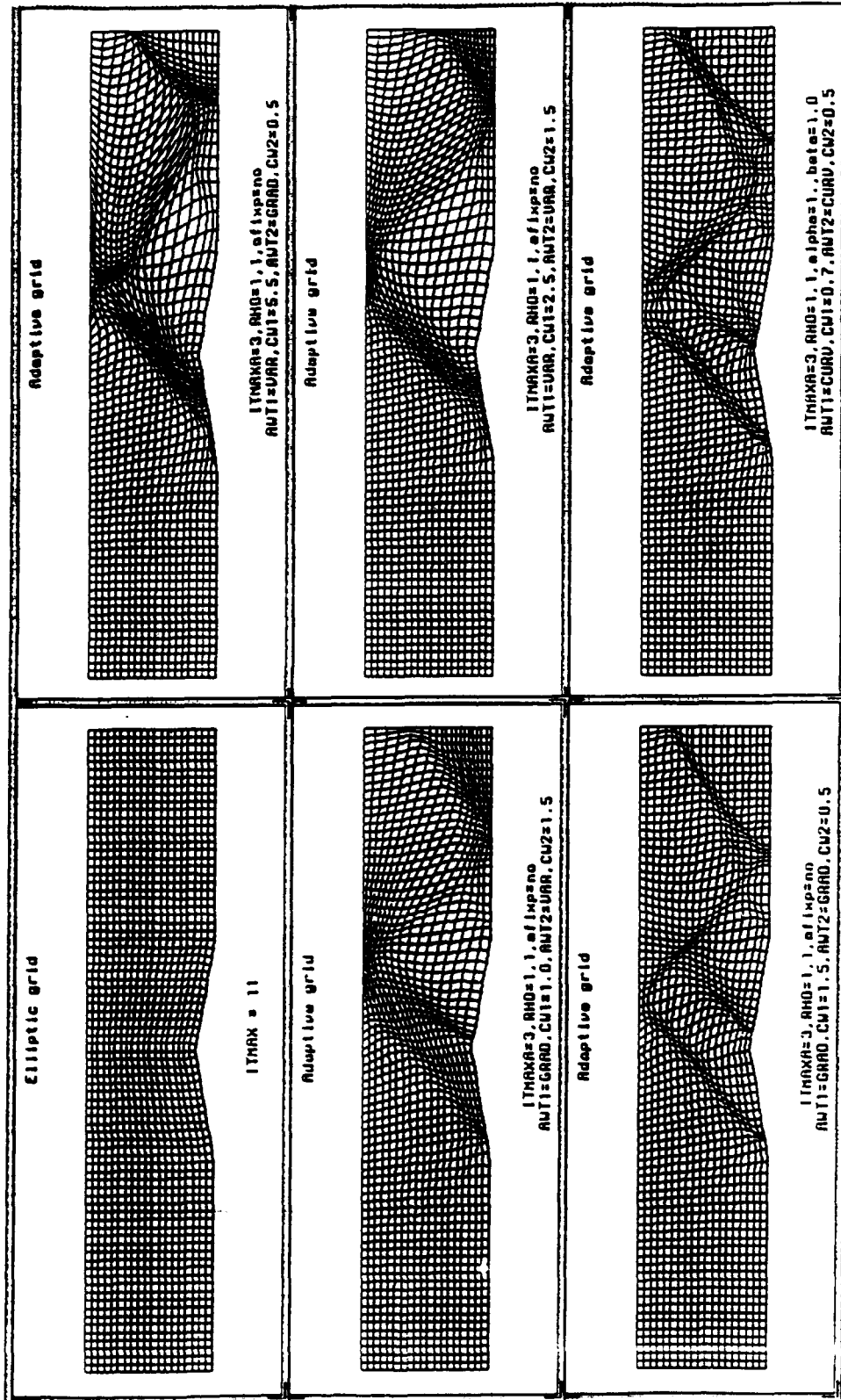


Figure 16. Initial grid and adaptive grids for different parameters and adaptive mechanisms.

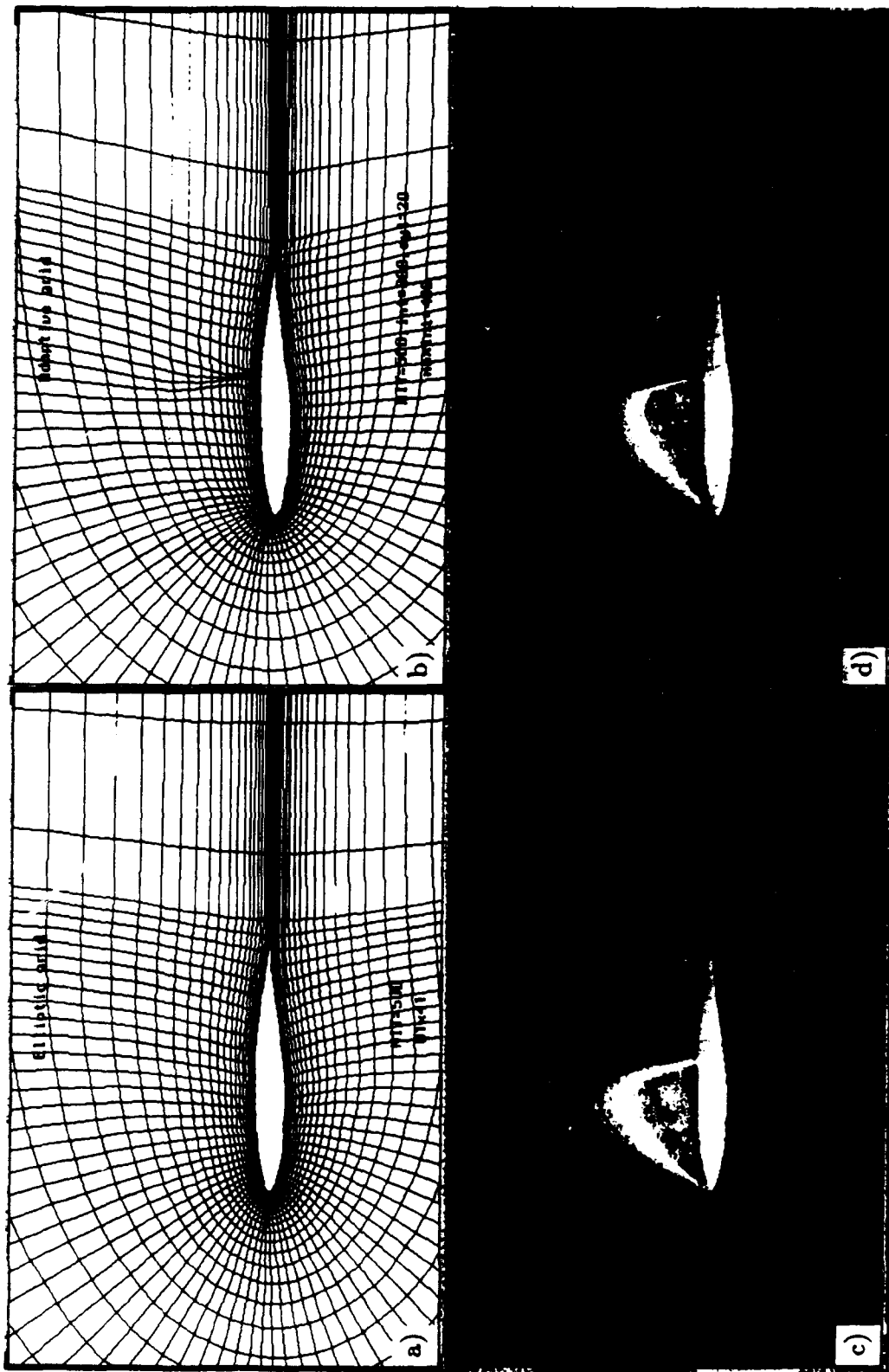


Figure 17. Contour plots of density on initial and adaptive grids.

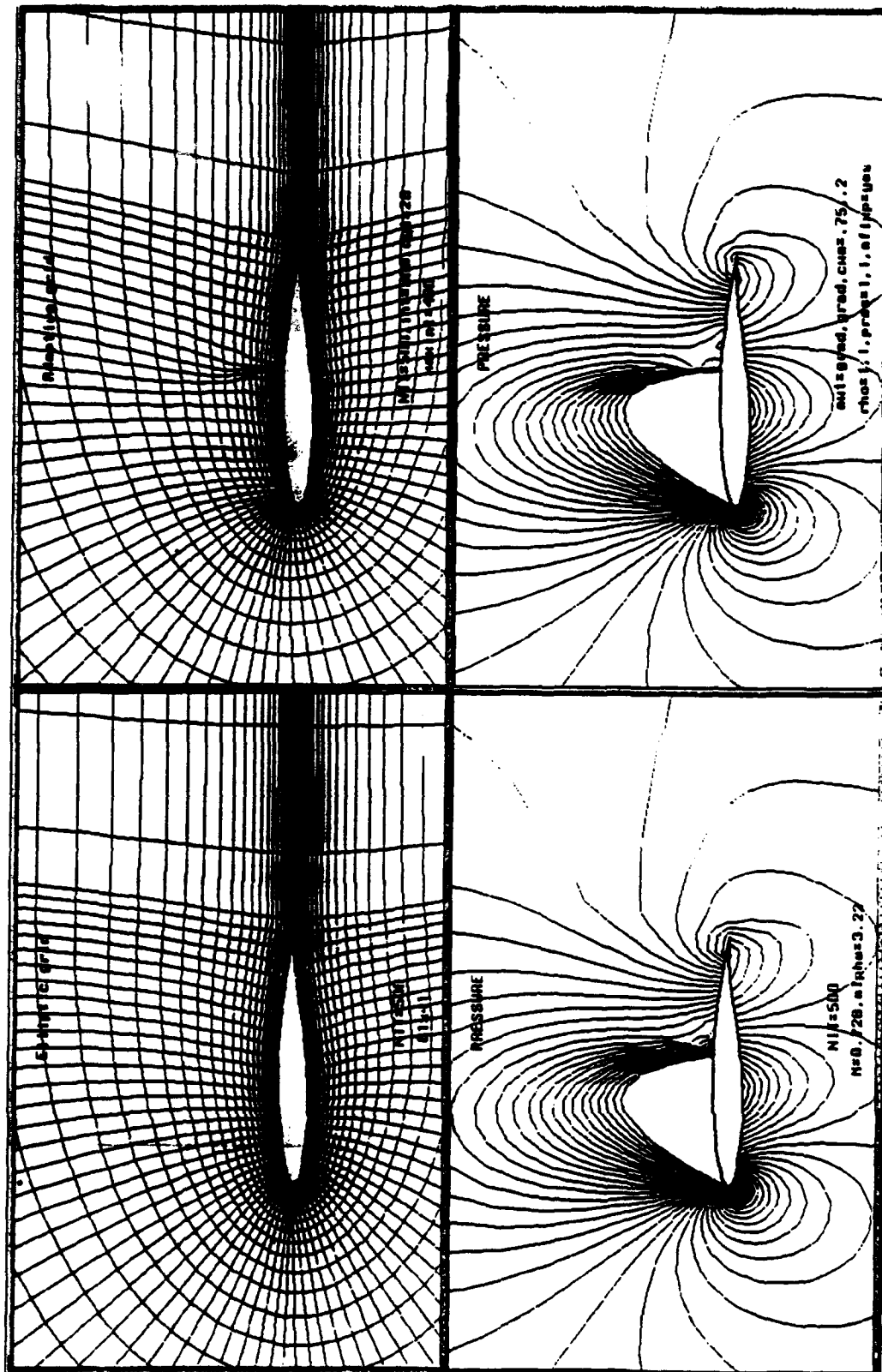


Figure 18. Contour plots of pressure on initial and adaptive grids.

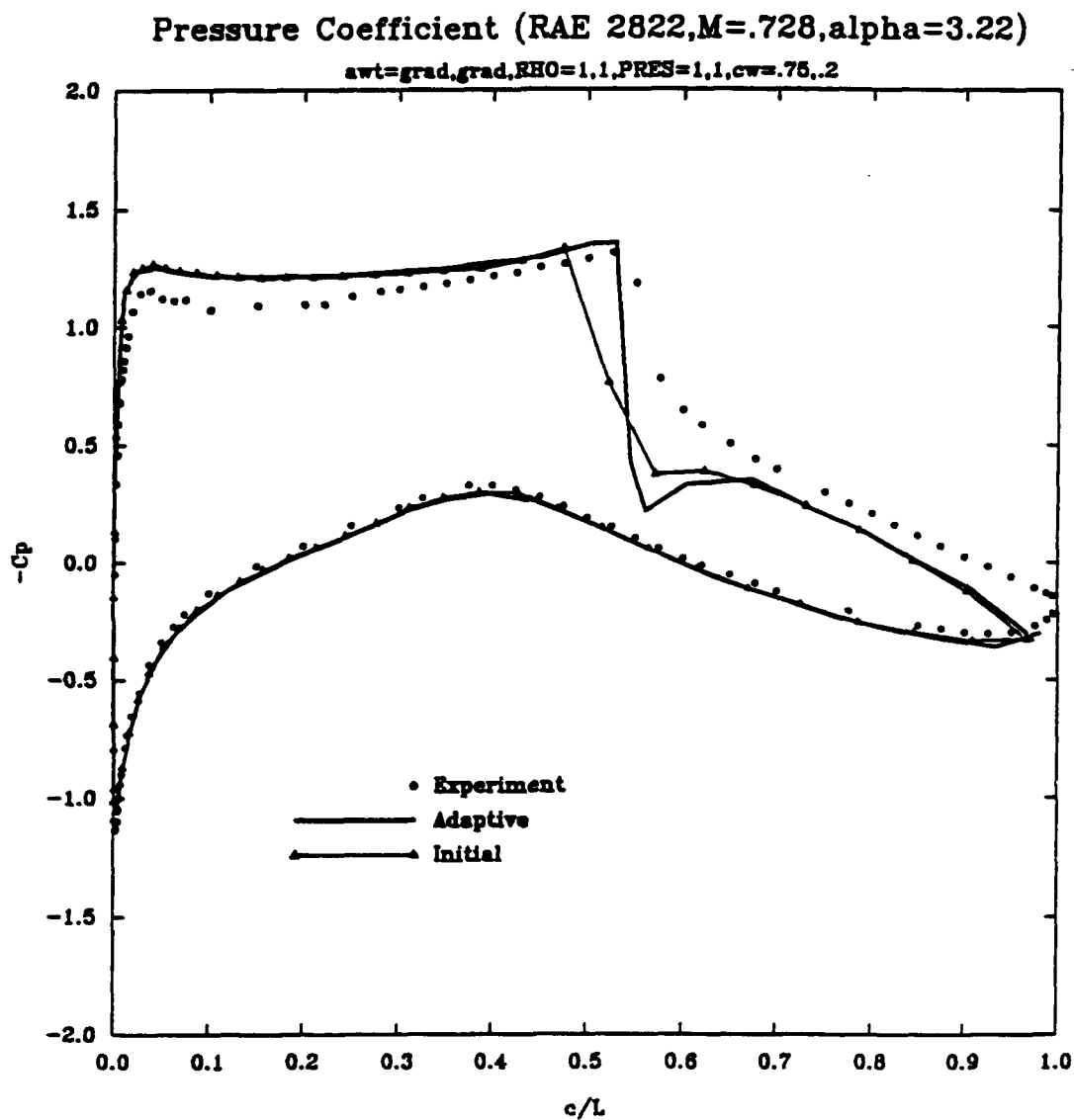


Figure 19. Pressure coefficients of the lower and upper surfaces.

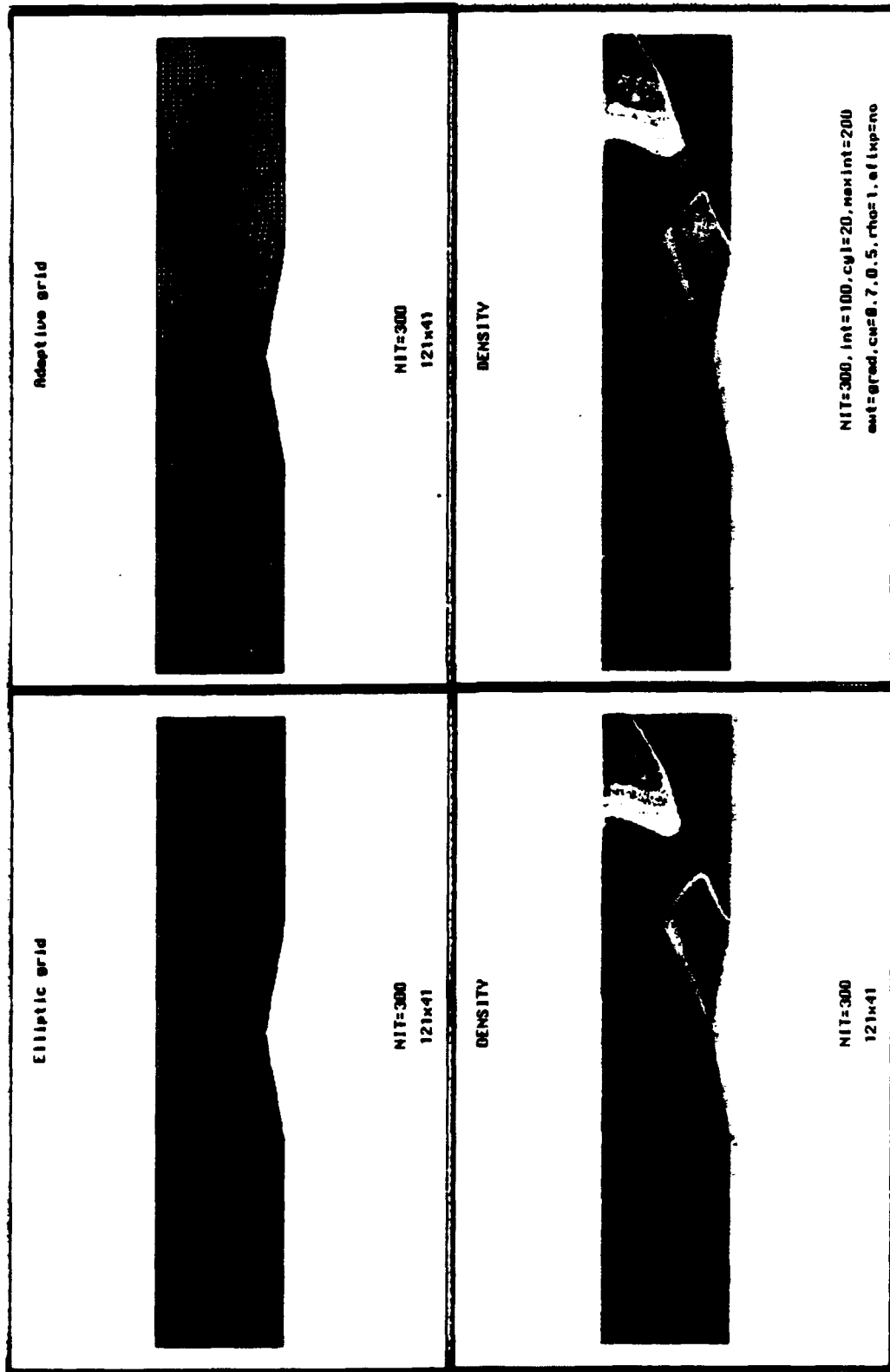


Figure 20. Contour plots of density on initial and adaptive grids.

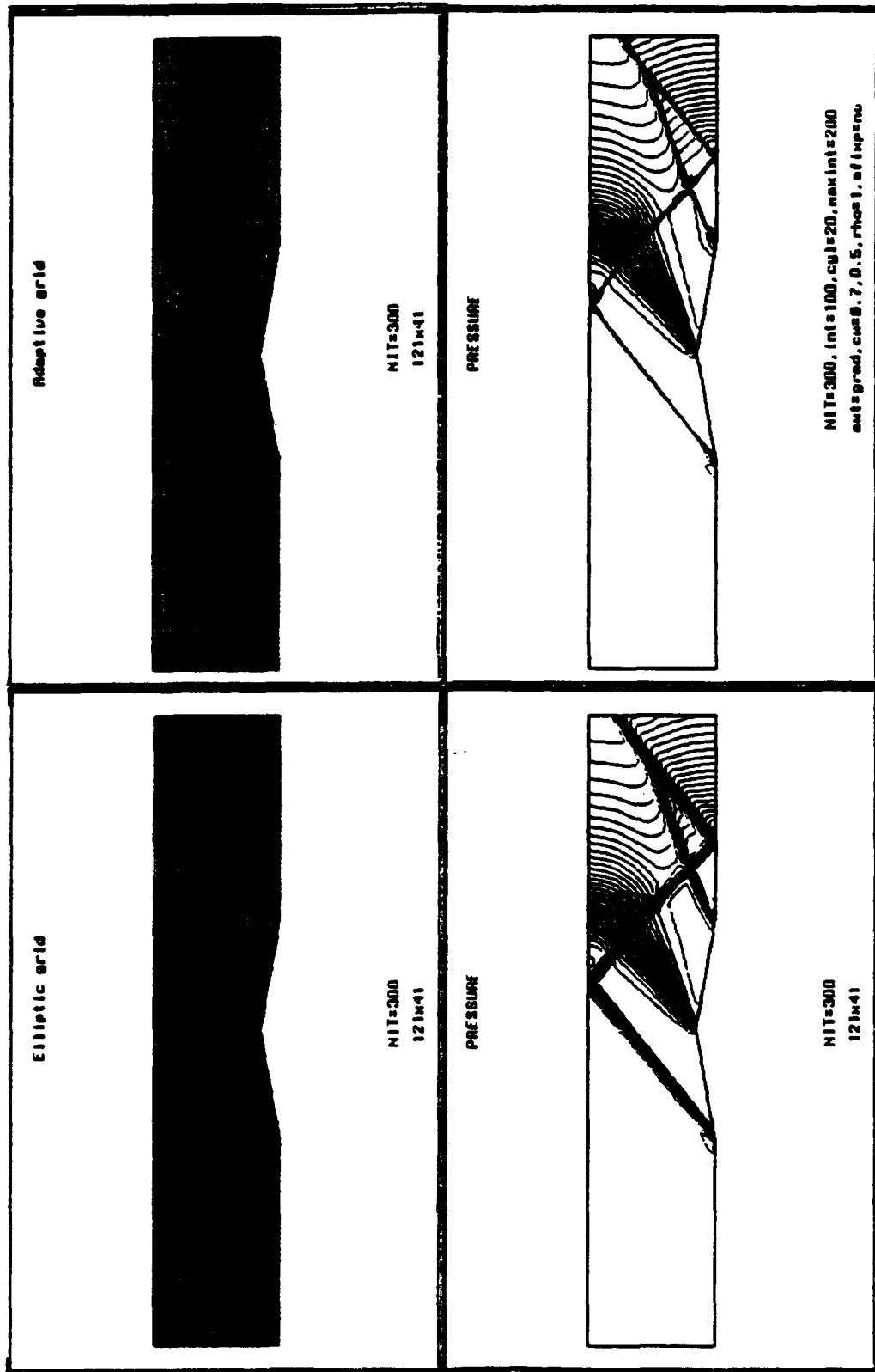


Figure 21. Contour plots of pressure on initial and adaptive grids.

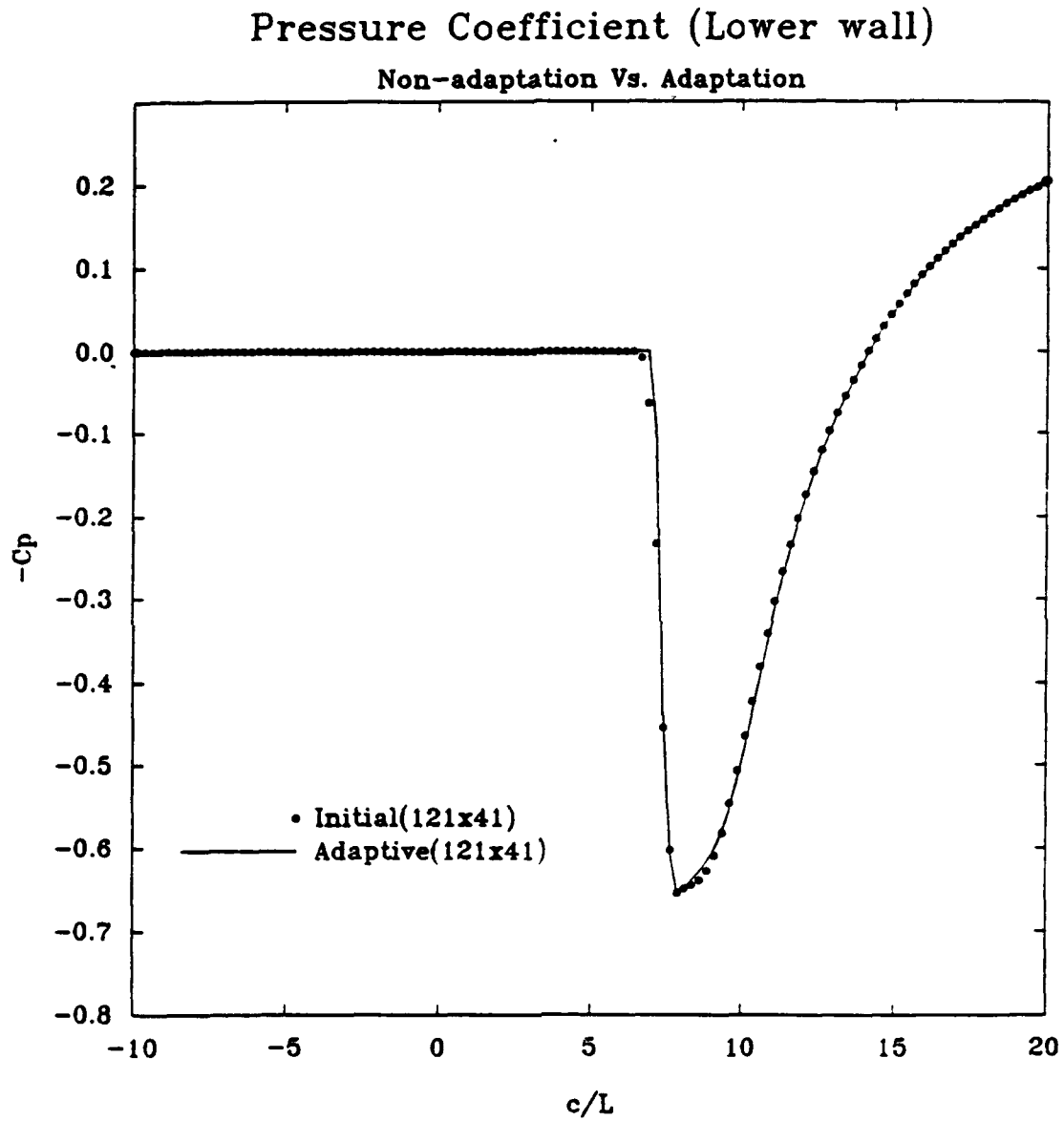


Figure 22. Adaptation with AWT=GRAD,GRAD, PRES=1.1. CW=0.7,0.7.

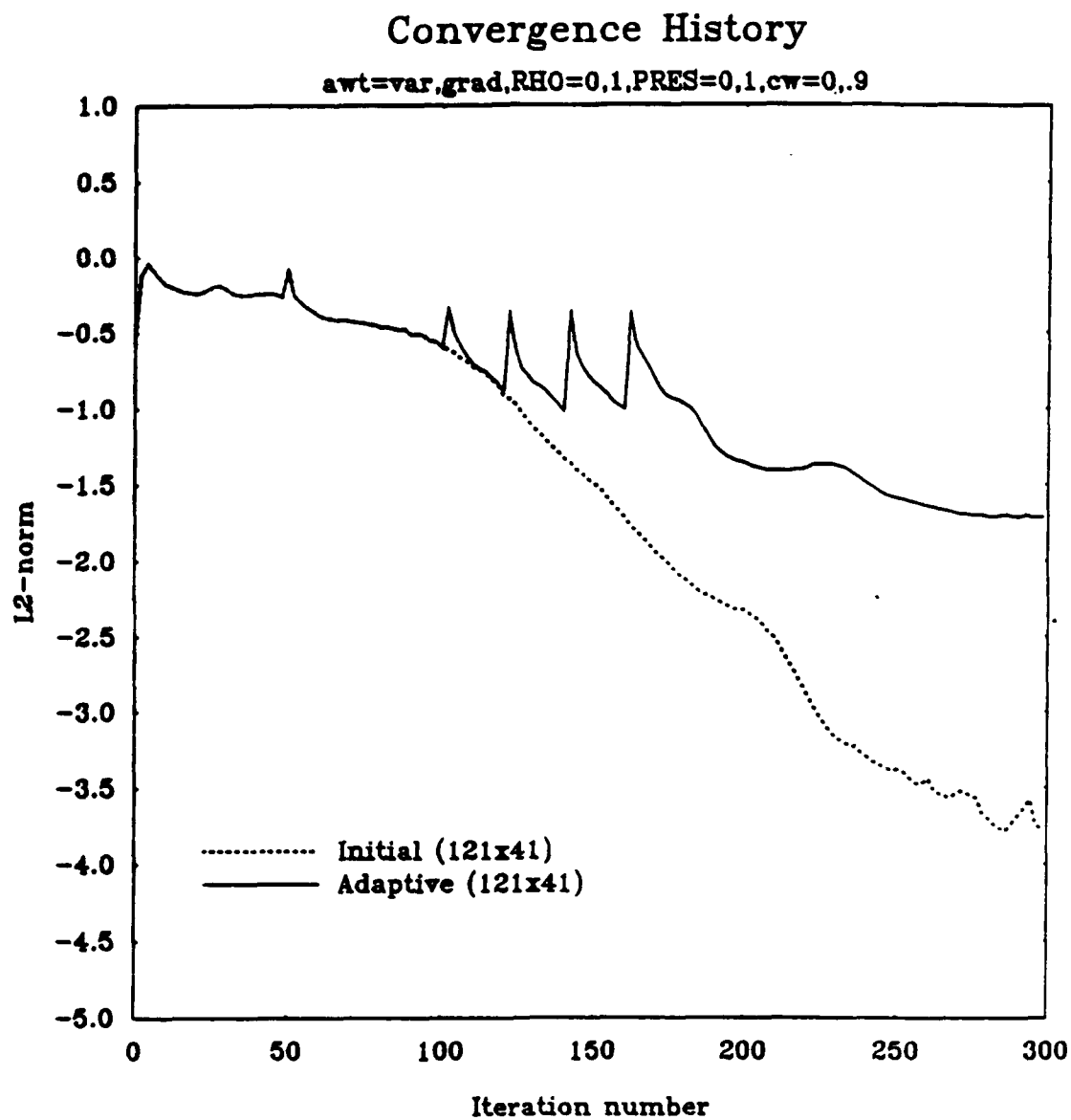


Figure 23. Convergence history of the initial and adaptive grids solutions.

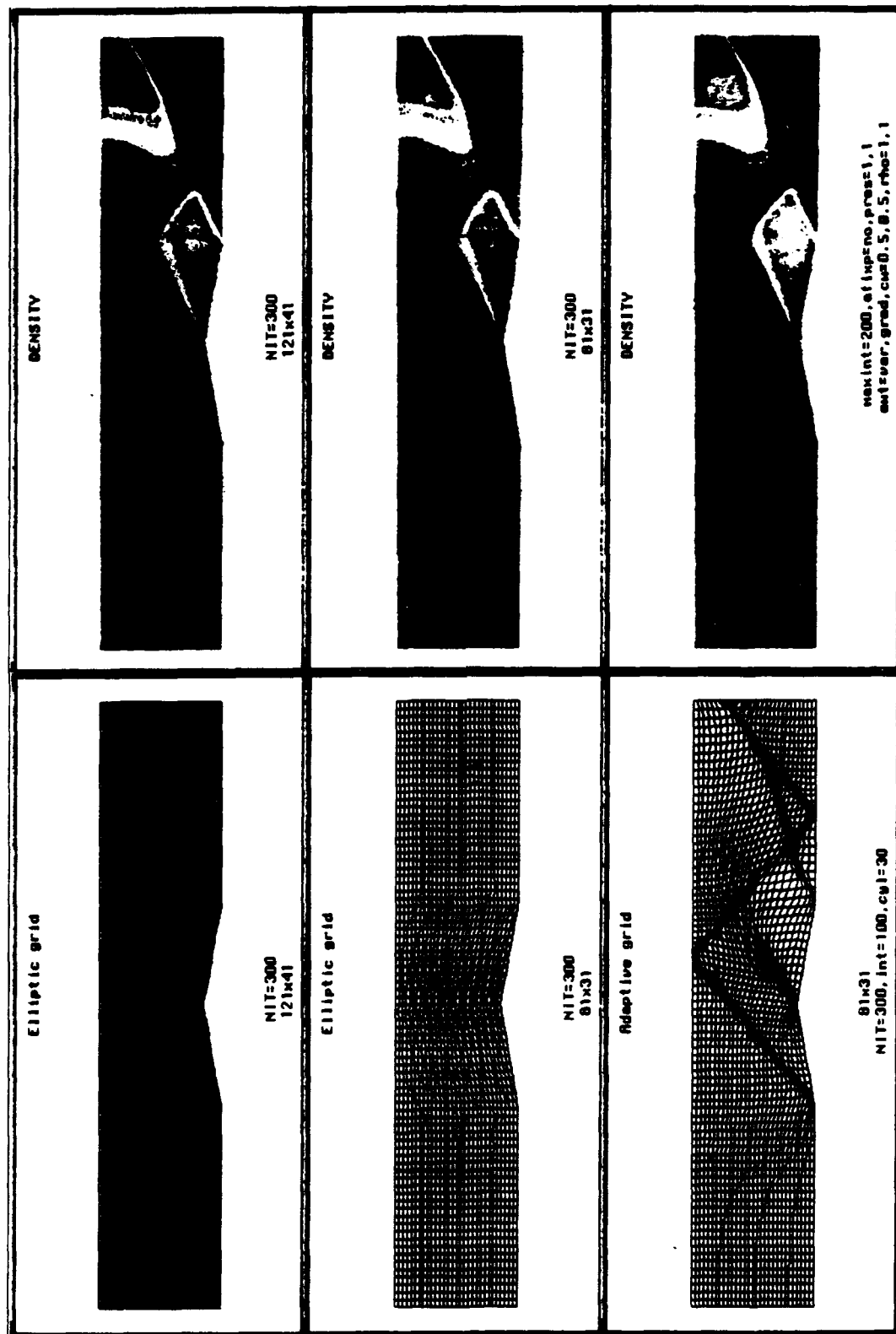


Figure 24. Contour plots of density on fine, coarse, and adaptive coarse grids.

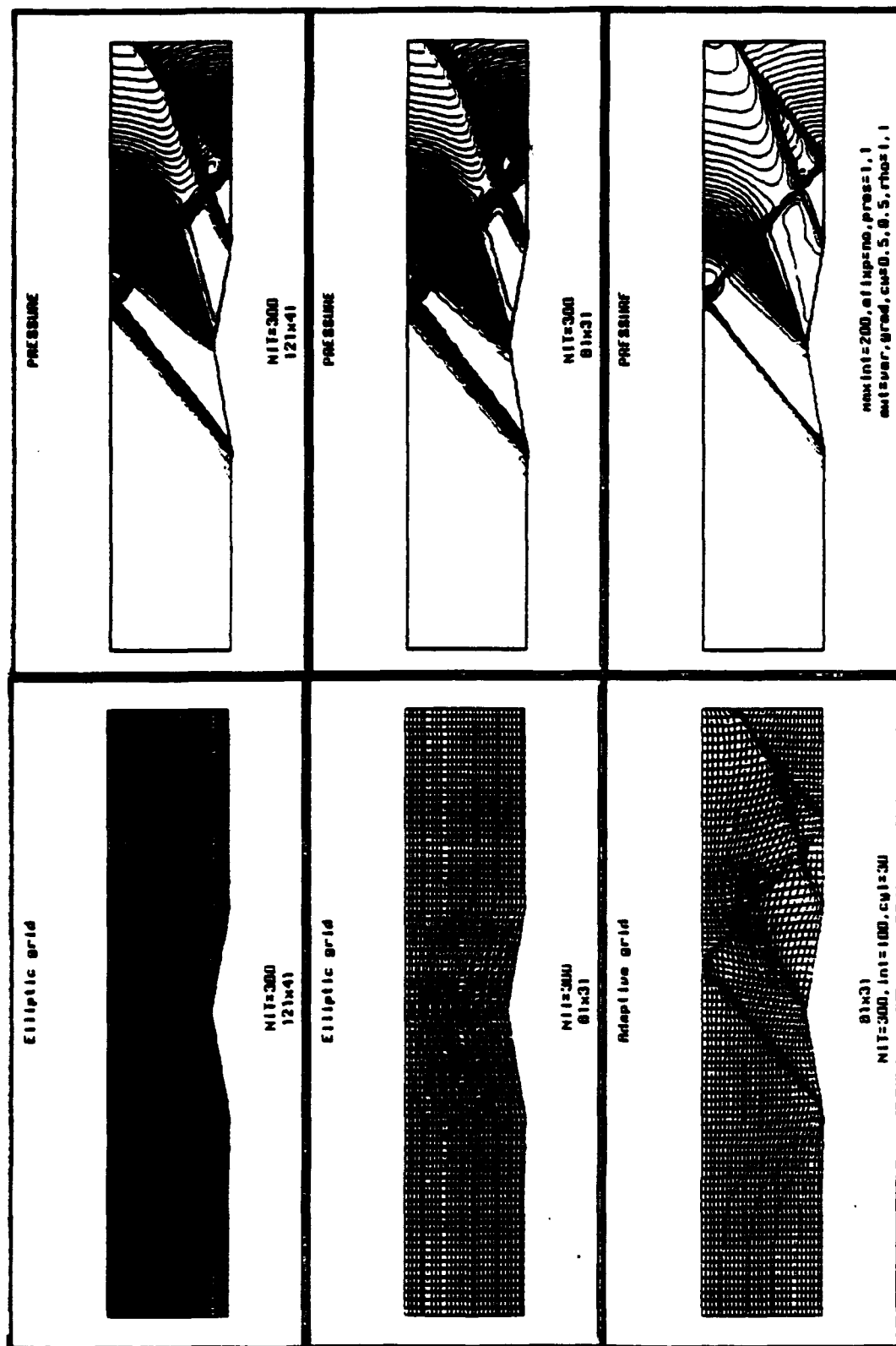


Figure 25. Contour plots of the pressure on fine, coarse, and adaptive grids.

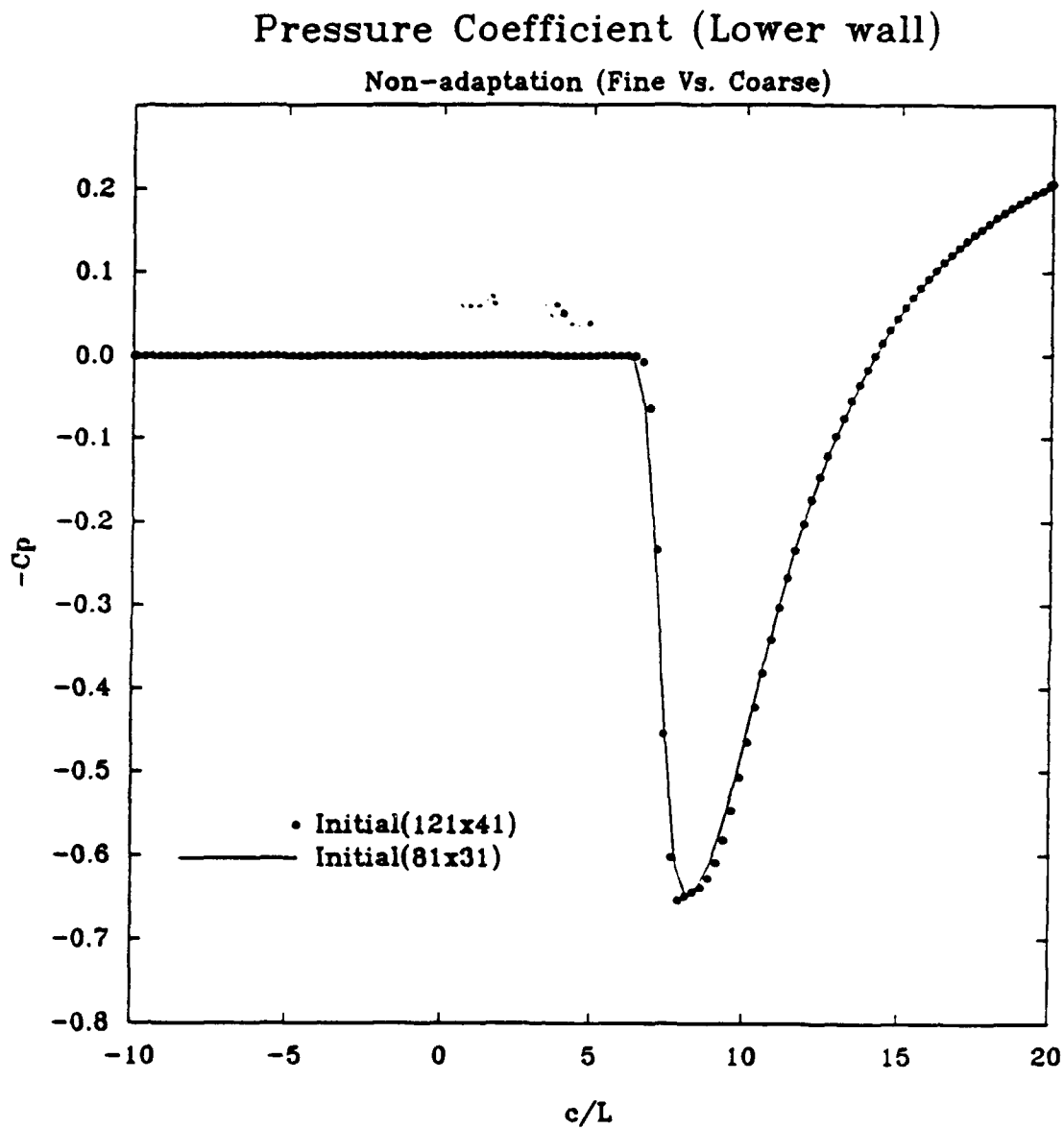


Figure 26. Pressure coefficients of the lower wall obtained from coarse and fine grids without adaptation.

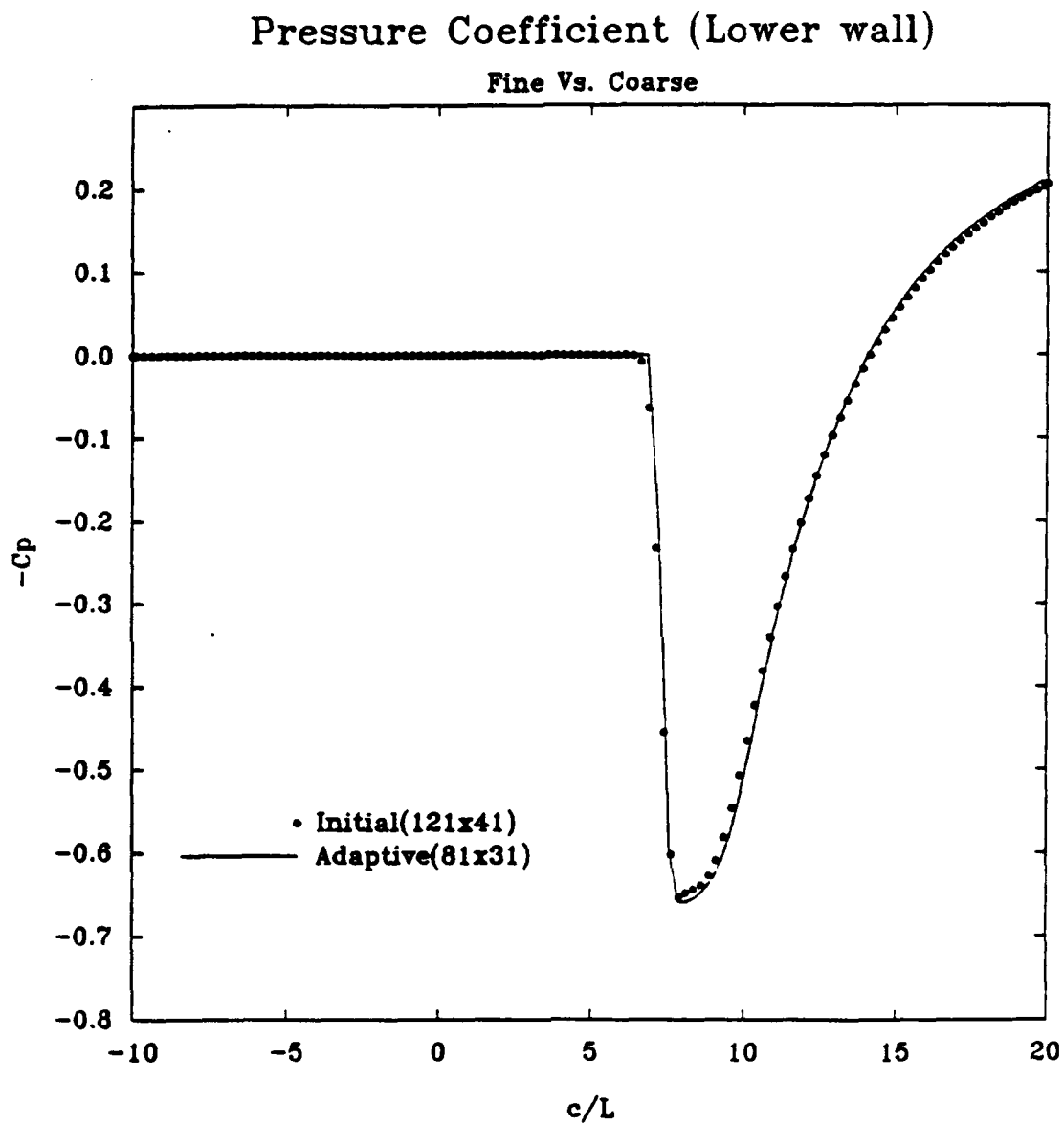


Figure 27. Adaptation with AWT=VAR,GRAD. RHO=1,1, CW=0.5,0.5.

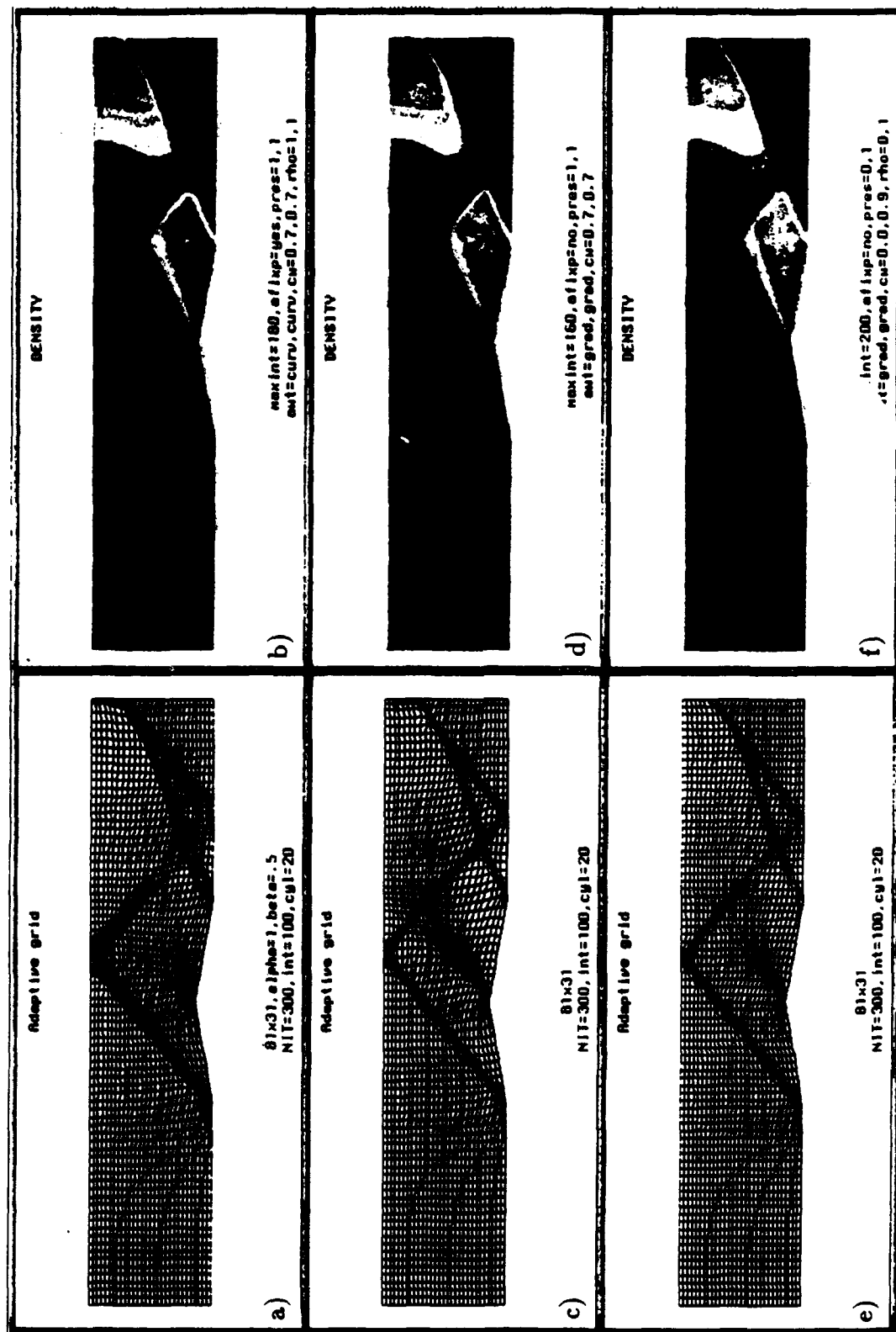


Figure 28. Contour plots of density on adaptive grids.

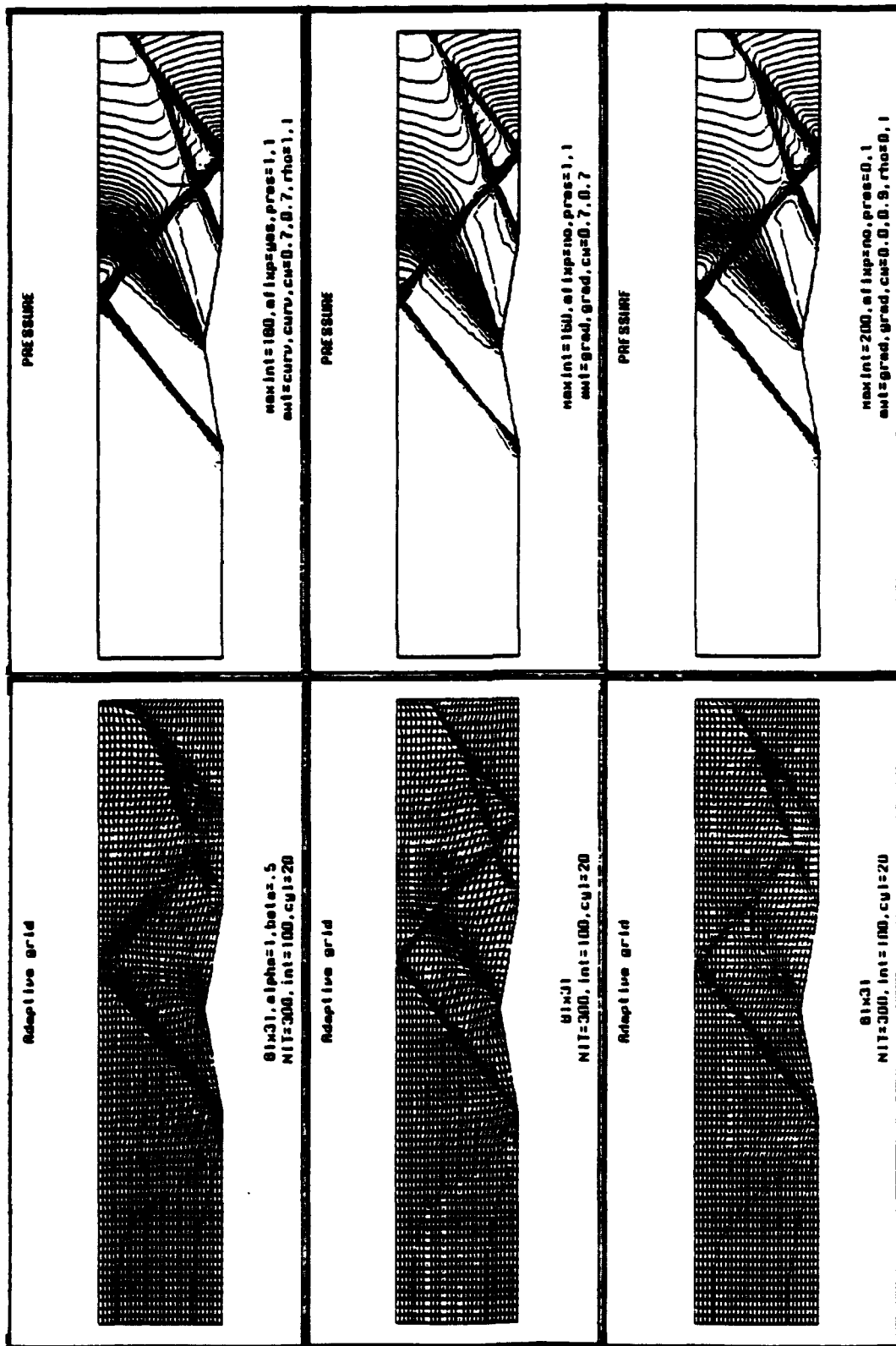


Figure 29. Contour plots of pressure on adaptive grids.

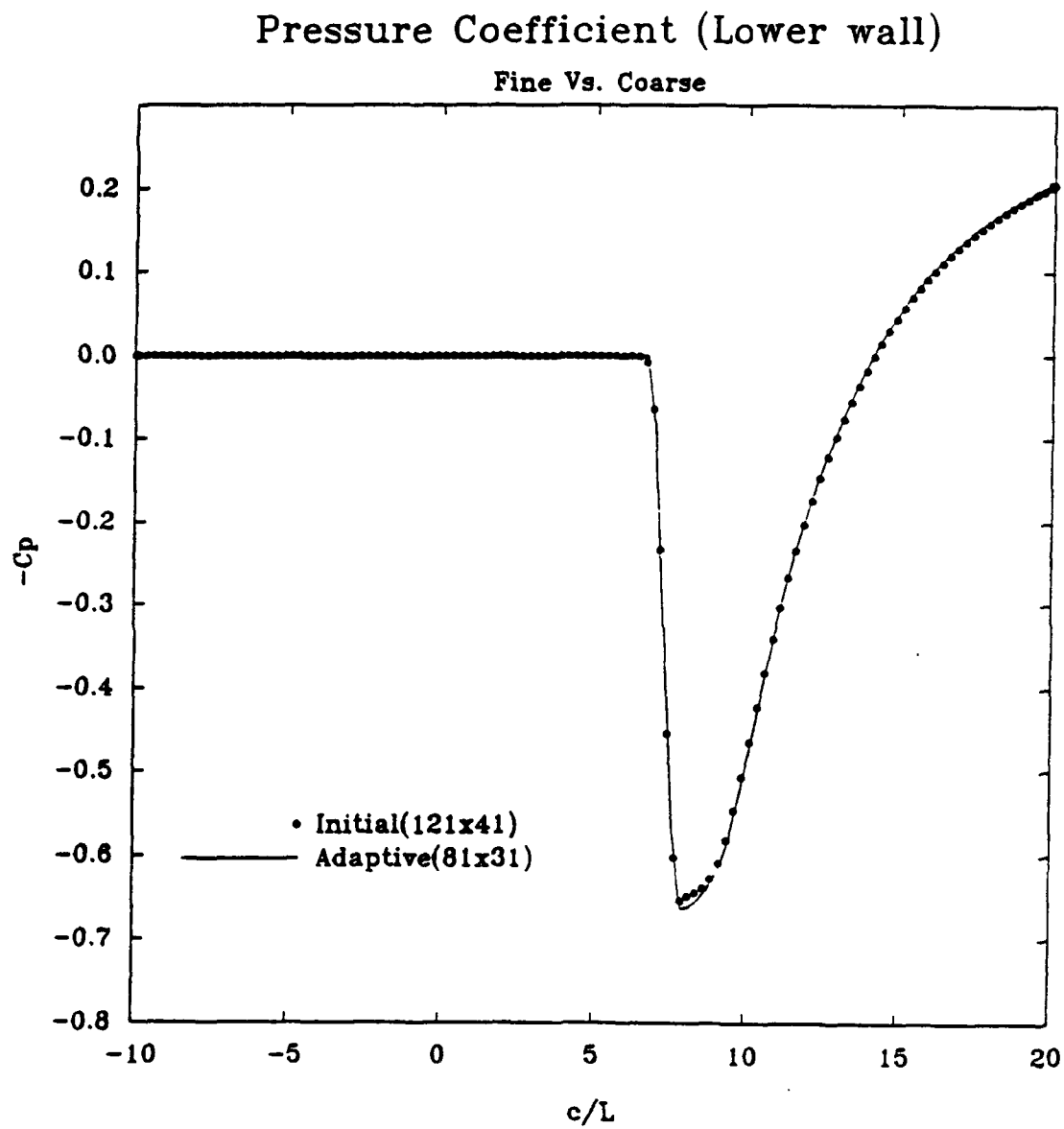


Figure 30. Adaptation with AWT=CURV,CURV, RHO=1.1, PRES=1.1 CW=0.7.0.7.

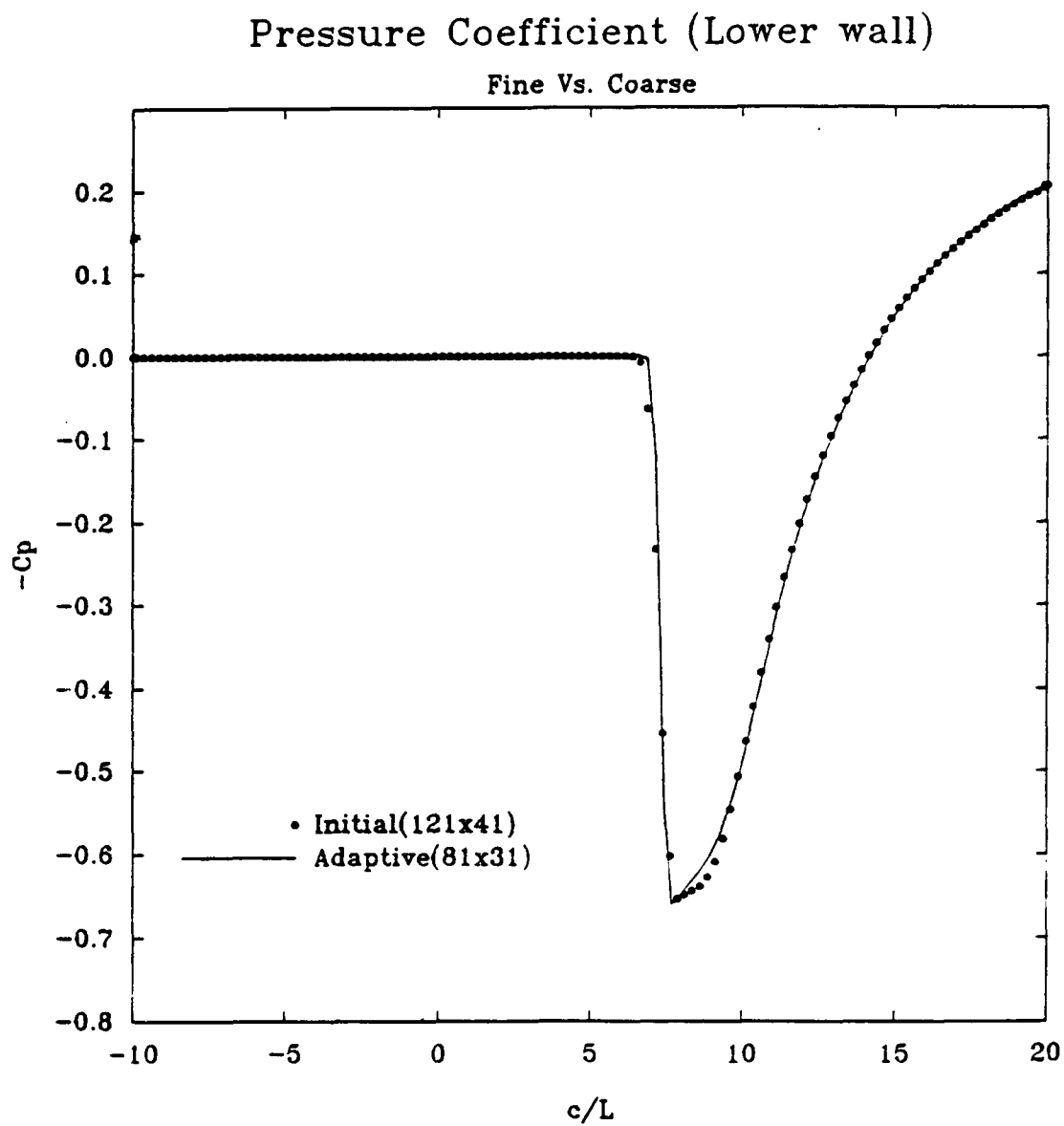


Figure 31. Adaptation with AWT=GRAD.GRAD, PRES=1.1, CW=0.7,0.7.

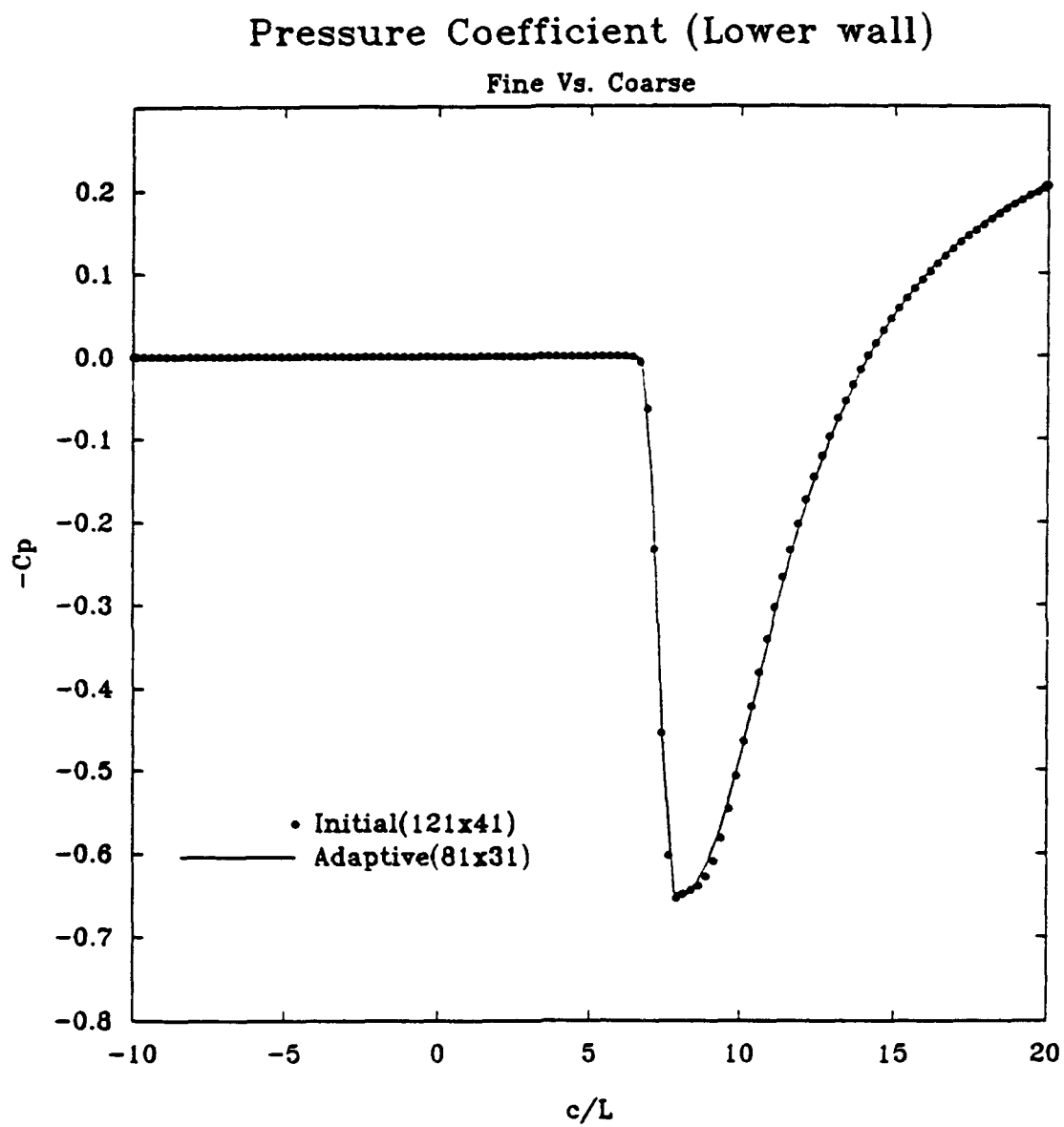


Figure 32. Adaptation with AWT=VAR,GRAD, RHO=0,1, PRES=0.1 CW=0.0.0.9.

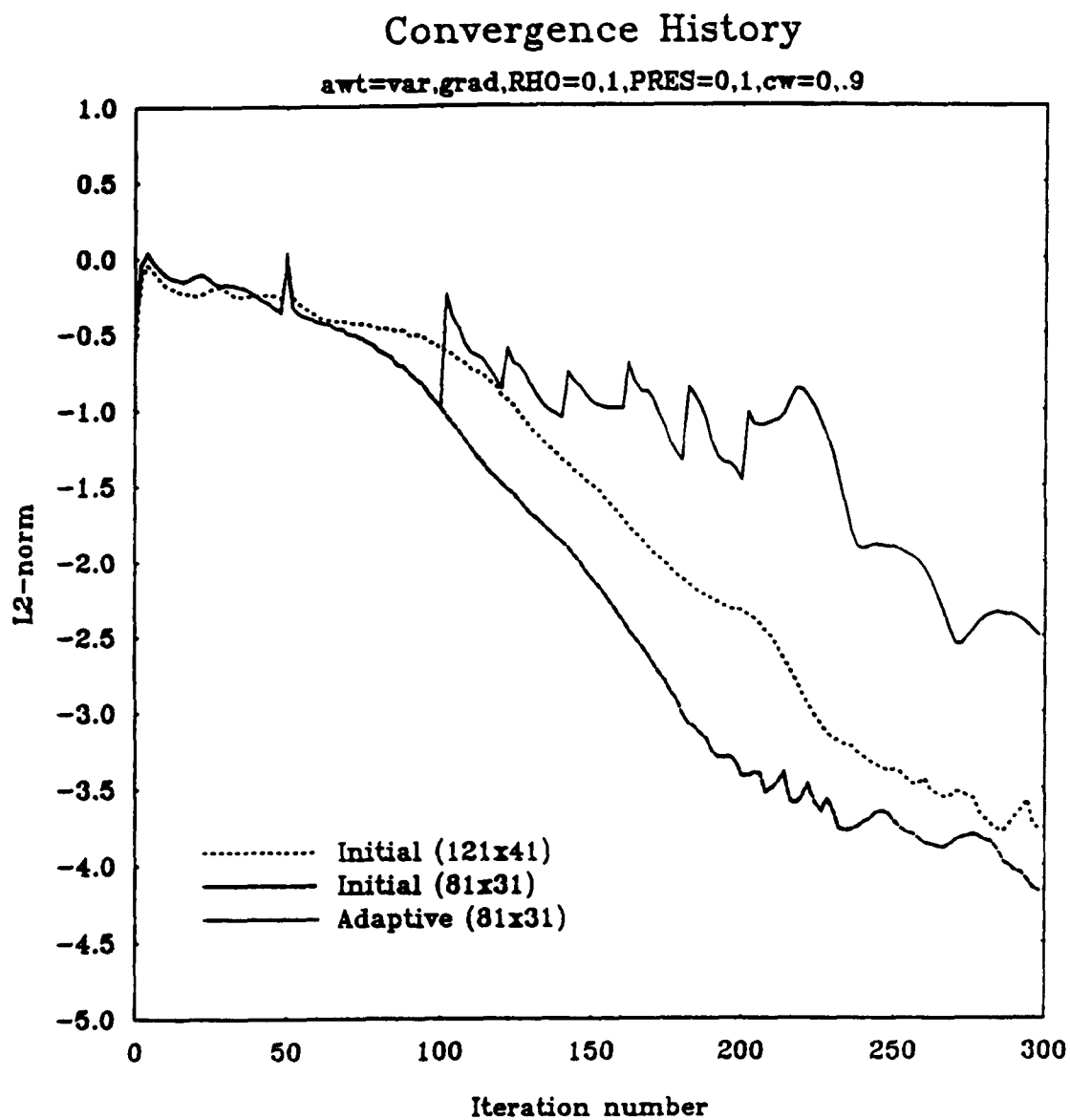


Figure 33. Convergence history of the fine, coarse, and adaptive grid solutions.

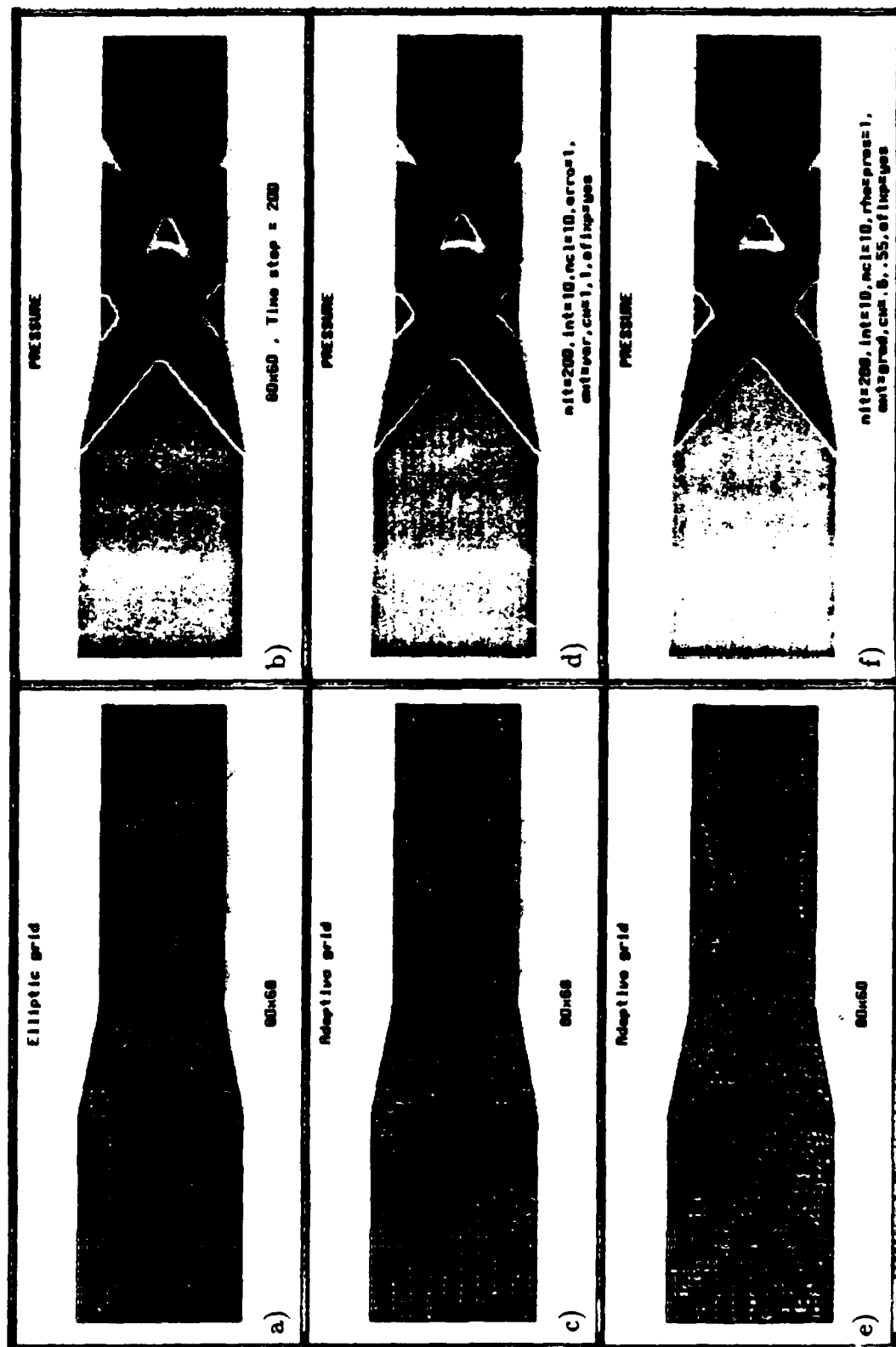


Figure 34. Contour plots of pressure on initial and adaptive grids.

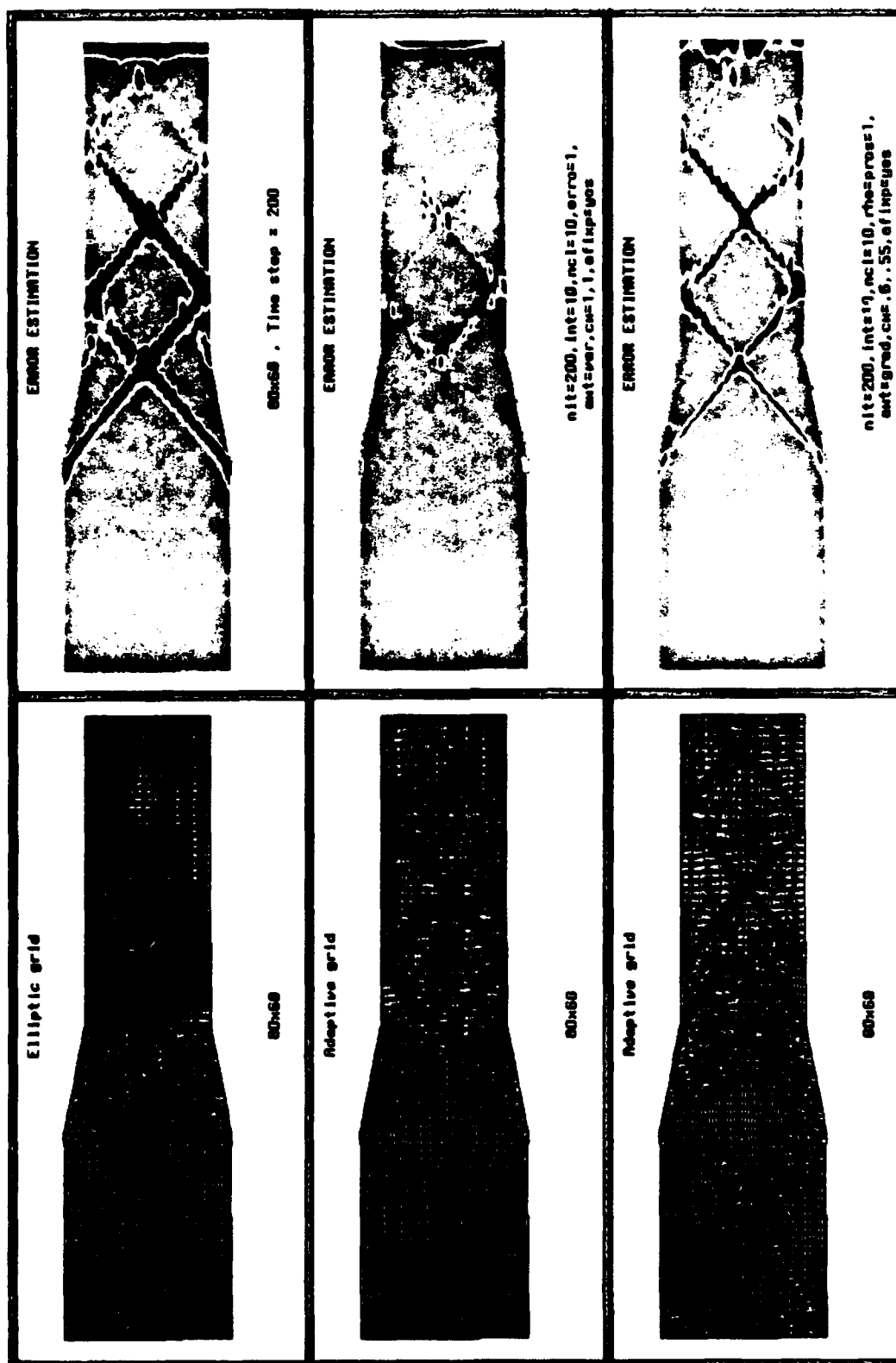


Figure 35. Contour plots of error estimation on initial and adaptive grids.

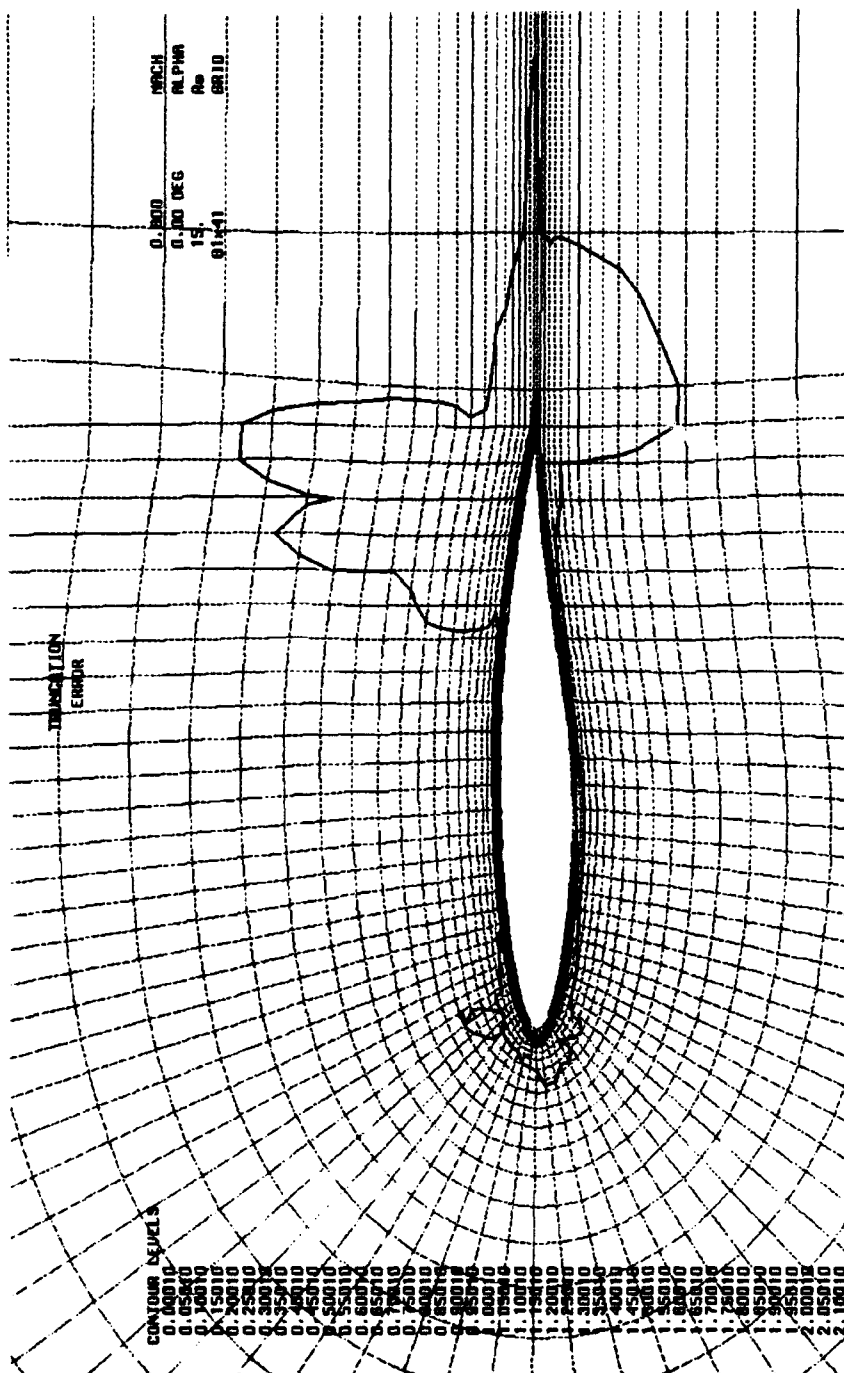


Figure 37. Contour plots of truncation error estimate.

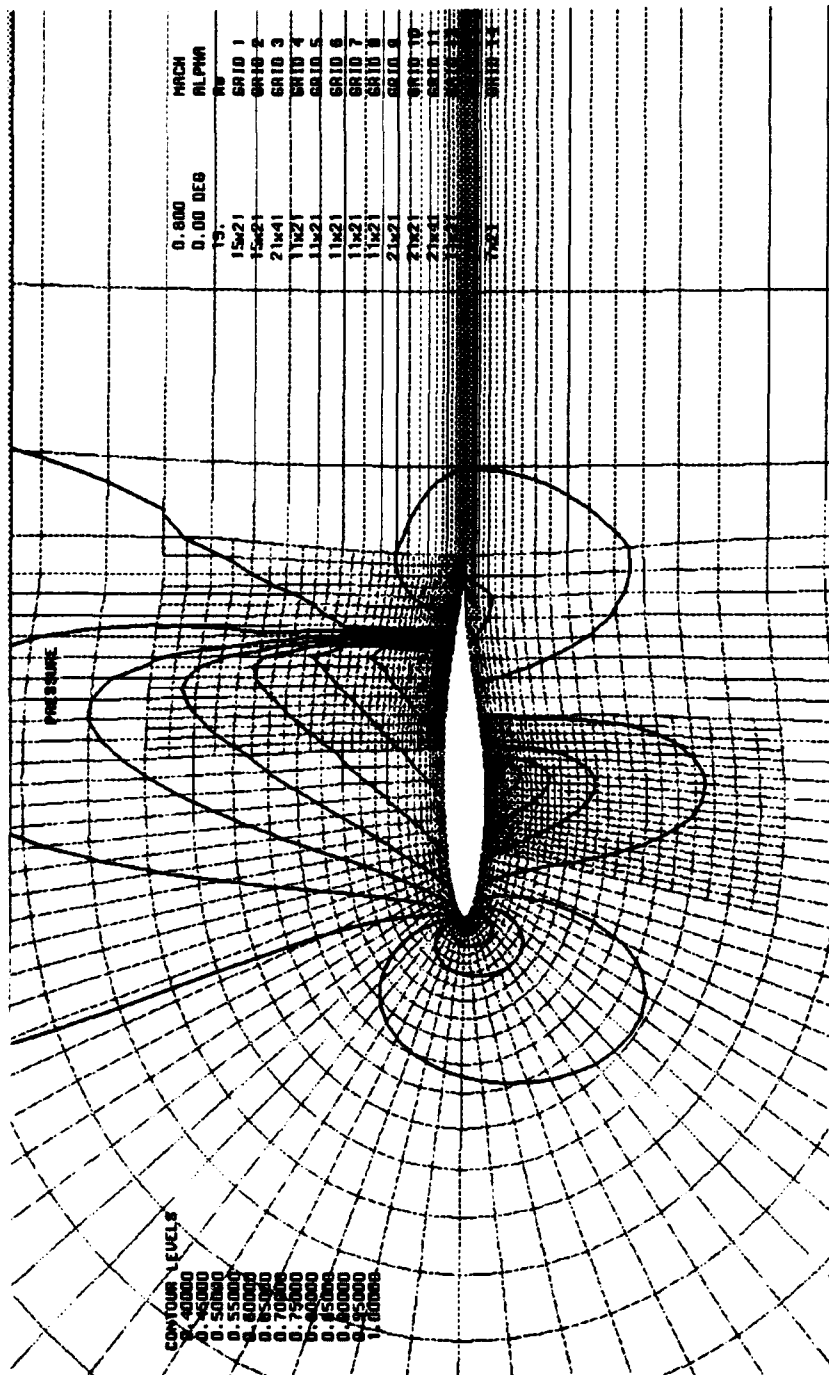


Figure 38. Contour plots of pressure on a locally refined grid.

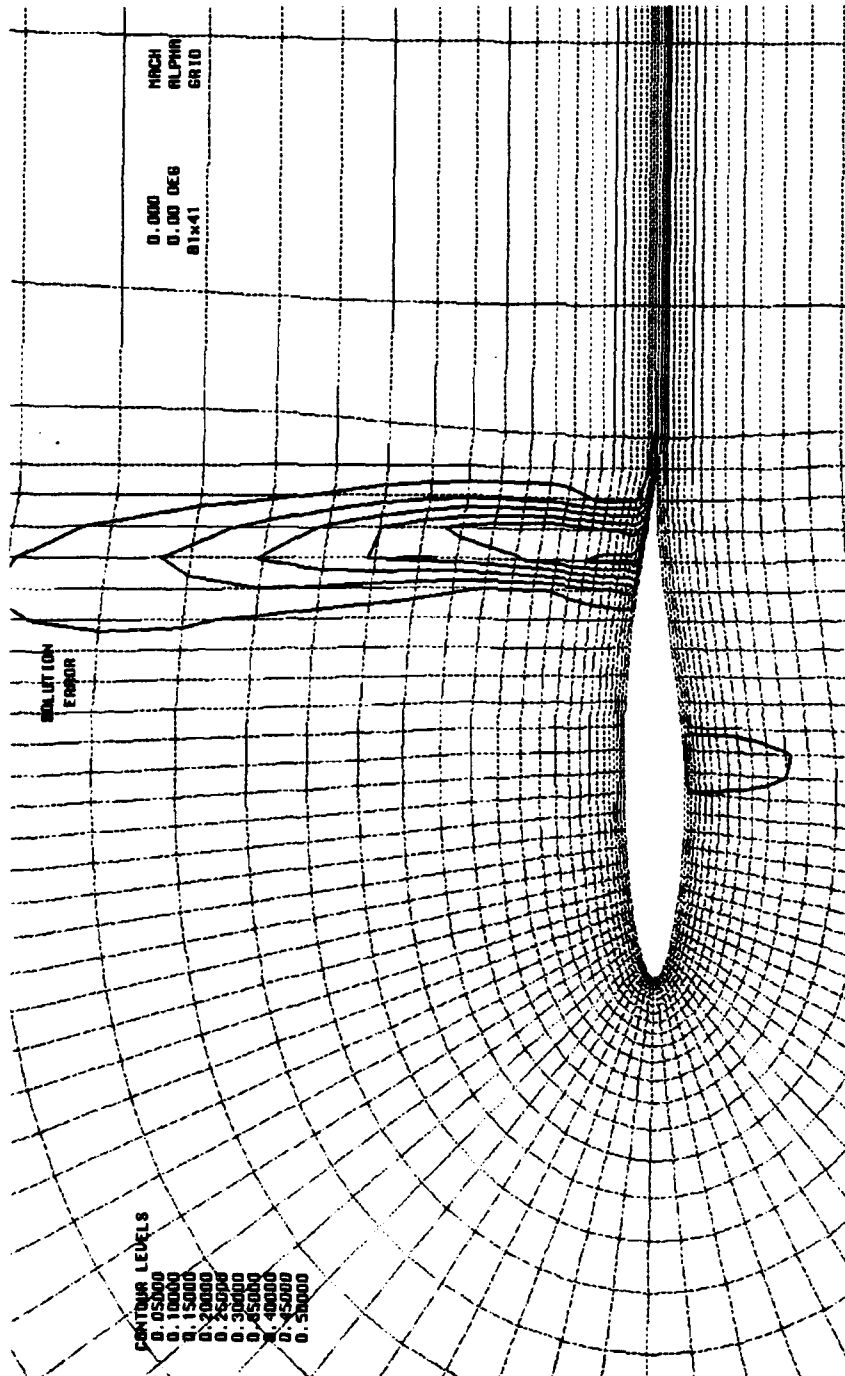


Figure 40. Contour plots of solution error computed by extrapolation.

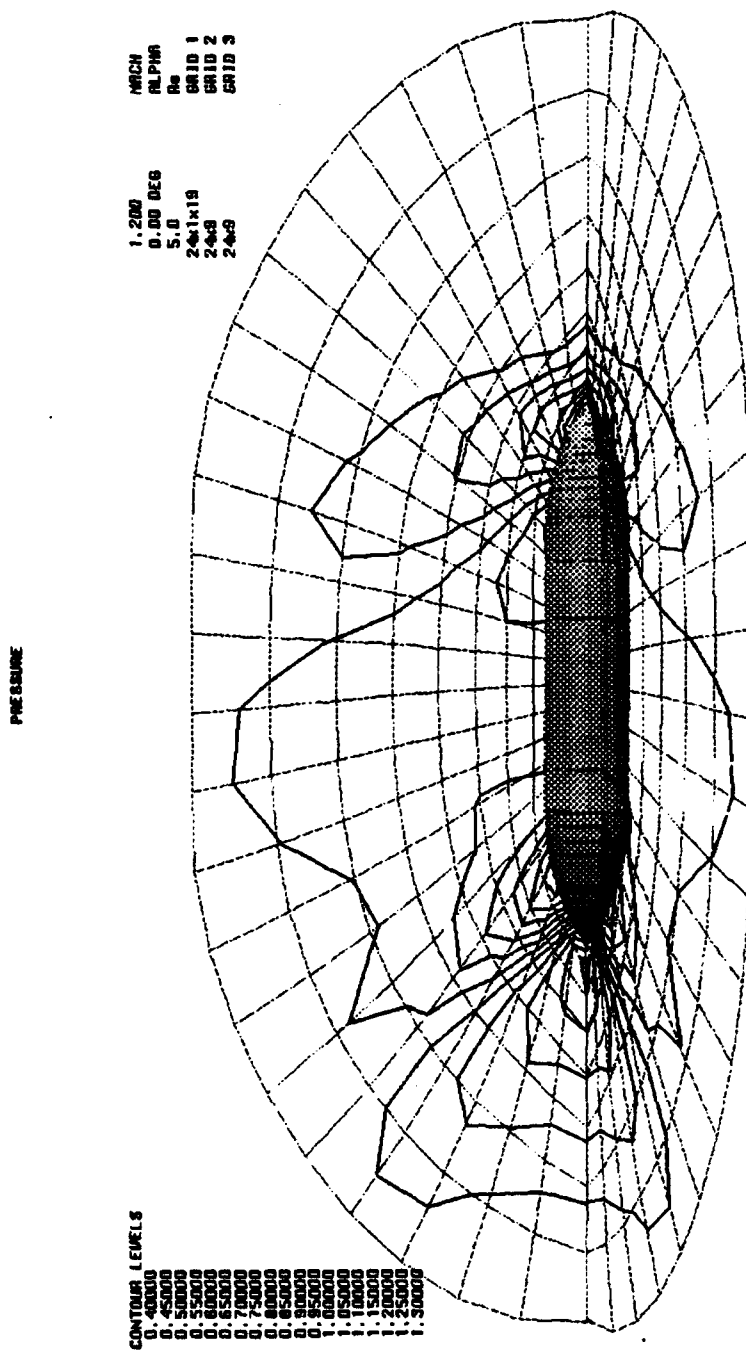


Figure 41. Contour plots of pressure on a coarse grid.

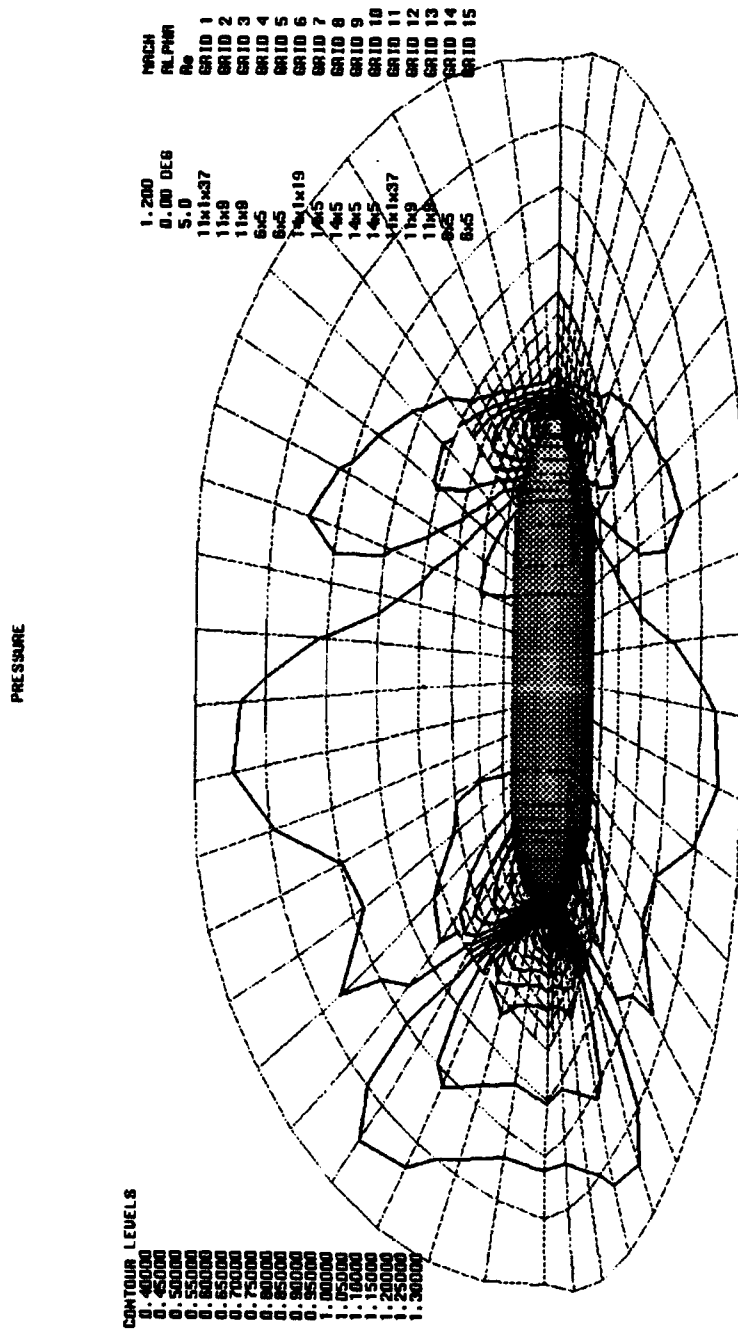


Figure 42. Contour plots of pressure on a locally refined grid.

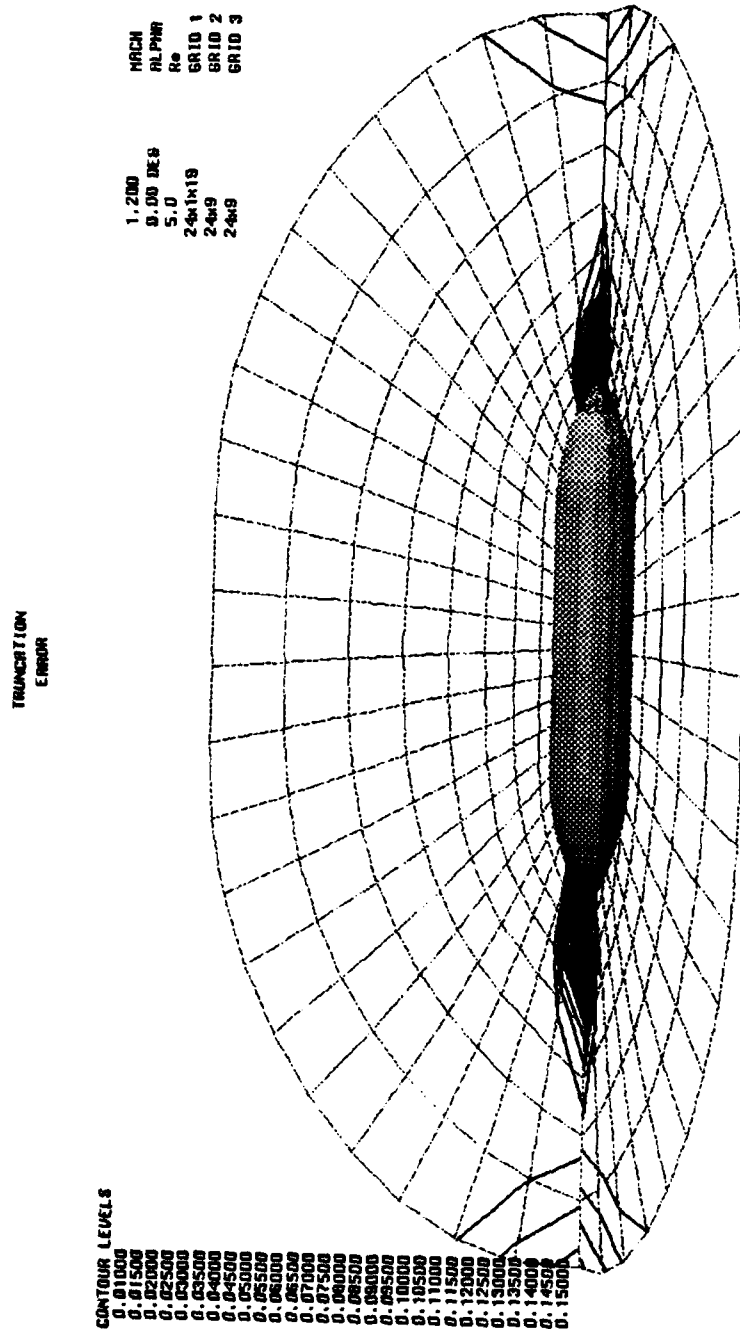


Figure 43. Contour plots of truncation error estimate.

Distribution
WL/MN-TR-91-83

90

Defense Tech Info Center
Attn: DTIC-DDAC
Cameron Station
Alexandria VA 22304-6145
2

AUL/LSE
Maxwell AFB AL 36112-5564
1

AFCSA/SAMI
The Pentagon, Room 1D363
Washington DC 20330-5425
1

Eglin AFB offices:

WL/MNOI (Scientific & Tech. Info. Facility) 1
WL/CA-N 1

WL/FIES/SURVIAC
Wright Patterson AFB OH 45433-6553
1

HQ USAFE/INATW
APO NY 09012-5001
1

Wright-Patterson AFB OH 45433-6503

ASD/ENSTA 1
ASD/XRH 1

Wright-Patterson AFB OH 45433-6553

WL/CA-F 1
WL/FIM 1
WL/FIB 1
WRDC/FIBA 1
V/L/FIGX 1
WL/FIGCC 1

WL/TXA
Wright-Patterson B OH 45433-6523
1

AFIA/INT
Bolling AFB DC 20332-5000
1

EOARD/LDV
Box 14
FPO NY 09510-0200
1

Commander
U.S. Army Missile Command
Redstone Sci Info Center
Attn: AMSMI-RD-CS/R Documents
Redstone Arsenal AL 35898-5241
2

Commander
Naval Weapons Center (Code 3431)
Attn: Technical Library
China Lake CA 93555-6001
1

NASA Ames Research Center
Attn: Dr. Tony Strawa, MS 229-3
Moffett Field CA 94035-0000
1

NASA Langley Research Center
Technical Library, MS 185
Attn: Document Cataloging
Hampton VA 23665-5225
1

Eglin AFB offices:

WL/MNAA 4
WL/MNPX 1
WL/MNSI 1
ASD/XRC 1
AFDTC/PA 1

Engineering Research Center
Attn: Dr. Joe Thompson
Box 6176
Mississippi State MS 39762
3

Calspan Corporation
Attn: Dr. Jack Benek, MS 600
Arnold AFB TN 37389-9998
1

END