DESCRIPTION OF A HYPERBOLIC GRID

GENERATING PROCEDURE FOR ARBITRARY

TWO-DIMENSIONAL BODIES

DON W. KINSEY

TIMOTHY J. BARTH

AERODYNAMICS & AIRFRAME BRANCH

AEROMECHANICS DIVISION

JULY 1984

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION
IS UNLIMITED

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433
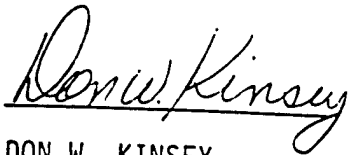
19980303 123

DTIC QUALITY INSPECTED

FOREWARD

· This report was prepared by Don W. Kinsey and Timothy J. Barth of the Aerodynamics and Airframe Branch, Aeromechanics Division, Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio. The work was performed under Work Units 24041074 and 2307N603.
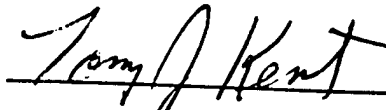
Most finite difference simulations of fluid flow about complex bodies require the development of a computational grid. This report describes the procedures developed for creating a two-dimensional computational grid around arbitrary bodies.

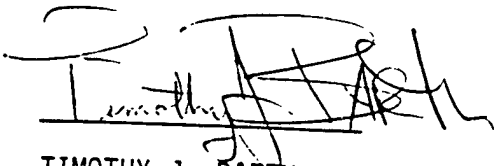This report has been reviewed and is approved.

DON W. KINSEY

Aerodynamics Engineer

TIMOTHY J. BARTH

Research Scientist

TOMMY J. KENT, MAJ, USAF

Chief, Aerodynamics and

Airframe Branch

ABSTRACT

This report describes the procedures used to create O and C-type solution grids around arbitrary two-dimensional bodies. The grid generation procedure is a modification of the O-type hyperbolic method of Steger and Chaussee[1]. The procedure described in Reference (1) was modified by including additional dissipation terms and by changing the form of the dissipation described. The modifications necessary to produce a C-type grid are discussed and examples are provided.

A brief description of the theory and development of the governing hyperbolic equations is provided. Example results for both an airfoil section and a complex body (X-24C) are shown. A discussion of the user definable input variables and their effects on the resulting grid is included.

For many solution procedures it is desirable that the distribution of points around the airfoil be "second order smooth."[2] Since most airfoils are defined with a somewhat random distribution of points, a routine was used to redefine the coordinates in a smooth distribution. The routine also allows selective clustering of grid points in regions of interest. A brief description of this routine and the effects of the different input parameters are included as Appendix A.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF SYMBOLS

ALPHA	Parameter defining the finite difference representation for marching in the η direction
C	Grid pattern shaped like the letter C
DSETA	Initial step size in the η direction
ESCAL	Scaling parameter for grid spacing in the η direction
JMAX	Maximum number of points defining the body surface
KMAX	Maximum number of grid surfaces surrounding the body
L	Reference body length (usually 1)
O	Grid pattern shaped like the letter O
PDE	Partial differential equation
RL	Reynolds number based on length L
S	Perimeter of body measured clockwise from XMAX
SETAMX	Maximum value for η (in reference lengths)
SMU	Parameter defining the amount of explicit smoothing
SMUIM	Parameter defining the amount of implicit smoothing
X	Cartesian coordinate defining the body
XMAX	Maximum value of the coordinates defining the body (including the zero-thickness wake extension)
Y	Cartesian coordinate defining the body
ξ	Transformed coordinate that encircles the body
η	Transformed coordinate that extends radially from the body
ε	Exponential stretching function
δ	Central difference operator
∇	Backward difference operator
Δ	Forward difference operator

I. INTRODUCTION

Finite difference solutions of the partial differential equations that govern fluid flow about arbitrary bodies require the development of a computational grid around the body. In particular, a grid system that conforms to the shape of the body has the very important advantages of simplifying the partial differential equation solution technique and simplifying the satisfaction of the boundary conditions at the body. In many cases, the character of the entire flow field is determined in the high gradient regions near the body. Therefore, the development of a body-conforming coordinate system is a necessary and very important first step in numerical solutions of fluid flow problems.

Algebraic transformation techniques for grid generation have been used for some applications and have the advantage of being relatively simple and fast. However, the most popular techniques use the solution of partial differential equations (PDE). Procedures exist that solve elliptic, hyperbolic and parabolic PDEs. Elliptic routines·are most often used and probably are more generally applicable to a wider range of problems. Hyperbolic routines are relatively fast and simple, but are not easily adapted to physical far field constraints (such as internal flows and multiple bodies). Parabolic routines have been recently developed and show some promise. The PDE methods attempt to produce nearly orthogonal grids with assurance of no crossover of adjacent grid lines.

This report will discuss the characteristics of a hyperbolic grid generation routine modified to produce a computational grid over arbitrary two-dimensional bodies. The hyperbolic grid generation routine was chosen because it is quite fast, provides nearly orthogonal grids and has good user control of the grid spacings. The routine should work well for generating a grid around bodies where there are no physical constraints (such as other bodies, walls, etc.)

1

within the region of the grid. The actual grid generation starts at the body at user defined locations and marches out to an outer boundary.

The core of this program was adapted from the procedure described by Steger and Chaussee in Reference 1. The purpose of this report is to describe and document the modifications made to the O-grid algorithm of Steger and Chaussee and to provide examples of how the various user defined variables affect the results.

Another program called SMOOTH has been developed to allow a smoothly varying distribution of the points describing the body. The same process that insures smoothness also allows arbitrary clustering and/or spreading-out of points around the body. This program is briefly described and some representative results are provided in Appendix A.

II. THEORY DEVELOPMENT

In the grid generation algorithm developed by Steger and Chaussee, definitions of grid orthogonality and transformation Jacobian were chosen to devise a scheme mapping (x,y) to (ξ,η). A typical transformation for an airfoil is shown in Figure 1.

$$X_\xi X_\eta + Y_\xi Y_\eta = 0 \qquad\qquad \text{Orthogonality} \qquad (1)$$

$$X_\xi Y_\eta - X_\eta Y_\xi = 1/J \qquad\qquad \text{Inverse Jacobian} \qquad (2)$$

Note that in general, area integrands in physical and computational domains can be written

$$dA = dXdY = 1/J \; d\xi d\eta$$

Numerically $\Delta\xi = \Delta\eta = 1$, so the inverse Jacobian approximates the physical cell area. Equations (1) and (2) are locally linearized using
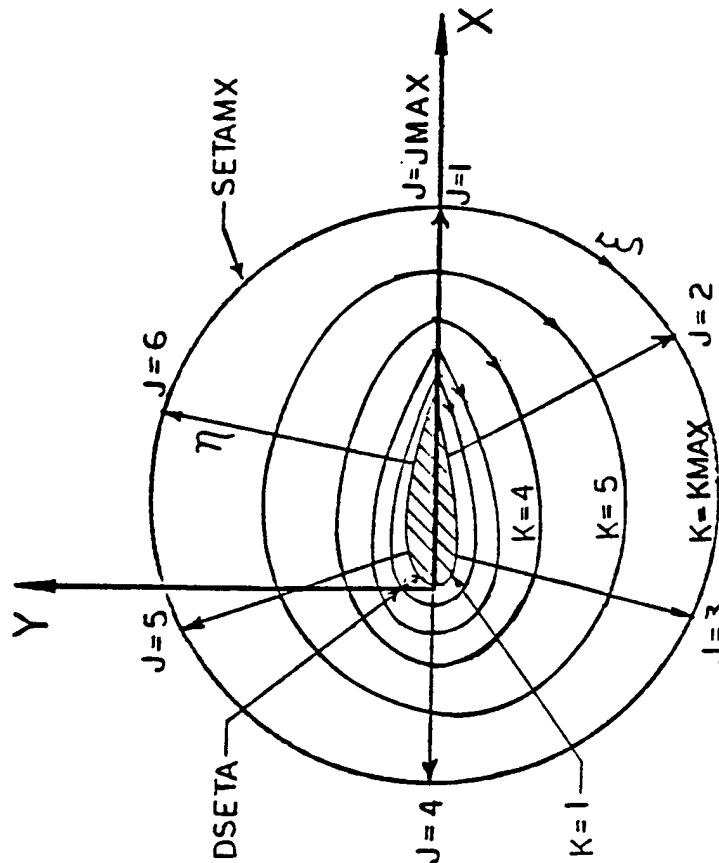
$$X = X^\circ + \Delta X$$
$$Y = Y^\circ + \Delta Y$$

where X°, y° is a nearby location. The linearized set of equations become (after some algebraic manipulation)

$$\begin{bmatrix} X_\eta^\circ & Y_\eta^\circ \\ Y_\eta^\circ & -X_\eta^\circ \end{bmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}_\xi + \begin{bmatrix} X_\xi^\circ & Y_\xi^\circ \\ -Y_\xi^\circ & X_\xi^\circ \end{bmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}_\eta = \begin{pmatrix} 0 \\ V+V^\circ \end{pmatrix}$$

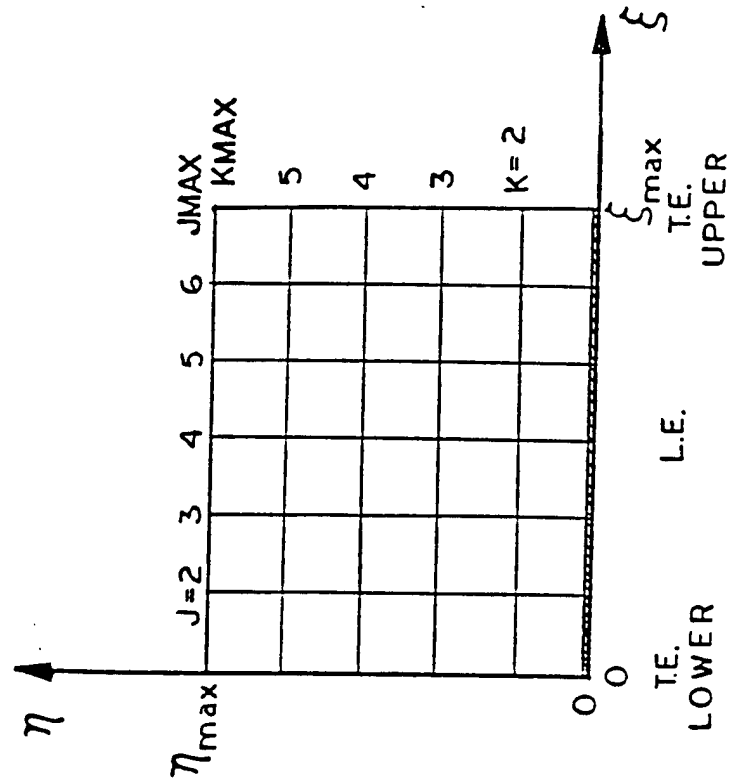COORDINATE TRANSFORMATION

TRANSFORMED COORDINATES

CARTESIAN COORDINATES

FIGURE 1

or

$$A \vec{R}_\xi + B \vec{R}_\eta = \vec{f} \tag{3}$$

The specific numerical procedure for solving equation (3) depends on the class of equations being solved (elliptic, hyperbolic, or parabolic). The class can be determined by inspecting $B^{-1} A$.

$$B^{-1} A = 1/\hat{J} \begin{bmatrix} X_\xi^\circ X_\eta^\circ - Y_\xi^\circ Y_\eta^\circ & X_\xi^\circ Y_\eta^\circ + Y_\xi^\circ X_\eta^\circ \\ X_\eta^\circ Y_\xi^\circ + Y_\eta^\circ X_\xi^\circ & Y_\xi^\circ Y_\eta^\circ - X_\eta^\circ X_\xi^\circ \end{bmatrix}$$

where

$$\hat{J} = X_\xi^{\circ 2} + Y_\xi^{\circ 2}$$

Since $B^{-1} A$ is symmetric, it has real eigenvalues, specifically:

$$\lambda_{1,2} = \pm 1/\hat{J} \sqrt{(X_\xi^\circ X_\eta^\circ - Y_\xi^\circ Y_\eta^\circ)^2 + (X_\xi^\circ Y_\eta^\circ + Y_\xi^\circ X_\eta^\circ)^2}$$

This indicates that the system is hyperbolic and can be marched in η.

The finite difference solution scheme used is centrally differenced in ξ and backward differenced in η. The scheme can be written as

$$[I + B^{-1} A \, \delta_\xi] \vec{R}_{j,k+1} = \vec{R}_{j,k} + B^{-1} \vec{f}_{j,k+1} + \epsilon_e (\nabla_j \Delta_j)^2 \vec{R}_{j,k}$$

where f_{k+1} is known at the k level, and $\epsilon_e (\nabla_j \Delta_j)^2 \vec{R}_{j,k}$ is an added, fourth order dissipation term in ξ as discussed in Beam and Warming.[3]

5

Typically X_ξ°, Y_ξ° are approximated as follows:

$$X_\xi^\circ = (X_{j+1,k} - X_{j-1,k})/2$$
$$Y_\xi^\circ = (Y_{j+1,k} - Y_{j-1,k})/2$$

While X_η°, Y_η° are extracted from the nonlinear differential equations (1) and (2).

$$x_\eta^\circ = \frac{-Y_\xi^\circ V^\circ}{\hat{J}} \quad , \qquad Y_\eta^\circ = \frac{X_\xi^\circ V^\circ}{\hat{J}}$$

This allows the nonlinearity to be maintained. It is important to note that the cell areas at the k and k+1 levels are assumed to be known. This can be accomplished in several ways. Steger and Chaussee construct polar grids about two individual circles whose circumference is the arc length of the body to be meshed. The grids on these two circles have the same spacing in the radial direction, (usually an exponential stretching to cluster points near the body). However, the two circles differ in grid spacing in the circumferential direction. One circle has points placed in equal spaced increments around the circumference. Cell areas are extracted from these polar grids such that when near the body, cell areas are calculated using the nonuniform circumferential distribution of points. When far from the body, areas from the uniform circumferentially distributed circle are used. A smooth function transitions from one area type to the other for points between these two extremes.

The solution algorithm based on these equation's exhibits difficulties for geometries with slope discontinuities and regions of reverse (concave) curvature. Discontinuities can propagate into the grid interior with undesirable results. Drawing from experience in other hyperbolic systems it is possible that these problems may be circumvented by carefully including other

6

forms of dissipation without drastically compromising the orthogonal qualities of the grid. If an analogy is made between marching in time and marching in η, it seems reasonable that adding both temporal and spacial dissipation would be beneficial in insuring smoothness. To accomplish this a more general class of integration in η was chosen:

$$\frac{\vec{R}_{k+1} - \vec{R}_k}{\Delta\eta} = (1-\alpha) \left(\frac{\partial R}{\partial\eta}\right)_k + \alpha \left(\frac{\partial R}{\partial\eta}\right)_{k+1}$$

When $\alpha = 1$, the original backward differenced scheme is obtained. From a Taylor expansion it can be shown that the numerical error term for this scheme is: $\frac{1}{2}(1-2\alpha) \Delta\eta \frac{\partial^2 R}{\partial\eta^2}$ + higher order terms. Note that for $\alpha = \frac{1}{2}$ the integration is a trapezoidal type and is formally second order in η. For $\alpha > \frac{1}{2}$ the error term has a dissipative effect in the η direction. Rewriting the algorithm in the so called "delta" form, such that increments in $\vec{R}$, $(\vec{R}_{k+1} - \vec{R}_k)$ are solved, the algorithm becomes

$$[I + \alpha B^{-1}A \, \delta_\xi] \, (\vec{R}_{k+1} - \vec{R}_k) = B_k^{-1}(\alpha V^\circ_{k+1} + (1-\alpha)V_k) + \varepsilon_e (\nabla_j \Delta_j)^2 \vec{R}_k$$

Adding a second order implicit dissipation term to this algorithm in delta form serves to augment the amount of explicit dissipation that may be added, while not formally degrading the accuracy of the method. Therefore the final form of the algorithm is:

$$[1 + \varepsilon_i(\nabla_j \Delta_j) + \alpha B^{-1}A \, \delta_\xi] \, (\vec{R}_{k+1} - \vec{R}_k) = B^{-1} (\alpha V^\circ_{k+1} + (1-\alpha)V_k) + \xi_e (\nabla_j \Delta_j)^2 \vec{R}_k$$

The marching algorithm resulting from the hyperbolic equations provides an efficient means for constructing a computational grid. Figure 2 is a flow chart of the computer program based on this algorithm. The program begins by initializing the variables and reading in the user defined body coordinates. The marching procedure loop consists of four steps. First, the areas at the k+1 level are approximated. Then a system of equations is set up to determine the increments in X and Y. This results in a 2x2 block tridiagonal matrix which is solved. The increments in X and Y are now known at the k level for each j station, and the grid is computed at the k+1 level using these increments. The process is repeated until the KMAX level is reached.

Table I is a listing of the subroutines used in a computer program that utilizes the algorithm described above. A brief description of the purpose of each subroutine is also provided.

FLOW DIAGRAM

Control Parameters

Initialization
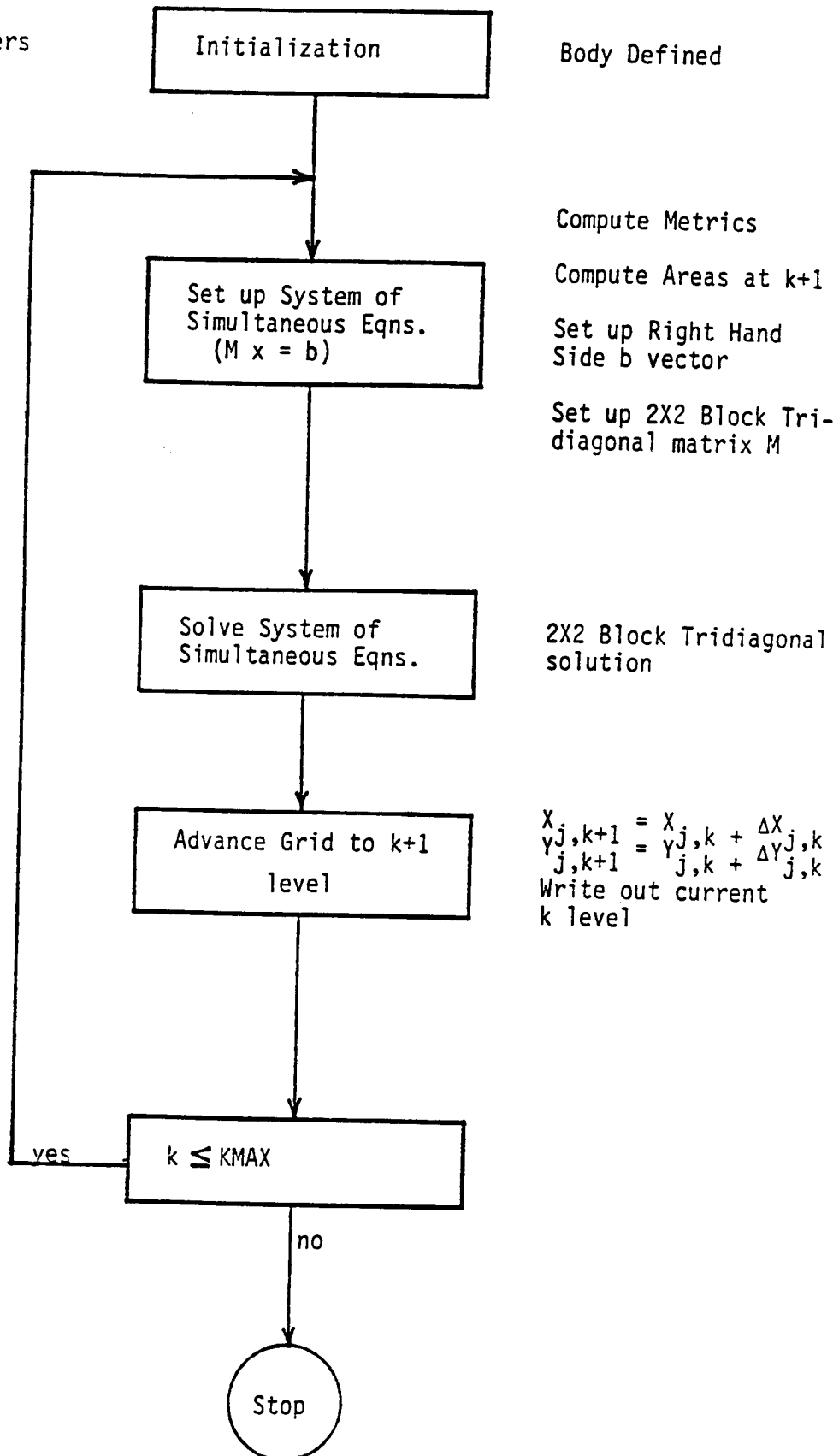
Body Defined

Set up System of
Simultaneous Eqns.
(M x = b)

Compute Metrics

Compute Areas at k+1

Set up Right Hand
Side b vector

Set up 2X2 Block Tri-
diagonal matrix M

Solve System of
Simultaneous Eqns.

2X2 Block Tridiagonal
solution

Advance Grid to k+1
level

$$X_{j,k+1} = X_{j,k} + \Delta X_{j,k}$$
$$Y_{j,k+1} = Y_{j,k} + \Delta Y_{j,k}$$
Write out current
k level

$k \leq$ KMAX

yes

no

Stop

FIGURE 2

TABLE I

SUBROUTINE DEFINITIONS

SUBROUTINE	CALLED FROM	CALLS TO	PURPOSE
MAIN	-	INITIA STEP OUTPUT	PROGRAM CONTROL
INITIA	MAIN	BODIS SARC EPSIL	READS INPUT, DEFINES INITIAL GRID SPACING
BODIS	INITIA	-	READS BODY COORDINATES
SARC	INITIA	-	COMPUTES BODY PERIMETER
EPSIL	INITIA	-	DEFINES DISTRIBUTION
STEP	MAIN	METRIC RHS FILTRY BTRI	MARCHES GRID GENERATOR FROM BODY TO OUTER BOUNDARY
METRIC	STEP	-	
RHS	STEP	-	CALCULATES THE FORCING FUNCTION (RIGHT HAND SIDE)
FILTRY	STEP	GMATRX	FILLS THE TRIDIAGONAL MATRIX
GMATRX	FILTRY	-	DEFINES G MATRIX
BTRI	STEP	LUDEC	BLOCK TRIDIAGONAL SOLUTION
LUDEC	BTRIP	-	COMPUTES L-U
OUTPUT	MAIN	-	WRITES DESIRED INFORMATION TO OUTPUT
BC	STEP	-	ROUTINE ADDED TO C-GRID TO DEFINE AFT BOUNDARY CONDITIONS.

III. PROGRAM RESULTS

Two computer programs were developed using the algorithm described in the proceeding section. One program produces an O-type computational grid and the other provides C-type grids. Results from the O-type grid will be shown first, followed by a discussion of the modification necessary to produce a C-type grid and some example results. A step by step examination of the effects of the various input parameter options is provided in the next section.

Figure 3 shows a far-field and near-field view of one of the first attempts at fitting an O-grid around a NLR 7301 airfoil section.[4] This shape represents a relatively simple test condition except for the region of the trailing edge. Careful selection of the points used to define the geometry at the trailing edge can significantly improve the results. Figure 4 is an example of an O-type grid around a square. Figure 4a has equally distributed body geometry points, whereas, Figure 4b shows the results of clustered and stretched body definition points. Figures 5 and 6 indicate the O-type grid results for a complicated geometry. The geometries shown provide a very severe test for any grid generation scheme and is particulary difficult for a marching (hyperbolic) procedure.

O-type computational grids are widely used for a variety of solution procedures. For some applications, however, a C-type grid is preferred. For example, an attempt to determine the viscous flow characteristics around an airfoil section and the resulting wake area behind the airfoil, require fine grid spacings close to the body surface and in the wake. A C-type grid can easily provide this kind of grid distribution.

The most obvious change required to convert from an O-grid to a C-grid is the establishment of a downstream boundary condition. The boundary conditions for a C-grid, however, are only slightly more difficult than for an O-grid. A
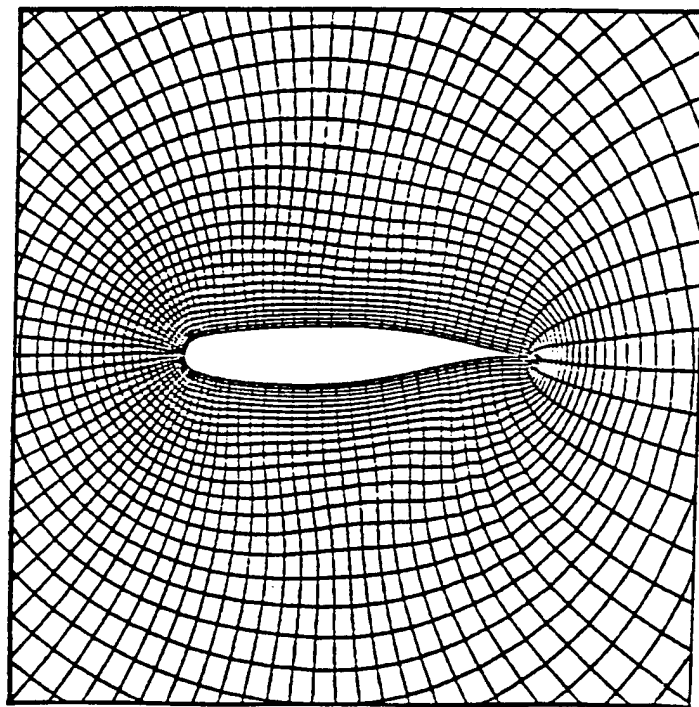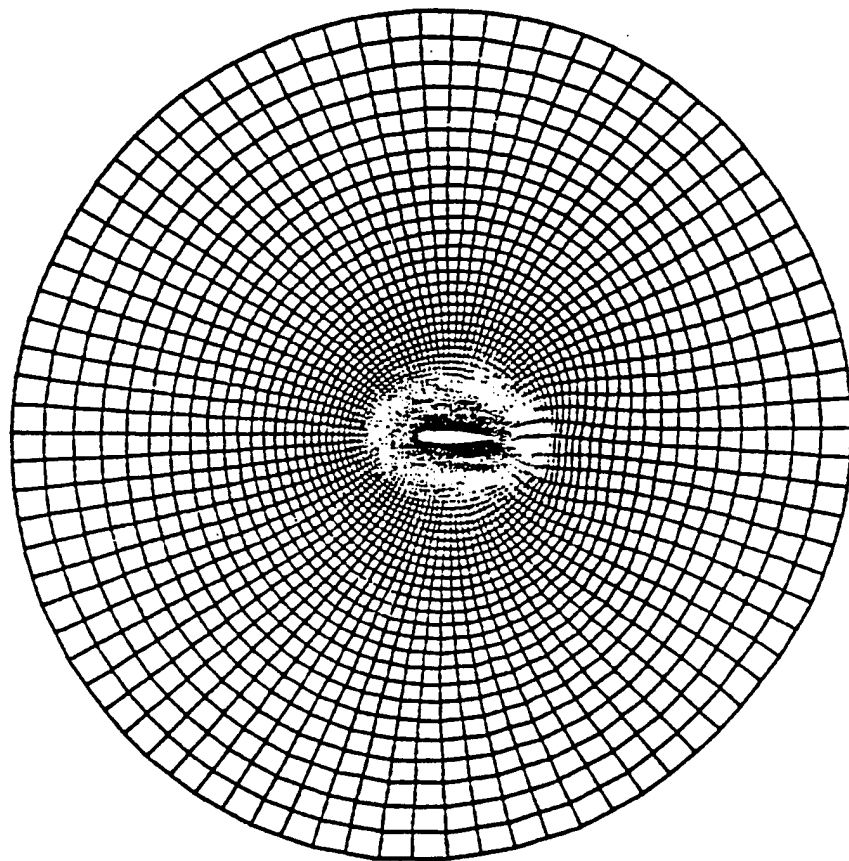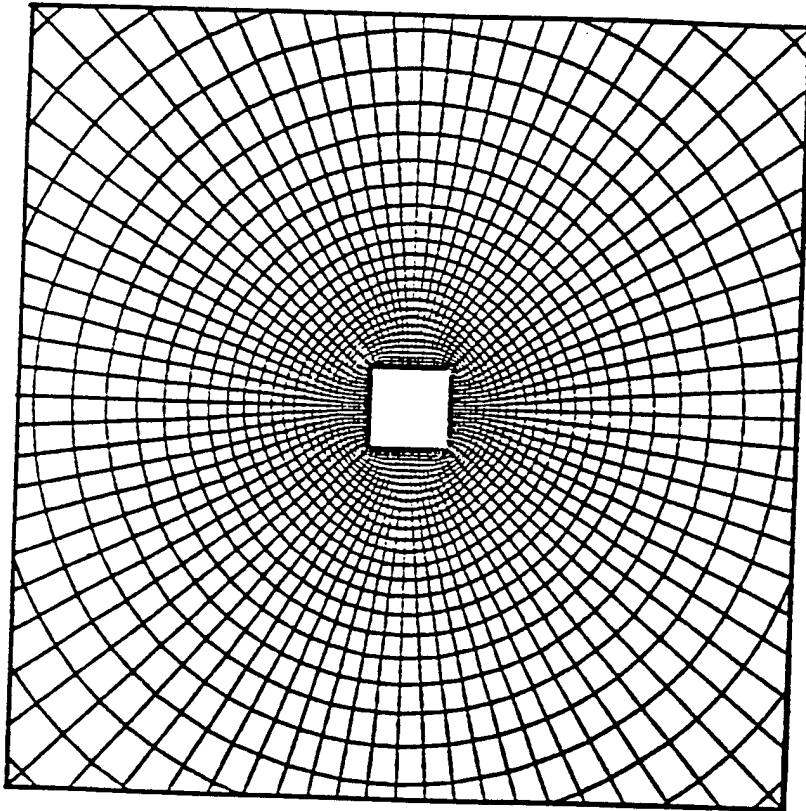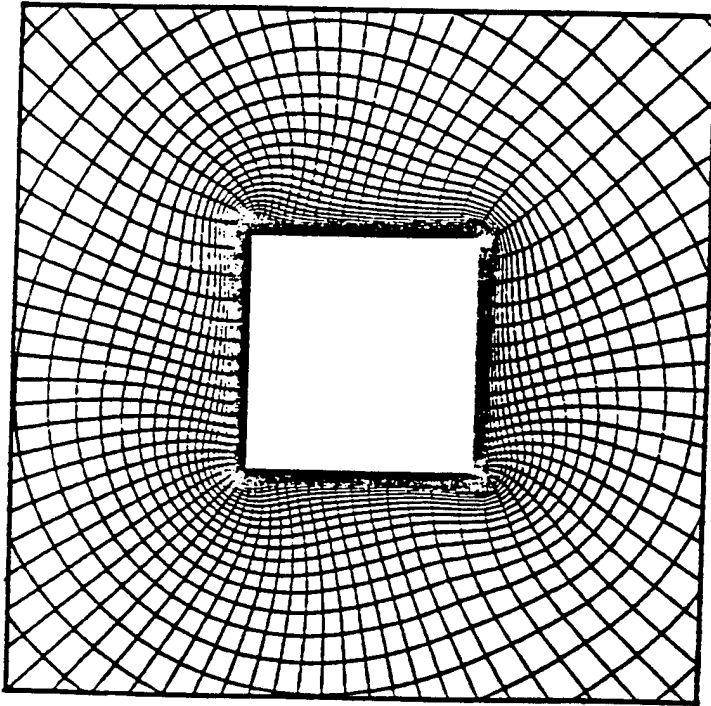
11

FIGURE 3

O-GRID ON A SQUARE
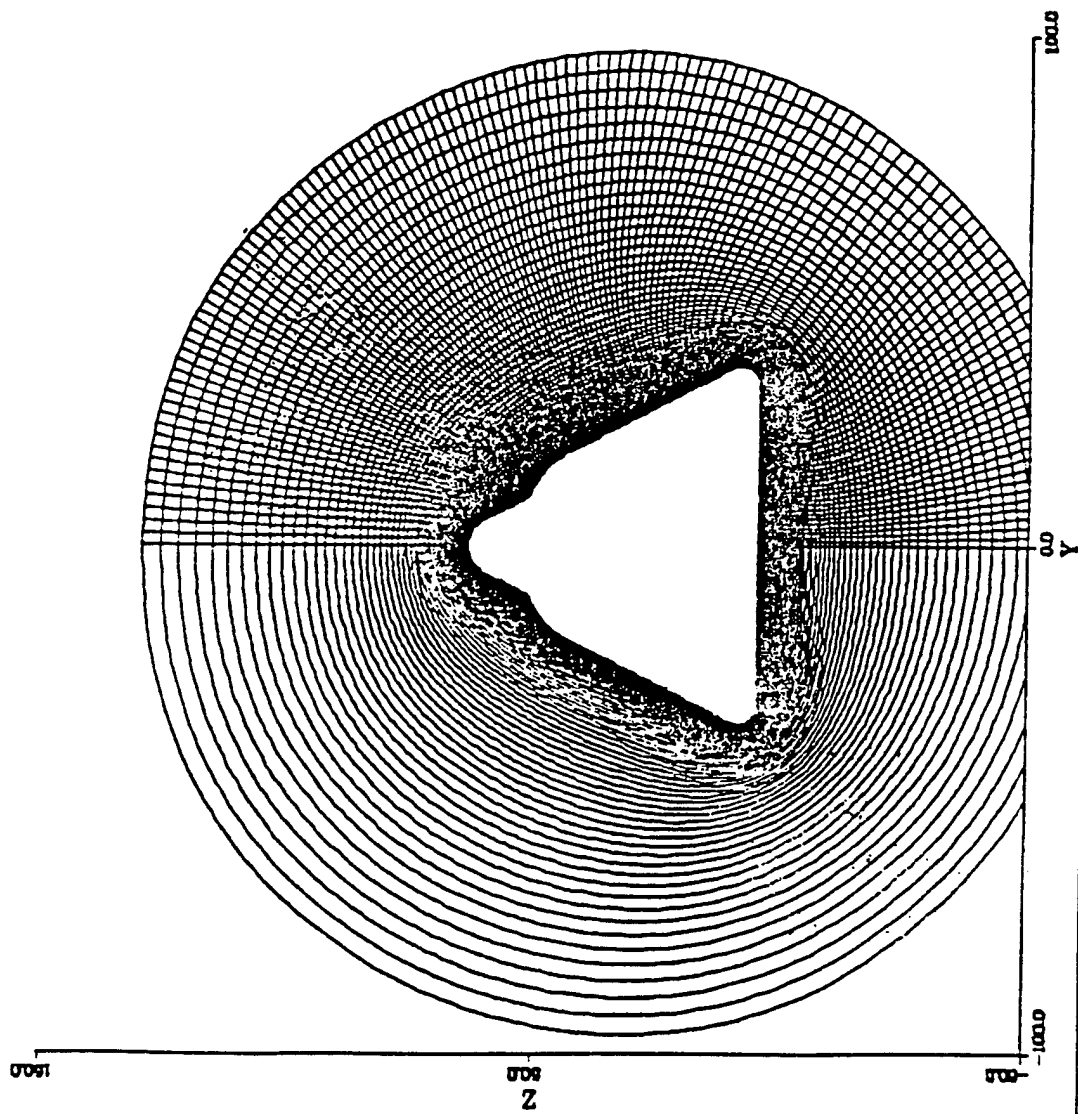

a

b

FIGURE 4

13

GRID

X24C-10D..SECTION 15B

FIGURE 5

14

GRID

X24C-10D_SECTION 560

GRID

X24C-10D_SECTION 560
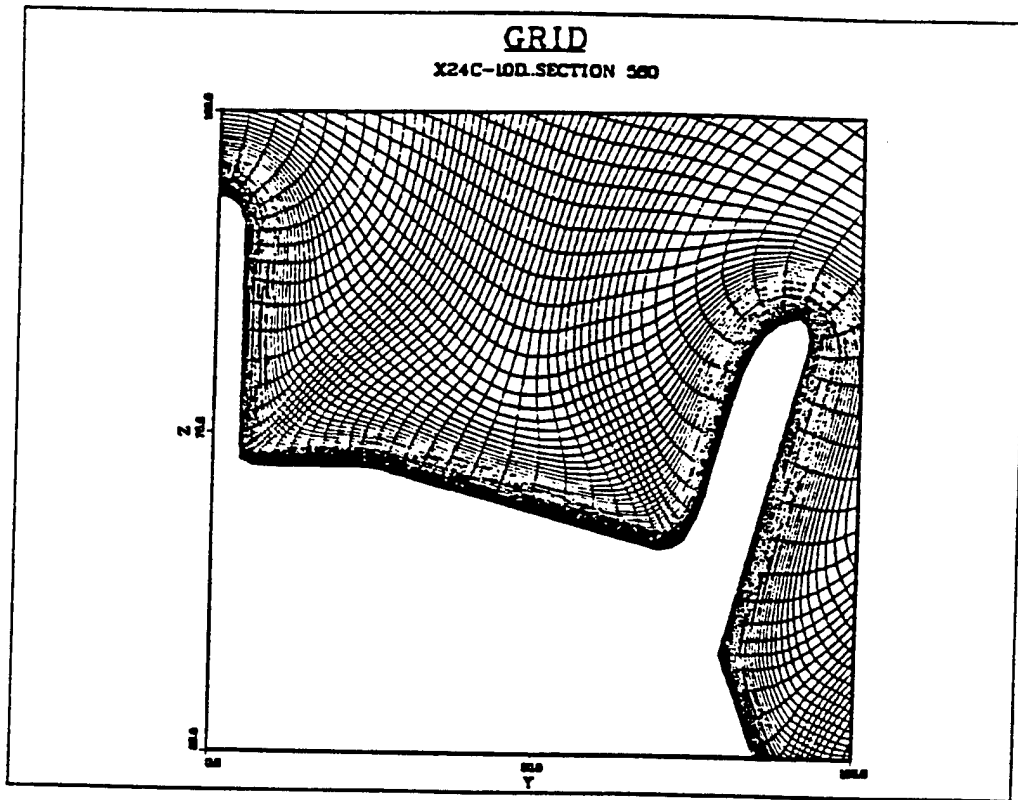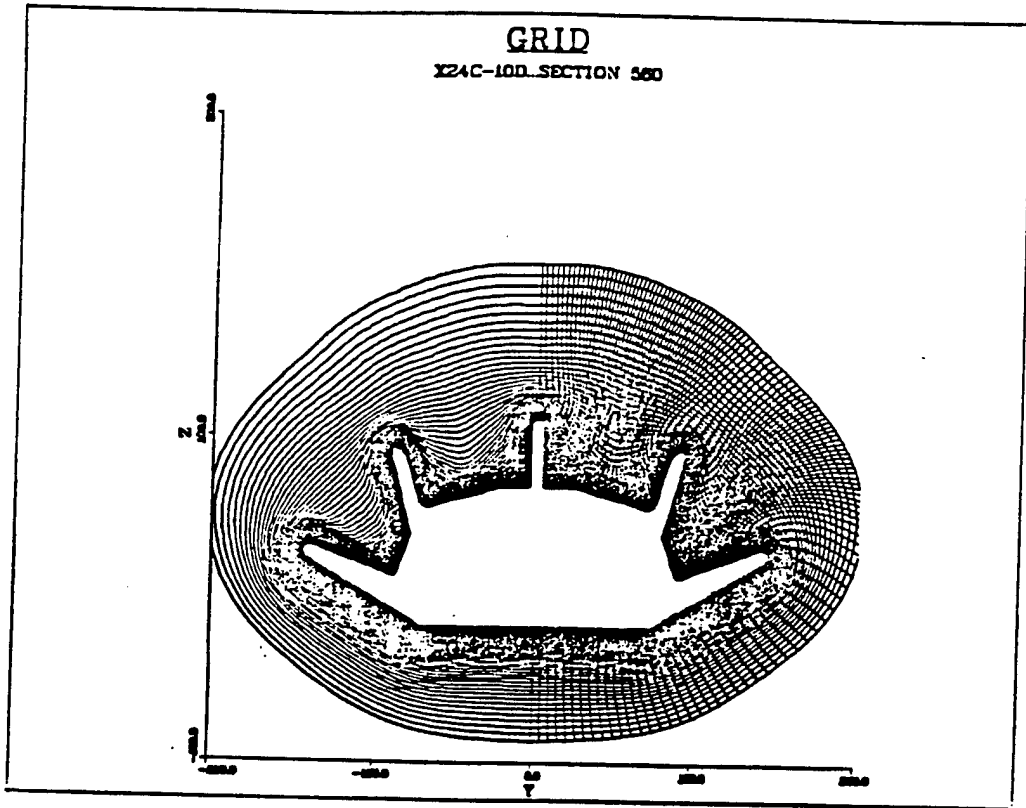
FIGURE 6

separate subroutine (BC) was provided to compute the downstream boundary conditions, but essentially X is fixed at XMAX and Y is incremented as ± ΔR.

The C-type grid algorithm alters the form of the tridiagonal matrix slightly from the form required by the O-grid procedure. Figure 7 shows the general arrangement of the tridiagonal matrix used. Figure 8 shows a typical C-grid for the NLR 7301 airfoil. Many examples of a C-type grid around the NLR 7301 airfoil are provided in the next section which includes a discussion of the effects of the various user definable parameters.
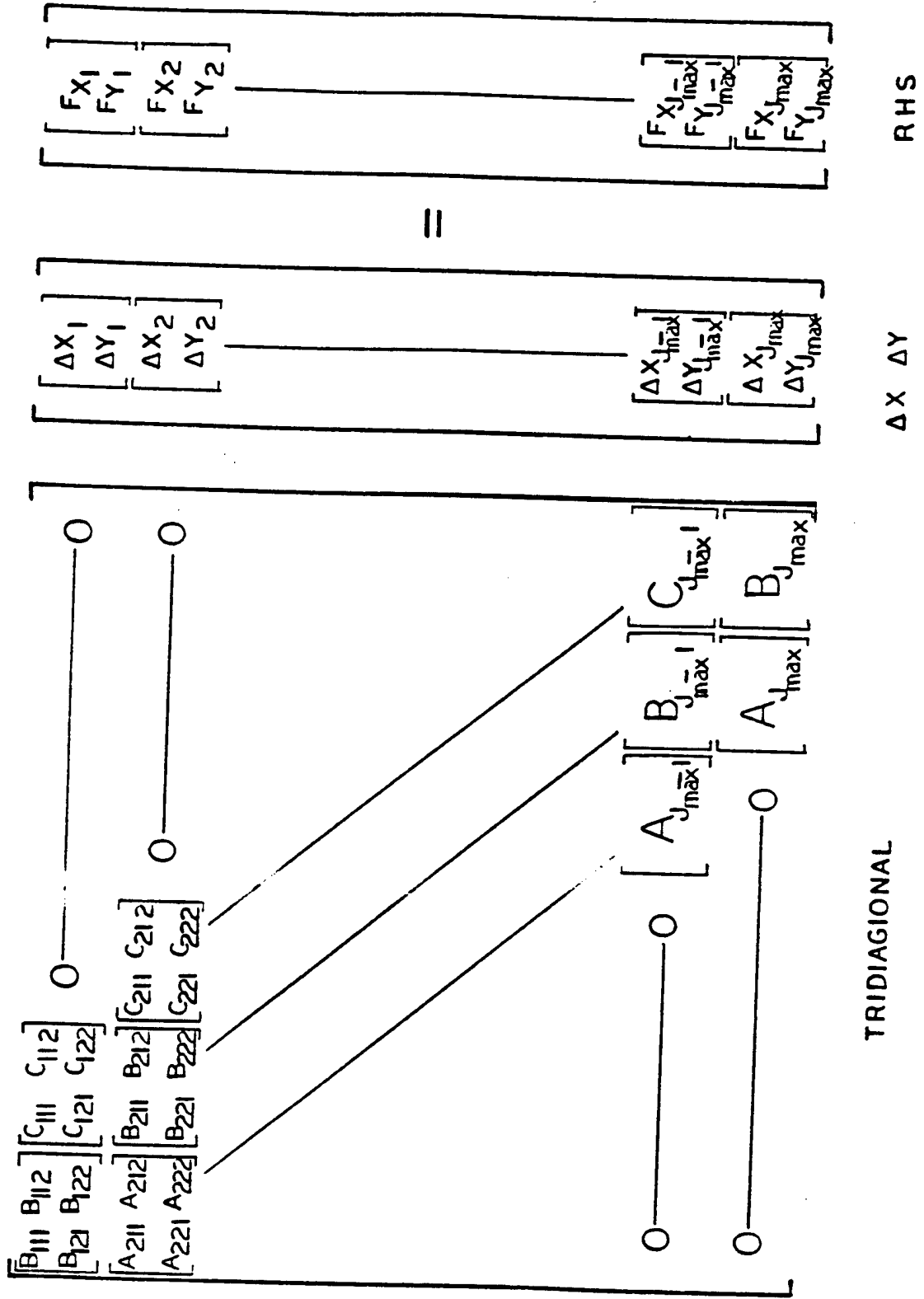
TRIDIAGIONAL EQUATION
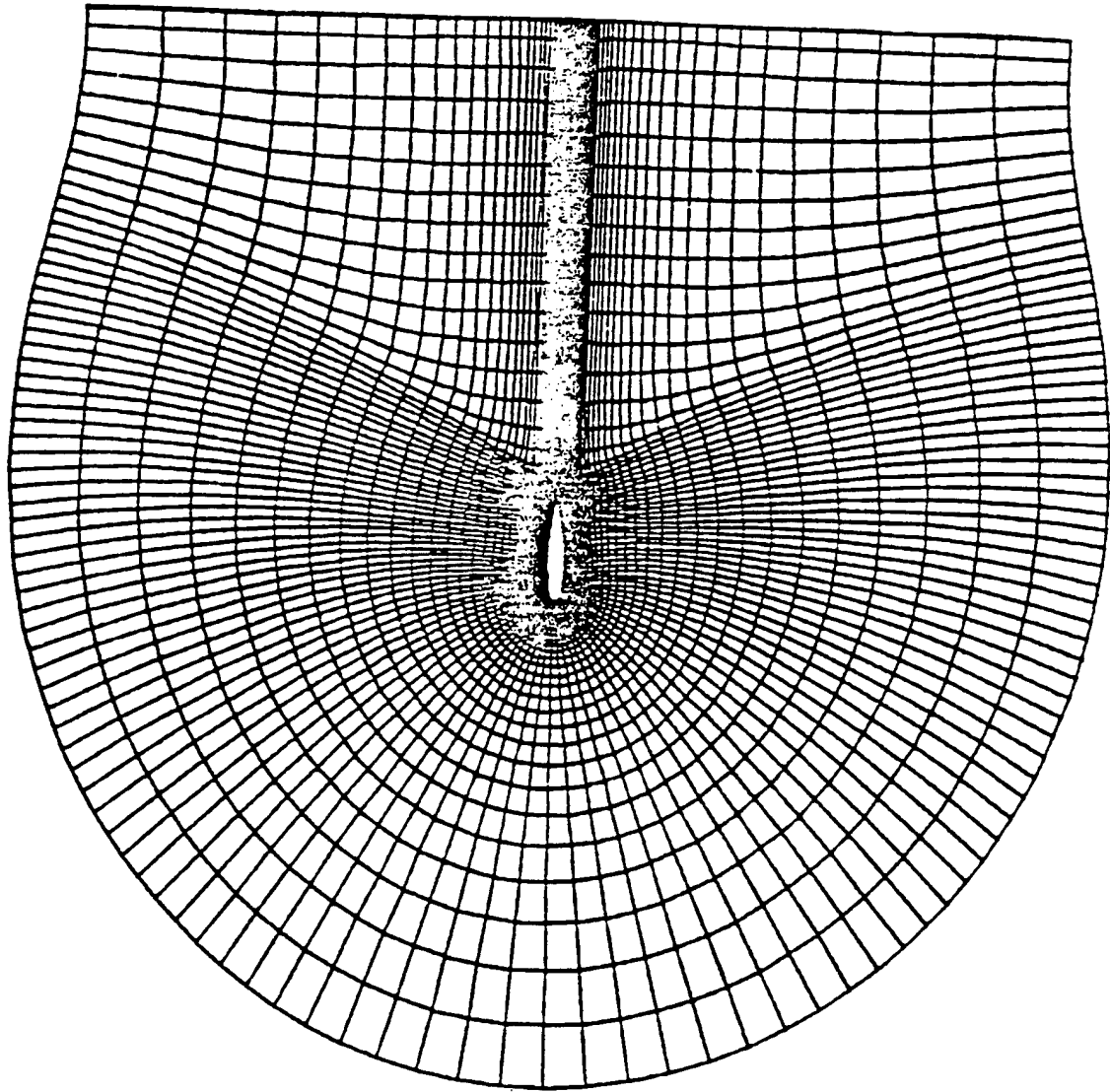
FIGURE 7

17

TYPICAL C-GRID ON NLR 7301 AIRFOIL

18

FIGURE 8

IV. EFFECT OF USER DEFINABLE PARAMETERS

It is appropriate to begin this section with a discussion of how the "body" is defined for the grid generating routines. In addition to the actual body, as defined for the 0-grid routine, a zero thickness extension of the body from the trailing edge to the downstream boundary is required for the C-grid routine. The complete "body" (actual body plus zero thickness extension) is input in the same way as the real body was input to the 0-grid procedure. That is, the body geometry is input as JMAX Cartesian coordinates, starting at XMAX and proceeding around the body in a clockwise direction. A more aesthetically pleasing grid is obtained if the upper and lower (X, Y) locations for the wake extension are equal.

This section includes a description and discussion of the effects of varying the user defined input variables. It is necessary to define a baseline condition from which variations of individual parameters will be discussed. Table II defines the baseline parameters used to determine a grid for the NLR 7301 airfoil section which will be used for most of the examples of this section. Figure 9 shows a far field and a near field view of the grid produced using these baseline parameters. The variables in Table II will each be discussed in turn.

JMAX defines the total number of points used to describe the body. Probably more important than the total number of points is the distribution of these points. In a practical situation both the maximum number (JMAX) and the distribution will be governed by the body geometry and the finite difference program that will be using the grid. Figure 10 shows the baseline configuration (JMAX = 100), and grids with 25% fewer (JMAX = 75) and 50% more (JMAX = 150) points. The distribution of the points around the body is approximately proportional in all three cases.

TABLE II

BASELINE PARAMETER SETTINGS

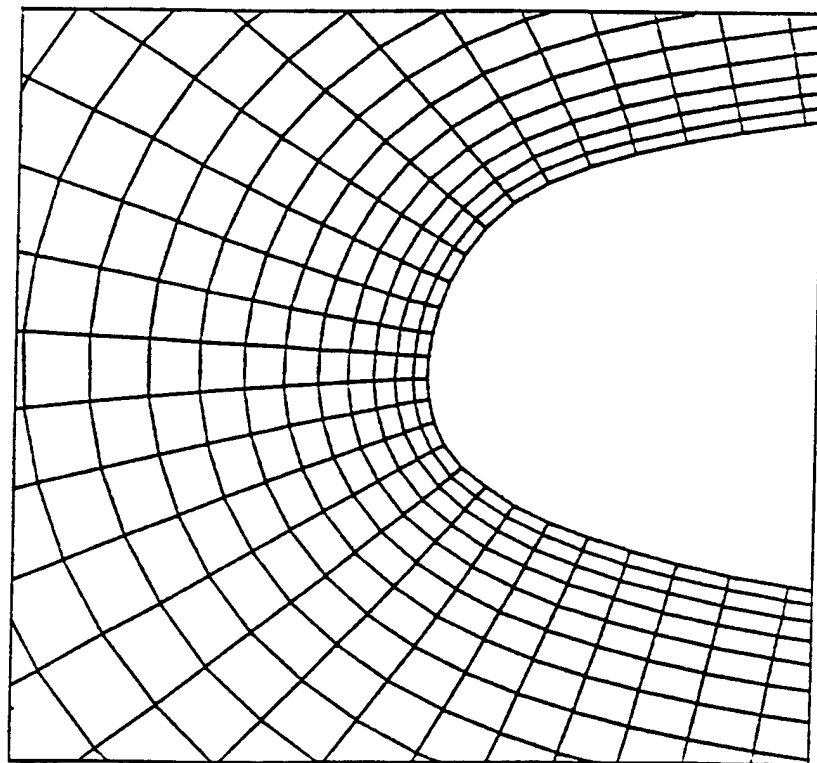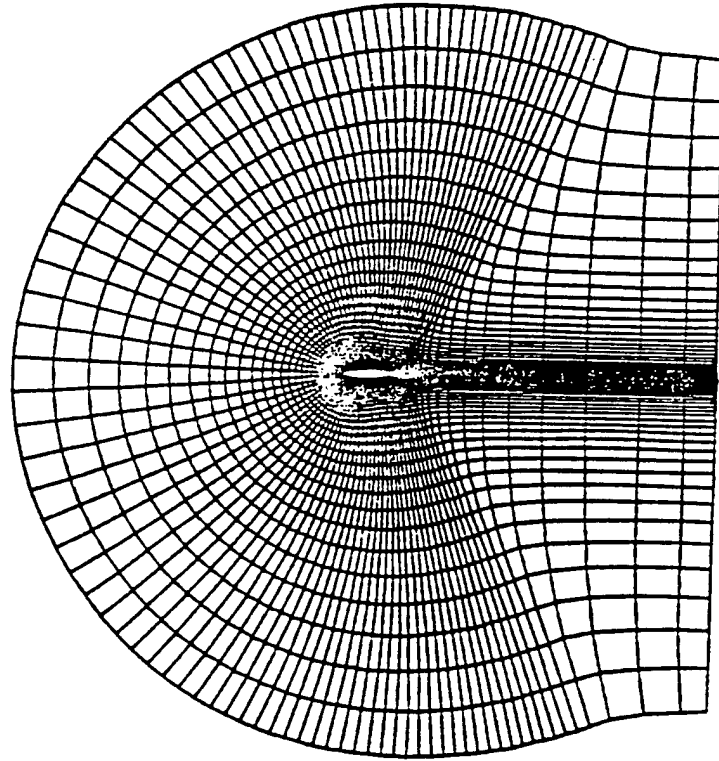| PARAMETER	VALUE	REASON
JMAX	100	Number of coordinates for body.
KMAX	40	Typical value.
DSETA	.004	Provides approximately 5 grid lines in a turbulent boundary layer with R_L = 10 million.
SETMX	6	Typical value.
ESCAL	.005	Nominal value.
SMU	.1	Nominal value.
SMUIM	.5	Nominal value.
ALPHA	1.0	Implicit finite difference of order 1.

FIGURE 9

VARIATION OF JMAX

K = 40   DSETA = .004   SETAMX = 6.   ESCAL = .005   SMU = 0.1
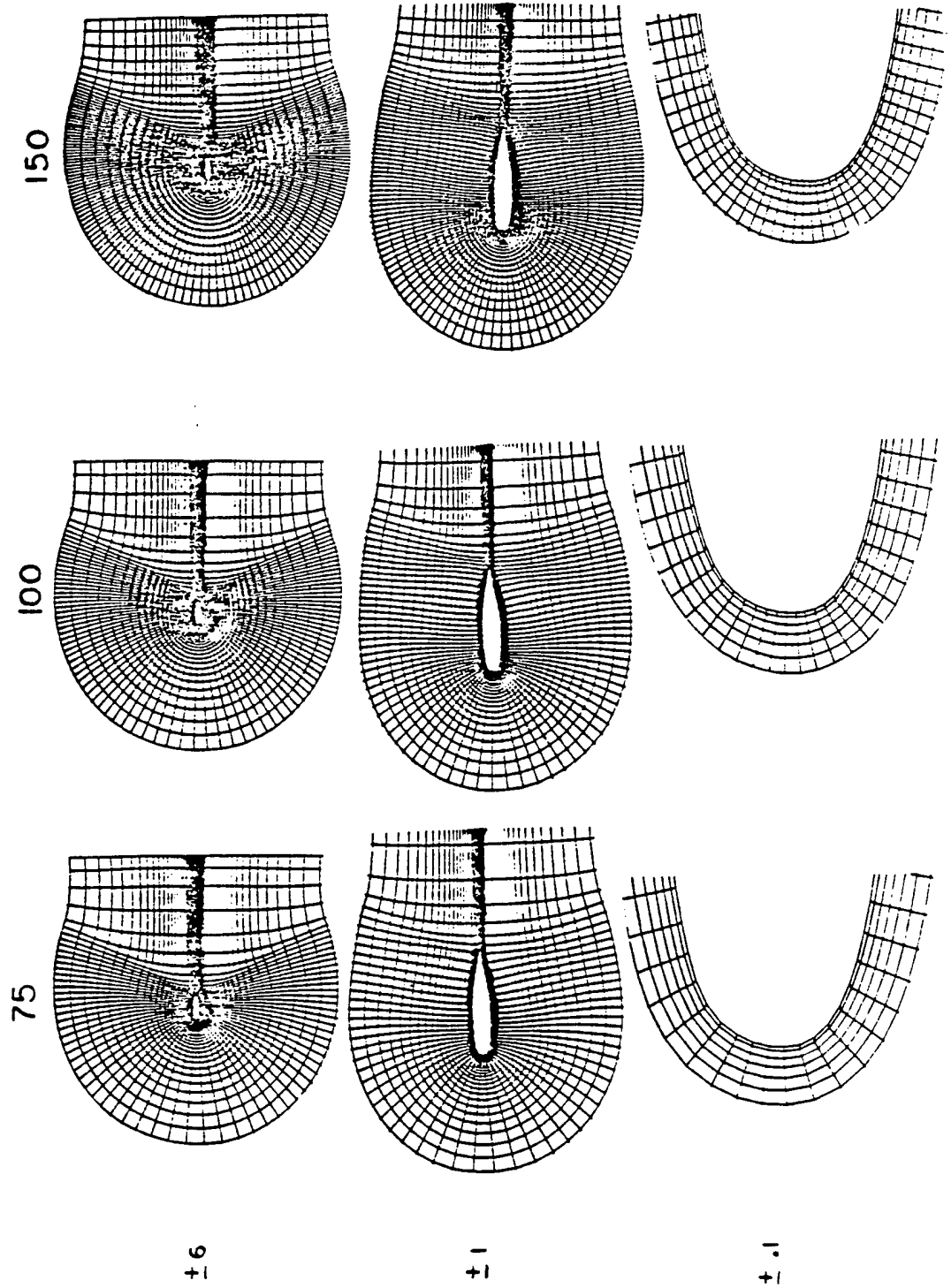
SMUIM = 0.5   ALPHA = 1.0

FIGURE 10

22

KMAX defines the maximum number of grid levels or contours. Note that the body geometry definition constitutes the first physical contour (K = 1). Each succeeding contour is determined by marching radially from the previous contour. The initial cell height in the radial (η) direction is also approximately specified by the user (DSETA). The program uses a Newton-Raphson routine to determine the values of an exponential stretching function (ε) that will smoothly vary from the initial increment (DSETA) at the body to the location of the outer boundary (SETAMX) in KMAX steps:

$$R_k = R_{k+1} + DSETA\ (1+ \varepsilon)^{k-2}$$

Figure 11 shows a typical variation of Δ R with distance from the body. With JMAX, SETAMX and DSETA fixed, the aspect ratio (length to width) of the grid cells away from the body are governed by KMAX. Figure 12 shows the effect of reducing the value of KMAX by 50% (KMAX = 20) and of increasing the value of KMAX by 50% (KMAX = 60) over the baseline value. The aspect ratio of the cells and the rate of transition to equal area cells are both significantly influenced by the value of KMAX.

DSETA defines the initial cell height normal to the body surface. The proper magnitude for DSETA depends on the purpose for which the grid is being created. For example, if a detailed analysis of the flow conditions in the boundary layer of a high Reynolds number flow is desired, several grid levels will be needed within this thin layer near the body. On the other hand, if the flow is assumed to be inviscid (no boundary layer) the grid distribution near the body may be governed by other factors, such as cell aspect ratio. If the total number of contours (KMAX) and the outer boundary location (SETAMX) are fixed, then obviously having more contours near the body requires a wider

GRID DISTRIBUTION
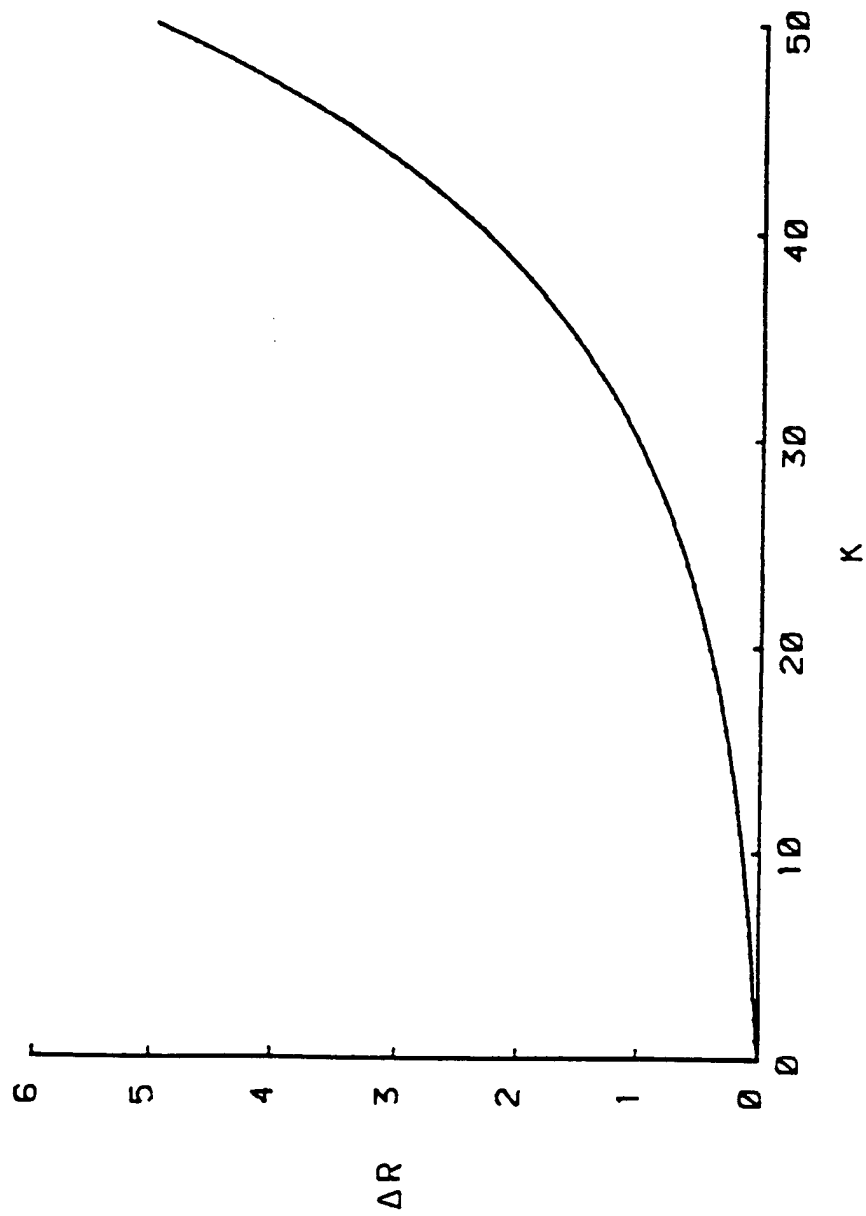
KMAX=50 SETAMX=5 DSETA=.01

FIGURE II

24

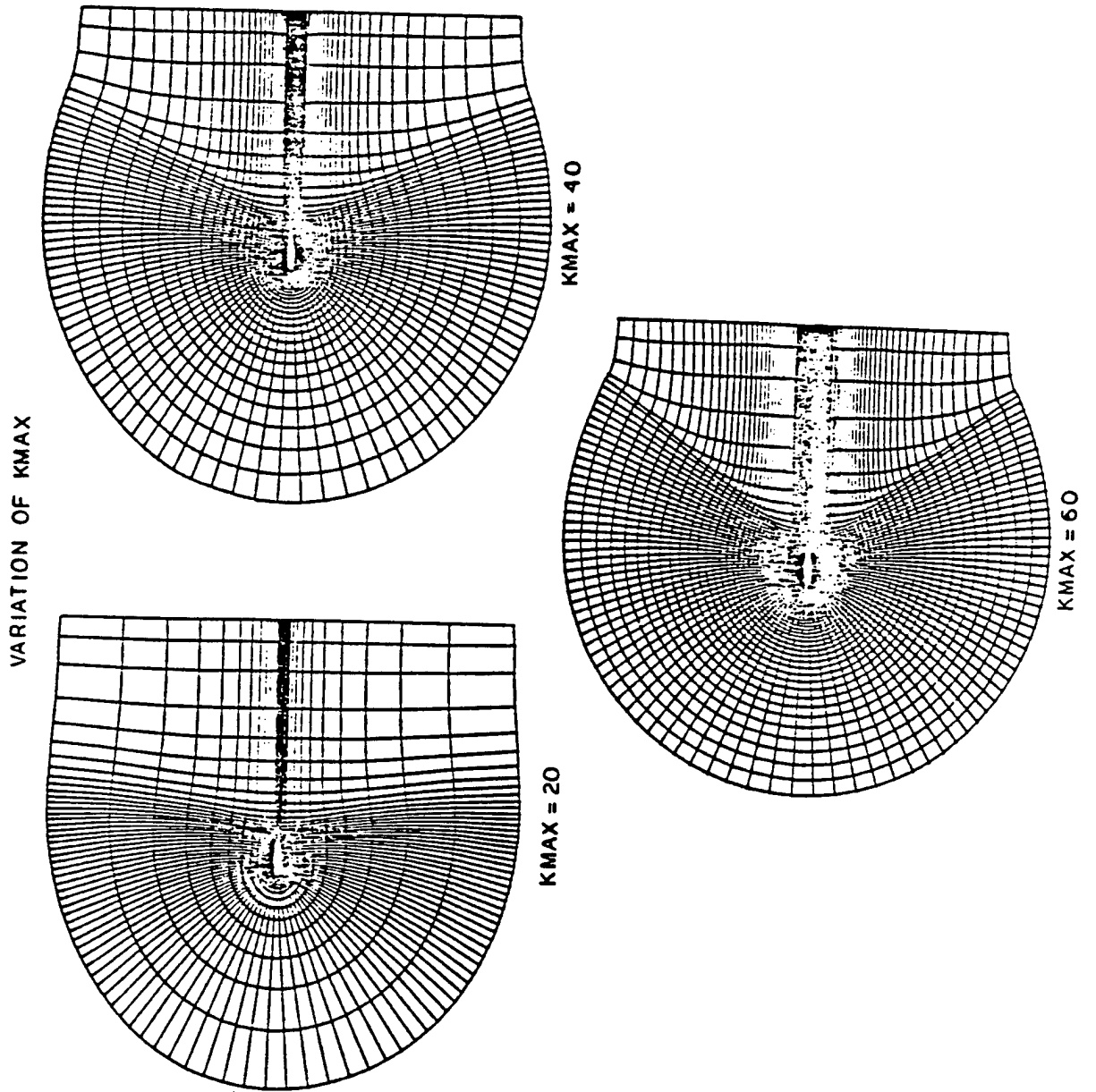VARIATION OF KMAX

KMAX = 40

KMAX = 60

KMAX = 20

FIGURE 12

spacing away from the body. Figure 13 shows the variation in grid spacing for DSETA = .001 and .01 compared to the baseline value of .004.

The next parameter in the list, SETAMX, defines the approximate location of the outer boundary for the grid. For the baseline airfoil it defines (approximately) the number of chord lengths away from the body for the outer most grid level (KMAX). Figure 14 shows a comparison of SETAMX = 3, 6 and 10 with all other parameters at baseline values. Note that the downstream boundary has been maintained at six chord lengths for this comparison. A practical grid would probably have the downstream boundary approximately the same distance from the body as the other far field boundaries.

ESCAL defines the rate at which the grid distribution around the body will transition from the input coordinate distribution to a grid with equal cell areas for a given radial location. The scale factor used to transition to equal cell areas is of the form:

$$Scale_k = (1. - ESCAL)^{k-2} \qquad\qquad k = 2, 3, 4.....KMAX$$

Figure 15 shows how the value of ESCAL influences this scale factor. ESCAL = 0 means there will be no tendency for the grid cells to seek equal areas with increasing distance from the body. In this case the radial lines tend to march out from the body with the same distribution as the input data; however, the procedure will not allow radial lines to coalesce or cross. Figure 16 compares results for ESCAL = 0. and 0.025. Notice that the ESCAL = 0.025 grid has very undesirable characteristics along a line of "disturbance" from the trailing edge. Additional problems results when the disturbance reaches the rear boundary of the C-grid routine. The bad grid points occur between regions of relatively dense grid spacing and regions of sparse grid spacing. Some relief for this problem can be obtained with the following smoothing parameters.
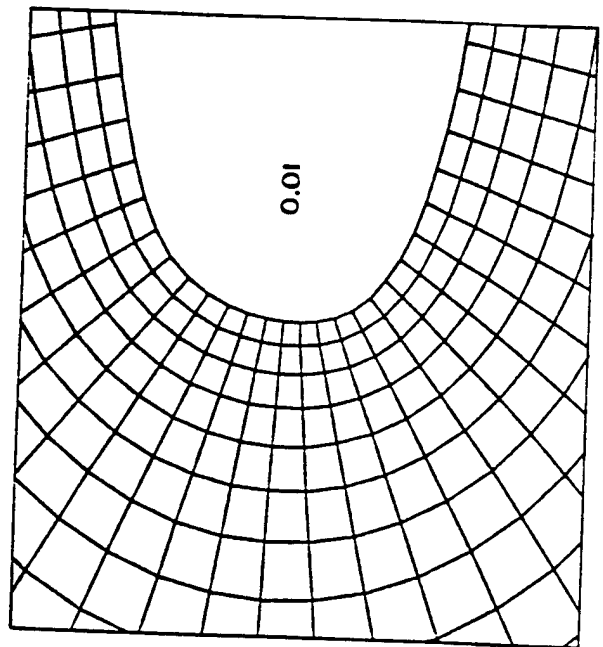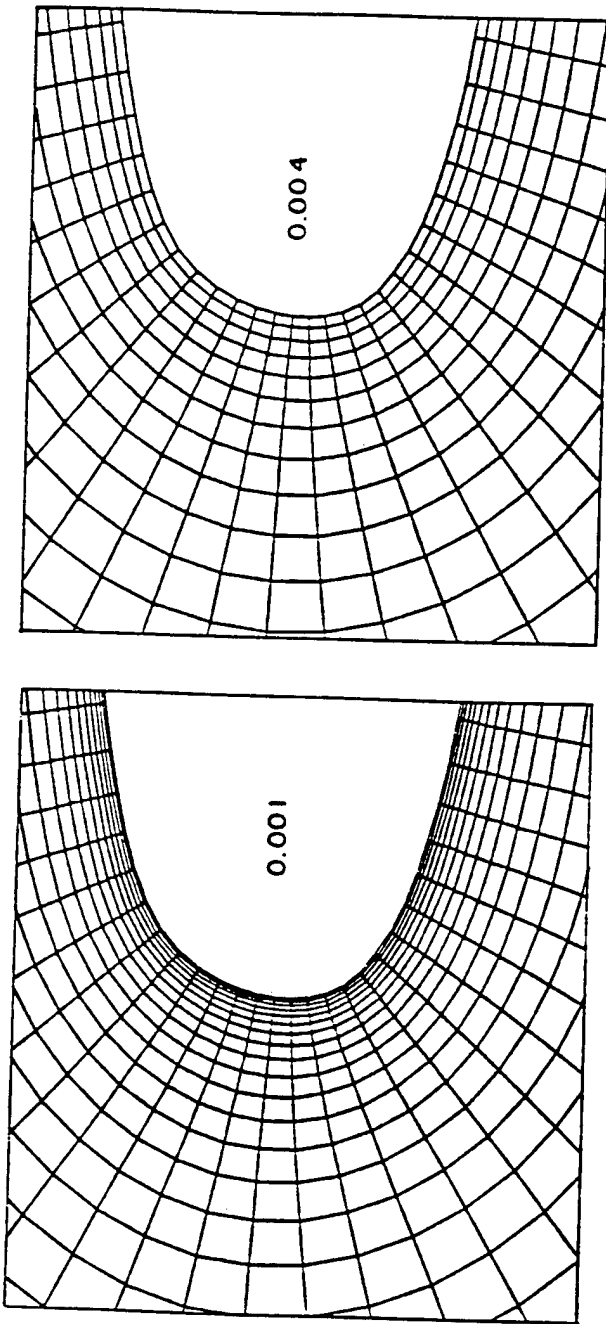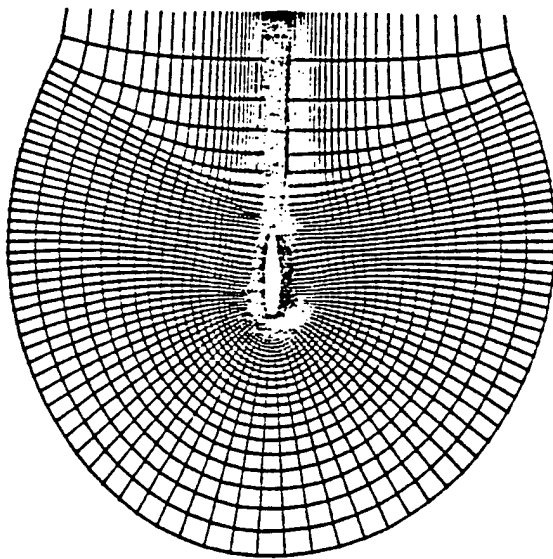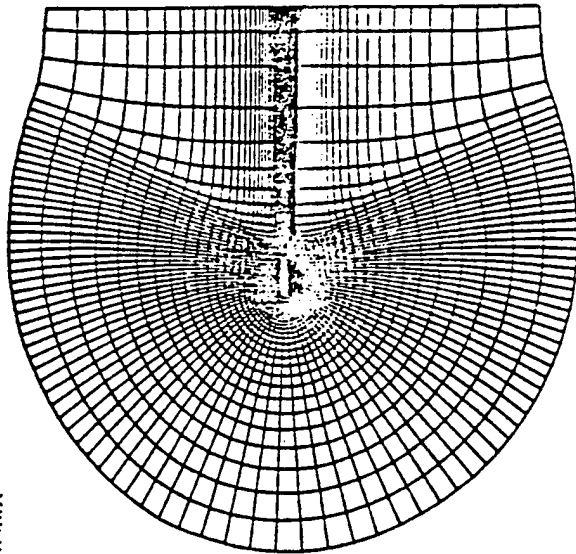
VARIATION OF DSETA

0.004

0.01

0.001

FIGURE 13

27
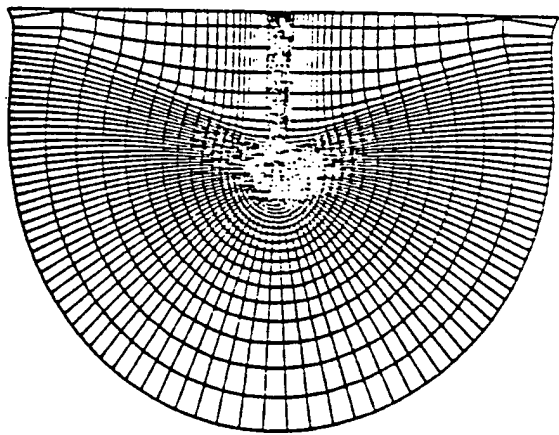
VARIATION OF SETAMX

SETAMX = 6

SE TAMX = 10

SETAMX = 3

FIGURE 14

EFFECT OF ESCAL PARAMETER

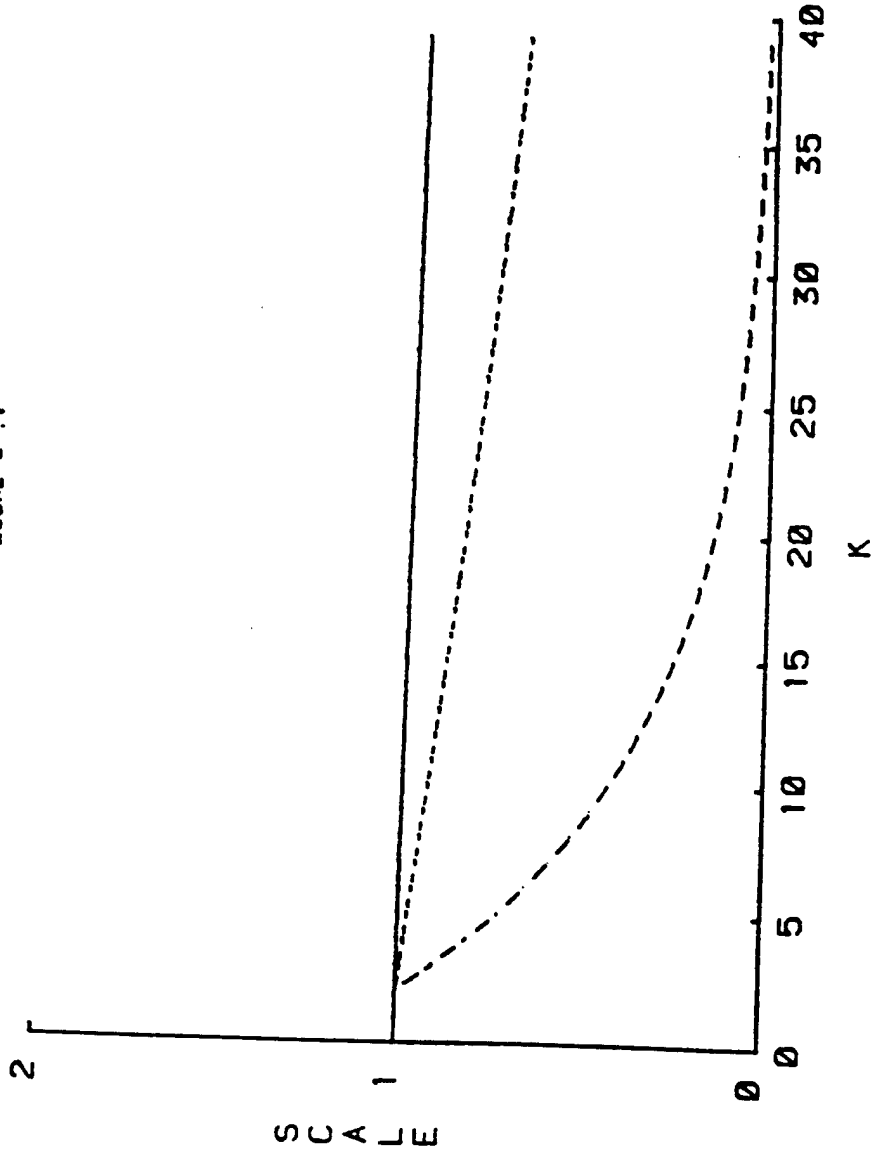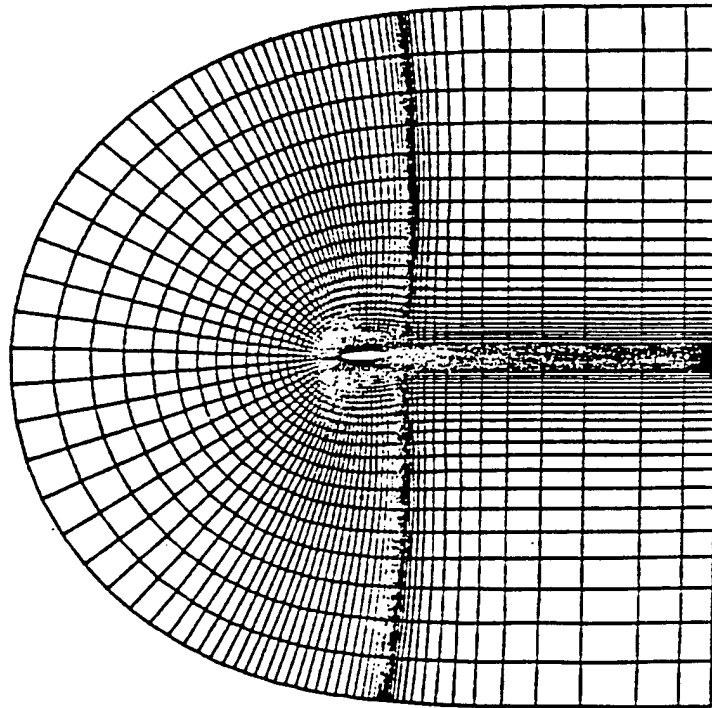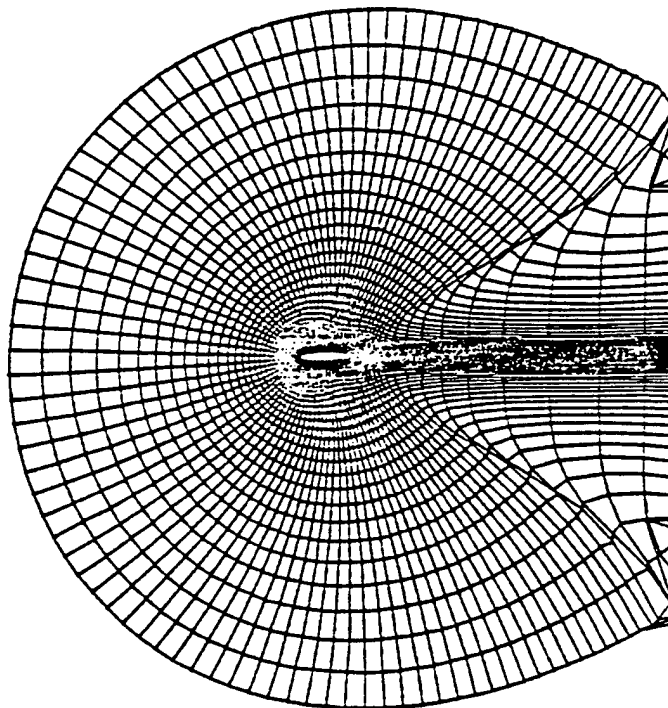ESCAL = .001
ESCAL = .01
ESCAL = .1

FIGURE 15

ESCAL = 0.0

ESCAL = 0.025

FIGURE 16   30

The next input parameter for the routine is an explicit smoothing parameter (SMU). This parameter defines the amount of fourth order dissipation (smoothing) to be used to damp numerical oscillations. An explanation of why the smoothing is needed and how it is implemented is given in Section II and Reference (1).

SMU has an adverse effect on orthogonality and too large a value for SMU could cause the numerical procedure to become unstable. The value of SMU input represents the maximum amount of smoothing for any grid level. The actual value used in the program varies from zero at the body to this maximum value far from the body at a rate equal to the rate of transition to equal cell areas.

$$SMU_K = SMU_{INPUT} \ (1 - SCALE_K)$$

For the baseline parameters, variations of SMU seem to have very little effect on the grid produced until values of approximately 4.5 are input. Figure 17 demonstrates this by showing grids for SMU = 0.0, 0.5 and 4.5. At SMU = 4.5, very obvious instabilities are occurring at grid points far from the body. It is important to remember that the value for ESCAL, SMUIM and the input geometry definition can all have a strong influence on how SMU affects the grid.

SMUIM is another smoothing parameter. This implicit smoothing is not as effective as the explicit smoothing (SMU) but it will not cause the numerical instability that large values of SMU does. Implicit smoothing has the dual effect of adding some higher order smoothing itself plus increasing the amount of explicit smoothing that can be added before the procedure becomes unstable.

VARIATION OF SMU
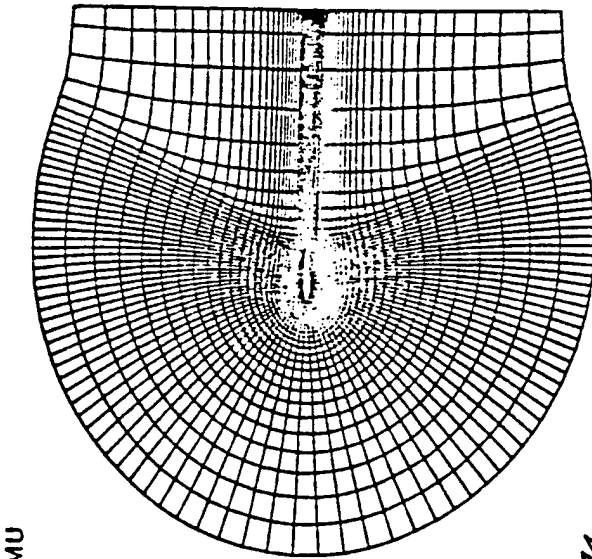
SMU = 0.5
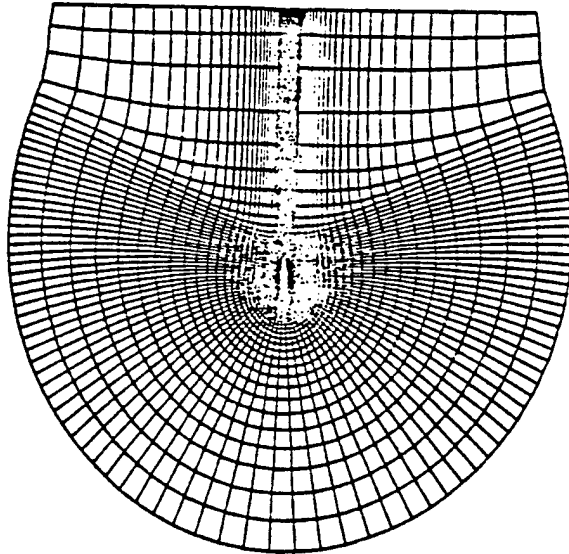
SMU = 4.5

SMU = 0

FIGURE 17

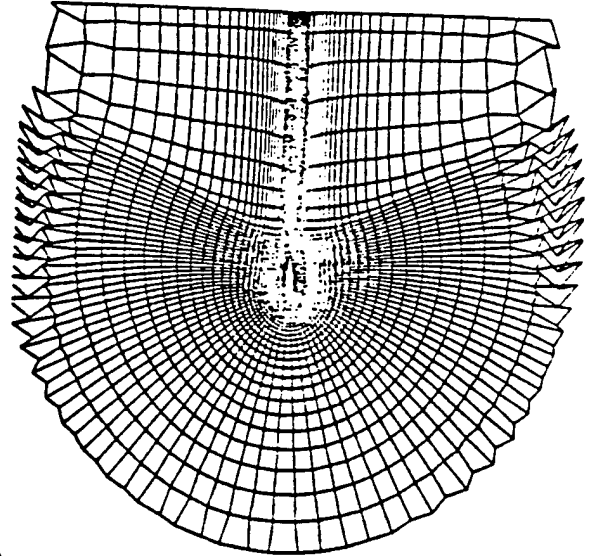Although there is no theoretical stability limit to the amount of implicit smoothing (SMUIM) that can be added, there is a practical limit because both orthogonality and accuracy of the grid are affected. Figure 18 shows the effect of SMUIM on orthogonality. It is obvious from Figure 18 that large values of SMUIM have a significant effect on the propagation of disturbances throughout the grid.

The ALPHA parameter controls the nature of the finite difference marching algorithm used to march the grid from the body to the outer boundary (see Section II). A value of ALPHA greater than 1 tends to weight the procedure in favor of the implicit method and has the result of improving the smoothness of the grid. The trade-off is between the accuracy (and improved orthogonality) provided by the ALPHA = ½ choice and the "robustness" of the ALPHA greater than 1 choice. The grid characteristics required by the finite difference program for which the grid is intended and the shape of the body about which the grid is produced will probably determine what value of ALPHA is appropriate. The payoff from using large values of ALPHA are most evident on complex bodies. Figure 19 shows the effect of ALPHA on the grid generated about sections of the X-24C. For the baseline parameters on the airfoil section, the effect of changing ALPHA from 0.5 to 2.0 is quite small (see Figure 20). There is some stretching of the grid in the radial direction as ALPHA gets larger. Very large values of ALPHA (up to 10) produce significant stretching in the radial direction and causes some loss of orthogonality near the body.

VARIATION OF SMUIM

SMUIM = 50.

SMUIM = 50

SMUIM = O.

FIGURE 18

34
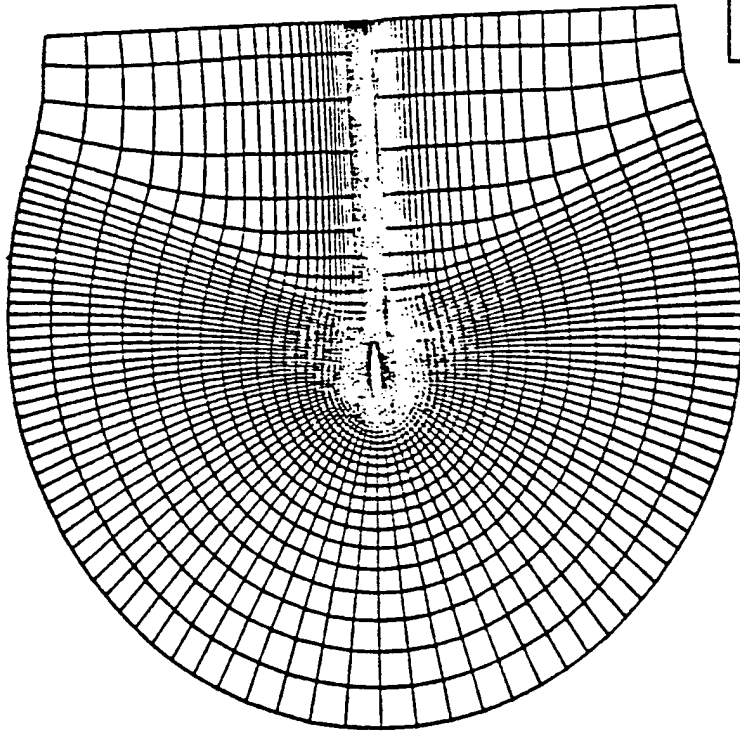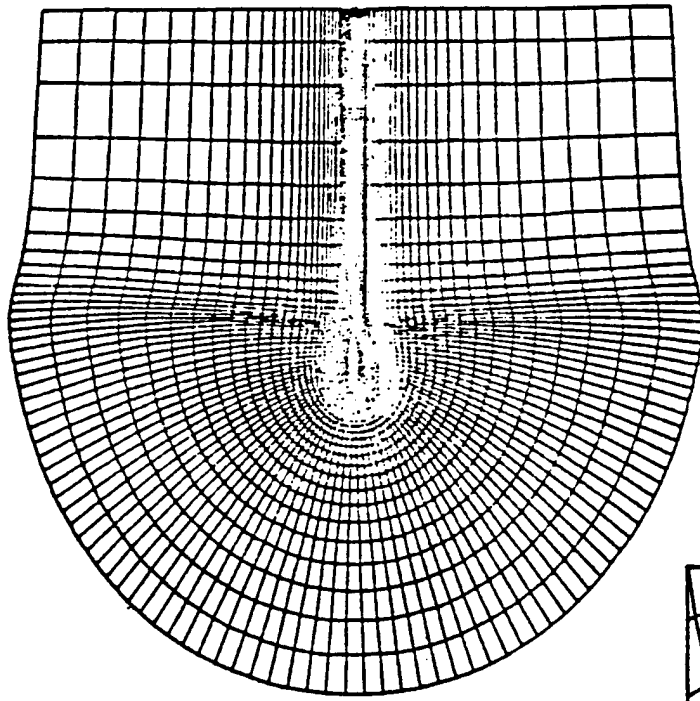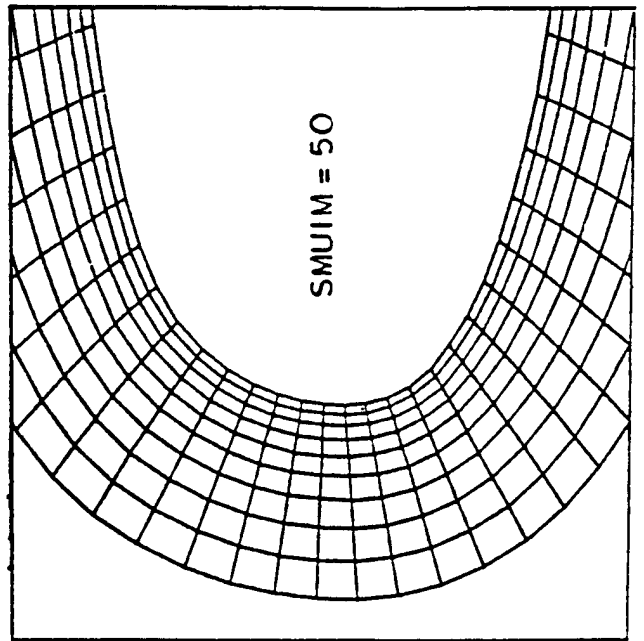
ALPHA = 2
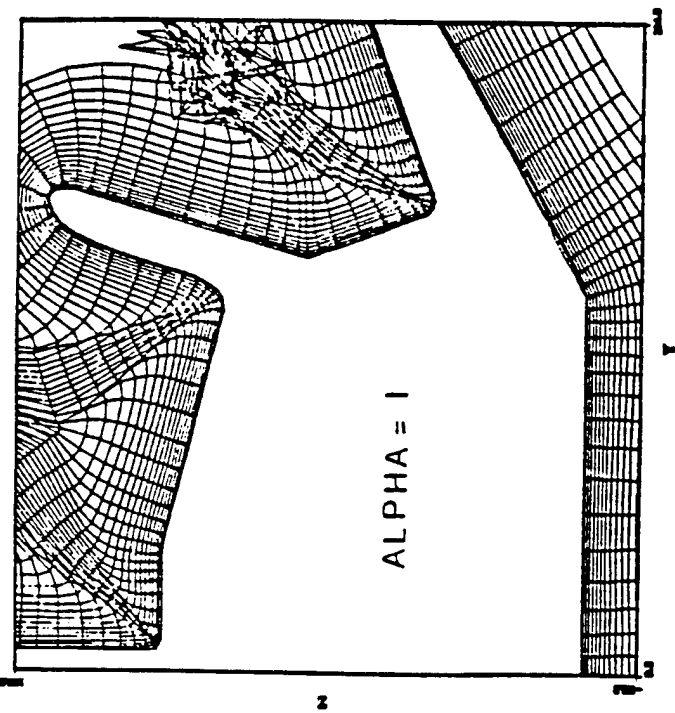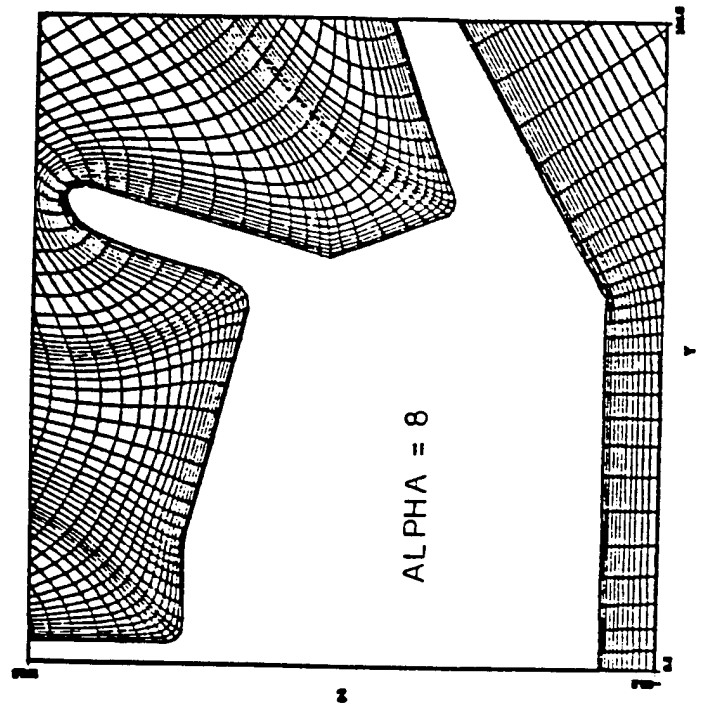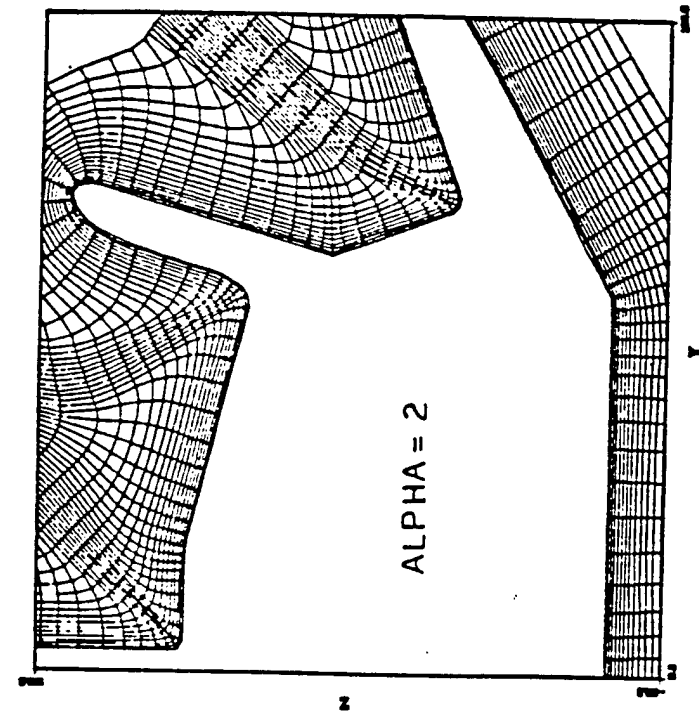
ALPHA = 8

ALPHA = 1

FIGURE 1   VARIATION OF ALPHA ON X-24C

35
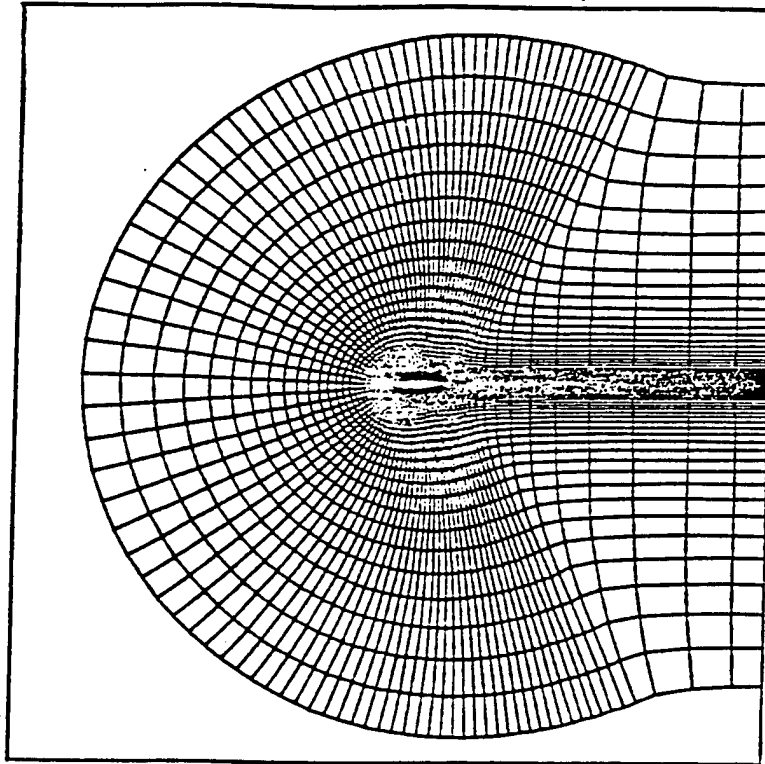
ALPHA =0.5

ALPHA = 2

FIGURE 20

36
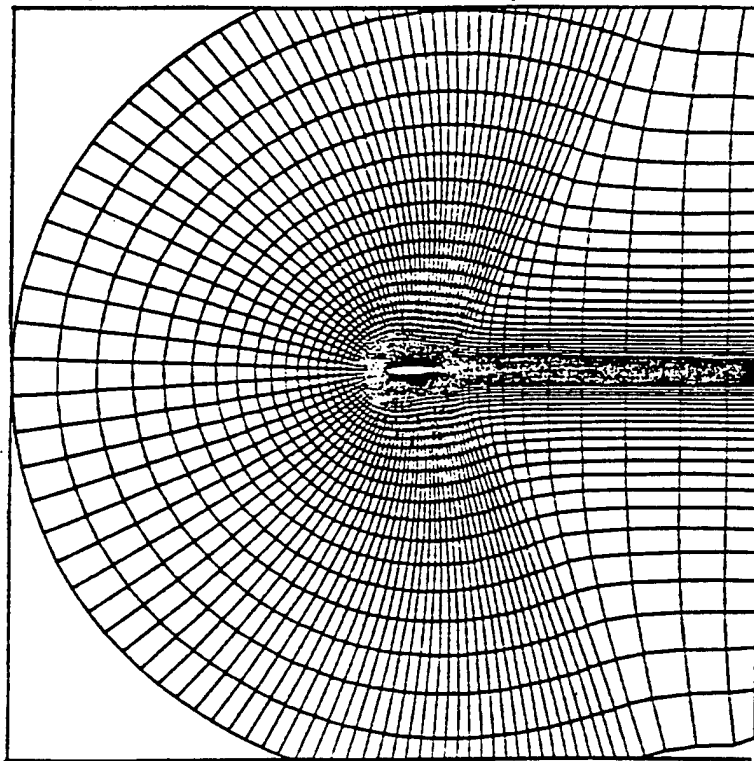
V. CONCLUSIONS

In conclusion, the hyperbolic grid generating procedure described is very fast and provides a good grid pattern with a lot of user control. The routine is tolerant to variations of all input parameters ("robust") except for the parameter that controls transition to equal cell areas (ESCAL) when rapid changes in grid spacings around the body are used. Tolerance to ESCAL could be improved by changing some of the other parameters (SMUIM for example) and/or adding or deleting input points. Additional work is needed in reducing the effect of the disturbance between high and low density grid regions. Another desirable feature for clustered grids would be to delay the transition to equal cell areas in the clustered regions until about a body length away from the body ($\eta = 1$).

APPENDIX A - SMOOTHING INPUT COORDINATES

Most finite difference procedures for which a grid is required, need the distribution of points around the body to be second order smooth.[2] In other words, a plot of grid point number (J) versus body perimeter length (S) should be such that the rate of change of curvature is mathematically smooth. Smoothing of the geometry coordinates (which constitutes the grid distribution at the surface) must be accomplished by a separate program before the geometry is input to the grid generating procedure. Any clustering of grid points, such as around the leading edge or at shock locations, must also be included in the body coordinates read by the grid generating routine. These clustered coordinates should also be second order smooth.

A program called SMOOTH was written that allows the distribution of coordinates around the body and wake to be broken into any number of intervals. Within an interval any number of points can be defined such that the distribution within the interval and across the end points is smooth. An exponential point distribution is used within the intervals and a tension spline is used to interpolate between the original body coordinates to define the new, smooth and (if desired) clustered points.

The end points of the intervals are identified by grid point numbers (J). The first interval always begins at the lower surface at X = XMAX (J = 1). The last interval ends at the upper surface at X = XMAX (J = XMAX). The program must also have an interval defined at the leading edge, X = XMIN, location to avoid the possibility of having more than one Y for a given X in an interval. The curve fitting routines require monotonically increasing or decreasing values for X in a given interval. Other points where intervals should be defined are the lower and upper locations of the trailing edge of the body. This would minimize the possibility of incorrectly defining the geometry in that region.

In addition to defining the location of the start (and thus stop) points of each interval, the user must define the number of points in each interval, and the grid spacing at each interval point. A typical set of input values for the SMOOTH routine is shown in Table III. This Table is for the NLR 7301 airfoil and wake with 100 points in both the original input geometry and the new smooth coordinates. Figure 21 shows a plot of coordinate point (J) versus body length (S) for the original coordinates and the new smooth coordinates. Table IV is a listing of the original coordinates and the smooth coordinates.

As mentioned above, a tension spline routine is used to determine the new coordinate values. The value of the tension parameter, SIGMA, must be provided by the user. The range of values is from zero (normal cubic spline) to infinity (straight line between points). For Figure 21 a value of 1000 was used. Too small a value for SIGMA can produce undesirable (not smooth) perturbations in the grid distribution; whereas, too large a value may produce "straight line" geometry definition. Figure 22 shows the coordinate distribution obtained by using the parameters in Table III and SIGMA = 1000. Figure 23 shows a coordinate distribution and resulting grid for clustered points between 50 and 60% chord on the upper surface. This is typical of what might be desired to provide a close grid spacing in the region of a shock wave. The value of the parameter ESCAL in the grid procedure will need to be kept small to keep the grid points clustered at points away from the body.

39

TABLE III

INPUT PARAMETERS FOR SMOOTH

STARTING J	NUMBER OF POINTS IN INTERVAL	GRID SPACING AT POINT J	COMMENTS
1	16	0.5	Lower surface rear boundary to lower surface trailing edge.
9	35	.04	Lower surface T.E. to L.E.
48	36	.006	L.E. to upper surface T.E.
92	16	.04	Upper surface T.E. to rear boundary.
100	0	0.5	Number of points is not needed here.

Note: The number of new coordinates defined is the sum of the number of points in each internal minus the number of internals, plus two; i.e., (16 + 35 + 36 + 16) - 5 + 2 = 100.
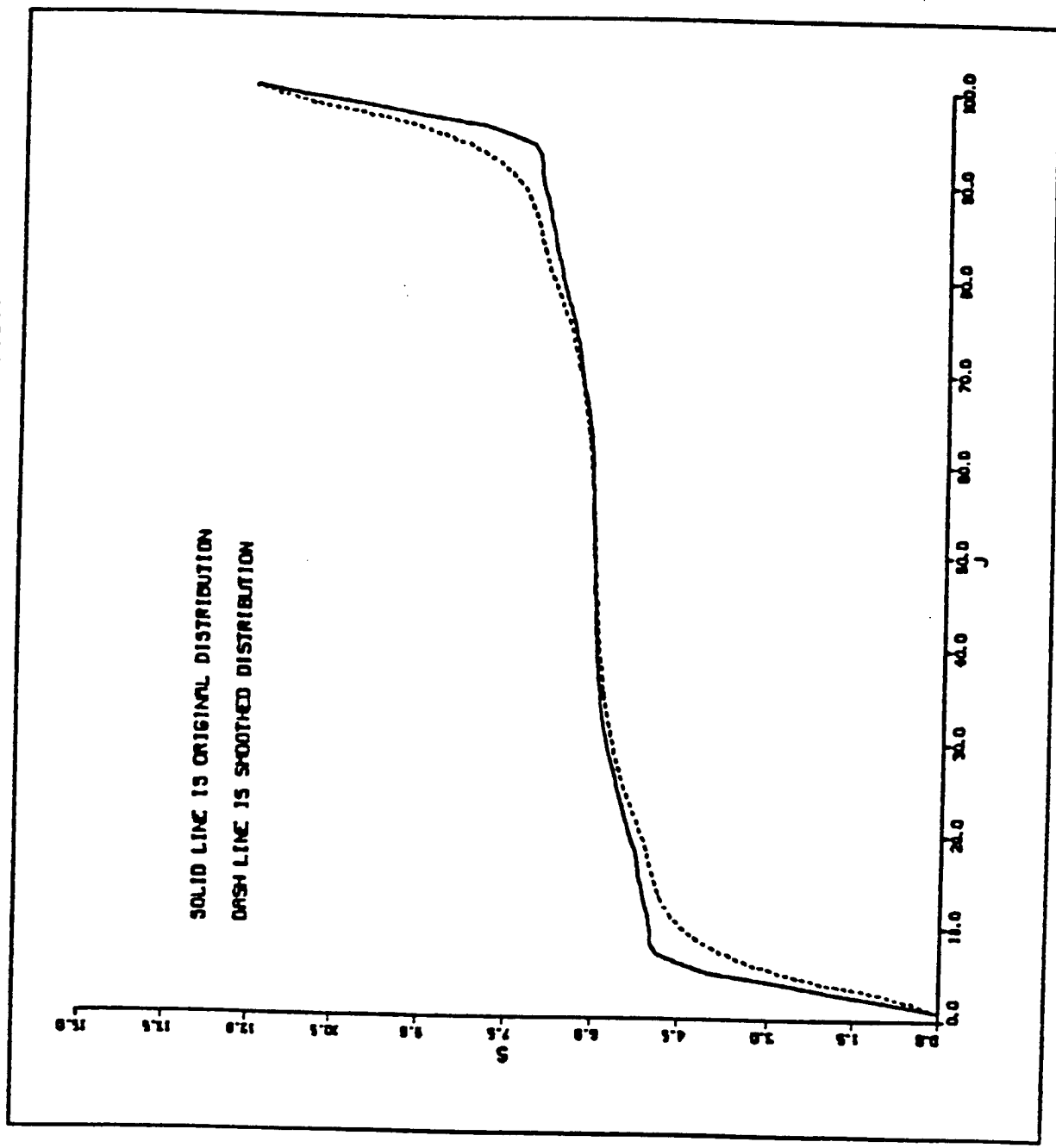
SMOOTH COORDINATE DISTRIBUTION

SOLID LINE IS ORIGINAL DISTRIBUTION
DASH LINE IS SMOOTHED DISTRIBUTION

41

FIGURE 21

TABLE IV

J	ORIGINAL		SMOOTH	
	X	Y	X	Y
1	6.00000	0.00000	6.00000	0.00000
2	5.00000	0.00000	5.46707	-0.00000
3	4.00000	0.00000	4.77434	0.00000
4	3.00000	0.00000	3.87389	-0.00000
5	2.00000	0.00000	3.18116	-0.00000
6	1.50000	0.00000	2.64807	0.00000
7	1.12500	0.00000	2.23818	0.00000
8	1.01560	0.00000	1.92291	0.00000
9	1.00000	-0.00040	1.68014	-0.00000
10	0.98850	0.00050	1.49351	-0.00000
11	0.95790	0.00220	1.34523	0.00000
12	0.91290	0.00180	1.23904	0.00000
13	0.87630	-0.00080	1.15432	0.00000
14	0.83960	-0.00550	1.09032	0.00000
15	0.80400	-0.01130	1.04076	0.00001
16	0.77520	-0.01690	1.00000	-0.00040
17	0.75060	-0.02200	0.96003	0.00211
18	0.72230	-0.02810	0.91671	0.00186
19	0.66360	-0.04080	0.86905	-0.00170
20	0.61960	-0.05050	0.81745	-0.00909
21	0.56340	-0.06140	0.76124	-0.01979
22	0.52000	-0.06790	0.69984	-0.03296
23	0.47440	-0.07280	0.63324	-0.04750
24	0.43720	-0.07520	0.57206	-0.05974
25	0.39660	-0.07690	0.51548	-0.06841
26	0.35600	-0.07780	0.46359	-0.07352
27	0.31530	-0.07780	0.41586	-0.07610
28	0.27640	-0.07670	0.37230	-0.07745
29	0.23610	-0.07440	0.33247	-0.07781
30	0.19970	-0.07150	0.29604	-0.07727
31	0.16670	-0.06830	0.26280	-0.07594
32	0.13380	-0.06440	0.23246	-0.07412
33	0.11140	-0.06110	0.20478	-0.07191
34	0.09650	-0.05860	0.17945	-0.06955
35	0.08100	-0.05560	0.15643	-0.06709
36	0.06190	-0.05100	0.13546	-0.06461
37	0.04510	-0.04580	0.11634	-0.06184
38	0.03580	-0.04210	0.09863	-0.05897
39	0.02950	-0.03920	0.08264	-0.05593
40	0.02440	-0.03650	0.06830	-0.05257
41	0.01570	-0.03360	0.05546	-0.04905
42	0.01070	-0.02660	0.04369	-0.04527
43	0.00730	-0.02260	0.03321	-0.04094
44	0.00460	-0.01830	0.02388	-0.03620
45	0.00210	-0.01280	0.01606	-0.03090
46	0.00090	-0.00890	0.00969	-0.02553
47	0.00050	-0.00590	0.00497	-0.01899
48	0.00000	0.00000	0.00200	-0.01258
49	0.00060	0.00880	0.00051	-0.00601
50	0.00120	0.01310	0.00000	0.00000
51	0.00180	0.01580	0.00039	0.00614
52	0.00290	0.01980	0.00109	0.01254
53	0.00570	0.02750	0.00282	0.01953
54	0.00830	0.03260	0.00539	0.02676

TABLE IV CONCLUDED

|	ORIGINAL		SMOOTH	
J	X	Y	X	Y
55	0.01210	0.03850	0.00939	0.03441
56	0.01660	0.04380	0.01470	0.04169
57	0.02040	0.04730	0.02202	0.04858
58	0.02420	0.05020	0.03134	0.05447
59	0.02770	0.05250	0.04245	0.05915
60	0.03120	0.05440	0.05493	0.06306
61	0.03800	0.05750	0.06859	0.06644
62	0.04740	0.06090	0.08365	0.06958
63	0.06160	0.06490	0.10063	0.07245
64	0.08100	0.06910	0.11868	0.07504
65	0.09690	0.07190	0.13852	0.07747
66	0.12270	0.07560	0.16018	0.07985
67	0.15340	0.07920	0.18391	0.08206
68	0.18320	0.08200	0.20988	0.08415
69	0.20660	0.08390	0.23819	0.08618
70	0.22960	0.08560	0.26902	0.08801
71	0.25490	0.08730	0.30265	0.08941
72	0.28300	0.08870	0.33933	0.09034
73	0.31380	0.08980	0.37932	0.09042
74	0.34760	0.09050	0.42309	0.08987
75	0.38370	0.09040	0.47035	0.08925
76	0.42070	0.08990	0.52223	0.08718
77	0.47310	0.08920	0.57843	0.08273
78	0.51600	0.08760	0.63974	0.07498
79	0.56160	0.08440	0.70589	0.06501
80	0.59950	0.08060	0.76649	0.05342
81	0.63110	0.07620	0.82169	0.04136
82	0.67440	0.07010	0.87217	0.02966
83	0.70960	0.06440	0.91840	0.01890
84	0.73920	0.05900	0.96047	0.00924
85	0.77050	0.05260	1.00000	0.00040
86	0.80400	0.04540	1.04079	-0.00001
87	0.84070	0.03700	1.09034	-0.00000
88	0.88060	0.02770	1.15431	-0.00000
89	0.92400	0.01760	1.23903	-0.00000
90	0.96370	0.00850	1.34922	-0.00000
91	0.98310	0.00420	1.49350	0.00000
92	1.00000	0.00040	1.68013	0.00000
93	1.01560	0.00000	1.92290	-0.00000
94	1.12500	0.00000	2.23817	-0.00000
95	1.50000	0.00000	2.64805	-0.00000
96	2.00000	0.00000	3.18112	0.00000
97	3.00000	0.00000	3.87387	0.00000
98	4.00000	0.00000	4.77432	-0.00000
99	5.00000	0.00000	5.46705	0.00000
100	6.00000	0.00000	6.00000	0.00000
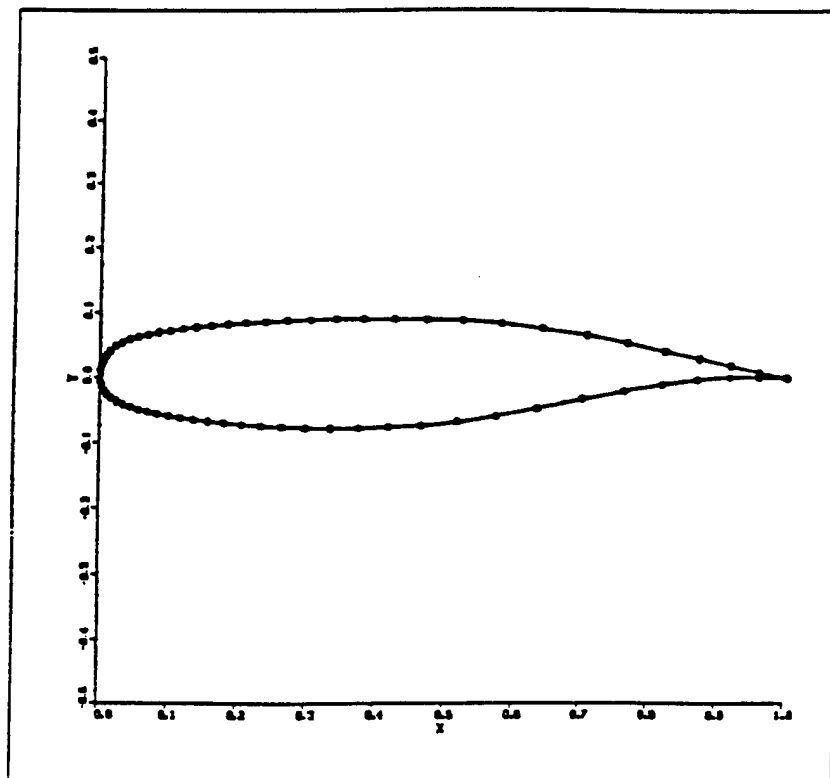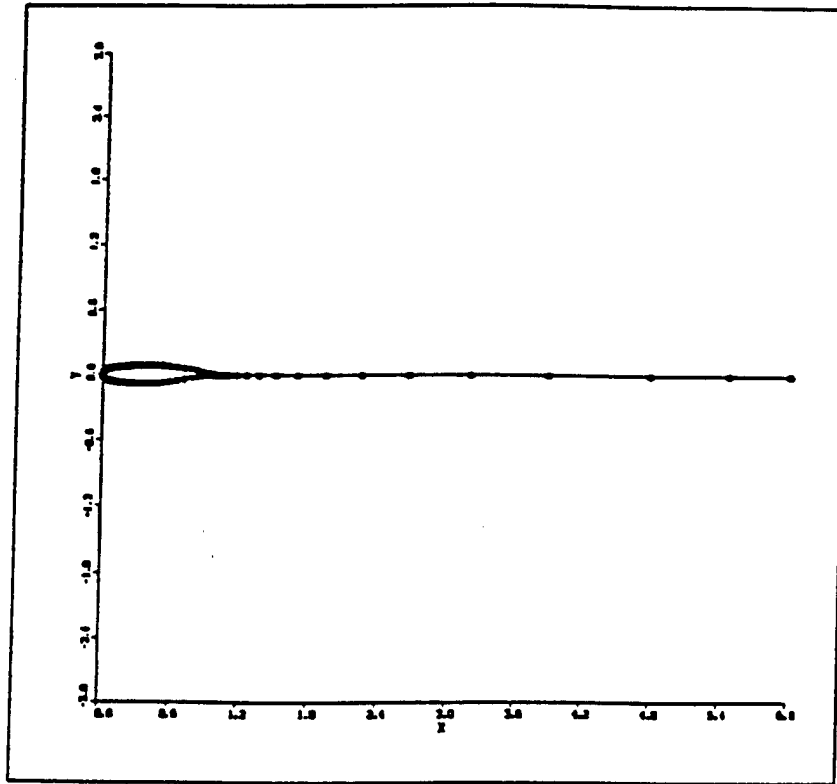
FIGURE 22

DASH LINE IS NEW BODY DISTRIBUTION.
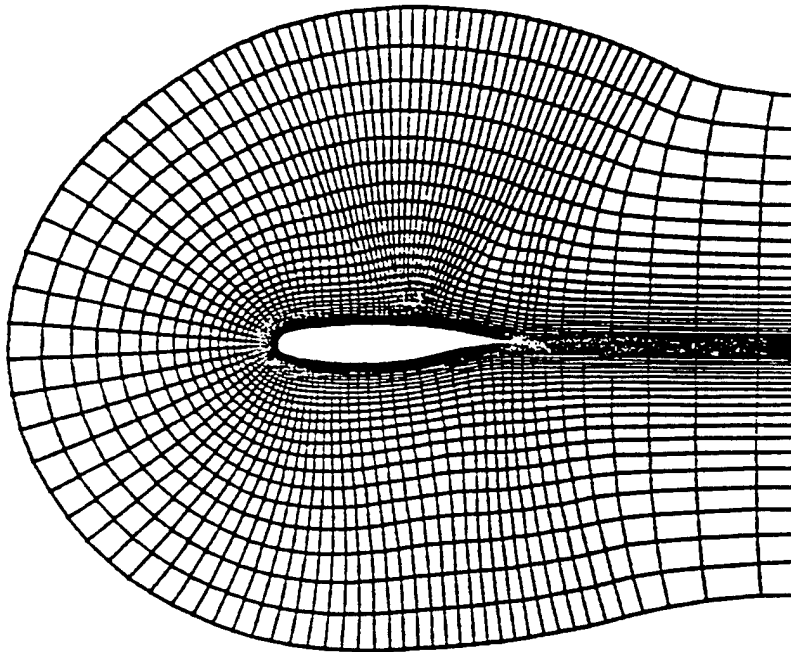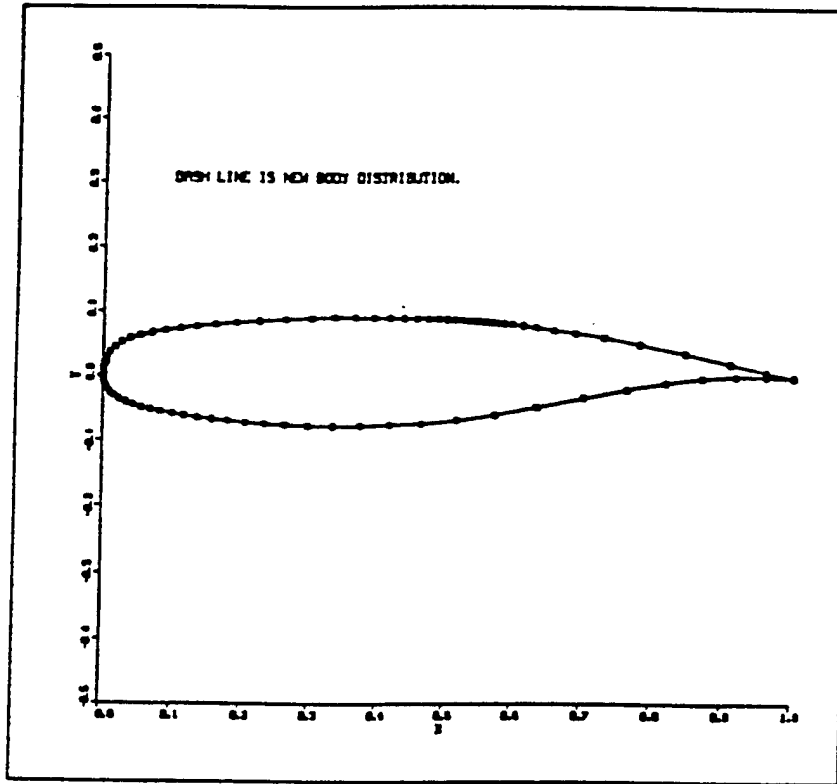
FIGURE 23

REFERENCES

1. Steger, J. and Chaussee, D., "Generation of Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations," SIAM J. Sci. Stat. Comput., Vol 1, Dec 1980, pp. 431-437.

2. Thompson, J. F. (Editor), Numerical Grid Generation, Elseiver Science Publishing Co., New York, 1982, pp. 31-40.

3. Beam, R. W. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form", J. Comp. Phys., 22 (1976), pp. 87-110.

4. Davis, S. and Malcolm, G., "Experimental Unsteady Aerodynamics of Conventional and Supercritical Airfoils," NASA TM 81221, Aug 1980.