NASA Technical Memorandum 73252

Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations

Reese L. Sorenson and Joseph L. Steger

AUGUST 1977

NASA

Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations

Reese L. Sorenson and Joseph L. Steger

Ames Research Center
Moffett Field, California

SIMPLIFIED CLUSTERING OF NONORTHOGONAL GRIDS GENERATED BY

ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

Reese L. Sorenson and Joseph L. Steger

Ames Research Center

SUMMARY

A simple clustering transformation is combined with the Thompson, Thames, and Mastin (TTM) method of generating computational grids to produce controlled mesh spacings. For various practical grids, the resulting hybrid scheme is easier to apply than the inhomogeneous clustering terms included in the TTM method for this purpose. The technique is illustrated in application to airfoil problems, and listings of a FORTRAN computer code for this usage are included.

INTRODUCTION

Thompson, Thames, and Mastin (TTM) have developed a practical way of generating two-dimensional computational grids about arbitrary bodies with coordinate lines coincident with all boundaries (refs. 1,2). Curvilinear nonorthogonal coordinates are generated by conventional numerical solution of two nonlinear elliptic partial differential equations. Because a refined grid is generally needed in regions where a rapid change in gradient of the flow variables occurs, the TTM method includes a provision for concentrating coordinate lines in any region of the field. However, this clustering feature can be either complicated or unsatisfactory, at least for the uninitiated user. The purpose of this note is to illustrate a simple alternative procedure for clustering about any grid line.

The authors wish to acknowledge the helpful discussions with L. B. Schiff with whom we first investigated the TTM scheme, and the assistance of P. Kutler who later programmed an airfoil grid generation scheme (some portions of which we adapted) to investigate the effectiveness of the P and Q terms. The authors also wish to acknowledge the assistance given by C. Schulbach in writing computer code for the SC4020 plotter.

THE THOMPSON-THAMES-MASTIN METHOD

In its simplest form, the TTM method solves Laplace equations in the physical plane

$$\xi_{xx} + \xi_{yy} = 0 \tag{1}$$

$$\eta_{xx} + \eta_{yy} = 0 \tag{2}$$

to generate curvilinear coordinates $\xi(x,y)$ and $\eta(x,y)$, which form a rectangular grid in the transformed computational plane. Equations (1) and (2) are transformed into the computational plane as

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \tag{3}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \tag{4}$$

where $\alpha = x_\eta^2 + y_\eta^2$, $\beta = x_\xi x_\eta + y_\xi y_\eta$, and $\gamma = x_\xi^2 + y_\xi^2$. Although equations (3) and (4) are mildly nonlinear, it is usually a simple matter to solve for x and y on the rectangular ξ,η computational grid by conventional relaxation methods. Then, if the transformation Jacobian at each point is nonzero, the locally one-to-one mapping between the computational plane and the physical plane yields the solution for ξ,η in the physical plane as well.

Simple boundary conditions usually accompany equations (3) and (4). For example, Dirichlet conditions can be picked for the blunt-body problem sketched in figure 1, where lines of constant η are chosen to coincide with the body and outer shock boundaries and the end points of lines of constant ξ are arbitrarily spaced along these boundaries. Lines of constant ξ terminate the blunt body flow region, with end points of lines of constant η distributed along these boundaries. This ability to arbitrarily locate boundary points by specifying their locations in the x,y plane is one of the most desirable and significant aspects of the TTM method.

While the Laplace equations automatically generate grids, the nodes are often poorly concentrated in regions where a large change of gradient in the flow variables occurs. To illustrate this, a grid is constructed about an NACA 0012 airfoil using Dirichlet boundary conditions as in figure 2. The resulting grid is shown in figure 3(a), and the lines nearest the airfoil are shown in detail in figure 3(b). The computed grid has the deficiency that the spacing between the airfoil-wake boundary and the next outward line of constant η is too large and not uniform, especially near the airfoil trailing edge. Clustering of lines of constant η to the airfoil-wake boundary (*abcde* in fig. 2) is needed.

The provision for clustering in the TTM method consists of adding inhomogeneous terms to equations (1) and (2), producing Poisson equations

$$\xi_{xx} + \xi_{yy} = P(\xi,\eta) \tag{5}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi,\eta) \tag{6}$$

which in the computational plane become

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(Px_\xi + Qx_\eta) \tag{7}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(Py_\xi + Qy_\eta) \tag{8}$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is the transformation Jacobian. Here P and Q are summations of exponential terms, as defined in reference 2 (however, the sign of these terms in ref. 2 is in error).

The harmonic functions P and Q can cause clustering about any number of nodes or grid lines. The TTM method of clustering is versatile (with respect to where clustering takes place in the field), but exact control of mesh spacings can be difficult. Particularly difficult is the imposition of a predetermined and uniform spacing between two adjacent grid lines along their entire length.

To illustrate this last remark, consider the 0012 airfoil application. Here terms of Q giving clustering near the airfoil-wake boundary line of constant η are used. The closeup of the resultant grid, shown in figure 4, is improved over the unclustered grid shown in figure 3(b) but is still not sufficiently uniform around the airfoil.

COMBINING SIMPLE STRETCHING WITH THE TTM METHOD

In this section we show that for particular problems in which only grid refinement near a boundary is desired, a simple stretching procedure combined with the simplest, unclustered form of the TTM method can produce good results. This technique is easier to implement than the inhomogeneous form of the TTM method.

In the airfoil problem, the ξ spacing can be sufficiently controlled by placement in the x,y plane of the nodes along the η_{min} and η_{max} boundaries, so only clustering of lines of constant η is needed. Instead of trying to enforce a specific clustering by choice of Q, the lines of constant ξ from the inhomogeneous form of the TTM method can be accepted and the lines of constant η can be discarded. Then points on lines of constant η are redistributed along each line of constant ξ by means of simple monotonic stretching functions. This method is detailed below for an airfoil application where an exponential stretching function is chosen to give a specified minimum grid spacing adjacent to the airfoil-wake boundary.

Given the grid resulting from the simplest form of the TTM method, the arc length $S(\eta)|_\xi$ along each constant ξ line is computed from the relations

$$S_1 = 0 \tag{9}$$

$$S_k = S_{k-1} + \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2} \qquad 2 \le k \le k_{max} \tag{10}$$

where k is the index of the kth line of constant η. Along each line of constant ξ, equations (9) and (10) allow tabulation of x_k and y_k as functions of S_k. The η spacing from the TTM solution can now be discarded.

An exponential stretching transformation is introduced such that $\tilde{S}(\eta)\big|_{\xi}$ will be the arc length along the constant ξ line with grid-point distribution defined by

$$\tilde{S}_1 = 0 \tag{11}$$

$$\tilde{S}_k = \tilde{S}_{k-1} + \Delta\tilde{S}_{\min}(1 + \varepsilon)^{k-2} \qquad 2 \leq k \leq k_{\max} \tag{12}$$

where $\Delta\tilde{S}_{\min}$ is the desired minimum physical spacing along lines of constant ξ near the body, and ε is chosen so that $\tilde{S}_{k_{\max}} = S_{k_{\max}}$. Values of x_k and y_k corresponding to the new $\tilde{S}_k$ are obtained by interpolating the $x = x(S_k)$ and $y = y(S_k)$ tables generated with equations (9) and (10). These locations define the new lines of constant η. A new grid has therefore been defined where the lines of constant η are uniformly spaced in the physical plane and equal to $\Delta\tilde{S}_{\min}$ along the entire airfoil-wake boundary and exponentially stretched toward the outer ($\eta = \eta_{\max}$) boundary.

For each constant ξ line, the value of ε is determined such that $\tilde{S}_{k_{\max}} = S_{k_{\max}}$ using a Newton-Raphson iterative procedure. With ε_n defined as the value of ε after the nth iteration

$$\varepsilon_n = \varepsilon_{n-1} - \frac{F(\varepsilon_{n-1})}{F'(\varepsilon_{n-1})} \tag{13}$$

where

$$F(\varepsilon) = S_{k_{\max}} - \tilde{S}_{k_{\max}} \tag{14}$$

Combining equations (11), (12), and (14) and reducing yields

$$F(\varepsilon) = S_{k_{\max}} - \frac{\Delta\tilde{S}_{\min}}{\varepsilon}\left[(1 + \varepsilon)^{k_{\max}-1} - 1\right] \tag{15}$$

from which it is seen that

$$F'(\varepsilon) = \frac{\Delta\tilde{S}_{\min}}{\varepsilon^2}\left\{1 + (1 + \varepsilon)^{k_{\max}-2}\left[\varepsilon(k_{\max} - 2) - 1\right]\right\} . \tag{16}$$

A value for ε_0 on the first constant ξ line for which the Newton-Raphson procedure has been found to be stable is obtained from

$$\varepsilon_0 = \left(\frac{S_{k_{\max}}}{\Delta\tilde{S}_{\min}}\right)^{1/(k_{\max}-2)} - 1 \tag{17}$$

For all subsequent constant ξ lines, the final solution ε for the immediately previous line is used for ε_0.

4

The above procedure has been applied to the NACA 0012 airfoil with $\Delta \tilde{S}_{min}$ as small as 0.00005 chords. An example with $\Delta \tilde{S}_{min} = 0.01$ chords is shown in figure 5(a) and a closeup of the lines nearest the airfoil is seen in figure 5(b). The control of the grid line spacing is evident. In comparing figure 5(a) with figure 3(a), one observes that the outermost line of constant η (in the shape of three sides of a rectangle) has been discarded and the next inner line of constant η is taken for the $\eta = \eta_{max}$ boundary. This was done before application of the simple clustering procedure for the purpose of minimizing variations in the lengths of different constant ξ lines. The appendix contains a listing of the FORTRAN computer code used to generate these grids and notes on its use.

CONCLUDING REMARKS

A simple stretching transformation has been combined with the simplest form of the Thompson-Thames-Mastin grid generation method to produce grids with certain types of controlled spacings. The technique has been used to generate various grids including those for viscous flow calculations about airfoils.

Ultimately, the clusterings in both ξ and η will probably be done by an improved version of the Thompson-Thames-Mastin method. Such an algorithm should include a procedure for dynamic adjustment of the parameters in the P and Q terms during solution of the Poisson equations. For the present, the much simpler method is adequate for many cases.

Ames Research Center
National Aeronautics and Space Administration
Moffett Field, Calif. 94035, March 8, 1977

5

FORTRAN PROGRAM FOR AIRFOIL GRIDS

The FORTRAN computer program that combines simple exponential stretching with the TTM method to produce two-dimensional grids about airfoils is designed to be run on the Control Data Corporation 7600 computer system but should prove easily adaptable to other installations. It is approximately 700 cards in length and consists of a main program and twelve short subroutines. Of these subroutines, five deal with plotting the resultant grids on the Stromberg-Carlson SC4020 plotter and can be easily removed if this feature is not desired. Another two of the included decks are proven utility subroutines for standard mathematical operations. Thus, only the main program and five of the subroutines constitute the operative part of the code that creates the grids.

MAIN PROGRAM

The main program first reads input data cards, which are detailed in a later section of this appendix. Subroutine INNER is called to distribute points on the airfoil-wake and rear boundaries. Points on the bottom-front-top boundary are located by subroutine OUTER. With all boundary points fixed, initial conditions for the TTM method are provided by equally spacing grid nodes along straight lines of constant ξ between corresponding boundary points. The TTM method is applied by subroutine RELAX, exponential stretching is applied by subroutine CLUSTR, and the finally resultant grid is output using a FORTRAN unformatted WRITE statement.

```
*DECK MAIN
      PROGRAM MAIN(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT,TAPE48,TAPE8)
C
C         LOGICAL UNIT NUMBERS USED IN THIS PROGRAM:
C
C         5  -  READ INPUT DATA CARDS
C         6  -  WRITE PRINTER OUTPUT
C         8  -  UNFORMATTED WRITE OF FINAL SOLUTION GRID
C        48  -  SC4020 PLOTTER DATA
C
      COMMON/TCOM/ITITLE(8)
C
      DIMENSION XB(140),YB(140),X(140,80),Y(140,80)
      DIMENSION LINE(8)
C
C         GRIDS ARE PRESENTLY LIMITED BY DIMENSION SIZE TO 140 POINTS
C         IN XI DIRECTION AND 80 POINTS IN ETA DIRECTION.  THESE
C         DIMENSION SIZES ARE FOUND IN SUBROUTINES INNER, OUTER,
C         RELAX, CLUSTR, EPSIL, AND PLAWT AS WELL AS HERE IN THE
C         MAIN PROGRAM.
```

```
C
      DATA PI/3.141592654/
C
C         READ IN PARAMETERS.
      READ(5,101) JWAKE,KMAX,MAXIT
      READ(5,102) XGMX,XGMN,YMAX,YMIN,XNOSE,XTAIL
      READ(5,102)  DY1
      READ(5,102)  XORG,ETACD,BETA
      READ(5,102)  OMEGA
      READ(5,102)  DY2
      CALL READIN
      READ(5,100)ITITLE
      READ(5,100)  LINE
C
      WRITE(6,112) JWAKE,KMAX,MAXIT,XGMX,XGMN,YMAX,YMIN,XNOSE,XTAIL
      WRITE(6,113)  DY1,XORG,ETACD,BETA,OMEGA,DY2
      WRITE(6,137)ITITLE
      WRITE(6,109)  LINE
      ETAC=ETACD*PI/180.
C
C         READ IN AIRFIOL SHAPE.
      N = 0
    1 N = N +1
      READ(5,102) XB(N),YB(N)
      IF( EOF(5)) 2,1
    2 NBOD = N-1
      WRITE(6,103)
      WRITE(6,104)  (N,XB(N),YB(N),N=1,NBOD)
C
C         DISTRIBUTE POINTS ON INNER AND REAR BOUNDARIES.
      CALL INNER(NBOD,JWAKE,KMAX,XB,YB,XNOSE,XTAIL,XGMX,YMAX,YMIN,
     1  DY1,JMAX,X,Y)
C
C         DISTRIBUTE POINTS ON BOTTOM-FRONT-TOP BOUNDARY.
      CALL OUTER(XGMX,XGMN,YMAX,YMIN,XORG,ETAC,BETA,JMAX,KMAX,X,Y)
C
C         FILL-IN BETWEEN ETA-MAX AND ETA-MIN BOUNDARIES.
C
34    JMM = JMAX -1
      KMM = KMAX-1
      RKMM=1.0/KMM
C
      DO 13 J=2,JMM
      DELX=(X(J,KMAX)-X(J,1))*RKMM
      DELY=(Y(J,KMAX)-Y(J,1))*RKMM
C
      DO 13 K=2,KMM
      X(J,K) = X(J,1) + ( K-1)*DELX
   13 Y(J,K) = Y(J,1) + ( K-1)*DELY
C
C         PLOT INITIAL CONDITIONS.
      CALL PLAWT(JMAX,KMAX,X,Y,XGMX,XGMN,YMAX,YMIN)
C
C         GENERATE UN-CLUSTERED GRID.
      CALL RELAX(JMAX,KMAX,X,Y,OMEGA,MAXIT)
C
C         PLOT UN-CLUSTERED GRID.
      CALL PLAWT(JMAX,KMAX,X,Y,XGMX,XGMN,YMAX,YMIN)
```

```
C
C            APPLY EXPONENTIAL CLUSTERING.
      CALL CLUSTR(JMAX,X,Y,DY2,KMAX)
C
C            PLOT EXPONENTIALLY CLUSTERED GRID.
      CALL PLAWT(JMAX,KMAX,X,Y,XGMX,XGMN,YMAX,YMIN)
C
C        WRITE FINAL SOLUTION.
      WRITE(8) ((X(J,K),J=1,JMAX),K=1,KMAX),((Y(J,K),J=1,JMAX),K=1,KMAX)
C
C            PLOT BLOW-UP OF INNERMOST 10 LINES.
      NJW=JWAKE/4
      NJM=NBOD+2*NJW
      JJJ=JWAKE-NJW-1
      XMAX=X(JWAKE,1)
      XMIN=XMAX
      YMAX=Y(JWAKE,1)
      YMIN=YMAX
C
      DO 35 K=1,10
      DO 35 J=1,NJM
      JJ=JJJ+J
      XX=X(JJ,K)
      YY=Y(JJ,K)
      IF(XX.GT.XMAX)   XMAX=XX
      IF(XX.LT.XMIN)   XMIN=XX
      IF(YY.GT.YMAX)   YMAX=YY
      IF(YY.LT.YMIN)   YMIN=YY
      X(J,K)=XX
35    Y(J,K)=YY
C
      CALL PLAWT(NJM,10,X,Y,XMAX,XMIN,YMAX,YMIN)
      CALL EOFTV
C
      STOP
C
10C   FORMAT(8A10)
101   FORMAT(16I5)
102   FORMAT(8F10.0)
103   FORMAT(/27H AIRFOIL POINTS AS READ IN.)
104   FORMAT(3H N=,I3,5X,3HXB=,E12.5,5X,3HYB=,E12.5)
109   FORMAT(34H CARD NO. 9:  AIRFIOL DESCRIPTION:,10X,8A10)
112   FORMAT(24H1 PARAMETERS AS READ IN://
     1 60H CARD NO. 1:  JWAKE, NUMBER OF POINTS IN EACH SIDE OF WAKE =,
     2 I5/14X,41HKMAX, NUMBER OF POINTS IN ETA DIRECTION =,I5/
     3 14X,57HMAXIT, MAXIMUM NUMBER OF ITERATIONS IN MAKING UN-CLUSTERE,
     4 8HD GRID =,I5/
     5 57H CARD NO. 2:  XGMX, X-DIRECTION COORDINATE OF DOWNSTREAM ,
     6 10HBOUNDARY =,F20.10/14X,32HXGMN, X-DIRECTION COORDINATE OF ,
     7 19HUPSTREAM BOUNDARY =,F20.10/14X,25HYMAX, Y-DIRECTION COORDIN,
     8 23HATE OF UPPER BOUNDARY =,F20.10/14X,23HYMIN, Y-DIRECTION COORD,
     9 25HINATE OF LOWER BOUNDARY =,F20.10/14X,19HXNOSE, X-DIRECTION ,
     A 39HCOORDINATE OF LEADING EDGE OF AIRFOIL =,F20.10/14X,
     B 59HXTAIL, X-DIRECTION COORDINATE OF TRAILING EDGE OF AIRFOIL =,
     C F20.10)
```

```
113    FORMAT(51H CARD NO. 3:   DY1, MINIMUM Y-INCREMENT ON REARWARD ,
      1 33HBOUNDARY FOR INITIAL CONDITIONS =,F20.10/14H CARD NO. 4:   ,
      2 61HXORG, X-DIRECTION LOCATION OF ORIGIN FOR ANGULAR CLUSTERING =,
      3  F20.10/14X,45HETAC, ANGLE (IN DEGREES) ABOUT WHICH ANGULAR ,
      4 20HCLUSTERING IS DONE =,F8.2/14X,25HBETA, ANGULAR CLUSTERING ,
      5 11HPARAMETER =,F20.10/35H CARD NO. 5:   OMEGA, PARAMETER FOR ,
      6 17HTHOMPSON SOLVER =,F20.10/31H CARD NO. 6:   DY2, MINIMUM ETA-,
      7 40HDIRECTION SPACING FOR FINAL CLUSTERING =,F20.10)
137    FORMAT(25H CARD NO. 8:   PLOT TITLE:,10X,8A10)
C
       END
```

Subroutine INNER distributes points on the airfoil-wake ($\eta = \eta_{min}$)
boundary and the rear boundaries. The distribution of points on the airfoil
part of the $\eta = \eta_{min}$ boundary is taken from the input data cards describing
the airfoil shape; no redistribution of these points or modifications of their
number is allowed for in the program. The airfoil shape, however, is shifted
and scaled to allow arbitrary placement on the x axis of the leading and
trailing edges. Points on the wake part of the $\eta = \eta_{min}$ boundary are exponen-
tially stretched toward the rearward boundary using the transformation

$$\tilde{S}_1 = 0 \tag{18}$$

$$\tilde{S}_k = \tilde{S}_{k-1} + \Delta\tilde{S}_{min}\left(1 + \frac{\epsilon}{\sqrt{k-1}}\right)^{k-2} \tag{19}$$

much like equations (11) and (12). Distribution of points in the y direction
on the rear ($\xi = \xi_{min}$ and $\xi = \xi_{max}$) boundaries is also obtained by exponential
stretching from the wake to the bottom and top boundaries using the transfor-
mation in equations (11) and (12). It should be noted that the y-direction
point distribution on the rear boundaries fa and ei of figure 2 is used only
as a boundary condition in the TTM method and is discarded along with all
other η spacing in the application of the final exponential clustering.

```
*DECK INNER
      SUBROUTINE INNER(NBOD,JWAKE,KMAX,XB,YB,XNOSE,XTAIL,XGMX,YMAX,YMIN,
     1 DY1,JMAX,X,Y)
C
C         THIS SUBROUTINE DISTRIBUTES POINTS ON THE ETA-MIN (AIRFOIL-
C         WAKE, INNER) BOUNDARY AND THE REAR (OUTFLOW) BOUNDARY.
C
      DIMENSION XB(1),YB(1),X(140,80),Y(140,80)
C
C         SHIFT AND SCALE AIRFOIL POIMTS.
      XMIN = XB(1)
      XMAX = XB(1)
      DO 3 N=2,NBOD
      IF( XB(N) .LT. XMIN) XMIN = XB(N)
    3 IF( XB(N) .GT. XMAX) XMAX = XB(N)
C
      SCALE = (XTAIL-XNOSE)/( XMAX - XMIN)
      DO 4 N=1,NBOD
      XB(N) = ( XB(N) - XMIN)*SCALE+XNOSE
    4 YB(N) = YB(N)*SCALE
      WRITE(6,105)
      WRITE(6,106)  (N,XB(N),YB(N),N=1,NBOD)
C
      JWT = JWAKE + NBOD -1
      N = 0
      DO 5 J = JWAKE,JWT
      N = N +1
      X(J,1) = XB(N)
    5 Y(J,1) = YB(N)
```

```
C
C          DISTRIBUTE POINTS ON WAKE BOUNDARY.
      DXO = (X(JWT,1) - X(JWT-1,1) +X(JWAKE,1) -X(JWAKE+1,1))*.5
      N = -1
      JMAX = NBOD +2*JWAKE -2
      JWTP = JWT + 1
      XX = ( X(JWT,1) + X(JWAKE,1))*.5
      YY = (Y(JWT,1) + Y(JWAKE,1))*.5
      EPSW=EPSIL(XGMX,XX,DXO,JMAX-JWT+1,0.001,100,2,1)
C
      DO 6 J=JWTP,JMAX
      N = N +1
      JJ = JWAKE -1 -N
      XX=XX+DXO*(1.0+EPSW/SQRT(FLOAT(N+1)))**N
      X(J,1) = XX
      X(JJ,1) = XX
      Y(J,1) = YY
    6 Y(JJ,1) = YY
      NONE=1
      WRITE(6,107)
      WRITE(6,108)   (J,NONE,X(J,1),J,NONE,Y(J,1),J=1,JMAX)
C
C          DISTRIBUTE POINTS IN Y DIRECTION ON REARWARD BOUNDARY.
      N=-1
      EPSY1T=EPSIL(YMAX,YY,DY1,KMAX,0.001,100,1,1)
      EPSY1B=EPSIL(-YMIN,-YY,DY1,KMAX,C.001,100,1,2)
C
      DO 8 K=2,KMAX
 
      N=N+1
      Y(JMAX,K)=Y(JMAX,K-1)+DY1*(1.0+EPSY1T)**N
      Y(1,K)=Y(1,K-1)-DY1*(1.0+EPSY1B)**N
      X(JMAX,K)=XGMX
    8 X(1,K)=XGMX
C
      WRITE(6,110)
      WRITE(6,108)   (NONE,K,X(1,K),NONE,K,Y(1,K),K=1,KMAX)
      WRITE(6,111)
      WRITE(6,108)   (JMAX,K,X(JMAX,K),JMAX,K,Y(JMAX,K),K=1,KMAX)
C
      RETURN
C
105   FORMAT(/32H AIRFOIL POINTS AFTER RESCALING.)
106   FORMAT(3H N=,I3,5X,3HXB=,F12.5,5X,3HYB=,F12.5)
107   FORMAT(/30H BOUNDARY ON WAKE AND AIRFOIL:)
108   FORMAT(3H X(,I3,1H,,I3,2H)=,F12.5,5X,2HY(,I3,1H,,I3,2H)=,F12.5)
110   FORMAT(/33H LOWER PART OF REARWARD BOUNDARY:)
111   FORMAT(/33H UPPER PART OF REARWARD BOUNDARY:)
C
      END
```

SUBROUTINE OUTER

Subroutine OUTER distributes points on the bottom-front-top boundary by first locating a special "clustering origin" at an arbitrary point on the x axis, and then placing the boundary points in an angular fashion about that origin. The angular distribution of boundary points is derived from an angular clustering transformation given by Davy and Reinhardt (ref. 3). The user specifies the angle about which clustering is performed and a parameter determining the strength of clustering. Equal angular distribution is available as a special case.

```
      SUBROUTINE OUTER(XMAX,XMIN,YMAX,YMIN,XORG,ETAC,BETA,JMAX,KMAX,X,Y)
C
C         THIS SUBROUTINE PLACES POINTS ON BOTTOM-FRONT-TOP BOUNDARY IN
C         ANGULAR FASHION.
C
      DIMENSION X(140,80),Y(140,80)
C
      LOGICAL CLUSTR
C
      DATA PI/3.141592654/
C
      SINH(X)=0.5*(EXP(X)-EXP(-X))
C
      ETARU=ATAN2(YMAX,XMAX-XORG)
      ETARL=ATAN2(-YMIN,XMAX-XORG)
      DETA=(2.0*PI-(ETARU+ETARL))/(JMAX-1)
      CLUSTR = .FALSE.
      IF( BETA.GT.0.0) CLUSTR = .TRUE.
      IF(.NOT. CLUSTR) GO TO 14
      FACT=PI/(2.0*PI-(ETARU+ETARL))
      FACTR=1.0/FACT
      ETACT=(ETAC-ETARU)*FACT
      B=0.5*ALOG((1.+(EXP(BETA)-1.)*ETACT/PI)/(1.+(EXP(-BETA)-1.)*
     1 ETACT/PI))
      RSB = 1./SINH(B)
14    ETA=ETARU
      ANG1=ATAN2(YMAX,XMIN-XORG)
      ANG2=ATAN2(YMIN,XMIN-XORG)+2.0*PI
      NSIDE = 1
C
      ETARUD=ETARU*180./PI
      ETARLD=ETARL*180./PI
      ANG1D=ANG1*180./PI
      ANG2D=ANG2*180./PI
      WRITE(6,109) CLUSTR,ETARUD,ETARLD,ANG1D,ANG2D
      IF(CLUSTR.AND.ETAC.LT.ETARU.OR.CLUSTR.AND.ETAC.GT.(2.0*PI-ETARL))
     1 GO TO 22
```

```fortran
C
      DO 9 JJ=2,JMAX
      J=JMAX+1-JJ
      ETA = ETA + DETA
      IF(.NOT.CLUSTR)  GO TO 26
      ETAT=(ETA-ETARU)*FACT
      PHIT=ETACT*(SINH(BETA*ETAT/PI-B)*RSB+1.)
      PHI=ETARU+PHIT*FACTR
      GO TO 27
26    PHI=ETA
C
27    GO TO (1,2,3),NSIDE
1     IF(PHI.GT.ANG1)  NSIDE=2
      GO TO 3
2     IF(PHI.GT.ANG2)  NSIDE=3
C
3     GO TO (10,11,12),NSIDE
   10 Y(J,KMAX) = YMAX
      X(J,KMAX) = YMAX/TAN(PHI) + XORG
      GO TO 21
   11 X(J,KMAX) = XMIN
      Y(J,KMAX) = ( XMIN - XORG)*TAN(PHI)
      GO TO 21
12    X(J,KMAX)=XORG+YMIN/TAN(PHI)
      Y(J,KMAX)=YMIN
21    CONTINUE
C
      ETAD=ETA*180./PI
      PHID=PHI*180./PI
      WRITE(6,113)   J,KMAX,X(J,KMAX),J,KMAX,Y(J,KMAX),ETAD,PHID,NSIDE
    9 CONTINUE
C
      RETURN
C
22    WRITE(6,114)
      STOP
C
109   FORMAT(/8H CLUSTR=,L1,5X,6HETARU=,F8.2,5X,6HETARL=,F8.2,5X,
     1 5HANG1=,F8.2,5X,5HANG2=,F8.2//24H OUTER BOUNDARY ON TOP, ,
     2 18HFRONT, AND BOTTOM:)
113   FORMAT(3H X(,I3,1H,,I3,2H)=,F12.5,5X,2HY(,I3,1H,,I3,2H)=,F12.5,
     1 5X,4HETA=,F8.2,5X,4HPHI=,F8.2,5X,6HNSIDE=,I1)
114   FORMAT(/23H ERROR EXIT.  BAD ETAC.)
C
      END
```

SUBROUTINE RELAX

Subroutine RELAX creates a grid by solving equations (3) and (4) in the uniform ξ-η domain using successive line overrelaxation (SLOR). Second-order accurate central difference operators are used throughout, and tridiagonal matrices are solved along lines of constant η using the utility program TRIB.

```
*DECK RELAX
      SUBROUTINE RELAX(JMAX,KMAX,X,Y,OMEGA,MAXIT)
C
C         THIS SUBROUTINE SOLVES BY SLOR THE DIFFERENTIAL EQUATIONS
C         THAT CONSTITUTE THOMPSON-THAMES-MASTINES METHOD OF GENERATING
C         GRIDS.
C
      DIMENSION X(140,80),Y(140,80)
      DIMENSION A(140),B(140),C(140),D(140),F(140),G(140)
C
      KMM=KMAX-1
      JMM=JMAX-1
      ICOUNT=0
C
2     ICOUNT=ICOUNT+1
      RSUM=0.
C
      DO 1 K=2,KMM
C
      DO 3 J=2,JMM
      XXD=(X(J+1,K)-X(J-1,K))*0.5
      XED=(X(J,K+1)-X(J,K-1))*0.5
      YXD=(Y(J+1,K)-Y(J-1,K))*0.5
      YED=(Y(J,K+1)-Y(J,K-1))*0.5
      AD=XED**2+YED**2
      BD=XXD*XED+YXD*YED
      GD=XXD**2+YXD**2
      XXED=(X(J+1,K+1)-X(J+1,K-1)-X(J-1,K+1)+X(J-1,K-1))*0.25
      YXED=(Y(J+1,K+1)-Y(J+1,K-1)-Y(J-1,K+1)+Y(J-1,K-1))*0.25
      BD=-2.0*BD
      A(J)=AD
      B(J)=-AD-AD-GD-GD
      C(J)=AD
      F(J)=-BD*XXED-GD*(X(J,K+1)+X(J,K-1))
3     G(J)=-BD*YXED-GD*(Y(J,K+1)+Y(J,K-1))
C
      F(2)=F(2)-A(2)*X(1,K)
      G(2)=G(2)-A(2)*Y(1,K)
      F(JMM)=F(JMM)-C(JMM)*X(JMAX,K)
      G(JMM)=G(JMM)-C(JMM)*Y(JMAX,K)
C
      CALL TRIB(A,B,C,D,F,2,JMM)
      CALL TRIB(A,B,C,D,G,2,JMM)
```

```
C
      DO 4 J=2,JMM
      XC=OMEGA*(F(J)-X(J,K))
      YC=OMEGA*(G(J)-Y(J,K))
      RSUM=RSUM+ABS(XC)+ABS(YC)
      X(J,K)=X(J,K)+XC
4     Y(J,K)=Y(J,K)+YC
   1 CONTINUE
C
      WRITE(6,100)RSUM,ICOUNT
      IF(ICOUNT.LT.MAXIT)  GO TO  2
C
      WRITE(6,121)
      DO 24 J=1,JMAX,10
      WRITE(6,115)J
      WRITE(6,117)  (X(J,K),K=1,KMAX)
      WRITE(6,116)J
24    WRITE(6,117)  (Y(J,K),K=1,KMAX)
C
      RETURN
C
100   FORMAT(20H SUM OF RESIDUALS = ,F20.10,
     1 7H AFTER ,I5,12H ITERATIONS.)
115   FORMAT(/31H X≥S FOR CONSTANT XI LINE AT J=,I5)
116   FORMAT(/31H Y≥S FOR CONSTANT XI LINE AT J=,I5)
117   FORMAT(10E13.6)
121   FORMAT(/48H AFTER THOMPSON-SOLVER, BEFORE FINAL CLUSTERING:)
      END
```

Subroutine CLUSTR applies along each line of constant ξ, the exponential clustering transformation discussed at length in the main body of this paper.

```
*DECK CLUSTR
      SUBROUTINE CLUSTR(JMAX,X,Y,DY2,KMAX)
C
C        THIS SUBROUTINE APPLIES TNE EXPONENTIAL CLUSTERING
C        TO THE LINES OF CONSTANT XI.
C
      DIMENSION X(140,80),Y(140,80)
      DIMENSION T(80),S(80),XXX(80),YYY(80)
C
      KMAX = KMAX-1
      KMM = KMAX-1
C
      DO 20 J=1,JMAX
C
      T(1) = 0.
      DO 16 K=2,KMAX
   16 T(K) = T(K-1) + SQRT( ( X(J,K) -X(J,K-1))**2 +( Y(J,K)-Y(J,K-1))
     1  **2)
C
      EPSY2=EPSIL(T(KMAX),0.0,DY2,KMAX,0.001,100,1,J)
C
      S(1)=0.0
      N=-1
      DO 15 K=2,KMAX
      N=N+1
   15 S(K)=S(K-1)+DY2*(1.0+EPSY2)**N
C
      DO 17 K=1,KMAX
      XXX(K) = X(J,K)
   17 YYY(K) = Y(J,K)
C
      PTL = 0.
      DO 18 K=2,KMM
   18 CALL TAINT(T,XXX,S(K),X(J,K),KMAX,2,NER,PTL)
C
      PTL = 0.
      DO 19 K=2,KMM
   19 CALL TAINT(T,YYY,S(K),Y(J,K),KMAX,2,NER,PTL)
C
   20 CONTINUE
C
      WRITE(6,120)
      DO 23 J=1,JMAX
      WRITE(6,115)J
      WRITE(6,117)  (X(J,K),K=1,KMAX)
      WRITE(6,116)J
   23 WRITE(6,117)  (Y(J,K),K=1,KMAX)
      RETURN
```

```
C
120     FORMAT(/24H AFTER FINAL CLUSTERING:)
115     FORMAT(/31H X2S FOR CONSTANT XI LINE AT J=,I5)
116     FORMAT(/31H Y2S FOR CONSTANT XI LINE AT J=,I5)
117     FORMAT(10E13.6)
C
        END
```

Function EPSIL applies a Newton-Raphson iterative technique to find a value of ε for use in the exponential stretching transformations given by equations (11), (12), (18), and (19).

```
      FUNCTION EPSIL(FMX,FMIN,DFM,NPT,FPCC,ICC,KEY,NCALL)
C
C        THIS SUBROUTINE APPLIES A NEWTON-RAPHSON ROOT-FINDING
C        TECHNIQUE TO FIND A VALUE OF EPSILON FOR A PARTICULAR USE
C        OF THE EXPONENTIAL STRECHING TRANSFORMATION.
C
      DIMENSION R(140)
C
      FMXL=FMX
      FMINL=FMIN
      DFML=DFM
      FPCCL=FPCC
      ICCL=ICC
C
      GO TO (1,2),KEY
C
1     FNPTM2=NPT-2
      IF(NCALL.EQ.1)  EPS=(FMXL/DFML)**(1.0/FNPTM2)-1.0
C
      DO 3 NIT=1,ICCL
      EP1=EPS+1.0
      EP1TN=EP1**FNPTM2
      REPS=1.0/EPS
      DFMOE=DFML*REPS
      F=FMXL-FMINL-DFMOE*(EP1TN*EP1-1.0)
      IF(ABS(F).LT.FPCCL)  GO TO 4
      DFMOE2=DFMOE*REPS
      FPN=DFMOE2*(1.0+EP1TN*(EPS*FNPTM2-1.0))
      EPS=EPS+F/FPN
3     CONTINUE
      GO TO 5
C
2     NPTM=NPT-1
      IF(NCALL.EQ.1)
     1 EPS=((FMXL/DFML)**(1.0/(NPT-2))-1.0)*SQRT(FLOAT(NPTM))
      DO 6 L=1,NPTM
6     R(L)=1.0/SQRT(FLOAT(L))
C
      DO 7 NIT=1,ICCL
      SUM1=0.0
      SUM2=0.0
      DO 8 L=1,NPTM
      FLM2=L-2
      FACT1=1.0+EPS*R(L)
      FACT2=FACT1**FLM2
      SUM1=SUM1+FACT2*FACT1
8     SUM2=SUM2+(L-1)*FACT2*R(L)
      F=FMXL-FMINL-DFML*SUM1
      IF(ABS(F).LT.FPCCL)  GO TO 4
      FPN=DFML*SUM2
      EPS=EPS+F/FPN
7     CONTINUE
C
5     EPSIL=EPS
      WRITE(6,100)
      RETURN
C
```

```
4       EPSIL=EPS
        WRITE(6,101)  EPSIL,F,NIT
        RETURN
C
100     FORMAT(/42H EXCEEDED MAX. NO. OF ITERATIONS IN EPSIL.)
101     FORMAT(/7H EPSIL=,F12.5,5X,7H AND F=,F12.5,5X,7H AFTER ,I3,
      *  12H ITERATIONS.)
C
        END
```

Subroutine TAINT is a utility subroutine that inputs a tabulated function of one independent variable, then outputs the value of the function at a given value of the independent variable by passing a polynomial of arbitrary order (≤ 9) through the nearest input points. It is used for second-order interpolation.

Subroutine TRIB solves a tridiagonal system of linear equations using the so-called Thomas algorithm (i.e., Gaussian elimination or L-U decomposition).

```
      SUBROUTINE TAINT(XTAB,FTAB,X,FX,N,K,NER, MON)
C
C         SYSTEM LIBRARY SUBROUTINE TAINT FOR POLYNOMIAL INTERPOLATION
C         OF A TABULATED FUNCTION.
C
      DIMENSION XTAB(1),FTAB(1),T(10),C(10)
CPS0400  TAINT SUBROUTINE- IN FORTRAN II.
      IF (N - K) 1,1,2
    1 NER=2
      RETURN
    2 IF (K-9) 3,3,1
    3 IF ( MON) 4,4,5
    5 IF ( MON-2) 6,7,4       .
    4 J=0
      NM1=N-1
      DO 8 I=1,NM1
      IF (XTAB(I)-XTAB(I+1)) 9,11,10
   11 NER=3
      RETURN
    9 J=J-1
      GO TO 8
   10 J=J+1
    8 CONTINUE
      MON=1
      IF (J) 12,6,6
   12 MON=2
    7 DO 13 I=1,N
      IF (X-XTAB(I)) 14,14,13
   14 J=I
      GO TO 18
   13 CONTINUE
      GO TO 15
    6 DO 16 I=1,N
      IF (X-XTAB(I)) 16,17,17
   17 J=I
      GO TO 18
   16 CONTINUE
   15 J=N
   18 J=J-(K+1)/2
      IF (J) 19,19,20
   19 J=1
   20 M=J+K
      IF (M-N) 21,21,22
```

20

```
   22 J=J-1
      GO TO 20
   21 KP1=K+1
      JSAVE=J
   26 DO 23 L=1,KP1
      C(L)=X-XTAB(J)
      T(L)=FTAB(J)
   23 J=J+1
      DO 24 J=1,K
      I=J+1
   25 T(I)=(C(J)*T(I)-C(I)*T(J))/(C(J)-C(I))
      I=I+1
      IF (I-KP1) 25,25 ,24
   24 CONTINUE


      FX=T(KP1)
      NER=1
      RETURN
      END
```

```
      SUBROUTINE TRIB(A,B,C,X,F,NL,NU)
      DIMENSION A(2),B(2),C(2),X(2),F(2)
C
C        THIS SUBROUTINE SOLVES A TRI-DIAGONAL SYSTEM OF LINEAR
C        EQUATIONS.
C
      X(NL)=C(NL)/B(NL)
      F(NL)=F(NL)/B(NL)
      NLP1 = NL +1
      DO 1 J=NLP1,NU
      Z=1./(B(J)-A(J)*X(J-1))
      X(J)=C(J)*Z
    1 F(J)=(F(J)-A(J)*F(J-1))*Z
      NUPNL=NU+NL
      DO 2 J1=NLP1,NU
      J=NUPNL-J1
    2 F(J)=F(J)-X(J)*F(J+1)
      RETURN
      END
```

SC4020 PLOT SUBROUTINES

Subroutines PLAWT, AXIS, LABEL, PLOT, and TITLE are used to illustrate the grids on the SC4020 plotter. They contain calls to the standard SC4020 software subroutines; thus, the SC4020 software subroutines must be made available to the loader. If the user's computer installation does not include an SC4020, the five subroutines listed above should be removed from the deck; calls to subroutines READIN, PLAWT, and EOFTV should be removed from the main program; and data card 7 should be deleted.

```
      SUBROUTINE PLAWT(N,M,X,Y,XMAX,XMIN,YMAX,YMIN)
C
C         THIS SUBROUTINE IS ONE OF THE SUBROUTINES THAT PLOTS GRIDS
C         ON THE SC4020 PLOTTER.
C
      DIMENSION X(140,80),Y(140,80),XX(80),YY(80)
      COMMON/TCOM/ITITLE(8)
C
'C        READJUST PLOT LIMITS SO AS TO AVIOD A STRECHED PLOT.
      XDIF=XMAX-XMIN
      YDIF=YMAX-YMIN
      IF(XDIF.LT.YDIF)  GO TO 4
      XDIFH=XDIF*0.5
      YMID=(YMAX+YMIN)*0.5
      YMX=YMID+XDIFH
      YMN=YMID-XDIFH
      XMX=XMAX
      XMN=XMIN
      GO TO 5
4     YDIFH=YDIF*0.5
      XMID=(XMAX+XMIN)*0.5
      XMX=XMID+YDIFH
      XMN=XMID-YDIFH
      YMX=YMAX
      YMN=YMIN
5     CONTINUE
C
C         PLOT THE LINES.
      CALL AXIS(XMN,XMX,YMN,YMX,0,0)
      CALL TITLE(ITITLE,80)
      DO1J=1,M
1     CALL PLOT(X(1,J),Y(1,J),N,0)
      DO2I=1,N
      DO3J=1,M
      XX(J)=X(I,J)
    3 YY(J)=Y(I,J)
2     CALL PLOT(XX,YY,M,0)
      RETURN
      END
```

```
            SUBROUTINE AXIS(XMIN,XMAX,YMIN,YMAX,NXINTS,NYINTS)
C
C         THIS SUBROUTINE IS ONE OF THOSE THAT PLOTS GRIDS
C         ON THE SC4020 PLOTTER.
C
C      THIS SUBROUTINE DRAWS THE X AND Y AXES, MARKS OFF THE
C      INTERVALS, SETS THE SCALE VALUES, AND MOVES THE FILM TO A NEW
C      FRAME.
C
C      XMIN IS THE MINIMUM VALUE OF THE X (HORIZONTAL) VARIABLE.
C      XMAX IS THE MAXIMUM VALUE OF THE X VARIABLE.
C      YMIN IS THE MINIMUM VALUE OF THE Y (VERTICAL) VARIABLE.
C      YMAX IS THE MAXIMUM VALUE OF THE Y VARIABLE.
C      NXINTS IS THE NUMBER OF INTERVALS INTO WHICH THE X AXIS IS TO
C         BE DIVIDED.
C      NYINTS IS THE NUMBER OF INTERVALS INTO WHICH THE Y AXIS IS TO
C         BE DIVIDED.
C
C      DO INITIALIZATION
       CALL CAMRAV(35)
       CALL SMALLV
       CALL FRAMEV(0)
       NX=NXINTS
       NY=NYINTS
C      IF NX OR NY .LE. 0 MAKE THEM EQUAL TO 1
       IF (NX .LE. 0) NX=1
       IF (NY .LE. 0) NY=1
C      SET UP VALUES TO BE USED FOR MARGINS
       ML=123
       MR=923
       MB=123
       MT=923
C      DRAW X AXIS
       CALL LINEV(ML,MB,MR,MB)
C      DRAW Y AXIS
       CALL LINEV(ML,MB,ML,MT)
C      DETERMINE INCREMENTS FOR TIC MARKS
       DX=(XMAX-XMIN)/NX
       DY=(YMAX-YMIN)/NY
C      SCALE X AND Y VALUES
       CALL XSCALV(XMIN,XMAX,ML,100)
       CALL YSCALV(YMIN,YMAX,MB,100)
C      DRAW TIC MARKS ON THE X AXIS
       CALL LINRV(1,MB-20,MB-5,MB+5,XMIN,XMAX,DX,0,-1,-3,8)
C      DRAW TIC MARKS ON Y AXIS
       CALL LINRV(2,ML- 90,ML-5,ML+5,YMIN,YMAX,DY,0,-1,-3,10)
       RETURN
       END
```

```
      SUBROUTINE LABEL(IXLABL,NX,IYLABL,NY)
C
C         THIS SUBROUTINE IS ONE OF THOSE THAT PLOTS GRIDS
C         ON THE SC4020 PLOTTER.
C
C     THE MAXIMUM NUMBER OF CHARACTERS ALLOWED FOR THE X LABEL IS 54.
C     THE MAXIMUM NUMBER OF CHARACTERS ALLOWED FOR THE Y LABEL IS 38.
      INX=IABS(NX)
      INY=IABS(NY)
      IF (INX .GT. 54) INX=54
      IF (INY .GT. 38) INY=38
      IX=500-(INX/2)*18
      IY=510+(INY/2)*26
C     LABEL THE X AXIS
      CALL RITE2V(IX,60,1000,90,1,INX,1,IXLABL,NLAST)
C     LABEL THE Y AXIS
      DO 100 I=1,INY
            ITEMP=IY-26*(I-1)
            CALL RITE2V(10,ITEMP,50,90,1,1,I,IYLABL, NLAST)
  100 CONTINUE
      RETURN
      END




      SUBROUTINE PLOT(X,Y,NBR,NSYM)
C
C         THIS SUBROUTINE IS ONE OF THOSE THAT PLOTS GRIDS
C         ON THE SC4020 PLOTTER.
C
      DIMENSION X(1),Y(1),MARKPT(5)
      DATA MARKPT /42,16,55,38,44/
C     SYMBOLS ARE, IN ORDER:  . + X O *
      IF (NSYM .GT. 0 .AND. NSYM .LE. 5) GO TO 100
      J=IABS(NBR)-1
      DO 110 I=1,J
            CALL LINEV( IXV(X(I)), IYV(Y(I)), IXV(X(I+1)),IYV(Y(I+1)) )
  110 CONTINUE
      RETURN
  100 CALL APLOTV(NBR,X,Y,1,1,1,MARKPT(NSYM))
      RETURN
      END
```

```
      SUBROUTINE TITLE(ITITLE,NCHARS)
C
C        THIS SUBROUTINE IS ONE OF THOSE THAT PLOTS GRIDS
C        ON THE SC4020 PLOTTER.
C
C     THE MAXIMUM NUMBER OF CHARACTERS ALLOWED IN THE TITLE IS 108.
C
      ICHARS=IABS(NCHARS)
      IF (ICHARS .GT. 108) ICHARS=108
      IF (ICHARS .GT. 54) IX=14
      IF (ICHARS .LE. 54) IX=510-(ICHARS/2)*18
      IY=990
      CALL RITE2V(IX,IY,1010,90,1,ICHARS,1,ITITLE,NLAST)
      RETURN
      END
```

Input data cards are as follows. All integers are read using 16I5 format. All floating-point variables are read using 8F10.0 format. All alphanumeric data are read using 8A10 format.

Card 1. Card 1 contains three integers.

Columns	Name	Description
1 to 5	JWAKE	Number of grid points in the x direction in the wake including the trailing edge point
6 to 10	KMAX	Number of grid points in the η direction
11 to 15	MAXIT	Number of iterations to be used in creating the unclustered grid

Card 2. Card 2 contains floating point numbers that define the physical size of the grid and the location of the airfoil.

Columns	Name	Description
1 to 10	XGMX	x-direction coordinate of rearward boundary
11 to 20	XGMN	x-direction coordinate of front boundary
21 to 30	YMAX	y-direction coordinate of top boundary
31 to 40	YMIN	y-direction coordinate of bottom boundary
41 to 50	XNOSE	x-direction coordinate of leading edge of airfoil
51 to 60	XTAIL	x-direction coordinate of trailing edge of airfoil

Card 3. Card 3 contains in columns 1 to 10 one floating-point number $DY1$, which is the physical spacing in the y direction on the rear boundary immediately above and below the wake.

Card 4. Card 4 contains floating-point numbers determining the location of points on the bottom-front-top boundary.

Columns	Name	Description
1 to 10	XORG	Location on the x axis of the special "clustering origin"
11 to 20	ETACD	Angle (in degrees), measured counterclockwise from the x axis, about which clustering should be performed
21 to 30	BETA	Parameter determining the strength of clustering. Zero (for equal-angular distribution) or values in the range 1 to 5 are recommended.

Card 5. Card 5 contains one floating-point number in columns 1 to 10, OMEGA, the relaxation parameter for the SLOR procedure in subroutine RELAX. Values must be between 0 and 2, with a typical value being 1.55.

Card 6. Card 6 contains one floating-point number in columns 1 to 10, the parameter $DY2$, which is the physical spacing in the η direction between the airfoil-wake boundary and the next line of constant η.

Card 7. Card 7, read by SC4020 subroutine READIN, contains up to 80 alphanumeric characters to be used for a comment on the title frame of the SC4020 plots.

Card 8. Card 8 contains up to 71 alphanumeric characters to be placed as a title on each of the four SC4020 plots that result from each run.

Card 9. Card 9 contains up to 80 alphanumeric characters describing the airfoil to be used. This description will appear on the printed output.

Cards 10, 11, ... All cards following card 9 contain in columns 1 to 10 and 11 to 20 two floating-point numbers, x and y, respectively, which are coordinates of points on the airfoil surface, defining the airfoil shape. The points should be ordered to begin and end at the trailing edge and proceed counterclockwise.

Two notes of caution with respect to the input data cards are in order for those installing this program on computers of manufacture other than Control Data. First, the cards defining the airfoil shape are read until an end-of-file is encountered. End-of-file conditions are handled differently by different computers. Second, alphanumeric data is stored 10 characters per word on Control Data machines. Care should be taken on machines having different word lengths.

SAMPLE DATA CASE

The data cards used to produce the grid shown in figure 5 follow:

```
    12    28   2C0
8.0        -8.0       8.0        -8.0       0.0        1.0
0.04

1.55
0.01
REESE SORENSON   NASA-AMES   STOP 5124   PH. 965-5125   TAPE CO
NACA 0012
NACA 0012
   1.000000   -.001251
    .981540   -.003816
    .959694   -.006776
    .929881   -.010690
    .892711   -.015378
    .848960   -.020639
    .799449   -.026286
    .745313   -.032084
    .687413   -.037853
    .626854   -.043365
```

.564684	-.048424
.501943	-.052817
.439612	-.056319
.378638	-.058748
.319904	-.059927
.264227	-.059712
.212357	-.058027
.164970	-.054863
.122669	-.050245
.085973	-.044279
.055327	-.037107
.031109	-.028868
.013637	-.019743
.003244	-.009929
-.000000	.000002
.003283	.009932
.013651	.019728
.031111	.028866
.055331	.037100
.085975	.044274
.122670	.050244
.164972	.054858
.212357	.058027
.264227	.059712
.319904	.059926
.378638	.058748
.439613	.056319
.501943	.052815
.564685	.048424
.626855	.043366
.687413	.037653
.745314	.032084
.799450	.026289
.848961	.020640
.892711	.015380
.929881	.010691
.959694	.006777
.981540	.003816
1.000000	.001252

REFERENCES

1. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies. J. Comp. Phys., vol. 15, no. 3, July 1974, pp. 299-319.

2. Thames, F. C.; Thompson, J. F.; and Mastin, C. W.: Numerical Solution of the Navier-Stokes Equations for Arbitrary Two-Dimensional Airfoils. NASA SP-347, 1975, Pt. 1, pp. 469-530.

3. Davy, W. C.; and Reinhardt, W. A.: Computation of Shuttle Nonequilibrium Flow Fields on a Parallel Processor. NASA SP-347, 1975, Pt. 2, pp. 1351-1366.
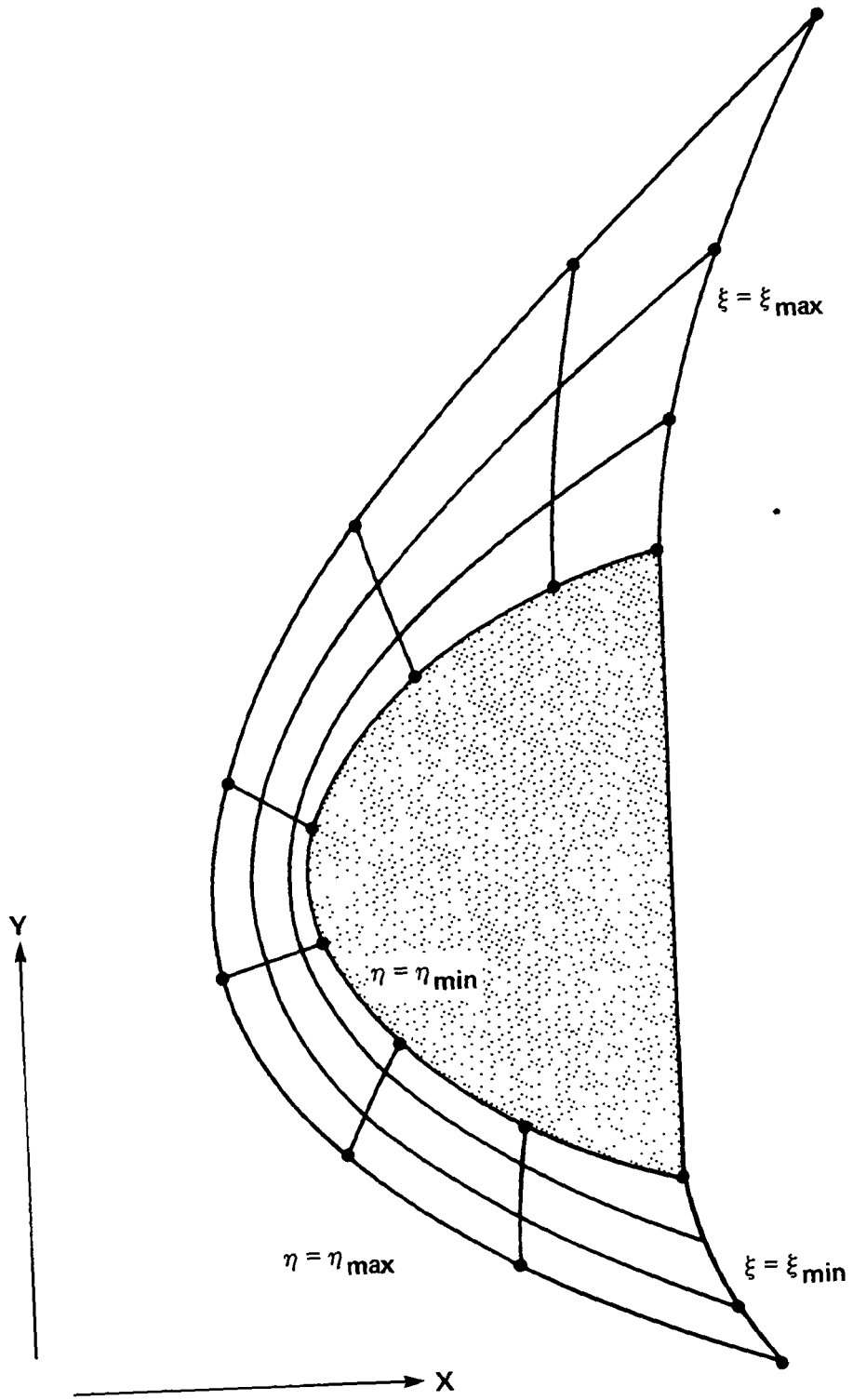
Figure 1.— Location of boundaries on a typical physical domain.
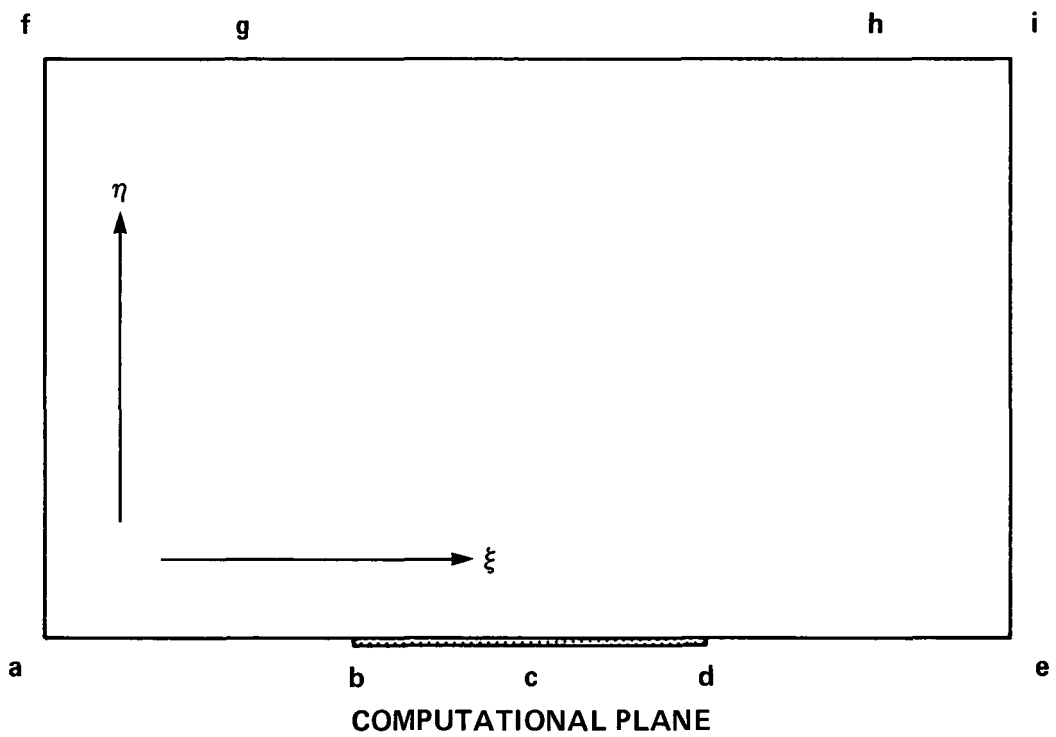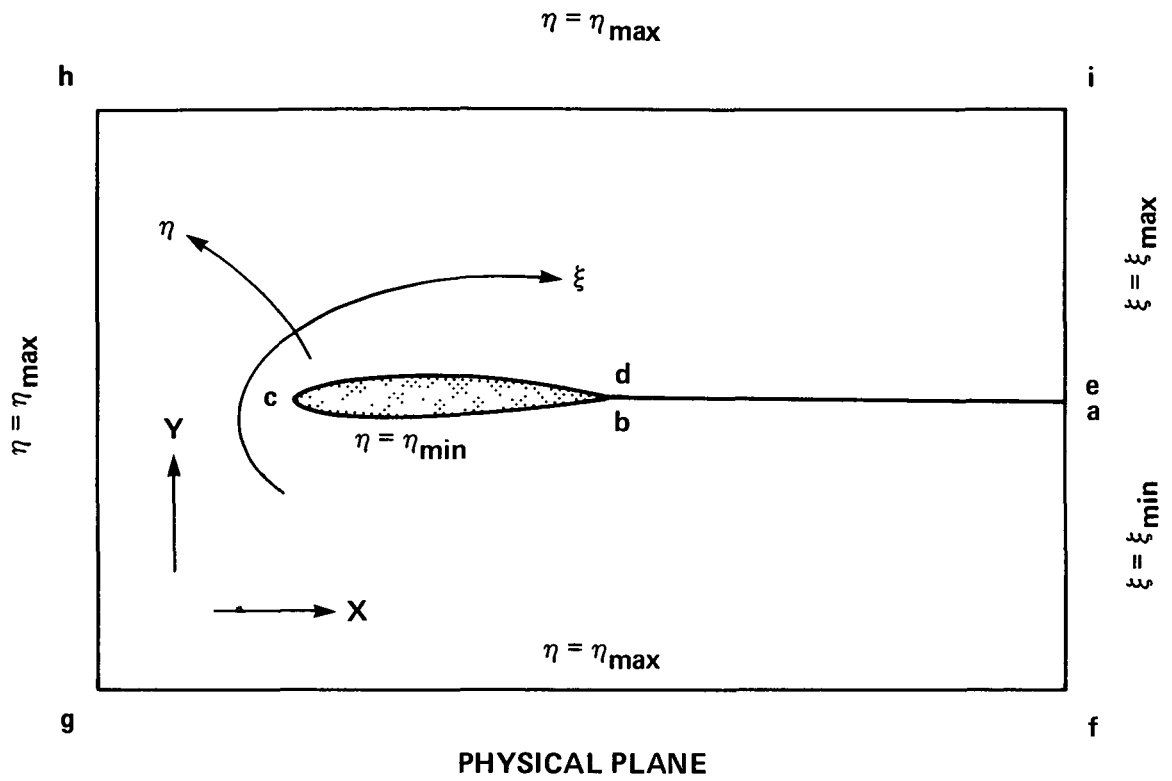
Figure 2.— Schematic showing boundary locations for one kind of airfoil grid generation.
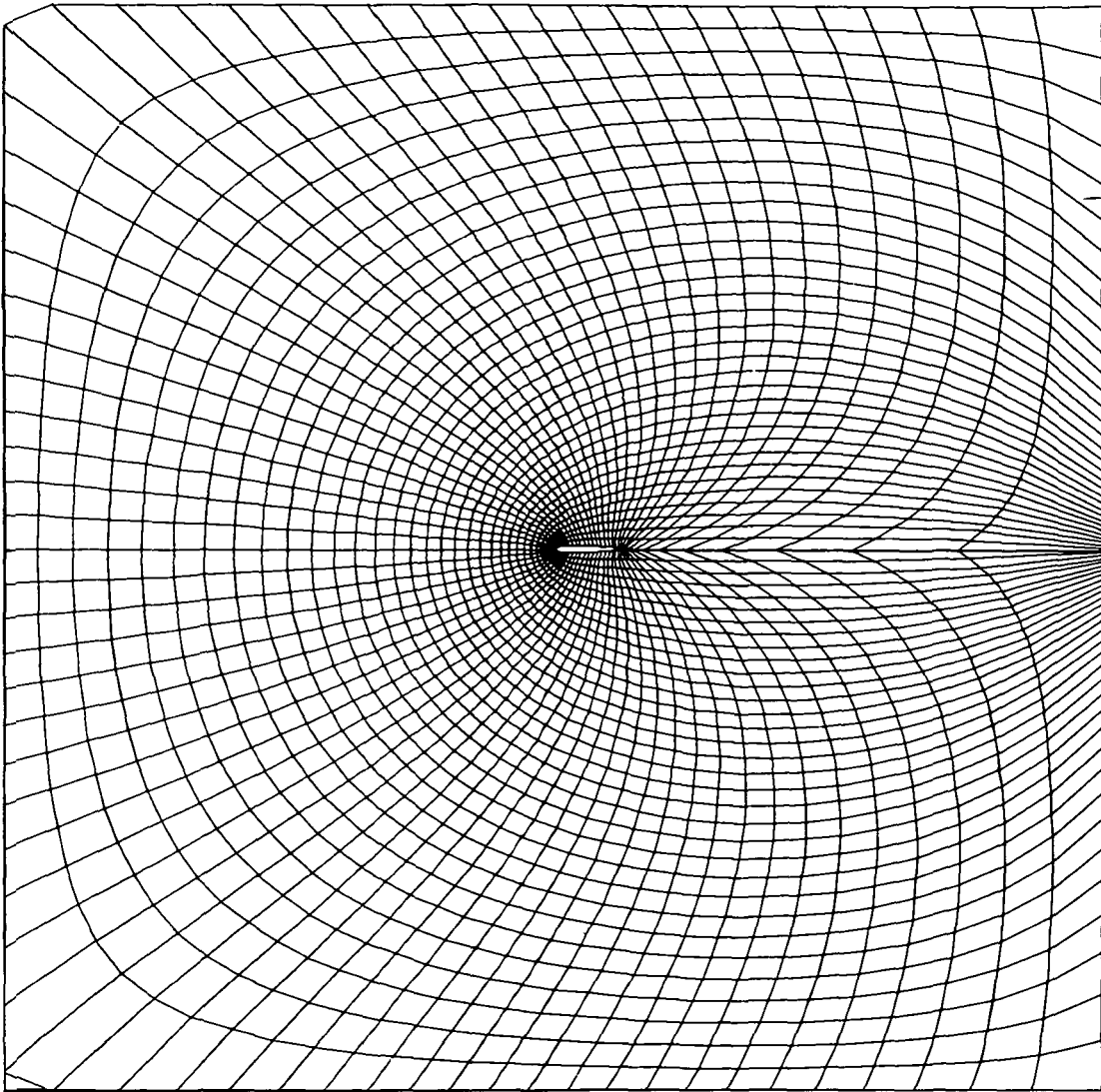
(a) Overview.

Figure 3.— Computed grid using equations (3) and (4) with specified boundaries.
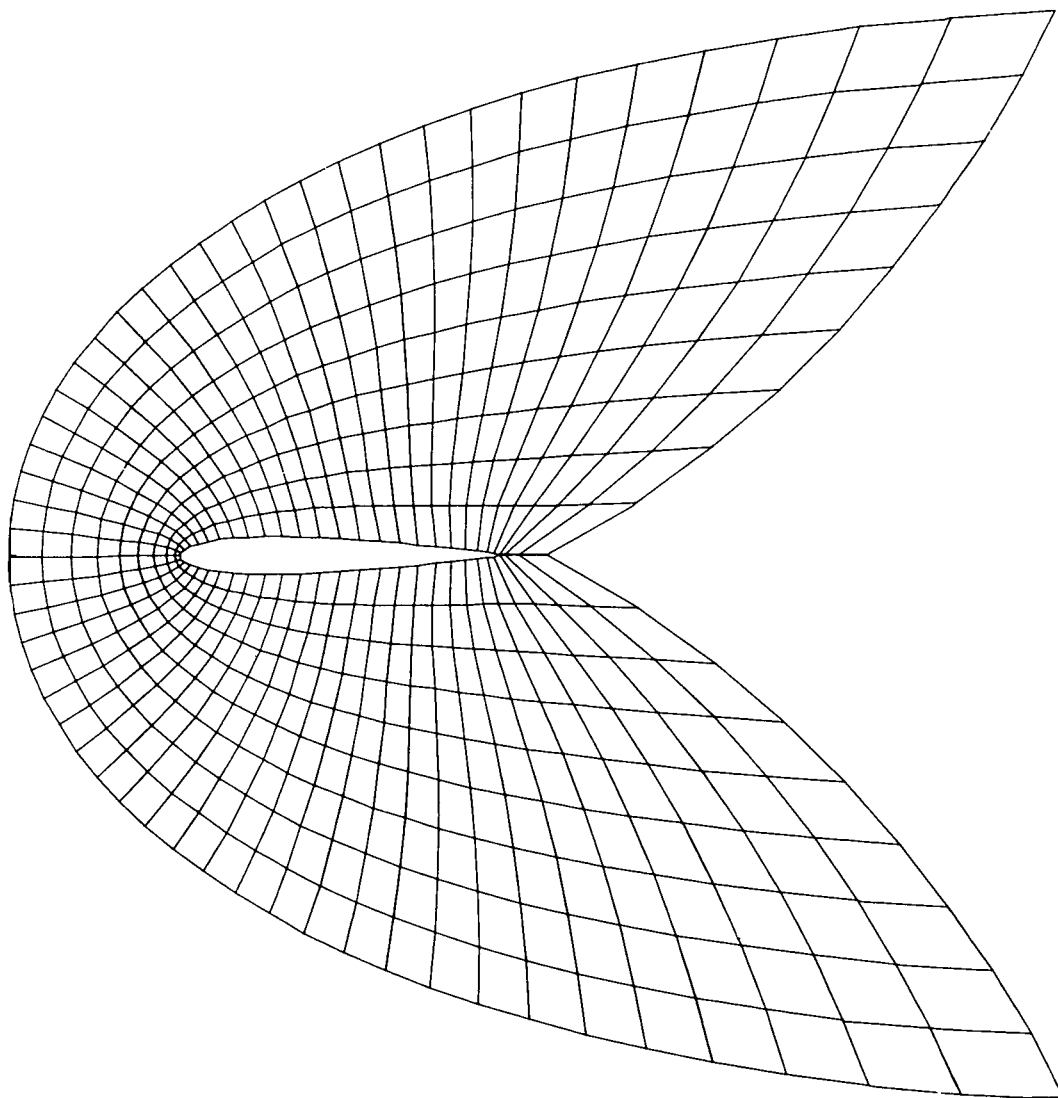
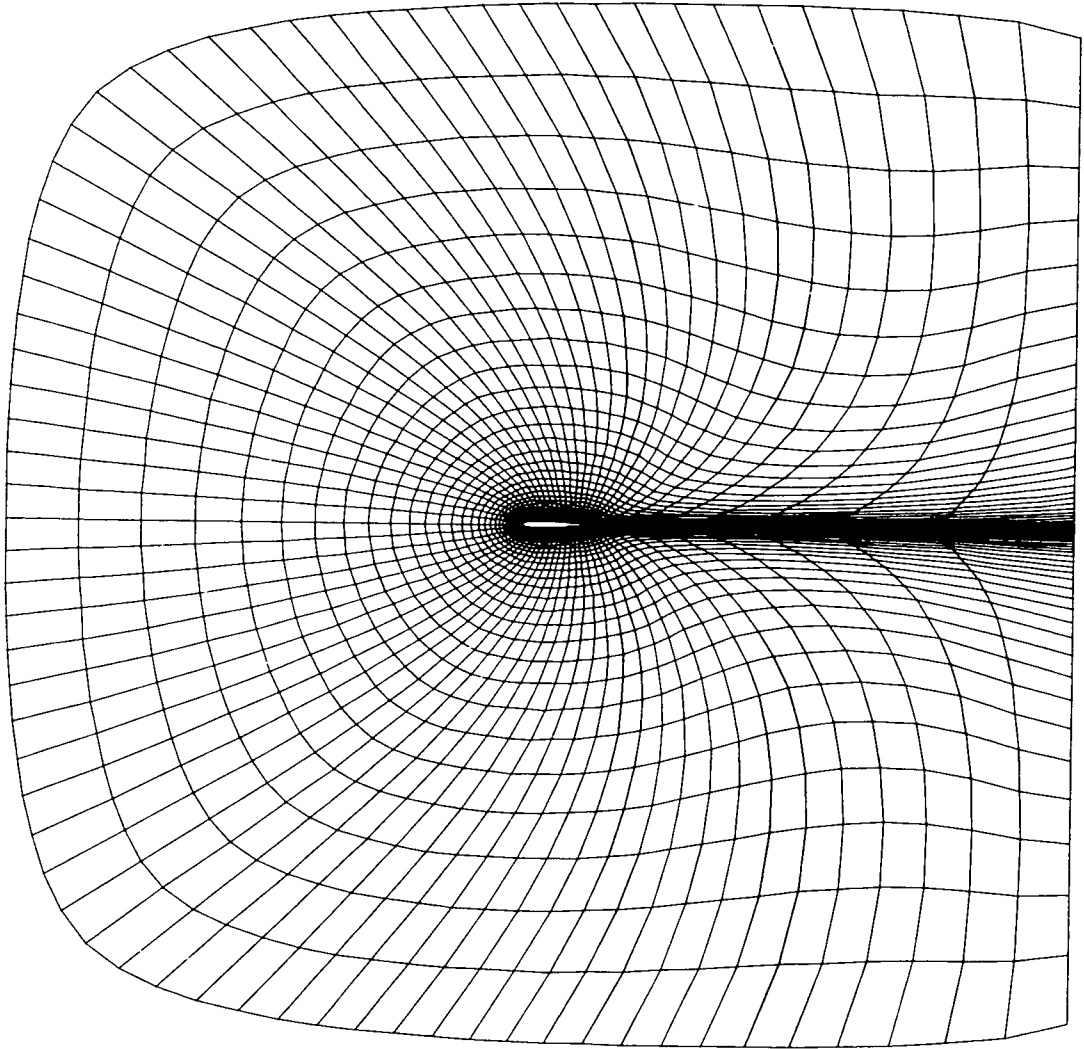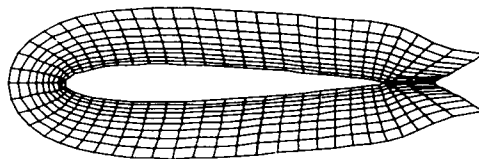(b) Grid detail near boundary.

Figure 3.— Concluded.
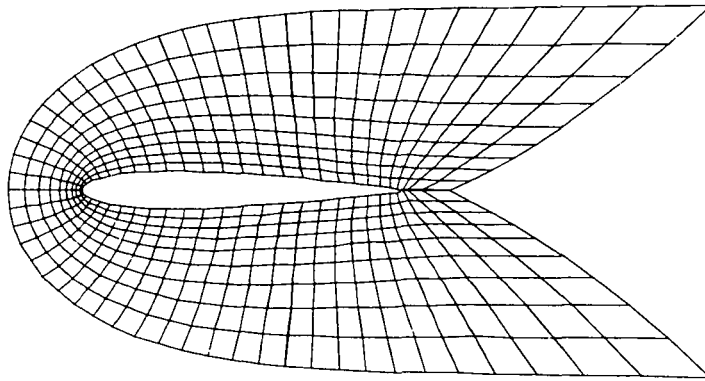
Figure 4.— Detail of computational grid using equations (7) and (8).

(a) Overview.

(b) Grid detail near boundary.

Figure 5.— Computational grid after application of clustering along lines of constant ξ.

1 Report No. NASA TM 73,252	2 Government Accession No	3. Recipient's Catalog No
4 Title and Subtitle SIMPLIFIED CLUSTERING OF NONORTHOGONAL GRIDS GENERATED BY ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS		5. Report Date August 1977
		6. Performing Organization Code
7 Author(s) Reese L. Sorenson and Joseph L. Steger		8 Performing Organization Report No A-6950
9. Performing Organization Name and Address Ames Research Center, NASA Moffett Field, Calif. 94035		10 Work Unit No 514-52-02-01
		11 Contract or Grant No
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		13 Type of Report and Period Covered Technical Memorandum
		14 Sponsoring Agency Code

15 Supplementary Notes

16 Abstract

    A simple clustering transformation is combined with the Thompson, Thames, and Mastin (TTM) method of generating computational grids to produce controlled mesh spacings. For various practical grids, the resulting hybrid scheme is easier to apply than the inhomogeneous clustering terms included in the TTM method for this purpose. The technique is illustrated in application to airfoil problems, and listings of a FORTRAN computer code for this usage are included.

17. Key Words (Suggested by Author(s)) Computational fluid dynamics Numerical grid generation	18 Distribution Statement Unlimited STAR Category - 64	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of Pages 37	22. Price* $3.75

THIRD-CLASS BULK RATE

U.S. MAIL

NASA

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return

26 APR 80 S   C/C. Lee 343/32²/265/29991   "1 JUL 31 S

8. NOV 81 MO   CC Lee 343/32²/36491   23 DEC 83 S