

AD-A090 829

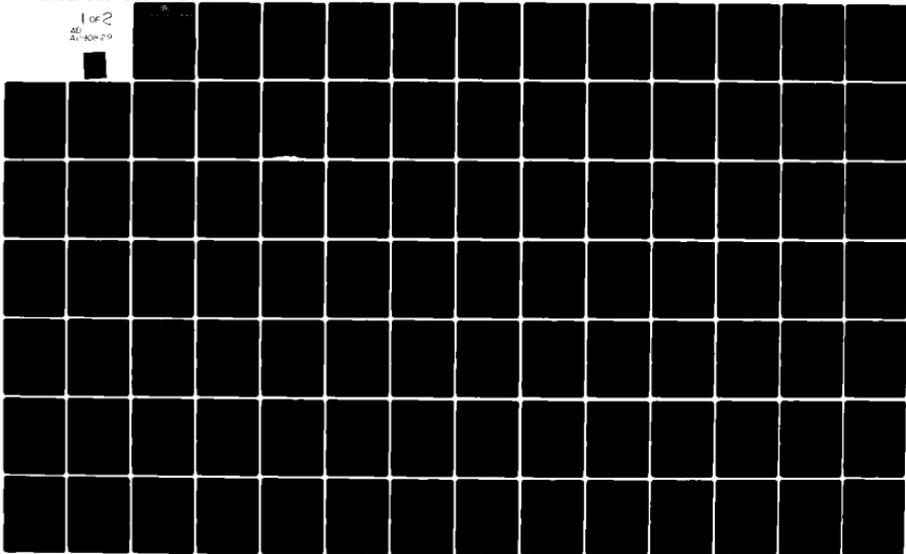
FLOW RESEARCH CO KENT WA
DEVELOPMENT OF FINITE-VOLUME METHODS FOR THREE-DIMENSIONAL TRAN--ETC(U)
AUG 80 J E MERCER, W JOU, D A CAUGHEY
FLOW-RR-166

F/G 20/4
N00014-78-C-0079

NL

UNCLASSIFIED

1 of 2
40
21-508-20



LEVEL

12
P.S.

FLOW RESEARCH COMPANY

A DIVISION OF FLOW INDUSTRIES, INC.

AD 11090829

DDC FILE COPY



DTIC
ELECTE
S OCT 27 1980 **D**
A

HEADQUARTERS
21414 - 68th Avenue South
Kent, Washington 98031 (206) 854-1310
Seattle Ex. 622-1500 TWX 910-447-2762

DISTRIBUTION STATEMENT A
Approved for release;
Distribution unlimited

80 9 26 046

12 117

14 FLOW-RR-166

9 Flow Research Report ~~166~~ on P. 2

Development of Finite-Volume Methods for Three-Dimensional Transonic Flows*

By

John E. Mercer
Wen-Huei Jou
David A. Caughey
Antony Jameson
David Nixon

11 August 1980

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution unlimited

Flow Research Company
A Division of Flow Industries, Inc.
21414 - 68th Avenue South
Kent, Washington 98031
(206) 872-8500

*This work was supported by the Office of Naval Research and NASA Ames Research Center under ONR Contract N00014-78-C-0079

390 404

Table of Contents

	Page
Abstract	i
List of Figures	iii
1. Introduction	1
2. Finite-Volume Algorithm	4
3. Field-Point Calculation	12
4. Wing-Body Code Development	14
4.1 Introduction	14
4.2 Grid Generation	15
4.3 Boundary Conditions	19
4.4 Check Cases	21
5. Slit Transformations	29
6. Consistent Vortex Wake Model	41
7. Embedded Tail-Plane Mesh	45
7.1 Introduction	45
7.2 Embedded Tail-Plane Mesh	46
7.3 Grid Generation	49
7.3.1 Wraparound C-Type Lines	49
7.3.2 X = Constant Lines	50
7.4 Finite-Volume Computation With Tail-Plane Mesh	51
8. Conclusions	56
References	57
Appendix: A Preliminary Study of an Overlapping Grid-Generation System	A-i

List of Figures

	Page
Figure 1. Mapping from Physical Space to Computational Space	5
Figure 2. Primary and Secondary Cell in Computational Plane	7
Figure 3. Decoupled Solution Arising from Difference Operator	8
Figure 4. Shift in ϕ_X Used to Recouple Solutions	10
Figure 5. Field-Point Velocities Along $X = 0$	13
Figure 6. Coordinate System for Wing-Body Code	16
Figure 7. Sequential Mappings from Physical to Computational Space	18
Figure 8. Sample Coarse Mesh	20
Figure 9. Coarse Grid on Learjet	23
Figure 10. Cordwise Pressure Distribution on Learjet	24
Figure 11. Comparison of FLO 30 with Experiment for A-7	25
Figure 12. Comparison of Computed and Experimental Cordwise Pressure Distribution on Modified A-7 Model	26
Figure 13. Effect of Fuselage on Computed Spanwise Loading of Modified A-7 Model	28
Figure 14. Grid Plots	30
Figure 15. Zero Angle of Attack	32
Figure 16. Two-Degree Angle of Attack	36
Figure 17. Tail-On-Fuselage	47
Figure 18. High-Tail	48
Figure 19. Tail-Plane Mesh	52
Figure 20. Detail of the Mesh	53
Figure 21. Finite-Volume Computation on Tail-Plane Mesh (No Lift)	54
Figure 22. Finite-Volume Computation on Tail-Plane Mesh (With Lift)	55

1. Introduction

The development of transonic calculations has progressed rapidly over the past several years. Small-disturbance codes have been developed (Bailey and Ballhaus, 1975; Mason et al., 1978; Boppe and Stern, 1980) which can model a wide range of geometrical configurations and even account for viscous effects. Small-disturbance codes do have limitations, though. In general they cannot accurately treat the flow around leading edges (Hinson and Burdges, 1980), especially for advanced airfoil designs. They also have difficulty treating moderate to strong shocks.

In order to overcome these deficiencies, transonic full-potential codes have been developed. One of the most promising schemes in terms of extension to arbitrary configurations is the Jameson-Caughey finite-volume algorithm (Jameson and Caughey, 1977). This scheme decouples the geometry from the differencing algorithm so that the main difficulty in applying the method is to develop a geometry package which can generate a grid that adequately defines the configuration and has certain smoothness requirements.

The work reported in this document has been sponsored jointly by the Office of Naval Research and NASA Ames Research Center. It covers the second phase of an effort to develop a computer code to model arbitrary wing-body configurations.

The first phase of the effort was reported in Caughey, Jameson, and Nixon (1979). This work produced two codes, one using a quasi-conservative differencing formulation of the finite-volume scheme developed by David Caughey and the other a fully conservative differencing of the scheme developed by Antony Jameson. The quasi-conservative code, FLO 25, can treat wing-body combinations where the body has infinite length extending upstream and downstream of the physical fuselage. The fully-conservative code, FLO 28, can also treat wing-body combinations and can handle finite-length fuselages. The two codes use entirely different techniques to generate the grids. The grid generated by FLO 25 is entirely boundary conforming while the grid generated by FLO 28 does not conform to the fuselage. The latter grid system causes pressure oscillations in the fuselage-wing junction area (Caughey and Jameson, 1979).

The goal of the Phase II work reported here was to overcome the deficiencies noted in the codes developed in the Phase I work and to demonstrate the resultant code on test cases. Another goal was to investigate grid-generation schemes which might readily be extended to wing-body-tail configurations.

This report covers the wing-body code development and an investigation of grid systems which could model more elaborate configurations, such as wing-body-tail configurations. A general summary of the Phase II wing-body code development was presented in Mercer and Murman (1980), in which code comparisons with experimental data are given.

Previous work led to the development of two wing-body codes, FLO 28, which is based on Joukowski transformation for the fuselage representation, and FLO 30, which is based on a cylindrical mapping for the fuselage representation. The present work evaluated the two codes and selected one (FLO 30) for further development and demonstration. As part of the development, a reconsideration of the vortex wake model was accomplished to determine the adequacy of the original assumptions. Additionally, a special subroutine was developed so that velocities and pressures at any point in the computational field could be printed.

Along with the wing-body code development was an investigation of grid systems. The grid work covered two separate areas. The first was aimed at obtaining a better wing-body mesh; the second was aimed at developing a tail mesh compatible with the wing-body mesh. The need for the first area was diminished when the cylindrical mesh system in FLO 30 was adopted. The work did demonstrate, however, that two mesh systems could be overlapped and the numerics would converge. A description of the method is included in the appendix. Another effort devoted to improving the basic wing-body mesh was an investigation of slit transformations. This work showed that a direct application of the mesh is not feasible. The mesh does offer possibilities of modeling more complex geometries provided that the inherent singularities associated with the mesh could be overcome.

The tail mesh generation was approached using two different philosophies. The first is based on a ratio of inner to outer boundary coordinates, while the second is based on a family of super ellipses and

shearing. The latter approach eliminated problems in dealing with mesh-line kinks and was found to be more desirable. Both methods are reported here. The first method is presented in the appendix.

All the work described in the appendix was performed by Dr. David Nixon while at Flow Research Company.

2. Finite-Volume Algorithm

The finite-volume algorithm assumes that the six-sided elements comprising the mesh in the physical space can be transformed to cubes in the computational space. The mapping to each cube is assumed to be local so that transformations can be based on the physical values of the vertices of the six-sided elements. The location of the vertices (or mesh points) in physical space may be determined by any suitable procedure, and two specific examples are given in following sections. The mapped cubes have trilinear variations of coordinates ranging from $-\frac{1}{2}$ to $\frac{1}{2}$ (Figure 1), and the potential is assumed to vary trilinearly within each cell. With the coordinate variation assumption, the corresponding points in the physical space can be located from points in the computational space by the local trilinear mapping formula:

$$x = 8 \sum_{k=1}^8 x_k \left(\frac{1}{4} + X_k X \right) \left(\frac{1}{4} + Y_k Y \right) \left(\frac{1}{4} + Z_k Z \right), \quad (1)$$

where X_k , Y_k , and Z_k are the mapped vertices of the cubes ($\pm\frac{1}{2}$) and the x_k terms are the corresponding physical values. There are equivalent formulas for y , z , and ϕ , the velocity potential. With this mapping, continuity of x , y , z , and ϕ is preserved at the cell boundaries. The mapping also allows derivatives of the transformation and potential to be evaluated anywhere in the cell.

The flow equation that we wish to solve is the conservation relation:

$$\frac{\partial}{\partial x} (\rho u) + \frac{\partial}{\partial y} (\rho v) + \frac{\partial}{\partial z} (\rho w) = \frac{\partial}{\partial x_i} (\rho u_i) = 0. \quad (2)$$

The finite-volume algorithm is a conservative differencing scheme which satisfies the above equation using the cubical cells in the computational space. Density is computed from the isentropic relation:

$$\rho = \left[1 + \frac{\gamma-1}{2} M_\infty^2 (1 - q^2) \right]^{\frac{1}{\gamma-1}} \quad (3)$$

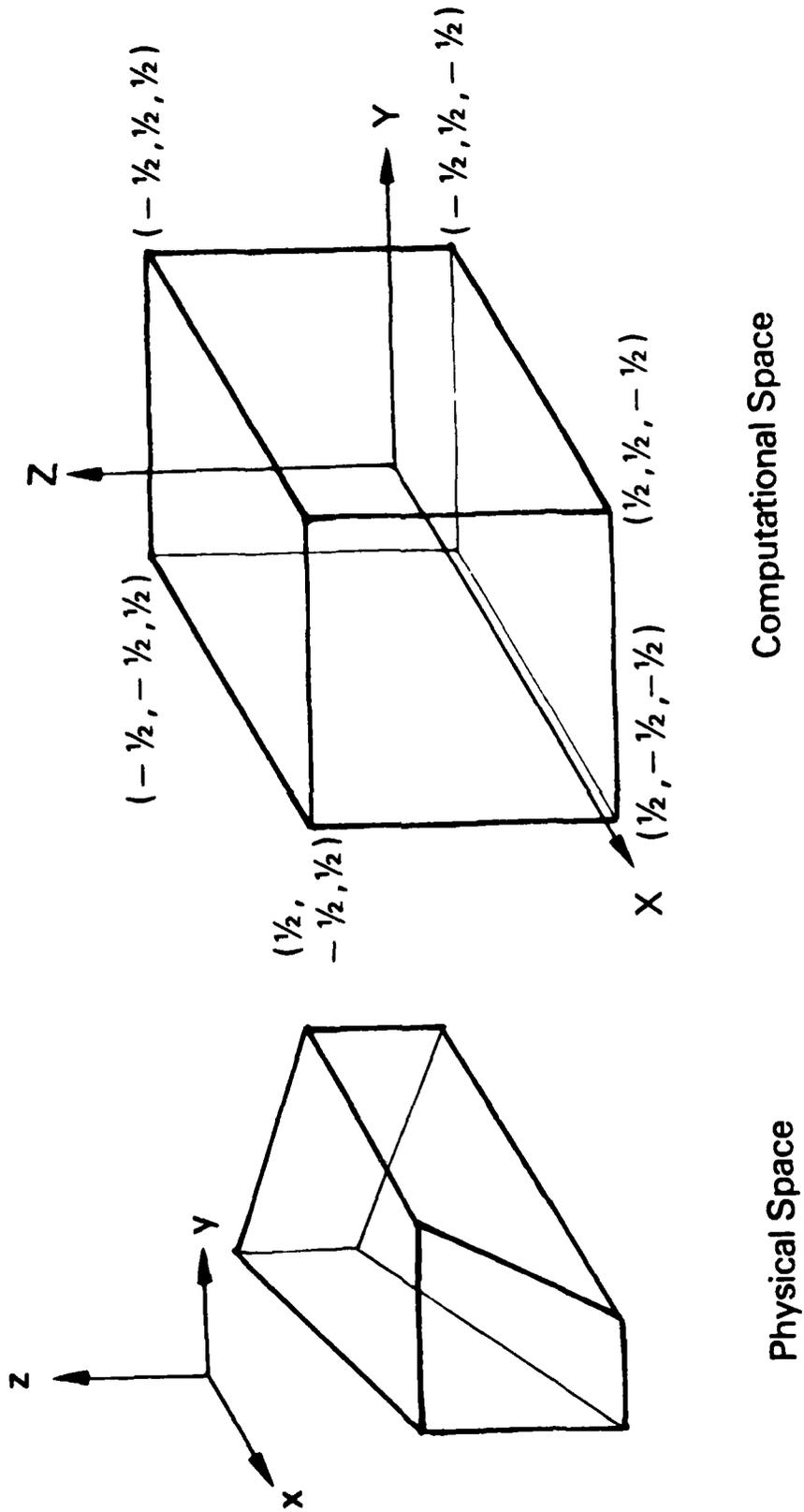


Figure 1. Mapping from Physical Space to Computational Space.

-6-

where $q^2 = u^2 + v^2 + w^2$. (4)

The first step in the procedure is to determine the governing equation [Equation (2)] in computational space. The result is

$$\frac{\partial}{\partial X^i} (\rho h U^i) = 0 \quad (5)$$

where X^i are the transformed coordinates [X, Y, and Z in Equation (1)] , U^i are the contravariant velocity components, and h is the determinant of the transformation matrix \tilde{H} with the elements $\partial x^i / \partial X^j$. The contravariant velocity is defined by

$$U^i = g^{ij} \frac{\partial \phi}{\partial X^j} \equiv (\tilde{H}^T \tilde{H})^{-1} \frac{\partial \phi}{\partial X^j} . \quad (6)$$

A differencing algorithm which conserves $\rho h U^i$ on the cubical cells is derived by creating a set of secondary cells whose vertices lie at the centers of the primary cubical cells. The flux quantity $\rho h U^i$ is evaluated at the center of each primary cell (the vertices of the secondary cell, Figure 2). The flux computed at the corner is assumed to be constant over that portion of the secondary cell face that lies within the primary cell. If the global mapping is sufficiently smooth to allow a Taylor series expansion of the physical coordinates in terms of the computational coordinates, then the local linear truncation error terms for the flux will cancel and the flux conservation formula will be accurate to the second order.

With this approach a problem arises in that the difference operator decouples odd and even points as shown in Figure 3. This results in a homogeneous solution where ϕ can be 1 at odd points and -1 at even points. This problem is overcome by displacing the flux evaluation point away from the vertices by adding a higher-order correction term. This displacement recouples the odd and even points and eliminates the

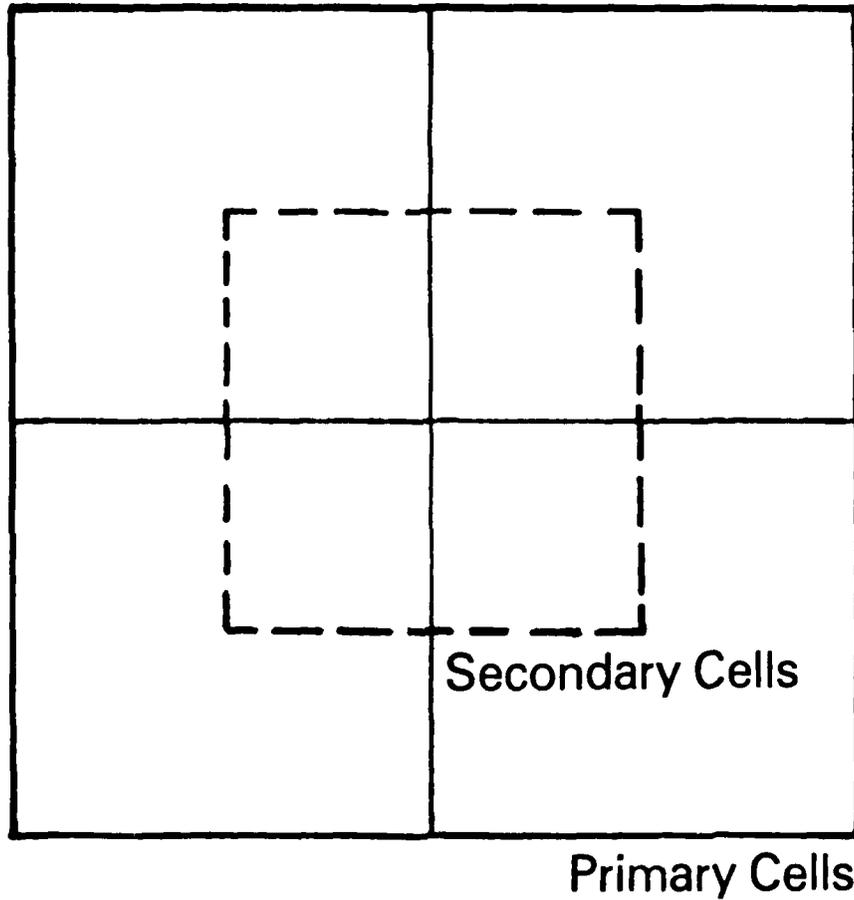
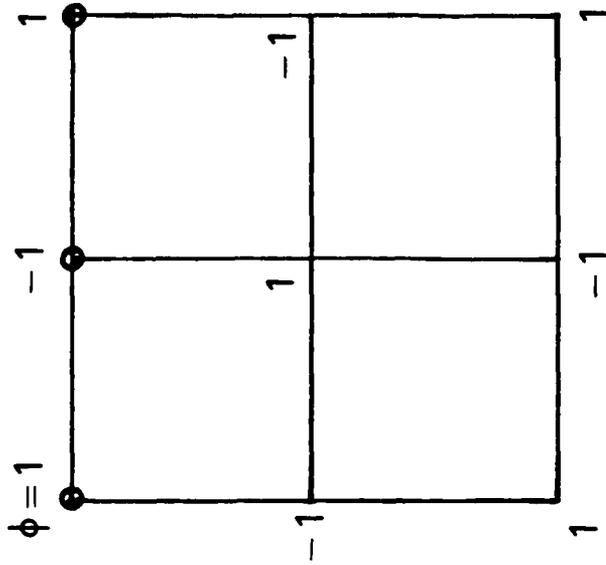
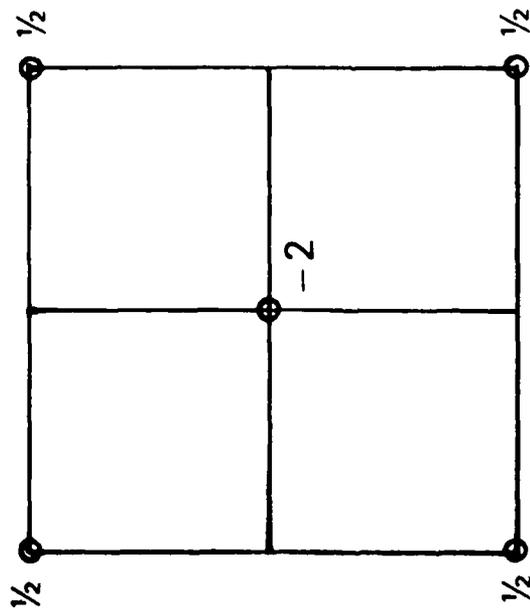


Figure 2. Primary and Secondary Cell in Computational Plane.



Distribution of ϕ Which Produces
a Homogeneous Result Using
the Difference Operator



Stencil of Difference
Operator

Figure 3. Decoupled Solution Arising from Difference Operator.

homogeneous solution. For the simple case of the flux being given by ϕ_x , the displacement relation used by Jameson and Caughey is

$$\phi_x = \phi_{x_0} + \epsilon \phi_{xy_0}, \quad (7)$$

where the subscript o represents the center of the primary cell and ϵ can vary from 0 to $\frac{1}{2}$ where the cell height is assumed to be 1 (Figure 4). Computation of these recoupling terms requires time, and other methods involving averaging which do not require adding terms are currently under investigation by other researchers.

In regions where the flow is supersonic, upwind differencing is employed. This is accomplished by adding terms to the conservation equation which produce an upwind bias. The terms are selected such that the proper domain of dependence is used in the differencing. The effect of this is to produce a rotated difference operator of the form

$$\frac{\partial \phi}{\partial s} = \frac{U^i}{q_c} \frac{\partial \phi}{\partial X^i}, \quad (8)$$

where s is the streamwise direction, q_c is the contravariant velocity, and the first-order difference operators $\partial/\partial X^i$ are chosen to be in the upwind direction. The terms added to the flux equation are

$$P^i = -\mu |U^i| \rho_X^i, \quad (9)$$

where μ is a switching function

$$\mu = \max \left[0, (1 - a^2/q^2) \right] \quad (10)$$

and q/a is the local Mach number. The presence of these terms has the effect of adding artificial viscosity to the solution. This does require, however, that the mesh be smooth in the supersonic zone or the effect of the higher-order derivatives associated with the artificial viscosity will cause the solution to give erroneous results.

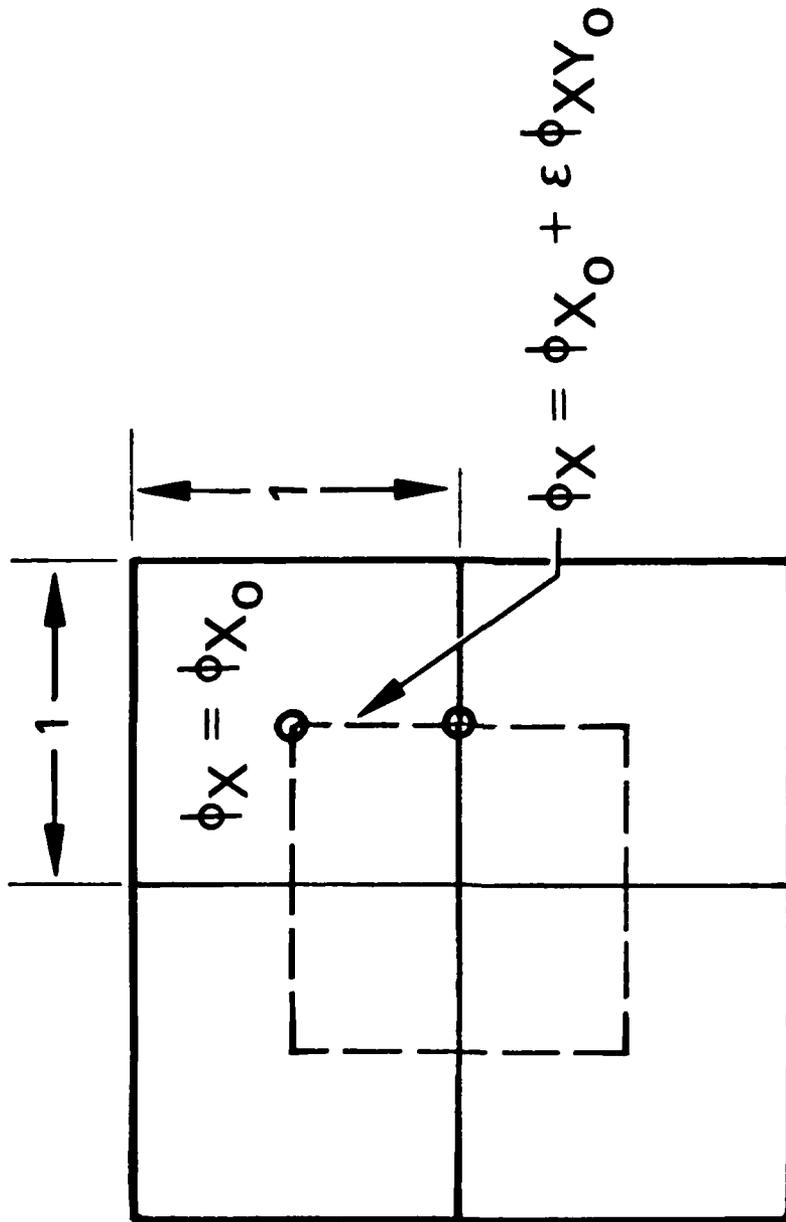


Figure 4. Shift in ϕ_X Used to Recouple Solutions.

The last terms which have to be added to the equation are timelike derivatives which have the effect of embedding the steady-state equation in an artificial, time-dependent equation. The final equation that is solved is a discrete approximation to

$$\frac{\partial}{\partial x^i} \rho h U^i + P^i = \alpha \phi_{XT} + \beta \phi_{YT} + \gamma \phi_{ZT} + \delta \phi_T, \quad (11)$$

where the P^i are the upwind biasing terms in the supersonic zones, α , β , and γ are chosen to make the flow direction timelike, as in the steady state, and $\delta \phi_T$ is a damping factor.

The complete numerical scheme is outlined below.

- (1) Evaluate the contravariant velocity components and densities at the centers of the primary cells.
- (2) Satisfy continuity on the secondary cells using the flux values calculated in step 1 plus the recoupling terms.
- (3) Add artificial viscosity in the supersonic zones to produce an upwind bias and enforce the entropy condition.
- (4) Add the time-dependent terms to embed the steady-state equation in a convergent, time-dependent process which evolves to the solution.

The main difficulty associated with developing a computer code based on the finite-volume algorithm is that of generating a grid system and incorporating boundary conditions. A desirable grid is one which conforms to all the solid boundaries. Boundary-conforming grids provide an accurate and convenient means of specifying boundary conditions. They also can be made very efficient in that the grid density can be readily controlled at the boundaries where the gradients of the flow parameters can vary most rapidly.

Since the finite-volume method only requires sets of coordinates corresponding to the corner points of the six-sided computational cells, there is no need to have a single mapping function to generate the grid. The procedure chosen is one that uses a sequence of rather simple transformations. The overall mapping is required to be smooth so that the higher-order effects of the transformations do not cause numerical instabilities, particularly in the vicinity of shocks.

3. Field-Point Calculations

In order to calculate velocities in the field, the location of the field point relative to the grid points must first be determined. This is done by transforming the field point to the computational grid. In this Cartesian coordinate system, the eight grid points surrounding the field point can be readily identified. Once this has been determined, the velocities, pressures, and local Mach numbers at the grid points can be calculated using the chain rule

$$U_i = \frac{\partial \phi}{\partial x^i} = \frac{\partial X^j}{\partial x^i} \frac{\partial \phi}{\partial X^j} \quad (12)$$

Central differencing is used to calculate the derivatives. Once the U_i components have been determined, the local Mach number and pressure can be calculated. The values at the field points are then found by using trilinear interpolation which is consistent with the order of variation assumed in the finite-volume algorithm.

Figure 5 shows a comparison of velocities computed using the field-point interpolation data with an analytical solution. The case shown is a Karman-Treftz airfoil for which an analytical solution exists. The line of computed points starts at the leading edge of the airfoil where velocities change rapidly. This provides a good test of the interpolation procedure. The agreement is excellent. The calculation was done using a code which models wings in a wind tunnel. This code uses the same grid-generation technique as the FLO 30 code. However, there is no body in this code, so that the spanwise grid sections are planes rather than cylinders. Unfortunately, time did not permit us to fully implement the field-point calculation into the FLO 30 code, so this should be done later.

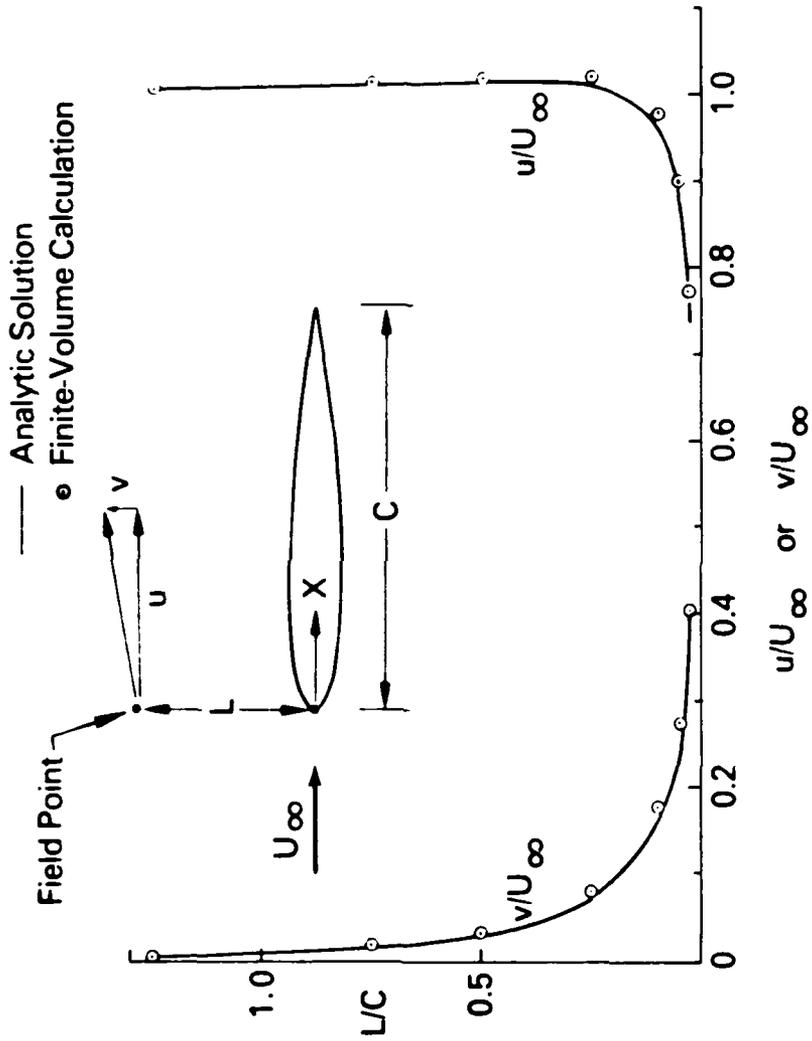


Figure 5. Field-Point Velocities Along $X = 0$.

4. Wing-Body Code Development

4.1 Introduction

An investigation was made of the two wing-body finite-volume codes developed in the Phase I work by Caughey, Jameson, and Nixon (1979). Descriptions of the grid systems for the two codes and the results of the study are presented below.

The first task which was accomplished was to complete the work started in Phase I for converting the quasi-conservative differencing scheme used in FLO 25 to one which is fully conservative. The resulting code has been named FLO 30. This new code differs from FLO 28 in the grid system used to define the computational space. The FLO 28 grid system uses a Joukowski transformation to map the noncircular fuselage to a slit with the wing extending outward. A grid system is then established with planes parallel to the freestream cutting the wing, and a parabolic C-type mesh is used within each plane. Results presented in Caughey and Jameson (1979) revealed oscillations in the pressure distribution in the wing root area. Further analysis revealed that these were boundaries which resulted in an irregular fuselage geometry in the computation. Although ways could be developed to modify this grid and overcome this problem, this approach was not undertaken as the other grid system appeared to provide better wing root-fuselage geometry modeling and did not suffer from the problem mentioned above. The problem noted became more severe for high- and low-mounted wings which are the typical configurations.

The FLO 30 grid system described by Caughey and Jameson (1979) uses a cylindrical-type system. Quasi-cylindrical shells surround the fuselage. The inner shell corresponds to the actual fuselage geometry and the outer shell to a cylinder on which the far-field boundary condition is applied. On each shell a parabolic C-type mesh is used. This system provides excellent modeling in the wing root area and also provides more mesh points on the fuselage than the slit-type transformation. Two expected drawbacks from this system did not present any significant difficulties. For a closed body, the cylindrical system collapses to a

line. In practice a very small cylindrical extension to the body is used and the results appear satisfactory. Also, since the system is cylindrical, the vertical mesh spacing above and below the wing increases with distance outboard from the body. However, in practice, vertical mesh spacing near the wing tip is comparable for the cylindrical- and slit-type systems, with the result that the wing root mesh spacing is better.

The remainder of this section will describe the mesh system in more detail and explain some improvements which were necessary to make the earlier version reported in Caughey and Jameson (1979) more robust. Example calculations for a Learjet and an A-7 will be presented.

4.2 Grid Generation

The cylindrical computational surfaces are formed by first defining the fuselage surface as

$$r = R_f(x, \theta) \quad (13)$$

where

$$\theta = \tan^{-1} (y/z) . \quad (14)$$

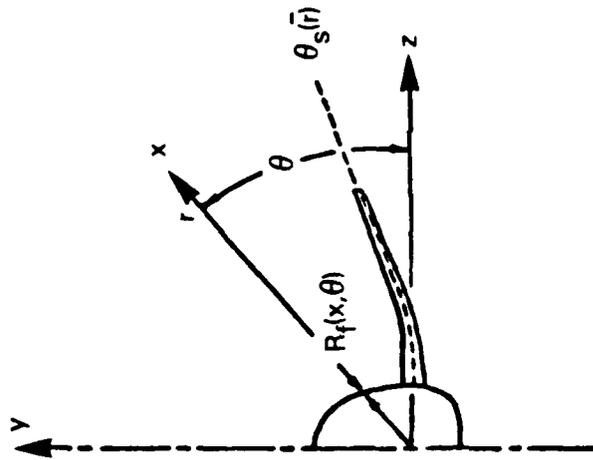
The coordinate system is shown in Figure 6. A nondimensional radius is formed by

$$\bar{r} = \frac{r - R_f(x, \theta)}{R_t - R_f(x, \theta)} . \quad (15)$$

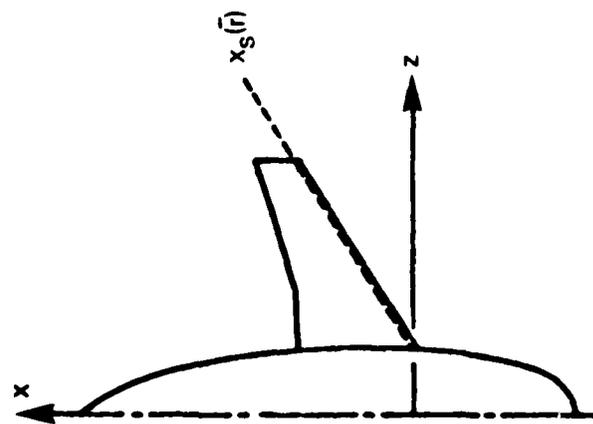
Here R_t is the radius of the cylinder passing through the wing tip.

Within each surface of constant \bar{r} , the configuration appears as a wing in a wind tunnel with θ being the ordinate and the "wind tunnel walls" as the symmetry planes. A C-type mesh is used in this plane. The mesh conforms to the wing's surface and the "wind tunnel walls."

The mesh is generated by "unwrapping" the airfoil about a line which starts at the center of curvature of the leading edge and proceeds downstream bisecting the trailing-edge angle and coincident with the wake. The wake position is assumed such that it bisects the trailing-edge angle and follows a body contour line downstream. The purpose of



(a) Plan View



(b) Front View

Figure 6. Coordinate System for Wing-Body Code.

the unwrapping is to create a Cartesian grid system in a transformed plane which will allow a convenient distribution of mesh lines to be specified. The grid points in the transformed plane can then be transformed back to the physical plane.

The transformation procedure consists of several transformations and is outlined in Figure 7. First a coordinate shift is made to remove the sweep and place one coordinate axis at the center of curvature of the wing's leading edge. Next an elliptic transformation is made to shift the wing to the center of a tunnel whose walls are at $\pm\pi$

$$(\bar{y} - a)^2 + [2(\theta - \theta_s) - b]^2 = R^2, \quad (16)$$

where a , b , and R are selected to meet the constraints

$$\begin{aligned} \bar{y} &= +\pi \text{ at } \theta = +\pi/2, \\ \bar{y} &= -\pi \text{ at } \theta = -\pi/2, \text{ and} \\ \bar{y} &= 0 \text{ at } \theta = \theta_s. \end{aligned} \quad (17)$$

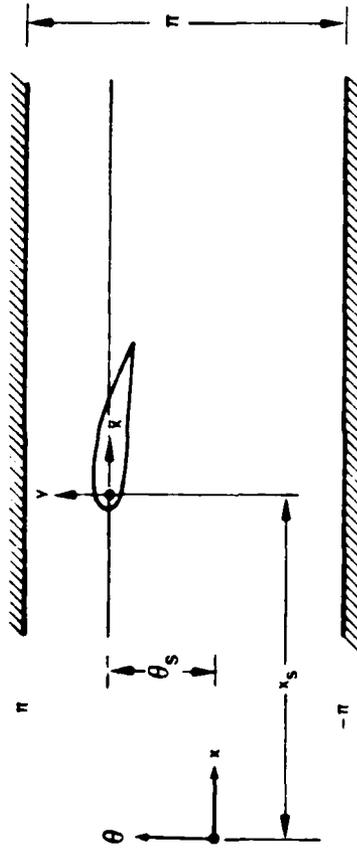
Here θ_s is the angular location of the center of curvature of the wing section formed by the intersection of the cylindrical grid surface and the wing. The factor 2 appears on the $\theta - \theta_s$ term so that the global scaling of the θ to y transformation can be accounted for. This latter transformation allows the wing to be displaced from the centerline ($\theta = 0^\circ$) by as much as $\pm 65.88^\circ$. This has been found to be adequate for all the test cases run so far. If still further displacements are required, an exponential transformation could be used. This function would always provide a unique mapping regardless of the amount of displacement from the centerline.

Next an unwrapping function is used:

$$\xi + i\eta = \cosh^{-1} (1 - 2e^{\bar{x} + i\bar{y}}). \quad (18)$$

This transformation makes the wing and wake appear as a slowly varying curve about $\eta = \pi$ in the ξ, η plane. Finally the ξ, η plane is sheared using

$$Y = \eta/\eta_{\text{wing-wake}}. \quad (19)$$

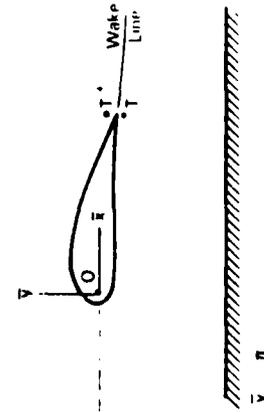


- $\bar{x} = (x - x_s) \times 2$
Removes Wing Sweep and Scales x According to the Mean θ Dimensional Scale
- $(\bar{y}-a)^2 + |2(\theta-\theta_s) - b|^2 = R^2$
Moves Model to Center of Tunnel and Scales Dimension so that the Tunnel Walls are at $\pm \pi$

(a) Origin Shift, X Scaling, and θ Distortion.

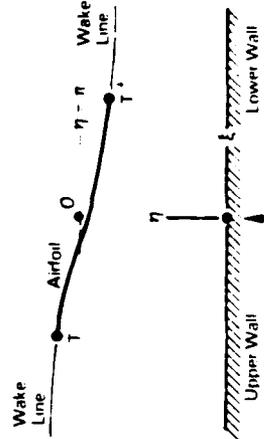
$$\xi + i\eta = \cosh^{-1}(1 - 2e^{\bar{x} + i\bar{y}})$$

$$Y = \eta/\eta_{\text{upper}}$$



(b) Distorted Plane.

(c) Unwrapped Space.



(d) Computational Space.

Figure 7. Sequential Mappings from Physical to Computational Space.

This produces the desired parallel line representation of the wing-wake and tunnel wall shown in Figure 7d. The remaining procedure is to distribute Cartesian grid lines in this space and transform the intersection points back to the physical plane by reversing the transformation procedure just described. Figure 8 shows a coarse grid generated by the procedure. The fine grid used for the final computation has four times as many divisions in each direction.

One additional transformation was found to be necessary to handle highly swept wing configurations. For swept wings that are highly tapered, the mesh system described above becomes very highly swept far upstream or downstream. This causes numerical instability problems. The reason that the mesh sweep increases upstream or downstream is that for each cylindrical surface the nondimensionalization used is based on the local wing chord. With a highly tapered wing, the mesh lines advance upstream more rapidly at the root than at the tip. This adds to the basic sweep of the mesh system due to wing sweep. To overcome this problem, the grid points obtained by the transformations described so far were shifted according to:

$$x = x_{LE} + (x - x_{LE}) \left[1 + \left(\frac{C_R}{C} - 1 \right) \tanh^2 (x - x_{LE}) \right] \text{ for } x < x_{LE} \quad (20)$$

$$X = x_{TE} + (x - x_{TE}) \left[1 + \left(\frac{C_R}{C} - 1 \right) \tanh^2 (x - x_{TE}) \right] \text{ for } x > x_{TE} \quad (21)$$

where x_{LE} is the local wing leading edge, x_{TE} is the local wing trailing edge, C is the local wing section chord, and C_R is the root section chord. These stretching functions have the effect of changing the local scaling from the local chord to the root chord far upstream and downstream of the wing. This removes much of the added sweep due to taper and provides the more stable computational grid.

4.3 Boundary Conditions

Boundary conditions for the wing-body code can be divided into specifications on the configuration, specifications on the far-field boundaries, and specifications on the lifting-surface wake. The actual

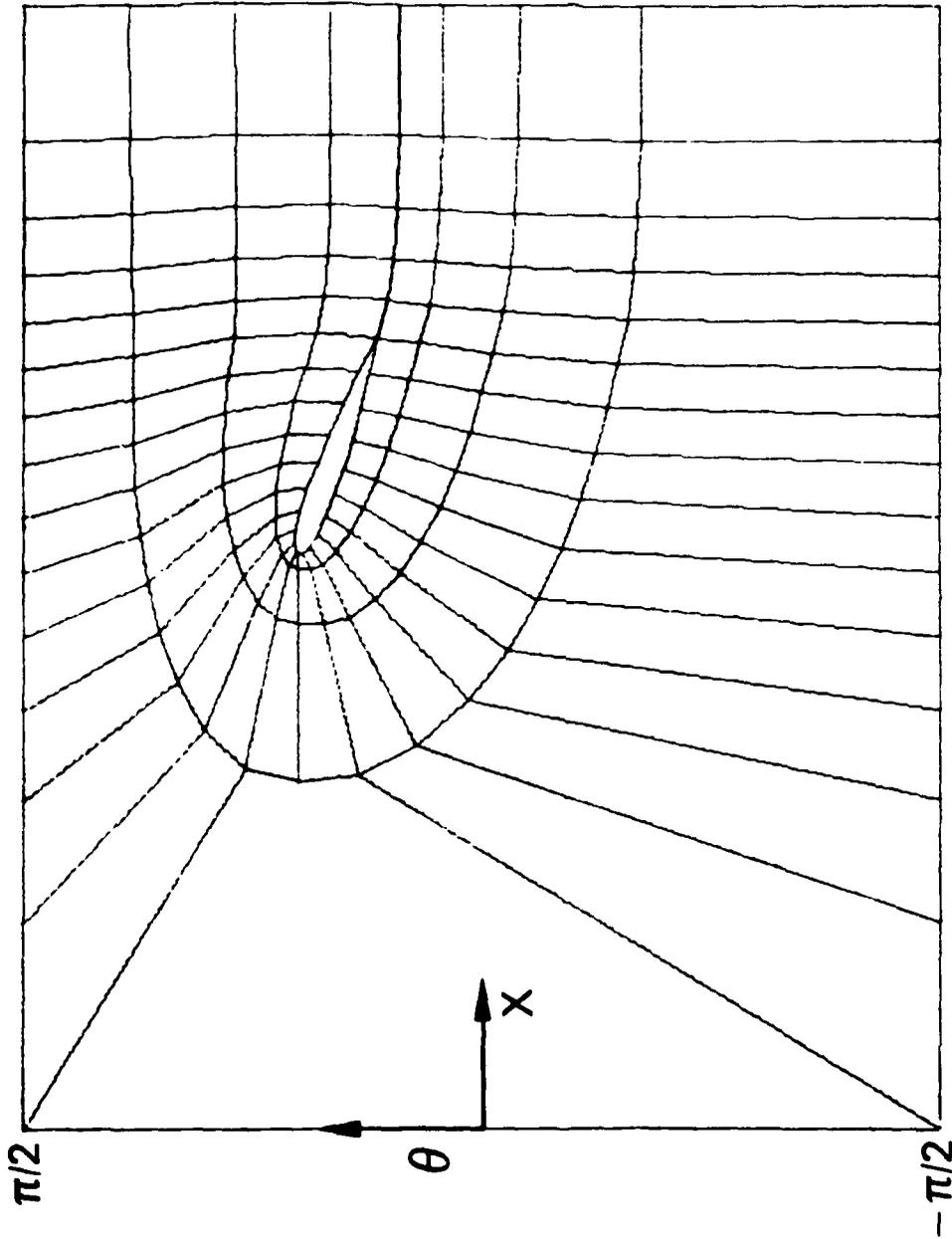


Figure 8. Sample Coarse Mesh.

wake position is part of the solution since it corresponds to a streamline surface. However, it has been demonstrated in the past that an approximation to this surface is actually good enough (see Section 6). An approximation which works fairly well is to assume that the wake leaves the wing's trailing edge at the bisecting angle, and its position varies smoothly downstream. The wake is not assumed to be a streamline surface (i.e., normal flow is enforced). The wake does have a jump in velocity potential across it but the downstream and normal components of velocity are forced to be continuous. Therefore, there is only a discontinuity in the spanwise component of velocity, and the pressure is continuous to the first order.

On the body and wing, no normal flow is enforced. Also, no normal flow is enforced at the upper and lower boundaries of the two-dimensional grids on the cylindrical ($\bar{r} = \text{constant}$) surfaces since these lines correspond to the planes of symmetry. Flow normal to these lines corresponds to cross flow which must be zero for symmetry reasons.

Upstream, the Mach number and angle of attack are specified, and, the perturbation velocity potential vanishes. Downstream, the perturbation velocity in the x-direction is assumed to vanish. This produces a first-order approximation to a return to freestream pressure. On the outer shell, all the perturbation velocity components are assumed to vanish. The far-field boundaries in the finite-volume algorithm are actually at a finite distance from the configuration. This in itself introduces some error; however, comparisons with other analyses and wind tunnel data would indicate the effect to be small.

4.4 Check Cases

The wing-body code has been exercised for several representative configurations. Results have shown good agreement with other numerical techniques in their common range of validity. Two sample results are presented. The first example is a Learjet for which no wind tunnel data are available for comparison. The second configuration is a Navy attack aircraft (A-7) with a nonstandard supercritical wing which was designed for the configuration using numerical optimization techniques (see Haney, Johnson, and Hicks, 1979, and Haney and Johnson, 1980). The wing-body configuration was tested at NASA Ames Research Center to

verify the new design goals and, hence, wind tunnel data is available for comparison. The redesigned wing configuration resulted from a design exercise to test transonic numerical design techniques.

Figure 9 shows a coarse computational grid on the Learjet. The final computational mesh has four times as many grid lines in each direction and is formed dividing the mesh spacing shown in half and then in half again. Figure 10 shows the pressure distribution on the wing at a span station near the root and one near the tip. Unfortunately, no wind tunnel data were available for comparison.

Comparisons of wind tunnel results with FLO 30 calculations are shown in Figure 11. Except for very close to the root, the computed lower surface pressure agrees almost exactly with experimental results. Upper surface pressures, in general, are lower than computed. This could be caused by either wind tunnel interference or code modeling accuracy. The angle of attack used for the computer analysis was identical to the wind tunnel angle of attack of 4.68° . Closer upper surface pressure agreement might be obtained if lift coefficients were matched. Also, the effect of viscosity has an influence on the pressures. Near the tip ($\eta = 0.878$), the boundary layer reduces the amount of recompression behind the shock and weakens the shock strength. Overall the comparison shows excellent agreement.

Haney and Johnson (1980) describe the models and test procedure used to obtain the wind tunnel data. Comparisons in that report with a computer code that does not include the fuselage are not as good as those shown in Figure 12. In order to demonstrate the effect of the fuselage on the pressure data, the FLO 28 code was used to model the wing alone. For the wing-alone calculation, the wing planform was extended to the plane of symmetry. The first station shown on Figure 12 ($\eta = 0.146$) is close to the wing-body juncture ($\eta = 0.12$). These results show the strong influence of the fuselage and the good agreement of the wing-body code with the wind tunnel data. Both the wing-body and wing-alone computer codes were run at the same angle of attack as the wind tunnel model; as for the previous comparisons, there was no attempt to match overall lift.

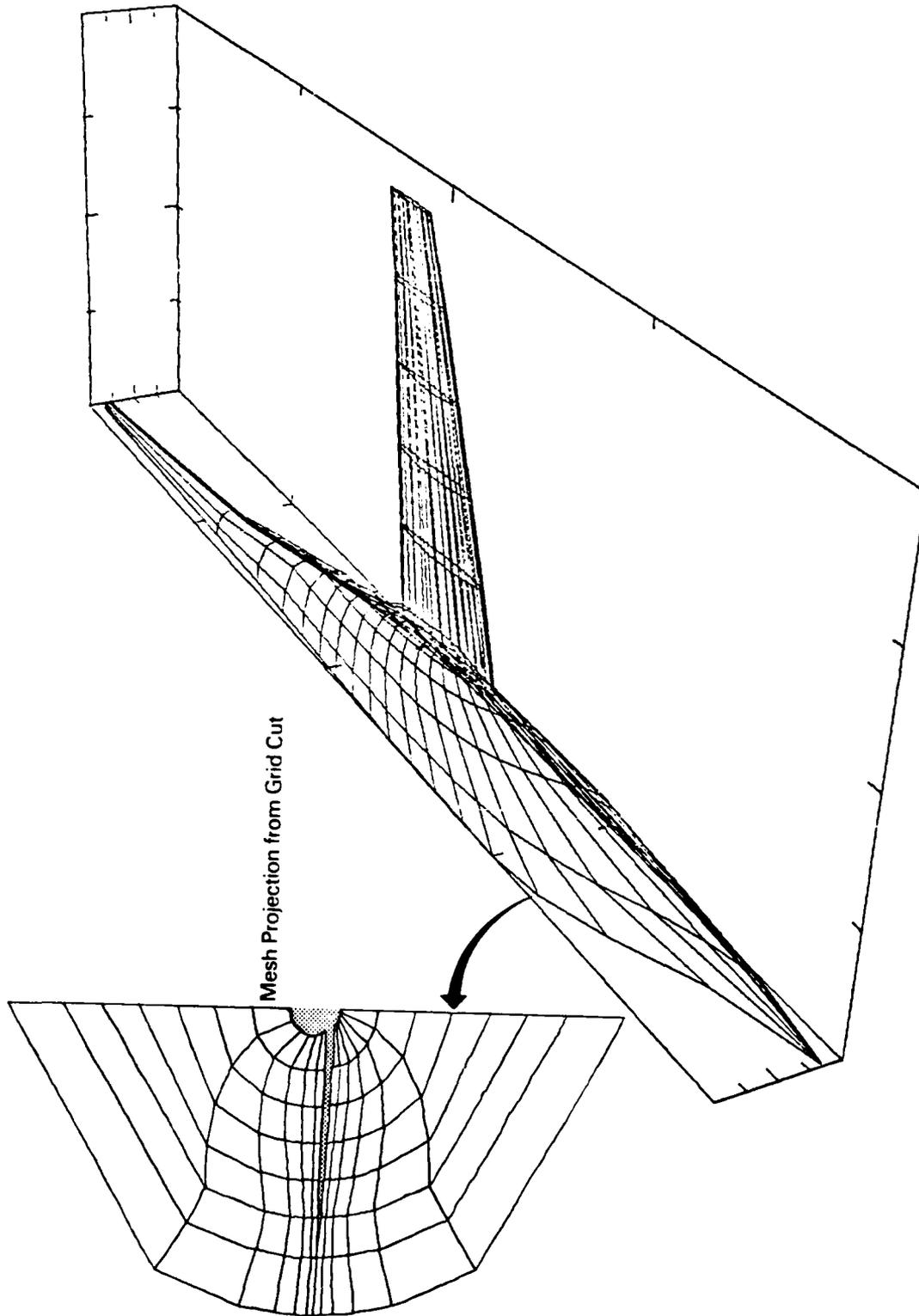


Figure 9. Coarse Grid on Learjet.

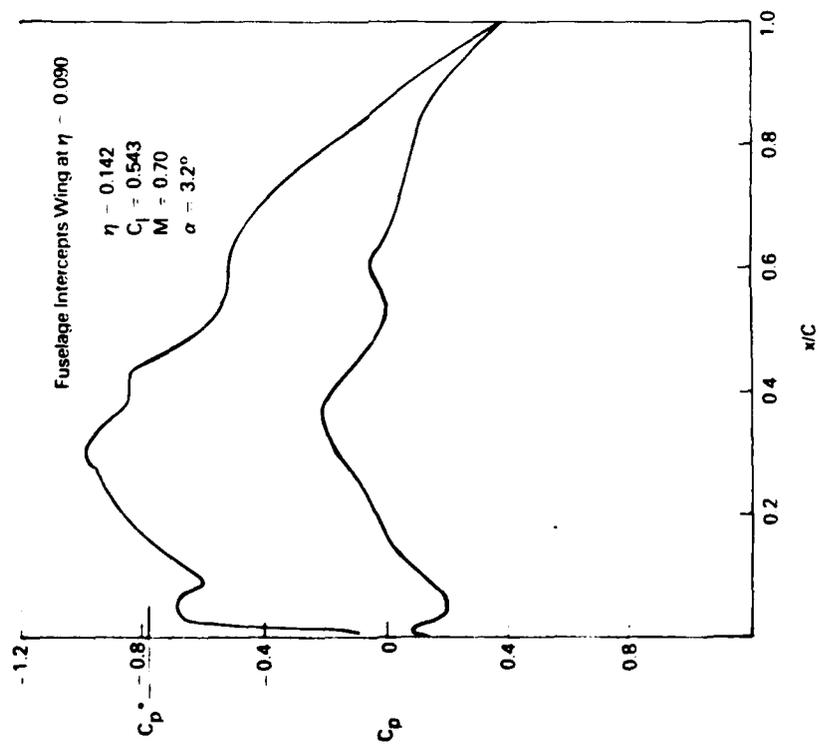
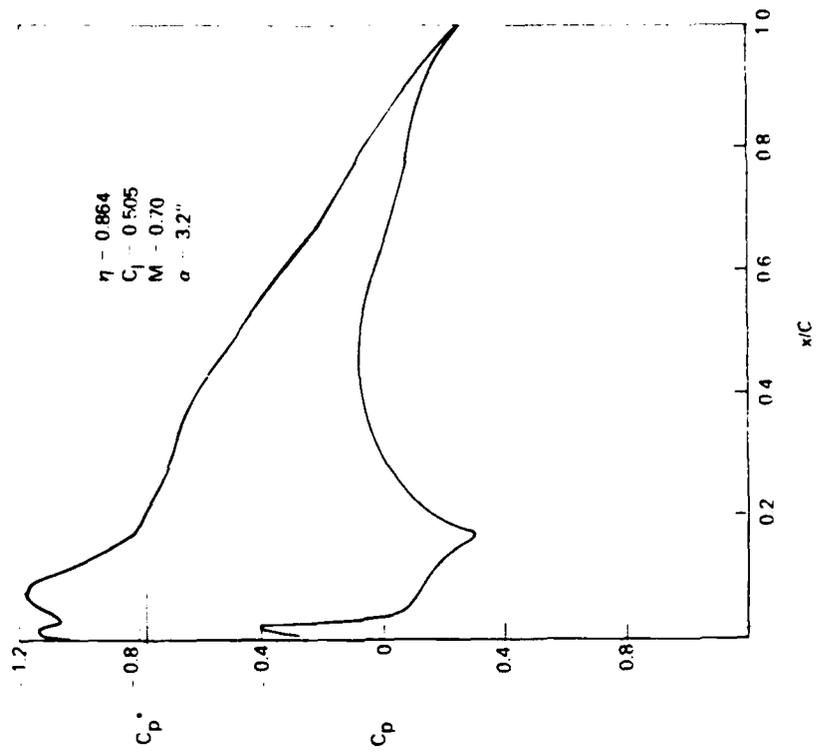


Figure 10. Cordwise Pressure Distribution on Learjet.

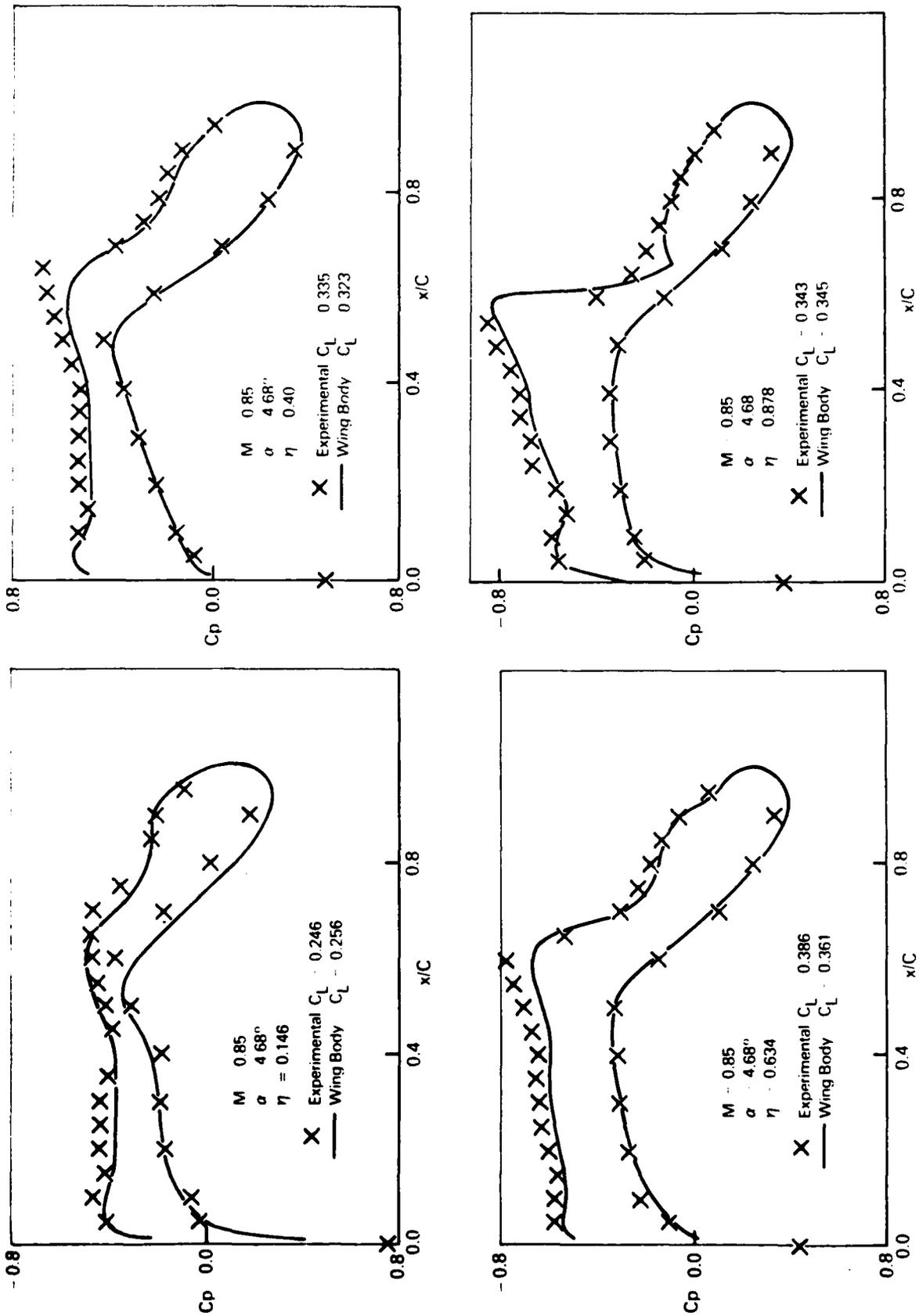


Figure 11. Comparison of FLO 30 with Experiment for A-7.

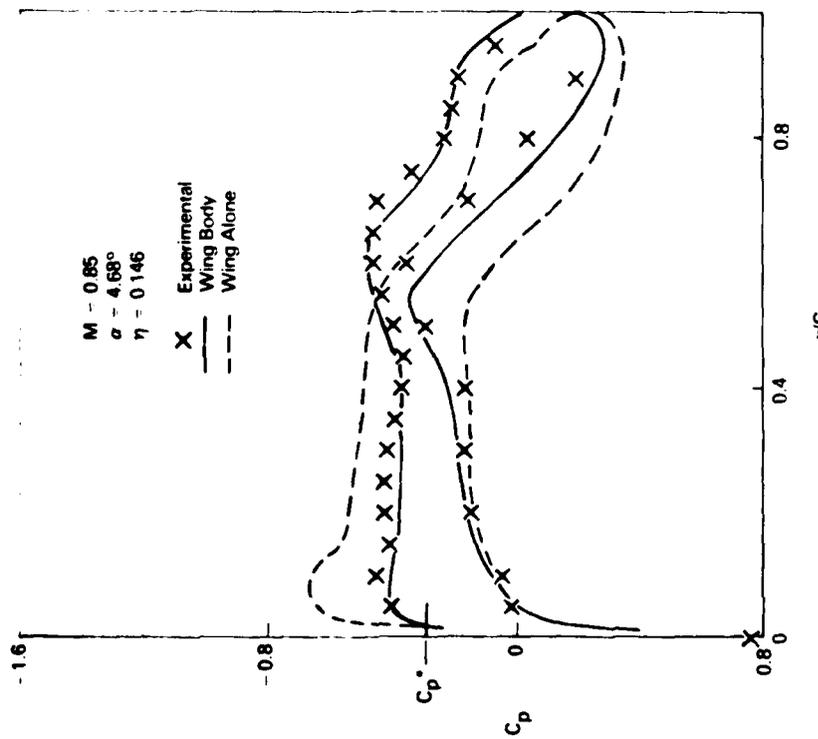
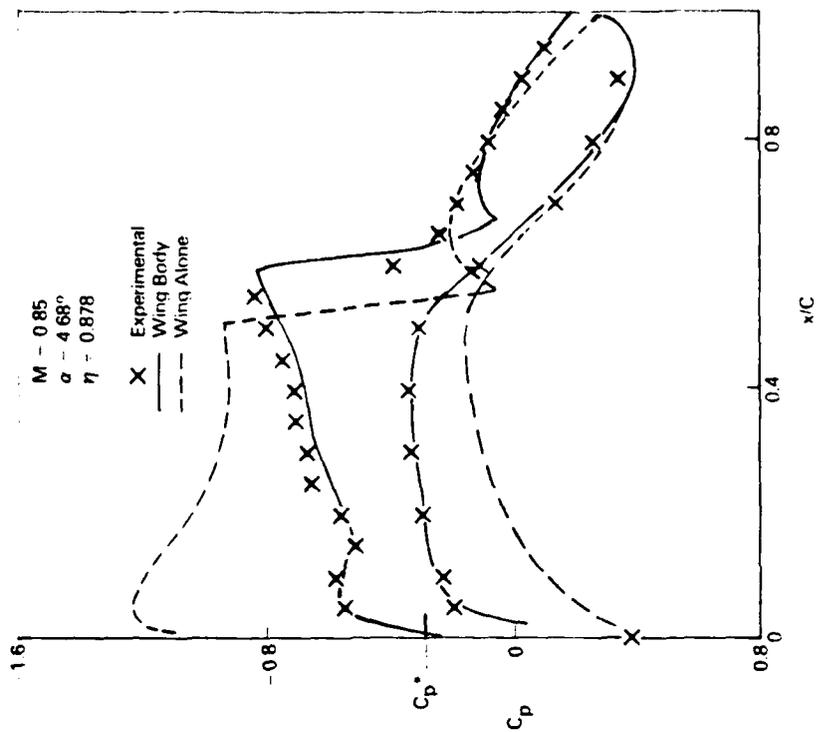


Figure 12. Comparison of Computed and Experimental Cordwise Pressure Distribution on Modified A-7 Model.

Figure 13 shows the effect of the fuselage on the spanwise loading. The results are nondimensionalized by the total lift coefficient so that the comparison shows the distribution effect. The presence of the fuselage tends to increase the nondimensionalized loading inboard on the wing. The effect of the fuselage on the total lift is indicated on the figure. For an angle of attack of 4.68° , the fuselage reduced the total lift by 38 percent from 0.485 for the wing-alone case to 0.300 for the wing-fuselage case.

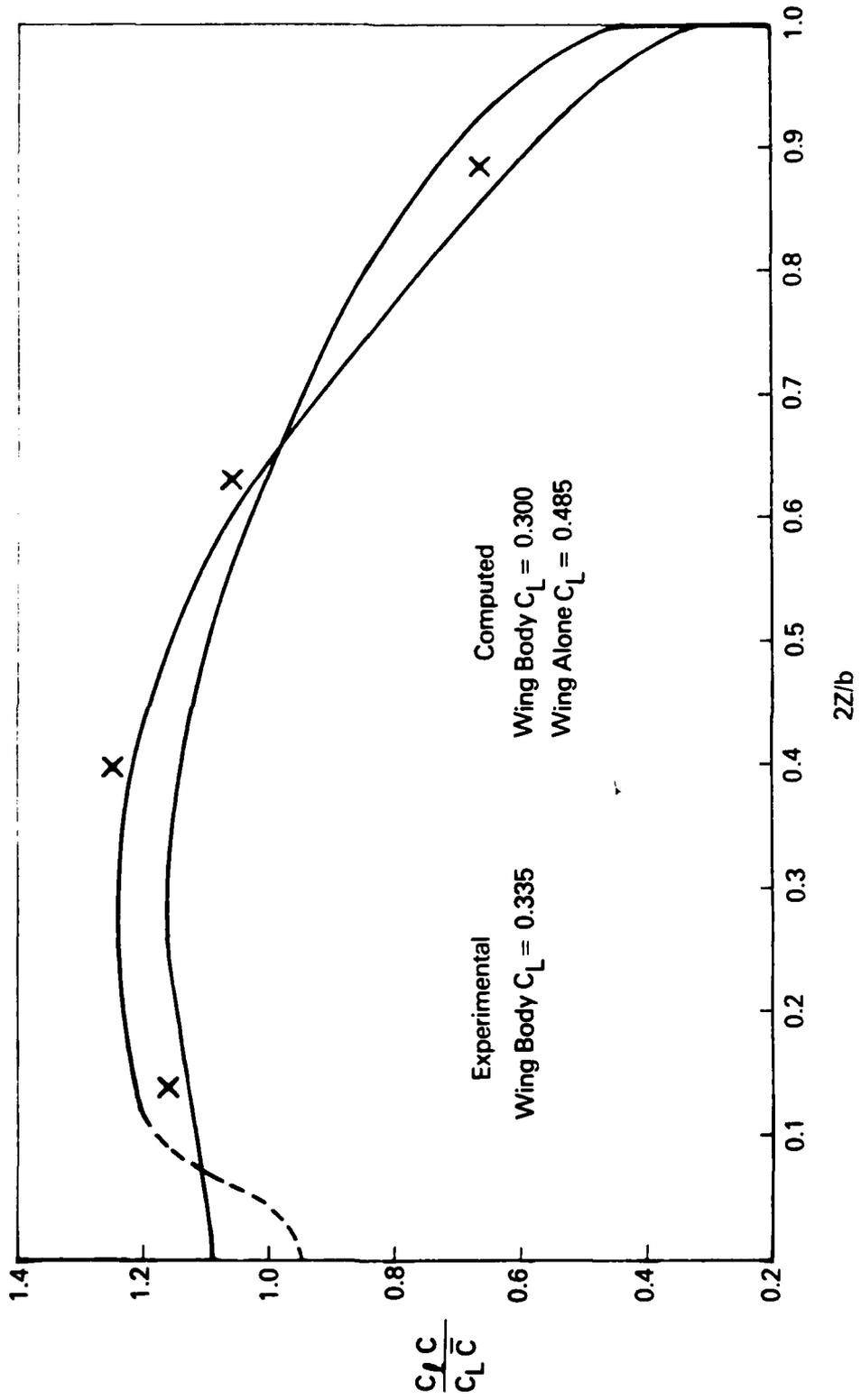


Figure 13. Effect of Fuselage on Computed Spanwise Loading of Modified A-7 Model.

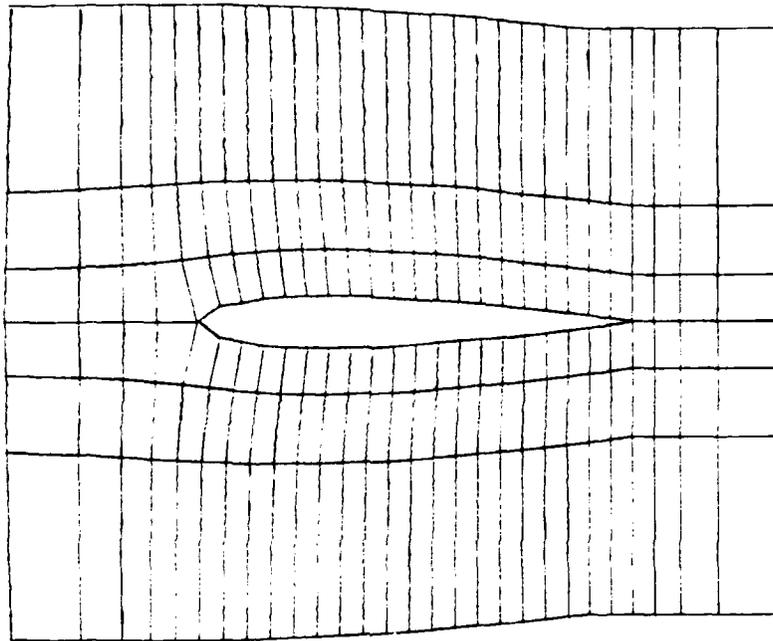
5. Slit Transformations

A slit transformation is a mapping which compresses a finite-thickness body into a zero-thickness sheet. One example of such a transformation is the Joukowski and shearing transformations that FLO 28 uses for the fuselage representation. By employing a sequence of such transformations, a complex wing-body-tail-nacelle configuration can be reduced to a configuration whose fuselage, wing, tail, and nacelles are all slits being on Cartesian planes. Generation of the computational grid for such a configuration would be quite easy. Therefore, a preliminary study using a two-dimensional airfoil code was initiated to test the feasibility of such a transformation. Results for a NACA 0012 airfoil are presented.

The grids obtained by a slit transformation are shown in Figure 14. The grid lines are similar to streamlines and potential lines. The distribution of lines is selected to give good resolution in the areas of interest and where the flow variables are changing rapidly. Away from the airfoil, the streamlines spread.

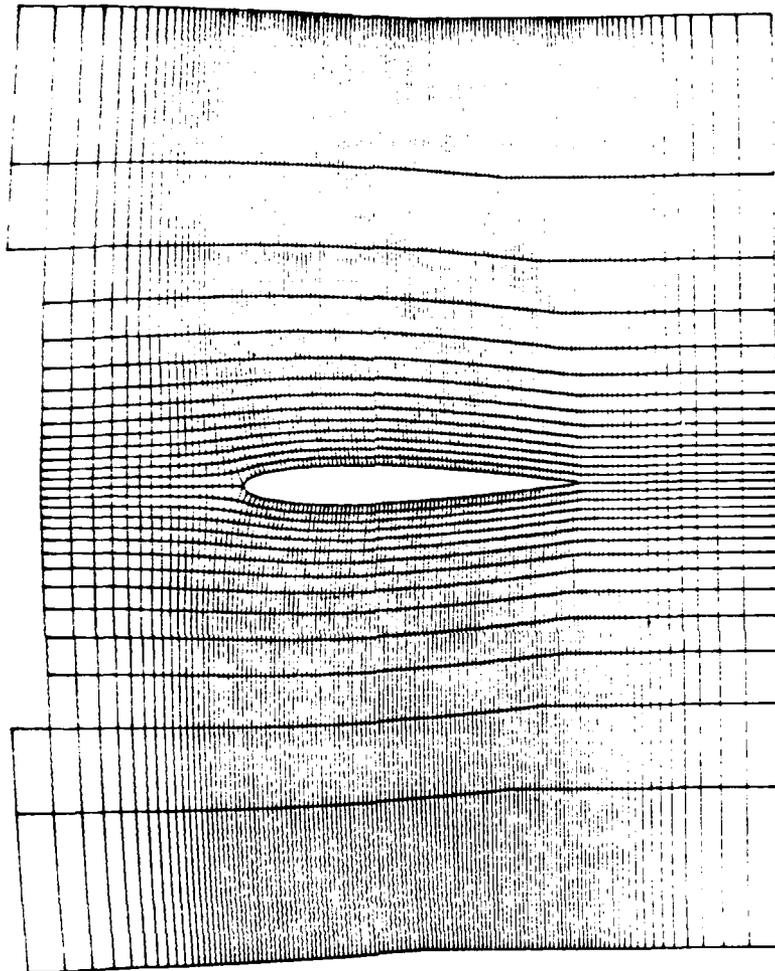
This type of mesh can easily be blended into a Cartesian mesh for the far field. A disadvantage of this mesh system is that the streamline and potential-like lines maintain their spacing throughout the flow field so that the dense spacing of lines emanating from the airfoil leading edge remain densely spaced in the far field where this density is not required. Another disadvantage is that the transformation is singular at the leading edge of the airfoil. This transformation singularity causes the velocity potential variation to also become singular, even though the velocity potential variation is regular in the physical domain.

A computer code was assembled which implemented the slit transformation into a two-dimensional finite-volume code. Figures 15 and 16 show results for a sample calculation. For this test case, no special treatment was made for the leading edge region. Rather the fact that the discretization does not produce a singularity was used as a means to eliminate any special treatment of the leading edge. The results show substantial differences between the coarse and fine meshes in both minimum pressure level and shock position. The convergence rate was



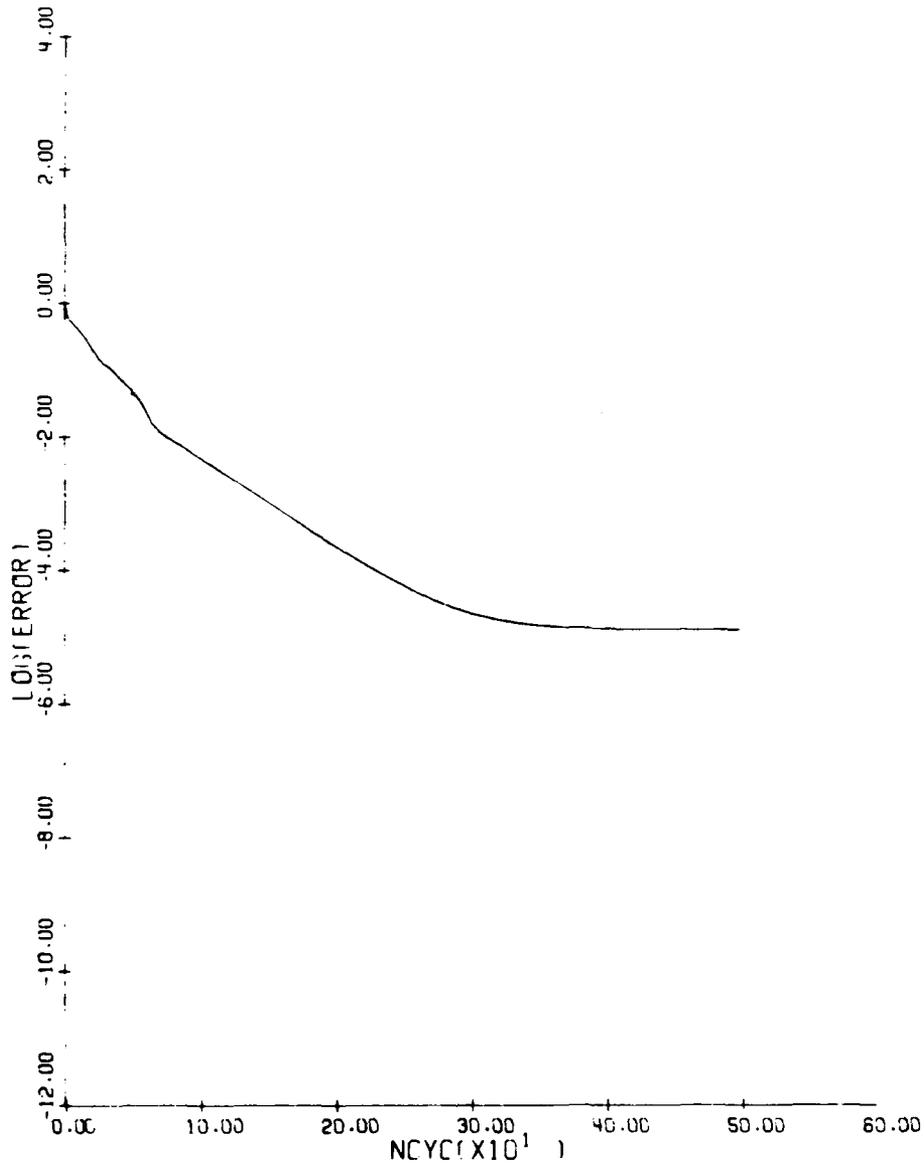
NACA 0012
GRID 32 X 8

Figure 14. Grid Plots.



NACA 0012
GRID 128 X 32

Figure 14. (Cont.)



NACA 0012
MACH .800 ALPHA 0.000
RESID1 .953E-02 RESID2 .118E-06
WORK 433.00 RATE .9776
GRID 32 X 8

Figure 15. Zero Angle of Attack.

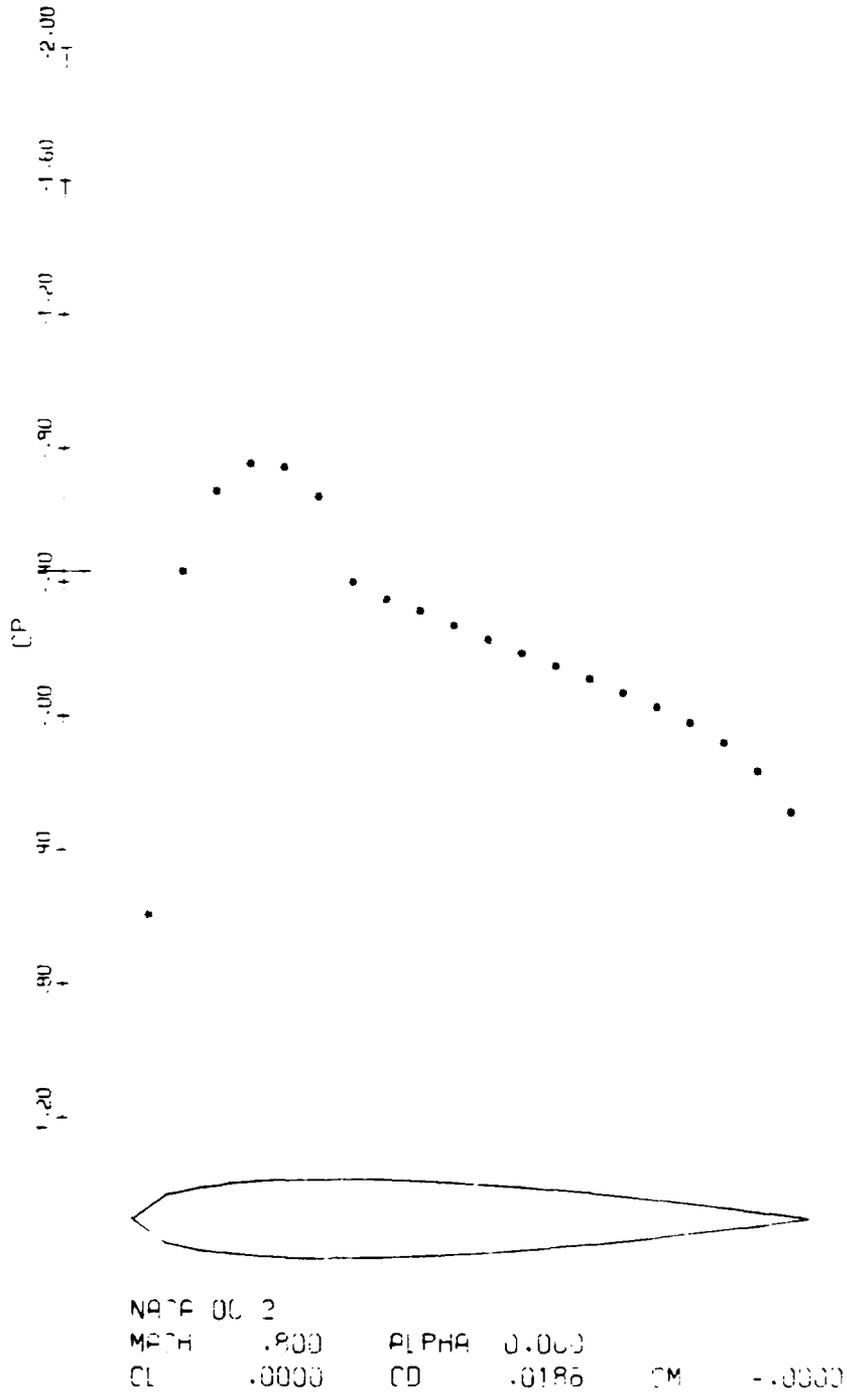
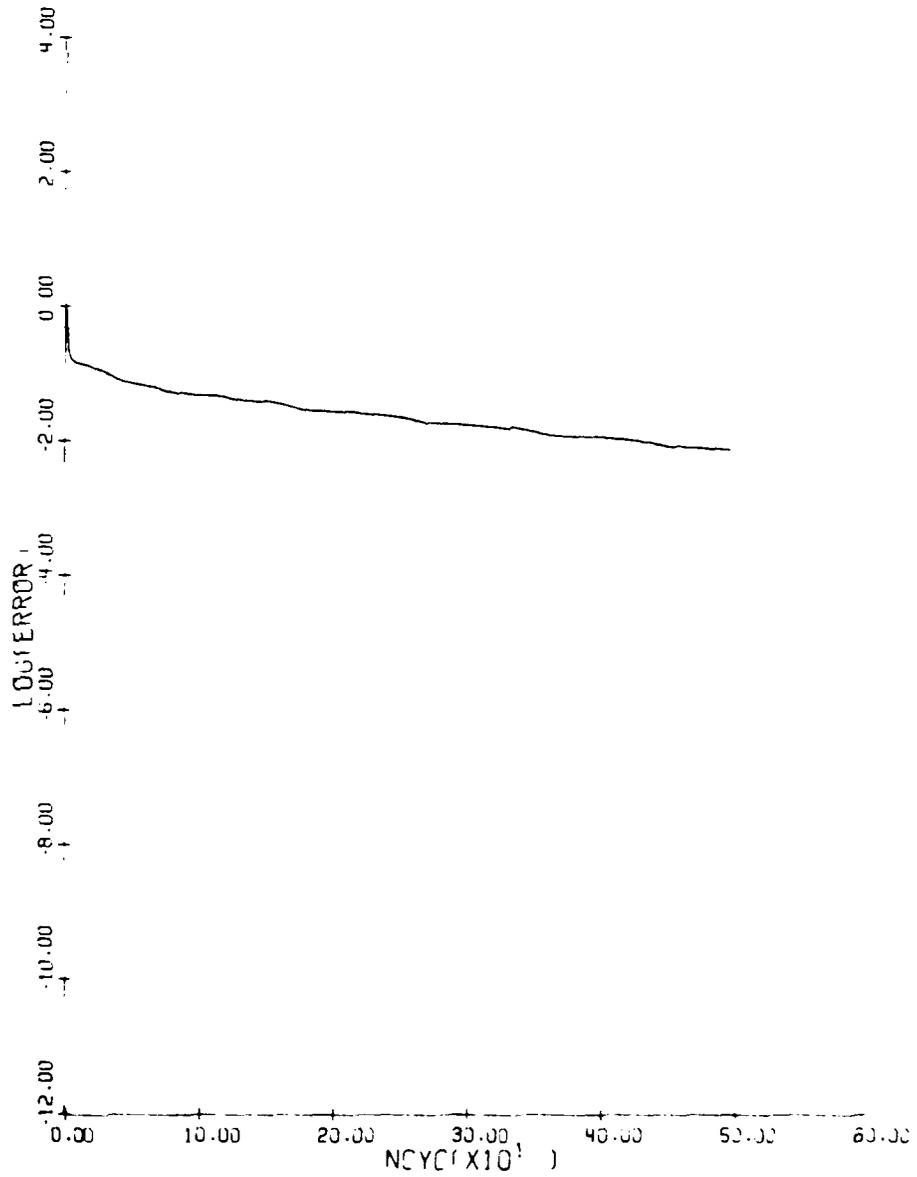
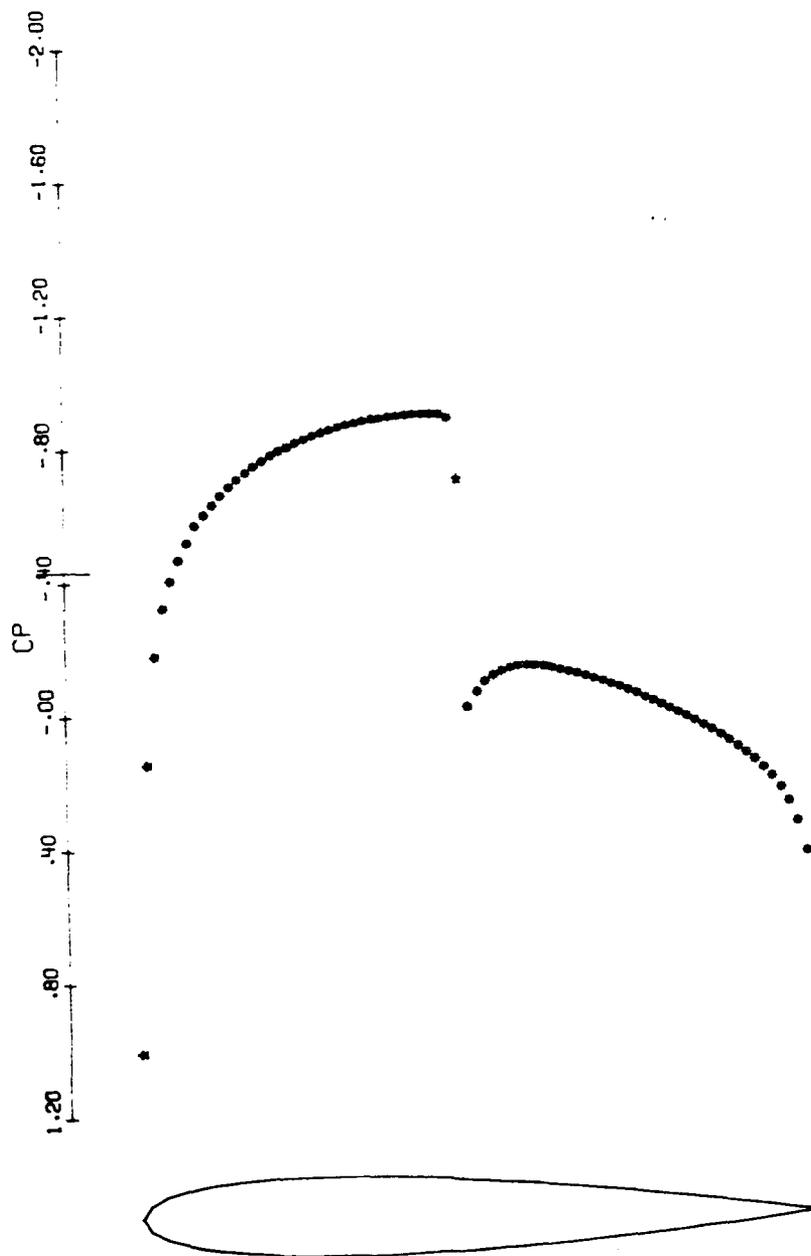


Figure 15. (Cont.)



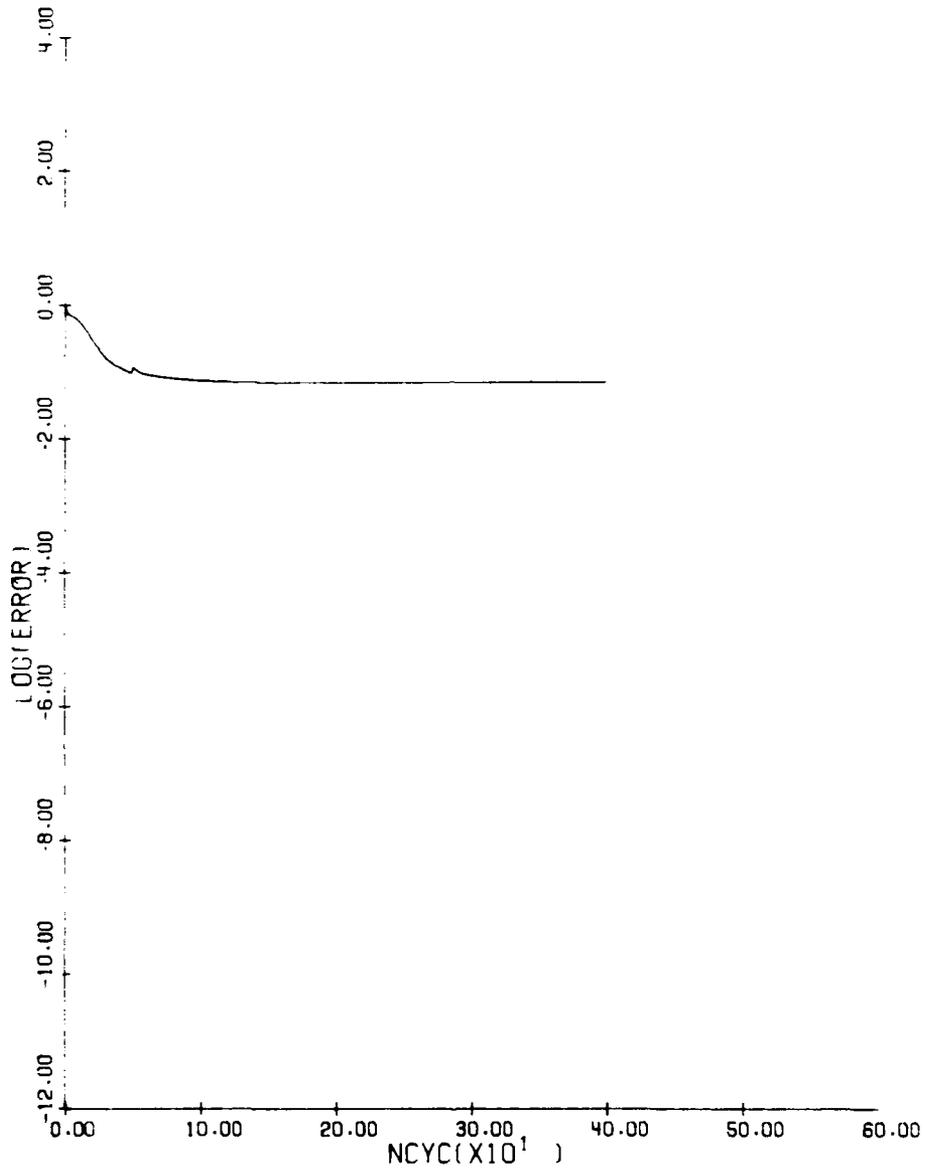
NACA 0012
MACH .800 ALPHA 0.000
RESID1 .104E-02 RESID2 .745E-05
WORK 439.00 RATE .9301
GRID 128 X 32

Figure 15. (Cont.)



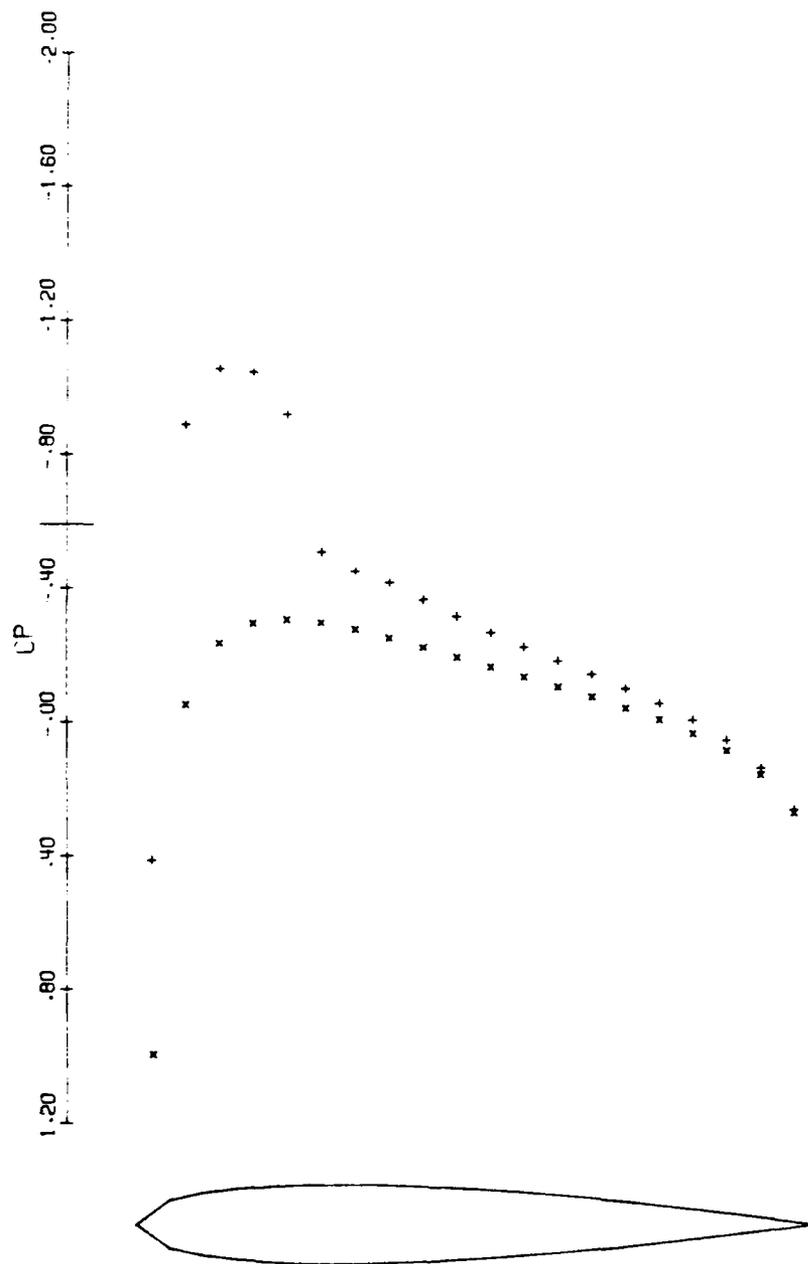
NACA 0012
MACH .800 ALPHA 0.000
CL .0000 CD .0165 CM .0000

Figure 15. (Cont.)



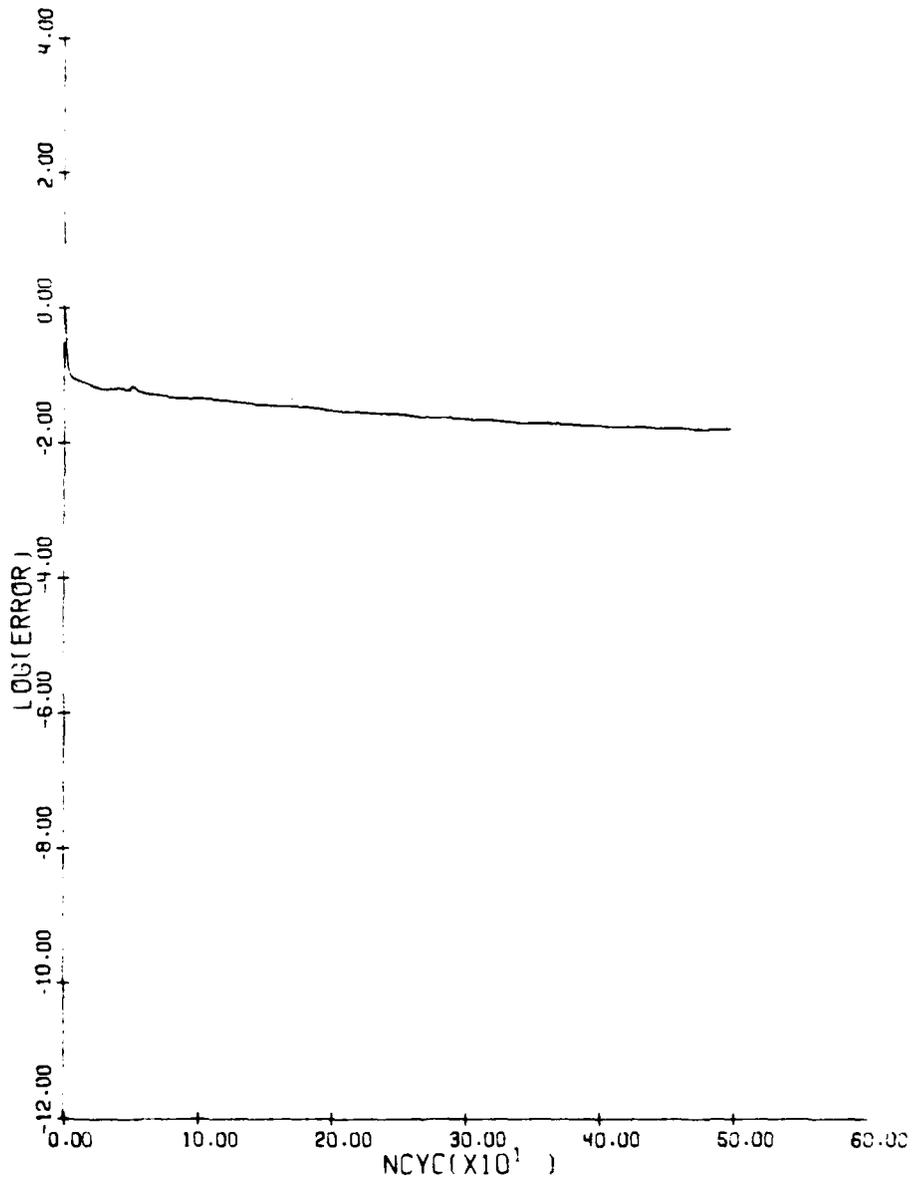
NACA 0012			
MACH	.750	ALPHA	2.000
RESID1	.944E-02	RESID2	.647E-03
WORK	401.00	RATE	.9933
GRID	32 X 8		

Figure 16. Two-Degree Angle of Attack.



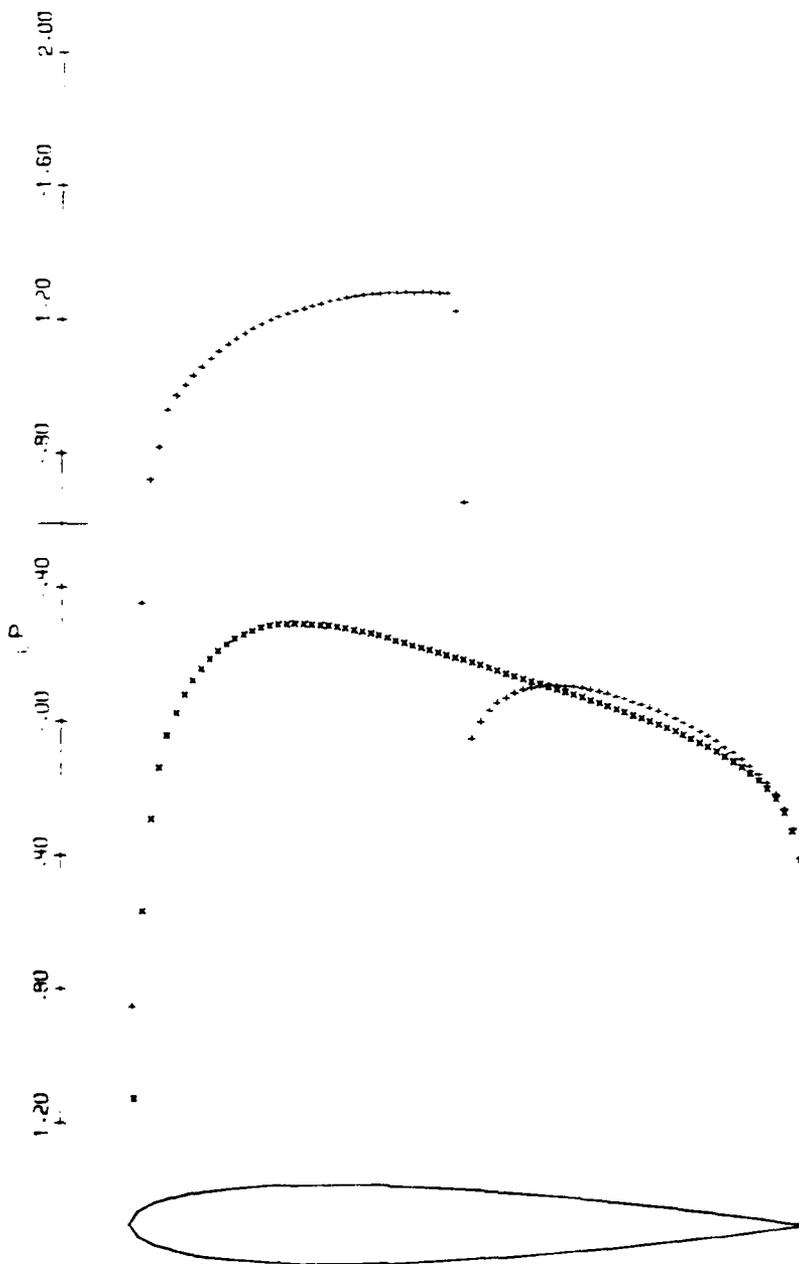
NACA 0012
MACH .750 ALPHA 2.000
CL .2490 CD .0361 CM .0052

Figure 16. (Cont.)



NACA 0012			
MACH	.750	ALPHA	2.000
RESID1	.229E-02	RESID2	.351E-04
WORK	499.00	RATE	.9317
GRID	128 X 32		

Figure 16. (Cont.)



NACA 0012
MACH .750 ALPHA 2.000
CL .4723 CD .0235 CM -.0084

Figure 16. (Cont.)

slow. Although the fine-mesh results appear to be in general agreement with results obtained from other grids, the computing time was too large.

Several possible reasons could cause the slow convergence. The two most likely are the treatment of the mesh singularity at the leading edge and the relatively high aspect ratio of the mesh cells in the far field caused by the narrow potential-like lines.

Unfortunately, time did not allow a thorough investigation of the reasons for the slow convergence so that the study was limited to the results presented. The advantages of this type of coordinate system are substantial, particularly for complex configurations. Therefore, the study should be continued to resolve the current problems.

6. Consistent Vortex Wake Model

The vortex wake behind a lifting surface is a contact discontinuity in the context of inviscid aerodynamics. Certain "slip conditions" are allowed across the vortex sheet. The conditions to be satisfied on the vortex sheet can be stated by a kinematic condition and a dynamic condition. The kinematic condition to be imposed is that the vortex sheet is a stream surface so that no fluid is allowed to pass through. The dynamic condition is that the pressure at both sides of the surface must be equal. These two conditions together with the potential equation in the flow field determine the position of the sheet and the flow condition on both sides of the sheet, much like the determination of the shock wave in the flow field in the compressible flow.

The approximate vortex wake model is assumed in the present code FLO 30 as follows. The position of the vortex wake is assumed to come off the trailing edge at its bisection angle and follow a prescribed trajectory downstream. Thus, the kinematic condition is satisfied only at the trailing edge. The flow is allowed to pass through the wake. The dynamic pressure continuity condition is also treated in an approximate manner. The numerical procedure in the code implies that there is no pressure jump to first order. That is enforced by requiring the component of velocity in the X (freestream) direction to be continuous across the wake. Also the Y-component of velocity is forced to be continuous (no sources in the wake). The possible source of error in the dynamic condition across the wake is the contribution to the pressure by the Z-component of the velocity.

In the following, we shall make an estimation of the error based on the parameters in the practical range. The deviation of the pressure from the freestream value to the second order can be written as:

$$p' = O(u') + O(u'^2) + O(v'^2) + O(w'^2) \quad . \quad (22)$$

The first three terms on the right-hand side are continuous across the wake. The possible error may come from the last term. Let us make an estimation based on the lifting line theory.

-42-

The circulation Γ at a certain section can be expressed in terms of the sectional lift coefficient C_ℓ as

$$\rho_\infty U_\infty \Gamma = \frac{1}{2} \rho_\infty U_\infty^2 c C_\ell \quad (23)$$

where c is the chord length.

The shed vortex is the spanwise derivative of Γ :

$$\sigma = \frac{d\Gamma}{dy} = \frac{1}{2} U_\infty \frac{d}{dy} (C_\ell c) \quad (24)$$

Let $\overline{C_\ell c}$ be the reference quantity. Then:

$$\sigma \sim \frac{1}{2} U_\infty \frac{\overline{C_\ell c}}{\frac{1}{2} b} = U_\infty \frac{\overline{C_\ell}}{AR} \quad (25)$$

where b is the wing span and AR is the wing's aspect ratio.

The perturbation velocity w' is related to σ by:

$$w' = \frac{1}{2} \frac{\sigma}{U_\infty} = \frac{\overline{C_\ell}}{2AR} \quad (26)$$

Compare the contribution of w'^2 to the perturbation pressure p^1 to that of $u' = 0(\overline{C_\ell})$, we have:

$$\frac{w'^2}{u'^2} = 0 \left(\frac{\overline{C_\ell}}{4 A^2 R^2} \right) \quad (27)$$

For $\overline{C_\ell} = 0.3$ and $AR = 4$, the contribution from the cross-flow component w'^2 of velocity to the pressure is only 0.5 percent. This estimation is based on the assumption that the w'^2 component is completely wrong or ignores.

To actually demonstrate that the pressures computed on two sides of the vortex sheet satisfy the dynamic condition to a high degree of accuracy, we have printed the pressure coefficients as well as the velocity components U , V , and W on either side of the sheet.

For an M-6 wing with a Mach number of 0.84 and an angle of attack of 3.06° , the largest error $(\Delta C_p / C_p)_{\max}$ is 0.014. We do not believe this error will cause a significant impact on the calculated pressures on the solid surfaces.

An inspection of the computational results in the w' component of the velocity at both sides of the wake shows that they are indeed approximately equal and opposite in the far field where the lifting line theory is applicable. Their contributions to the pressure disturbance will be equal across the wake.

The numerical computation shows that the approximate wake model in the code is adequate for all practical purposes. The improvement of the model either in kinematic condition or in dynamic condition requires substantial effort which is not justifiable in terms of the actual improvement in the accuracy of the computation.

Another consequence of assuming a wake position comes into account when considering the effect of a wake on the horizontal tail. If we assume that the wing span loading is approximately elliptical, then the downwash induced by the wake in the Trefftz plane will be uniform across the wake. If we now examine the variation of the downwash as a function of the vertical displacement from the wake at the wake centerline

$$w = \frac{1}{2\pi} \int_{-1}^1 \frac{-x}{\sqrt{1-x^2}} \frac{x}{x^2 + y^2} dx \quad (28)$$

we find that for a 0.1 semispan displacement, the induced downwash is 0.9 times the downwash on the wake. At 0.2 semispans away, the downwash drops to 0.8. Since the downwash in general represents a fraction of the angle of attack,

$$\epsilon_T = \frac{4}{AR + 2} \quad (\text{lifting line}), \quad (29)$$

we see that the effect of wake displacement is small for even displacements as large as 0.2 semispans.

The net result is that the exact wake location is not necessary to model the flow on the horizontal tail. Of course this analysis assumes that the span of the tail is less than the span of the wing. If this assumption is violated, then not only will the vertical variation be stronger, the effects of wake roll-up will also become very important.

7. Generation of Embedded Tail-Plane Mesh

7.1 Introduction

Due to the complexity of the general geometry of an aircraft, it is felt that it will be extremely difficult, if not impossible, to describe the flow space around it by a single mesh system, particularly when the mesh must be subjected to several constraints. For finite-volume computation of transonic flow, the following requirements on the mesh system must be satisfied.

- (1) Any aircraft surface must be a coordinate surface.
- (2) The mesh lines near wings and tail planes must wrap around the leading edge.
- (3) The mesh must be dense near the expected regions of fast variation, i.e., the leading edges and the expected position of shock waves.
- (4) The vortex sheets behind lifting surfaces must be coordinate surfaces.

The FLO 30 code developed in the present contract satisfies the above requirements for the wing-body combination. But, it is unlikely that one can extend the mesh-generation scheme used in FLO 30 to include tail planes and other appendices.

Instead, thoughts are given to the approach of embedded mesh systems and interactive computation, as described in the following. Local meshes may be installed around the tail planes and other appendices. These local meshes will be embedded in the wing-body mesh of FLO 30 by using common boundaries. Computations can be performed separately in the local meshes, and the interaction between the wing-body computation and the local computations can be made by iterating through common boundary values.

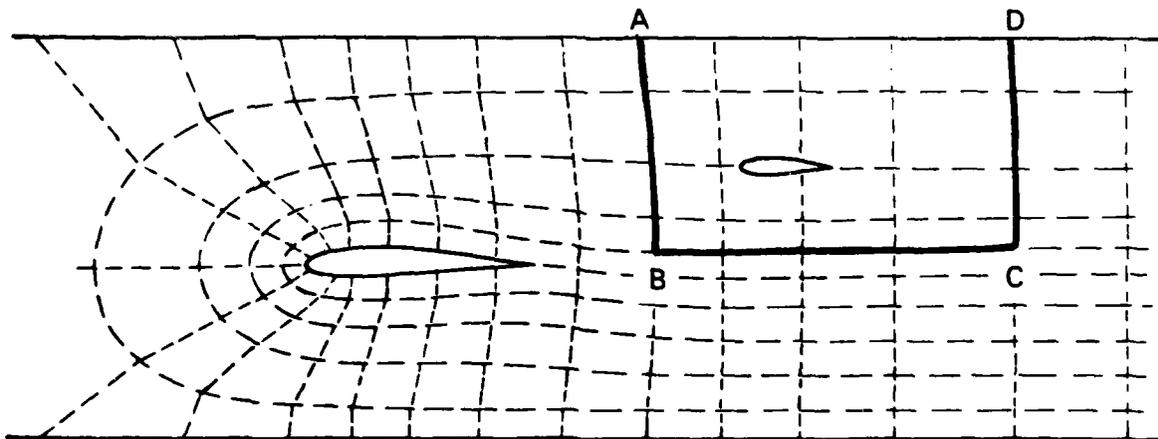
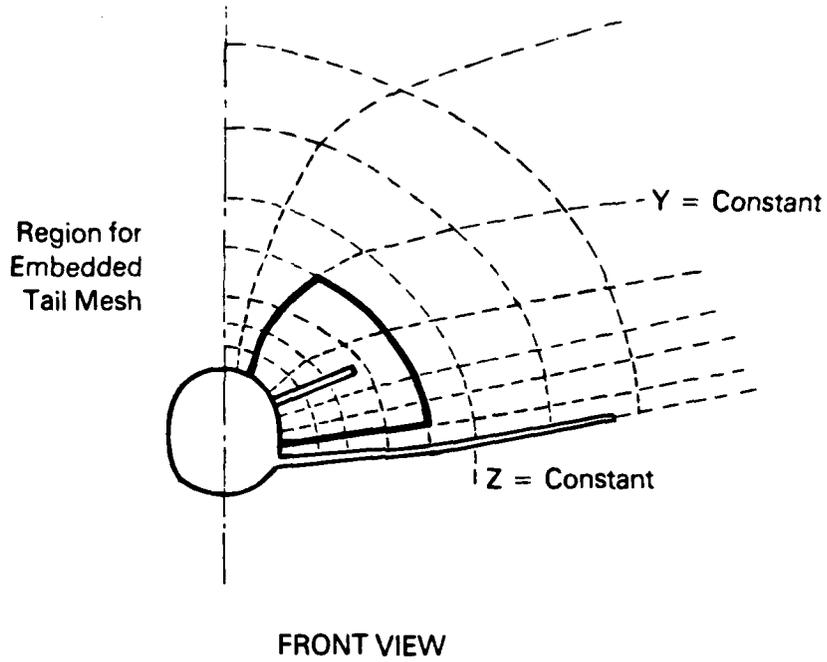
To demonstrate the feasibility of the idea given above, the local mesh around a tail plane will be chosen as an example. We shall first discuss the common boundaries in three-dimensional space for both the tail-on-fuselage and the high-tail cases. The mesh system on a "two-dimensional cut" will be generated. A two-dimensional finite-volume computation will be performed on the generated mesh system to demonstrate its computational quality. Suggestions on the method of interactive computation will be given.

7.2 Embedded Tail-Plane Mesh

Two views of the tail-plane location in the wing-body mesh of FLO 30 are shown in Figure 17 for the tail-on-fuselage case. It is natural to choose the fuselage surface and a spanwise surface $Z = Z_0$ as two boundaries of the tail mesh. The other four tail mesh boundaries can be chosen as the box ABCD shown in Figure 17. Within the tail region, we can again choose the $Z = \text{constant}$ surfaces as the coordinate surfaces, although it may be desirable to increase the number of spanwise stations locally. Now on the $Z = \text{constant}$ surfaces, the tail plane becomes a wing-in-a-box configuration. A family of wraparound C-type coordinate lines must be generated so that they vary continuously from the shape of the airfoil to the rectangular shape ABCD. Here, it should be remembered that any deviation of the shape of ABCD from a rectangle can be accounted for by a shearing transformation. In the rest of the work, we shall assume that this transformation has been done. The generation of the mesh on the $Z = \text{constant}$ surface will be discussed in Section 7.3.

For the high-tail case, it is advisable to modify the $Z = \text{constant}$ surface in the wing-body mesh so that the tail planes are on the surface, as shown in Figure 18. This modification can be done either by shearing transformation or by use of the "super ellipse", similar to that used later in Section 7.3. In this case, the tail region is bounded on two sides by the plane of symmetry and the $Y = Y_0$ surface as shown in Figure 18. Now, we can use $Y = \text{constant}$ surfaces as our coordinate surfaces in the local mesh. On these surfaces, again, the wing-in-a-box configuration is obtained.

From the discussion given above, it is clear that the first step toward the goal is to generate a two-dimensional local mesh for the wing-in-a-box configuration.



Developed Surface of $Z = \text{Constant}$ with Lateral Dimension Normalized.

Figure 17. Tail-On-Fuselage.

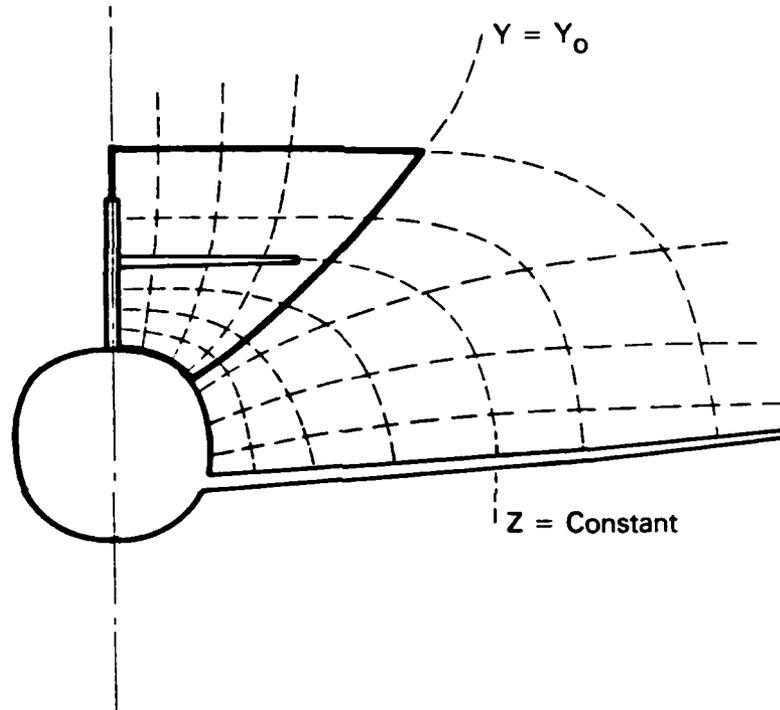


Figure 18. High-Tail.

7.3 Grid Generation

As discussed in Section 7.2, the mesh in the tail region consists of planes on which the cross section of the tail plane and the common boundaries with the wing-body mesh form the wing-in-a-box configuration. In this section, we shall generate the mesh lines in this plane.

7.3.1 Wraparound C-Type Lines

Consider a family of curves

$$\left(\frac{X}{A}\right)^m + \left(\frac{Y}{B}\right)^n = 1, \quad (30)$$

where m , n , A , and B are the parameters with A greater than B . These curves are a family of super ellipses with major axis A and minor axis B . For $m = n = 2$, it is the regular ellipse. As m approaches infinity and n approaches infinity, it becomes a rectangle with $2A$ and $2B$ as its sides. If m and n are varied from 2 to infinity, the family of curves will vary continuously from an ellipse to a rectangle.

In this work we shall use this property of the super ellipse to generate the C-type mesh line for an airfoil in a box.

Let $Y \in (0, 1)$ be the computational variable with $Y = 0$ on the airfoil surface and $Y = 1$ on the rectangular outer boundary. In Equation (30), the parameters A , B , m , and n are assumed to be functions of the computational variable Y . Consider a family of super ellipses with the center at the trailing edge. On the airfoil surface $Y = 0$, we choose a regular ellipse $m(0) = n(0) = 2$ with the major axis $A(0) = L_c$, where L_c is the chord length of the airfoil. The minor axis $B(0)$ is chosen so that the curvature of the ellipse at the leading edge $X = -L_c$ matches that of the given airfoil. For $Y > 0$, the parameters are given the following variations:

$$m(Y) = n(Y) = \frac{2}{\sqrt{1-Y}} \quad (31)$$

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} A(0) \\ B(0) \end{pmatrix} + \begin{pmatrix} A(1) - A(0) \\ B(1) - B(0) \end{pmatrix} F(Y) \quad (32)$$

where $F(Y)$ is a monotonically increasing polynomial function with $F(0) = 0$ and $F(1) = 1$. These ellipses are terminated at the trailing edge forming a family of C-type mesh lines which vary continuously from a regular ellipse on the airfoil to a rectangular shape as the outer boundary is approached. Now the $Y = 0$ line is oscillating to the airfoil at the nose and deviating from the given airfoil on other points. A simple shear transformation moves the $Y = 0$ line to the airfoil surface. These C-type lines are first continued horizontally into the wake region. As it turns out, this process produces an extremely large aspect ratio $\frac{\Delta x}{\Delta y} \gg 1$ of the mesh cell as one proceeds into the far wake. The large aspect ratio of the mesh causes the numerical scheme to diverge for the two-dimensional lifting case.

To avoid this difficulty, the vertical spacing of the mesh lines is adjusted smoothly to a new distribution within a distance of the order of one chord length downstream of the trailing edge. This smooth transition is achieved by use of $\tanh^2(x - x_{TE})$.

7.3.2 X = Constant Lines

To form a computational mesh, another family of curves intersecting the C-type mesh lines must be generated. The generation of this family of curves can be described in two parts.

The first part is the generation of a mesh upstream of the trailing edge. For each "loop" where $Y = \text{constant}$, the parametric form of the super ellipse $X(s)$ and $Y(s)$ is computed, where (X, Y) is the physical coordinate and $S \in (0, 1)$ is the normalized arc length. On the airfoil surface where $Y = 0$, the arc length distribution $S_i(Y = 0)$ of mesh points is chosen so that the meshes are concentrated near the vicinity of the leading and trailing edges. On the outer boundary where $Y = 1$, the other arc length distribution $S_i(Y = 1)$ is also chosen. For intermediate "loops" where $0 < Y < 1$, the combinations of the two

$$S_i(Y) = (1 - f(Y)) S_i(Y = 0) + f(Y) S_i(Y = 1) \quad (33)$$

are used where $f(Y)$ is a monotonic polynomial with $f(0) = 0$ and $f(1) = 1$. The choices of $S_i(Y = 1)$ and $f(Y)$ are quite arbitrary.

They are chosen in such a way that the resulting mesh is not unreasonably skewed as judged by the visual inspection.

Figure 19 shows the mesh generated by the method discussed above. Figure 20 shows the detail of the mesh lines in the neighborhood of the airfoil. The mesh lines are nonorthogonal though they are not unreasonably skewed.

It is felt that the success of a mesh-generation scheme must be demonstrated by actual finite-volume computations of the transonic flow in the supercritical regime.

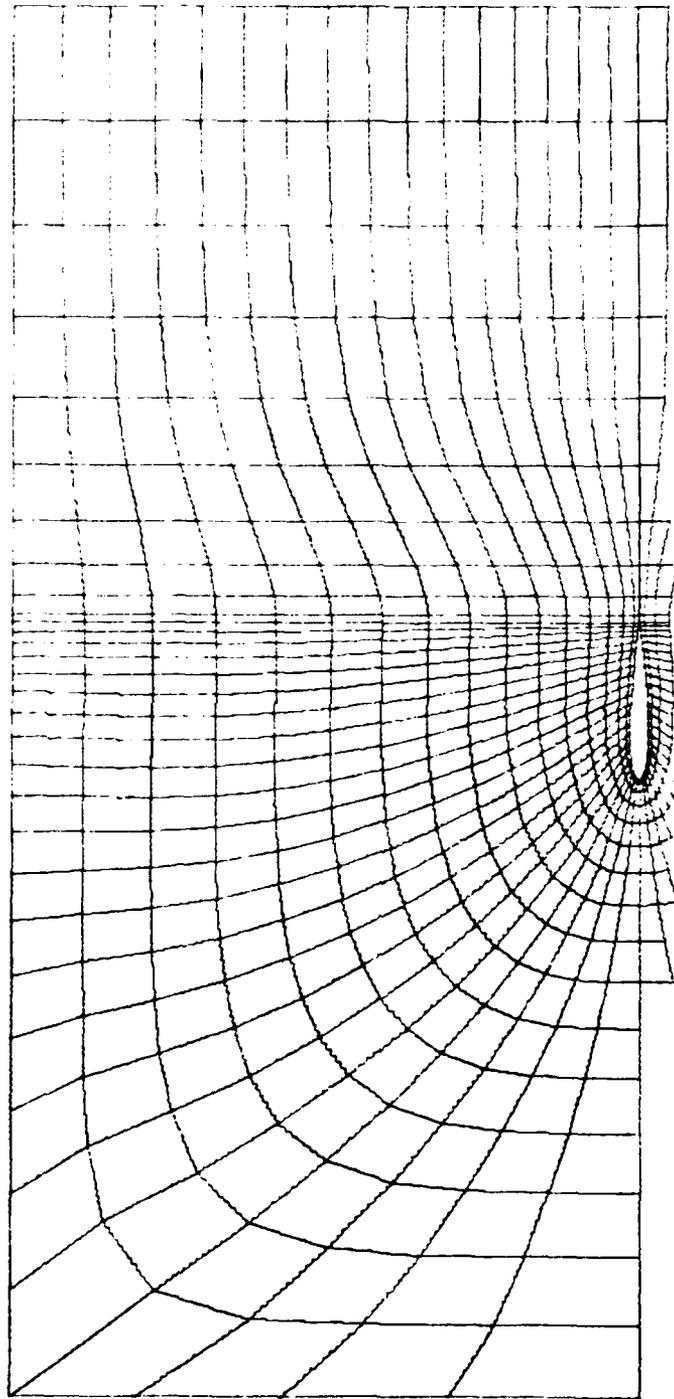
7.4 Finite-Volume Computation with Tail-Plane Mesh

It has been mentioned by Jameson and Caughey (1977) and demonstrated by Jameson and Jou (1979) that the smoothness of the mesh is a requirement of the finite-volume computation using the artificial viscosity model. The discontinuity of the transformation matrix x_x can be tolerated by the subsonic flow but will give erratic results if the flow in the region of mesh singularity is supersonic.

To demonstrate the success of the mesh-generation scheme developed here, the two-dimensional finite-volume code by Jameson, FLO 26, is modified to accommodate the new mesh.

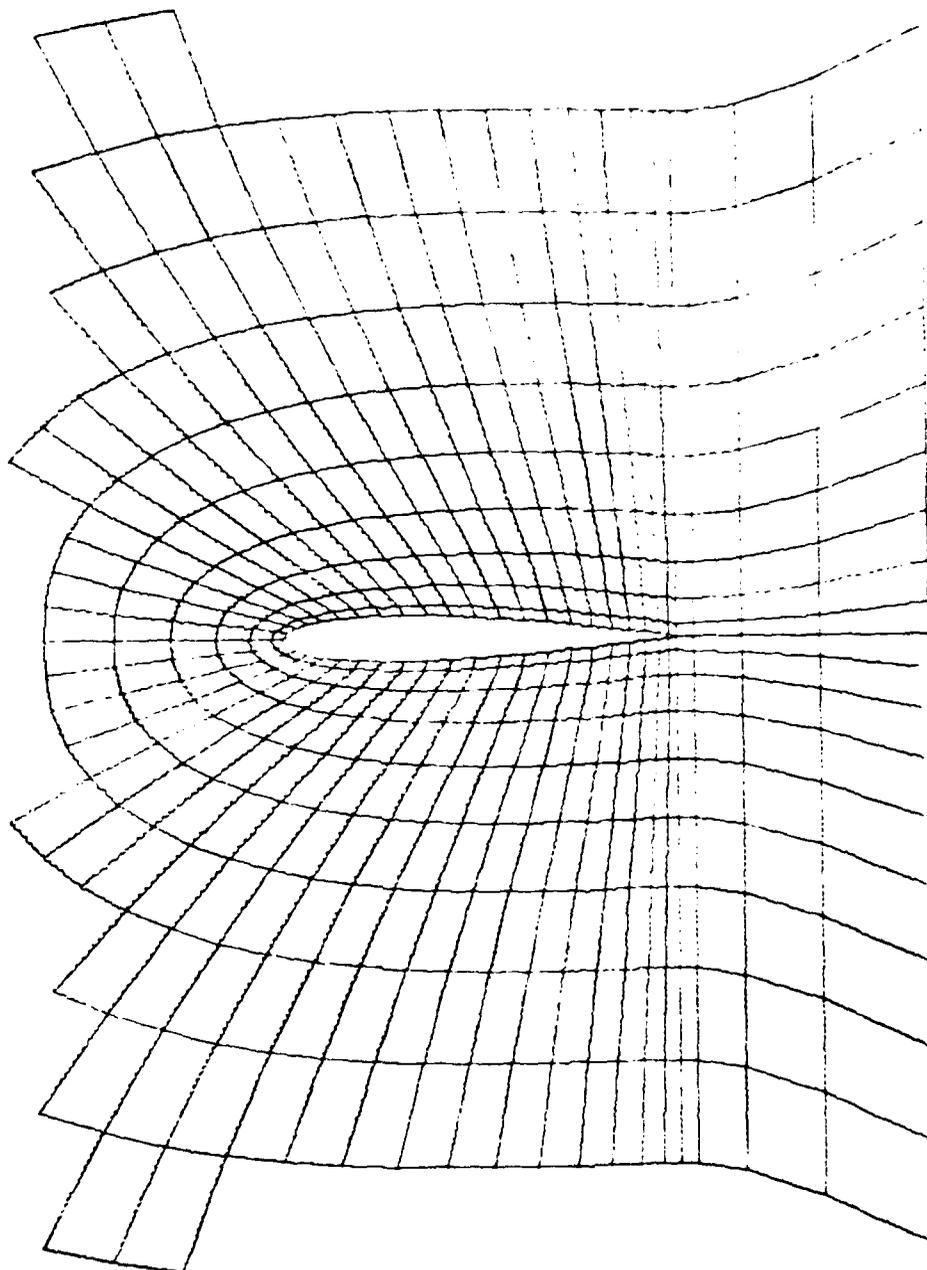
The calculations are performed in the rectangular region (9 chord lengths by 8 chord lengths) in terms of the NACA 0012 airfoil. Figure 21 shows the pressure distribution for the nonlifting case where the Mach number is 0.85 and the angle of attack is 0° . Figure 22 shows the pressure distribution for the lifting case where the Mach number is 0.75 and the angle of attack is 2° . Both are in good agreement with the computations performed by using the original parabolic mesh system.

It is demonstrated that the mesh generated here for a wing-in-a-box configuration gives good computational results for supercritical flow. The only place of mesh discontinuity is along the vertical line through the trailing edge. For the usual operating Mach number for which the potential flow computation is useful, the flow at the trailing edge is subsonic. There should not be any difficulty associated with the mesh property there.



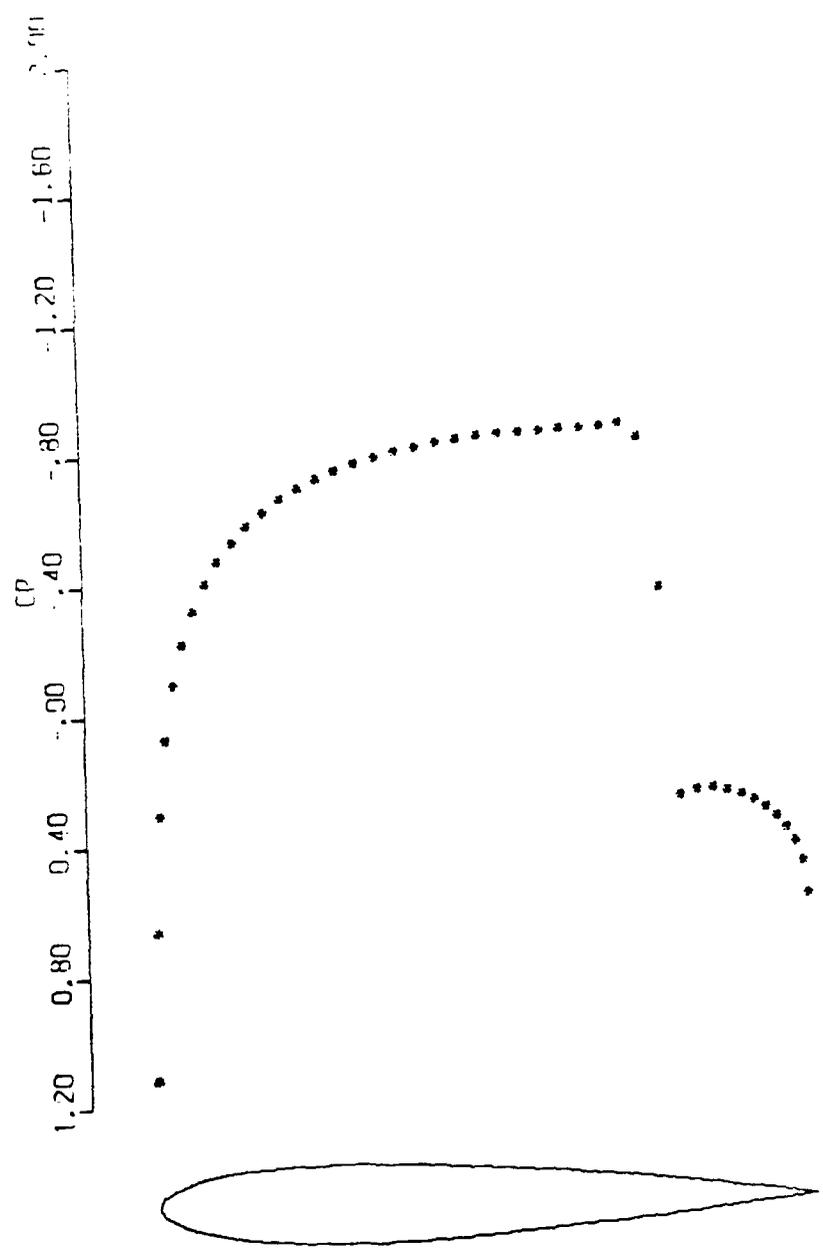
NACA 0012
GRID 64 X 16

Figure 19. Tail-Plane Mesh.



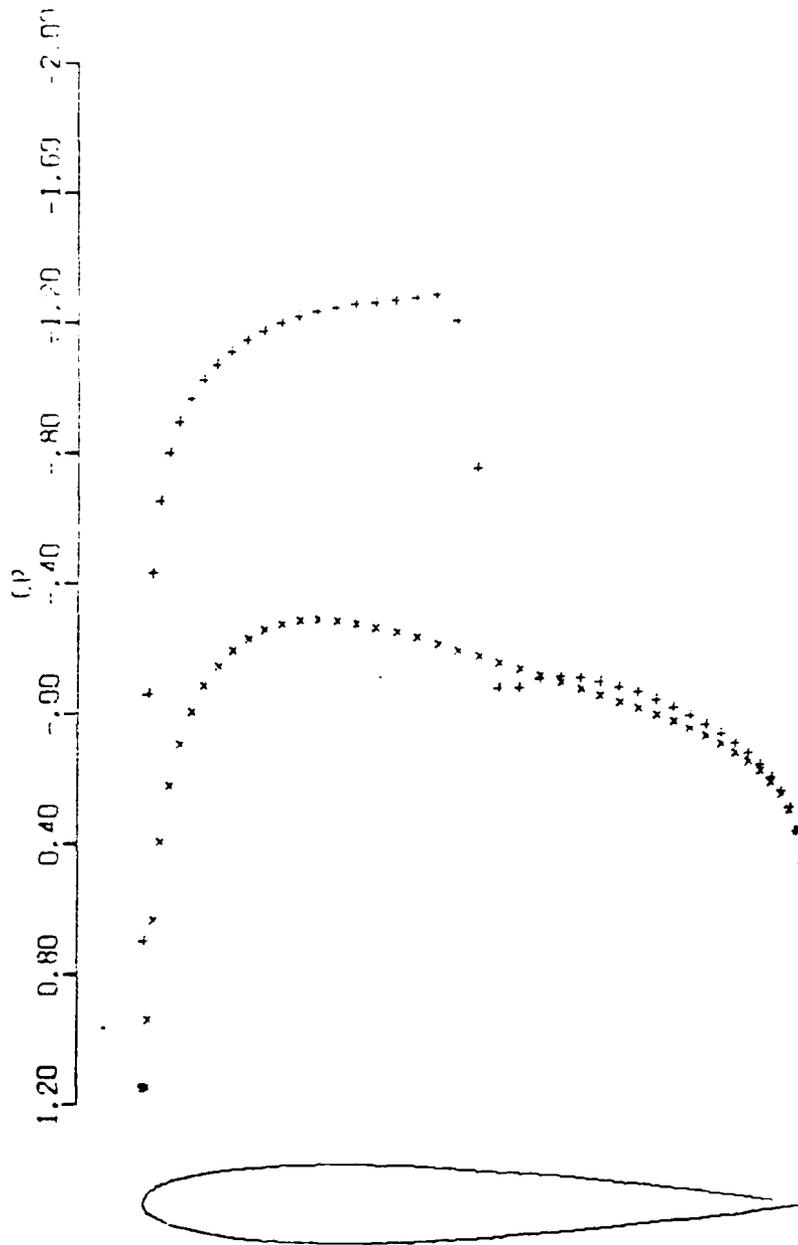
NACA 0012
GRID 64 X 16

Figure 20. Detail of the Mesh.



NACA 0012
MACH .850 ALPHA 0.000
CL .0000 CD .0004 CM .0000

Figure 21. Finite-Volume Computation on Tail-Plane Mesh (No Lift).



NACA 0012
MACH .750 ALPHA 2.000
CL .5030 CD .0145 CM -.0100

Figure 22. Finite-Volume Computation on Tail-Plane Mesh (with Lift).

8. Conclusions

Over the past two years of development much progress has been made with the full-potential transonic wing-body code. Several aircraft companies have used the code to develop confidence in it as a design and analysis tool. The work has been very successful in showing good comparisons with available experimental data. Issues regarding modeling and computed results raised on the Phase I effort were resolved in the Phase II effort. The general direction of future work has been outlined and some preliminary efforts started.

The second phase of the transonic wing-body work reported here has demonstrated that realistic configurations can be analyzed with the code developed (FLO 30). Comparisons of computed and experimental results show that the primary differences can be attributed to the effects of viscosity. FLO 30 appears to be very robust in that it has been used to model several configurations without any changes to the code.

Improvements made to the basic grid-generation scheme used in FLO 25 allow FLO 30 to treat highly swept and tapered wings. Additional testing of the code should be done to establish the limits to which the code can be applied. Other means of unsweeping the mesh upstream and downstream than the one described in this report may be required for some extreme wing configurations.

A systematic means of mesh embedding needs to be developed in order to model more complex configurations. Some of the techniques described in this report could be generalized to meet the requirements for horizontal tails and nacelles. The next stage in the code development should be to model a wing-body-tail configuration. This would provide an opportunity to resolve the techniques of grid fitting and mesh patching which will be necessary. Some work has already been done on this problem (see the appendix), but more effort remains to see what is practical.

References

- Bailey, F. R., and Ballhaus, W. F. (1975) "Comparisons of Computed and Experimental Pressures for Transonic Flows About Isolated Wings and Wing Fuselage Configurations," NASA SP-347, Part II, March, pp. 1213-1232.
- Boppe, E. W., and Stern, M. A. (1980) "Simulated Transonic Flows for Aircraft with Nacelles, Pylons and Winglets," AIAA Paper 80-0130, January.
- Caughey, D. A., and Jameson, A. (1979) "Recent Progress in Finite-Volume Calculations for Wing-Fuselage Combinations," AIAA Paper 79-1513, July.
- Caughey, D. A., Jameson, A., and Nixon, D. (1979) "Development of Finite Volume Methods for Three-Dimensional Transonic Flows," Flow Research Report No. 134, February.
- Haney, H. P., and Johnson, R. R. (1980) "Application of Numerical Optimization to the Design of Wings with Specified Pressure Distributions," NASA Contractor Report 3238.
- Haney, H. P., Johnson, R. R., and Hicks, R. M. (1979) "Computational Optimization and Wind Tunnel Test of Transonic Wing Designs," AIAA Paper 79-0080, January.
- Hinson, B. L., and Burdges, K. R. (1980) "An Evaluation of Three-Dimensional Transonic Codes Using New Correlation-Tailored Test Data," AIAA Paper 80-0003, January.
- Jameson, A., and Caughey, D. A. (1977) "A Finite Volume Method for Transonic Potential Flow Calculations," AIAA Paper 77-635 in Proceedings of AIAA 3rd Computational Fluid Dynamics Conference, June, pp. 35-54.
- Jameson, A., and Jou, W.-H. (1979) Unpublished results presented at FFA Meeting, Stockholm, September.
- Mason, W., MacKenzie, D. A., Stern, M. A., and Johnson, J. K. (1978) "A Numerical Three-Dimensional Viscous Transonic Wing-Body Analysis and Design Tool," AIAA Paper 78-101, January.
- Mercer, J. E., and Murman, E. M. (1980) "Application of Transonic Potential Calculations to Aircraft and Wind Tunnel Configurations," AGARD Conference on Subsonic/Transonic Configuration Aerodynamics, May 5-7.

Appendix: A Preliminary Study of an Overlapping
Grid-Generation System

Abstract

A study is made of existing and novel grid-generation schemes for complex airplane configurations. It is suggested that existing grid-generation schemes all have disadvantages when complex multicomponent bodies are considered. Accordingly, the concept of an overlapping mesh system has been developed in which a mean optimum grid for each component of the body is generated and these grids are coupled by an overlapping system. A preliminary example of a wing-body calculation using this type of grid system is presented. An example of a horizontal tail grid embedded in a wing grid is also presented, although no flow calculations have been programmed.

Table of Contents

	Page
Abstract	A-i
List of Figures	A-iii
1. Introduction	A-1
2. Mesh-Generation System	A-4
2.1 Jameson-Caughey Parabolic Mesh	A-6
2.2 Caughey-Jameson Cylindrical Mesh	A-8
2.3 Numerical Generation of Coordinates	A-8
2.4 Matching Mesh Systems	A-10
2.5 Overlapping Mesh Systems	A-12
3. Overlapping Grid System	A-15
3.1 Statement of the Problem	A-15
3.2 Overlapping Mesh Scheme for an Arbitrary Semiclosed Body-Wing Combination	A-19
3.3 Example	A-25
4. Grid-Generation Procedure for a Tail Plane	A-33
4.1 Type and Restrictions of Tail-Plane Grid	A-33
4.2 General Formulation	A-33
5. Conclusions	A-43
References	A-45

List of Figures

	Figure	Page
Figure 1.	Test Configuration	A-5
Figure 2.	Sketch of Jameson-Caughey Grid	A-7
Figure 3.	Caughey-Jameson Cylindrical Grid	A-9
Figure 4.	Matched or Patched Grids	A-11
Figure 5.	Sketch of an Overlapping Mesh	A-13
Figure 6.	Jameson Mesh System for a Wing-Body Combination	A-21
Figure 7.	Overlapping Grid System for the Transformed Body	A-23
Figure 8.	Body Grid Along a $\xi = \xi_{23}$ Line	A-26
	Body Grid Along a $\xi = \xi_{30}$ Line	A-27
Figure 9.	Wing Grid Along a $\xi = \xi_{23}$ Line	A-28
	Wing Grid Along a $\xi = \xi_{30}$ Line	A-29
Figure 10.	Wing-Body Configuration	A-30
Figure 11.	Pressure Distribution Around a Wing-Body Combination	A-31
Figure 12.	Tail-Plane Embedding in Transformed Domain	A-35
Figure 13.	Sketch of Tail-Plane Embedded Coordinate System	A-37
Figure 14.	Embedded Tail-Plane Mesh System	A-42

1. Introduction

In order to estimate the flow distribution around a body using a finite difference computational method, a finite difference mesh or grid must first be constructed. When such finite difference computations were being developed, the main problem was the derivation of stable and rapid algorithms. The wing or airplane was represented by a somewhat primitive model. For example, wings were represented by a plane using the well-known thin-wing approximations (Bailey and Ballhaus, 1972); in such cases, the computational grid was Cartesian. For a wing-body combination (Ballhaus, Bailey, and Frick, 1976), the body was represented in a fairly crude fashion by simply taking the nearest (Cartesian) grid points. As stable algorithms were developed, there arose a need to devise more suitable computational grids.

The main criteria for judging the efficiency of a particular grid is the number of grid points required for a given accuracy, since the total computation cost decreases with the number of grid points. In practice, this requires that mesh points be clustered near regions of rapid gradient in the flow and be more sparsely distributed in regions of moderate gradients. Areas of rapid flow gradients, such as the leading edge of an airfoil, can often be specified in advance, which helps in determining the clustering. A further problem is the accurate and computationally simple representation of the body in the grid. This is best achieved by the use of a body-conforming grid in which the body surface coincides with an extreme member of one family of the coordinate surface that constitutes the grid.

For two-dimensional airfoil computations, a circle plane mapping which transformed the flow field exterior to the airfoil to the interior of a unit circle (Sells, 1968) proved very satisfactory. The actual grid is constructed by a net of concentric circles and radial lines. In addition to being a body-conforming grid (the unit circle identifies with the airfoil surface), this transformation clusters grid points in regions of high curvature, such as the airfoil leading edge. This particular grid is orthogonal. In principle, the circle plane mapping could be used for a three-dimensional wing by transforming each wing section to a circle, although it may be difficult to treat, for instance, a wing-body-tail configuration.

An alternative means of generating a mesh for an airfoil section is by using a parabolic-shearing mesh system. Initially it is assumed that the airfoil section is parabolic and the parabolic section is then "unwrapped" about a regular line inside the airfoil. The airfoil leading edge is approximately parabolic. This square-root transformation transforms the airfoil section to a quietly undulating curve which can then be removed by a simple shearing. This does lead to a nonorthogonal grid. Application of this grid-generation system to sections of a wing have been used by Jameson and Caughey (1977) for finite wing computations.

There are other variations of "unwrapping" grids which are similar to those discussed above. These mesh-generation systems can be termed analytic since they are complemented by a series of analytic operations. A second type of classification is numerical generation of the computational mesh in which the unit is constructed by the numerical solution of a set of partial differential equations. These equations are usually Poisson's equations for the transformed (computational) coordinates. The forcing terms are inserted to control clustering or possibly orthogonality. It is perhaps worth noting for two-dimensional grid generations that if the forcing terms are neglected and the set of equations solved subject to the Cauchy-Reamann conditions, then the numerically generated coordinates are identical to a conformal transformation generation scheme. Generally, numerically generated coordinate systems are non-orthogonal. Applications of this type of approach to three-dimensional bodies are, in principle, straightforward. The most common applications of this approach follow the work of Thompson, Thames, and Mastin (1974) and Mastin and Thompson (1978). Further developments concerned mainly with clustering and orthogonality are by Sorenson and Steger (1977; Steger and Sorenson, 1978).

The present generation of transonic flow computer codes can be used for (fairly simple) wing-body configurations. The grid-generation schemes used are based on the above techniques. For example, Jameson and Caughey (1977) transform the body, section by section, to a plane using a combination of conformal transformations and simple shearing. This transforms the problem to that of a (transformed) wing on a wall,

and the grid can then be constructed using the parabolic-shearing transformation discussed above. A recent version of this computer code is that of Caughey and Jameson (1977) in which the mesh system consists of a cylindrical coordinate surface based on the body axis followed by a logarithmic unwrapping of the wing section in the intersecting cylindrical surface. Three-dimensional solutions of the Navier-Stokes equations have been obtained in a numerically generated grid by Thompson, Thames, and Mastin (1974). The physical problems treated in these computer codes are fairly simple wing-body configurations. Even then, the existing mesh-generation schemes sometimes have disadvantages.

In Section 2, existing and novel mesh-generation schemes are critically discussed regarding their applicability to realistic configurations. It is suggested that the best system may be an overlapping mesh system in which a "master" grid is chosen (for example, around an isolated wing) and each additional component (body, tail, nacelle, etc.) has its own "slave" grid. Each slave grid ideally has common points with the master grid and possibly other slave grids, although some form of interpolation may be used. In principle, this allows the grid system to be constructed as the complexity of the configuration increases. Some general rules regarding uniqueness and orthogonality of the system are derived. The main disadvantage seems to be an increase in the number of iterations of the difference schemes, although this could well be offset by the reduced local complexity of the mesh.

A grid system for a semi-infinite body, closed at the front, with a wing has been generated using these techniques, and an example has been computed. Comparison of these results with those of Jameson and Caughey (1977) indicates that some problems in the generated grid remain but that in principle the basic approach is sound. A second grid system suitable for a wing-body horizontal tail configuration has also been prepared and the grid constructed. However, no flow computations have been programmed.

2. Mesh-Generation System

Before discussing different mesh-generation systems, perhaps it is advisable to lay down some ground rules upon which the discussion will be based. The basic object of the present work is to devise a mesh-generation system capable of treating realistic airplane configurations. Consequently, the discussion of grid generation will to some extent center around the capability to represent a wing-body combination with both a vertical and horizontal tail and with a nacelle somewhere on the wing. The configuration is sketched in Figure 1. It is also helpful to list the desirable properties of a mesh system even though it may be impossible to satisfy all such requirements. It is suggested that the following properties are desirable in any such system.

- a. The mesh system should ultimately be capable of treating realistic airplane configurations.
- b. The mesh should be rapidly generated.
- c. The mesh should have sufficient smoothness, compatible with the difference scheme.
- d. The mesh should not have excessive "skewness" which can cause numerical inaccuracies or instability.
- e. The mesh should be capable of development to an adaptive grid in order to cluster points in regions of rapid change. Also, it should ultimately be capable of alteration during a computation in order to treat unsteady effects.
- f. The mesh system should not require a high level of computer input.
- g. The mesh should reduce to a Cartesian (or universal) mesh outside the neighborhood of the airplane. Since the metrics for this universal mesh can be permanently stored in the computer, this removes the need to difference the transformation metrics in this region for each case and thus saves computing time.

These then are the general guidelines upon which the following discussion is based.

A-5

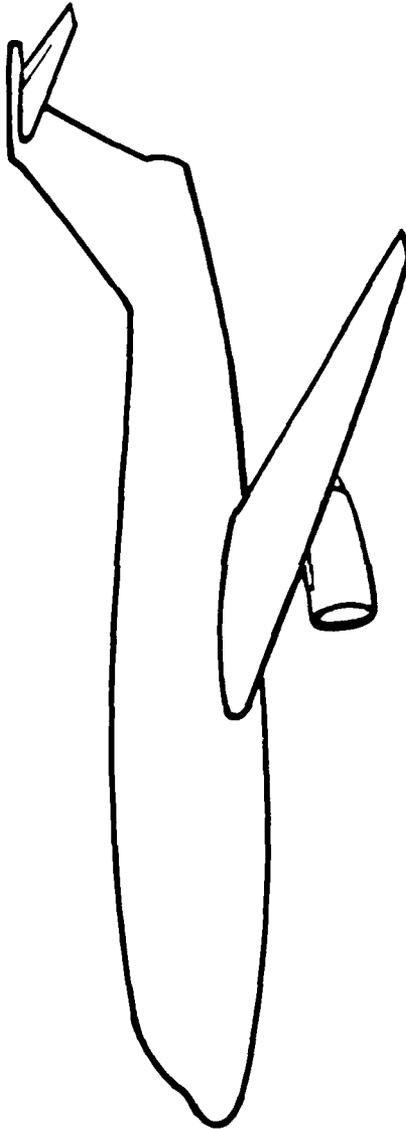


Figure 1. Test Configuration.

2.1 Jameson-Caughey Parabolic Mesh

This mesh system, described by Jameson and Caughey (1977), is capable of treating a wing/finite body combination. In this system an enveloping circle is constructed at each body section and this circle is then conformally transformed to a slit. The representation of the body is a slowly varying curve which is then collapsed to the slit by a shearing transformation. The resulting configuration is a (transformed) wing on a wall. A parabolic transformation is used to unwrap the wing about a singular line just inside the wing. The wing at any section is then represented by a slowly varying curve which is reduced to a plane by a shearing transformation. The "slowly varying curve" arises from the assumption that the rapid curvature region of the airfoil section at the leading edge can be approximated by a parabola. The wing wake geometry can be treated adequately in this system.

This parabolic grid system is satisfactory for an isolated wing, the main disadvantage being the tendency to give a coarse mesh in the physical space near the wing trailing edge. However, since this grid is "wing dominated" it can produce undesirable effects on the body. After the initial conformal and shearing transformation, the body is represented by its profile. Unless this profile coincides with one of the wing-generated "parabolic" lines, the body profile is distorted by the appearance of "fins". This is sketched in Figure 2. Although these fins diminish in size as the grid is refined, they still constitute a strong disadvantage of this grid. Furthermore, the grid spacing is determined by its desirability for the wing calculation which may not be at all suitable for certain fuselage geometries; some of the grid cells near the wing root or the symmetry plane may be highly skewed.

Turning to the problem of incorporating extra components such as a horizontal tail or a nacelle, it is not at all clear how these could be incorporated into the wing-dominated grid with satisfactory accuracy. For example, it is difficult to see how a similar wraparound grid for a horizontal tail could be incorporated into the existing grid for the wing wake. However, one advantage of analytically generated grids such as this is the speed with which it can be implemented. This would make it easily adaptable to changing flow conditions for time-varying flows.

A-7

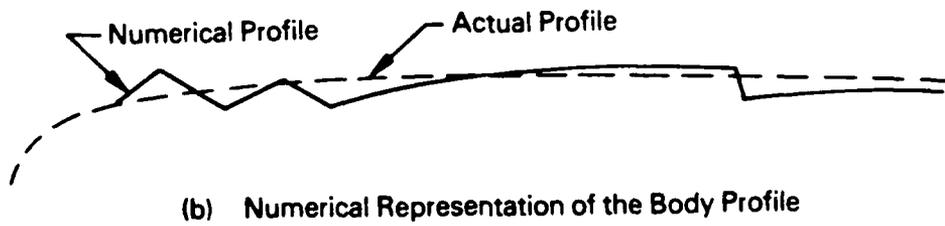
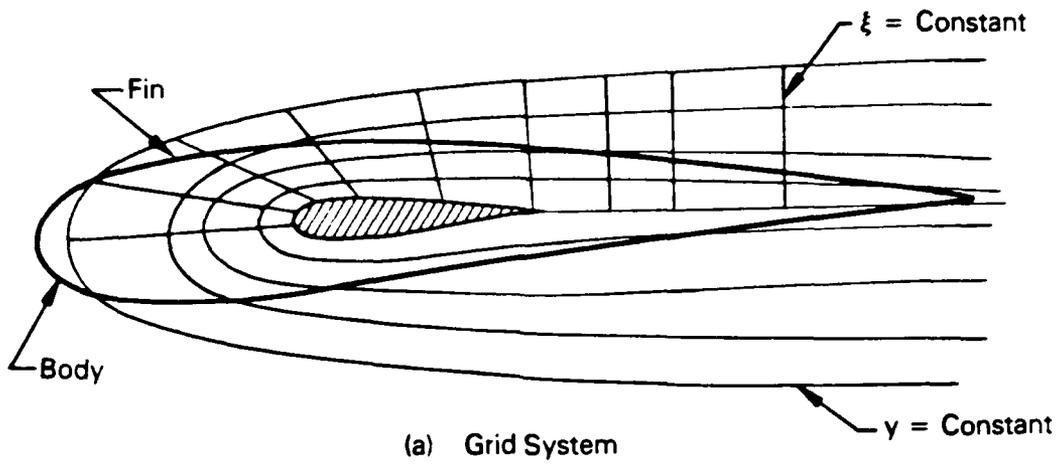


Figure 2. Sketch of the Jameson-Caughey Grid.

2.2 Caughey-Jameson Cylindrical Mesh

The grid system developed by this method is in some respects similar to the previous mesh but differs significantly in the treatment of the body. The basis of the system is to introduce a distorted cylindrical coordinate system about the axis of the body. The distortion is such that at streamwise section the body surface constitutes a coordinate line. When one of the coaxial surfaces cuts the wing, the wing section is similar to a wing between two walls, as shown in Figure 3. Each wing section is then unwrapped in a similar fashion as in the Jameson-Caughey mesh except that a logarithmic transformation is used instead of the parabolic transformation. This mesh can be termed body dominated and can cause difficulties in resolution near the wing tips if the radial surfaces are far apart. Also there are difficulties associated with a closed body since the "body radius" is then zero; special difference operators have to be used for the axial points. In practice it has been found that a very small radius can be used without difficulty. Furthermore, the location of the upstream boundary is represented in the transformed plane by a singular point so that a special treatment of this region is necessary.

2.3 Numerical Generation of Coordinates

The basic idea is to solve elliptic partial differential equations for the computational coordinates. In two dimensions, two equations are required while for three dimensions, three equations are required. Since the basic equations are elliptic, values of the transformed coordinates can be specified on the boundaries; one of these coordinate surfaces is taken to be coincident with the body or bodies. Examples of this method of grid generation are given in Thompson, Thames, and Mastin (1974); Mastin and Thompson (1978); Sorenson and Steger (1977); and Steger and Sorenson (1978). Initially, the elliptic differential equations were Laplace's equation, but this did not really allow much flexibility in clustering the grid points at will, for example in regions of high curvature. Consequently, nonhomogeneous terms were added to the equations so that the desired clustering occurred. In spite of these modifications, the grids generated can exhibit undesirable features. For example, in two-dimensional applications the generated

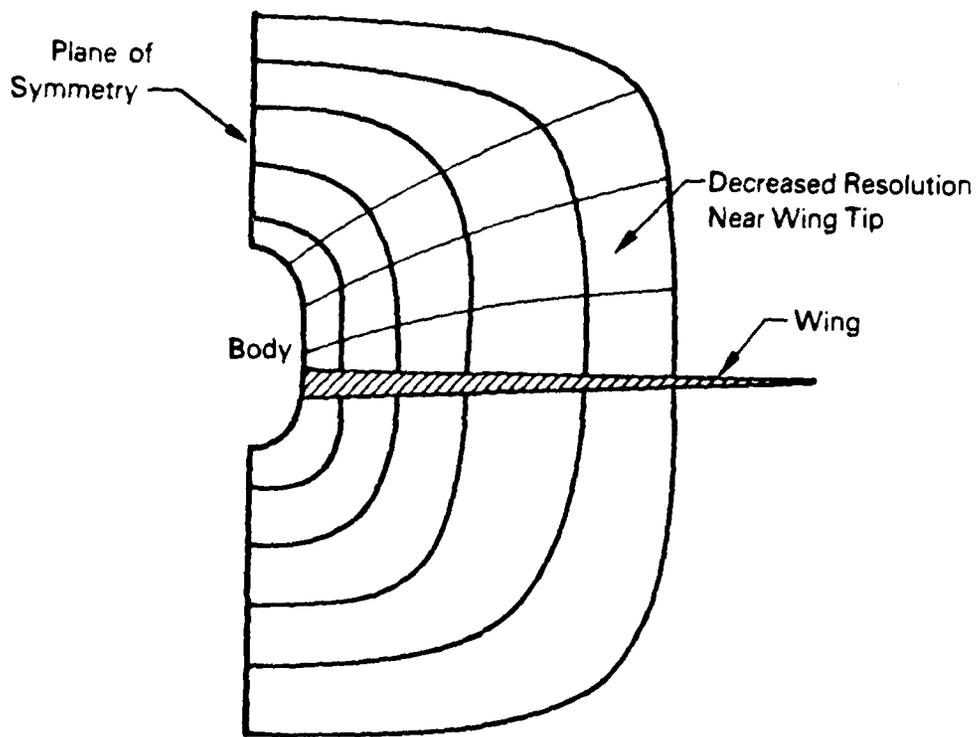


Figure 3. Caughey-Jameson Cylindrical Grid.

grid may have excessive skewness (Sorenson and Steger, 1977) or the clustering may still be inadequate. However, at least in two dimensions, it is possible to introduce some orthogonality condition into the equations which alleviates the problem of skewness (see Steger and Sorenson, 1978). It is not clear whether this technique can be applied to complex three-dimensional configurations.

Generally, numerically generated coordinate systems can be altered during the iteration or for a time-dependent computation. All that is required is that the equations for the coordinate system be solved simultaneously with the equations governing the physical process. If the physical equations are complex, e.g., the Navier-Stokes equations, these additional equations for the mesh do not require a significant portion of the total computing time. However, for potential equation calculations the additional complexity of numerically generated coordinate systems may increase the total computation time significantly.

2.4 Matching Mesh Systems

One possible grid-generation system is to have a separate system for each component and match or patch each individual grid at the interface in such a way as to minimize any undesirable interference difficulties. A sketch of the proposed scheme is shown in Figure 4. From a study of the difference equations used in the finite-volume method, it seems that across each interface the grid should have at least continuity of second derivatives. Although superficially attractive because of the choice of a near optimum grid for each component and the possibility of reversion to a Cartesian mesh outside the body, there are some difficulties with this type of mesh. One main difficulty is the construction of the individual mesh system at junctions of two components, e.g., wing-fuselage since if the fuselage surface, say, is the inner coordinate surface for the wing, it must, at least at the wing root, also be the outer coordinate surface for this body subgrid. A second difficulty is the necessary matching of the smoothness at the interface. A third difficulty is in the ordering of the grid lines from one subgrid to another. In spite of these difficulties it is probable that a good modular mesh of this type could be constructed by the same variant of

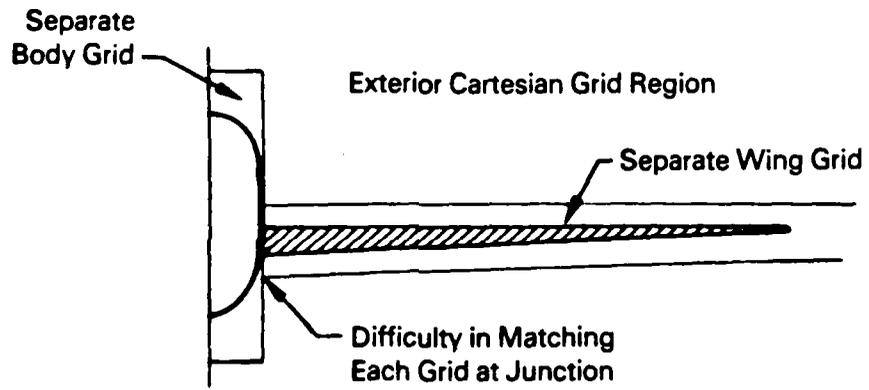


Figure 4. Matched or Patched Grids.

the numerical generation technique discussed in the previous section which allowed some control over the smoothness of the interfaces. However, this would be a major piece of work which the possible gain in efficiency may not warrant.

2.5 Overlapping Mesh Systems

The overlapping mesh-generation scheme was developed to have most of the advantages of the modular grid system mentioned above but without most of the disadvantages. Thus, the overlapping grid has a nearly optimum separate grid for each component but does not have the difficulties of enforcing smoothness at the interface of the modular system or the difficulties at junctions of components. The basic idea is best illustrated in two dimensions as shown in Figure 5.

In this scheme, a master grid, say one suitable for a wing, is chosen, and a series of slave grids which are optimum for the body, tail plane, nacelle, etc., are embedded or attached to this master grid. The main requirement is that the boundary of the master grid should be either at a known physical boundary or consist of grid points that are common with one of the slave grids. A similar condition applies to each of the slave grids. By this means, an optimum grid for each component can be used in the computation. In this scheme, each component is solved in isolation with the boundary conditions on the grid boundary taking into account the interaction effects.

Any means, numerical or analytic, may be used to generate both master and slave grids, provided the rules for overlapping (discussed later) are satisfied. Note that the grids need only overlap by the two coordinate surfaces necessary to obtain a central difference with a Dirichlet boundary condition. This overlapping mesh scheme will allow not only a near optimum mesh for each component but can, in principle, allow each component to be solved to a different order of accuracy or convergence level. It will also allow additional components, such as nacelles, stores, etc., to be "plugged" in once the ground rules for the overlapping are established. Furthermore, one such overlapping could be a wing-body mesh combination with the exterior Cartesian mesh. The main disadvantage is the multiple iteration required in the overlap region.

A-13

- Wing Mesh
- Body Mesh
- Wing Mesh Boundary Points
- × Body Mesh Boundary Points

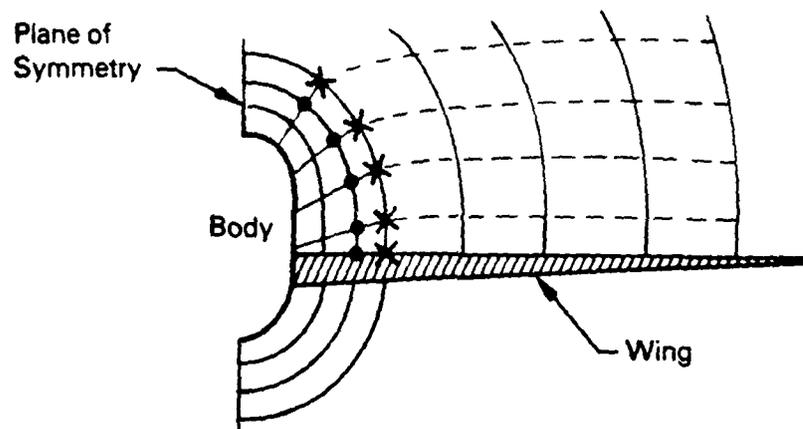


Figure 5. Sketch of an Overlapping Mesh.

This problem may not be critical since the better mesh systems may require fewer total iterations than a universal mesh. Furthermore, if the overlapping meshes are chosen in the optimum fashion, the overlap region need only consist of two coordinate surfaces. Also, if it is desirable, each component can be solved to a different convergence level if the effect of that component is required in only a global sense.

The basic idea of overlapping meshes as discussed above requires some ground rules to determine how universal the scheme is and how (if possible) to predict its characteristics. These are discussed in the next section. It is felt, however, that the overlapping mesh technique shows considerable promise for the computation of complex configurations.

3. Overlapping Grid System

If a primary grid for a particular component of the configuration is chosen, the question of determining the grid for the secondary component arises. Obviously there are certain restrictions on the choice of this second grid since one coordinate surface must be coincident with the secondary component. Also, by the definition of the overlapping grid concept, certain points in this secondary grid must be coincident with the specified points in the primary grid. In this section, the necessary restrictions or ground rules for the secondary mesh are examined by writing each grid curvilinear coordinate system as a function of known coordinate surfaces (x, y, z) and constructing the necessary equations that each coordinate surface must satisfy for the overlap to exist. Also, the degree of orthogonality and the question of a one-to-one mapping for a point in each coordinate is considered. A typical example of an overlap grid scheme is also given.

3.1 Statement of the Problem

In a three-dimensional physical space a point in a curvilinear coordinate system is defined by the point of intersection of three families of coordinate surfaces, each of which in general only intersects a member of another family once. Each member of a particular family of surfaces does not intersect another member of the same family. Each point of the physical space is therefore represented uniquely by the specification of a particular member of each of the three families of coordinate surfaces. If these coordinate surfaces intersect each other at right angles, then the coordinate system is orthogonal. It follows fairly obviously that any given point in the physical space can be represented in any coordinate system that covers all or at least the necessary part of the physical space.

As outlined in the introduction, the basic idea of the overlapping mesh system is to first choose a master grid suitable for the most important part of the configuration (e.g., the wing) and then attach slave grids suitable for the other components. The slave grids are bounded by a coordinate surface of one family representing the component and another surface of the same or another family consisting entirely of points in the master system or known far-field boundary points. The

wing grid has a bounding surface consisting entirely of points in the slave grid. Consider now two such overlapping grids.

Let the master grid, fitted to component A, be defined by the families of coordinate surfaces.

$$\left. \begin{aligned} \psi_1(x, y, z) &= E_\ell^{(1)} & \ell &= 1, N_1 \\ \psi_2(x, y, z) &= E_m^{(2)} & m &= 1, N_2 \\ \psi_3(x, y, z) &= E_n^{(3)} & n &= 1, N_3 \end{aligned} \right\} \quad (1)$$

where the parameters $E_\ell^{(1)}$, $E_m^{(2)}$, and $E_n^{(3)}$ denote the members of each of the families of coordinate surfaces ψ_1 , ψ_2 , and ψ_3 . The specifications of the parameters $E_\ell^{(1)}$, $E_m^{(2)}$, and $E_n^{(3)}$, i.e., the points of intersection of the coordinate surfaces, are the coordinates of a point in space. In the system, Equation (1), there are $N_1 N_2 N_3$ such points. It is assumed that one of the coordinate surfaces coincides with the component A, that is, the component A is defined by:

$$\psi_1(x, y, z) = E_\ell^{(1)} \quad (2)$$

Let the specified set of points (i.e., the common points of the overlap region) in this coordinate system be defined as $\bar{E}_\ell^{(1)}$, $\bar{E}_m^{(2)}$, and $\bar{E}_n^{(3)}$. Now introduce a new system of coordinate surfaces:

$$\left. \begin{aligned} \phi_1(x, y, z) &= \delta_p^{(1)} & p &= 1, N_p \\ \phi_2(x, y, z) &= \delta_q^{(2)} & q &= 1, N_q \\ \phi_3(x, y, z) &= \delta_r^{(3)} & r &= 1, N_r \end{aligned} \right\} \quad (3)$$

It is assumed that one of these coordinate surfaces, say

$$\phi_1(x, y, z) = \delta_p^{(1)} \quad (4)$$

A-17

coincides with component B. That is, the surface of component B is defined by Equation (4). The problem is to construct the slave coordinate system such that one coordinate surface, say:

$$\phi_1(x, y, z) = \delta_{Np}^{(1)}, \quad (5)$$

contains all the specified (common) points, $\bar{E}_\ell^{(1)}$, $\bar{E}_m^{(2)}$, $\bar{E}_n^{(3)}$, of the coordinate system, Equation (1).

If it is assumed that the master grid is satisfactorily close to orthogonality as regards component A, then a second problem is to control, if possible, the orthogonality of the slave coordinate system.

Third, it is imperative that this slave coordinate system be unique so that the mapping from the system in Equation (1) to the system in Equation (3) is one to one.

Any point in some domain of physical space can be described not only in the Cartesian coordinates x, y, z but also in the master grid system ψ_1, ψ_2, ψ_3 .

Now the slave coordinate system, Equation (3), can be written in terms of the master system; thus

$$\begin{aligned} \bar{\phi}_1(\psi_1, \psi_2, \psi_3) &= \delta_p^{(1)} \\ \bar{\phi}_2(\psi_1, \psi_2, \psi_3) &= \delta_q^{(2)} \\ \bar{\phi}_3(\psi_1, \psi_2, \psi_3) &= \delta_r^{(3)} \end{aligned} \quad (6)$$

If the coordinate surface, Equation (5), contains all the common points $\bar{E}_\ell^{(1)}$, $\bar{E}_m^{(2)}$, and $\bar{E}_n^{(3)}$, then the equation

$$\bar{\phi}_1\left(\bar{E}_\ell^{(1)}, \bar{E}_m^{(2)}, \bar{E}_n^{(3)}\right) = \delta_{Np} \quad (7)$$

is satisfied for all $\bar{E}_\ell^{(1)}$, $\bar{E}_m^{(2)}$, and $\bar{E}_n^{(3)}$. This is the first restriction on the choice of the slave grid surface ϕ_1 . Also, the

specific points on the surface must be contained in the surface, hence the choice of the other coordinate surfaces, ϕ_2 and ϕ_3 , must be such that the equations

$$\begin{aligned} \bar{\phi}_2 \left(\bar{E}_\ell^{(1)}, \bar{E}_m^{(2)}, \bar{E}_n^{(3)} \right) &= \delta_q^{(2)} \quad \text{for all } q \\ \bar{\phi}_3 \left(\bar{E}_\ell^{(1)}, \bar{E}_m^{(2)}, \bar{E}_n^{(3)} \right) &= \delta_r^{(3)} \quad \text{for all } r \end{aligned} \quad (8)$$

are satisfied in total for all $\bar{E}_\ell^{(1)}$, $\bar{E}_m^{(2)}$, and $\bar{E}_n^{(3)}$. This ensures that every coordinate point on the surface

$$\bar{\phi}_1 (\psi_1, \psi_2, \psi_3) = \delta_{Np}^{(1)}$$

coincides with one of the specified points, $\bar{E}_\ell^{(1)}$, $\bar{E}_m^{(2)}$, or $\bar{E}_n^{(3)}$, of the master system. The primary requirements for the above grid are therefore that it must satisfy Equations (5), (7), and (8). Other considerations are that it must be smooth, unique, and possibly orthogonal. It is assumed that a smooth mesh can be constructed; the other points will be discussed later. As well as requirements for the slave grid, there are some requirements for the master grid; that is, one extreme of the master grid must consist solely of points in the slave grid. This requirement can be easily formulated by writing the master coordinate system in terms of the slave coordinates. Thus, if the specified overlap points in the slave grid are defined by $\bar{\delta}_p^{(1)}$, $\bar{\delta}_q^{(2)}$, and $\bar{\delta}_r^{(3)}$, and if the extreme surface of the master grid is chosen to be

$$\bar{\psi}_3 (\phi_1, \phi_2, \phi_3) = E_n^{(3)}, \quad (9)$$

then the master grid must be such that the equation

$$\bar{\psi}_3 \left(\bar{\delta}_p^{(1)}, \bar{\delta}_q^{(2)}, \bar{\delta}_r^{(3)} \right) = E_n^{(3)} \quad (10)$$

is satisfied for all values of $\bar{\delta}_p^{(1)}$, $\bar{\delta}_q^{(2)}$, and $\bar{\delta}_r^{(3)}$. Also, by the same argument as the slave grid, the equations

$$\begin{aligned} \bar{\psi}_1 \left(\bar{\delta}_p^{(1)}, \bar{\delta}_q^{(2)}, \bar{\delta}_r^{(3)} \right) &= E_\ell^{(1)} \quad \text{for all } \ell \\ \bar{\psi}_2 \left(\bar{\delta}_p^{(1)}, \bar{\delta}_q^{(2)}, \bar{\delta}_r^{(3)} \right) &= E_m^{(2)} \quad \text{for all } m \end{aligned} \quad (11)$$

must be satisfied in total for all $\bar{\delta}_p^{(1)}$, $\bar{\delta}_q^{(2)}$, and $\bar{\delta}_r^{(3)}$. Thus, the master grid must satisfy Equations (2), (10), and (11). If the master grid is single valued, then the slave grid is single value (McConnell, 1957) if the Jacobian of the transformation from the master coordinate system is nonzero. Thus, if

$$J = \left| \frac{\partial \bar{\phi}_i}{\partial \psi_j} \right| \neq 0 \quad (\text{all } i, j), \quad (12)$$

then the slave grid is single valued. Also the mesh is orthogonal if

$$\frac{g_{ij}}{[g_{ij}g_{ii}]^{1/2}} = 0 \quad (i \neq j) \quad (13)$$

where g_{ij} is the transformation metric given by

$$g_{ij} = \frac{\partial \bar{\phi}_k}{\partial \psi_i} \frac{\partial \bar{\phi}_k}{\partial \psi_j} \quad (14)$$

The above analysis gives the requirements for an overlapping mesh system. The next problem is to devise some means of constructing such a mesh system either numerically or analytically.

3.2 Overlapping Mesh Scheme for an Arbitrary Semiclosed Body-Wing Combination

It is proposed in this section to indicate how to construct an overlapping mesh system for a wing-body combination when the body is a semiclosed arbitrary shape. The body is closed at the front and open at the rear, a configuration typical of a representation of a closed fuselage with a wake model. A wing is mounted at any location on the body. The object of this section is to suggest a scheme for generating a body-conforming grid for the body matched to a wing grid.

In the Jameson-Caughey computer code, the body is transformed to a slit by means of a Joukowski transformation in conjunction with a simple shearing. This reduces the problem of grid generation to that for a wing on a wall. In this transformed domain the wing grid is formed at selected spanwise stations in a similar manner as in two-dimensional airfoil problems. This scheme is sketched in Figure 6. As noted in Section 2.1, this grid-generation scheme can cause difficulties in the representation of the body due to the appearance of "fins" (see for example, Figure 2). The present idea is to use the overlapping grid scheme to avoid this difficulty.

Considering the transformed "wing-on-a-wall" problem, the procedure is initially the same as in the previous section. First, the usual parabolic transformation used by Jameson is made for the wing sections. In this scheme the physical coordinates x, y, z are transformed to the parabolic coordinates ξ, η, ζ as follows

$$\xi + i\eta' = \left[(x - x_0) + i(y - y_0) \right]^{1/2} \quad (15)$$

$$\zeta = z$$

where x_0, y_0 is the location of a singular line inside the wing. This transformation unwraps the wing about the singular line giving a slowly undulating surface $S(\xi)$ representing the wing. A shearing transformation

$$\eta = \eta' - S(\xi) \quad (16)$$

then makes the wing a surface of constant η . Thus the final transformation is

$$\left. \begin{aligned} x &= x_0 + \xi^2 - (\eta + S)^2 \\ y &= y_0 + 2\xi(\eta + S) \\ z &= \zeta \end{aligned} \right\} \quad (17)$$

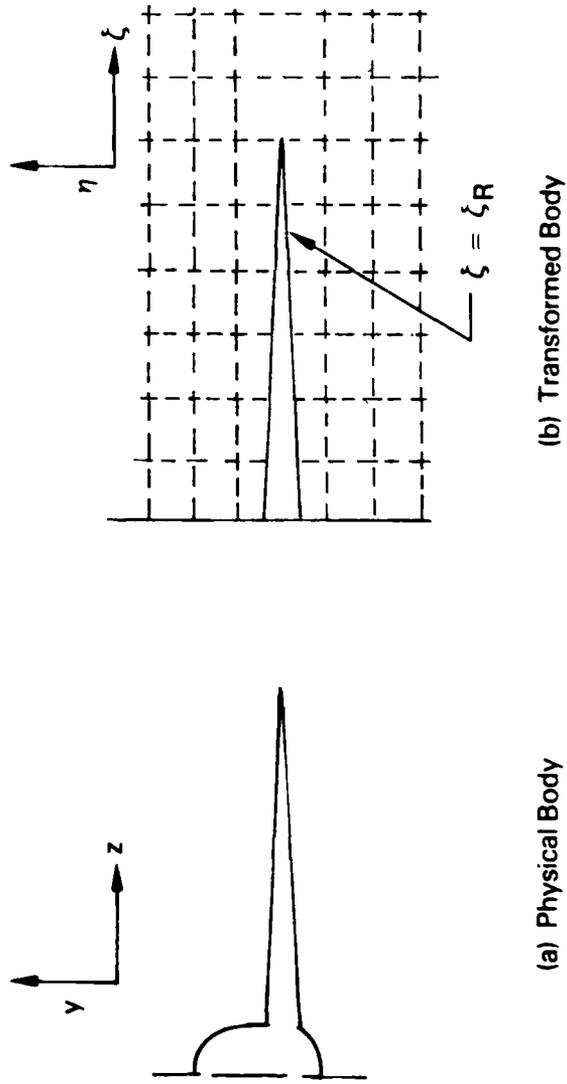


Figure 6. Jameson Mesh System for a Wing-Body Combination.

Now let the preliminary master coordinate ψ_1 be given as

$$\left. \begin{aligned} \psi_1 &= \xi \\ \psi_2 &= \eta \\ \psi_3 &= \zeta \end{aligned} \right\} \quad (18)$$

where ξ, η, ζ are defined for the transformed wing by Equations (15) and (16). Retaining the ξ coordinate for the body (in this case the transformed "wall"), it can be seen that a $\xi = \text{constant}$ line will cut the body profile at some location "a", as shown in Figure 7. Now this line, AB in Figure 7, is the limiting form of a family of ellipses, thus the body grid is constructed by using a family of ellipses in a η, ζ plane as one coordinate surface. A coordinate surface orthogonal to this is a family of hyperbolas. Thus, the body grid is then a system of elliptic cylindrical coordinates and, in the notation of Section 3.1,

$$\left. \begin{aligned} \psi_1 &= \phi_2 \\ \psi_2 &= a \cosh \phi_1 \cos \phi_3 \\ \psi_3 &= a \sinh \phi_1 \sin \phi_3 \end{aligned} \right\} \begin{aligned} 0 &\leq \phi_1 \leq \phi_{1L} \\ 0 &\leq \phi_3 \leq \frac{\pi}{2} \end{aligned} \quad (19)$$

where a is the intercept made by a $\xi = \text{constant}$ line on the transformed body profile. The outer limit of ϕ_1, ϕ_{1L} , is chosen such that the point defined by $(\phi_{1L}, \phi_2, 0)$ coincides with a specified $\zeta = \zeta_L$ line in the wing grid. Thus,

$$\phi_{1L} = \ln \left\{ \left[\left(\frac{a}{\zeta_L} \right)^2 + 1 \right]^{\frac{1}{2}} + \frac{a}{\zeta_L} \right\}. \quad (20)$$

Equation (19) then gives the body grid in the transformed domain. In order to match the wing and body grids, it is simpler to replace the ψ_3 grid lines by a continuation of the family of ellipses in Equation (19).

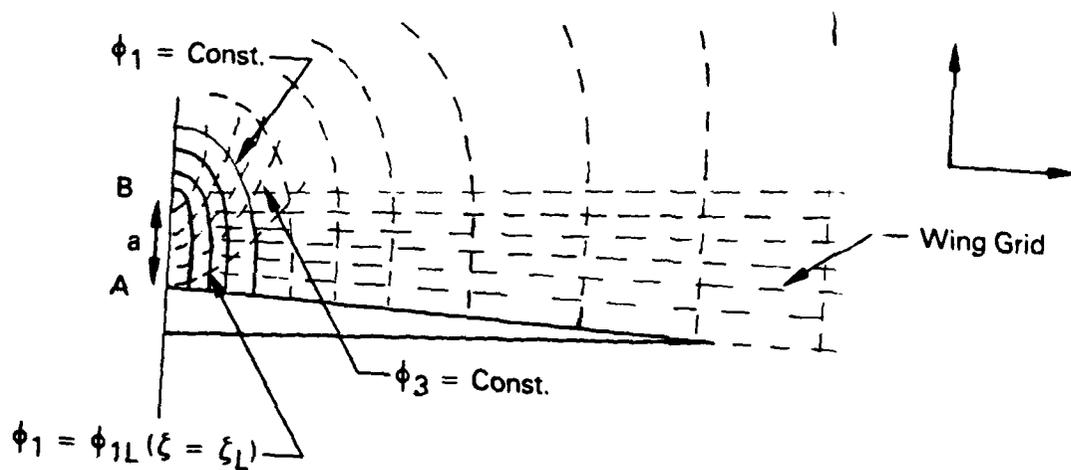


Figure 7. Overlapping Grid System for the Transformed Body.

A-24

Thus the wing grid is now

$$\begin{aligned}\psi_1 &= \xi \\ \psi_2 &= \eta \\ \psi_3 &= a \sinh u \sin v .\end{aligned}\tag{21}$$

It should be noted that the outer parts of the wing grid ellipses are the far-field representation, where

$$u = \frac{1}{\zeta} \ln \left\{ \left[\left(\frac{a}{\xi} \right)^2 + 1 \right]^{1/2} + \frac{a}{\zeta} \right\}\tag{22}$$

and v is given by

$$\left(\frac{\eta}{a} \right)^2 \sec^2 v - \left(\frac{\zeta}{a} \right)^2 \operatorname{cosec}^2 v = 1 .\tag{23}$$

Having obtained the overlapping wing-body grid in the transformed domain, the physical grid is obtained by reversal of the Joukowski and shearing transformation. In summary then

- a. Use the existing Joukowski and shearing transformations to reduce the wing-body problem to that of a wing on a wall.
- b. Construct the usual wing parabolic coordinates ξ, η, ζ .
- c. Determine the intercept, "a", of a $\xi = \text{constant}$ line with the body profile.
- d. Construct the transformed body coordinate system, Equation (19).
- e. Construct the modified wing coordinate system, Equation (21).
- f. Use the reversal of the shearing and Joukowski transformations to construct the mesh points in the physical plane.

Note that since both the ψ_1 and ϕ_2 surfaces and the ψ_3 and ϕ_3 surfaces are coincident in the overlap region, interpolation of the ψ_2 or ϕ_1 coordinates for the overlap region is simple to implement if this should be necessary.

The above mesh-generation scheme has been coded for a cylindrical body with a nose consisting of an elliptical body of revolution. Sections of both the body grid (along a $\xi = \text{constant}$ line) and the wing grid are shown in Figures 8 and 9. It can be seen that the body grid has a highly skewed mesh cell adjoining the plane of symmetry. A probable cause of this error is a computer "bug" at the plane of symmetry itself where certain functions in the Joukowski transformation can have branch points. It is suggested that in the generation of the body grid this singular behavior is not correctly represented.

3.3 Example

The transonic flow around a semi-infinite body with a mid-mounted wing was computed on the overlap mesh by using the finite-volume method. The body is closed at the front and extends an infinite distance downstream. The wing is the ONERA M-6 wing at 0° angle of attack. The freestream Mach number is 0.85. A sketch of the configuration is shown in Figure 10. The basic code is FLO 28 with a modified mesh system. Separate iterations for both body and wing are programmed with the overlap concept being used to generate the necessary boundary conditions. A one-dimensional interpolation (in the y-direction) is used in the overlap region. It was found necessary to replace the iteration subroutine YSWEEP with XSWEEP in the code because for a swept wing the marching direction could be in the upstream direction with YSWEEP.

The pressure distributions at specified spanwise stations computed using the present mesh are shown in Figure 11. Also shown are results of Jameson's FLO 28 finite-volume code with a different mesh structure. It can be seen that the present grid gives pressures that are in qualitative agreement with those of FLO 28 but which are in general much too low. The probable cause of this error is that the representation of the "far field" in the present grid is much too close to the wing surface, and modification to alter the location of the outer grid point should be incorporated into the mesh. A further point that should be considered is the accuracy in the overlap region itself. The present indications are that the accuracy is worse in this region than in other parts of the

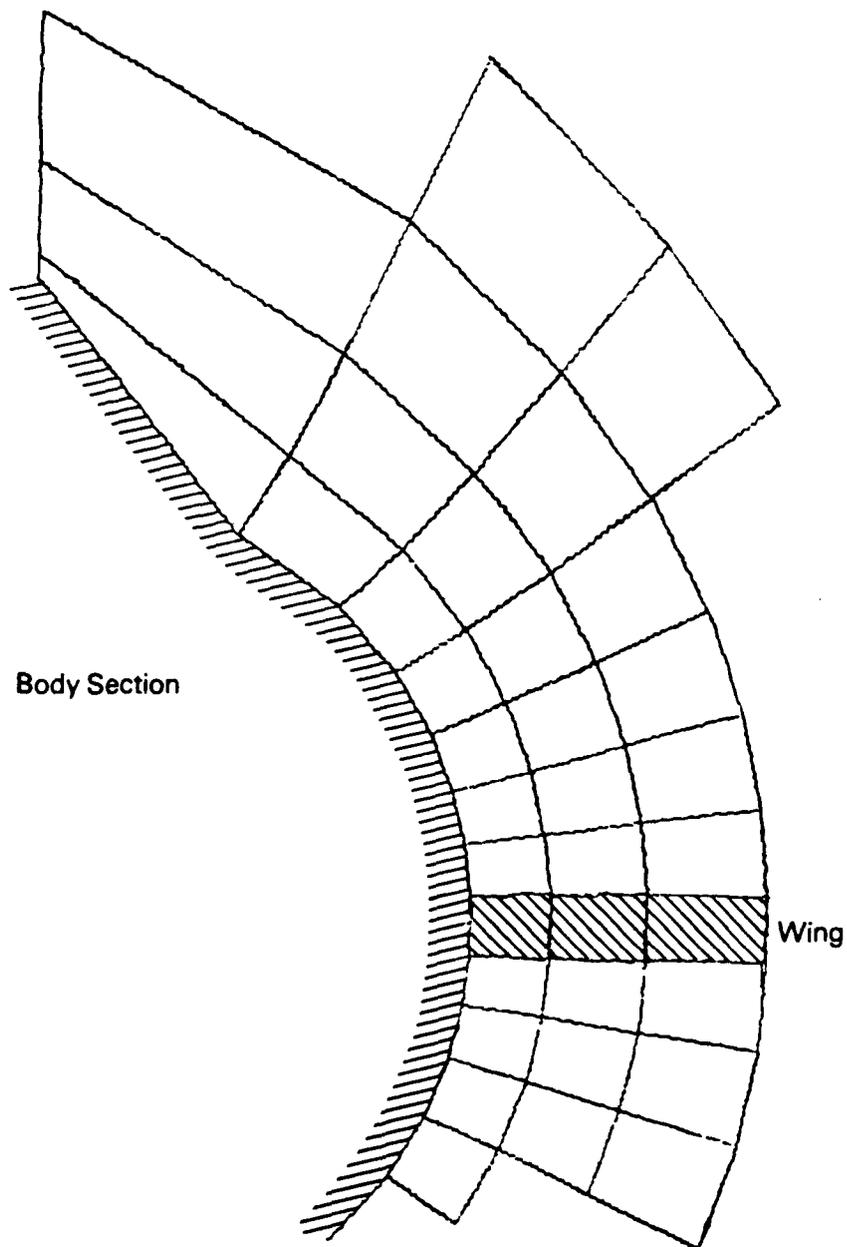


Figure 8(a). Body Grid Along a $\xi = \xi_{23}$ Line (32 x 8 x 4 Grid).

A-27

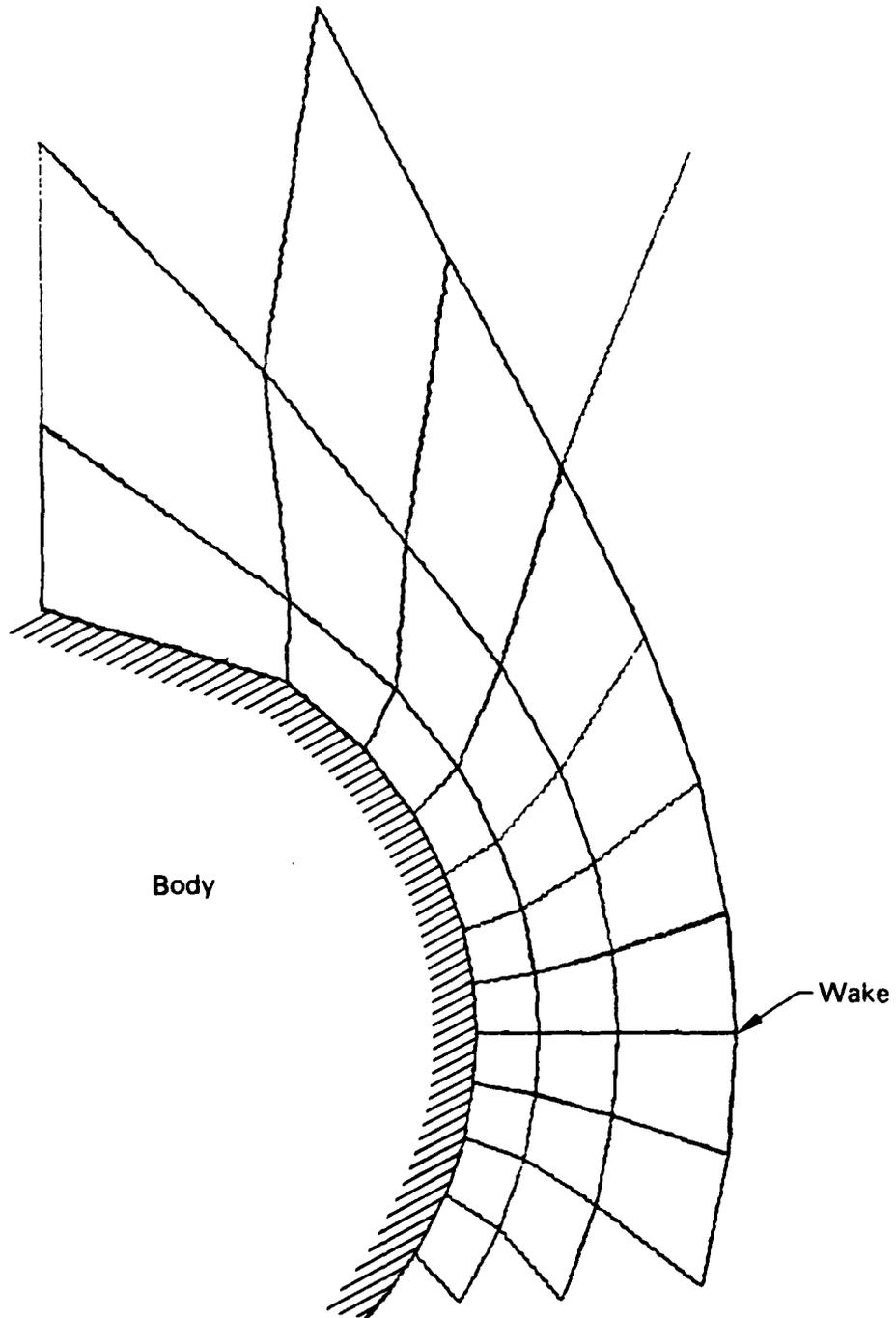


Figure 8(b). Body Grid Along a $\xi = \xi_{30}$ Line ($32 \times 8 \times 4$ Grid).

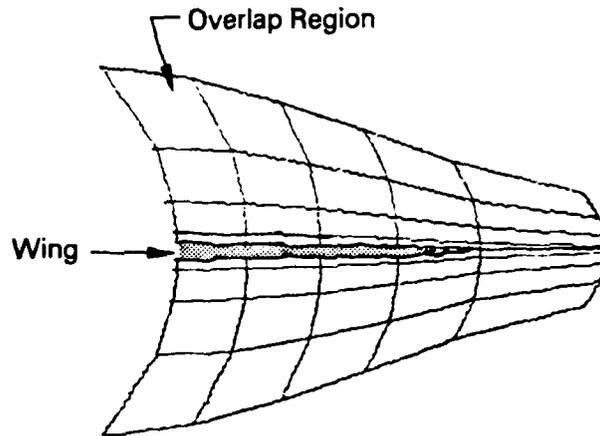


Figure 9(a). Wing Grid Along a $\xi = \xi_{23}$ Line (32 \times 8 \times 4 Grid).

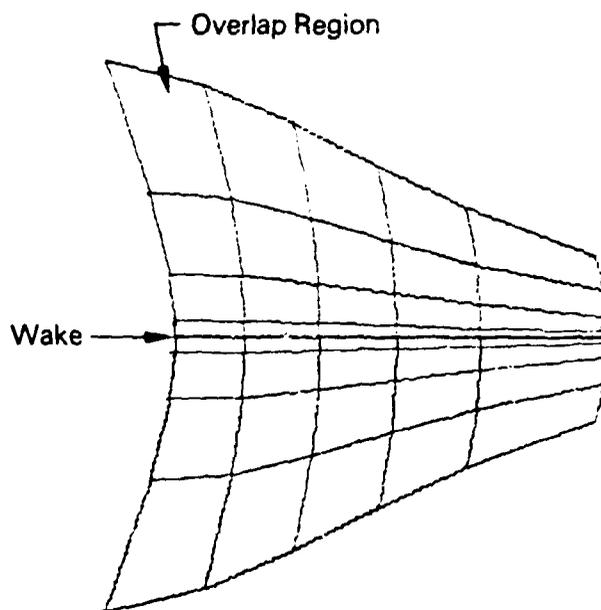


Figure 9(b). Wing Grid Along a $\xi = \xi_{30}$ Line (32 \times 8 \times 4 Grid).



Figure 10. Wing-Body Configuration.

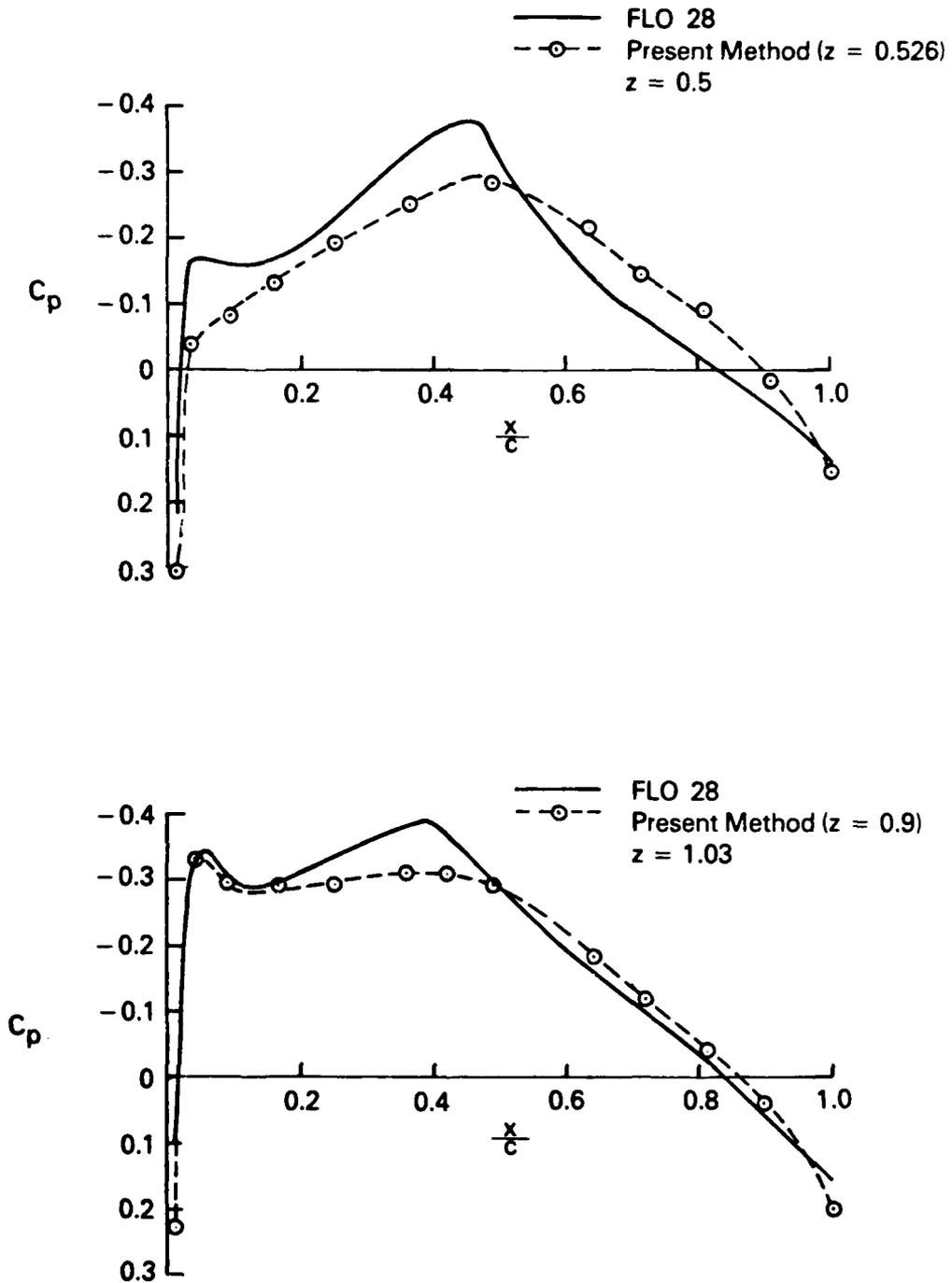


Figure 11. Pressure Distribution Around a Wing-Body Combination $M_\infty = 0.85$.

AD-A090 829

FLOW RESEARCH CO KENT WA

F/G 20/4

DEVELOPMENT OF FINITE-VOLUME METHODS FOR THREE-DIMENSIONAL TRAN--ETC(U)

AUG 80 J E MERCER, W JOU, D A CAUGHEY

N00014-78-C-0079

UNCLASSIFIED

FLOW-RR-166

NL

Q18



END
DATE
FILMED
42-90
DTIC

flow field. This may be due to the interpolation procedure used in this region or simply another manifestation of the "far-field representation" error. Finally, the option of iterating the body and wing calculations at different rates should be investigated since this could reduce the computation time.

4. Grid-Generation Procedure for a Tail Plane

4.1 Type and Restrictions of Tail-Plane Grid

The basic idea is to use the overlapping grid technique developed in Section 3 for a wing-body combination. Basically, this technique develops a separate grid for each component of an airplane, e.g., the wing and body and tail plane. One grid, probably that for the wing, is designated a master grid, and the others are slave grids. Each grid overlaps the grid of a neighboring component by at least one grid cell which allows a (usually spanwise) marching procedure for each component. The boundary, the extreme limit of a component grid system, is either a far-field grid line or a common line with another grid. Dirichlet boundary conditions can then be applied along this boundary.

The existing wing-body grid (see Section 3.2) consists of the following steps:

- a. Conformally transforming and shearing the body to a slit as in the existing Jameson-Caughey finite-volume code.
- b. Generating a "shell" type body grid enclosing the body which is assumed to be semi-infinite. These shells will probably extend about 1/3 of the wing semispan.
- c. Constructing the wing grid using a continuation of the shell-ordinates but changing the "normal" coordinate, the ψ_2 coordinate in Equation (21), to avoid coarse mesh cells near the wing tip.

For a tail-plane grid, it is assumed that the tail will not extend sufficiently in a lateral direction as to lie outside the body grid (i.e., less than one-third semispan). Consequently, the tail-plane grid will be embedded in the body grid alone and will not intersect the wing grid. This is the main restriction on the tail-plane dimensions. A second restriction is that the tail-plane lies at least one grid line from the wing wake in order to avoid double valuedness on the ξ -coordinate lines.

4.2 General Formulation

First let the body be compressed to a slit as in Section 3.2, with the corresponding changes to the tail-plane input stations X_T ,

Y_T, Z_T to the transformed coordinates $\bar{X}_T, \bar{Y}_T, \bar{Z}_T$. Then find where these points $(\bar{X}_T, \bar{Y}_T, \bar{Z}_T)$ occur in the body grid as follows.

The body grid is given by a series of ellipses and hyperbolas together with the original wing grid value of ξ , as shown in Figure 7. The body grid lines are given by

$$\mu = a \cosh u \cos v \quad (24)$$

$$\zeta = a \sinh u \sin v$$

where ξ, η, ζ are the coordinates found from the original wing grid.

Hence, the transformed tail-plane input stations $\bar{X}_T, \bar{Y}_T, \bar{Z}_T$ are transformed to the original wing coordinates ξ_T, η_T, ζ_T .

From Equation (24) we have the equivalent stations in ξ, U, V coordinates where ξ_T is as before, V_T is found from

$$\left(\frac{\mu_T}{a \cos V_T} \right)^2 - \left(\frac{\zeta_T}{a \sin V_T} \right)^2 = 1, \quad (25)$$

and U_T is given by

$$U_T = \cosh^{-1} \left(\frac{\mu_T}{a \cos V_T} \right). \quad (26)$$

Having obtained the input stations of the tail plane in the body grid, bounding lines are then constructed as follows.

A search is made to find the body grid surfaces $\bar{\xi}_N, \bar{U}_N, \bar{V}_N$ that always completely enclose the tail plane. A second set of grid surfaces is then found, probably obtained by simply changing the grid index by one such that they move further from the tail plane. For each $U = \text{constant}$ surface for $U \leq \bar{U}_N$, the tail plane (and wake) will look as shown in Figure 12. Hence, if the tail-plane geometry is known for this section, the grid generation reduces to a two-dimensional problem. The tail-plane geometry at the required computational ξ, U stations can be obtained from the input values ξ_T, U_T, V_T given by Equations (25) and (26) by using interpolation. Thus, we now know the ξ_T, V_T data for the

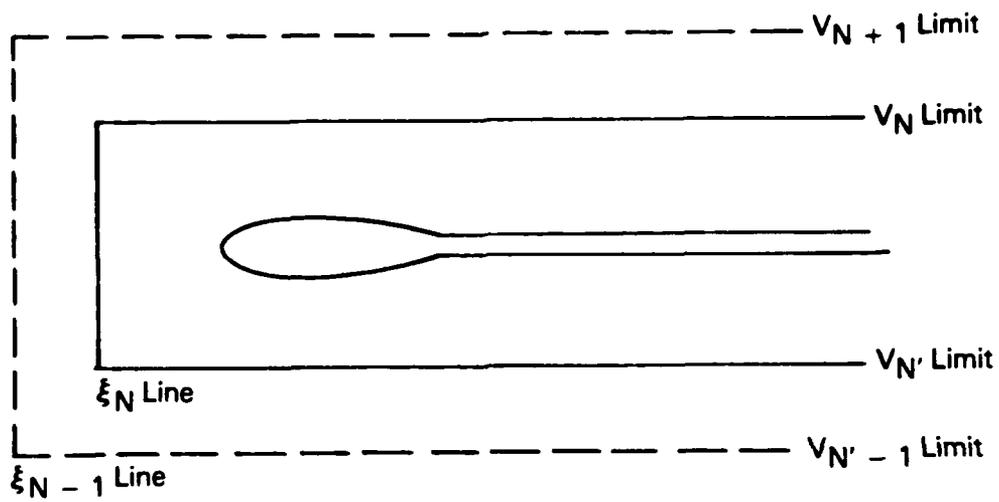


Figure 12. Tail-Plane Embedding in Transformed Domain.

computational section shown in Figure 12. The problem is now to develop a coordinate system for this configuration.

Let the coordinate lines be denoted as follows. The outer boundary grid lines are denoted by

$$g(\xi, V) = 0 \quad . \quad (27a)$$

The inner bounding grid lines are denoted by

$$g_2(\xi, V) = 0 \quad , \quad (27b)$$

and the tail-plane section and wake are denoted by

$$f(\xi, V) = 0 \quad . \quad (27c)$$

We want a system of grid lines that coincide with the body $f(\xi, V) = 0$ and also, on the outer bound, essentially coincide with the curve $g_1(\xi, V) = 0$ (the corners are "rounded" to avoid singularities). This arrangement is sketched in Figure 13.

Consider the set of parameter curves

$$\psi(\xi, V) = f(\xi, V) (1-\lambda) + \lambda g_1(\xi, V) = 0 \quad , \quad (28)$$

where

$$0 \leq \lambda \leq 1 \quad .$$

When

$$\lambda = 0 \quad , \quad \psi \equiv f(\xi, V)$$

$$\lambda = 1 \quad , \quad \psi \equiv g_1(\xi, V) \quad .$$

Hence, Equation (28) is one family of coordinate lines, for each λ , if the lines do not intersect. Two members of the family (λ_1, λ_2) will

A-38

intersect at (ξ_1, V_1) if $\psi(\xi_1, V_1) = \lambda_1 = \lambda_2$. This is obviously impossible unless $\lambda_1 = \lambda_2$, a trivial condition. Hence, we now have one of our two coordinate lines:

$$\frac{f(\xi_1; V)}{f(\xi_1, V) - g_1(\xi_1, V)} = \lambda \quad (29)$$

For the second coordinate line there is probably no easier method than to use the parabolic mapping of the type used for the wing. Thus for the tail plane we have

$$\hat{\xi} + i \hat{\eta}' = \left[(\xi - \xi_s) + i (V - V_s) \right]^{1/2} \quad (30)$$

where ξ_s, V_s is the location of a singular line just inside the tail plane. Thus

$$\begin{aligned} \hat{\xi}^2 - (\hat{\eta}')^2 &= \xi - \xi_s \\ 2\hat{\xi}\hat{\eta} &= V - V_s \end{aligned} \quad (31)$$

As before, shear the $\hat{\eta}'$ coordinates to remove the "slowly undulating curve" referred to in Section 3.2. Thus

$$\hat{\eta} = \hat{\eta}' - V_T(\hat{\xi}) \quad (32)$$

Combination of Equations (31) and (32) gives

$$\hat{\xi}^2 - \frac{(V - V_s)^2}{\hat{\eta}^2} = \xi - \xi_s \quad (33)$$

Note that the $f(\xi, V) = 0$ line in Equation (28) is given by

$$\hat{\eta}(\xi, V) = 0 \quad (34)$$

Now the bounding line $g_1(\xi, V) = 0$ consists solely of lines of $\xi = \text{constant}$ $\hat{\eta}$ and $V = \text{constant}$. Hence, the point of intersection of any $\hat{\xi}$ line with the outer boundary, including the location of the

corner points, is simple. An important difficulty arises at the corner points in the tail-plane mesh since in the two-dimensional problem a mesh cell containing these corner points may be a five-sided cell rather than the required four-sided cell. This difficulty can be removed by first ensuring that $\xi_1 \neq \xi_c$ and then, as far as the tail-plane mesh is concerned, replacing the "corner" by a smooth curve joining the $\xi = \text{constant}$ and $V = \text{constant}$ lines. For the overlapping points at the corners, the body grid extrapolation can easily be used. Denote a typical point as (ξ_1, V_1) . The tail-plane coordinate lines are now

$$\begin{aligned}\hat{\xi} &= \text{constant} \\ \lambda &= \lambda(\hat{\xi}_1)\end{aligned}$$

The next task is the construction of the intermediate coordinate lines $\lambda_j(\hat{\xi}_1)$. Ideally we would like the $\lambda_j(\hat{\xi}_1)$ lines to essentially coincide with the $g_2(\xi, V) = 0$ bounding line (again the corners are "rounded" to avoid singularity). For a constant $\lambda_j(\xi_1)$ (i.e., independent of $\hat{\xi}_1$) this would probably be impractical. Hence choose $\lambda_j = \lambda_j(\hat{\xi}_1)$ such that

$$f(\xi, V) \left[1 - \lambda_j(\hat{\xi}_1) \right] + \lambda_j(\hat{\xi}_1) g_1(\xi, V) - g_2(\xi, V) = 0 \quad (35)$$

Again, since the curve $g_2(\xi, V) = 0$ is either a line of constant ξ or a line of constant V , the point of intersection of a $\hat{\xi}_1 = \text{constant}$ line with g_2 is easily found from Equation (33), and the required value of $\lambda_j(\hat{\xi}_1)$ is found from Equation (35). It is proposed that $\lambda_j(\hat{\xi}_1)$ be monotonic functions of j and that λ_j be one of this set, possibly the penultimate value (the extremes are $\lambda = 0, 1$). When the curve $g_2(\xi, V) = 0$ changes from a $\xi = \text{constant}$ line to a $V = \text{constant}$ line, it is possible that the curves for a given λ_j are not continuous. $\lambda_j(\hat{\xi}_1)$ must therefore be chosen such that each $\lambda_j(\hat{\xi}_1)$ curve is continuous through the corner junction. Thus we now have a grid system $(\hat{\xi}_1, \lambda_j(\hat{\xi}_1))$ that coincides with the tail plane, the outer boundary line $g_1(\xi, V) = 0$, and the overlap boundary $g_2(\xi, V) = 0$. The actual ξ_1 intersection point on these second two lines will not in general coincide with the body grid point, and interpolation will have to be used.

A-40

Having chosen both the $\hat{\xi}_i, \lambda_j(\hat{\xi}_i)$ and the U_k lines, the reverse set of transformations are as follows.

The $\hat{\xi}_i$ are known, but for the inverse transformation of Equation (31) it is necessary to know $\hat{\eta}(\xi, V)$. By definition,

$$\hat{\eta}(\xi, V) = R^{1/2} \sin \theta/2, \quad (36)$$

where

$$R = \left[(V - V_s)^2 + (\xi - \xi_s)^2 \right]^{1/2} \quad (37)$$

$$\theta = \tan^{-1} \left(\frac{V - V_s}{\xi - \xi_s} \right)$$

and ξ_s, V_s are the coordinates of the singular point of the tail-plane section. Using Equation (35), it follows for $\hat{\xi}_i < \hat{\xi}_c$

$$\left[\xi - \xi_T(\hat{\xi}_i) \right] \left[1 - \lambda_j(\hat{\xi}_i) \right] + \lambda_j(\hat{\xi}_i) (\xi - \xi_i) = 0 \quad (38a)$$

where $\xi_T(\hat{\xi}_i)$ are the ξ coordinates corresponding to the $\hat{\xi}_i$ on the section surface and $\xi = \xi_1$ is the bounding line of $g_1(\xi, V) = 0$ for $\hat{\xi}_i$ less than the corner value $\hat{\xi}_c$.

For $\hat{\xi}_i > \hat{\xi}_c$

$$\left[V - V_T(\hat{\xi}_i) \right] \left[1 - \lambda_j(\hat{\xi}_i) \right] + \lambda_j(\hat{\xi}_i) (V - V_1) = 0 \quad (38b)$$

where $V_T(\hat{\xi}_i)$ is the V-coordinate corresponding to the $\hat{\xi}_i$ on the section surface, and $V = V_1$ is the bounding line of $g_1(\xi, V) = 0$ for $\hat{\xi}_i > \hat{\xi}_c$.

Using Equation (36) and Equation (38) with Equation (31) and Equation (32), the coordinates ξ_{ij}, V_{ij} corresponding to the points $\xi_i, \lambda_j(\hat{\xi}_i)$ can be obtained. Having obtained ξ_{ij}, V_{ij} (and U) the inverse transformation for the wing-body mesh can then be used to recover the physical grid points.

The suggested iteration sequence is as follows.

A-41

- (1) Sweeping down each of the shells ($U = \text{constant}$ lines), sweep to end boundary*, noting the values of ϕ on the g_1 grid lines.
- (2) Using interpolation, get the Dirichlet boundary conditions for the tail plane.
- (3) Compute the tail-plane flow, noting the values on the g_2 grid lines.
- (4) Sweep out along the span as before. No extra work is involved at the tail-plane tip since it should be identical to the wing-body procedure.
- (5) Replace ϕ on the body mesh lines, g_2 by their values from step (3) using interpolation.
- (6) Repeat.

An example of an embedded tail-plane mesh is shown in Figure 14. The section shown is the mesh in a $\zeta = \text{constant}$ surface of a typical wing grid. No attempt was made at clustering the tail-plane grid lines, although this is easy to do by simply choosing the $\lambda_j(\xi_1)$ in Equation (35).

*Not strictly necessary but avoids complex logic.

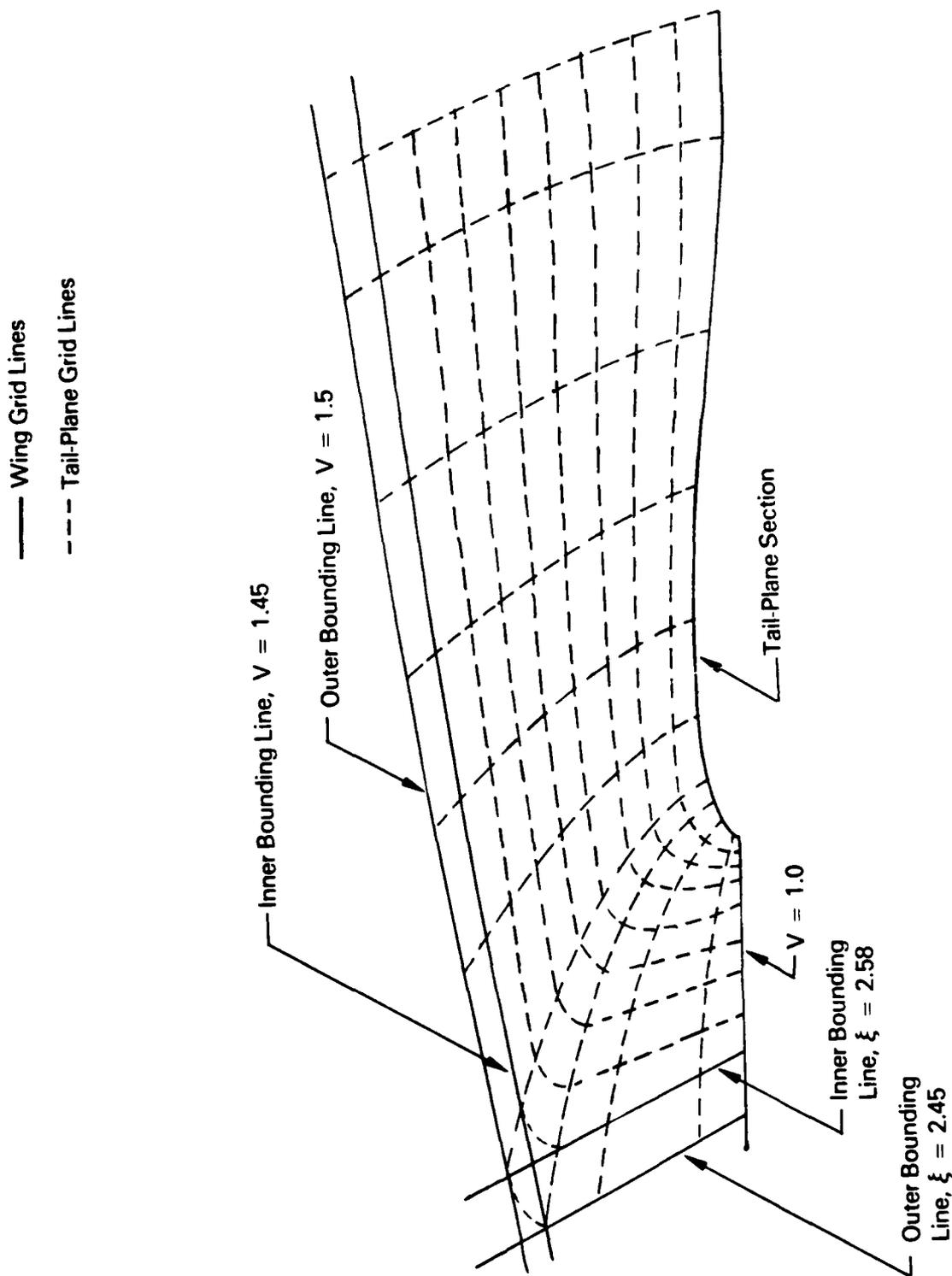


Figure 14. Embedded Tail-Plane Mesh System.

5. Conclusions

An attempt has been made to evaluate existing mesh-generation schemes for their ability to model complex airplane geometries. It is felt that all existing generation techniques have significant drawbacks in this regard, mainly because of the different topological characteristics of the airplane components, such as the fuselage, wing, and nacelles. A new method of overlapping meshes has been formulated to overcome this problem. In this system, a near optimum mesh system for each component is overlapped with adjacent component meshes allowing a satisfactory iteration procedure with little or no interpolation. A detailed mesh-generation scheme for a semi-infinite body closed at the front with a wing mounted at any location has been derived. It is felt that the present overlapping mesh approach shows considerable promise for complex airplane configurations since additional component meshes can be eventually "plugged in" to existing or master meshes. However, the example presented does have deficiencies which should be corrected before any further extension is made. It is suggested that the existing problems are due to a computer coding error and to a lack of flexibility in locating the far-field point. Neither of these problems should be insurmountable.

A means of embedding a mesh system for a horizontal tail into a wing or body mesh has also been developed. A simple example of such a mesh has been computed and appears satisfactory. No flow computations have been performed.

Authors Note

This report represents the state of the work when the author (David Nixon) left Flow Research Company. The report should not be considered as an account of a completed research program.

References

- Bailey, F. R., and Ballhaus, W. F. (1972) "Relaxation Methods for Transonic Flow about Wing-Cylinder Combinations and Lifting Swept Wings", Lecture Notes in Physics, Vol. 19, Springer-Verlag.
- Ballhaus, W. F., Bailey, F. R., and Frick, J. (1976) "Improved Computational Treatment of Transonic Flow about Swept Wings," Advances in Engineering Science, Vol. 4, NASA CP-2001.
- Caughey, D. A., and Jameson, A. (1977) "Numerical Calculation of Transonic Potential Flow About Wing-Fuselage Combinations," AIAA Paper 77-677.
- Jameson, A., and Caughey, D. A. (1977) "A Finite Volume Method for Transonic Potential Flow Calculations," Proceedings of the AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque, New Mexico.
- Mastin, C. W. J., and Thompson, J. F. (1978) "Three-Dimensional Body Fitted Coordinate Systems for Numerical Solution of the Navier-Stokes Equations," AIAA Paper 78-1147.
- McConnell, A. J. (1957) Applications of Tensor Analysis, Dover.
- Sells, C. C. L. (1968) "Plane Subcritical Flow Past a Lifting Aerofoil," Proc. Roy. Soc. A., 308, pp. 377-401.
- Sorenson, R. L., and Steger, J. L. (1977) "Simplified Clustering of Non-Orthogonal Grids Generated by Elliptic Partial Differential Equations," NASA Technical Memo 73252.
- Steger, J. L., and Sorenson, R. L. (1978) "Automatic Boundary Clustering in Grid Generation with Elliptic Partial Differential Equations." Under review, to be published.
- Thompson, J. F., Thames, F. C. J., and Mastin, C. W. (1974) "Automatic Numerical General of Body Fitted Curvilinear Coordinate System for Fields Containing any Number of Arbitrary Two Dimensional Bodies," J. Comp. Phys., Vol. 15, No. 3.

