AIAA-80-0156

A Computer Code to Model Swept Wings in an Adaptive Wall Transonic Wind Tunnel

J. E. Mercer, E. W. Geller, M. L. Johnson, Flow Research Co., Kent, Wash., A. Jameson, Courant Institute of Mathematical Sciences, New York, N.Y.

AIAA 18th

AEROSPACE SCIENCES MEETING

January 14-16, 1980/Pasadena, California

A COMPUTER CODE TO MODEL SWEPT WINGS IN AN ADAPTIVE WALL TRANSONIC WIND TUNNEL

J. E. Mercer, E. W. Geller, Research Scientists,
and M. L. Johnson, Research Engineer
Flow Research Company
Kent, Washington

and A. Jameson, Professor of Computer Science
Courant Institute of Mathematical Sciences
New York, New York

Abstract

A computer program has been developed to calculate inviscid transonic flow over a swept wing in a wind tunnel with specified normal flow at the walls. An approximately orthogonal computational grid which conforms to the wing and the tunnel walls was developed for application of the Jameson-Caughey finite volume algorithm. The code solves the full potential equations in fully conservative form using line relaxation. This program is to be used in place of the wind tunnel for preliminary studies of the adaptive wall concept for three dimensional configurations. It can also be used to assess the magnitude of wall interference in a conventional tunnel.

Introduction

It is possible to simulate free air flow about a wind tunnel model by controlling inflow and outflow through porous or slotted tunnel walls. The correct normal flow distribution changes with model configuration, angle of attack and Mach number and is not known apriori. It has been proposed by Ferri and Baronti[1] and by Sears[2] to match the flow field provided by the tunnel to a computed flow field beyond the tunnel walls by an iterative adjustment of the boundary conditions at the interface of these two regions. This wedding of the wind tunnel and the computer eliminates the major problems associated with using either of these tools by itself, namely the incorrect flow constraints of the tunnel walls, and the inaccuracy and/or the expense of near field flow calculation, particularly for transonic flow and separated flow.

A typical iterative procedure for matching the inner flow solution supplied by the wind tunnel to the outer flow solution supplied by computer calculation is illustrated in Figure 1. The pressure is measured at or near the tunnel walls with the current wall normal flow velocity distribution. This pressure distribution is used for the boundary condition for the outer flow solution which provides an updated wall normal flow to close the iteration loop. Other iterative procedures are possible. The iteration could be made in the direction opposite to that shown in Figure 1. One alternative has been proposed using a laser velocimeter to measure the normal velocity on two surfaces. One surface is used to provide a boundary condition for the outer flow calculation. Another surface, farther removed from the model, is used for the matching of the tunnel flow with the outer calculation.

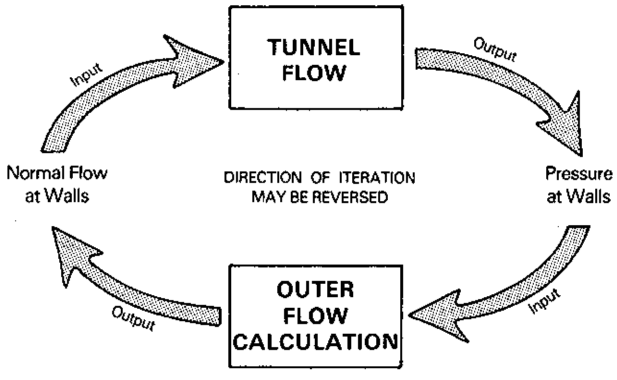


Figure 1. Iterative Matching of the Inner and Outer Flows.

The Arnold Engineering Development Center is sponsoring a preliminary investigation of the adaptive wall wind tunnel concept to answer questions about plenum compartmentalization required for wall normal flow distribution, sensitivity to pressure measurements, iteration variables, convergence, and other considerations. In order to make this investigation without actually building an adaptive wind tunnel, the tunnel flow shown in Figure 1 will be replaced by a computer calculation of that flow, and it is this part of the project that is the subject of this paper. Although it is not possible to model all the phenomena involved with transonic flow over a wing, the code developed will handle the dominate inviscid effects for large flow perturbations. In addition, the code developed can be used to assess the severity of wall interference in a conventional tunnel. Such calculations could be used as a basis for correcting wind tunnel data. This technique would be very useful in the transonic regime where conventional wall correction methods fail.

Wind Tunnel Code

The code developed for the tunnel flow solution is based on the finite volume method of Jameson and Caughey.[3] The code solves the full potential equation with non-linear boundary conditions using fully conservative differencing. One great advantage of this method is that the grid generation scheme can be treated independently from the numerical algorithm. This allows considerable freedom in selecting the grid. Once the grid is generated, the equation solver deals directly with the corner points of the mesh. No analytical mapping of the differential equation is necessary.

Governing Equations and Boundary Conditions

We assume that vorticity generated by shock waves can be ignored so that the velocity (u,v,w) is the gradient of the potential ϕ

$$u = \phi_x \ , \quad v = \phi_y \ , \quad w = \phi_z \qquad (1)$$

Conservation of mass requires

$$\frac{\partial}{\partial u}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0 \qquad (2)$$

where the density is given by

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

$$q^2 = u^2 + v^2 + w^2 \qquad (4)$$

Discontinuities in velocity across surfaces are allowed provided the following shock jump conditions apply:

(a) Continuity of ϕ , implying continuity of the tangential velocity component.

(b) Continuity of ρu_n , where u_n is the component of velocity normal to the discontinuity surface.

(c) Increase of density across the discontinuity surface in the direction of flow.

Two types of tunnel wall boundary conditions are considered:

Case A Normal Velocity Prescribed on Tunnel Wall

The Neumann type of boundary condition on ϕ at the tunnel wall is equivalent to specification of the normal velocity according to

$$v_n = \frac{\partial \phi}{\partial n} \qquad (5)$$

Case B Pressure Prescribed on the Tunnel Wall

The Dirichlet type of boundary condition on ϕ at the tunnel wall is equivalent to specification of the pressure coefficient, C_p at the wall according to the small perturbation relation

$$C_p = -2(u - U_\infty) \qquad (6)$$

and

$$\phi = \int_{\substack{\text{along}\\ \text{wall}}} u \, dx \qquad (7)$$

The exact relation between the pressure and velocity is more complex than Equation (6). However, for this application, the small perturbation assumption should be sufficient at the wall location, and it provides a convenient simplification. Equations (6) and (7) assume the choice for Cartesian axes given in Figure 2.

The boundary conditions used for the upstream and downstream boundaries of the calculation region are

$$\text{Upstream:} \quad \frac{\partial \phi}{\partial x} = u = U_\infty \qquad (8)$$

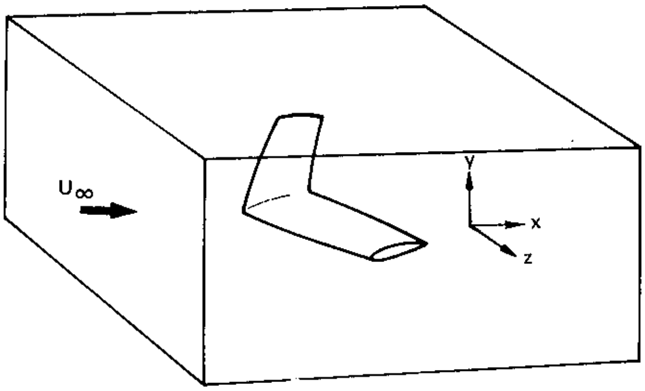


Figure 2. Cartesian Coordinates Chosen.

$$\text{Downstream:} \quad \frac{\partial \phi}{\partial x} = u = U_\infty + \delta u \qquad (9)$$

$$\delta u = \frac{1}{\rho_d A_d} \int_{\substack{\text{tunnel}\\ \text{walls}}} \rho v_n \, dS \qquad (10)$$

ρ_d = density at downstream boundary

A_d = tunnel cross-sectional area at downstream boundary

v_n = normal velocity at wall (positive into tunnel)

At the downstream boundary, the increment δu over freestream, velocity U_∞ assures conservation of mass.

Mesh Generation

One of the principal advantages of the finite volume method of Jameson and Caughey,[3] which was applied to the wind tunnel flow problem at hand, is that transformation to boundary conforming coordinates is defined by a table of grid points, rather than by mapping functions. Thus, many and complex sequential mapping functions may be used to generate these grid points, but once generated these functions are discarded, and three-dimensional linear interpolation in the transformed space, is used when coordinate values are needed at other than grid points. Similarly, scalar fields needed for the numerical calculation (e.g., the velocity potential) are defined by their values at the grid points and tri-linear variation in the transformed space is assumed between grid points.

The mapping from x, y, z physical space to the X, Y, Z computational space that was developed for the flow about a swept wing in a wind tunnel will now be defined. Although the description involves a number of sequential mappings, it should be remembered that the final product needed for the numerical solution is a table of grid points. For the physical space, Cartesian axes are chosen (as shown in Figure 2) such that the x-axis is in the direction of the undisturbed flow and the z-axis is in the spanwise direction of the wing.

The mapping in z is uniform in the region between the centerline and the tip and is a stretching in the region beyond the tip as illustrated in Figure 3. For purposes of grid generation, the wing planform is extended to the tunnel wall as shown in Figure 3.

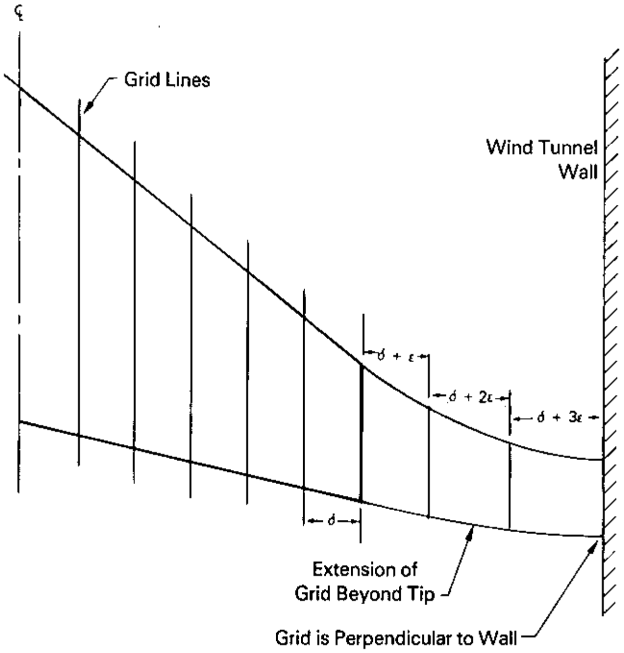
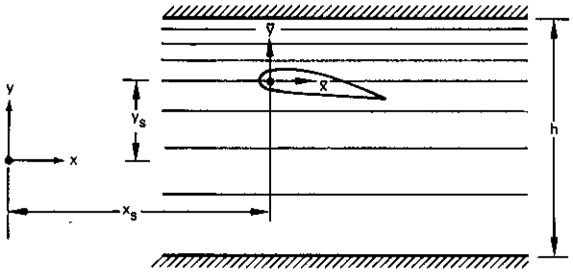


Figure 3. Z Grid Planes and Extension of Planform for Grid Generation.



- $x = (x - x_s) \times 2\pi/h$

  Removes Wing Sweep and Scales x According to the Mean y Dimensional Scale

- $y' = (y - y_s)$

  Places Origin of Coordinate System at Reference Point on Model

- $y = ay' + by'^2$

  Moves Model to Center of Tunnel and Scales Dimension so that the Tunnel Walls are at ± π

Figure 4. Origin Shift, X Scaling, and Y Distortion.

There remains the task of obtaining a two-dimensional boundary conforming coordinate system for a physical x - y cut. The procedure is similar to that outlined by Caughey and Jameson.[4] After a shift in origin to the center of curvature of the airfoil nose, and an appropriate scaling in x , a quadratic distortion in y is chosen such that values for the upper and lower tunnel walls are π and -π respectively (see Figure 4). Next an intermediate mapping is applied:

$$\bar{x} + i\bar{y} = \log\left[1-\cosh(\xi + i\eta)\right] - \log 2 \qquad (11)$$

This mapping allows unwrapping the domain about an arbitrarily chosen slit line emanating from the origin such that the upper and lower tunnel walls become the lower boundary $\eta = 0$, upstream infinity becomes the origin, and the two sides of the slit plus the airfoil become the upper boundary (located near $\eta = \pi$) as illustrated in Figure 5 (b) and (c). This slit line is chosen to pass thru the trailing edge thus eliminating an acute angle on the transformed boundary. It is also convenient to make this slit coincide with the location of the shed vorticity sheet which is assumed to be tangent to the trailing edge bisector at the trailing edge and parallel to the tunnel wall far downstream.

Finally, the computational domain with a coordinate conforming upper boundary (see Figure 5d) is obtained by stretching the ordinate according to

$$Y = \eta/\eta_{\text{upper}} \qquad (12)$$

An example of the result of these sequential mappings is the computer generated mesh shown in Figure 6.

Numerical Method

Details of the finite volume method are given in reference 3. The method is summarized in the following.

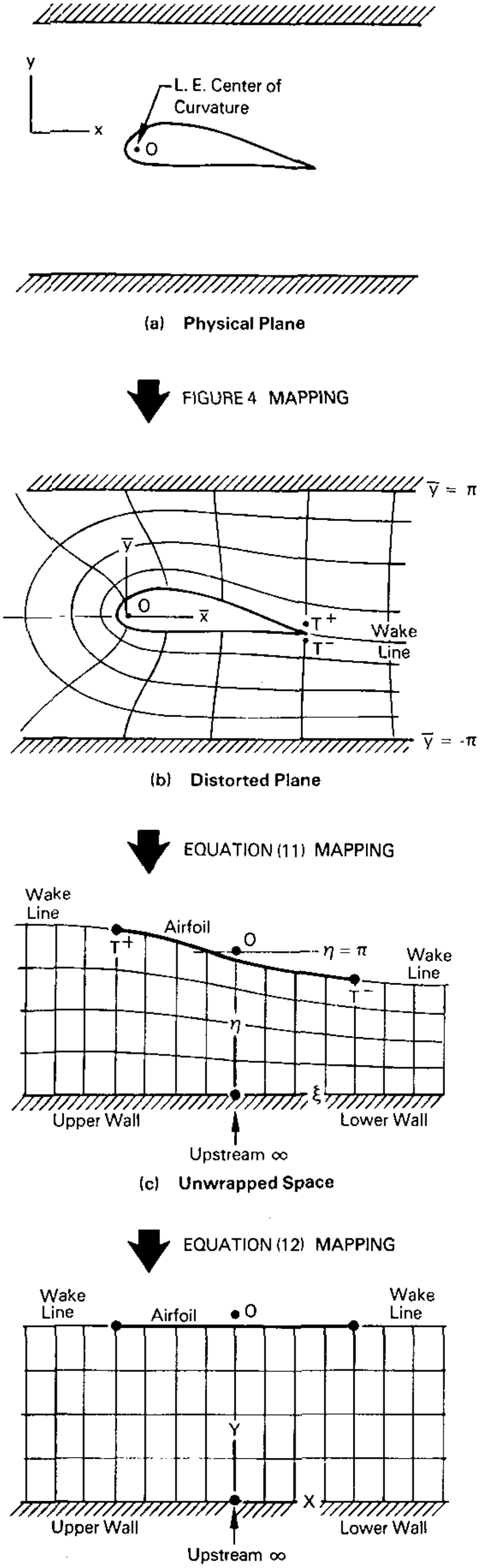
The grid in computational space divides the space into cubical cells. This leads to simple difference schemes for the numerical analysis. It can be shown that in terms of quantities associated with computational space, the divergence of the mass flux vector is given by

$$\frac{\partial}{\partial x^i}(\rho u^i) = \frac{\partial}{\partial X^i}(\rho h U^i) \qquad (13)$$

u^i = physical velocity components
U^i = contravariant velocity components
x^i = physical coordinates
X^i = transformed coordinates
h = determinant of the transformation matrix.

Also it can be shown that

$$U^i = g^{ij}\frac{\partial\phi}{\partial X^j} \qquad (14)$$



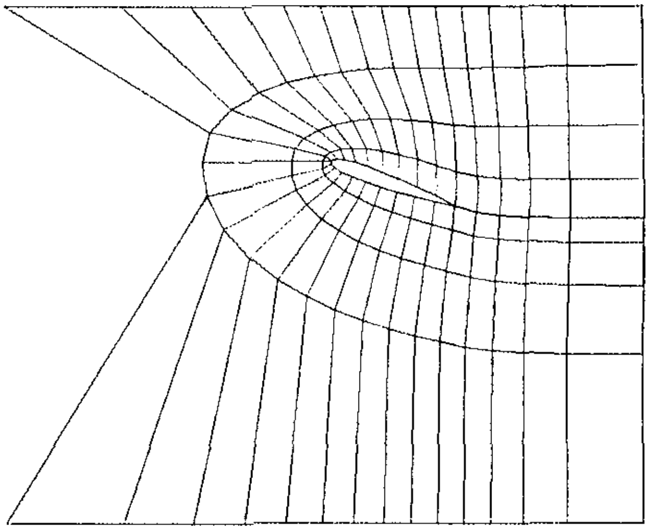
(a) **Physical Plane**

(b) **Distorted Plane**

(c) **Unwrapped Space**

(d) **Computational Space**

Figure 5. Sequential Mappings from Physical to Computational Space.



COARSE GRID

Figure 6. A Computer Plot of the Boundary Conforming Coordinates.

where g^{ij} is the inverse of the metric tensor of the transformation. Thus to satisfy the governing equation, Eq. (2), the finite volume algorithm uses an iterative procedure to make the right hand side of Eq. (13) approach zero.

The steps for one iteration for the algorithm are listed below.

(1) Using the current values for ϕ, the contravariant velocity U^i is computed using a difference scheme for Eq. (14).
(2) The density is computed from Eq. (3).
(3) The divergence of the mass flux vector is computed at each grid point according to Eq. (13) using a box difference scheme on a set of secondary cells nested in the primary set.
(4) Additional terms are added to the divergence to obtain an artificial viscosity, to offset certain lumping errors, and to embed the steady state equation in a convergent time dependent process (the iteration) which evolves to the solution (see reference 3 for details).
(5) New grid point values for ϕ are calculated so as to reduce the divergence to zero using successive line over-relaxation.

The numerical solution initially gives the values of ϕ at the grid points. The velocity is then calculated as follows. Since gradients are easily evaluated in the computational plane, Eq. (1) is first reformulated using the chain rule to obtain

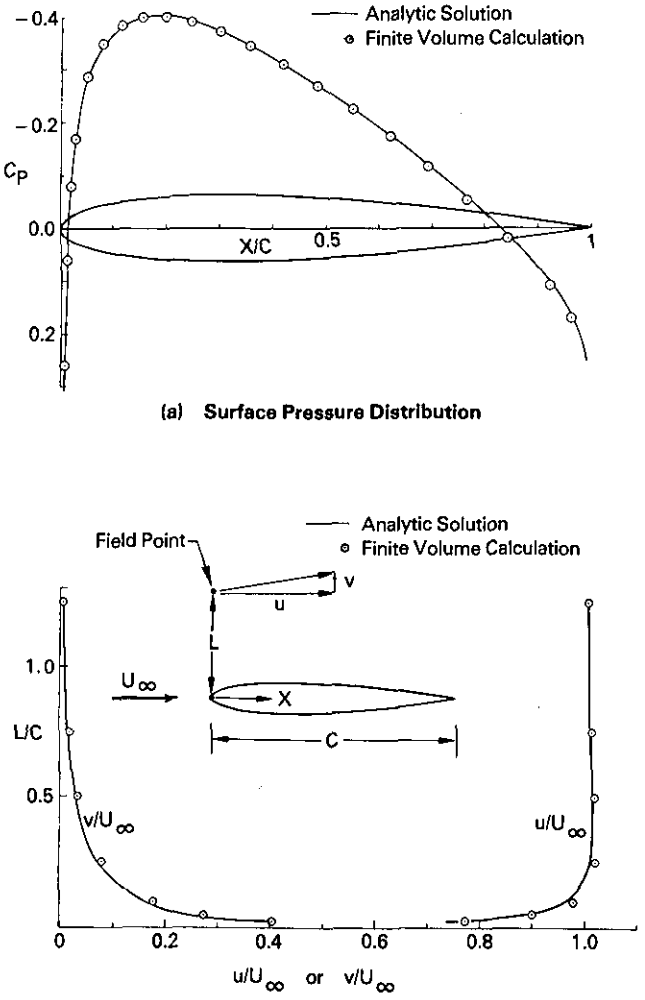
$$u^i = \frac{\partial X^j}{\partial x^i} \frac{\partial \phi}{\partial X^j} \qquad (15)$$

The grid point values of the velocity are calculated according to Eq. (15) by central differencing and are interpolated linearly in computational space to obtain the velocity at any point desired.

Example Calculations and Code Validation

Various example calculations are presented, including those made for comparison with experimental and other theoretical results. All of the comparisons with experimental data are for insignificant wall effects. No three dimensional experimental results with significant wall effects and known wall boundary conditions were found to compare with calculations. The ability of the code to properly treat the tunnel wall boundary condition was checked out using the results of other theoretical methods. These methods are two-dimensional so that the code calculations are made with unswept wings that span the tunnel.

Comparison with an exact incompressible potential flow calculation is presented in Figure 7 for a Karman-Trefftz airfoil in a tunnel. The exact solution was obtained by calculating the free-air flow about the airfoil using analytical procedures. The normal component of velocity was calculated along a line parallel to the free stream and above the airfoil as the upper tunnel wall boundary condition for the finite volume method computer calculation and similarly for the lower wall. The agreement at the airfoil surface and at field points given in Figure 7 is excellent. Field point comparison along a line going to the leading edge was chosen to illustrate the accuracy of the calculations where the errors would be most noticeable. Maximum descretization error for the finite volume method is expected near the leading edge where velocity gradients are largest. This comparison provides a validation for proper treatment of wall boundary conditions and for calculation of field point velocities.



(a) Surface Pressure Distribution

(b) Field Point Velocities Along X = O

Figure 7. Incompressible Potential Flow about a Karman Trefftz Airfoil.

To provide a transonic check case with tunnel walls, we calculated the two-dimensional flow about an NACA 0012 airfoil at a Mach number of 0.8 with solid tunnel walls (no normal flow) four chord lengths apart. Figure 8 shows a comparison of the finite volume method to the small disturbance method described by Murman, Bailey, and Johnson.[5] Here again the agreement is quite good. The pressure levels are very close and the shock positions are within one grid spacing. The full potential equation shock position is slightly upstream of the small disturbance equation. These relative positions are expected due to the differences in the two equations being solved.

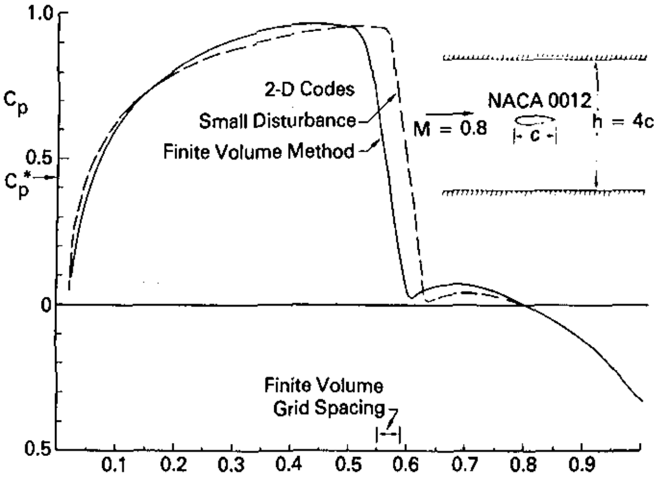


Figure 8. Calculation of the Transonic Flow about an NACA 0012 Airfoil in the Wind Tunnel.

In order to validate the code's treatment of the downstream boundary condition according to Equation 9, the tunnel flow problem of Figure 8 was repeated with normal flow at the wall. The field point velocities and densities calculated at the downstream boundary were found to satisfy Equation 9.

Three-dimensional transonic calculations were made for the ONERA Wing M6 for which experimental data are given in Reference 6 for a slotted wall tunnel that was simulating the free air case. The comparison is shown in Figure 9. The calculation was repeated for solid tunnel walls. The effect is shown in Figure 10. No comparison with experiment is available for this calculation.

Discussion and Conclusions

A computer code for a wing in a transonic wind tunnel has been developed which can be used to evaluate active wall control configurations.

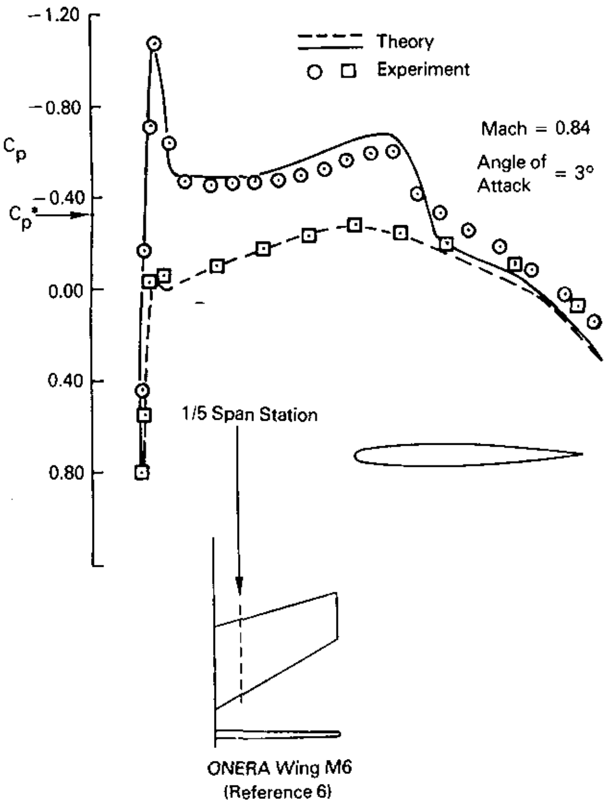


Figure 9. Finite Volume Method and Wind Tunnel Results for Transonic Flow about a Swept Wing in Free Air.

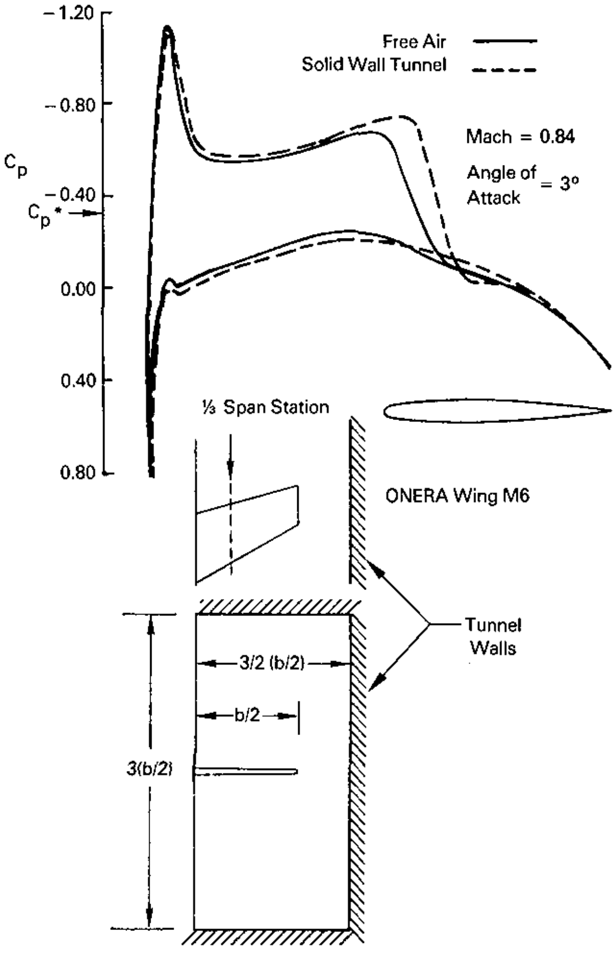


Figure 10. Effect of Solid Tunnel Walls on Free Air Calculation of Figure 9.

Currently the code can treat arbitrary wall normal flow boundary conditions. Comparisons of the code with analytical and numerical results have proven its accuracy. Comparisons with wind tunnel data have been limited to cases with insignificant wall effects because of a lack of appropriate data. Effort is currently being directed to gather appropriate experimental data to further check the code with wall interference.

The output from the code provides pressures, velocities and local Mach numbers on the wing and at any points in the flow field specified by the user. This feature allows for flexibility in checking various active wall control convergence algorithms. For example, measurements on surfaces inside the physical walls can be simulated.

Future work will be devoted to further verify the code and extend its modeling capability to include wing-bodies. Also, pressure boundary conditions for the wall will be included. This latter change should allow the code to model all of the wall control algorithms currently under consideration.

Acknowledgement

This work was sponsored by the Arnold Engineering Development Center under Contract F40600-79-C-001, "Development of a Computer Code to Model Three-Dimensional Self-Correcting Transonic Wind Tunnels."

References

1. Ferri, A. and Baronti, P., "A Method for Transonic Wind Tunnel Corrections," AIAA Journal, Vol. 11, No. 1, January 1973, pp. 63-66.

2. Sears, W.R., "Self-Correcting Wind Tunnels," Calspan Report No. RK-5070-A-2, July 1973 (The Sixteenth Lanchester Memorial Lecture); also the Aeronautical Journal, Vol. 78, No. 758/759, February/March 1974, pp. 80-89.

3. Jameson, Antony, and Caughey, D.A., "A Finite Volume Method for Transonic Potential Flow Calculations," AIAA Paper 77-635, June 1977.

4. Caughey, D.A., and Jameson, Antony, "Numerical Calculation of Transonic Potential Flow about Wing-Fuselage Combinations," AIAA Paper 77-677, June 1977.

5. Murman, E.M., Bailey, F.R., and Johnson, M.H., "TSFOIL - A Computer Code for 2-D Transonic Calculations, Including Wind-Tunnel Wall Effects and Wave-Drag Evaluation." NASA SP-347, Part II, 1975, pp. 769-788.

6. Monnerie, B., and Charpin, F., "Essais de buffeting d'une aile en fleche en transsonique," 10e d'Aerodynamique Appliquee, Lille, November 1973.