NASA Technical Memorandum 86716

A Consistent Spatial Differencing Scheme for the Transonic Full-Potential Equation in Three Dimensions

Scott D. Thomas and Terry L. Holst

December 1985

National Aeronautics and Space Administration

NASA Technical Memorandum 86716

A Consistent Spatial Differencing Scheme for the Transonic Full-Potential Equation in Three Dimensions

Scott D. Thomas,
Terry L. Holst, Ames Research Center, Moffett Field, California

December 1985

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

A CONSISTENT SPATIAL DIFFERENCING SCHEME FOR THE TRANSONIC FULL-POTENTIAL EQUATION IN THREE DIMENSIONS

Scott D. Thomas* and Terry L. Holst

Ames Research Center

SUMMARY

A full-potential, steady, transonic wing, flow solver has been modified so that free-stream density and residual are captured in regions of constant velocity. The numerically precise free-stream consistency is obtained by slightly altering the differencing scheme without affecting the implicit solution algorithm. The changes chiefly affect the fifteen metrics per grid point, which are computed once and stored. With this new method, the outer boundary condition is captured accurately, and the smoothness of the solution is especially improved near regions of grid discontinuity.

INTRODUCTION

Free-stream consistency is defined as the ability to numerically capture the free-stream density and vanishing residual from the transformed governing equations on an arbitrary computational mesh. The calculation of these quantities must be algebraically precise; that is, accurate to within machine roundoff, instead of to within some order of magnitude associated with the mesh spacing.

Following the work of Flores et al. (ref. 1), modifications have been made to the three-dimensional (3-D) full-potential flow solver described by Thomas and Holst (ref. 2) in order to implement a free-stream-consistent spatial differencing scheme. This program uses a coordinate transformation from physical to computational space in order to represent accurately the aerodynamic surface. It models the inviscid, compressible, irrotational flow of air past a lifting wing mounted on a wall. The vectorized computer code TWING, Cycle 3, has been upgraded to Cycle 4. It is designed to run on a Cray supercomputer.

The spirit of the present approach follows the description in reference 1 in that it only requires careful numerical definition and use of metric terms required by the coordinate transformation. This is distinct from other approaches involving the subtraction of "artificial flux" terms as mentioned, for example, by Caughey and Jameson (ref. 3).

*Informatics General Corporation, Palo Alto, CA 94303.

Beneficial features of TWING, Cycle 4, include:

1. Elimination of grid-induced perturbations in the flow field near the outer (or free-stream) boundary

2. Reduction of perturbations in the flow field that otherwise would be introduced by grid discontinuities

3. Improved ability to treat wings with sharp leading edges

On the negative side, instead of seven metric quantities per point, now fifteen are required. However, a new data compaction and expansion technique keeps the storage requirement close to that for the previous cycle.

The operation count per iteration per grid point remains about the same as before, reduced by the elimination of some averaging operations, but this is offset by the addition of the data expansion operations. The number of iterations required for convergence, which is usually taken as a two-order drop in the maximum modulus of the residual, increases for some cases, yet decreases for others. The initial residual for Cycle 4 is usually less than two orders of magnitude smaller than before.

GOVERNING EQUATIONS

This description is taken from reference 2 and is included here for future reference. The 3-D full-potential equation written in strong conservation-law form in Cartesian coordinates is given by

$$(\rho\phi_x)_x + (\rho\phi_y)_y + (\rho\phi_z)_z = 0 \tag{1a}$$

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}(\phi_x^2 + \phi_y^2 + \phi_z^2)\right]^{[1/(\gamma-1)]} \tag{1b}$$

where the density ρ is scaled by the stagnation density ρ_s, the velocity components ϕ_x, ϕ_y, ϕ_z are scaled by the critical speed of sound a_*, and γ is the ratio of specific heats. The value 1.4 is assigned to γ in all cases under discussion.

The orthogonal physical coordinates x, y, z are oriented in the flow, span, and vertical directions, respectively. A general coordinate transformation is applied to the above equations, and the implicit AF2 relaxation takes place in the new coordinate system. Equation (1), after the transformation, is

$$(\rho U/J)_\xi + (\rho V/J)_\eta + (\rho W/J)_\zeta = 0 \tag{2a}$$

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}\left(U\phi_\xi + V\phi_\eta + W\phi_\zeta\right)\right]^{[1/(\gamma-1)]} \tag{2b}$$

where ξ, η, and ζ represent wraparound, spanwise, and radial-like directions, respectively, and the contravariant velocity components U, V, and W are defined by

$$\begin{vmatrix} U \\ V \\ W \end{vmatrix} = H^T H \begin{vmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{vmatrix} \tag{3}$$

The quantities H and J are defined by

$$H = \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{vmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{vmatrix} \tag{4}$$

and $J = \det(H)$. Dirichlet boundary conditions for the potential are imposed at the outer free-stream boundaries, and Neumann boundary conditions that yield flow tangency are imposed at the symmetry wall and wing surface boundaries.

The discretization of the coordinate space replaces the real variables ξ, η, and ζ with integer variables i, j, and k, respectively.

CONSISTENT FREE-STREAM DENSITY

Assume that the free-stream potential function is given by

$$\phi = u_\infty x + v_\infty y + w_\infty z \tag{5}$$

so that the gradient, $\nabla\phi = (u_\infty, v_\infty, w_\infty)^T$, gives the free-stream velocity vector. From equation (1b) it is seen that the density is a function of the magnitude of the velocity vector, so that

$$\rho_\infty = \rho(u_\infty^2 + v_\infty^2 + w_\infty^2) \tag{6}$$

The method used to capture the free-stream density is simply to difference ϕ in i, j, and k the same way that x, y, and z are differenced in order to find the metrics. The metrics are defined to be the elements of the matrix $H^T H$ (see eqs. (3) and (4) above). An algebraic proof that this yields free-stream density is given below. This will also serve as an introduction to the next section on consistent residual computation.

From equation (2b), one sees that it will suffice to show that

$$U\phi_\xi + V\phi_\eta + W\phi_\zeta = u_\infty^2 + v_\infty^2 + w_\infty^2 \tag{7}$$

when finite differences replace derivatives. Equation (3) can be used to establish that the inner product

$$\begin{aligned} U\phi_\xi + V\phi_\eta + W\phi_\zeta &= (U,V,W)(\phi_\xi,\phi_\eta,\phi_\zeta)^T \\ &= (\phi_\xi,\phi_\eta,\phi_\zeta)H^TH(\phi_\xi,\phi_\eta,\phi_\zeta)^T \\ &= A^TA \end{aligned} \tag{8}$$

where the column vector $A = H(\phi_\xi,\phi_\eta,\phi_\zeta)^T$. The method used to compute H is first to form a matrix of finite differences of x, y, and z,

$$K = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix} \tag{9}$$

and then invert to find $H = K^{-1}$. For instance, the quantity x_ξ at $(i+1/2,j,k)$ is found to be $x_{i+1,j,k} - x_{i,j,k}$.

By the chain rule, the following holds analytically:

$$\begin{vmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{vmatrix} = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix} \begin{vmatrix} \phi_x \\ \phi_y \\ \phi_z \end{vmatrix} \tag{10}$$

When differenced numerically, this becomes an approximation. The symbol K^* shall denote the matrix on the right-hand of equation (10) when the difference formulas for ϕ are identical to those used for x, y, and z. Thus K^* is the same as K of equation (9). Finally, it is seen that equation (3) and the definition of ϕ leads to the identity

$$A = K^{-1}K^*(\phi_x,\phi_y,\phi_z)^T = (u_\infty,v_\infty,w_\infty)^T \tag{11}$$

Equation (7) follows from (8) and (11), which completes the proof.

This technique has been implemented in TWING Cycle 4 to find the densities at the $(i+1/2,j,k)$ grid locations. Six density metrics are required per grid point since the metric array, H^TH, is symmetric. Whenever the density is needed at other

locations, like (i,j+1/2,k), the value is obtained from an average of neighboring points. In a free-stream region of flow all of the densities will be the same, so that the generality of the formulation is retained.

CONSISTENT FREE-STREAM RESIDUAL

This section examines how the metrics must be defined in order to have the numerically obtained residual vanish identically, independent of the shape of the grid. Since the Jacobian of the transformation appears in the denominator, we make the assumption that it remains positive. This corresponds to the condition that grid lines of the same family do not cross. The density is constant, so it may be removed from equation (2a), leaving:

$$(U/J)_\xi + (V/J)_\eta + (W/J)_\zeta = 0 \tag{12}$$

Treating this analytically, the left-hand side of equation (12) may be rewritten as an inner product:

$$\left(\frac{\partial}{\partial\xi}, \frac{\partial}{\partial\eta}, \frac{\partial}{\partial\zeta}\right) \left([\det(K_1)(K_2^{-1})^T K_3^{-1}] K^* \begin{vmatrix} \phi_x \\ \phi_y \\ \phi_z \end{vmatrix} \right) \tag{13}$$

Once again, the letter K denotes a transformation matrix as defined in equation (10). The superscript and subscripts on K will be used below to indicate slightly different differencing methods used to approximate the analytically defined K. As in the previous section, K^* denotes the matrix that arises from the method used to difference ϕ.

In equation (13) we actually have control over the part of the right-hand side that appears in square brackets. It is evident that by forming K_3 with the same difference formulas as those used in forming ϕ, then $K_3^{-1}K^* = I$ may be dropped from the equation. The free-stream condition on ϕ then allows us to rewrite this expression as:

$$\left(\frac{\partial}{\partial\xi}, \frac{\partial}{\partial\eta}, \frac{\partial}{\partial\zeta}\right) \left([\det(K_1)(K_2^{-1})^T] \begin{vmatrix} u_\infty \\ v_\infty \\ w_\infty \end{vmatrix} \right) \tag{14}$$

It only remains to show how to form the remaining K matrices so that this expression will vanish identically when finite differences replace the derivatives.

The values assigned to u_∞, v_∞, and w_∞ are arbitrary, and equation (14) is linear, so it will suffice to treat the case for which $u_\infty = 1$ and $v_\infty = w_\infty = 0$. This amounts to an assumption about the direction of the free-stream velocity. The general case will follow by permutation.

Make the assumption that $K_1 = K_2 = K$ (drop the subscripts), since analytically

$$(K^{-1})^T = \frac{1}{\det(K)} \begin{vmatrix} y_\eta z_\zeta - y_\zeta z_\eta & \cdots & \cdots \\ y_\zeta z_\xi - y_\xi z_\zeta & \cdots & \cdots \\ y_\xi z_\eta - y_\eta z_\xi & \cdots & \cdots \end{vmatrix} \tag{15}$$

so the determinant will divide out. Six elements of the matrix in equation (15) have been omitted because the assumption on the free-stream direction renders them immaterial. The problem has been reduced to showing how the following expression can be made to vanish:

$$\left(\frac{\partial}{\partial\xi}, \frac{\partial}{\partial\eta}, \frac{\partial}{\partial\zeta}\right) \begin{vmatrix} y_\eta z_\zeta - y_\zeta z_\eta \\ y_\zeta z_\xi - y_\xi z_\zeta \\ y_\xi z_\eta - y_\eta z_\xi \end{vmatrix} \tag{16}$$

This is known as the "third consistency condition" in reference 1. It should be clear that, if this were an analytic expression, then it would indeed vanish by the property of equality of mixed partial derivatives.

The first summand of equation (16), $(y_\eta z_\zeta - z_\eta y_\zeta)_\xi$, is approximated by

$$(y_\eta z_\zeta - z_\eta y_\zeta)_{i+1/2,j,k} - (y_\eta z_\zeta - z_\eta y_\zeta)_{i-1/2,j,k} \tag{17}$$

A natural way to expand the first derivative in this expression, y_η, is $(y_{i+1/2,j+1/2,k} - y_{i+1/2,j-1/2,k})$. Now replace $y_{i+1/2,j+1/2,k}$ with an average of y at cell centers, namely $(y_{i+1/2,j+1/2,k+1/2} + y_{i+1/2,j+1/2,k-1/2})/2$. Such averages may be applied to all the other terms in equation (17). This procedure may be carried to the point where every term in equation (16) is replaced by finite difference expressions involving only the values of x, y, and z at the eight neighboring cell centers surrounding the grid point at (i,j,k).

Complete cancellation can be obtained by forming the primitive quantities for x, y, and z at the cell centers using averages of the eight nearest neighbors. For instance,

$$x_{i+1/2,j+1/2,k+1/2} = (1/8)(x_{i,j,k} + x_{i+1,j,k} + x_{i,j+1,k} + x_{i+1,j+1,k} + x_{i,j,k+1} + x_{i+1,j,k+1} + x_{i,j+1,k+1} + x_{i+1,j+1,k+1}) \tag{18}$$

As a point of interest, Pulliam and Steger (ref. 4) suggested a similar averaging for the Euler equations, but they studied this "third consistency condition" without the other features mentioned above.

Verification of the cancellation property is left as an exercise for the reader. Numerical verification has been conducted with a small Fortran program called RFTEST. It was set up to run through a sequence of random perturbations of a 39-point difference molecule surrounding the single point of interest. The residual, the left-hand side of equation (2a), was found to be consistently close to zero when three different machine precisions were used. Four, eight, and sixteen-byte real-number representations were tested on a DEC VAX computer.

It is customary to discretize equation (2a) as a sum of three flux terms (shown one per line)

$$(\rho U/J)_{i+1/2,j,k} - (\rho U/J)_{i-1/2,j,k} + (\rho V/J)_{i,j+1/2,k} - (\rho V/J)_{i,j-1/2,k} + (\rho W/J)_{i,j,k+1/2} - (\rho W/J)_{i,j,k-1/2} = 0 \qquad (19)$$

The flux terms at adjacent points are shared, so at each point of the mesh there are three "flux metrics" required for each of the three coordinate directions. Although three of these metrics are computed at the same half-points as the density metrics, they must be kept separate for free-stream consistency. Thus, six plus nine (or fifteen) metrics per point are computed in order to impose free-stream consistency for the full-potential equation in three dimensions.

It is an interesting exercise to show that if the 3-D grid is composed of a stack of parallel 2-D grids, then some of the density and flux metrics may be shared while retaining free-stream consistency. At the same time, the compactness of the differencing molecule for density and residual may remain as described above. In this case, thirteen metric quantities would be needed per grid point. An early version of the RFTEST program was used to show that this property does not hold for a general computational mesh. This method was also tested in TWING but was later replaced by the fifteen-metric method described above. It should be noted that a remnant of the earlier method remains implemented in Cycle 4 in the wing extension region beyond the tip for the innermost k plane.

In reference 1 it is shown that a consistent differencing scheme can be formulated with only thirteen metrics by forming the ϕ differences for density in a manner consistent with the way the metrics are formed for the residual. This would lead to a larger density differencing molecule and, thus, to more numerical smearing of the solution in the vicinity of a shock; therefore, it is not recommended.

PERFORMANCE NEAR A GRID DISCONTINUITY

There are several cases in which it is convenient to use a computational grid that does not have optimal smoothness. A simple example of this is in the treatment

of a wing with break discontinuities in which the sweep angle of the leading and trailing edges may change abruptly. The consistency method guarantees good results using a simple grid which may propagate the break discontinuity to the outer boundary. In another example, one may wish to use a block approach in which a transonic wing flow solver just treats wing-like elements of a complicated geometry. In this case, it may be necessary to make the outer boundary of the wing block conform to the surfaces presented by the other blocks.

Figure 1 shows an example of a grid about a wing that is mounted on a fuselage-like bump on a wall as an example of a case for which one might want to use a grid that is not made of a stack of two-dimensional grids. In this case, we may choose to propagate the bump throughout the grid from the root wall to the opposite free-stream boundary. The approach presented above applies in a limited sense to regions of the flow field where the velocity changes slowly, such as fore and aft of the wing. The region near the wing enjoys denser packing of grid points owing to the topology chosen for the flow solver, so consistent differencing is not as major an issue.

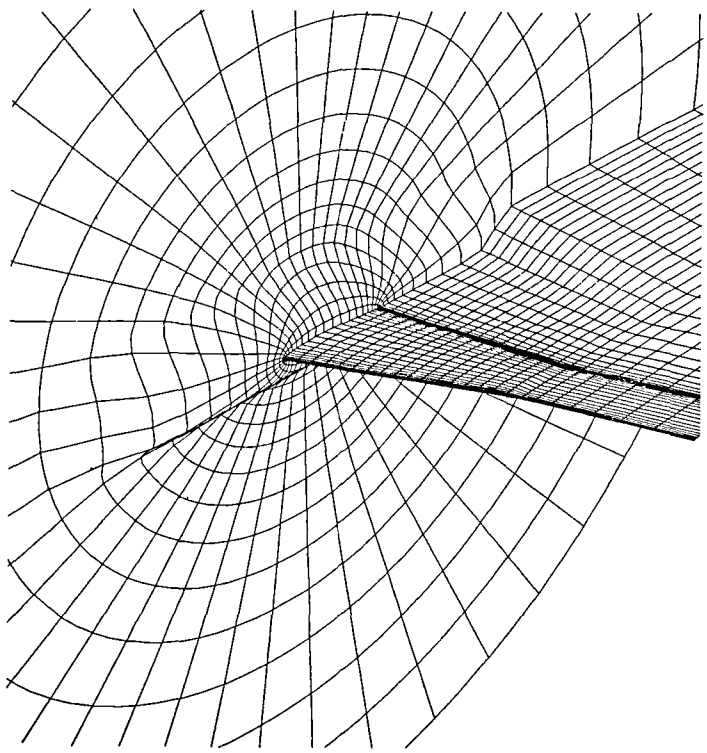

Figure 1.- Grid about a wing on a fuselage.

In the following discussion, computed results of Cycle 3 and Cycle 4 of the transonic wing analysis code TWING are compared. Cycle 4 of TWING uses the consistent spatial differencing described in the previous section, but the earlier Cycle 3 version does not.

Comparison of Cycle 3 and Cycle 4

The grid generator supplied with TWING is designed to build a three-dimensional grid by stacking two-dimensional grids, located at j = constant values, along the spanwise direction. Near the tip of the wing there is an abrupt transition from airfoil cross sections of positive thickness to zero thickness. The grid propagates this abrupt change from the inner to the outer boundaries, and its effect can be clearly seen by observing the velocity vectors in a fixed k plane. Figure 2 was generated by Cycle 3 with the default M6 wing at Mach 0.84 and zero angle of attack on a grid of 40,050 points (89 × 25 × 18). It shows a plot of the projections of the velocity vectors in the computational plane for $k = 6$ (about one root chord downstream) into a plane that is located aft of the wing that is aligned perpendicular to the free-stream direction. Since the angle of attack is zero, we expect that the projections will have almost no magnitude. Observe that this does not happen at points lying in the three j planes just beyond the tip. In addition, the directions of the projections change abruptly as we move from j planes that contain wing sections to j planes beyond the wing tip, which do not. In figure 3, we see the same case as above except that the velocity vectors were generated with Cycle 4. We see that each projection does have almost zero magnitude and also that the directions of the projections transition smoothly throughout the entire region that is shown.

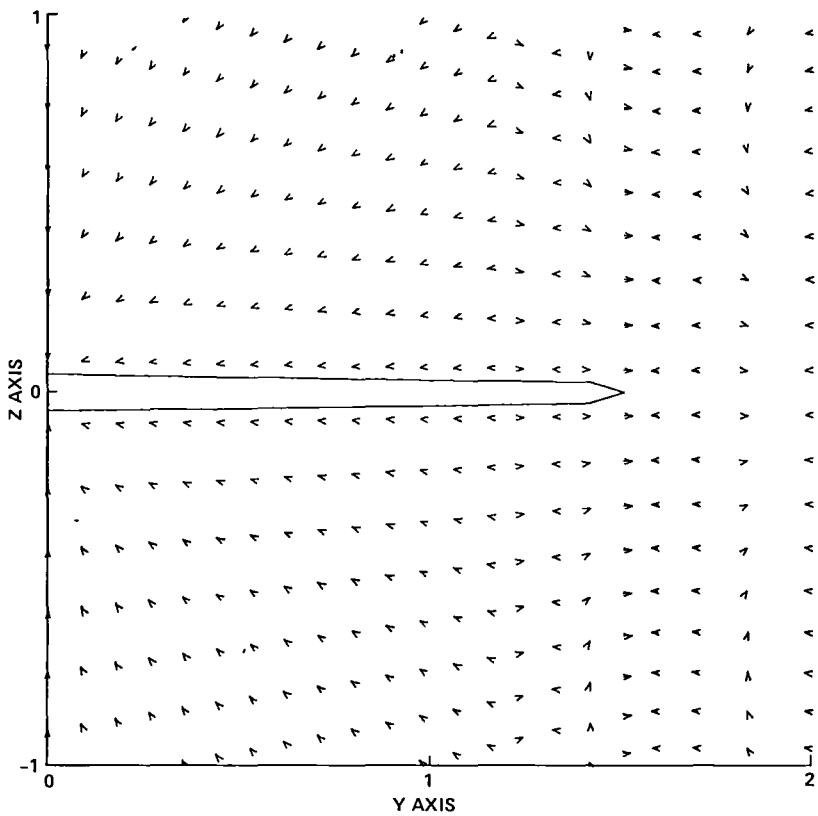

Figure 2.- Velocity vector projection for $M_\infty = 0.84$, $\alpha = 0°$, TWING Cycle 3.

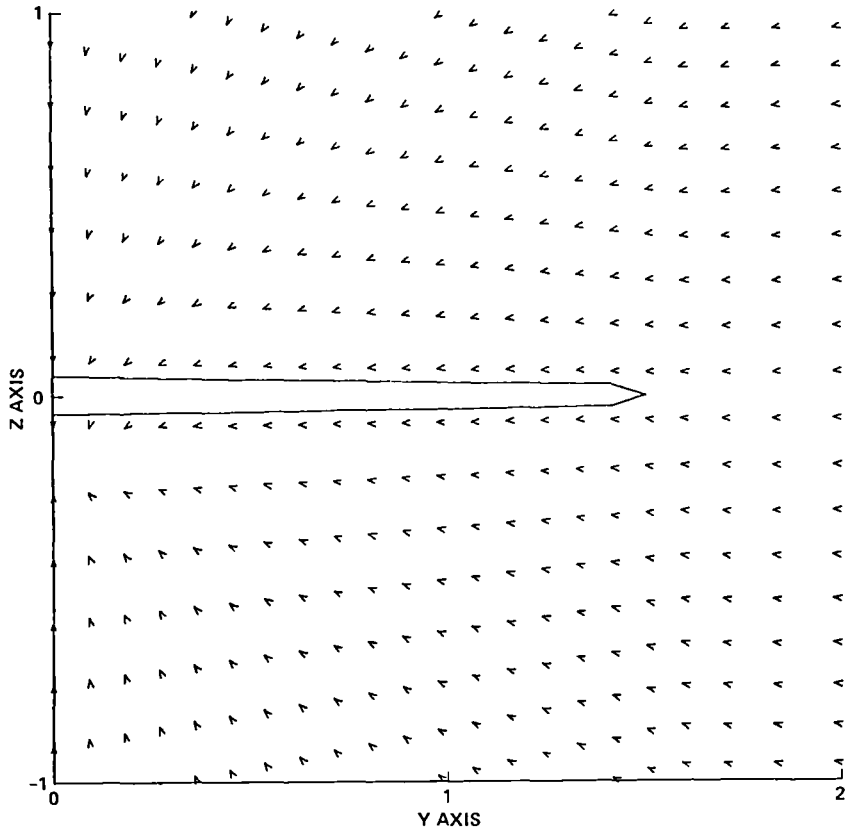

Figure 3.- Velocity vector projection for M_∞ = 0.84, α = 0°, TWING Cycle 4.

Again we compare the two cycles of TWING in the next two figures, but in this case the angle of attack assumes its default value of 3.06°. Figure 4 shows the projections of the velocity vectors for the same k plane as before, using Cycle 3. Since the angle of attack is nonzero, the plane of projection has been inclined 3.06° so the wake region appears to be offset from the wing location. A pattern that resembles a vortex is generated at the edge of the wake region near the tip, despite the inherent potential assumption of flow irrotationality. The appearance of this flow pattern can be explained by the fact that a circulation model is added to the governing equations in order to simulate a lifting wing. This is a common approach among wing codes (flow solvers) that are based on a potential formulation. Upon comparison with figure 5, which was generated by Cycle 4, we see that the irregularities in the flow field near the tip have been eliminated.

CONSIDERATIONS FOR A SHARP LEADING EDGE

Some applications require the analysis of a wing with a sharp leading edge. This is considered to be another type of grid discontinuity, especially with the "O"

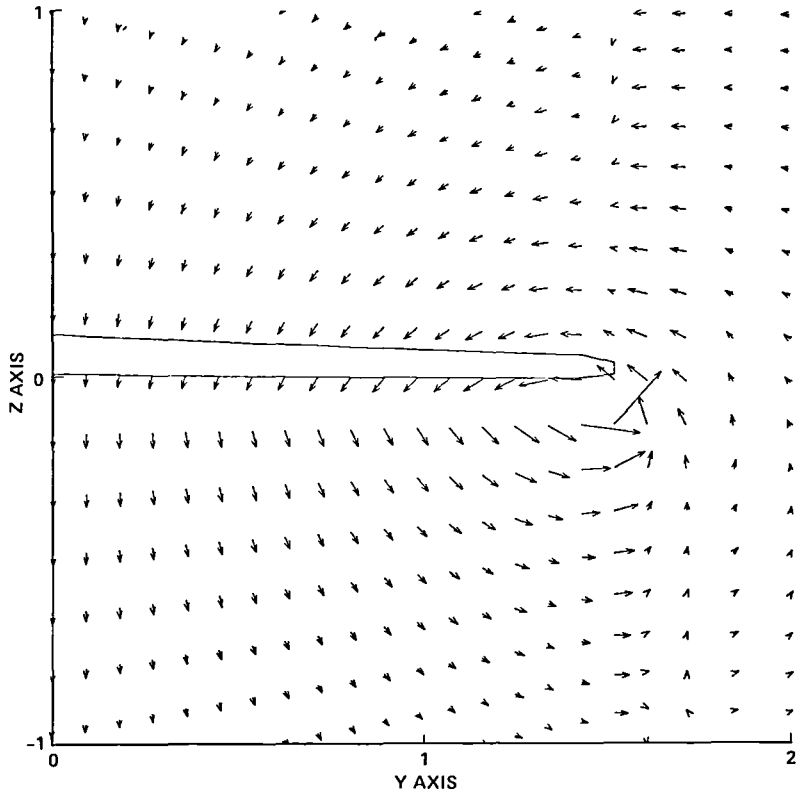

Figure 4.- Velocity vector projection for $M_\infty = 0.84$, $\alpha = 3.06°$, TWING Cycle 3.

mesh topology used in TWING. It was found that Cycle 4 (with consistent spatial differencing) is capable of treating a wing with a sharp leading edge, as opposed to Cycle 3 which is not. To demonstrate this point, a simple wing shape was devised having rectangular planform and an airfoil cross section for which the upper part is taken from a 12%-thick circular-arc airfoil and the lower part is taken to be a flat plate. This results in an airfoil that is 6% thick.

Figure 6 shows a plot of the coefficient of pressure for two wing stations (root and mid-span) which illustrates that a solution generated by Cycle 3 exhibits erratic behavior near the leading edge discontinuity. This excessive gradient can cause stability problems under some conditions. The Mach number for this case is subsonic, at 0.2, and the angle of attack is zero. The leading edge did not present as much of a problem for Cycle 4. Figure 7 shows the kind of solution that can be expected for the same case as above. This figure was generated without any of the options discussed in the following paragraphs.

In Cycle 4, the user may request the grid generator to use a double spline interpolation, one for each of the upper and lower surfaces, that will guarantee the midpoint in the ξ direction to be positioned at the leading edge. This feature was added because the flow solver makes an assumption about the flow direction at the wing surface when its magnitude is supersonic. It arose from the considerations that some wing cross sections have different upper and lower arc lengths, and that a

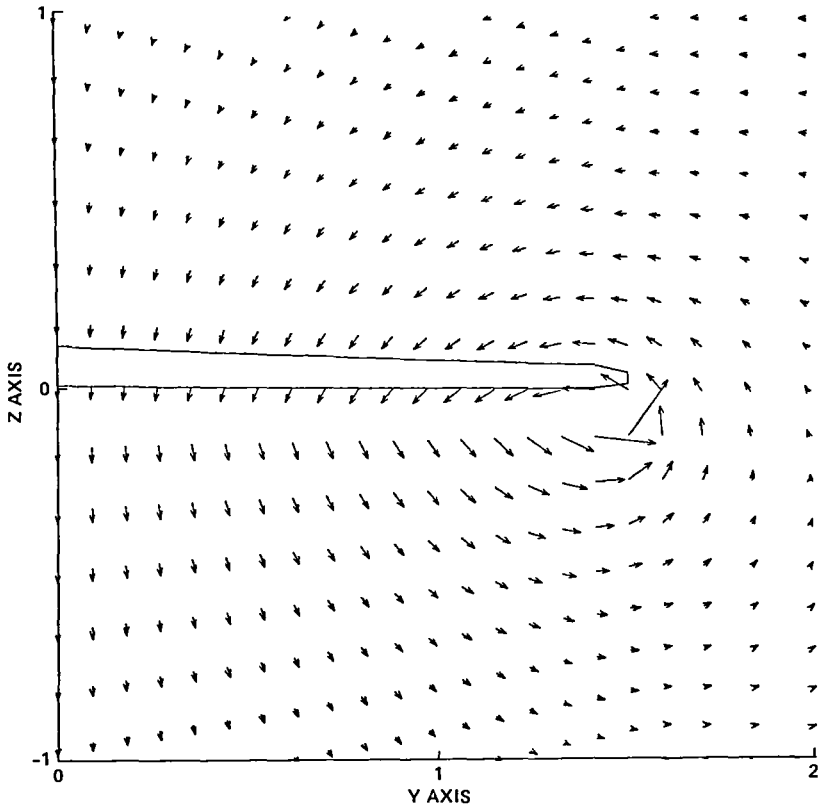

Figure 5.- Velocity vector projection for $M_\infty = 0.84$, $\alpha = 3.06°$, TWING Cycle 4.

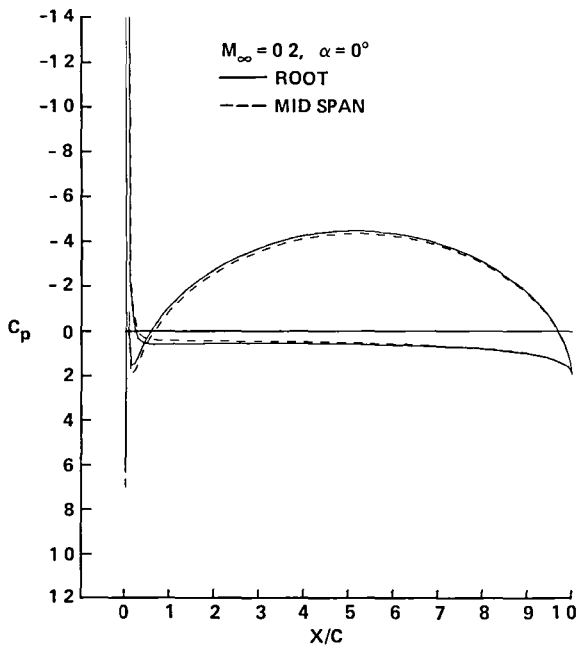

Figure 6.- Sharp leading edge, TWING Cycle 3.

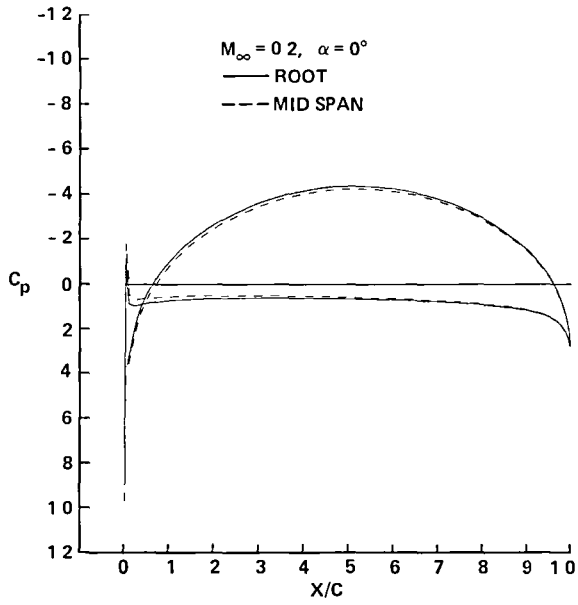

Figure 7.- Sharp leading edge, TWING Cycle 4.

large velocity gradient is expected near a sharp leading edge, especially at nonzero angle of attack. To select this option in the grid generator, set SLEG = 1.0 in the input list.

Assuming that the above option has been taken, another problem arises. Call the leading edge point i = IHALF, the wing surface at k = NK, and assume that j ranges along wing span stations. With the consistent differencing scheme, three metrics at (IHALF, j+1/2, NK) must be computed. It is not unusual for such points to have grid lines leaving the surface at obtuse angles. Since one-sided differences are still used at inner boundary points, the Jacobian may be numerically zero or even slightly negative at such points. Given a sophisticated grid generator, this problem can be eliminated. In the present case, however, a different approach was taken. The metrics and fluxes at the leading edge are split into upper and lower contributions, located at (IHALF-1/4, j+1/2, NK) and (IHALF+1/4, j+1/2, NK), respectively (identical to trailing edge treatment). A study involving the M6 wing, built into TWING, shows negligible impact on a wing with a blunt leading edge, while some improvement has been observed for a wing with a sharp leading edge. Finally, the user has the option of averaging the densities at the (IHALF-1/2, j, NK) and (IHALF+1/2, j, NK) points with their two neighbors at IHALF-3/2 and IHALF+3/2. To turn this on, set SLET = 1.0 in the input list. This leads to a more accurate plot of the coefficient of pressure in the vicinity of the leading edge for sharp leading edge cases.

COMPARISONS WITH A TWO-DIMENSIONAL FLOW SOLVER

The 2-D flow solver developed by Flores et al. in reference 1 was a modified version of TAIR, described in reference 5. It is a 2-D full-potential transonic flow code for airfoils, and is very similar to TWING. The features of the former flow solver are embedded in the code known at NASA Ames Research Center as TAIR10K. It was used in order to make two final comparisons.

Using the same cross section as that of the wing presented in figures 6 and 7, we show the effect of higher Mach number. For these comparisons, the wing grid cross sections of TWING and the TAIR10K grid have the same number of points, 151 × 31, and the wing has aspect ratio 20.0. Figure 8 shows a good comparison between the two codes for Mach 0.70. The flow on the upper surface is slightly supersonic without a shock. The wing code does not exhibit the peak at the leading edge as does the airfoil code. This is a consequence of the double interpolation at the leading edge lower and upper surfaces in TWING. Figure 9 shows what happens at Mach 0.75. In this case, a strong shock forms at about 80% of chord. The pressure peak at the leading edge for the 2-D flow solver is observed again. It is interesting that the shock for the wing is located slightly forward of the shock for the airfoil. It probably is caused by a 3-D relief effect in the wing calculation.

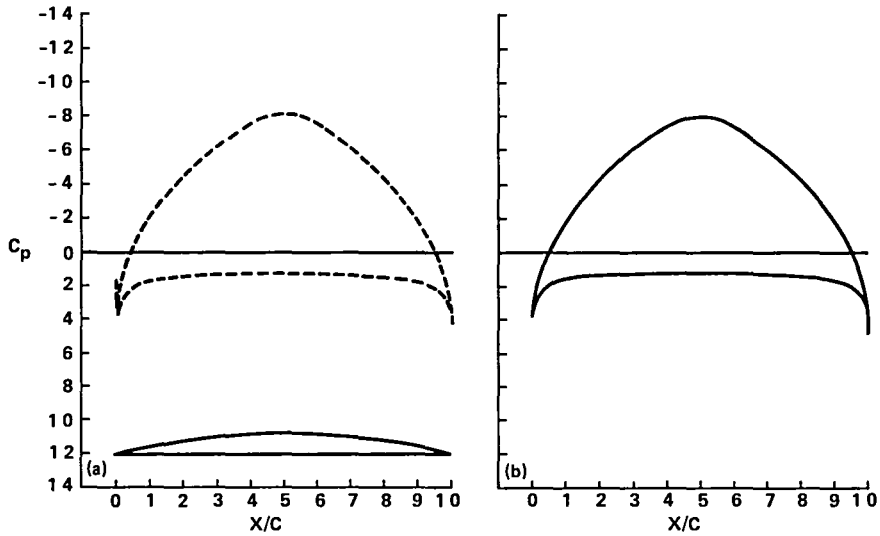

Figure 8.- Sharp leading edge, $M_\infty = 0.70$, $\alpha = 0$.
(——) TWING Cycle 4. (----) TAIR10K.

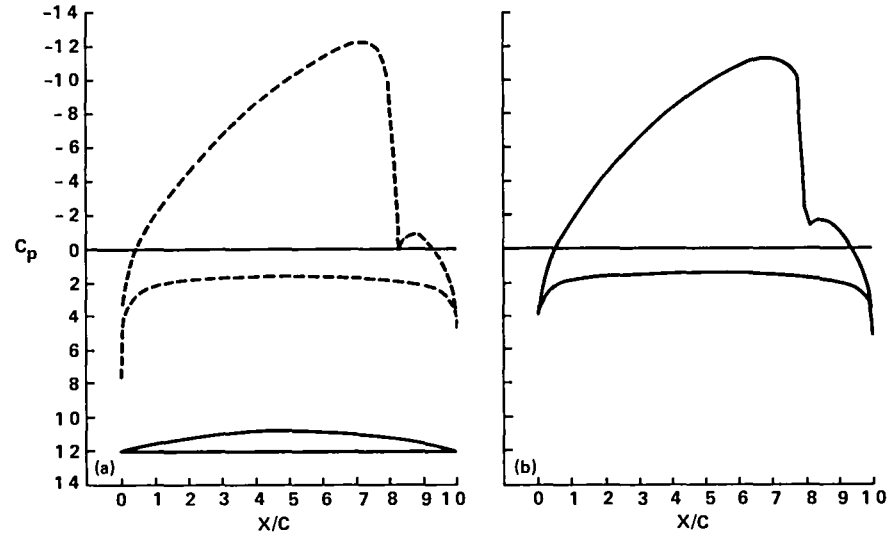

Figure 9.- Sharp leading edge, $M_\infty = 0.75$, $\alpha = 0$.
(——) TWING Cycle 4. (----) TAIR10K.

CONCLUSIONS

The full-potential, steady, transonic, wing-flow solver, TWING, has been modified so that free-stream density and vanishing residual are captured in regions of constant velocity. Numerically precise consistency, which does not depend on the fineness of the computational grid, is obtained by slightly altering the differencing scheme without affecting the implicit AF2 solution algorithm.

The computational efficiency remains about the same as before because the changes chiefly affect the fifteen metrics per grid point, which are computed once and stored. For each iteration, operations that were added for data expansion are offset by averaging operations that were eliminated. The number of iterations to convergence has been observed to increase for some cases, but decrease for others.

With this new method, the outer boundary condition is accurately captured. We have seen that the smoothness of the solution is especially improved near regions of grid discontinuity.

REFERENCES

1. Flores, J.; Holst, T. L.; Kwak, D.; and Batiste, D. M.: A New Consistent Spatial Differencing Scheme for the Transonic Full-Potential Equation. AIAA J., vol. 22, no. 8, Aug. 1984, p. 1027.

2. Thomas, S. D.; and Holst, T. L.: Numerical Computation of Transonic Flow About Wing-Fuselage Configurations on a Vector Computer. AIAA Paper 83-0499, Jan. 1983.

3. Caughey, D. A.; and Jameson, A.: Basic Advances in the Finite Volume Method for Transonic Potential Flow Calculations. Presented at the Numerical and Physical Aspects of Aerodynamic Flows Symposium, California State University, Long Beach, California, Jan. 1981.

4. Pulliam, T. H.; and Steger, J. L.: Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow. AIAA J., vol. 18, Feb. 1980, pp. 159-167.

5. Dougherty, F. C.; Holst, T. L.; Gundy, K. L.; and Thomas, S. D.: TAIR--a Transonic Airfoil Analysis Computer Code. NASA TM-81296, May 1981.

1 Report No	2 Government Accession No	3 Recipient's Catalog No
NASA TM 86716		

4 Title and Subtitle	5 Report Date
A CONSISTENT SPATIAL DIFFERENCING SCHEME FOR THE TRANSONIC FULL-POTENTIAL EQUATION IN THREE DIMENSIONS	December 1985
	6 Performing Organization Code
7 Author(s)	8 Performing Organization Report No
Scott D. Thomas and Terry L. Holst	85213
	10 Work Unit No
9 Performing Organization Name and Address	
Ames Research Center Moffett Field, CA 94035	11 Contract or Grant No
	13 Type of Report and Period Covered
12 Sponsoring Agency Name and Address	Technical Report
National Aeronautics and Space Administration Washington DC, 20546	14 Sponsoring Agency Code 505-31-01

15 Supplementary Notes

Point of contact: Scott D. Thomas, Ames Research Center, MS 202A-14, Moffett Field, CA 94035 (414) 694-6329 or FTS 448-6329

16 Abstract

A full-potential steady transonic wing flow solver has been modified so that freestream density and residual are captured in regions of constant velocity. This numerically precise freestream consistency is obtained by slightly altering the differencing scheme without affecting the implicit solution algorithm. The changes chiefly affect the fifteen metrics per grid point, which are computed once and stored. With this new method, the outer boundary condition is captured accurately, and the smoothness of the solution is especially improved near regions of grid discontinuity.

17 Key Words (Suggested by Author(s))	18 Distribution Statement
Consistent spatial differencing scheme Full-potential equation Free-stream density	Unlimited Subject category: 02

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of Pages	22 Price*
Uncl.	Uncl.	17	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document