NASA Technical Memorandum 110435

On Approximate Factorization Schemes for Solving the Full Potential Equation

Terry L. Holst

February 1997

National Aeronautics and
Space Administration

NASA Technical Memorandum 110435

On Approximate Factorization Schemes for Solving the Full Potential Equation

Terry L. Holst, Ames Research Center, Moffett Field, California

February 1997

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

ON APPROXIMATE FACTORIZATION SCHEMES FOR SOLVING THE FULL POTENTIAL EQUATION

Terry L. Holst[§]
Ames Research Center
Moffett Field, California 94035

SUMMARY

An approximate factorization scheme based on the AF2 algorithm is presented for solving the three-dimensional full potential equation for the transonic flow about isolated wings. Two spatial discretization variations are presented; one using a hybrid first-order/second-order-accurate scheme and the second using a fully second-order-accurate scheme. The present algorithm utilizes a C-H grid topology to map the flow field about the wing. One version of the AF2 iteration scheme is used on the upper wing surface and another slightly modified version is used on the lower surface. These two algorithm variations are then connected at the wing leading edge using a local iteration technique. The resulting scheme has improved linear stability characteristics and improved time-like damping characteristics relative to previous implementations of the AF2 algorithm. The presentation is highlighted with a grid refinement study and a number of numerical results.

INTRODUCTION

The long term objective of this research is to develop a chimera-based full potential flow solver which will be compatible with the well-established OVERFLOW Euler/Navier-Stokes flow solver developed at NASA Ames.[1] Thus, the user will have an option of which flow solver to use in the chimera-based zonal grid approach: full potential, Euler or Navier-Stokes. Of course, the full potential option will not be applicable for all applications, but for those that are applicable, the execution time should be up to two orders of magnitude less than for the Navier-Stokes formulation. Indeed, a chimera-based full potential solver should have modest execution times on even moderate-speed workstations. In a parametric study the bulk of the required computations could utilize the full potential approach and then a few selected conditions could be "checked" with a more complete, and thus more accurate, Euler or Navier-Stokes simulation. Such an approach would be extremely cost effective especially considering that all of these approaches would utilize the same problem setup and post processing software and to a large extent the same grid generation software. Applications of this new approach are quite numerous and include providing a fast mechanism for assessing wind tunnel wall and support interference effects associated with wind tunnel testing, or it could be used directly in the industrial preliminary design environment.

The specific goal of this report is to document recent advances in approximate factorization algorithms for solving the full potential equation that will eventually be useful in a multi-zone chimera environment. However, only single-grid-zone methodology will be presented herein. In particular, the presentation will focus on numerical solution of the transonic flow about an isolated wing utilizing a single-zone C-H-topology grid. For this grid all constant span stations on the wing exhibit the familiar C-grid topology, and all constant chord stations across the wing exhibit the equally familiar H-grid topology. Utilization of the chordwise C-grid topology is useful because it lends itself more readily to boundary layer correction implementation, which is future goal of this work.

In the present study the AF2 full potential algorithm developed in Refs. 2-3, which uses an O-H grid topology for isolated wing computations, is modified for the present C-H grid topology. Direct

---

§ Research Scientist, Advanced Computational Methods Branch.

application of the Ref. 2-3 AF2 iteration scheme on C-type grids is difficult because of a cell-aspect-ratio stability limitation, which exists at the wing surface and is greatly accentuated by the C-type grid topology. The nature of this instability, which was first described by South and Hafez,[4] will be briefly described in the iteration scheme section of this report. The present AF2 scheme variation is designed to control this instability and provides efficient and reliable convergence for a wide range of isolated-wing transonic flow simulations involving C-H topology grids.

This presentation begins with a discussion of the governing equation formulation followed by a detailed discussion of the numerical algorithm including spatial discretization scheme, boundary conditions, vortex sheet conditions, and the newly modified AF2 iteration scheme. Next, typical transonic wing computational results simulating the flow about the ONERA M6 Wing are presented. These results include a grid refinement study showing the levels of error in lift and drag relative to available Euler results. Finally, the presentation ends with concluding remarks and recommendations for future work.

GOVERNING EQUATION FORMULATION

The steady, three-dimensional, full potential equation written in strong conservation-law form is given by

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

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

where ρ is the fluid density; x, y, and z are Cartesian coordinates; γ is the ratio of specific heats; and ϕ is the full or exact velocity potential related to the velocity vector $\vec{q}$ by

$$\nabla\phi = \vec{q}$$

The velocity components can be expressed using Cartesian coordinates as follows:

$$\nabla\phi = \vec{q} = u\,\vec{i} + v\,\vec{j} + w\,\vec{k} = \phi_x\,\vec{i} + \phi_y\,\vec{j} + \phi_z\,\vec{k}$$

where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are the standard unit vectors in the x, y, and z directions, respectively. The mere existence of the velocity potential implies that the curl of the velocity vector must vanish. Thus, flows governed by the full potential equation must be irrotational. In addition, derivation of the density relation [Eq. (1b) above] requires the assumption of isentropic flow. In Eqs. (1) the density (ρ) and velocity components (ϕ_x, ϕ_y, and ϕ_z) are nondimensionalized by the stagnation density (ρ_s) and the critical speed of sound (a_*), respectively. Additional relations valid for these flow assumptions and this nondimensionalization include

Isentropic equation of state:

$$\frac{p}{\rho^\gamma} = \frac{\gamma+1}{2\gamma}$$

Bernoulli's equation:

$$\frac{q^2}{2}+\frac{a^2}{\gamma-1}=\frac{1}{2}\frac{\gamma+1}{\gamma-1}$$

Speed of sound definition:

$$a^2=\gamma\frac{p}{\rho}$$

where p is the fluid pressure, a is the local speed of sound, and q is the magnitude of the local fluid velocity.

To complete the above system, boundary conditions are required along all boundaries. Specifically, these boundaries fall into three categories: freestream, symmetry planes, and geometric surfaces. The freestream boundary condition, simply stated, is given by

$$x^2+y^2+z^2\to\infty, \qquad \phi\to\phi_\infty$$

where ϕ_∞ is the freestream distribution of the velocity potential, usually uniform flow. The latter two boundary conditions, symmetry planes and geometric surfaces, are both treated in the same manner, i.e., with a flow tangency assumption given by

$$\vec{q}\bullet\vec{n}=0$$

where $\vec{n}$ is a unit vector normal to the geometry of interest. More on boundary conditions including numerical implementation will be presented in the Numerical Approach section.

Equations (1) express mass conservation for flows that are isentropic and irrotational. Despite these limiting assumptions, the full potential formulation can be used in a shock-capturing context providing the shock waves are weak. The corresponding shock-jump conditions are valid approximations to the Rankine-Hugoniot shock jump conditions (derived from the Euler equations) for many applications. The key parameter in this situation is the normal component of the Mach number just upstream of the shock wave in question, which must remain below about 1.3 for the full potential formulation to be a reasonable approximation to the Euler equations. This is well within the scope of many transonic flow applications and includes the cruise conditions for most transonic transport aircraft. More discussion on this point including a comparison of the Euler and isentropic full potential shock polars is presented in Steger and Baldwin.[5]

Equations (1) are transformed from the physical domain (Cartesian coordinates) into the computational domain using a general, independent-variable transformation. This general transformation, indicated by

$$\begin{aligned}\xi&=\xi(x,y,z)\\ \eta&=\eta(x,y,z)\\ \zeta&=\zeta(x,y,z)\end{aligned} \qquad (2)$$

maintains the strong conservation-law form of Eqs. (1). The final transformed version of the full potential equation is given by

$$\left(\frac{\rho U}{J}\right)_\xi+\left(\frac{\rho V}{J}\right)_\eta+\left(\frac{\rho W}{J}\right)_\zeta=0 \qquad (3a)$$

where the density expression becomes

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}(U\phi_\xi + V\phi_\eta + W\phi_\zeta)\right]^{\frac{1}{\gamma - 1}} \tag{3b}$$

and

$$\begin{aligned}
U &= A_1\phi_\xi + A_4\phi_\eta + A_5\phi_\zeta \\
V &= A_4\phi_\xi + A_2\phi_\eta + A_6\phi_\zeta \\
W &= A_5\phi_\xi + A_6\phi_\eta + A_3\phi_\zeta \\
A_1 &= \nabla\xi \bullet \nabla\xi = \xi_x^2 + \xi_y^2 + \xi_z^2 \\
A_2 &= \nabla\eta \bullet \nabla\eta = \eta_x^2 + \eta_y^2 + \eta_z^2 \\
A_3 &= \nabla\zeta \bullet \nabla\zeta = \zeta_x^2 + \zeta_y^2 + \zeta_z^2 \\
A_4 &= \nabla\xi \bullet \nabla\eta = \xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z \\
A_5 &= \nabla\xi \bullet \nabla\zeta = \xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z \\
A_6 &= \nabla\eta \bullet \nabla\zeta = \eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z \\
J &= \xi_x\eta_y\zeta_z + \xi_y\eta_z\zeta_x + \xi_z\eta_x\zeta_y \\
&\quad - \xi_z\eta_y\zeta_x - \xi_y\eta_x\zeta_z - \xi_x\eta_z\zeta_y \\
&= (x_\xi y_\eta z_\zeta + x_\eta y_\zeta z_\xi + x_\zeta y_\xi z_\eta \\
&\quad - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi)^{-1}
\end{aligned} \tag{4}$$

In Eqs. (4), U, V, and W are the contravariant velocity components along the ξ, η, and ζ coordinate directions, respectively; $A_1 - A_6$ are metric quantities; and J is the determinant of the transformation Jacobian. The above full potential governing equation formulation can be used for general geometries in which the aerodynamic surface of interest is mapped to a constant coordinate line in the computational domain. This makes the flow-tangency boundary condition easy and accurate to implement. In transformed coordinates the flow-tangency boundary condition becomes (e.g., for an η=constant surface)

$$\begin{aligned}
&\vec{q} \bullet \nabla\eta = 0 \\
&\vec{q} \bullet \nabla\eta = \left(\phi_x \vec{i} + \phi_y \vec{j} + \phi_z \vec{k}\right) \bullet \left(\eta_x \vec{i} + \eta_y \vec{j} + \eta_z \vec{k}\right) = V \\
&V = 0
\end{aligned}$$

More simply stated, the contravariant velocity component in the η-direction V must vanish at the η=constant surface where flow tangency is required. The above flow tangency boundary condition is also used at y=0 as a symmetry plane boundary condition.

For all computations presented in this report the ξ coordinate is aligned with the C-grid wrap-around direction. Hence, it is aligned with the negative flow direction below the wing and with the positive flow direction above the wing. The η coordinate is aligned with the spanwise direction, and the ζ coordinate is in the wing normal-like direction. The orientation of each of these coordinate directions, especially the ξ coordinate, will have an influence on the construction of the iteration scheme as will be seen in the next section.

NUMERICAL APPROACH

Spatial differencing scheme

A second-order-accurate spatial differencing approximation to the full potential equation written in the general transformed coordinate system is given by

$$\overleftarrow{\delta}_\xi\left(\frac{\rho U}{J}\right)_{i+1/2,j,k}+\overleftarrow{\delta}_\eta\left(\frac{\rho V}{J}\right)_{i,j+1/2,k}+\overleftarrow{\delta}_\zeta\left(\frac{\rho W}{J}\right)_{i,j,k+1/2}=0 \tag{5}$$

where (for example)

$$\begin{aligned}U_{i+1/2,j,k}=&A_{1_{i+1/2,j,k}}(\phi_{i+1,j,k}-\phi_{i,j,k})\\&+\frac{1}{4}A_{4_{i+1/2,j,k}}(\phi_{i+1,j+1,k}-\phi_{i+1,j-1,k}+\phi_{i,j+1,k}-\phi_{i,j-1,k})\\&+\frac{1}{4}A_{5_{i+1/2,j,k}}(\phi_{i+1,j,k+1}-\phi_{i+1,j,k-1}+\phi_{i,j,k+1}-\phi_{i,j,k-1})\end{aligned} \tag{6}$$

The operators $(\overleftarrow{\delta}_\xi, \overleftarrow{\delta}_\eta, \overleftarrow{\delta}_\zeta)$ are standard backward difference operators in the three coordinate directions defined by

$$\begin{aligned}\overleftarrow{\delta}_\xi(\_)_{i,j,k}&=(\_)_{i,j,k}-(\_)_{i-1,j,k}\\\overleftarrow{\delta}_\eta(\_)_{i,j,k}&=(\_)_{i,j,k}-(\_)_{i,j-1,k}\\\overleftarrow{\delta}_\zeta(\_)_{i,j,k}&=(\_)_{i,j,k}-(\_)_{i,j,k-1}\end{aligned} \tag{7}$$

The i, j, and k subscripts used in the Eqs. (5)-(7) are used to denote position in the finite-difference grid such that $\xi = i\Delta\xi$, $\eta = j\Delta\eta$, $\zeta = k\Delta\zeta$. For convenience, the $\Delta\xi$, $\Delta\eta$ and $\Delta\zeta$ values are taken to be unity.

Values of ρ, $A_1 - A_6$, and J are computed using the freestream preserving spatial differencing scheme described by Flores et al.[6] and Thomas and Holst.[7] This scheme produces a zero residual for each interior grid cell with a freestream distribution of the velocity potential and generally produces a solution with improved accuracy, especially near grid singularities or in regions of rapid grid stretching.

At supersonic points the centrally-differenced, second-order-accurate spatial discretization scheme just presented must be upwind biased to maintain stable operation. This is accomplished by replacing the density in the ξ-flux computation with an upwind biased value of the density. Thus, the new spatial differencing scheme is given by

$$\overleftarrow{\delta}_\xi\left(\frac{\tilde{\rho} U}{J}\right)_{i+1/2,j,k}+\overleftarrow{\delta}_\eta\left(\frac{\rho V}{J}\right)_{i,j+1/2,k}+\overleftarrow{\delta}_\zeta\left(\frac{\rho W}{J}\right)_{i,j,k+1/2}=0 \tag{8}$$

where the density coefficient $\tilde{\rho}_{i+1/2,j,k}$ is defined by one of two options. The first option is given by

$$\tilde{\rho}_{i+1/2,j,k}=\rho_{i+1/2,j,k}-\nu_{i+1/2,j,k}(\rho_{i+1/2,j,k}-\rho_{i-1/2,j,k}) \tag{9}$$

where

$$\nu_{i+1/2,j,k} = \begin{cases} 2.46625(2\rho^* - \rho_{i+1/2,j,k} - \rho_{i-1/2,j,k})C & \text{if} \quad \rho_{i,j,k} \le \rho^* \\ 0 & \text{if} \quad \rho_{i,j,k} > \rho^* \end{cases} \tag{10}$$

The quantity ρ^* is a constant equal to the sonic value of the density, which for $\gamma = 1.4$, is $0.6339382\ldots$. The quantity C is a user specified coefficient usually set near a value of one, and $\nu_{i+1/2,j,k}$ is a switching parameter that controls the amount of upwinding that exists in the numerical scheme. Equation (10) is designed such that the value of $\nu_{i+1/2,j,k}$ will be zero at all subsonic grid points, i.e., a second-order-accurate, centrally-differenced scheme is retained for subsonic regions of the flow domain, and larger than zero at all supersonic grid points, i.e., a first-order-accurate upwind scheme is utilized for supersonic regions of flow. The supersonic branch of Eq. (10) approximates $(M^2_{i,j,k} - 1)C$, and thus, the amount of upwinding increases dramatically as the extent of supersonic flow increases. To keep the value of $\tilde{\rho}_{i+1/2,j,k}$ bounded by $\rho_{i+1/2,j,k}$ and $\rho_{i-1/2,j,k}$, the value of $\nu_{i+1/2,j,k}$ is constrained to be less than or equal to one.

The second option for defining the density coefficient ($\tilde{\rho}_{i+1/2,j,k}$) (inspired by Kinney et al.[8] and Jameson[9]), is given by

$$\begin{aligned} \tilde{\rho}_{i+1/2,j,k} = \rho_{i+1/2,j,k} - \nu_{i+1/2,j,k}[&\rho_{i+1/2,j,k} - \rho_{i-1/2,j,k} \\ &- \Psi_{i+1/2,j,k}(\rho_{i-1/2,j,k} - \rho_{i-3/2,j,k})] \end{aligned} \tag{11}$$

where the limiter Ψ is defined by

$$\Psi_{i+1/2,j,k} = \begin{cases} 1 - C_2\Delta & \text{if} \quad r_{i+1/2,j,k} \ge 0 \\ 0 & \text{if} \quad r_{i+1/2,j,k} < 0 \end{cases} \tag{12}$$

and $r_{i+1/2,j,k}$ is the ratio of successive density gradients defined by

$$r_{i+1/2,j,k} = \frac{\rho_{i+1/2,j,k} - \rho_{i-1/2,j,k}}{\rho_{i-1/2,j,k} - \rho_{i-3/2,j,k}} \tag{13}$$

In Eq. (12) the quantity C_2 is a constant typically set to a value near one and Δ is the local ξ-direction grid spacing which is approximated using

$$\Delta = (ITEU - ILE)^{-1}$$

where *ILE* is the ith grid index at the wing leading edge and *ITEU* is the ith grid index at the wing upper-surface trailing edge. To improve stability, especially for finer grids, the limiter function is decreased in magnitude with increasing distance away from the wing surface. A function of the form

$$\Psi_{i+1/2,j,k} = \Psi_{i+1/2,j,1}(C_3)^{-(k-1)} \tag{14}$$

where C_3 is a constant set to a value just above one, e.g., 1.08, and k is the normal-like ζ-direction grid index with k=1 defining the wing surface, has worked well for this purpose. The resulting scheme retains second-order-accuracy at the wing surface while allowing increased stability associated with increased dissipation away from the wing surface.

At subsonic grid points, the second density coefficient option (identical to the first option) produces a zero value of $\nu_{i+1/2,j,k}$, leading to a second-order-accurate, centrally-differenced, spatial-differencing scheme. At most supersonic points r will be greater than zero and the resulting spatial-difference scheme will be upwind-biased and second-order-accurate. At a supersonic point which is an extremem $r<0$. This produces a zero limiter function ($\Psi = 0$) at these points, which in turn produces the original first-order density upwinding option.

The lead truncation error term generated by the first density upwinding option at supersonic grid points can be approximately written as

$$\Delta\xi(\nu\rho_\xi\phi_\xi)_\xi$$

This expression is characteristically dissipative and leads to a first-order-accurate scheme. Larger values of this term, achieved by increasing the value of C in Eq. (10), will produce increased smearing of the solution in supersonic regions of flow.

The lead truncation error term generated by the second density upwinding option at supersonic grid points can be approximately written as

$$\Delta\xi^2(\nu\rho_{\xi\xi}\phi_\xi)_\xi + \Delta\xi C_2\Delta(\nu\rho_\xi\phi_\xi)_\xi$$

The first term is characteristically dispersive, and the second term is characteristically dissipative. However, providing $C_2\Delta$ approaches zero as $\Delta\xi$ approaches zero, both terms are second-order terms and the resulting scheme is second-order accurate.

The two supersonic spatial differencing schemes represented by Eqs. (8)-(10) and Eqs. (8), (10)-(14) are valid for supersonic flows which are approximately aligned with the positive ξ-coordinate direction. In practice, even for C-H topology wing grids involving large amounts of sweep, this type of supersonic upwinding is suitable. It is the only type of supersonic flow stabilization used in the present study. Nevertheless, generalization of the present scheme for arbitrary orientations of a curvilinear coordinate system is easy to accomplish. See Refs. 2-3 for examples involving a generalized form of the first-order density upwinding option.

Boundary conditions

Flow tangency and symmetry plane boundary conditions (as described above) require that the velocity component normal to the applicable boundary must vanish. For a general nonorthogonal mapping, such as that described by Eq. (2), the general condition for flow tangency requires the ζ contravariant velocity component at $\zeta = \zeta_{\min}$ (for example) be zero (i.e., $W_{\zeta=\zeta_{\min}} = 0$). This is implemented in the present study using a mass-flux reflection condition given by

$$\left(\frac{\rho W}{J}\right)_{i,j,1/2} = -\left(\frac{\rho W}{J}\right)_{i,j,3/2} \tag{15}$$

where in this case k=1 corresponds to the tangency or symmetry plane surface. In other expressions where ϕ_ζ is required at k=1 [e.g., in Eq. (6) or in the density computation at k=1], the $W_{\zeta=\zeta_{\min}} = 0$ boundary condition is used again to obtain

$$\phi_\zeta\Big|_{\zeta=\zeta_{\min}} = -\frac{A_5}{A_3}\phi_\xi - \frac{A_6}{A_3}\phi_\eta \tag{16}$$

Thus, a value of ϕ_ζ at the boundary condition surface can be obtained without using one-sided differences on the velocity potential.

At the wing-surface/symmetry-plane line of intersection, two contravariant velocity components must vanish (i.e., $V_{\eta=\eta_{min}} = 0$, $W_{\zeta=\zeta_{min}} = 0$). This is implemented using two mass-flux reflection conditions given by

$$\left(\frac{\rho V}{J}\right)_{i,1/2,1} = -\left(\frac{\rho V}{J}\right)_{i,3/2,1}$$
$$\left(\frac{\rho W}{J}\right)_{i,1,1/2} = -\left(\frac{\rho W}{J}\right)_{i,1,3/2} \qquad (17)$$

where in this case j=1, k=1 corresponds to the line of intersection. In other expressions where ϕ_η and ϕ_ζ are required at the line of intersection [e.g., in Eq. (6) or in the density computation at j=1, k=1], the $V_{\eta=\eta_{min}} = 0$, $W_{\zeta=\zeta_{min}} = 0$ boundary conditions are used again to obtain

$$\phi_\eta\Big|_{\text{intersection}} = \frac{A_5A_6 - A_3A_4}{A_2A_3 - A_6^2}\phi_\xi$$
$$\phi_\zeta\Big|_{\text{intersection}} = \frac{A_4A_6 - A_2A_5}{A_2A_3 - A_6^2}\phi_\xi \qquad (18)$$

Thus, values of ϕ_η and ϕ_ζ at the symmetry-plane/wing-surface line of intersection can be obtained without using one-sided differences on the velocity potential.

At the outer boundary a standard freestream boundary condition is used which is given by

$$\phi(x,y,z) = u_\infty x + w_\infty z$$

where u_∞ and w_∞ are standard Cartesian velocity components in the freestream associated with the x and z directions, respectively.

Vortex sheet conditions

For lifting computations involving the full potential formulation, circulation is accommodated with the usual vortex sheet or wake cut emanating downstream of the lifting surface (for wings, downstream of the wing trailing edge). The amount of lift or circulation is equal to the jump in velocity potential across the vortex sheet. The jump or discontinuity in velocity potential, as well as the double-stored characteristic of the vortex sheet, must be accounted for in the residual mass flux and density computational logic described above. For example, the U contravariant velocity component computation of Eq. (6) must be modified as follows at the vortex sheet

$$\begin{aligned} U_{i+1/2,j,1} = {} & A_{1_{i+v2,j,1}}(\phi_{i+1,j,1} - \phi_{i,j,1}) \\ & + \frac{1}{4}A_{4_{i+v2,j,1}}(\phi_{i+1,j+1,1} - \phi_{i+1,j-1,1} + \phi_{i,j+1,1} - \phi_{i,j-1,1}) \\ & + \frac{1}{4}A_{5_{i+v2,j,1}}(\phi_{i+1,j,2} - \phi_{NI-i,j,2} - \Gamma_j + \phi_{i,j,2} - \phi_{NI-i+1,j,2} - \Gamma_j) \end{aligned} \qquad (19)$$

where the $k = k_{min}$ ($k = 1$) boundary has been mapped to the wing/vortex-sheet surface using a C-grid topology. The point *i,j,1* is a typical point on the upper side of the vortex sheet and *NI-i+1,j,1* is the identical point on the lower side of the vortex sheet. The quantity Γ_j is the *j*th value of the circulation (where *j* is measured along the span of the wing) and *i=NI* corresponds to the maximum value of the *i*th coordinate. The above logic requires identical grid point distributions to exist on both the upper and lower wing/vortex-sheet surfaces with a grid point at the wing leading edge, thus, the *i*th point along the upper (or lower) vortex sheet exactly matches the *(NI-i+1)*th point along the lower (or upper) vortex sheet. The value of Γ_j is computed using

$$\Gamma_j^n = \phi_{ITEU,j,1}^n - \phi_{ITEL,j,1}^n \tag{20}$$

where *ITEU,j,1* corresponds to the *j*th location of the upper wing trailing edge and *ITEL,j,1* corresponds to the same point on the lower wing trailing edge, i.e., *ITEU=NI-ITEL+1*. The *j*th-location jump in velocity potential established at the wing trailing edge is maintained downstream along the vortex sheet, i.e.,

$$\phi_{i,j,1}^n = \phi_{NI-i+1,j,1}^n + \Gamma_j^n \quad i = ITEU+1, ITEU+2, \ldots, NI \tag{21}$$

Because of Eq. (21), only one velocity potential residual computation [using Eq. (8)] is required for each two grid nodes on the vortex sheet downstream of the wing trailing edge. This single residual value for each two vortex sheet points is computed by performing normal residual computations for both points and then averaging the values.

Iteration scheme

The iteration scheme utilized in the present study is called the AF2 scheme and was first introduced by Ballhaus and Steger[10] in 1975 for solving the transonic small-disturbance potential equation for two-dimensional applications.§ The present implementation is closely related to the AF2 scheme described in Holst and Thomas,[3] which was designed for solving the three-dimensional full potential equation for isolated-wing applications using an O-H grid topology (as mentioned in the Introduction). The present algorithm is also designed for solving isolated wing applications, but uses a C-H grid topology. The present implementation's C-type grid topology is more amenable to boundary layer correction implementation and is the major motivation for the present modification. More on the differences between the Ref. 2-3 AF2 scheme and the present version will be presented when the actual scheme is discussed.

A general iteration scheme for solving the full potential equation can be expressed as

$$NC_{i,j,k}^n + \omega L\phi_{i,j,k}^n = 0 \tag{22}$$

where the *n* superscript is an iteration index, $L\phi_{i,j,k}^n$ is the *n*th iteration residual at the *(i,j,k)*th position in the finite-difference mesh [spatial difference scheme for the full potential equation defined by Eq. (8)], *L* is the residual operator, ω is a relaxation parameter, $C_{i,j,k}^n$ is the correction defined by

$$C_{i,j,k}^n = \phi_{i,j,k}^{n+1} - \phi_{i,j,k}^n$$

§ The name AF2 (short for Approximate Factorization Scheme 2) was given to this scheme in Ref. 10 because it was the second scheme presented in that study.

and N is the "left-hand-side" operator that determines the type of iteration scheme. The AF2 iteration scheme used in the present formulation (one of two variations) can be specified using the following definition for N:

$$\alpha^2 NC^n_{i,j,k} = -\left(\alpha - \vec{\delta}_\xi R_i\right)\left[\left(\alpha - \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta\right)\left(\alpha - \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta\right) - \alpha^2 E_\xi^{-1}\right] C^n_{i,j,k} \qquad (23)$$

where α is an acceleration parameter (to be discussed shortly); R_i, S_j, and T_k are coefficients defined by

$$R_i = \left(\frac{\tilde{\rho} A_1}{J}\right)^n_{i-1/2,j,k}, \qquad S_j = \left(\frac{\rho A_2}{J}\right)^n_{i,j-1/2,k}, \qquad T_k = \left(\frac{\rho A_3}{J}\right)^n_{i,j,k-1/2} \qquad (24)$$

$\vec{\delta}_\xi, \vec{\delta}_\eta, \vec{\delta}_\zeta$ are standard forward-difference operators in the three coordinate directions defined by

$$\vec{\delta}_\xi (\_)_{i,j,k} = (\_)_{i+1,j,k} - (\_)_{i,j,k}$$

$$\vec{\delta}_\eta (\_)_{i,j,k} = (\_)_{i,j+1,k} - (\_)_{i,j,k} \qquad (25)$$

$$\vec{\delta}_\zeta (\_)_{i,j,k} = (\_)_{i,j,k+1} - (\_)_{i,j,k}$$

and E_ξ^{-1} is a shift operator defined by

$$E_\xi^{-1} (\_)^n_{i,j,k} = (\_)^n_{i-1,j,k}$$

A standard linearized von Neumann stability analysis of the interior AF2 scheme given by Eqs. (8), (22)-(23) yields a stable scheme providing $\alpha > 0$ and $0 \le \omega \le 2$. The quantity α behaves like the inverse of a time step in a typical time-dependent iteration, this implies that the linearized AF2 scheme is stable for any positive time step and hence is said to have unconditional linear stability.

The fact that α behaves like the inverse of the time step means that small values of α (large time steps) advance the solution rapidly and often cause (nonlinear) high frequency error growth. Large values of α (small time steps) advance the solution slowly and provide solution smoothing especially for the high frequency error components. Thus, a sequence of values for α produces optimal steady state convergence and can be obtained using

$$\alpha_k = \alpha_H \left(\frac{\alpha_L}{\alpha_H}\right)^{\frac{k-1}{M-1}} \qquad k = 1,2,\cdots,M \qquad (26)$$

where α_L and α_H are low- and high-frequency limits for the α_k parameter sequence, respectively, and M is the number of elements used in the α sequence (see Refs. 2-3 and 11 for more details and a variety of applications).

The AF2 factorization given by Eq. (23) is implemented in a three-sweep format, each involving a set of banded, scalar matrix inversions. These sweeps are given by

Sweep 1:

$$(\alpha - \vec{\delta}_\xi R_i) f^n_{i,j,k} = \alpha^2 \omega L \phi^n_{i,j,k} \tag{27a}$$

Sweep 2:

$$(\alpha - \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta) g^n_{j,k} = f^n_{i,j,k} + \alpha^2 C^n_{i-1,j,k} \tag{27b}$$

Sweep 3:

$$(\alpha - \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta) C^n_{i,j,k} = g^n_{j,k} \tag{27c}$$

In step 1, the f array is obtained from the residual by solving a simple bidiagonal matrix equation for each ξ line. The g array is then obtained from the f array by solving a tridiagonal matrix equation for each η line. Finally, the correction array is obtained in the third sweep from the g array by solving a tridiagonal matrix equation for each ζ line. Note that sweeps 2 and 3 can be performed together, i.e., immediately after the g values for the kth computational plane are obtained from sweep 2, values for the correction array can be obtained for the same kth computational plane from sweep 3. Thus, the g intermediate result array needs to be only two dimensional.

As can be seen from Eq. (23) or Eqs. (27) the left-hand-side difference operator in the ξ direction (the C-grid wrap-around direction in the present implementation) has been split between the first and second factors. This operator splitting is a telltale characteristic of the AF2 scheme. However, the operator splitting does not have to be in the first factor; any of the factors can be split. For example, the AF2 scheme of Ref. 3 splits the ζ direction factor. Like the present approach, the Ref. 3 implementation uses the ξ direction as the wrap-around direction, η is spanwise, and ζ is in the wing normal-like direction. However, unlike the present implementation, the Ref. 3 approach used an O-H grid topology.

Implementation of the AF2 scheme (like all implicit schemes) requires matrix "boundary conditions" for each sweep. In particular, conditions are required for f at $I = NI$, g at $j = 1$, g at $j = NJ$, C at $I = 1$, C at $k = 1$ and C at $k = NK$. The second and third of these conditions are associated with sweep 2 and are satisfied using a standard Dirchlet condition if the boundary is freestream or a standard Neumann condition if the boundary involves flow symmetry or flow tangency. The fifth and sixth conditions are associated with sweep 3 and likewise, are satisfied using a Dirchlet condition if the boundary is freestream or a Neumann condition if the boundary involves flow symmetry or flow tangency. The first and fourth conditions are nonstandard and result from the special AF2 left-hand-side operator splitting. These two special left-hand-side matrix "boundary conditions" are discussed next.

First, in sweep 1, a value of f at $I = NI$ is required for each value of j and k. A simple technique to satisfy this condition is given by

$$\vec{\delta}_\xi R_i f^n_{NI-1,j,k} = 0$$

This condition works, but a stability limitation at the i=NI boundary results, which is of the form[4]

$$\alpha \geq R_i \omega$$

where R_i is defined by Eq. (24) and ω is the relaxation factor from Eq. (22). Thus, there is a limit on the size of α, which can have a dramatic effect on convergence (even if the limitation exists only at the computational boundary). The key aspect of this stability limit is the A_1 / J contribution inside R_i, which behaves like a grid-cell aspect ratio (boundary tangential spacing over boundary normal spacing).

In the Ref. 3 AF2 implementation with the wing normal-like ζ operator split, the boundary stability limitation described above, exists at the wing surface. For the O-type grid in use for the Ref. 3 application, in which the cell aspect ratio is O(1) all around the inner boundary, this stability limitation has a minor consequence. However, direct application of the Ref. 3 AF2 iteration scheme for solving the full potential equation on C-type grids would be very inefficient because the cell aspect ratio for the inner boundary of a C-grid becomes quite large downstream of the wing trailing edge along the wake cut.

In the present approach the AF2 scheme is configured such that this approximate matrix inversion boundary condition is implemented away from the wing surface. In particular, it is implemented along the downstream outflow boundary where the flow is freestream and the cell aspect ratio is generally quite small.

The second special "boundary condition" for the left-hand side is associated with sweep 2. A value for the correction at $l = 1$ is required. If $i = 1$ represents an outer freestream boundary, then this condition is succinctly satisfied without loss of stability by using $C = 0$, i.e., the solution is required to be freestream along the $i = 1$ boundary with corrections for all iterations being zero. For the present AF2 implementation, as will be seen shortly, this condition must be satisfied along a surface of grid points that emanates from the wing leading edge that exactly divides the C-grid topology into two halves. The intermediate condition on the correction at this location used in the present scheme, involves a local iteration to ensure implicitness and will be subsequently discussed in detail.

An additional special condition is required at the vortex sheet downstream of the wing trailing edge. Values of the correction array are discontinuous across the vortex sheet during the iteration process due to the changing value of the circulation, namely

$$C^n_{i,j,1} - C^n_{NI-i+1,j,1} = \Gamma^{n+1}_j - \Gamma^n_j$$

where $k = 1$ corresponds to the wing/vortex-sheet-cut grid surface and Γ^n_j is the nth iterate of the circulation at the jth spanwise station. In the above expression the i subscript must be taken along the upper vortex sheet, and thus, NI-i+1 automatically is along the lower vortex sheet. This discontinuity must be taken into account when the sweep 3 matrix coefficients are computed downstream of the wing trailing edge.

Some comments about the AF2 scheme given by Eq. (23) are in order. First of all, the role played by each of the terms in this factorization can be understood more clearly by multiplying out each of the factors, which yields

$$\begin{aligned}\alpha^2 NC^n_{i,j,k} = -\Big(& \alpha^3 - \alpha^2 \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta - \alpha^2 \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta + \alpha \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta \\ & - \alpha^3 E_\xi^{-1} - \alpha^2 \vec{\delta}_\xi R_i + \alpha \vec{\delta}_\xi R_i \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta + \alpha \vec{\delta}_\xi R_i \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta \\ & - \vec{\delta}_\xi R_i \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta + \alpha^2 \vec{\delta}_\xi R_i E_\xi^{-1} \Big) C^n_{i,j,k}\end{aligned} \qquad (28)$$

Terms 1 and 5 when combined become

$$\alpha^3 - \alpha^3 E_\xi^{-1} = \alpha^3 \left(1 - E_\xi^{-1}\right) = \alpha^3 \overleftarrow{\delta}_\xi$$

which when operating on the correction produces a $\phi_{\xi t}$-like term that provides implicit (i.e., left-hand-side) or time-like dissipation to the iteration process in supersonic regions of flow.§ Existence of this term in the AF2 factorization is the chief reason that the AF2 scheme enjoys a convergence efficiency advantage over more traditional ADI-type algorithms for solving the full potential equation for transonic flows with shock waves.[12] This term provides favorable implicit dissipation in supersonic regions of flow providing the ξ coordinate is approximately aligned with the local streamwise direction "s" and providing the backward difference corresponds to an upwind difference. In other words, the $\phi_{\xi t}$-term must approximate an upwind-evaluated ϕ_{st}-term in supersonic regions of the flow. See Jameson[13] for more information on this point.

The $\phi_{\xi t}$ term arising from the Eq. (23) factorization is a direct result of having the ξ-direction factor split between sweeps 1 and 2. In contrast, the Ref. 3 AF2 scheme had the ζ-direction difference split between these two sweeps. Thus, the $\phi_{\xi t}$ term (or equivalently the ϕ_{st}- term) required for a convergent iteration in the Ref. 3 AF2 scheme had to be added as an extra term to the appropriate left-hand-side factor when the flow became supersonic.

In summary, the present scheme's ξ direction operator splitting is more efficient for two reasons. First, it requires fewer operations because no additional time-like dissipation terms have to be added in the ξ-direction. Second and most important, the boundary stability limitation resulting from the AF2 operator splitting is effectively removed. The present AF2 factorization with a ξ-direction splitting, represents a significant improvement over past AF2 approaches for solving the full potential equation on C-topology grids.

Terms 6 and 10 from Eq. (28) become

$$-\alpha^2 \vec{\delta}_\xi R_i + \alpha^2 \vec{\delta}_\xi R_i E_\xi^{-1} = -\alpha^2 \vec{\delta}_\xi R_i \left(1 - E_\xi^{-1}\right) = -\alpha^2 \vec{\delta}_\xi R_i \overleftarrow{\delta}_\xi$$

which, assuming the cross derivative metrics A_4 and A_5 are small, is a close approximation to the first-difference-operator term of Eq. (8). In the limit as an orthogonal grid is approached, i.e., as A_4 and A_5 go to zero, the approximation becomes exact.

This term coupled with terms 2 and 3 from Eq. (28) provides an approximation to the entire right-hand-side residual operator and can be written as

$$-\alpha^2 \left(\vec{\delta}_\xi R_i \overleftarrow{\delta}_\xi + \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta + \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta \right) C^n_{i,j,k} \approx -\alpha^2 L C^n_{i,j,k}$$

where a strict equality exists in the special case of an orthogonal grid.

The remaining terms in Eq. (28) (terms 4, 7, 8, and 9) are factorization error terms (FET) which are driven to zero as the iteration proceeds. Thus, Eq. (23) can be written as

$$\alpha^2 N C^n_{i,j,k} \approx \alpha^2 L C^n_{i,j,k} - \alpha^3 \overleftarrow{\delta}_\xi C^n_{i,j,k} + FET \qquad (29)$$

---

§ The time-derivative in the $\phi_{\xi t}$ term is obtained by assuming the relaxation or iteration process to be an iteration in real time. Thus,

$$\alpha \overleftarrow{\delta}_\xi C^n_{i,j,k} = \Delta t^{-1} \left(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i-1,j,k} - \phi^n_{i,j,k} + \phi^n_{i-1,j,k} \right) \sim \phi_{\xi t}$$

which shows that the AF2 scheme left-hand-side N operator represents a close approximation to the right-hand-side residual operator. This is an indication of rapid convergence.

Because of the inherent direction on the $\phi_{\xi t}$ term differencing operator, i.e., always backwards, the AF2 scheme given by Eq. (23) is suitable in supersonic regions of flow only when the positive ξ direction and the flow direction are approximately aligned. For supersonic regions of flow where the negative ξ direction is approximately aligned with the flow direction, i.e., below the wing, the AF2 scheme given by

$$\alpha^2 NC^n_{i,j,k} = -\left(\alpha + \overleftarrow{\delta}_\xi R_{i+1}\right)\left[\left(\alpha - \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta\right)\left(\alpha - \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta\right) - \alpha^2 E^{+1}_\xi\right] C^n_{i,j,k} \quad (30)$$

is an appropriate alternative. Note that the only changes between Eq. (30) above and Eq. (23) are in the first factor and in the sign of the shift operator superscript. When the factors of Eq. (30) are multiplied out the resulting expression for the N operator is given by

$$\alpha^2 NC^n_{i,j,k} \approx \alpha^2 LC^n_{i,j,k} + \alpha^3 \vec{\delta}_\xi C^n_{i,j,k} + FET \quad (31)$$

Thus, the same level of approximation to the residual operator is obtained by the factorization of Eq. (30). The only significant difference is in the direction and sign of the $\phi_{\xi t}$ term which is now upwind (as desired) for all regions of flow in which the negative ξ coordinate is aligned with the positive flow direction.

The AF2 factorization of Eq. (23) is appropriate (for example) for solving the transonic flow on the upper surface of a wing using a "C" or "O" type grid topology, and the factorization given by Eq. (30) is appropriate for the lower surface. This assumes the grid lines are wrapped in a clockwise direction with the flow from left to right. The two schemes need only be "connected" to each other to allow for general wing (or for that matter, any lifting surface) computations. This is accomplished using the following algorithm written in a three-step format, which is intended for C-type grid topology applications about isolated wing geometries. The i subscript is assumed to be in the wrap-around direction, j is assumed to be spanwise, and k is assumed to be in the normal-like direction. Each of the indicated operations is to be performed for all values of j and k, but only for the values of i that are indicated.

Step 1 (Sweep 1):

$$\begin{aligned}
(\alpha - \vec{\delta}_\xi R_i) f^u_{i,j,k} &= \alpha^2 \omega L \phi^n_{i,j,k} && i = NI-1, NI-2, \cdots, ILE \\
(\alpha + \overleftarrow{\delta}_\xi R_{i+1}) f^l_{i,j,k} &= \alpha^2 \omega L \phi^n_{i,j,k} && i = 2,3,\cdots, ILE \\
f^n_{i,j,k} &= f^u_{i,j,k} && i = NI-1, NI-2, \cdots, ILE-1 \\
f^n_{i,j,k} &= f^l_{i,j,k} && i = 2,3,\cdots, ILE-1 \\
f^n_{i,j,k} &= \frac{1}{2}\left(f^u_{i,j,k} + f^l_{i,j,k}\right) && i = ILE
\end{aligned} \quad (32a)$$

Step 2 (Sweep 2/3 leading edge iteration, $m=1,2,\ldots,MAXIT$):

$$\left.\begin{aligned}(\alpha-\vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)g^{u,m}_{j,k} &= f^n_{i,j,k}+\alpha^2 C^{n,m-1}_{i-1,j,k}\\ (\alpha-\vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)g^{l,m}_{j,k} &= f^n_{i,j,k}+\alpha^2 C^{n,m-1}_{i+1,j,k}\\ (\alpha-\vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta)C^{u,m}_{i,j,k} &= g^{u,m}_{j,k}\\ (\alpha-\vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta)C^{l,m}_{i,j,k} &= g^{l,m}_{j,k}\\ C^{n,m}_{i,j,k} &= \frac{1}{2}\left(C^{u,m}_{i,j,k}+C^{l,m}_{i,j,k}\right)\end{aligned}\right\} \quad i=ILE$$

$$\left.\begin{aligned}C^n_{i,j,k} &= C^{n,m}_{i,j,k}\\ \phi^{n+1}_{i,j,k} &= \phi^n_{i,j,k}+C^n_{i,j,k}\end{aligned}\right\} \quad i=ILE, \quad \text{if } m=MAXIT$$

$$\left.\begin{aligned}(\alpha-\vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)g^{u,m}_{j,k} &= f^n_{i,j,k}+\alpha^2 C^{n,m}_{i-1,j,k}\\ (\alpha-\vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta)C^{n,m}_{i,j,k} &= g^{u,m}_{j,k}\end{aligned}\right\} \quad i=ILE+1$$

$$\left.\begin{aligned}(\alpha-\vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)g^{l,m}_{j,k} &= f^n_{i,j,k}+\alpha^2 C^{n,m}_{i+1,j,k}\\ (\alpha-\vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta)C^{n,m}_{i,j,k} &= g^{l,m}_{j,k}\end{aligned}\right\} \quad i=ILE-1 \qquad (32b)$$

$$\left.\begin{aligned}C^n_{i,j,k} &= C^{n,m}_{i,j,k}\\ \phi^{n+1}_{i,j,k} &= \phi^n_{i,j,k}+C^n_{i,j,k}\end{aligned}\right\} \quad i=ILE+1, ILE-1, \quad \text{if } m=MAXIT$$

Step 3 (Sweep 2/3 downstream of the leading edge):

$$\left.\begin{aligned}(\alpha-\vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)g^{u}_{j,k} &= f^n_{i,j,k}+\alpha^2 C^{n}_{i-1,j,k}\\ (\alpha-\vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta)C^{n}_{i,j,k} &= g^{u}_{j,k}\\ \phi^{n+1}_{i,j,k} &= \phi^n_{i,j,k}+C^n_{i,j,k}\end{aligned}\right\} \quad i=ILE+2, ILE+3, \cdots, NI-1$$

$$\left.\begin{aligned}(\alpha-\vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)g^{l}_{j,k} &= f^n_{i,j,k}+\alpha^2 C^{n}_{i+1,j,k}\\ (\alpha-\vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta)C^{n}_{i,j,k} &= g^{l}_{j,k}\\ \phi^{n+1}_{i,j,k} &= \phi^n_{i,j,k}+C^n_{i,j,k}\end{aligned}\right\} \quad i=ILE-2, ILE-3, \cdots, 2 \qquad (32c)$$

where $i=ILE$ corresponds to the wing leading edge, $i=NI$ corresponds to the maximum value of i, the superscripts u and l denote upper- and lower-wing-surface values, and the superscript m is an index

for the local iteration used to obtain the nth level solution at $i = ILE, ILE \pm 1$. The initial values of the correction at the leading edge used in the local iteration are simply assumed to be zero, i.e.,

$$C^{n,0}_{ILE,j,k} = 0$$

The quantity *MAXIT* is the maximum number of iterations used for the local leading edge iteration. A value between 10 and 20 is typically used. Use of larger values will not improve the global convergence rate.

In step 1 [Eq. (24a)], the f array is obtained by solving a simple bidiagonal matrix equation for each ξ line. The upper and lower surface computations are completely independent of each other. The two values of f obtained along the grid surface emanating from the wing leading edge (i=*ILE*,j,k) are averaged to produce a single unique value of f at this location. Next, as summarized in step 2, the correction array is obtained by local iteration at the wing leading edge.§ This is achieved by alternate implementations of sweeps 2 and 3 at $i = ILE, ILE \pm 1$. After the leading edge correction is obtained, the remaining corrections in the three-dimensional field are obtained by sweeping the sweep 2/3 combination away from the leading edge on both the upper and lower wing surfaces. At the wing trailing edge the upper and lower sweep 3 matrix inversions are combined into a single inversion, thus providing the maximum amount of implicitness for the iteration scheme downstream of the trailing edge.

Despite the improved time-like dissipation arrangement of the scheme just presented, additional dissipation is required for some cases. For swept wing cases the ξ-coordinate direction may not be closely enough aligned to the streamwise direction because of wing-induced spanwise velocity components. Convergence instabilities, especially in areas with large cell-aspect-ratios in the i-j computational plane and/or around the grid singularity along the wing leading edge extension outboard of the wing tip, may exist. To correct this situation additional time-like damping in the spanwise or η direction is required. This is easily and efficiently accomplished with the following N operator modification

$$\alpha^2 N C^n_{i,j,k} = -\left(\alpha - \vec{\delta}_\xi R_i\right)\left[\left(\alpha - \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta\right)\left(\alpha' - \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta\right) - \alpha^2 E_\xi^{-1}\right] C^n_{i,j,k} \quad (33)$$

where the α parameter in the third factor has been changed to α' which is defined to be

$$\alpha' = \alpha + \alpha_d \alpha_v(\xi, \eta, \zeta)$$

i.e., the sum of the original α parameter and a new quantity, $\alpha_d \alpha_v$. The quantity α_v is a three-dimensional distribution array determined by trial and error that is permanently set, and α_d is a user-specified constant that allows the time-like dissipation level to be increased or decreased. The quantity $\alpha_d \alpha_v$ effectively provides additional time-like dissipation to the iteration process and stabilizes many of the difficulties encountered for swept wing computations. It also serves to stabilize the leading edge region of the computation in the vicinity of the local iteration and the region outboard of the wing tip along $k = 1$ where the grid cell aspect ratios are very large. The increment in computational cost for this modification is negligible.

§Several versions of the leading edge local iteration algorithm have been implemented and produce about the same results in terms of convergence efficiency. For brevity only one is presented.

NUMERICAL RESULTS

Problem setup and grid arrangement

To evaluate the attributes of the transonic full potential algorithm just presented the familiar ONERA M6 wing geometry is chosen. As mentioned in the previous section, a C-H grid topology is utilized; C topology in the chordwise direction and H topology in the spanwise direction. All grids have been generated using the HYPGEN grid generation code[14] which uses a fast hyperbolic grid generation scheme described in Steger and Rizk[15] and Chan and Steger.[16] Best grid results are typically achieved (especially for the coarse grid cases) by generating finer grids than required and then taking only every second or third point as desired. This is especially important for the normal-like direction in order to keep grid lines from crossing during the grid marching process. For all of the results presented herein the grids were generated using triple the desired number of points in the normal-like direction and then reduced in size after the grid generation was completed by taking every third point. In addition, because of the large amount of numerical dissipation required for the grid generation process when marching large distances from the initial data surface, grid skewness develops in the grid downstream of the wing trailing edge. To partially alleviate this situation, the grid point distribution downstream of the wing trailing edge along each ξ-coordinate line is redistributed using cubic spline interpolation. Despite this extra computational work, grid generation times range from a few seconds to a few tens of seconds on a Cray C-90 single processor computer.

Figures 1-5 show selected views of a typical grid about the ONERA M6 wing generated using HYPGEN. The grid displayed is a relatively fine grid consisting of 452,925 total points, with 225 points in the wrap-around direction, 61 in the spanwise direction, and 33 in the normal-like direction. Figure 1 shows a blowup of the wing surface grid in planform including a portion of the grid outboard of the wing tip and downstream of the wing trailing edge. In this figure only every third point in the wrap-around direction and every second point in the spanwise direction are displayed. Clustering in the wrap-around direction at the wing leading and trailing edges and in the spanwise direction at the wing tip are clearly visible. Figure 2 shows a close up view of the grid along a constant-chord surface near mid-chord. In this figure only every second point in the spanwise direction and every second point in the normal-like direction are displayed. This view shows the approximate treatment at the wing tip.

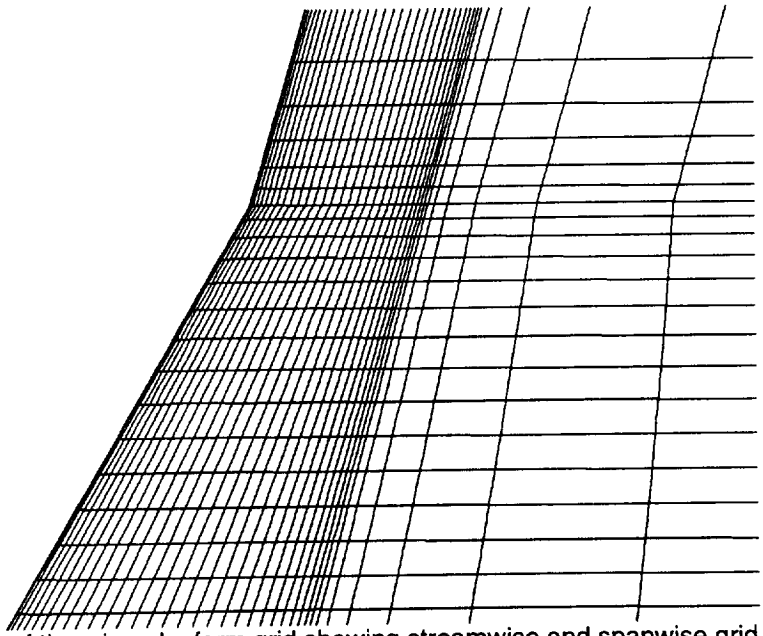


Fig. 1 Blow-up view of the wing planform grid showing streamwise and spanwise grid clustering in the vicinity of the wing (225X61X33=452,925 points). Only every third point in the wrap-around direction and every second point in the spanwise direction are displayed.

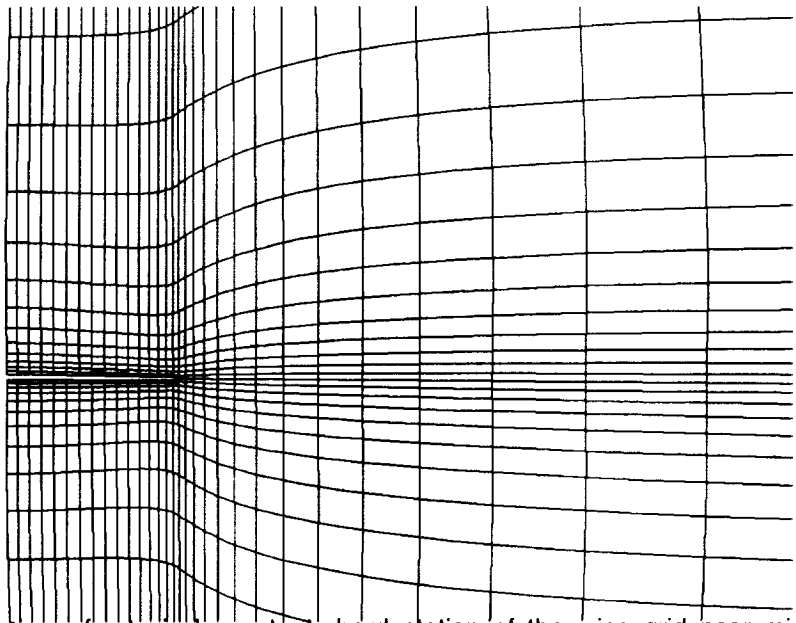


Fig. 2 Blow-up view of a typical constant-chord station of the wing grid near mid-chord showing selected grid lines in the normal-like and spanwise directions (225X61X33=452,925 points). Only every second point in the spanwise direction and every second point in the normal-like direction are displayed.

Figures 3 and 4 show two views of the wing in the wing-root cross-sectional plane; Fig. 3 shows a blow-up of the wing cross-section with surrounding grid and Fig. 4 shows an extreme blow-up of the same view. For clarity only every third point in the wrap-around direction and every second point in the normal-like direction are displayed in Fig. 3 while all points are displayed in Fig. 4. The grid is not orthogonal (especially downstream of the trailing edge) for the reason mentioned above. For all cases presented in this study the outer grid boundary is placed approximately 12 root chords above, below, upstream, and downstream of the wing and approximately 5.5 root chords outboard of the wing tip. An outer boundary position study conducted for the present approach and presented in Ref. 17, showed these distances to be acceptable for typical transonic wing computations.

The normal and streamwise tangential grid spacing on the wing surface are approximately controlled in the present grid generation approach. For the grid of Fig. 4 (as well as all grids used in the present study) the streamwise tangential spacings at the wing leading and trailing edges are set to 0.3 and 0.5 times the average tangential spacing, respectively. A smooth distribution of tangential grid spacing values in between the leading and trailing edges is obtained by numerically solving a fourth-order bi-harmonic differential equation for the tangential grid distribution. Use of such an approach allows easy specification of not only the location but also the spacing at both endpoints of this distribution. The resulting distribution is then interpolated onto the wing surface using cubic spline interpolation. The normal spacing all around the wing surface is set equal to the average tangential spacing. This value is automatically reduced somewhat at the wing leading edge to better match the streamwise grid clustering and to account for the inevitable flow gradients that exist in this area. Thus, the cells all around the wing surface are (on the average) approximately square.

Figure 5 shows a similar view as that of Fig. 4 except the constant-span station is outboard of the wing tip. Thus, the grid singularity outboard of the wing tip at the wing-extension leading edge is clearly visible. Across the $k = 1$ boundary at this span station flow periodicity is applied, i.e., all flow variables on the wing extension surface at grid point i must equal the corresponding flow variables on this surface at $NI-i+1$.

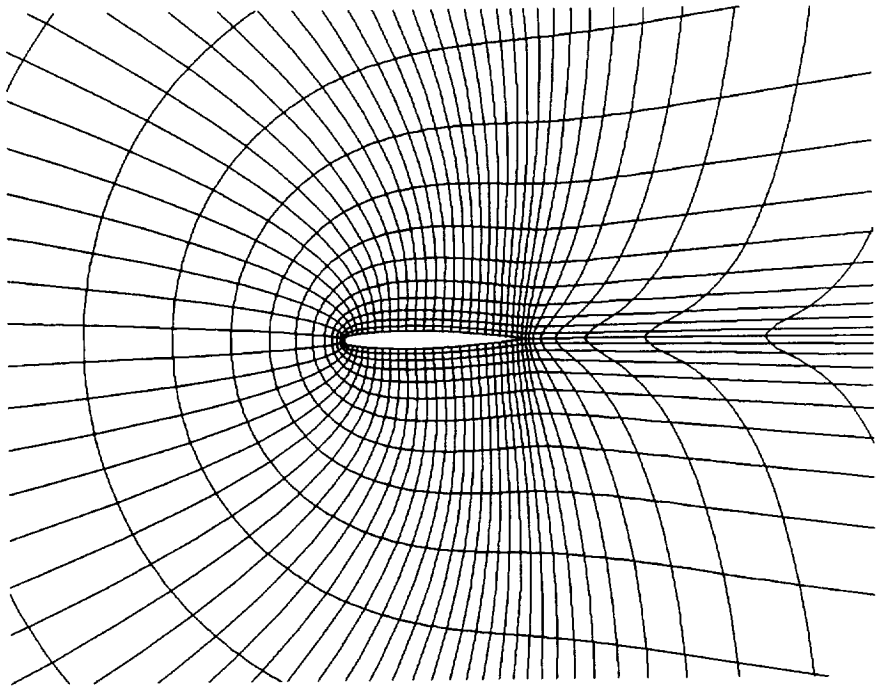


Fig. 3 Blow-up view of the wing root grid showing grid-cell clustering around the wing surface (225X61X33=452,925 points). Only every third point in the wrap-around direction and every second point in the normal-like direction are displayed.

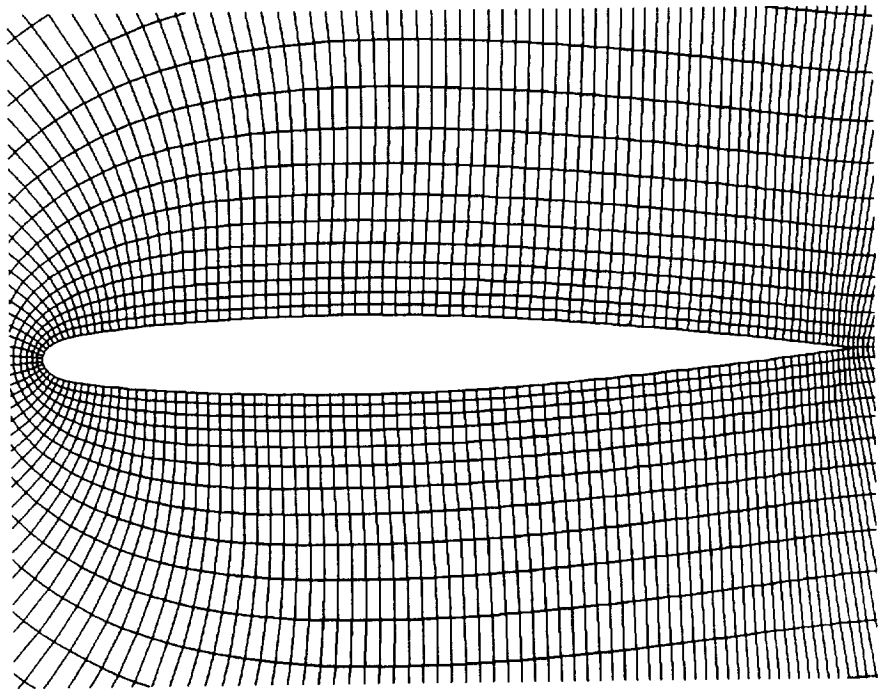


Fig. 4 Extreme blow-up view of the wing root grid showing grid-cell clustering around the wing surface (225X61X33=452,925 points). All grid points are displayed.

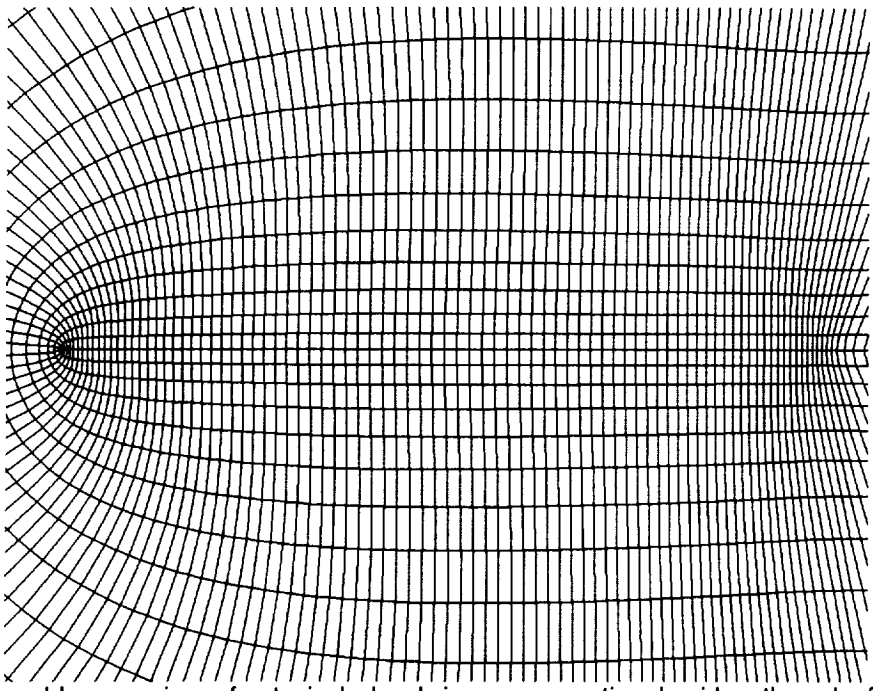


Fig. 5 Extreme blow-up view of a typical chordwise cross-sectional grid outboard of the wing tip showing grid-cell clustering around the wing extension and the grid singularity at the wing-extension leading edge (225X61X33=452,925 points). All grid points are displayed.

One last point about the grid displayed in Figs. 1-5 is in order. These grid views, as well as the Mach number contour plots to be shown subsequently, have been generated using PLOT3D[18]. As such, a special Euler-like *Q*-vector solution must be generated from the velocity potential solution. This is straight-forwardly accomplished from the stored values of density and from special velocity component values computed from the velocity potential. Because the density values are computed and stored at half points in the ξ-coordinate direction (i.e., at *i+1/2,j,k*), the *Q*-vector values are also computed and stored at these same half points. Using half points in the ξ-coordinate direction is also important because this allows use of the smallest computational stencil in the streamwise direction, which reduces smearing caused by the *Q*-vector post-processing computation. Likewise, the PLOT3D grid coordinate file is recomputed using simple second-order averages and is stored at these same half points in the ξ direction so as to be consistent with the *Q*-vector file. Thus, the grids displayed in Figs. 1-5 have been moved a small distance from where they have actually been used. This is why a grid space exists at the leading edge in Figs. 3-5 instead of a grid line as required by the leading edge flow-solver iteration scheme described in Eq. 32.

Hybrid and Second-Order Scheme Comparisons

In this section differences in accuracy between the hybrid spatial differencing scheme and the second-order-accurate spatial differencing scheme are examined. Keep in mind that the hybrid scheme is a second-order-accurate, centrally-differenced scheme at all subsonic points and a first-order-accurate, upwind-differenced scheme at all supersonic points. The first-order aspect in supersonic regions of flow is exclusively tied to the ξ-coordinate direction. The η- and ζ-coordinate directions are always second-order accurate and centrally differenced in both subsonic and supersonic regions of flow. For typical weak-shock transonic applications (amenable to simulation using the full potential formulation), the number of supersonic points is usually around 2-4% of the total number of points. Thus, even though the hybrid scheme does have first-order-accurate regions, most of the solution behaves as if it were second-order accurate. Nevertheless, differences caused by these "first-order pockets of flow" can be important and are now discussed.

Chordwise pressure coefficient distributions over the ONERA M6 Wing for both the hybrid and second-order-accurate schemes (as well as experimental results from Ref. 19) are compared in Fig. 6. These comparisons are presented at four semi-span stations: $2y/b$=0.2, 0.44, 0.65, and 0.95. Both computed results utilize the grid presented in Figs. 1-5. The flow conditions for this standard case ($M_\infty = 0.84, \alpha = 3.06°$), produce a rather benign flow field with a moderate-to-weak shock wave system with only a slight amount of shock/boundary-layer interaction. As can be seen in Fig. 6 the two numerical sclutions are in excellent agreement at all subsonic locations of the flow, i.e., on the entire lower wing surface and on the upper wing surface at and aft of the normal shock wave. In these regions the computational/experimental agreement is also generally quite good, except at the normal shock wave where some disagreement exists primarily due to viscous effects.§ Agreement between the two computational results in the supersonic region (as expected) is not as good. Generally, the forward shock, which is a supersonic-to-supersonic oblique shock, is "smeared out" by the hybrid scheme as exemplified in Figs. 6b and 6c. The second-order scheme does a better job in this region capturing the weak oblique shock wave in about the right location in Fig. 6b and about 6-7% of chord downstream of the experimental location in Fig. 6c. This second-order oblique shock location in Fig. 6c (2y/b=0.65) is only about 2% of chord downstream of the Euler results of Refs. 20-24.

Effect of C, C_2, and C_3 on Solution Accuracy

In this section the effect of various parameters associated with the hybrid and second-order spatial-discretization schemes is examined. These parameters are essentially fixed from case to case, and thus, are not meant to change as user-specified coefficients. However, it is of interest to see the effect their variation has on the solution to better appreciate the differences between the hybrid and second-order schemes. In all comparisons the $2y/b$ = 0.65 station from the ONERA M6 Wing at $M_\infty = 0.84, \alpha = 3.06°$ will be used in this evaluation. This is an appropriate station because the sensitive supersonic-to-supersonic shock is a prominent feature at this location on the wing.

Figure 7 shows a series of pressure coefficient comparisons for the hybrid scheme involving different values of the parameter C [defined by Eq. (10)]. Larger values of C produce larger amounts of first-order dissipation in the supersonic region of flow and smaller values produce smaller amounts of dissipation. Thus, as expected the largest value of C (C=1.7) completely smears the supersonic-to-supersonic shock wave and significantly "rounds out" the top of the normal shock. Conversely, the lowest value of C (C=0.5) does a somewhat reasonable job at the supersonic-to-supersonic shock but produces a rather significant overshoot at the normal shock. In addition, stability for the lowest value of C is marginal requiring about 1000 iterations for convergence (more than triple the number of iterations required for solution convergence using the default value of C=0.9). Values of C smaller than 0.5 are unstable for this case.

§Most of the computational/experimental disagreement at the normal shock wave (both in terms of shock position and strength) is caused by viscous effects. This qualitative conclusion is reached by comparing the present results with a variety of Euler results[20-24]. In these Euler-based computational studies the normal shock position and strength are in between the full potential and experimental results, but are more closely in agreement with the full potential results. That is, both types of inviscid computations produce shocks that are too strong and too far downstream relative to experiment. This overprediction of shock strength and position is a typical characteristic of transonic inviscid methods when used for computations that contain shock/boundary-layer viscous effects. The fact that the full potential results produce shocks which are somewhat stronger and further downstream of the Euler results implies that this additional (although generally smaller) error is a direct result of the isentropic/irrotational assumptions that are inherent in the full potential formulation.

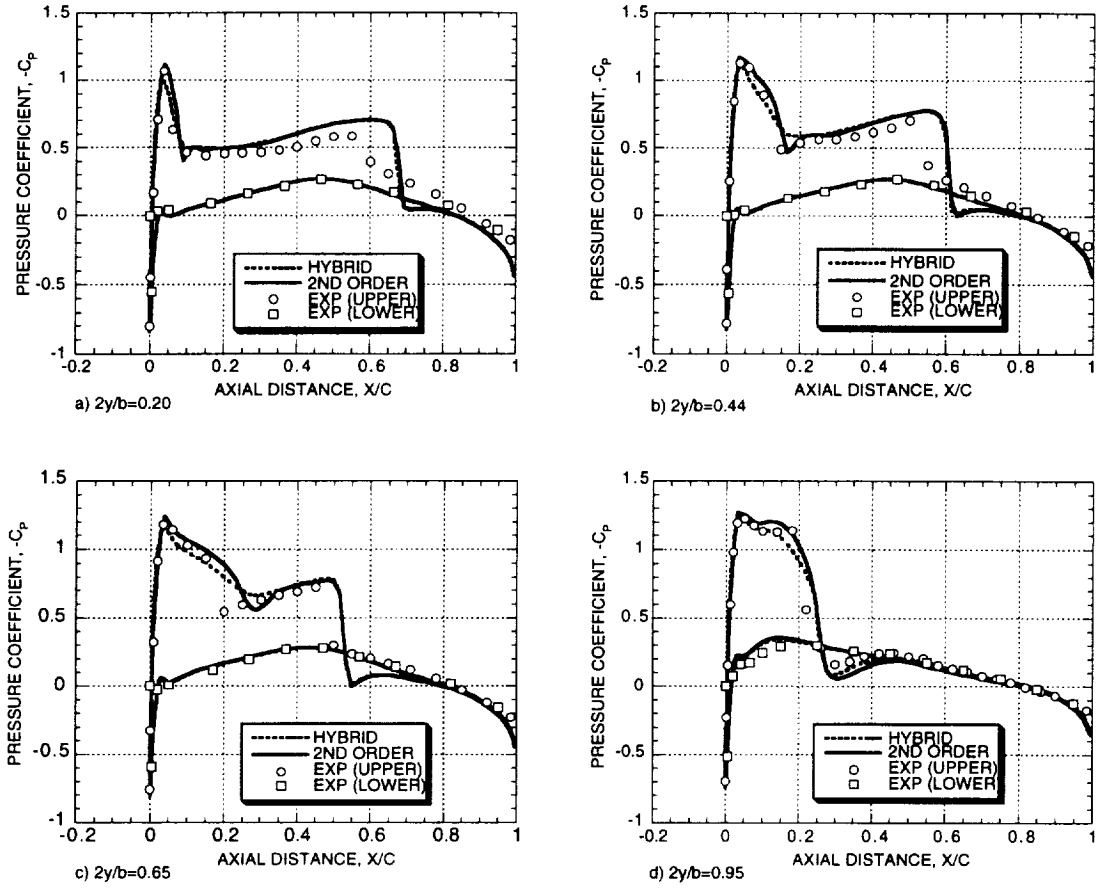


Fig. 6 Pressure coefficient comparisons at selected semi-span wing stations showing differences between the hybrid and second-order schemes, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$. Both computations utilized the same grid consisting of 225X61X33=452,925 points.

Figure 8 shows a series of pressure coefficient comparisons for the second-order scheme also involving different values of the parameter C. The supersonic domain pressure variations are not as dramatic for this case as compared to that of the hybrid scheme (Fig. 7). This is due to the small effect that the C parameter has in the second-order scheme, i.e., the term it multiplies is much smaller. Nevertheless, for smaller values of C the overshoot problem at the normal shock still exists. In addition, smaller values of C produce less stable convergence, and in particular, C=0.5 diverged for this case. For second-order cases the default value of C is 1.3. This improves convergence stability somewhat with a small sacrifice in accuracy.

Figures 9 and 10 present a similar set of pressure coefficient comparisons for the second-order scheme involving C_2 variations for Fig. 9 and C_3 variations for Fig. 10. These parameters are inherently associated with the second-order scheme only and are defined by Eqs. (12) and (14), respectively. The effect of C_2 on the supersonic domain pressure distribution as seen from Fig. 9 is quite small. Any value, including a very large value, produces acceptable results. The default for this parameter is 1.0. The effect of C_3 on the supersonic domain pressure distribution as seen from Fig. 10 is more pronounced. The smallest value (C_3=1.0), which eliminates any growth in the second-order dissipative term into the flow field interior [see Eq. (14)], produces a very strong supersonic-to-supersonic shock and an oscillatory solution between the two shocks. Slightly elevated values of C_3 eliminate this behavior. The default value of C_3 is 1.08.

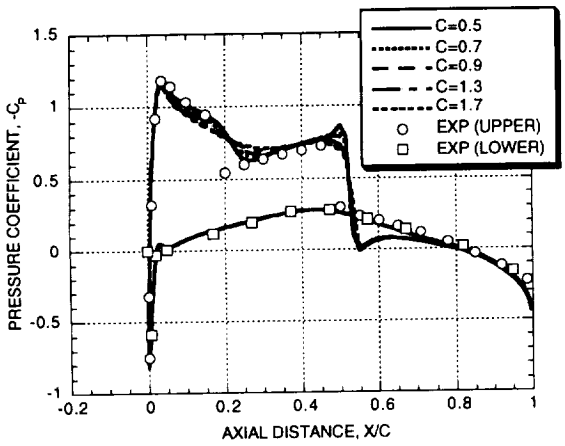


Fig. 7 Pressure coefficient comparisons at the $2y/b$=0.65 semi-span wing station showing the effect of C on solution accuracy (hybrid scheme), ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$, grid consists of 225X61X33=452,925 points.

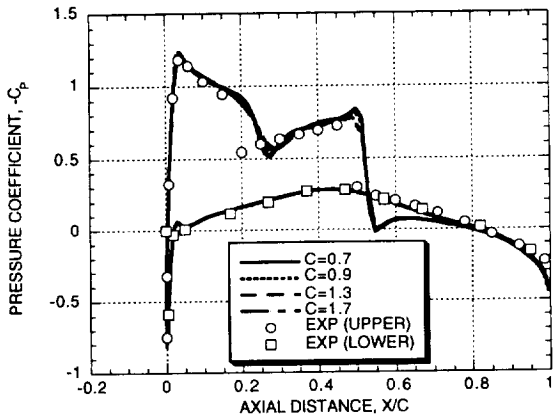


Fig. 8 Pressure coefficient comparisons at the $2y/b$=0.65 semi-span wing station showing the effect of C on solution accuracy (second-order scheme), C_2=1.0, C_3=1.08, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$, grid consists of 225X61X33=452,925 points.

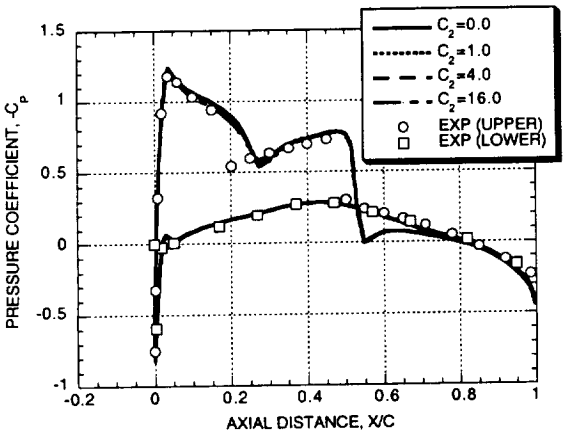


Fig. 9 Pressure coefficient comparisons at the $2y/b$=0.65 semi-span wing station showing the effect of C_2 on solution accuracy (second-order scheme), C=1.3, C_3=1.08, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$, grid consists of 225X61X33=452,925 points.

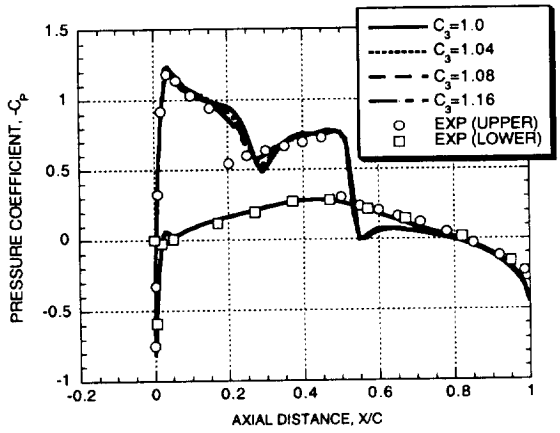


Fig. 10 Pressure coefficient comparisons at the $2y/b$=0.65 semi-span wing station showing the effect of C_3 on solution accuracy (second-order scheme), C=1.3, C_2=1.0, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$, grid consists of 225X61X33=452,925 points.

Grid Refinement Study

A grid refinement study for the ONERA M6 wing geometry utilized in the previous section is described next. Four grids, outlined in Table 1, are used for this purpose. In Table 1, NI is the number of grid points in the wrap-around direction, NJ is the number of points in the spanwise direction, and NK is the number of points in the normal-like direction. The total number of points ranges from just under 60,000 to over 875,000. The number of points defining the wing surface ranges from 1445 to 8651. The second-finest grid from Table 1 (case L4) is identical to the grid used in the last section and to the grid presented in Figs. 1-5..

Table 1. Summary of grid statistics for the grid refinement study.

CASE NO.	GRID DIMENSIONS			TOT SURF POINTS	TOTAL POINTS
	NI	NJ	NK		
L2	113	31	17	1445	59551
L3	169	46	25	3175	194350
L4	225	61	33	5577	452925
L5	281	76	41	8651	875596

Selected chordwise pressure coefficient distributions are presented in Fig. 11 at four different semi-span locations, $2y/b$=0.20, 0.44, 0.65 and 0.95. In each plot solutions for all four grids described in Table 1 are compared with experimental results from Ref. 19. As can be seen from Fig. 11 most of the solution variation affected by grid refinement is associated with the supersonic flow domain (all values of $-C_p$ above 0.3269). The shocks get sharper and the pressure minimums are more well defined as the grid is refined. The largest solution variation associated with the grid refinement process is at $2y/b$=0.95. This is probably a direct result of the approximate treatment of the wing tip geometry. The entire wing tip region between 2y/b=0.95 and 1.00 is represented by less than two grid cell widths in the span direction for the coarsest grid. Thus, as the grid is refined in this relatively high gradient region the solution improves remarkably.

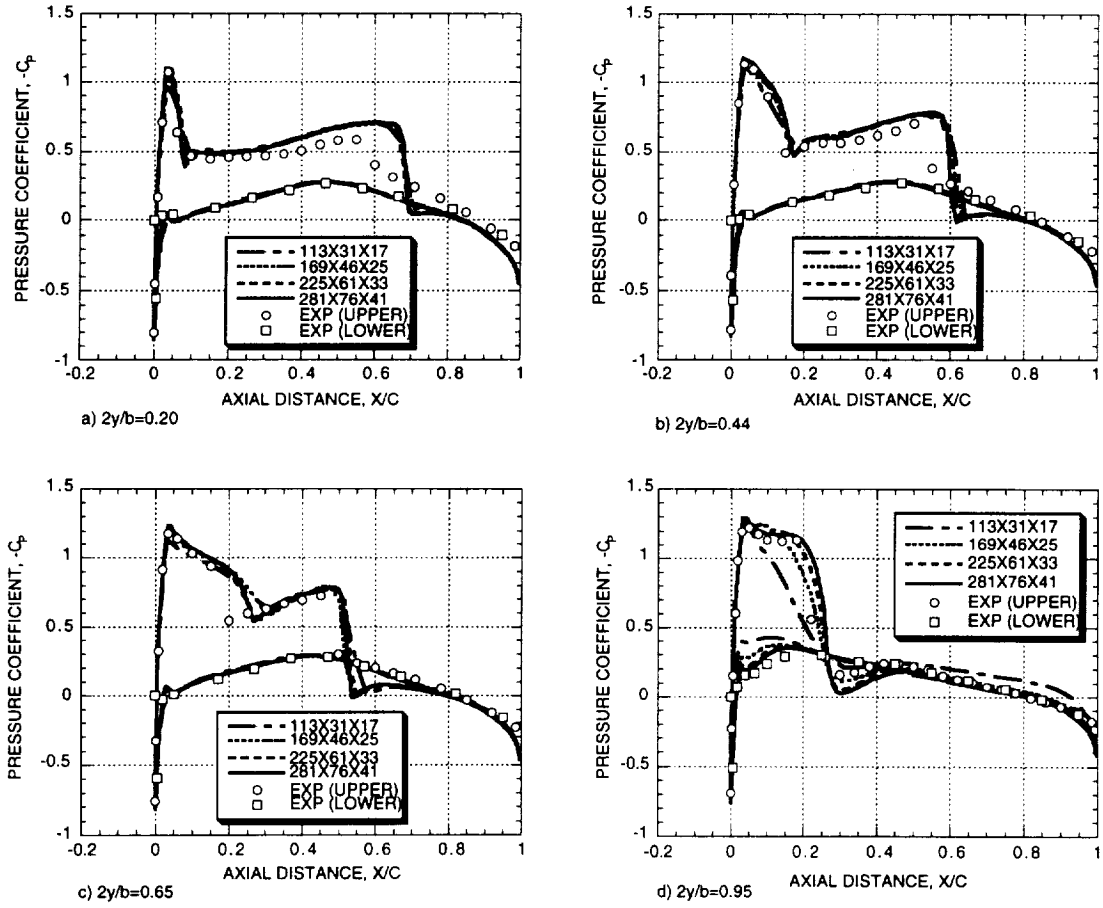
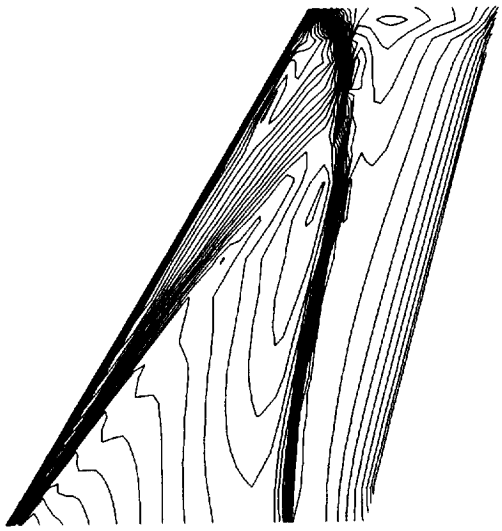
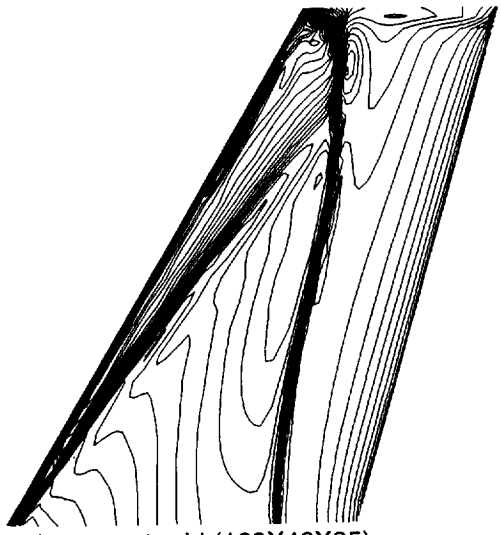


Fig. 11 Pressure coefficient comparisons at selected semi-span wing stations showing the effect of grid refinement (second-order scheme); ONERA M6 Wing, $M_\infty = 0.84$, $\alpha = 3.06°$.

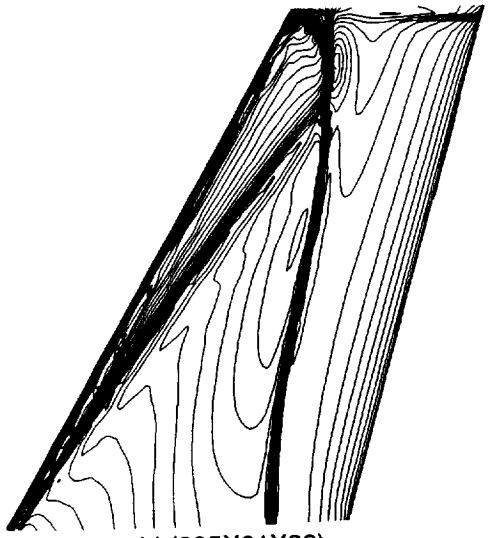
Another more qualitative technique for showing the effects of grid refinement on solution accuracy is shown in Fig. 12 where Mach number contours are displayed for the ONERA M6 Wing upper surface. These contours were generated using the PLOT3D plotting program[18] with contours plotted in 0.025 increments. A contour plot is displayed for each grid described in Table. 1. Figure 12a shows contours for the coarsest grid, Fig. 12b the second-coarsest grid, Fig. 12c the second-finest grid, and Fig. 12d shows contours for the finest grid. Evolution of the solution with grid refinement, in particular, the upper-surface shock system, is clearly evident from this sequence of plots. The supersonic-to-supersonic oblique shock is the most noticeable feature that forms and sharpens as the grid is refined. More on grid refinement will be presented in the last results section where lift and drag coefficient variations with grid refinement will be compared with Euler results.



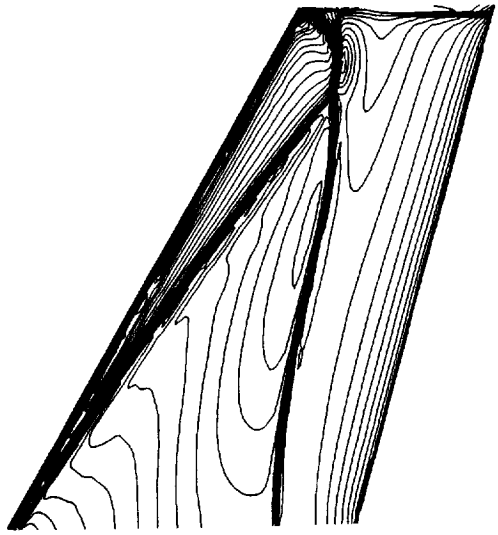
a) Coarse grid (113X31X17)



b) Second-coarsest grid (169X46X25)



c) Second-finest grid (225X61X33)



d) Fine grid (281X76X41)

Fig. 12 Mach number contours on the upper surface of the ONERA M6 Wing showing the effect of grid refinement (second-order scheme), $\Delta M = 0.025$; $M_\infty = 0.84, \alpha = 3.06°$.

Convergence Efficiency

Convergence history results are presented next showing the spatial discretization scheme's effect on convergence efficiency. In addition, the time-dissipation coefficient α_d [see Eq. (33)], which affects convergence stability and efficiency, but not the spatial accuracy, will also be explored. Figure 13 shows a plot of maximum residual, average residual and the lift coefficient as a function of iteration number for both the hybrid and second-order spatial differencing schemes. Both of these results used the L4 grid described in Table 1. In addition, the solution parameters [mainly α_L, see Eq. (26)] have been optimized by a trail-and-error process for both of these solutions. Each symbol in Fig. 13 represents 16 iterations in the convergence history for each solution. This corresponds to two complete applications of the α sequence, which for all cases presented herein contained eight elements, i.e., M=8.

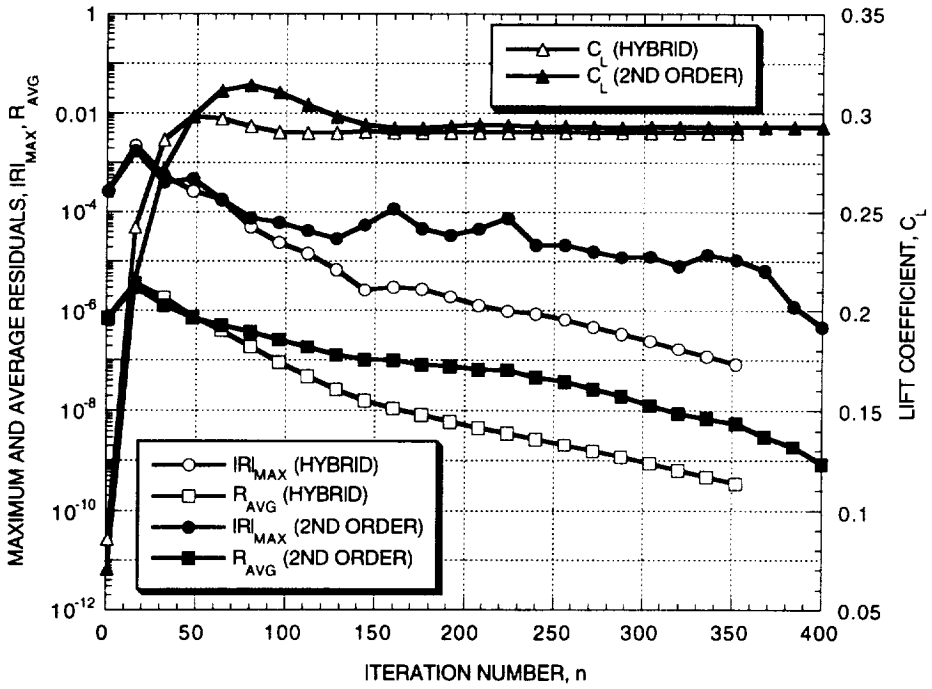


Fig. 13 Convergence history comparisons between the hybrid and second-order schemes, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$, grid consists of 225X61X33=452,925 points.

As can be seen from Fig. 13, the average residual is consistently about two orders of magnitude below the maximum residual. This characteristic allows a simple check for healthy convergence. When one or more points in the solution fail to converge smoothly (perhaps due to a localized grid problem) the difference between the maximum and average residuals will increase. The location of the maximum residual at this point in the convergence will then identify the region of difficulty. The average residual history curve is also somewhat smoother than the maximum residual curve, which (of course) is due to the high-frequency error content associated with the maximum residual. In fact, if values were plotted for every iteration, the average residual curve would have about the same smoothness, but the maximum residual curve would be much more oscillatory, peaking for each value of α_L as a consequence of high-frequency error growth. This high-frequency error growth is accompanied by a substantial reduction in the low-frequency error, and thus, is quite important for overall fast convergence. The lift convergence history (also shown in Fig. 13) epitomizes how the convergence process eliminates the low-frequency error content in the solution. Lift changes are small for values of α near α_H, but are quite large for values of α near α_L. Generally, a typical lift

convergence history curve from the present algorithm involves very rapid initial growth, a moderate overshoot and then a highly damped oscillation to the final answer.

Convergence histories for both solutions displayed in Fig. 13 posses the general characteristics described above. However, the hybrid scheme converges about 40 to 100% faster than the second-order scheme. This behavior is typical and is associated with increased high-frequency error components that exist in the second-order scheme because of reduced dissipation. For the specific convergence histories displayed in Fig. 13 the lift converges to within ±0.1% (a suitable condition for plotable accuracy) in 128 iterations for the hybrid scheme and in 240 iterations for the second-order scheme. These iteration counts correspond to approximately 23 and 43 sec of computer time, respectively, on a single processor Cray C-90. Again, for the convergence histories given in Fig. 13, the maximum residual is first reduced below a value of 10^{-7} (an alternate convergence criteria that represents tighter convergence than plottable accuracy) in 352 iterations for the hybrid scheme and in 496 iterations for the second-order scheme. Thus, depending on the convergence criteria (and many others can be used as well), the hybrid scheme is between 1.4 to 1.87 times faster than the second-order scheme for the case presented in Fig. 13.

The last topic for this section is to investigate and determine what effect the α_d parameter has on solution convergence. In this regard, Fig. 14 shows several convergence histories for cases with different values of α_d ranging from 0.4 to 3.0. All other solution parameters are fixed at default values. As can be seen $\alpha_d = 0.4$ produces an unstable result almost immediately. The other three values (0.8, 2.0, 3.0) all produce stable results with similar trends. Clearly, $\alpha_d = 3.0$ produces slower convergence than any other value. The two middle values of α_d (0.8 and 2.0) produce similar convergence histories with α_d=0.8 being somewhat faster for tighter convergence levels (especially if the maximum residual is used in the convergence criteria), and α_d=2.0 produces faster convergence for less tight levels of convergence. In terms of lift convergence, α_d=2.0 is the fastest producing convergence to 0.1% of the final lift value in 368 iterations; α_d=0.8 requires 384 iterations; and α_d=3.0 requires 480 iterations.

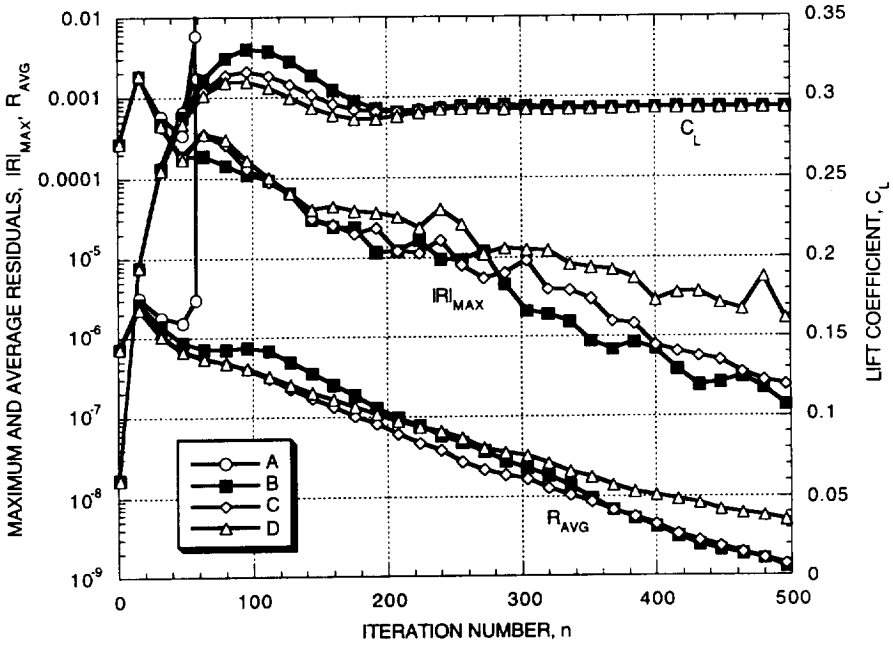


Fig. 14 Convergence history comparisons showing the effect of α_d on convergence (second-order scheme), ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$, grid consists of 225X61X33=452,925 points, A) $\alpha_d = 0.4$, B) $\alpha_d = 0.8$, C) $\alpha_d = 2.0$, D) $\alpha_d = 3.0$.

Lift and drag comparisons with Euler formulations

This section presents the effects of grid refinement on lift and drag for the standard case under study, i.e., the ONERA M6 Wing at $M_\infty = 0.84$, $\alpha = 3.06°$. In addition to results from the present full potential formulation, a number of other results collected from the literature are also included. The other-result formulations include both Euler and full potential methods using both structured and unstructured grids (Refs. 8, 20-25). In addition, there are newly computed full potential results (both hybrid and second-order) utilizing a multi-zone chimera variation of the present approach. In this approach the flow domain is divided into two grid zones: an inner C-H-topology grid surrounding the wing and an outer Cartesian-like grid that connects the inner grid to the far field using chimera interpolative boundary conditions. See Ref. 17 for more information about the full potential chimera grid scheme used in computing these results. The grids used in this series of full potential chimera computations are described in Table 2. The inner grid dimensions are given by NI1, NJ1, NK1, and the outer grid dimensions are given by NI2, NJ2, NK2. For each two-zone grid in Table 2 there is a single-zone grid in Table 1 with an exactly matching surface grid, both in terms of number and distribution of points. Thus, the surface solutions for these two sets of computations are comparable, despite the fact that the interior grid distribution/topology is quite different.

Table 2. Summary of grid statistics for the two-zone chimera grid refinement study.

CASE NO.	GRID DIMENSIONS NI1 NI2	NJ1 NJ2	NK1 NK2	TOTSURF POINTS	TOTAL POINTS
G2L2	101 37	23 19	9 26	1445	39185
G2L3	151 55	32 25	13 38	3175	115066
G2L4	201 73	41 33	17 50	5577	260547
G2L5	251 91	50 41	21 62	8651	494872

The variation of lift and drag with grid refinement for the above list of methods are displayed in Figs. 15 and 16, respectively. Values plotted along the horizontal axis are determined by taking the total number of wing surface nodes to the -1/2 power, which yields an approximate value for the average surface-cell grid spacing. Since surface quantities are being compared, it was decided to use this technique for computing the grid cell size. In this way structured and unstructured grid results can be compared.§ All results associated with open symbols utilize structured approaches, and all results associated with closed symbols utilize unstructured approaches. Results from Refs. 20-25 are Euler and from Ref. 8 (as well as the present results) are full potential. As can be seen from Fig. 15 the scatter is quite large. In fact, there is no clear cut single value for the asymptotic lift coefficient established by Fig. 15. A "best guess" range for the asymptotic lift coefficient limit is between 0.292 and 0.300. In actuality, the full potential formulation, because of its inherently isentropic nature which leads to the prediction of slightly stronger shock waves, should produce an asymptotic lift value that is slightly larger than the Euler value. However, it is difficult to reach such a conclusion from Fig. 15.

§ It is difficult to determine whether this technique for measuring grid refinement is appropriate or not for unstructured grid results or whether unstructured and structured grid results can be compared on this basis. Nevertheless, with this disclaimer, these comparisons are presented for the reader to evaluate.

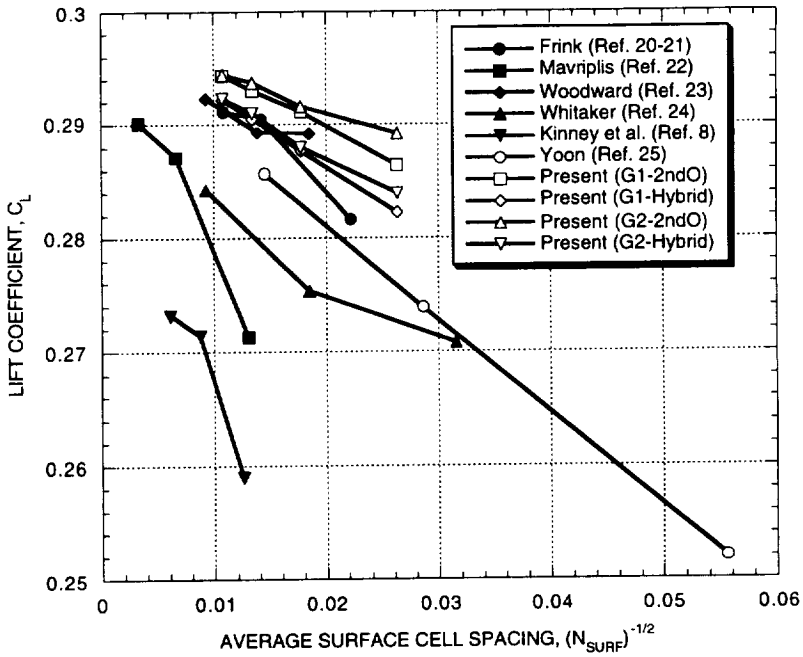


Fig. 15 Variation of lift coefficient with grid refinement for a variety of different approaches including both Euler and full potential methods, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$. Open symbols are structured methods and closed symbols are unstructured methods.

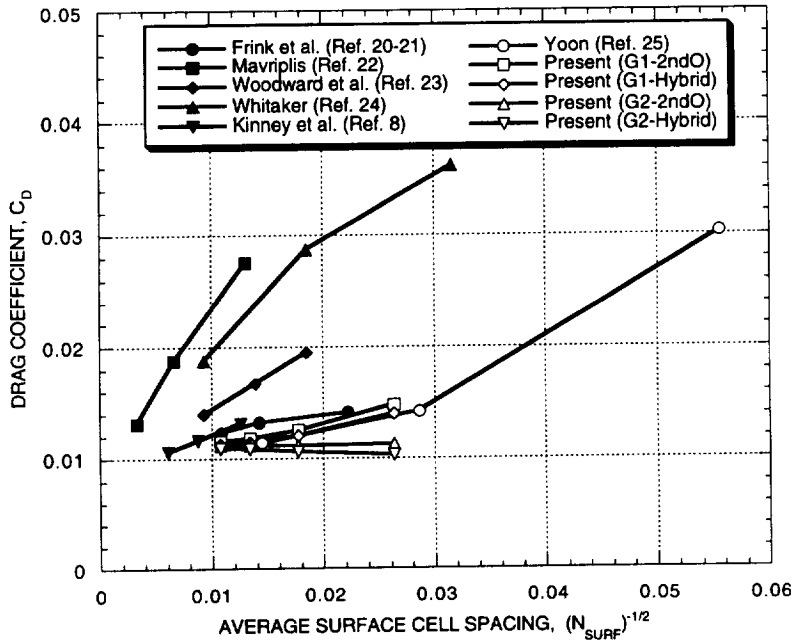


Fig. 16 Variation of inviscid drag coefficient with grid refinement for a variety of different approaches including both Euler and full potential methods, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$. Open symbols are structured methods and closed symbols are unstructured methods.

The variation of drag[§] with grid refinement for the above list of methods is displayed in Fig. 16. The same set of conventions as used in Fig. 15 are retained in Fig. 16. Again, the scatter is quite large; even larger than for the lift. The largest single-method variations in drag are from the unstructured grid results, which in some cases produce relatively large errors on moderately fine grids. For the drag results of Fig. 16, establishment of a definitive asymptotic value is even more difficult than for the lift. Nevertheless, a "best guess" range for the asymptotic drag coefficient limit is between 90 and 115 counts. The full potential asymptotic drag value, because of its stronger shock, should be slightly larger than the Euler value for this case. It is interesting to note that most of the individual full potential drag values are below most of the Euler values, but that the trends do seem to put the full potential drag asymptotic limit above the corresponding Euler value as expected. The two-zone full potential approach (G2 curves) produces generally more favorable drag results [relative to the single-grid results (G1 curves)] because the inner grid generation, also performed with HYPGEN, produces a better quality grid near the wing surface in the chimera approach.[§§]

A more quantitative picture of the present full potential lift, drag, and pitching moment results, including the newly computed two-zone chimera results generated with the Ref. 17 algorithm, is presented in Tables 3 and 4. Also included is a variety of convergence statistics for each case. No attempt has been made to optimize convergence for these cases (except for the L4 grid cases already presented in Fig. 13). All computer timings are from a single processor of the Cray C-90 computer using level 3 optimization for the inline, vector, and scalar options of the cf77 compiler. The scheme used for each computation is listed in the "SCH" column. An "H" corresponds to the hybrid scheme and a "2" corresponds to the second-order scheme. For the two-zone cases the first symbol corresponds to the scheme used for the inner grid zone and the second symbol corresponds to the scheme used for the outer Cartesian-like grid zone. Besides the lift, drag, and pitching moment information displayed in Tables 3 and 4, there are five additional quantities: the number of iterations required to achieve lift to within $\pm 0.1\%$ of the final value (n_{LIFT}), the computer time required for n_{LIFT} iterations (t_{LIFT}), the number of iterations required to achieve an average residual level of 10^{-8} (n_{RAVG}), the computer time required for n_{RAVG} iterations (t_{RAVG}), and the computer time required for solution overhead ($t_{O/H}$). The overhead time corresponds to grid generation, solution initialization including metric computation, and PLOT3D input file generation (a more difficult task for a potential solver than an Euler solver). The overhead time is larger for the single-zone results because the grid generation is more expensive. In the single-zone case the entire grid is generated using HYPGEN, and then (as mentioned previously) all points downstream of the wing trailing edge are redistributed to improve grid quality. In the two-zone case HYPGEN is used for only the inner grid zone (generally fewer than half the total points) without any point redistribution and the other Cartesian-like grid zone is generated using a very fast algebraic approach.

A definite degradation in solution convergence efficiency for the second-order scheme relative to the hybrid scheme can be seen in Tables 3 and 4 for most levels of grid refinement. This is true no matter which convergence criteria is chosen as was already established for the L4 grid in Fig. 13. The two-zone second-order results show less degradation relative to the two-zone hybrid results. This is primarily due to the fact that the outer grid zone in all of these cases actually used the hybrid spatial scheme. Using a two-zone chimera grid arrangement with the second-order scheme for the inner grid and the hybrid scheme for the outer grid is attractive because it is both efficient and accurate at least at the wing surface where the solution is essentially second order. Convergence efficiency comparisons between the two-zone and the single-zone results are mixed. Sometimes the single-zone results converge faster and sometimes the two-zone results are faster. One observation from this section is the speed with which the present full potential results are obtained. Solutions times

---

[§] It should be stressed that the drag being discussed here is inviscid drag consisting of wave drag and induced drag. No drag components due to viscous effects are included.

[§§] The grid quality near the wing surface for the two-zone chimera approach is better because HYPGEN doesn't have nearly as far to march to the outer boundary (one chord instead of 12). Thus, even though there are fewer normal-direction points in the two-zone inner grid relative to the single-zone grid, the stretching is less which contributes to better grid quality. In addition, because the marching distance is much smaller, less numerical smoothing for the grid generation process is required, again contributing to better grid quality.

range from 2 sec to less than 3 min. On a typical L4 grid for a two-zone case, each solution is obtained in 30-40 sec of computer time.

Table 3 Summary of computational statistics for the present single-grid full potential results displayed in Figs. 15 and 16, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$. Computer times are from a single processor Cray C-90 computer.

CASE NO.	SCH	C_L	C_D	C_M	n_{LIFT}	t_{LIFT} (sec)	n_{RAVG}	t_{RAVG} (sec)	$t_{O/H}$ (sec)
L2	H	.2822	.0140	-.1656	64	2	128	4	7
L3	H	.2877	.0120	-.1687	160	14	192	17	17
L4	H	.2905	.0114	-.1704	128	22	176	30	39
L5	H	.2922	.0113	-.1716	272	92	256	86	60
L2	2	.2864	.0148	-.1677	112	4	176	6	7
L3	2	.2911	.0125	-.1699	224	19	256	22	17
L4	2	.2930	.0118	-.1711	240	41	320	54	39
L5	2	.2943	.0115	-.1721	480	162	432	146	60

Table 4 Summary of computational statistics for the two-zone, chimera-grid full potential results displayed in Figs. 15 and 16, ONERA M6 Wing, $M_\infty = 0.84, \alpha = 3.06°$. These results use the chimera method presented in Ref. 17. Computer times are from a single processor Cray C-90 computer.

CASE NO.	SCH	C_L	C_D	C_M	n_{LIFT}	t_{LIFT} (sec)	n_{RAVG}	t_{RAVG} (sec)	$t_{O/H}$ (sec)
G2L2	H/H	.2839	.0103	-.1653	96	3	224	8	2
G2L3	H/H	.2880	.0106	-.1683	160	11	176	13	5
G2L4	H/H	.2910	.0108	-.1705	208	26	224	28	10
G2L5	H/H	.2923	.0109	-.1715	288	66	256	58	18
G2L2	2/H	.2892	.0113	-.1678	128	4	176	6	2
G2L3	2/H	.2916	.0111	-.1696	160	11	224	16	5
G2L4	2/H	.2937	.0112	-.1713	272	34	288	36	10
G2L5	2/H	.2944	.0112	-.1720	384	87	400	91	18

CONCLUDING REMARKS

In conclusion, a new scheme for solving the full potential equation has been presented and evaluated using a standard three-dimensional transonic wing computation. The new scheme includes both a hybrid spatial discretization option which is second-order accurate in subsonic regions of flow and first-order accurate in supersonic regions and an option which is fully second-order accurate irregardless of flow type. The new second-order scheme utilizes a solution limiter, somewhat similar to Euler flux limiters, to maintain stable operation at shock waves and other solution extrema.

The new iteration algorithm utilized in the present study is a variation of the AF2 scheme especially designed for obtaining transonic wing solutions on C-topology grids. Its special design removes (or at least controls) a stability limitation that exists in other implementations of this scheme. A key feature of the present scheme is the coupling of two different AF2-scheme variations, one above the wing and the other below the wing, using a local iteration at the wing leading edge. A typical transonic wing solution involving 450,000 point grids can be obtained in as little as 22 sec of computer time on a single processor of a Cray C-90 computer.

Numerous numerical results are presented including a grid refinement study to fully demonstrate the new scheme's capabilities. The results indicate that the new algorithm is a viable technique for solving the full potential equation and could provide a very fast computational tool for the aerodynamic analysis and design of geometrically complex aerodynamic configurations.

RECOMMENDATIONS FOR FURTHER STUDY

As stated in the introduction, the primary motivation for undertaking the present research path is to establish a full potential chimera capability, suitable for obtaining aerodynamic data for complex shapes. Thus, extending the present scheme's geometric handling capability is one area that needs additional work. In particular, development of a general methodology that is capable of handling multiple lifting surfaces with their associated vortex sheets is first on the list.

Another area of additional work is associated with viscous effects. In order to make the present capability suitable for realistic aerodynamic analysis, the inviscid flow assumption must be removed. Thus, a boundary layer correction capability must be added to the present methodology.

Application of the present capability to a variety of problems ranging from fast, nonlinear wind tunnel wall corrections to complex-geometry design applications is the next logical step for this research.

Finally, improvements in computational efficiency are required. Although not discussed in this report, the processing rate of the present algorithm is only marginally adequate, being about 300 MFLOPS on a single processor of a Cray C-90 computer when a fine grid is used. A variety of efforts in the parallel processing area could lead to dramatic improvement in this processing rate and a correspondingly dramatic improvement in the utility of this technology.

REFERENCES

1. Buning, P. G., Chan, W. M., Renze, K. J., Sondak, D. L., Chiu, I. T., Slotnick, J. P., "OVERFLOW User's Manual," Unpublished NASA report, Version 1.6ad, March 1993.

2. Holst, T. L., "Implicit Algorithm for the Conservative Transonic Full Potential Equation Using an Arbitrary Mesh," AIAA J., Vol. 17, No. 10, Oct. 1979, pp. 1038-1045.

3. Holst, T. L. and Thomas, S. D., "Numerical Solution of Transonic Wing Flow fields," AIAA J., Vol. 21, No. 6, June 1983, pp. 863-870.

4. South, J. C., Jr. and Hafez, M. M., "Stability Analysis of Intermediate Boundary Conditions in Approximate Factorization Schemes," AIAA Paper No. 83-1898, July 1983.

5. Steger, J. L. and Baldwin, B. S., "Shock Waves and Drag in the Numerical Calculation of Isentropic Transonic Flow," NASA TN D-6997, 1972.

6. 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, pp. 1027.

7. Thomas, S. D. and Holst, T. L., "A Consistent Spatial Differencing Scheme for the Transonic Full Potential Equation in Three Dimensions," NASA TM-86716, Dec. 1985.

8. Kinney, D. J., Hafez, M. M., and Gelhausen, P. A., "Validation of a New Unstructured Full Potential Formulation," AIAA Paper No. 95-1765-CP, June 1995.

9. Jameson, A., "Transonic Potential Flow Calculation Using Conservative Form," Proceedings of the AIAA 2nd Computational Fluid Dynamics Conference, June 1975, pp. 148-155.

10. Ballhaus, W. F., Jr. and Steger, J. L., "Implicit Approximate Factorization Schemes for the Low-Frequency Transonic Equation," NASA TM X-73,082, 1975.

11. Ballhaus, W. F., Jr., Jameson, A., and Albert, J., "Implicit Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems," AIAA J., Vol. 16, June 1978, pp. 573-579.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1997	Technical Memorandum

4. TITLE AND SUBTITLE

On Approximate Factorization Schemes for Solving the Full Potential Equation

5. FUNDING NUMBERS

522-31-12

6. AUTHOR(S)

Terry L. Holst

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER

A-975739

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-110435

11. SUPPLEMENTARY NOTES

Point of Contact: Terry L. Holst, Ames Research Center, MS T27B-1, Moffett Field, CA 94035-1000
(415) 604-6032

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified — Unlimited
Subject Category 02

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An approximate factorization scheme based on the AF2 algorithm is presented for solving the three-dimensional full potential equation for the transonic flow about isolated wings. Two spatial discretization variations are presented, one using a hybrid first-order/second-order-accurate scheme and the second using a fully second-order-accurate scheme. The present algorithm utilizes a C-H grid topology to map the flow field about the wing. One version of the AF2 iteration scheme is used on the upper wing surface and another slightly modified version is used on the lower surface. These two algorithm variations are then connected at the wing leading edge using a local iteration technique. The resulting scheme has improved linear stability characteristics and improved time-like damping characteristics relative to previous implementations of the AF2 algorithm. The presentation is highlighted with a grid refinement study and a number of numerical results.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Full potential, Numerical solution, Transonic	36
	16. PRICE CODE: A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

12. Holst, T. L. and Ballhaus, W. F., Jr., "Fast Conservative Schemes for the Full Potential Equation Applied to Transonic Flows," NASA TM 78469, 1978; See also, AIAA J., Vol. 17, No. 2, Feb. 1979, pp. 145-152.

13. Jameson, A., "Iterative Solutions of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1," Communications on Pure and Applied Mathematics, Vol. 27, 1974, pp. 283-309.

14. Chan, W. M., Chiu, I., and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791, October 1993.

15. Steger, J. L. and Rizk, Y. M., "Generation of Three-Dimensional Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations," NASA TM-86753, 1985.

16. Chan, W. M. and Steger, J. L., "Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme," Applied Mathematics and Computation, 51, pp. 181-205, 1992.

17. Holst, T. L., "Full Potential Equation Solutions Using a Chimera Grid Approach," Presented at the 14th AIAA Applied Aerodynamics Conference, New Orleans, LA, AIAA Paper No. 96-2423, June 1996.

18. Walatka, P. P., Buning, P. G., Pierce, L., and Elson, P. A., "PLOT3D User's Manual," NASA TM 101067, July 1992.

19. Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA M6-Wing at Transonic Mach Numbers," AGARD AR-138, May 1979.

20. Frink, N. T., "Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes," AIAA J., Vol. 30, No. 1, Jan. 1992, pp. 70-77.

21. Frink, N. T., Parikh, P., and Pirzadeh, S., "A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes," AIAA Paper No. 91-0102, Jan. 1991.

22. Mavriplis, D. J., "Three-Dimensional Unstructured Multigrid for the Euler Equations," AIAA Paper No. 91-1549-CP, 1991.

23. Woodward, P. R., Batina, J. T., and Yang, H. T., "Unstructured Mesh Quality Assessment and Upwind Euler Solution Algorithm Validation," J. of Aircraft, Vol. 31, No. 3, May-June 1994, pp. 644-650.

24. Whitaker, D. L., "Three-Dimensional Unstructured Euler Computations Using a Fully-Implicit, Upwind Method," AIAA Paper No. 93-3337-CP, June 1993.

25. Yoon, S., Private Communication, April 1996.