JOURNAL OF AIRCRAFT
Vol. 35, No. 3, May–June 1998

Multizone Chimera Algorithm for Solving the Full-Potential Equation

Terry L. Holst*
NASA Ames Research Center, Moffett Field, California 94035

A numerical scheme utilizing a chimera zonal grid approach for solving the three-dimensional full-potential equation is described. Within each grid zone a new approximate factorization scheme based on the approximation factorization scheme 2 algorithm is utilized to advance the solution one iteration. This is followed by the explicit advance of all common zonal grid boundaries using trilinear interpolation of the velocity potential. 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 presentation is highlighted with a grid refinement study and a number of transonic wing flowfield computations.

Introduction

THE long-term objective of this research is to develop a chimera-based full-potential flow solver that will be compatible with the well-established OVERFLOW Euler/Navier–Stokes chimera approach.[1,2] 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. 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 postprocessing 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.

Governing Equation Formulation

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

$$(\rho U/J)_\xi + (\rho V/J)_\eta + (\rho W/J)_\zeta = 0 \tag{1a}$$

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

where

$$U = A_1\phi_\xi + A_4\phi_\eta + A_5\phi_\zeta$$

$$V = A_4\phi_\xi + A_2\phi_\eta + A_6\phi_\zeta$$

$$W = A_5\phi_\xi + A_6\phi_\eta + A_3\phi_\zeta$$

$$A_1 = \nabla\xi\cdot\nabla\xi = \xi_x^2 + \xi_y^2 + \xi_z^2$$

$$A_2 = \nabla\eta\cdot\nabla\eta = \eta_x^2 + \eta_y^2 + \eta_z^2$$

$$A_3 = \nabla\zeta\cdot\nabla\zeta = \zeta_x^2 + \zeta_y^2 + \zeta_z^2$$

$$A_4 = \nabla\xi\cdot\nabla\eta = \xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z$$

$$A_5 = \nabla\xi\cdot\nabla\zeta = \xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z$$

$$A_6 = \nabla\eta\cdot\nabla\zeta = \eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z$$

$$J = \xi_x\eta_y\zeta_z + \xi_y\eta_z\zeta_x + \xi_z\eta_x\zeta_y - \xi_z\eta_y\zeta_x - \xi_y\eta_x\zeta_z - \xi_x\eta_z\zeta_y$$
$$= (x_\xi y_\eta z_\zeta + x_\eta y_\zeta z_\xi + x_\zeta y_\xi z_\eta - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi)^{-1}$$

In the preceding equations, ϕ is the full or exact velocity potential; ρ is the fluid density; ξ, η, and ζ are computational domain coordinates; γ is the ratio of specific heats; U, V, and W are the contravariant velocity components along ξ, η, and ζ, respectively; A_1–A_6 are metric quantities; and J is the determinant of the transformation Jacobian. In Eqs. (1) the density and all velocity components are nondimensionalized by the stagnation density and the critical speed of sound, respectively. To complete the preceding system, freestream or flow tangency boundary conditions are utilized along all boundaries. This full-potential governing equation formulation can be used for general geometries in which the aerodynamic surface of interest is mapped to a constant coordinate surface in the computational domain. This makes the flow-tangency boundary condition easy and accurate to implement. For a more complete description of these transformed equations see Refs. 3 and 4.

Equations (1) express mass conservation for flows that are isentropic and irrotational. Despite these limiting assumptions, the full-potential formulation can be used to capture shock waves, providing they are weak. The full-potential shock-jump conditions are valid approximations to the Euler shock-jump conditions providing the normal component of the Mach num-

Received March 17, 1997; revision received Nov. 14, 1997; accepted for publication Nov. 24, 1997.

*Research Scientist, Advanced Computational Methods Branch. Fellow AIAA.

ber just upstream of any captured shock wave is below about 1.3. 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 Ref. 5.

Numerical Approach

Spatial Differencing Scheme

A spatial differencing approximation to the full-potential equation suitable for both subsonic and supersonic regions of flow is given by

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

where $\overleftarrow{\delta}_\xi$, $\overleftarrow{\delta}_\eta$, and $\overleftarrow{\delta}_\zeta$ are standard backward-difference operators in the three coordinate directions; and $\tilde{\rho}$ is a special density coefficient that will be defined in the upcoming text. The i, j, and k subscripts in Eq. (2) are used to denote position in the finite difference grid along the ξ, η, and ζ coordinate directions, respectively. In Eq. (2) ρ, J, U, V, and W are all evaluated using the second-order-accurate, 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 improved accuracy, particularly near grid singularities or in regions of rapid grid stretching. At supersonic points the spatial discretization scheme just presented must be upwind biased to maintain stable operation. This is accomplished with the density coefficient $\tilde{\rho}_{i+1/2,j,k}$, which has two options. (The two supersonic spatial differencing options represented herein are valid for supersonic flows that 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 and 3 for examples involving a generalized form of the first-order density upwinding option.). The first option is given by

$$\tilde{\rho}_{i+1/2,j,k} = \rho_{i+1/2,j,k} - \nu_{i+1/2,j,k}(\rho_{i+1/2,j,k} - \rho_{i-1/2,j,k}) \quad (3)$$

where

$$\nu_{i+1/2,j,k} = \begin{cases} 2.46625(2\rho^* - \rho_{i+1/2,j,k} - \rho_{i-1/2,j,k})C & \text{if} \quad \rho_{i,j,k} \le \rho^* \\ 0 & \text{if} \quad \rho_{i,j,k} > \rho^* \end{cases} \quad (4)$$

The quantity ρ^* is a constant equal to the sonic value of the density, which for $\gamma = 1.4$, is 0.6339382 .... The quantity C is a user-specified coefficient (set to 0.9 for all results presented herein); and $\nu_{i+1/2,j,k}$ is a switching parameter that controls the amount of upwinding in the numerical scheme. Equation (4) 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 flow, 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. (4) 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

$$\tilde{\rho}_{i+1/2,j,k} = \rho_{i+1/2,j,k} - \nu_{i+1/2,j,k}[\rho_{i+1/2,j,k} - \rho_{i-1/2,j,k} - \Psi_{i+1/2,j,k}(\rho_{i-1/2,j,k} - \rho_{i-3/2,j,k})] \quad (5)$$

where the solution limiter Ψ is defined by

$$\Psi_{i+1/2,j,k} = \begin{cases} 1 - C_2\Delta & \text{if} \quad r_{i+1/2,j,k} \ge 0 \\ 0 & \text{if} \quad r_{i+1/2,j,k} < 0 \end{cases} \quad (6)$$

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

$$r_{i+1/2,j,k} = \frac{\rho_{i+1/2,j,k} - \rho_{i-1/2,j,k}}{\rho_{i-1/2,j,k} - \rho_{i-3/2,j,k}} \quad (7)$$

In Eq. (6) the quantity C_2 is a constant (set to 1.0 in the present study), and Δ is the average i-direction grid spacing on the upper wing surface normalized by chord. To improve stability, the limiter function is decreased in magnitude with increasing distance away from the wing surface using a function of the form

$$\Psi_{i+1/2,j,k} = \Psi_{i+1/2,j,1}(C_3)^{-(k-1)} \quad (8)$$

where C_3 is a constant (set to 1.08 in the present study) and $k = 1$ defines the wing surface. 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 option (just like the first) produces a zero value of $\nu_{i+1/2,j,k}$, leading to a second-order, centrally differenced scheme. At most supersonic points $r > 0$. The resulting spatial-difference scheme will be upwind-biased and second-order accurate. At a supersonic point, which is a local extrema, $r < 0$. This produces a zero limiter function, 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.

Boundary Conditions

Flow tangency and symmetry plane boundary conditions require that the velocity component normal to the applicable boundary must vanish. For a general nonorthogonal mapping, the general condition for flow tangency requires the appropriate nontangential contravariant velocity component to be zero. This is implemented in the present study using a mass-flux reflection condition. For lifting computations involving the

full-potential formulation, circulation is accommodated with the usual wake cut emanating 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 earlier. See Refs. 3 and 4 for details on boundary condition implementation.

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 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.] The present implementation is closely related to the AF2 scheme described in Holst and Thomas,[4] which was designed for solving the three-dimensional full-potential equation for isolated wing applications using an O–H grid topology. 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 a major motivation for the present modification. More on the differences of the AF2 scheme in Refs. 3 and 4 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

$$NC^n_{i,j,k} + \omega L\phi^n_{i,j,k} = 0 \tag{9}$$

where the n superscript is an iteration index; $L\phi^n_{i,j,k}$ is the nth iteration residual at i, j, and k [defined by Eq. (2)]; ω is a relaxation parameter; $C^n_{i,j,k}(=\phi^{n+1}_{i,j,k} - \phi^n_{i,j,k})$ is the correction at i, j, k; and N is the left-hand side (LHS) 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:

$$\alpha^2 NC^n_{i,j,k} = -(\alpha - \vec{\delta}_\xi R_i)[(\alpha - \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)(\alpha - \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta) - \alpha^2 E_\xi^{-1}]C^n_{i,j,k} \tag{10}$$

where α is an acceleration parameter (to be discussed shortly); $\vec{\delta}_\xi$, $\vec{\delta}_\eta$, and $\vec{\delta}_\zeta$ are standard forward-difference operators; E_ξ^{-1} is a shift operator defined by $E_\xi^{-1}(\_)^n_{i,j,k} = (\_)^n_{i-1,j,k}$; and R_i, S_j, and T_k are coefficients defined by

$$R_i = \left(\frac{\tilde{\rho}A_1}{J}\right)^n_{i-1/2,j,k}, \qquad S_j = \left(\frac{\rho A_2}{J}\right)^n_{i,j-1/2,k}, \qquad T_k = \left(\frac{\rho A_3}{J}\right)^n_{i,j,k-1/2} \tag{11}$$

A standard von Neumann stability analysis of the AF2 scheme yields a stable result 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 (even though the present scheme is not time accurate). 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, particularly for the high-frequency error components. Thus, a sequence of values for α produces optimal steady-state convergence and can be obtained using

$$\alpha_k = \alpha_H \left(\frac{\alpha_L}{\alpha_H}\right)^{(k-1)/(M-1)} \qquad k = 1, 2, \ldots, M \tag{12}$$

where α_L and α_H are low- and high-frequency limits, respectively; and M is the number of elements used in the sequence (see Refs. 3, 4, and 11 for more details and a variety of applications).

The AF2 factorization given by Eq. (10) is implemented in a three-sweep format, each involving a set of banded, scalar matrix inversions. These sweeps are given by

Sweep 1:

$$(\alpha - \vec{\delta}_\xi R_i) f^n_{i,j,k} = \alpha^2 \omega L\phi^n_{i,j,k} \tag{13a}$$

Sweep 2:

$$(\alpha - \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta) g^n_{j,k} = f^n_{i,j,k} + \alpha^2 C^n_{i-1,j,k} \tag{13b}$$

Sweep 3:

$$(\alpha - \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta) C^n_{i,j,k} = g^n_{j,k} \tag{13c}$$

In sweep 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 ith computational plane are obtained from sweep 2, values for the correction array can be obtained for the same ith computational plane from sweep 3. Thus, the g intermediate result array needs to be only two dimensional.

As shown in Eq. (10), the LHS difference operator in the ζ direction (the C-grid wraparound 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. 4 splits the ζ direction factor. Like the present approach, the Ref. 4 implementation uses the ξ direction as the wraparound direction, η is spanwise, and ζ is in the wing normal-like direction. However, unlike the present implementation, the Ref. 4 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, six of these matrix conditions are required in the present implementation and are enumerated as follows: 1) f at $i = NI$, 2) C at $i = 1$, 3) g at $j = 1$, 4) g at $j = NJ$, 5) C at $k = 1$, and 6) C at $k = NK$ (where NI, NJ, and NK are the maximum index values for i, j, and k, respectively). The third and fourth of these conditions are associated with sweep 2 and are satisfied using a standard Dirchlet condition if the boundary is freestream, or they are satisfied with 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 or Neumann condition. The first and second conditions are nonstandard, resulting from the special AF2 LHS operator splitting and 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[12]

$$\alpha \geq R_i \omega$$

where R_i is defined by Eq. (11), and ω is the relaxation factor from Eq. (9). 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 $i = NI$). The key aspect of this stability limit is a contribution inside R_i equal to $\nabla\xi \cdot \nabla\xi/J$, which behaves like a grid-cell aspect ratio (boundary tangential spacing over boundary normal spacing). If $R_i >> 1$ the restricted value of α that behaves like the time-step inverse will slow convergence. If $R_i << 1$ [or at most $R_i \approx \mathcal{O}(1)$] the restriction is small and efficient convergence is maintained. In the present approach the AF2 scheme [as presented in Eq. (13)] is configured such that this approximate matrix boundary condition is implemented at the downstream outflow boundary ($i = NI$), where R_i is generally quite small.

In the Ref. 4 AF2 implementation with the wing normal-like ζ operator split, the boundary stability limitation described earlier exists at the wing surface. For the O-type grid in use for the Ref. 4 application, in which the cell aspect ratio is $\mathcal{O}(1)$ all around the inner boundary, this stability limitation has a minor consequence. However, direct application of the Ref. 4 AF2 scheme 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 and, thus, the restriction on α would be quite large. The present AF2 implementation represents a significant improvement over the Ref. 4 AF2 implementation for C-grid applications.

The second special boundary condition for the LHS is associated with sweep 2. A value for the correction at $i = 1$ is required. If $i = 1$ represents an outer freestream boundary, then this condition is succinctly satisfied without loss of stability using $C = 0$, i.e., the solution is required to be freestream along the $i = 1$ boundary with zero corrections for all iterations. For the present C-grid AF2 implementation this condition is satisfied along a surface of grid points emanating from the wing leading edge exactly dividing the C-grid topology into two halves. The intermediate condition on the correction at this location involves a local iteration to ensure implicitness and will be discussed in the upcoming text.

Some comments about the AF2 scheme given by Eq. (10) 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 after simplification yields

$$\alpha^2 NC^n_{i,j,k} \approx \alpha^2(\vec{\delta}_\xi R_i \overleftarrow{\delta}_\xi + \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta + \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta) C^n_{i,j,k} - \alpha^3 \overleftarrow{\delta}_\xi C^n_{i,j,k} + FET \tag{14}$$

The first three terms are an approximation to the residual defined by Eq. (2), the fourth term is a $\phi_{\xi t}$-like term that provides time-like dissipation to the iteration process in supersonic regions of flow, and the set of terms indicated by *FET* are factorization error terms that are driven to zero as the iteration converges. Existence of the $\phi_{\xi t}$ term in the AF2 factorization is the chief reason that the AF2 scheme enjoys a convergence efficiency advantage over more traditional alternating direction implicit-type algorithms for solving the full-potential equation for transonic flows with shock waves.[13] This term provides favorable time-like 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[14] for more information on this point.

The $\phi_{\xi t}$ term arising from the Eq. (10) factorization is a direct result of having the ξ-direction factor split between sweeps 1 and 2. In contrast, the Ref. 4 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. 4 AF2 scheme had to be added as an extra term to the appropriate LHS 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.

Because of the inherent direction on the $\phi_{\xi t}$ term-differencing operator, i.e., always backwards, the AF2 scheme given by Eq. (10) 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

$$\alpha^2 NC^n_{i,j,k} = -(\alpha + \overleftarrow{\delta}_\xi R_{i+1})[(\alpha - \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta)(\alpha - \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta) - \alpha^2 E^{+1}_\xi] C^n_{i,j,k} \tag{15}$$

is an appropriate alternative. Note that the only changes between Eqs. (15) and (10) are in the first factor and in the sign of the shift operator superscript. When the factors of Eq. (15) are multiplied out the resulting expression for the N operator is given by

$$\alpha^2 NC^n_{i,j,k} \approx \alpha^2(\vec{\delta}_\xi R_i \overleftarrow{\delta}_\xi + \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta + \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta) C^n_{i,j,k} + \alpha^3 \vec{\delta}_\xi C^n_{i,j,k} + FET \tag{16}$$

Thus, the same level of approximation to the residual operator is obtained. 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. (10) 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. (15) 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 wraparound 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):

$$\begin{aligned}
&(\alpha - \vec{\delta}_\xi R_i) f^u_{i,j,k} = \alpha^2 \omega L \phi^n_{i,j,k} \quad i = NI - 1, NI - 2, \ldots, ILE \\
&(\alpha + \overleftarrow{\delta}_\xi R_{i+1}) f^l_{i,j,k} = \alpha^2 \omega L \phi^n_{i,j,k} \quad i = 2, 3, \ldots, ILE \\
&f^n_{i,j,k} = f^u_{i,j,k} \quad i = NI - 1, NI - 2, \ldots, ILE - 1 \\
&f^n_{i,j,k} = f^l_{i,j,k} \quad i = 2, 3, \ldots, ILE - 1 \\
&f^n_{i,j,k} = \tfrac{1}{2}(f^u_{i,j,k} + f^l_{i,j,k}) \quad i = ILE
\end{aligned} \tag{17a}$$

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

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

$i = ILE$

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

$i = ILE$ if $m = MAXIT$

(17b)

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

$i = ILE + 1$

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

$i = ILE - 1$

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

$i = ILE + 1, \quad ILE - 1$ if $m = MAXIT$

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

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

$i = ILE + 2, \quad ILE + 3, \ldots, NI - 1$

(17c)

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

$i = ILE - 2, \quad ILE - 3, \ldots, 2$

where $i = ILE$ corresponds to the wing leading edge; 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 of 20 is typically used. Use of larger values will not improve the global convergence rate.

In step 1 [Eq. (17a)], 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 ($I = ILE$) 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. (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.) 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.

Chimera-Induced Iteration Scheme Modifications

For chimera multizone applications the iteration scheme described earlier [e.g., by Eq. (13)] must be modified by including the traditional IBLANK array. This is accomplished as follows.

Sweep 1:

$$(\alpha - IB_{i,j,k} \vec{\delta}_\xi R_i) f^n_{i,j,k} = IB_{i,j,k} \alpha^2 \omega L \phi^n_{i,j,k} \tag{18a}$$

Sweep 2:

$$(\alpha - IB_{i,j,k} \vec{\delta}_\eta S_j \overleftarrow{\delta}_\eta) g^n_{j,k} = f^n_{i,j,k} + IB_{i,j,k} \alpha^2 C^n_{i-1,j,k} \tag{18b}$$

Sweep 3:

$$(\alpha - IB_{i,j,k} \vec{\delta}_\zeta T_k \overleftarrow{\delta}_\zeta) C^n_{i,j,k} = g^n_{j,k} \tag{18c}$$

where IB is the IBLANK integer array. This quantity is equal to one for all grid points that are activated or located inside the flow domain. Inactive grid points, e.g., grid points that lie inside a portion of the geometry and therefore outside of the flow domain, are assigned a value of zero. The computational algorithm is implemented at all points, but only the $IB = 1$ points produce a meaningful solution. The influence of all blanked out points, i.e., points with $IB = 0$, is removed by the IBLANK array. For grid points that lie outside the flow domain the AF2 scheme given earlier simply reduces to

$$C^n_{i,j,k} = 0$$

Chimera boundary conditions at grid-to-grid zonal interfaces are implemented using trilinear interpolation, which are updated after every iteration. This is the same type of iterpolative interface employed in typical Euler or Navier–Stokes formulations utilizing the chimera approach. That is, both approaches utilize trilinear interpolation on the scheme's dependent variables. For the present full-potential formulation, the velocity potential is used, and typically for the Euler/Navier–Stokes approach the standard conserved variables in the $\boldsymbol{Q}$ vector are used. Because the velocity potential is a very well-behaved function, i.e., it is continuous across shock waves (whereas the Euler/Navier–Stokes $\boldsymbol{Q}$-vector variables generally are not), interpolation of this function is also well behaved. However, most important quantities in fluid dynamics and aerodynamics, e.g., density, pressure, velocity, are only obtained from the velocity potential by taking derivatives. Because numerical differentiation is an error-aggravating process, it is expected that some error amplification may exist along interface boundaries for variables such as pressure and velocity.

In constructing the difference scheme for the full-potential equation care must be taken to avoid computational stencils that overlap from the interior grid points past the interpolated fringe points and into a region of the grid that is blanked out. One example of where this occurs in the present scheme is in the calculation of density in the η-direction flux, namely, $\rho_{i,j+1/2,k}$. Normally this value of density is computed from the density array (with values stored at $i + 1/2$, j, k) using a standard average given by

$$\rho_{i,j+1/2,k} = \tfrac{1}{4}(\rho_{i+1/2,j,k} + \rho_{i-1/2,j,k} + \rho_{i+1/2,j+1,k} + \rho_{i-1/2,j+1,k}) \tag{19}$$

At the η-direction flux that occurs directly between a fringe point and an interior point, values of $\rho_{i-1/2,j,k}$ and $\rho_{i+1/2,j,k}$ are not computable with the standard interior scheme. Instead, fringe values of the density are established from interior values using one-sided, first-order extrapolations. Then, the standard formula given by Eq. (19) is implemented in the usual way. Additional schemes for computing the density at fringe points that are second-order accurate are available but have not been used to date.

Care must be exercised when performing donor cell searches for fringe points that lie on the vortex sheet. This is because the x, y, and z coordinates for a particular fringe point on the upper side of the vortex sheet exactly match those of the corresponding lower point, i.e., either of these vortex-sheet points could end up finding a donor cell on the wrong side of the vortex sheet. The donor cell search algorithm must be modified so that only cells above the vortex sheet are searched for upper vortex-sheet fringe points and only cells below the vortex sheet are searched for lower vortex-sheet fringe points.

Numerical Results

Problem Setup and Grid Arrangement

To evaluate the attributes of the new chimera approach for solving the full-potential equation the familiar ONERA M6 wing geometry is chosen. The two-zone chimera grid chosen for this evaluation consists of an inner C–H grid embedded in an outer Cartesian-like grid (see Figs. 1 and 2). The Cartesian grid is sheared so that one of the grid planes is aligned with the wing leading edge and another with the wing trailing edge. In addition, the Cartesian grid is clustered in all three directions (streamwise, spanwise, and vertical) in the vicinity of the wing. These shearing and clustering operations provide a more consistent interface at the inner chimera interpolation boundary. For the present isolated-wing test case, both the inner and outer grids are constructed such that each η = constant surface is also a y = constant plane. This aspect is used to simplify the donor cell search algorithm, but not the flow solver or chimera interpolation schemes, which are fully general three-dimensional implementations.

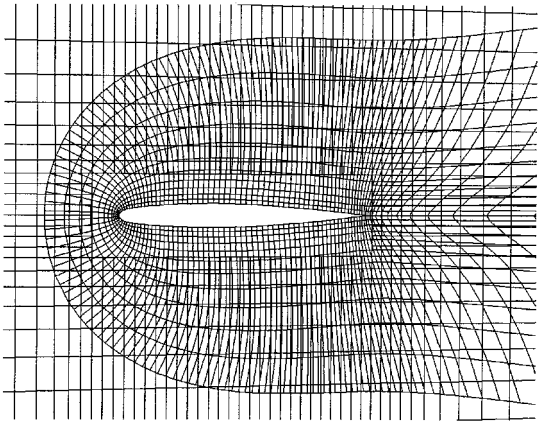

Fig. 1 Expanded view of a typical two-zone grid in the wing root plane, y = 0.0, inner grid: 151 × 32 × 13, outer grid: 73 × 33 × 50.

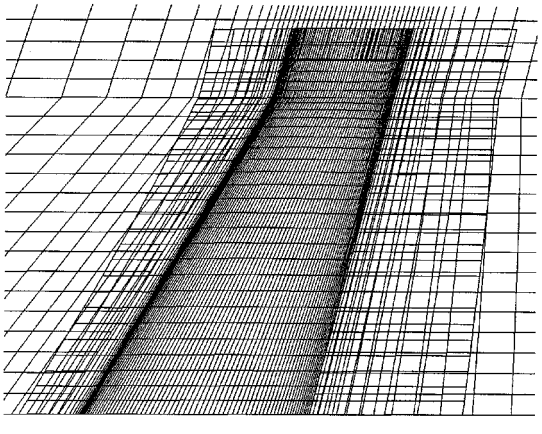

Fig. 2 Expanded view of a typical two-zone grid in the wing planform plane, z = 0.0, inner grid: 151 × 32 × 13, outer grid: 73 × 33 × 50.

All C–H grids used in this study are generated using the HYPGEN hyperbolic grid generation code[15] with full three-dimensional capability. Then after the grid is generated, the η = const surfaces are adjusted to achieve the desired constant y-plane distributions. Figures 1 and 2 show selected views from a typical two-zone grid generated in this fashion about the ONERA M6 wing. Figure 1 shows a close-up of both the inner C–H and outer Cartesian grids plotted in the wing-root symmetry plane. The inner grid is not orthogonal as one would expect with a hyperbolic grid generation technique because of a large amount of smoothing used in the grid generation process. Both inner and outer chimera interface boundaries are displayed. The outer interface boundary is located about 0.46 root chords upstream of the wing root leading edge, and about 0.86 root chords above and below the wing trailing edge. Note that the outer Cartesian grid is somewhat coarser than the inner C–H grid. This produces a somewhat larger truncation error on the Cartesian side of the interface boundary, a characteristic that will be particularly noticeable near all high-frequency components of the solution, particularly shock waves. Figure 2 shows both grids plotted around the wing in a planform view where the y = constant plane character is clearly evident. The outer Cartesian grid shearing and stretching and the inner C–H grid clustering at the wing tip are also clearly visible.

Grid Refinement Study

A grid refinement study for the ONERA M6 wing geometry is described next utilizing the four grids defined in Table 1. In each case the inner and outer grid dimensions are given by NI1 × NJ1 × NK1 and NI2 × NJ2 × NK2, respectively. The first dimension (either NI1 or NI2) is the number of grid points in the wraparound or streamwise direction; the second dimension (NJ1 or NJ2) is the number of points in the spanwise direction; and the third dimension (NK1 or NK2) is the number of points in the normal-like direction. The total number of points ranges from just under 40,000 to just under 500,000. The number of points defining the wing surface ranges from 1445 to 8651. Each grid in the grid refinement study utilizes the same grid distribution in each direction. The second-coarsest grid from Table 1 (case G2L3) is identical to the grid presented in Figs. 1 and 2.

Selected chordwise pressure coefficient distributions are presented in Fig. 3 at two different semispan locations, $2y/b$ = 0.44 and 0.95. In each plot solutions generated using the second-order scheme on all four grids described in Table 1 are compared with experimental results from Ref. 16. The flow conditions for this standard case (M_∞ = 0.84, α = 3.06 deg),

Table 1 Summary of grid statistics for the two-zone chimera grid refinement study

Case no.	Grid dimensions NI1 NI2	NJ1 NJ2	NK1 NK2	Total surface points	Total points
G2L2	101	23	9	1445	39,185
	37	19	26		
G2L3	151	32	13	3175	115,066
	55	25	38		
G2L4	201	41	17	5577	260,547
	73	33	50		
G2L5	251	50	21	8651	494,872
	91	41	62		

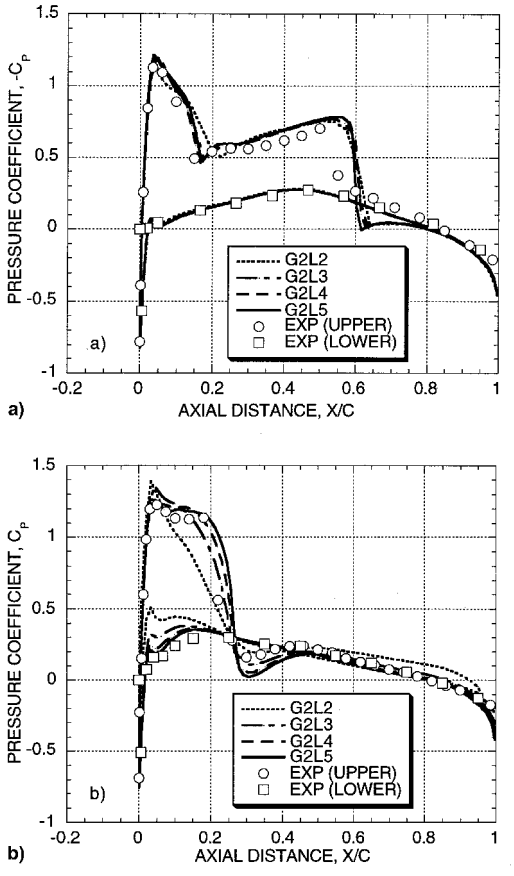

Fig. 3 Pressure coefficient comparisons at selected semispan wing stations showing the effect of grid refinement (two-zone chimera second-order scheme); ONERA M6 wing, $M_\infty = 0.84$, $\alpha = 3.06$ deg; see Table 1 for grid definitions; $2y/b$ = a) 0.44 and b) 0.95.

produce a rather benign flowfield with a moderate-to-weak shock wave system and only a small amount of shock/boundary-layer interaction. As can be seen from Fig. 3 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 most likely 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.

Another more qualitative technique for showing the effects of grid refinement on solution accuracy is shown in Fig. 4, where Mach number contours are displayed for the ONERA M6 wing upper surface. These contours were generated using the PLOT3D plotting program[17] with contours plotted in 0.025 increments. A contour plot is displayed for the coarsest (Fig. 4a) and finest (Fig. 4b) grids described in Table. 1. Evolution of the solution with grid refinement, in particular, the upper-surface shock system, is clearly evident. The supersonic-to-supersonic oblique shock is the most noticeable feature that forms and sharpens as the grid is refined. Additional grid refinement results using the base hybrid scheme for both the single- and two-zone-chimera approaches are presented in Refs. 18 and 19.

Hybrid/Second-Order and Single/Two-Zone Scheme Comparisons

In this section differences in accuracy between the hybrid and second-order differencing schemes and between the single- and two-zone chimera approaches are examined. Chordwise pressure coefficient distributions over the ONERA M6 wing for three different computations (single-grid hybrid, two-zone hybrid, and two-zone second-order) are compared with experimental results in Fig. 5. For brevity, these comparisons are presented at only one semispan station: $2y/b = 0.44$. Both two-zone results utilize the G2L4 grid described in Table 1. The single-zone results presented in Fig. 5 utilize a C–H topology, HYPGEN-generated grid with a wing surface point distribution identical to that of the G2L4 grid. This single-zone grid is topologically identical to the inner C–H grid shown in Figs. 1 and 2. However, the outer boundaries are extended (with additional grid points in the j and k directions) to approximately match the Cartesian-grid outer boundary locations in the two-zone grid.

As can be seen in Fig. 5, results from the single- and two-zone hybrid solutions are nearly identical, i.e., the effect of the

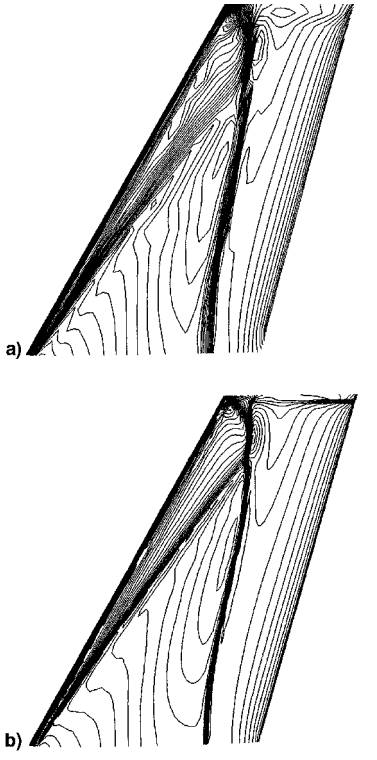

Fig. 4 Mach number contours on the upper surface of the ONERA M6 wing showing the effect of grid refinement; two-zone chimera second-order scheme; $M_\infty = 0.84$, $\alpha = 3.06$ deg; see Table 1 for grid definitions; a) coarse grid (G2L2) and b) finest grid (G2L5).

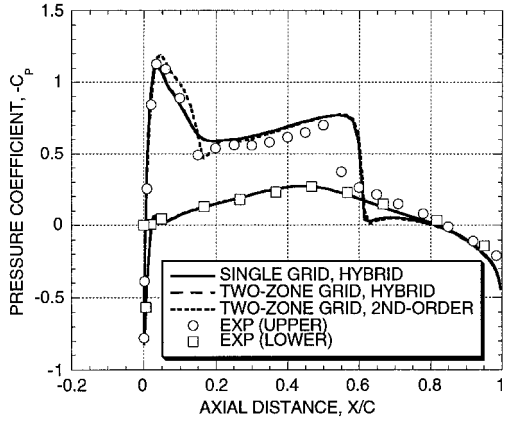

Fig. 5 Pressure coefficient comparisons between the single-grid hybrid, two-zone hybrid, and two-zone, second-order schemes; ONERA M6 wing, $2y/b = 0.44$, $M_\infty = 0.84$, $\alpha = 3.06$ deg.

chimera interface located less than one-chord away from the wing surface has a negligible effect on the wing surface solution. Agreement between the hybrid solutions and the second-order solution is excellent 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. Agreement between the hybrid and second-order results in the supersonic region (as expected) is not as good. The forward shock, which is a supersonic-to-supersonic oblique shock, is smeared out by the hybrid scheme and captured more accurately by the second-order scheme. Generally, the computational/experimental agreement is quite good everywhere except at the normal shock wave where some disagreement in strength and position exists. This disagreement, typical of a full-potential/experimental transonic flow comparison, is primarily caused by a lack of viscous flow modeling and to the isentropic assumption inherent in the full-potential formulation.

Convergence Efficiency

Convergence history results are presented showing the effect of several factors on convergence efficiency. Figure 6 shows a plot of maximum residual, average residual, and the lift coefficient as a function of iteration for the single-zone hybrid, the two-zone hybrid, and the two-zone, second-order schemes. Each simulation used the grid described in the previous section. The solution parameters [mainly α_L, see Eq. (12)] have been optimized by a trial-and-error process for all of these solutions. Each symbol in Fig. 6 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., $M = 8$ [see Eq. (12)].

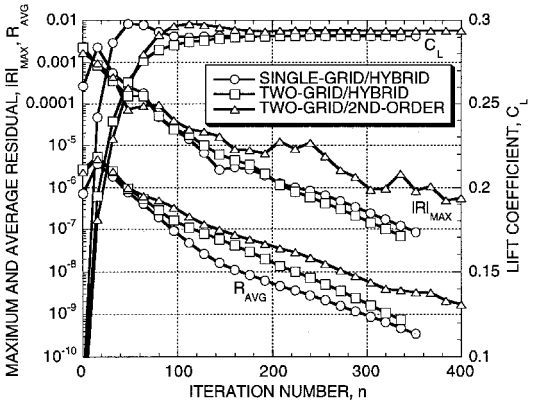

Fig. 6 Convergence history comparisons between the hybrid-single-zone, hybrid-two-zone, and second-order, two-zone schemes; ONERA M6 wing, $M_\infty = 0.84$, $\alpha = 3.06$ deg.

As can be seen from Fig. 6, convergence efficiency comparisons are dependent on which convergence criteria is used. In terms of lift convergence, the single-zone hybrid scheme converges the fastest, being about 1.6 times faster than the two-zone hybrid scheme and about 2.1 times faster than the two-zone, second-order scheme. In terms of maximum residual the single- and two-zone hybrid schemes converge at about the same rate, both being about 1.4 times faster than the two-zone, second-order scheme. These efficiency comparisons utilize the number of iterations required to achieve the desired convergence criteria. When computer time is used, the situation changes somewhat because each single-zone iteration requires about 35% more computer time than a two-zone iteration. This is because the single-zone grid requires about 35% more grid points than the two-zone grid to match the same surface distribution, i.e., the two-zone grid distributes points more efficiently than the single-zone grid.

A more quantitative picture of convergence efficiency is presented in Table 2, which contains iteration count and CPU timing information for the two-zone hybrid and two-zone, second-order schemes. Results are presented for all four grids described in Table 1. No attempt has been made to optimize convergence for these cases (except for the G2L4 grid cases already presented in Fig. 6). All computer timings are from a Cray C-90 computer using level 3 optimization for the inline, vector, and scalar options of the cf77 compiler. For completeness, lift, drag, and pitching moment coefficient values are displayed for each scheme and for each grid described in Table 1. The five other quantities displayed in Table 2 are related to convergence/execution efficiency and are defined as follows: n_{LIFT} is the number of iterations required to achieve lift to within $\pm 0.1\%$ of the final value; t_{LIFT} is the computer time required for n_{LIFT} iterations; $n_{R\text{av}}$ is the number of iterations required to achieve an average residual level of 10^{-8}; $t_{R\text{av}}$ is the computer time required for $n_{R\text{av}}$ iterations; and $t_{O/H}$ is the computer time required for solution overhead. The overhead time corresponds to grid generation, solution initialization, donor-cell search, interpolation coefficient computation, metric computation, and PLOT3D input file generation. As established in Fig. 6, there is a moderate degradation in convergence efficiency for the second-order scheme relative to the hybrid scheme for most of the grids utilized and for both the lift and average residual convergence criteria. In general, there is also a degradation in convergence efficiency with grid refinement. Nevertheless, solution times are quite modest, ranging from 3 to 91 s. On the G2L4 grid, a solution is obtained in 26–28 s for the hybrid scheme and in 34–36 s for the second-order scheme.

Outer-Grid Refinement Study

The effect of outer grid refinement on solution accuracy for the two-zone chimera approach is presented in Figs. 7 and 8.

Table 2 Summary of computational statistics for the two-zone, chimera-grid full potential results displayed in Figs. 3 and 4, ONERA M6 wing, $M_\infty = 0.84$, $\alpha = 3.06$ deg[a]

Case no.	C_L	C_D	C_M	n_{LIFT}	t_{LIFT}, s	$n_{R\text{av}}$	$t_{R\text{av}}$, s	$t_{O/H}$ s
			Hybrid scheme					
G2L2	0.2839	0.0103	−0.1653	96	3	224	8	2
G2L3	0.2880	0.0106	−0.1683	160	11	176	13	5
G2L4	0.2910	0.0108	−0.1705	208	26	224	28	10
G2L5	0.2923	0.0109	−0.1715	288	66	256	58	18
			Second-order scheme					
G2L2	0.2892	0.0113	−0.1678	128	4	176	6	2
G2L3	0.2916	0.0111	−0.1696	160	11	224	16	5
G2L4	0.2937	0.0112	−0.1713	272	34	288	36	10
G2L5	0.2944	0.0112	−0.1720	384	87	400	91	18

[a]Computer times are from a single processor Cray C-90 computer.

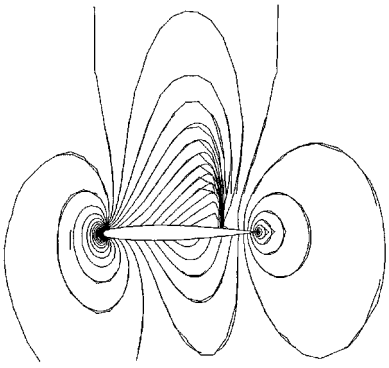

Fig. 7 Mach number contours about the ONERA M6 wing in the wing root symmetry plane; M_∞ = 0.84, α = 3.06 deg, inner grid = 151 × 32 × 13, outer grid = 37 ×19 × 26.

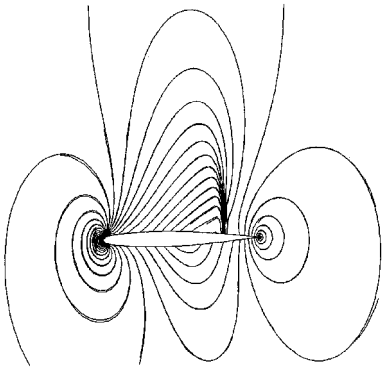

Fig. 8 Mach number contours about the ONERA M6 wing in the wing root symmetry plane; M_∞ = 0.84, α = 3.06 deg, inner grid = 151 × 32 × 13, outer grid = 157 × 65 × 90.

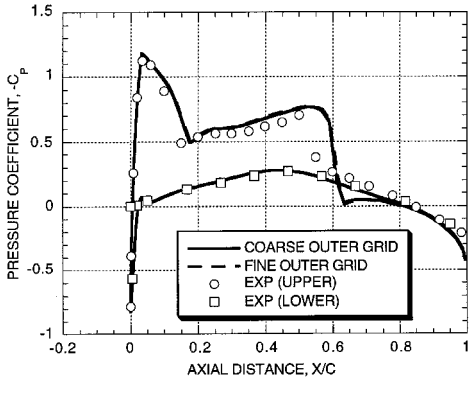

Fig. 9 Surface pressure coefficient comparison between the coarse outer grid result of Fig. 7 and the fine outer grid result of Fig. 8; ONERA M6 wing, 2y/b = 0.44, M_∞ = 0.84, α = 3.06 deg, inner grid = 151 × 32 × 13.

Mach number contours are displayed in the wing root symmetry plane for a coarse outer grid involving 37 × 19 × 26 points (Fig. 7) and for a very fine outer grid involving 157 × 65 × 90 points (Fig. 8). In each of these cases the inner grid is fixed at 151 × 32 × 13 points. In these figures contours from both the inner and outer solutions are plotted. The outer boundary location is fixed in the same location for both cases. However, the inner interface boundary is dependent on the outer grid refinement level and, thus, is somewhat closer to the wing surface for the fine outer grid case. For the coarse outer grid case (Fig. 7) the discrepancies between the inner and outer solutions at the inner grid interface are pronounced, particularly at the shock wave where the outer grid is not fine enough to support such solution detail. For the fine outer grid case (Fig. 8) the discrepancies are much smaller, particularly at the shock wave where the inner and outer grids have about the same streamwise grid spacing. Careful examination of the Mach number contours near the wing surface from the two inner grid solutions shows no discernible differences. In other words, the level of outer grid refinement plays a dramatically small role in determining solution accuracy at the wing surface. To accentuate this latter point, the surface pressure coefficient distributions from the two solutions displayed in Figs. 7 and 8 are compared in Fig. 9. Both results are almost identical. This implies that good surface aerodynamic results can be obtained with the present full-potential chimera approach more cost effectively using fine-inner/coarse-outer chimera grid systems. Of course, when interior flow details are required at distances moderately far removed from the aerodynamic surfaces, the fine-inner/coarse-outer grid philosophy may not be appropriate. But it is interesting to note that for the two almost-identical results presented in Fig. 9, the coarse-outer grid solution required about 10 times less computer time than the fine-outer grid solution.

Concluding Remarks

A new solution technique for solving the full-potential equation has been presented and evaluated using a standard three-dimensional transonic wing computation. The scheme has three new features including a spatial discretization scheme based on a solution limiter concept, a modified AF2 iteration scheme that is specifically designed for C-grid topologies, and a chimera-based zonal-grid implementation that will (eventually) allow complex geometry analysis. Numerous numerical results have been 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 and efficient computational tool for the aerodynamic analysis and design of geometrically complex aerodynamic configurations. Additional specific concluding remarks are given as follows:

1) The spatial discretization scheme has a traditional hybrid option, which is second-order accurate in subsonic regions of flow and first-order accurate in supersonic regions, and a new option that is fully second-order accurate regardless 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.

2) The second-order spatial discretization scheme is less computationally efficient than the hybrid scheme requiring 30–40% more iterations for a converged solution, but more accurate in supersonic regions of flow, particularly in terms of supersonic-to-supersonic shock wave capture.

3) The new iteration scheme utilizes two different AF2-scheme variations; one above the wing and the other below the wing, and connects the two at the wing leading edge using a local iteration. This special design removes (or at least controls) a stability limitation that exists in other implementations of this scheme and allows this implementation to be efficient for C-grid topologies.

4) Existence of the explicit chimera-interface boundaries causes only a moderate degradation in convergence efficiency, which can be largely compensated for by using more efficient fine-inner/coarse-outer zonal grid systems.

5) Numerical comparisons between single- and two-zone solutions demonstrate that the chimera interface has a negligible impact on solution accuracy at the wing surface, even when shocks pass through the chimera interface boundary. Further two-zone chimera comparisons between coarse- and fine-outer grid cases demonstrate that the outer grid refinement level has an almost negligible impact on wing surface solution accuracy. This suggests that good aerodynamic accuracy can be obtained at a fraction of the cost by using fine-inner/coarse-outer grid systems.

References

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

[2]Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera Grid Scheme," *Advances in Grid Generation*, edited by K Ghia and U. Ghia, FED-5, American Society of Mechanical Engineers, New York, 1983, pp. 59–69.

[3]Holst, T. L., "Implicit Algorithm for the Conservative Transonic Full Potential Equation Using an Arbitrary Mesh," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1038–1045.

[4]Holst, T. L., and Thomas, S. D., "Numerical Solution of Transonic Wing Flow fields," *AIAA Journal*, Vol. 21, No. 6, 1983, pp. 863–870.

[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 Journal*, Vol. 22, No. 8, 1984, pp. 1027–1034.

[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 95-1765, June 1995.

[9]Jameson, A., "Transonic Potential Flow Calculation Using Conservative Form," *Proceedings of the AIAA 2nd Computational Fluid Dynamics Conference*, AIAA, New York, 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, Nov. 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 Journal*, Vol. 16, No. 6, 1978, pp. 573–579.

[12]South, J. C., Jr., and Hafez, M. M., "Stability Analysis of Intermediate Boundary Conditions in Approximate Factorization Schemes," AIAA Paper 83-1898, July 1983.

[13]Holst, T. L., and Ballhaus, W. F., Jr., "Fast Conservative Schemes for the Full Potential Equation Applied to Transonic Flows," *AIAA Journal*, Vol. 17, No. 2, 1979, pp. 145–152.

[14]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.

[15]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, Oct. 1993.

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

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

[18]Holst, T. L., "Full Potential Equation Solutions Using a Chimera Grid Approach," AIAA Paper 96-2423, June 1996.

[19]Holst, T. L., "On Approximate Factorization Schemes for Solving the Full Potential Equation," NASA TM 110435, Feb. 1997.