

Transonic Navier-Stokes Computations for an Oscillating Wing Using Zonal Grids

Neal M. Chaderjian* and Guru P. Guruswamy†
NASA Ames Research Center, Moffett Field, California 94035

Modern jet transports and maneuvering tactical fighters operating in the transonic regime often give rise to time-dependent fluid physics that interact with flexible structural components, e.g., vortical flow, shocks, and separation. Efficient computational fluid dynamic methods are required to study such computationally intensive problems. A numerical method is presented to address this problem. Time-dependent, compressible, Navier-Stokes equations are used to simulate unsteady transonic flow about a three-dimensional rigid wing undergoing a forced periodic motion in angle of attack. An efficient, implicit, diagonal algorithm is utilized because of its low operation count per time step compared to other methods that solve systems of block matrix equations. The formal time accuracy is theoretically addressed and numerically demonstrated by comparison of computational results with experimental data. A zonal grid approach, capable of treating complex geometries, is presented and its time accuracy is demonstrated by comparing two- and three-zone computations with a single grid computation and experimental data.

Introduction

JET transports and maneuvering tactical fighters tend to operate in the transonic regime where the flowfield is primarily subsonic with regions of embedded supersonic flow. This often gives rise to complex fluid physics such as steady and time-dependent vortical flow, shocks, and separation. It has been observed that these phenomena interact with the flexible components of an aircraft and can lead to sustained aeroelastic oscillations and divergence.¹⁻³ The ability to predict accurately the time-dependent flowfield and aeroelastic response of an aircraft is essential to insure the integrity and safety of the vehicle and can help extend the maneuvering envelop to modern tactical fighters.

Computational fluid dynamics (CFD) can play an important role in the design process by providing detailed flowfield and structural displacement information. This will help reduce the design cycle time and provide information that is complementary to wind-tunnel and flight-test data. At present, the most advanced CFD codes for predicting the aeroelastic response of wing/fuselage configurations are based on potential flow theory.⁴ Even though these equations have a limited range of applicability, they have been used because the more complete Navier-Stokes equations require significantly more computer memory and computer time. Navier-Stokes equations, when coupled with the governing aeroelastic equations, can require two or more orders of magnitude of computer time. However, recent advances in numerical algorithms and computer technology make it possible to consider using these equations in the research environment and eventually in the design environment.^{5,6} The literature confirms this trend, as several time-dependent, three-dimensional, Navier-Stokes solutions for wing geometries have begun to emerge, e.g., Refs. 7 and 8, as well as aeroelastic computations, e.g., Refs. 9 and 10. It is our eventual goal to simulate the aeroelastic

response of a complete aircraft with the Navier-Stokes equations. However, as a first step toward this goal, we will address some accuracy and efficiency issues regarding the use of an Euler/Navier-Stokes code for predicting the time-dependent, three-dimensional, flowfield about a rigid wing undergoing a forced periodic motion in angle of attack.

In order to simulate time-dependent viscous flow about a realistic aircraft geometry, it is important for a Navier-Stokes code to be able to treat complex geometries, have an efficient and robust numerical algorithm, and be time accurate. The Transonic Navier-Stokes (TNS) code has been chosen as a base code because of its success in predicting the steady transonic viscous flow about the complete F-16A fighter aircraft.¹¹ This code uses a zonal grid approach for treating the geometric complexity of the aircraft. The flowfield is subdivided into an ensemble of simply shaped grids. Each of these grids can be refined by adding clustered grid points as appropriate to adequately capture viscous effects near all body surfaces and wake regions. Flowfield coupling between zones is accomplished by overlapping adjacent zonal grids. Moreover, this code uses an efficient and robust implicit approximate-factorization algorithm. This algorithm, due to Pulliam and Chaussee,¹² is a diagonal version of the Beam-Warming algorithm¹³ and uses central differencing. The operation count per time step is therefore low compared to most upwind schemes and methods requiring the solution of a system of block matrix equations. However, the TNS code has only been used for steady flow computations.

The Navier-Stokes Simulation (NSS) code is a new and extended version of the TNS code for treating time-dependent flows. It allows for a moving zonal grid system with time-dependent coordinate transformation metrics. A Fourier-analysis routine has also been included to facilitate comparison of computed surface pressure, force, and moment coefficients with experimental data.

The Pulliam-Chaussee diagonal algorithm is conservative in space but not conservative in time. This can affect the shock speed in a transonic computation. The form of this temporal conservation error will be presented in a later section, as well as ways to remove the error. The objective of this research is to verify the time accuracy of the NSS code algorithm and zonal boundary treatment by comparing computational results with experimental data for transonic flow over an oscillating three-dimensional wing with moving shocks.

In the following sections, a theoretical background is given describing the governing equations, turbulence model, nu-

Presented as Paper 90-0313 at the AIAA 28th Aerospace Sciences Meeting, Reno, NV, Jan. 8-11, 1990; received March 26, 1990; revision received April 9, 1991; accepted for publication April 17, 1991. Copyright © 1991 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

*Research Scientist, Applied Computational Fluids Branch. Member AIAA.

†Research Scientist, Applied Computational Fluids Branch. Associate Fellow AIAA.

merical algorithm, and formal time accuracy. This is followed by a description of the zonal grid approach, grid generation, boundary conditions, results, and concluding remarks.

Theoretical Background

Governing Equations

The compressible Reynolds-averaged Navier-Stokes equations are used to numerically simulate time-dependent viscous flow about a wing with oscillating angle of attack. For the high Reynolds number flows considered in this paper, the thin-layer approximation is employed together with body-fitted curvilinear coordinates. This improves the efficiency of the numerical simulation and simplifies the implementation of boundary conditions. Assuming a general coordinate transformation from physical space (x, y, z) to computational space (ξ, η, ζ) and that viscous terms are only important in the ζ -coordinate direction, the governing equations can be expressed in the following strong conservation-law form:

$$\partial_t \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = Re^{-1} \partial_\zeta \hat{S} \quad (1)$$

where

$$\hat{Q} = \frac{1}{J} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad \hat{E} = \frac{1}{J} \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ U(e + p) - \xi_t p \end{bmatrix}$$

$$\hat{F} = \frac{1}{J} \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ V(e + p) - \eta_t p \end{bmatrix}, \quad \hat{G} = \frac{1}{J} \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ W(e + p) - \zeta_t p \end{bmatrix}$$

$$\hat{S} = \frac{1}{J} \begin{bmatrix} 0 \\ \mu m_1 u_\zeta + (\mu/3)m_2 \zeta_x \\ \mu m_1 v_\zeta + (\mu/3)m_2 \zeta_y \\ \mu m_1 w_\zeta + (\mu/3)m_2 \zeta_z \\ \mu m_1 m_3 + (\mu/3)m_2 (\zeta_x u + \zeta_y v + \zeta_z w) \end{bmatrix}$$

with

$$m_1 = \zeta_x^2 + \zeta_y^2 + \zeta_z^2, \quad m_2 = \zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta,$$

$$m_3 = 0.5(u^2 + v^2 + w^2)_\zeta + Pr^{-1}(\gamma - 1)^{-1}(a^2)_\zeta$$

In Eq. (1), the contravariant velocity components are defined by

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w \quad (2a)$$

$$V = \eta_t + \eta_x u + \eta_y v + \eta_z w \quad (2b)$$

$$W = \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \quad (2c)$$

and the pressure is related to the other flow variables by the perfect gas law

$$p = (\gamma - 1)[e - \frac{1}{2} \rho(u^2 + v^2 + w^2)] \quad (3)$$

The transformation metrics are defined as

$$\xi_x = J(y_\eta z_\zeta - z_\eta y_\zeta), \quad \eta_x = J(z_\xi y_\zeta - y_\xi z_\zeta) \quad (4a)$$

$$\xi_y = J(z_\eta x_\zeta - x_\eta z_\zeta), \quad \eta_y = J(x_\xi z_\zeta - z_\xi x_\zeta) \quad (4b)$$

$$\xi_z = J(x_\eta y_\zeta - y_\eta x_\zeta), \quad \eta_z = J(y_\xi x_\zeta - x_\xi y_\zeta) \quad (4c)$$

$$\zeta_x = J(y_\xi z_\eta - z_\xi y_\eta), \quad \xi_t = -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z \quad (4d)$$

$$\zeta_y = J(z_\xi x_\eta - x_\xi z_\eta), \quad \eta_t = -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z \quad (4e)$$

$$\zeta_z = J(x_\xi y_\eta - y_\xi x_\eta), \quad \zeta_t = -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z \quad (4f)$$

and the coordinate-transformation Jacobian by

$$1/J = x_\xi(y_\eta z_\zeta - z_\eta y_\zeta) + x_\eta(z_\xi y_\zeta - y_\xi z_\zeta) + x_\zeta(x_\xi z_\eta - z_\xi y_\eta) \quad (5)$$

Turbulence Model

Fluid turbulence is modeled by the Baldwin-Lomax algebraic eddy viscosity model.¹⁴ This isotropic model is used primarily because it is computationally efficient. All viscous computations presented in this paper assume fully turbulent flow at the leading edge of the wing, i.e., transitional effects are ignored. This approximation is consistent with the high Reynolds number assumption.

Numerical Algorithm

An implicit approximate-factorization diagonal algorithm is used in the NSS code to integrate Eq. (1). This algorithm is based on the Beam-Warming algorithm¹³:

$$(I + h\delta_\xi \hat{A} - hD_i|_\xi)^n (I + h\delta_\eta \hat{B} - hD_i|_\eta)^n \times (I + h\delta_\zeta \hat{C} - hRe^{-1} \bar{\delta}_\zeta J^{-1} \bar{M} J - hD_i|_\zeta)^n \Delta \hat{Q}^n = \hat{R}^n \quad (6)$$

where

$$\hat{R}^n = -\Delta t(\delta_\xi \hat{E} + \delta_\eta \hat{F} + \delta_\zeta \hat{G} - Re^{-1} \bar{\delta}_\zeta \hat{S} + D_e \hat{Q})^n$$

and

$$\Delta \hat{Q}^n = \hat{Q}^{n+1} - \hat{Q}^n$$

In Eq. (6), the superscript n corresponds to the current time level t^n and $n + 1$ to the new time level $t^{n+1} = t^n + \Delta t$. Also, $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{M}$ are the Jacobian matrices of $\hat{E}$, $\hat{F}$, $\hat{G}$, and $\hat{S}$, respectively. The numerical metrics are evaluated so that uniform flow is an exact solution of the steady finite difference equation (see Ref. 15). The Beam-Warming algorithm is second-order accurate in time when $h = 1/2\Delta t$ (trapezoidal rule) and first-order accurate in time when $h = \Delta t$ (Euler implicit). The spatial operators use central differencing throughout, so fourth-order explicit D_e and second-order implicit D_i numerical dissipation terms are added to damp high frequency errors. One drawback to this method, however, is the required solution of a block tridiagonal system of equations, which is computationally costly. For more details of this algorithm, see Pulliam and Steger.¹⁶

In order to reduce the overall execution time of the NSS code, a more efficient diagonal form of the Beam-Warming algorithm is used. This algorithm, due to Pulliam and Chaussee,¹² given is by

$$(T_A)^n (I + h\delta_\xi \Lambda_A + hD_i|_\xi)^n (\hat{N})^n (I + h\delta_\eta \Lambda_B + hD_i|_\eta)^n \times (\hat{P})^n (I + h\delta_\zeta \Lambda_C + hD_i|_\zeta)^n (T_C^{-1})^n \Delta \hat{Q}^n = \hat{R}^n \quad (7)$$

where Λ_A is a diagonal matrix consisting of the eigenvalues of $\hat{A}$, and so on. Note that the viscous terms are treated explicitly. The right side of Eq. (7) is identical to the right side of Eq. (6). This algorithm uses both explicit and implicit fourth-order dissipation and only requires the solution of scalar pentadiagonal equations. This results in a lower operation count per time step and, hence, a reduced execution time per solution. Further details of this algorithm, including a description of the T_A , $\hat{N}$, and $\hat{P}$ matrices, are given in Refs. 12 and 17.

When a steady-state solution is desired, the rate of convergence can be greatly improved by using the spatially varying time step described by Srinivasan et al.¹⁸ and is given by

$$\Delta t = \frac{\Delta t_0}{1 + \sqrt{J}} \quad (8)$$

where Δt_0 is a user specified constant. Flores,¹⁹ using the diagonal algorithm with variable time step, reported convergence rates 40 times those obtained using the Beam-Warming algorithm with constant time step. A constant time step is used for time-accurate computations.

All of the computations reported here use the implicit diagonal algorithm. In its present coded form (NSS code), the implicit operator only requires 70% more computer time per time step than the explicit operator. Yet the implicit scheme has been used with a maximum Courant number of about 900. Therefore, this algorithm is far more efficient (overall computer time) than its explicit counterpart. However, due to its diagonal form, this algorithm is first-order accurate in time and temporally nonconservative. These two issues are addressed in the next section and in the Results section.

Time Accuracy

The strong conservation-law form of the governing equations, Eq. 1, when properly differenced, conserves numerical fluxes across flow discontinuities. These so called weak solutions, i.e., piecewise continuous solutions, satisfy the proper integral relations irrespective of the grid spacing or time step. If a nonconservative form is used, the shock strength, position, and wave speed will be dependent on the grid spacing and time step.

Although the diagonal algorithm, Eq. (7), is conservative in space, it is not conservative in time. Therefore, time-accurate computations involving shocks can be questionable. However, since the explicit side is differenced conservatively, one might expect the temporal error introduced in the implicit operator to be nominal for a variety of applications with moving shocks. This will be demonstrated computationally in the Results section.

This section will address the form of the conservation error and the formal time accuracy of the algorithm. Two approaches for removing this error will also be presented in the event that temporal conservation errors prove to be important for a specific application.

The temporal conservation error of the diagonal algorithm can be found by going back to its derivation from the Beam-Warming algorithm. Replacing the Jacobian matrices of Eq. (6) with

$$A = T_A \Lambda_A T_A^{-1}$$

$$B = T_B \Lambda_B T_B^{-1}$$

$$C = T_C \Lambda_C T_C^{-1}$$

and dropping the implicit viscous terms gives

$$\begin{aligned} & (T_A T_A^{-1} + h \partial_\xi T_A \Lambda_A T_A^{-1} + h T_A D_i|_\xi T_A^{-1})^n \\ & \times (T_B T_B^{-1} + h \partial_\eta T_B \Lambda_B T_B^{-1} + h T_B D_i|_\eta T_B^{-1})^n \\ & \times (T_C T_C^{-1} + h \partial_\zeta T_C \Lambda_C T_C^{-1} + h T_C D_i|_\zeta T_C^{-1})^n \Delta \hat{Q}^n = \hat{R}^n \end{aligned} \quad (9)$$

In Eq. (9), Λ_A is a diagonal matrix whose elements are the eigenvalues of A and T_A is the matrix whose columns are the eigenvectors of A . Similar terms apply for the B and C matrices. Spatial finite difference operators have been replaced by differential operators to avoid unnecessary details that do not affect the analysis. Application of the chain rule results in

$$\begin{aligned} & (T_A T_A^{-1} + h T_A \partial_\xi \Lambda_A T_A^{-1} + h (\partial_\xi T_A) \Lambda_A T_A^{-1} \\ & + h T_A D_i|_\xi T_A^{-1})^n \times (T_B T_B^{-1} + h T_B \partial_\eta \Lambda_B T_B^{-1} \\ & + h (\partial_\eta T_B) \Lambda_B T_B^{-1} + h T_B D_i|_\eta T_B^{-1})^n \\ & \times (T_C T_C^{-1} + h T_C \partial_\zeta \Lambda_C T_C^{-1} + h (\partial_\zeta T_C) \Lambda_C T_C^{-1} \\ & + h T_C D_i|_\zeta T_C^{-1})^n \Delta \hat{Q}^n = \hat{R}^n \end{aligned} \quad (10)$$

and collecting terms give the following form for the Beam-Warming algorithm:

$$\begin{aligned} & (T_A)^n (I + h \partial_\xi \Lambda_A + h D_i|_\xi)^n \\ & \times (\hat{N})^n (I + h \partial_\eta \Lambda_B + h D_i|_\eta)^n \times (\hat{P})^n (I + h \partial_\zeta \Lambda_C \\ & + h D_i|_\zeta) (T_C^{-1})^n \Delta \hat{Q}^n + h \epsilon^n = \hat{R}^n \end{aligned} \quad (11)$$

where

$$\begin{aligned} \epsilon^n &= [(\partial_\xi T_A) \Lambda_A T_A^{-1} + (\partial_\eta T_B) \Lambda_B T_B^{-1} \\ &+ (\partial_\zeta T_C) \Lambda_C T_C^{-1}]^n \Delta \hat{Q}^n + \text{higher order terms} \\ &\sim O(\Delta \hat{Q}^n) \end{aligned}$$

The Beam-Warming algorithm can therefore be expressed as the diagonal algorithm with an additional term. This extra term ϵ^n corresponds to the temporal conservation error of the diagonal algorithm. Now,

$$\Delta \hat{Q}^n = \hat{Q}^{n+1} - \hat{Q}^n = \frac{\partial \hat{Q}^n}{\partial t} \Delta t + O(\Delta t^2)$$

so that

$$\epsilon^n \sim O(\Delta t) \quad (12)$$

The diagonal algorithm is formally first-order accurate in time, even if trapezoidal rule is used. (This algorithm is also limited to first-order time accuracy due to the explicit treatment of the viscous terms.) This error ϵ^n can be important if the shock strength and speed become too large for a given time step. However, this error can be controlled by lowering the time step for a given grid. This provides a method for checking if this error is important for a specific application. We also remark that the time step used in Navier-Stokes computations are generally small because of stability considerations, hence the temporal conservation error should be small for a large class of applications.

There is another way of showing that the conservation error vanishes with decreasing time step. As the time step $h \rightarrow 0$, the dominant term in the implicit operator (see Eq. 7) is the identity matrix. This method approaches the Euler explicit algorithm for very small time steps, which is fully conservative.

Another method for removing the temporal conservation error is to use a Newton iteration method described by Chakravarthy.²⁰ The diagonal algorithm then has the form

$$\begin{aligned} & (T_A)^i (I + h \partial_\xi \Lambda_A + h D_i|_\xi)^i (\hat{N})^i (I + h \partial_\eta \Lambda_B + h D_i|_\eta)^i \\ & (\hat{P})^i (I + h \partial_\zeta \Lambda_C + h D_i|_\zeta) (T_C^{-1})^i \Delta \hat{Q}^i \\ & = -(\hat{Q}^i - \hat{Q}^n) + \hat{R}^i \end{aligned} \quad (13)$$

In this expression, $\Delta \hat{Q}^i \rightarrow 0$ and $\hat{Q}^i \rightarrow \hat{Q}^{n+1}$ as i becomes large. This usually requires only a few subiterations to converge to the new time level. This approach not only removes the conservation error, but also removes the approximate-factorization error and makes the scheme fully implicit. This approach, however, can significantly increase the computational time of the code.

Zonal Approach

Generating a single grid about a complex geometric shape, e.g., a fighter aircraft, with appropriate viscous clustering near all body surfaces and wakes can be a formidable (if not impossible) task. This difficulty can be overcome by adopting a zonal grid approach. The NSS code uses a zonal grid procedure that subdivides the flowfield about a complex geometry

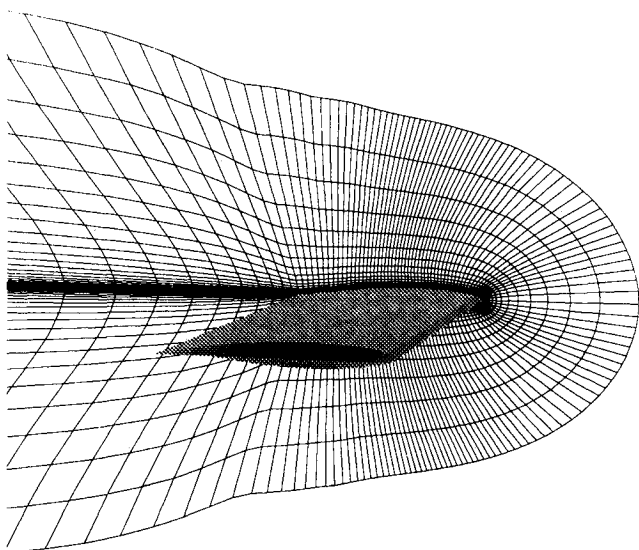


Fig. 1 Perspective view of a C-H-mesh grid for a NACA 64A010 rectangular wing.

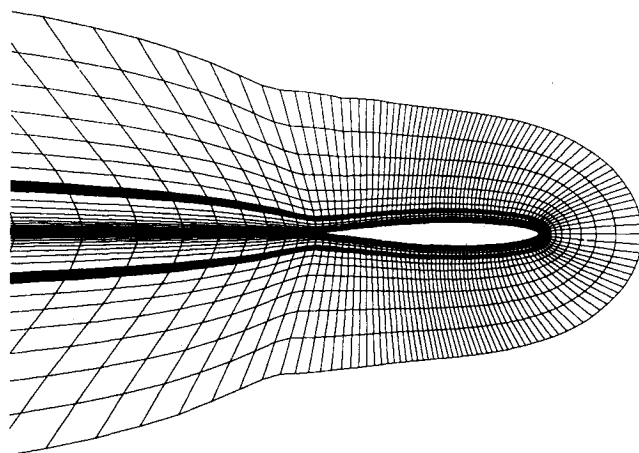


Fig. 2 Symmetry plane view of a two-zone C-H-mesh grid for a NACA 64A010 rectangular wing. One cell overlap between zones is shaded.

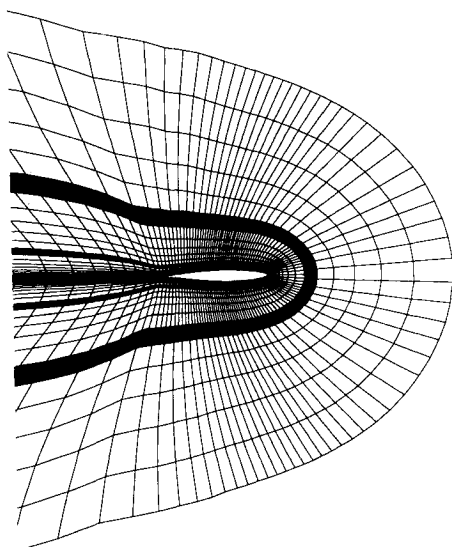


Fig. 3 Symmetry plane view of a three-zone C-H-mesh grid for a NACA 64A010 rectangular wing. One cell overlap between zones is shaded.

into an ensemble of simple geometric shapes. Adjacent zones overlap with one another, typically by one or two grid cells, to couple the flowfields between zones. These zones are constructed from a coarse global grid, and so they have a common interface surface. Each zonal grid is then clustered and refined with additional grid points as appropriate to accurately capture the pertinent flow physics. This procedure ensures that at an interface surface, the grid points of the coarser grid are coincident with grid points of the finer grid. Flow variable information is transferred from the finer grid to the coarser grid by direct injection. This step is fully conservative. Flow variable information is transferred from the coarser grid to the finer grid by interpolation along grid lines. This step is not conservative. When conservation between zones is important, e.g., when a shock passes through a zonal boundary, both zones are required to have identical grid points at the interface. This will insure complete flow conservation across the zonal boundary.

The flowfield is advanced to a new time level one zone at a time. The most recent data available at a zonal interface are used. Therefore, some zonal boundaries are updated explicitly, whereas others are updated implicitly. Any effects related to the explicit treatment of some zonal boundaries will vanish as the time step is decreased. A comparison between single, two-, and three-zone flow computations is given in the Results section. For more details of the zonal grid procedure and interface boundary treatment see Flores et al.²¹

Grid Generation

All computations presented in this paper use a C-H grid topology. A perspective view of a medium density grid is shown in Fig. 1. In particular, the surface geometry and part of the symmetry plane grid are highlighted. This grid consists of $151 \times 39 \times 39$ grid points (about 230,000). An additional plane has been added inboard of the symmetry plane to facilitate the symmetry plane boundary conditions.

The grid shown in Fig. 1 was constructed in the following manner. First, a surface grid for the wing was generated with the S3D code.²² Additional grid clustering was provided near the wing nose and shocks. Then, a three-dimensional grid was created with 3DGRAPE,²³ an elliptic grid generation program, with normal spacing to the wing surface appropriate for inviscid computations. Finally, viscous clustering was obtained by redistributing grid points along coordinate lines normal to the body with a clustering function described by Vinokur.²⁴ This one-dimensional stretching function provides a distribution of points that minimize the truncation error of a finite difference code.

A two-zone grid was also constructed by splitting the single-zone grid, see Fig. 1, in the ζ direction (body normal). A symmetry plane view is shown in Fig. 2 with the shaded region indicating the one cell overlap between zones. Zone 1 is adjacent to the wing surface and zone 2 extends from zone 1 to the far field, about six chord lengths away. The grid points at the zone 1 and zone 2 interfaces match up exactly. This was done in order to study the effect of the zonal boundary treatment on time accuracy without the additional complication of flow variable interpolation at zonal interfaces.

A three-zone grid was constructed from the two-zone grid by splitting zone 2 in the ζ direction, see Fig. 3. The third zone is the outermost zone connecting zone 2 to the far field. Notice that every other grid point in the ξ direction (streamwise) has been removed. This was done in order to demonstrate the capability of the NSS code to test grid interfaces that do not have identical grid points.

Boundary Conditions

The NSS code solves the Euler or Navier-Stokes equations depending on the specific application. For inviscid flow, flow tangency is imposed on the wing surface while density and pressure are found by extrapolation. The total energy per unit volume is then computed from the perfect gas law. The same

boundary conditions apply for viscous flow except the no-slip condition replaces flow tangency on the wing surface. Uniform flow is imposed at the far field, and zero gradient conditions are used at the outflow boundaries. Boundary values along the C cut (downwind of the wing trailing edge and outboard of the wingtip) are obtained by averaging flow variables across the cut. Flow symmetry is obtained by reflecting the flow variables about the symmetry plane in an explicit manner.

The zonal interface boundary conditions have already been described in the Zonal Approach section.

Results

Wind-Tunnel Model

All flow computations presented in this section correspond to a rectangular half-wing used in an AGARD wind-tunnel experiment described by Mabey et al.²⁵ The wing model was mounted on a fuselage-like body to displace it slightly from the wind-tunnel wall and its boundary layer. No experimental data was taken at the wind-tunnel wall (wing symmetry plane). The wing's cross section consists of a NACA 64A010 airfoil whose thickness/chord ratio has been modified to 10.6%. The wing has a complete aspect ratio = 4.0. The experiment was conducted in the Royal Aeronautical Establishment 8- × 8-ft tunnel with closed walls. (The model was originally designed for a smaller tunnel that was not available when the experiment was conducted.) Various C-H grids were generated for this wing, see for example, Fig. 1. No attempt was made to model the wind-tunnel test section.

Steady Flow Computations

Steady-state flow computations with different grid densities are used to study their effect on the spatial accuracy of the NSS code. The three grids that are used are 1) a coarse grid ($101 \times 39 \times 39$), 2) a medium grid ($151 \times 39 \times 39$), and 3) a fine grid ($233 \times 39 \times 39$). Each grid has the same grid point distribution in the spanwise and body-normal directions (η and ζ directions). The streamwise distribution of points (ξ direction) downwind of the wing trailing edge are also identical for all three grids (17 grid cells to the far field). However, each grid has 50% more points on the wing surface (in the streamwise direction) than its previous coarser grid. Grid points are clustered near the wing leading edge and shock locations to improve the flowfield resolution on each grid. The grid spacing in the viscous direction is $\Delta\zeta = 5.0 \times 10^{-6}$ chords.

Figure 4 shows a comparison between steady Navier-Stokes pressure coefficients C_p computed on the three grids just described with experimental values at $\eta = 50, 77$, and 94% semispan locations. These computations correspond to a freestream Mach number $M_\infty = 0.80$, angle of attack $\alpha = 0.0$ deg, and a freestream Reynolds number based on the wing root chord $Re = 2.4 \times 10^6$. At $\eta = 50\%$, Fig. 4a, the computed pressure coefficients compare well with each other and the experimental data, with the exception that the coarse grid shock is slightly smeared. The suction peaks are also slightly underpredicted, presumably due to wind-tunnel wall effects. All three grids have good C_p comparisons with each other and experiment at $\eta = 77\%$. At $\eta = 94\%$, computations with all three grids are in good agreement with each other, but their suction peaks differ somewhat from the experimental value. This is probably due to the use of a rounded wingtip in the computation vs a blunt (flat) tip in the wind-tunnel model. Overall, the medium and fine grids give the best C_p comparisons with the experiment and do not differ noticeably from each other. Therefore, the medium grid density (about 230,000 grid points) is used for the remainder of the viscous computations that are presented.

Unsteady Flow Computations

Time-accurate Euler and Navier-Stokes computations for a rigid wind with oscillating angle of attack are now presented.

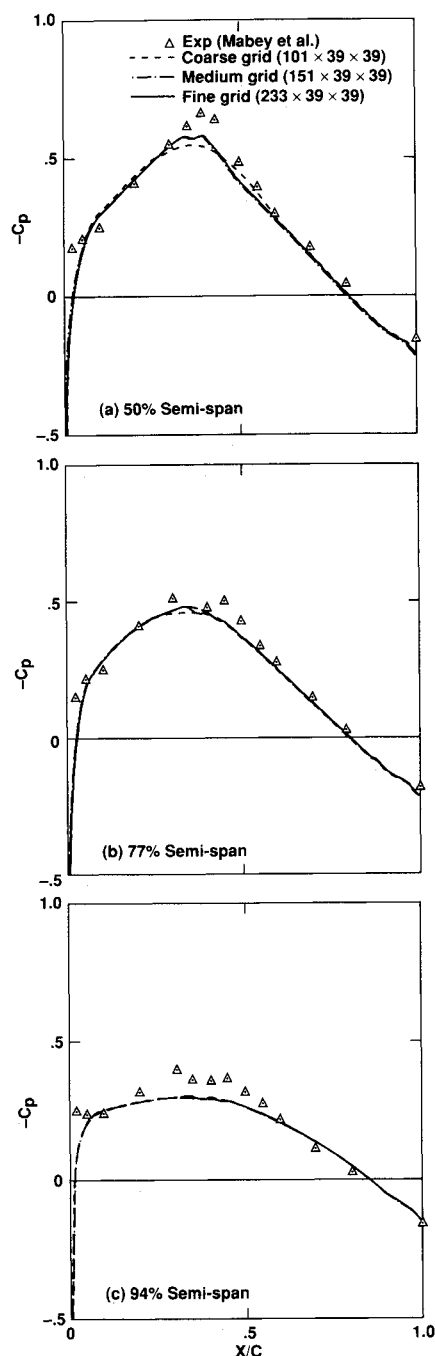


Fig. 4 Comparison of steady Navier-Stokes pressure coefficients (computed with different grid sizes) and experiment— $M_\infty = 0.80$, $\alpha = 0.0$ deg, $Re = 2.4 \times 10^6$: a) 50% semispan; b) 77% semispan; c) 94% semispan.

Unless otherwise noted, the freestream Mach number is $M_\infty = 0.80$ and the Reynolds number based on wing chord is $Re = 2.4 \times 10^6$. The angle of attack varies periodically according to

$$\alpha(t) = -\alpha_{\max} \sin(M_\infty k t)$$

where the amplitude is $\alpha_{\max} = 1.0$ deg, the reduced frequency is $k = 0.27$, and t is a nondimensional time based on wing chord and the freestream speed of sound. The reduced frequency is defined by $k = \omega c / U_\infty$, where ω is the circular frequency (rad/s), c the wing chord length, and U_∞ the freestream velocity. The computational grid undergoes a rigid body rotation according to the angle-of-attack formula. Real and imaginary unsteady pressure coefficients are used to compare computational and experimental results. These in-phase

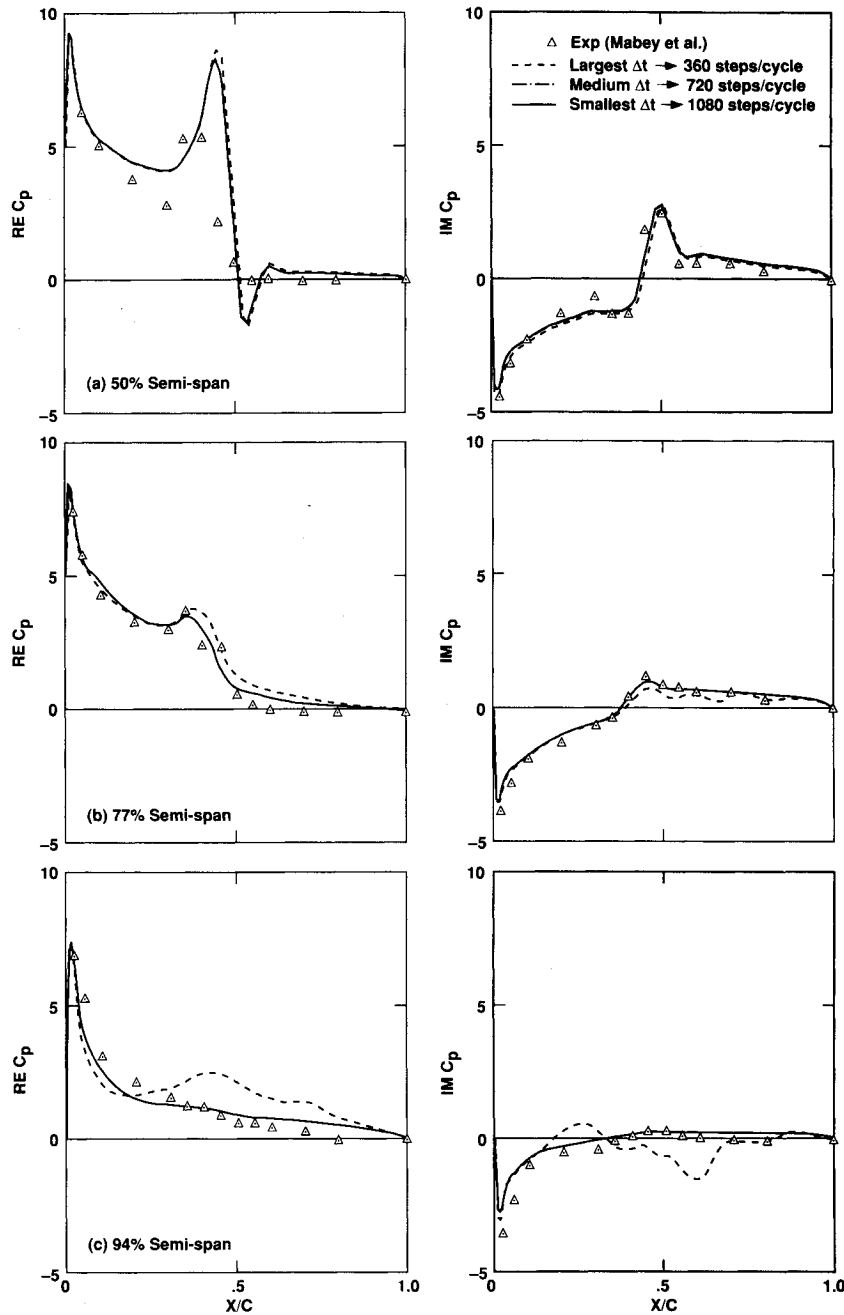


Fig. 5 Comparison of Euler pressure coefficients with different computational time steps and experiment— $M_\infty = 0.80$, $\alpha_{\max} = 1.0$ deg, $k = 0.27$, $\alpha(t) = -\alpha_{\max} \sin(M_\infty k t)$: a) 50% semispan; b) 77% semispan; c) 94% semispan.

and out-of-phase pressures correspond to the time-dependent Fourier coefficients normalized by $\alpha_{\max}$ (rad).

The time steps used in Navier-Stokes computations are usually determined by stability considerations rather than time accuracy. Therefore, the Euler equations are used to determine the time-step requirement for time accuracy with the first-order accurate diagonal algorithm. For these inviscid computations, the medium density viscous grid in Fig. 1 is unclustered by redistributing 21 grid points along coordinate lines normal to the wing surface. The inviscid and viscous grids have identical spacing in the spanwise and streamwise directions. Figure 5 compares computed real and imaginary pressures for three different time steps with experimental data at three semispan locations. These time steps, beginning from the largest to the smallest, are 360, 720, and 1080 time steps/cycle, respectively. Pressure coefficients computed with the medium and smallest time steps are virtually indistinguishable from one another. However, significant differences between

the largest time step (360 steps/cycle) and the smaller time steps can be noted, especially at $\eta = 77$ and 94%. Therefore, 720 time steps/cycle is the largest usable time step for time accuracy.

The real and imaginary pressure coefficients for the Euler equations, Navier-Stokes equations, and experimental data are compared in Fig. 6. These pressures correspond to the same freestream conditions and reduced frequency in Fig. 5. At 50% semispan location, the real C_p shock position and strength computed by the Navier-Stokes equations are in good agreement with the experiment. Reducing the time step in half or using a two-step Newton subiteration has no perceivable effect on any of the pressure coefficients. This indicates that the temporal conservation error is negligible for this transonic-flow example. It is evident from this figure that the shock moves about 25% of a chord length during a complete cycle. This provides a good test for the time accuracy of the diagonal algorithm. The Euler equations, on the other hand,

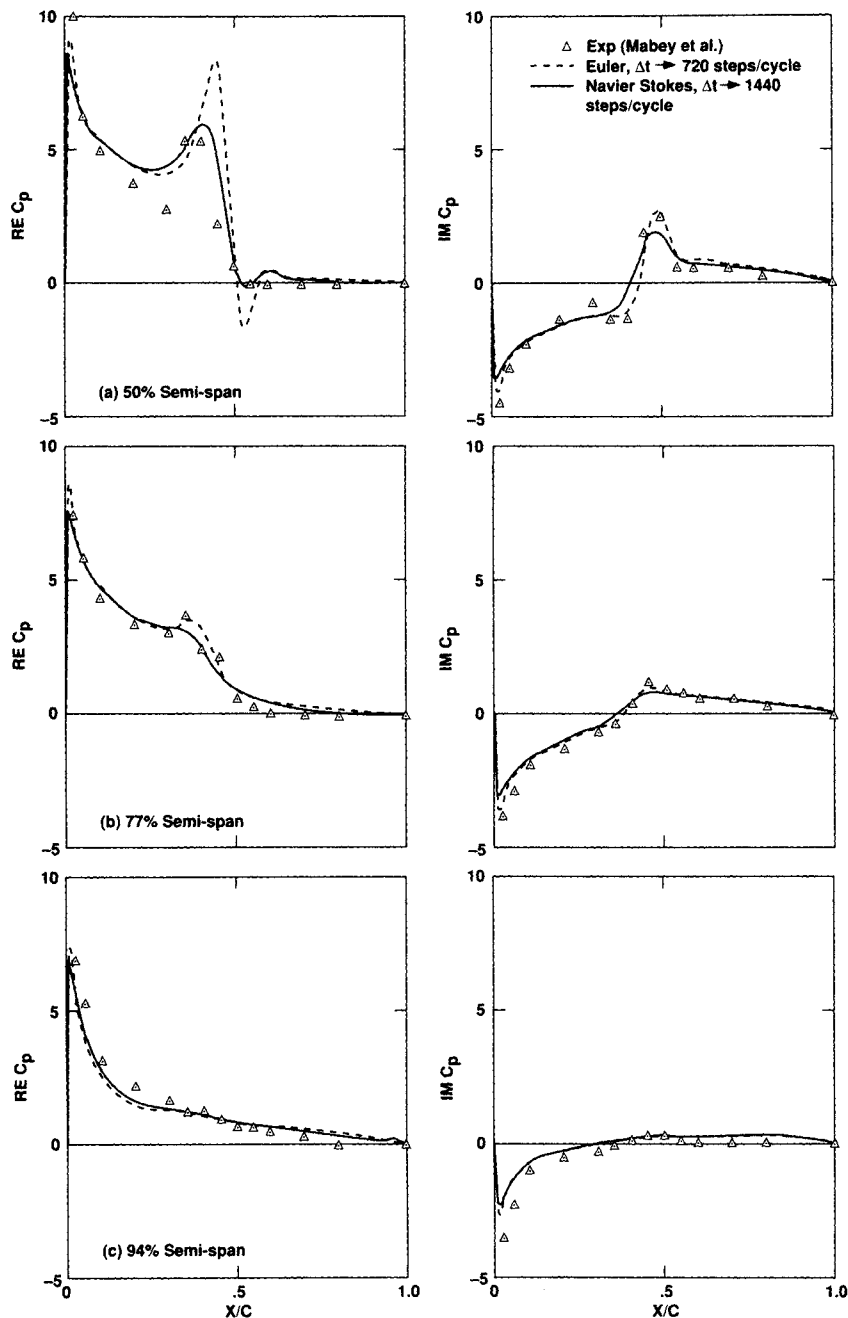


Fig. 6 Comparison of Euler pressure coefficients with Navier Stokes and experiment— $M_\infty = 0.80$, $Re = 2.4 \times 10^6$, $\alpha_{\max} = 1.0$ deg, $k = 0.27$, $\alpha(t) = -\alpha_{\max} \sin(M_\infty k t)$: a) 50% semispan; b) 77% semispan; c) 94% semispan.

predict a much stronger shock with a slightly downwind position than the experiment and the Navier-Stokes equations. Even for this relatively easy case that does not have any significant separation, the Euler equations would predict the wrong wave drag. The drop in pressure before the shock is not predicted very well by either the Euler or Navier-Stokes equations. This is attributed to inadequate grid resolution. Although the clustered grid gave good shock resolution for the steady-state case, there is inadequate grid resolution in the shock excursion region. With hindsight, it is obvious that time-dependent flows involving moving shocks require greater grid resolution than steady-state flows. This can be accomplished by adding more grid points in the shock excursion region or using a solution adaptive technique. The latter would be more efficient.

The real and imaginary pressure coefficients computed by the Euler and Navier-Stokes equations compare well with each other and the experimental data for $\eta = 77$ and 94% semispan locations. The time step used in the Navier-Stokes

computations at $\Delta t = 0.02$, or 1400 steps/cycle. Although this time step was chosen to maintain code stability, it is only one half of the time step required for time accuracy in the Euler computations. This is very encouraging, especially since the viscous grid spacing normal to the wing was two orders of magnitude smaller than the Euler grid spacing. This indicates the robustness of the diagonal algorithm.

In order to verify the time accuracy of the zonal boundary conditions, the single grid, shown in Fig. 1, is split into two zones, shown in Fig. 2. The first zone is close to the wing surface, whereas the second zone extends from the first zone to the far field. The Navier-Stokes equations are solved in the first zone and the Euler equations are solved in the second zone. The zones overlap by one grid cell as indicated by the shaded region in Fig. 2. The interface surfaces between the zones have identical grid points. The zonal overlap position was determined as the closest inviscid position, i.e., where the turbulent eddy viscosity is zero. The close proximity of the zonal interface to the wing surface and leading edge pro-

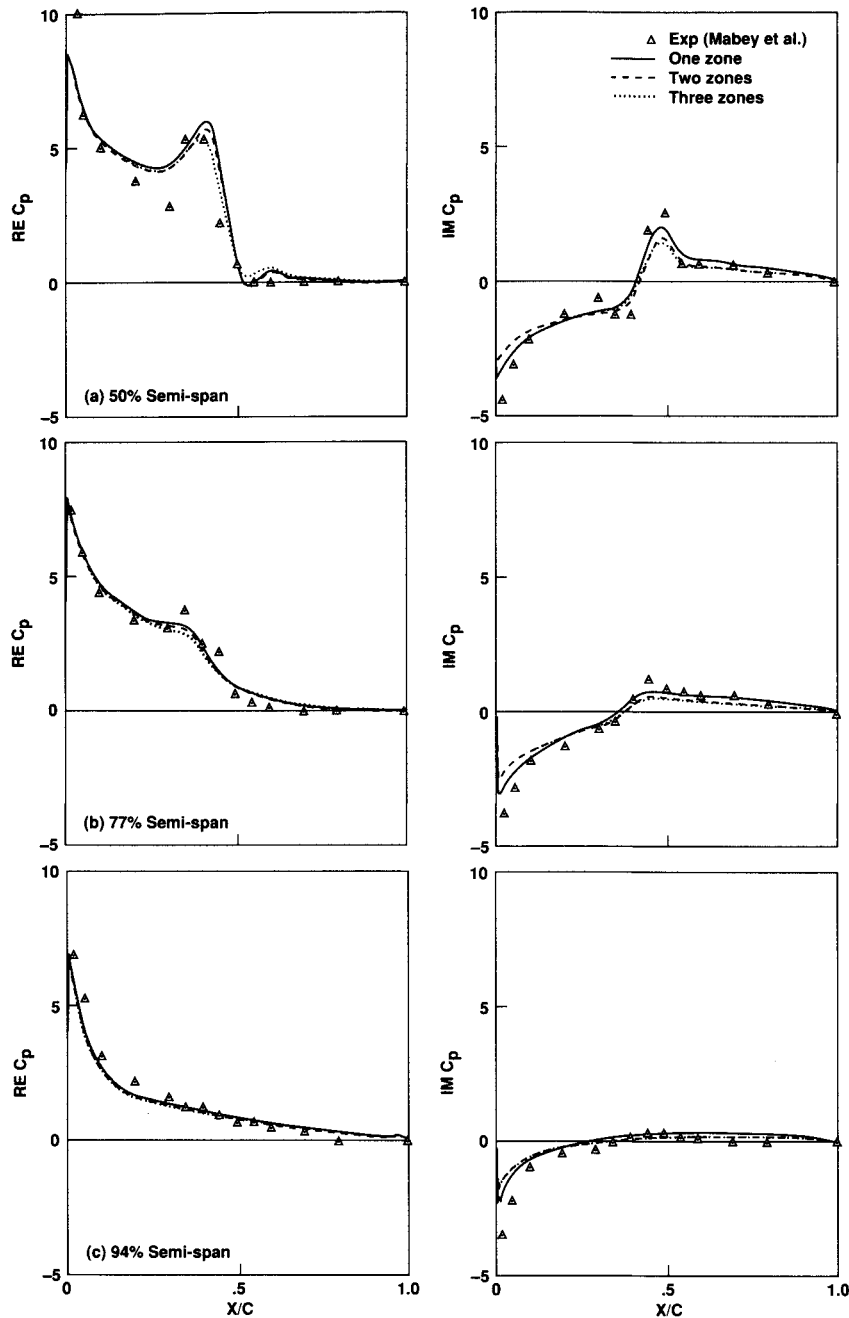


Fig. 7 Comparison of pressure coefficients for one-, two-, and three-zone Navier-Stokes computations with experiment— $M_\infty = 0.80$, $Re = 2.4 \times 10^6$, $\alpha_{\max} = 1.0$ deg, $k = 0.27$, $\alpha(t) = -\alpha_{\max} \sin(M_\infty kt)$: a) 50% semispan; b) 77% semispan; c) 94% semispan.

vides an extreme test for time accuracy of the zonal boundary condition treatment. A three-zone grid, shown in Fig. 3, is also used to demonstrate the capability of the NSS code to treat zonal interfaces in which all grid points do not coincide. Zone 3, the outer grid, has every other grid point removed in the streamwise direction. The Navier-Stokes equations are applied in zone 1, near the body, and the Euler equations in the other two zones, away from the body.

Figure 7 compares the real and imaginary pressure coefficients for the single-, two-, and three-zone Navier-Stokes computations with the experimental data. The single-zone computation applied the Navier-Stokes equations and Euler equations in the same region as the zonal cases. Once again, the time step used for these viscous computations was $\Delta t = 0.02$. Overall, the comparison of the single- and two-zone computations are very close. Slight differences can be noted in the imaginary pressure coefficients. Reducing the time step in half does not alter the results. This indicates that these slight differences are not due to the partially explicit zonal

boundary conditions. These differences are more likely due to differences in numerical dissipation. Although the diagonal algorithm uses fourth-order dissipation, it must drop to second order near grid boundaries. This means that the single-zone computation uses fourth-order dissipation in the zonal overlap region, whereas the two-zone computation uses second-order dissipation there. The close proximity of the zonal boundary to the wing surface and leading edge can easily account for these differences. Note that both solutions compare well with the experimental data. The real and imaginary C_p for the three-zone computation differs only slightly from the two-zone computation at the 50% semispan location. They are virtually identical at the other two semispan locations. These slight differences are most likely due to the different truncation errors of the zonal grid systems. Recall that zone 3, of the three-zone system, is twice as coarse as zone 2, of the two-zone system (see Figs. 2 and 3). Overall, the zonal grid approach is capable of simulating time-dependent flow with a moving grid system.

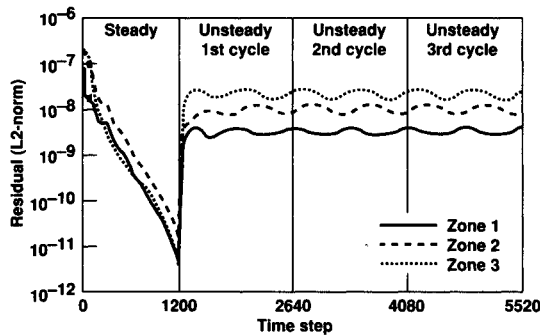


Fig. 8 Convergence history of steady and unsteady three-zone Navier-Stokes computation— $M_\infty = 0.80$, $Re = 2.4 \times 10^6$, $\alpha_{\max} = 2.0$ deg, $k = 0.27$, $\alpha(t) = -\alpha_{\max} \sin(M_\infty kt)$.

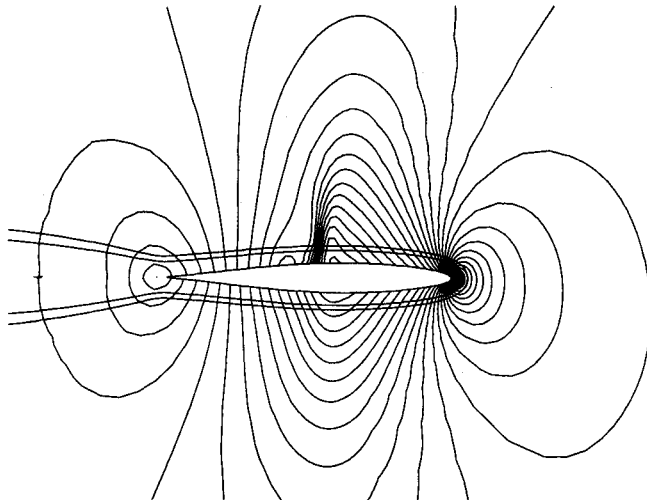


Fig. 9 Instantaneous pressure coefficient contours at the symmetry plane near $\alpha_{\max}$. Two-zone Navier-Stokes computations with $M_\infty = 0.80$, $Re = 2.4 \times 10^6$, $\alpha_{\max} = 2.0$ deg, $k = 0.27$, $\alpha(t) = -\alpha_{\max} \sin(M_\infty kt)$.

Figure 8 shows the convergence history of the three-zone computation. A steady solution was first obtained for the mean angle of attack ($\alpha = 0.0$ deg). A reduction of the L_2 norm of the residual by four orders of magnitude was obtained in 1200 time steps. A variable time step was used, according to Eq. (8), to accelerate the rate of convergence. The steady solution was then used as a starting point for the time-accurate oscillating wing case. The flow was completely periodic in the second cycle. This was verified by computing a third cycle and comparing unsteady pressure coefficients. When operating in steady-state mode, the NSS code requires about $14 \mu\text{s}/\text{grid point}/\text{time step}$ on a Cray Y-MP supercomputer. When operating in a time-accurate mode, with a forced periodic grid motion, it requires about $17 \mu\text{s}/\text{grid point}/\text{time step}$. Overall, a complete time-accurate computation for the present example requires about 4 h of Cray Y-MP time. This includes the steady solution and two cycles of forced periodic motion.

A final example of a time-accurate zonal computation is shown in Fig. 9. The same freestream conditions and grids of the previous two-zone example are used, except now $\alpha_{\max} = 2.0$ deg. Although experimental data is not available for this case, the stronger shock provides a qualitative check of the zonal boundary treatment. Figure 9 indicates instantaneous pressure coefficient contours when the upper surface shock is its strongest. The zonal overlap region is also indicated in the figure. Notice that the C_p contours are extremely smooth across the zonal boundaries, even near the shock and the wing leading edge. This shows that there is no significant flow disturbances introduced by the zonal boundary conditions.

Conclusions

The zonal, steady-state, Transonic Navier-Stokes code has been extended to treat unsteady flows. This new Navier-Stokes Simulation code has the capability of treating a moving zonal grid system. The Pulliam-Chaussee diagonal algorithm has been shown to be an efficient and robust time-dependent algorithm. The NSS code requires approximately $17 \mu\text{s}/\text{grid point}/\text{time step}$ on a Cray Y-MP supercomputer for a time-dependent flow with a forced periodic grid motion. A typical time-accurate result for 230,000 grid points requires about 4 h of computer time. Although this algorithm is conservative in space, it has a temporal conservation error $O(\Delta t)$. Since Navier-Stokes algorithms require small time steps for numerical stability, this error should be small for a large class of transonic applications.

The NSS code has been used to simulate unsteady transonic viscous flow about a three-dimensional rectangular wing with oscillating angle of attack. Computed real and imaginary pressure coefficients compared well with the experimental values and demonstrates the time accuracy of the diagonal algorithm. The temporal conservation error was shown to have no noticeable effect on the speed, strength, and position of the shock wave (for the present computations) through reduction of the time step and application of a Newton-iteration procedure. Two- and three-zone transonic computations for the oscillating wing compared well with the single-grid computation and the experimental data. This demonstrates the time-accurate capability of the zonal interface procedure. The zonal grid approach provides a way of simulating time-dependent flow about a complete aircraft with the Navier-Stokes equations.

References

- ¹Guruswamy, G. P., Goorjian, P. M., Ide, H., and Miller, G. D., "Transonic Aeroelastic Analysis of the B-1 Wing," *Journal of Aircraft*, Vol. 23, No. 7, 1986, pp. 547-553.
- ²Farmer, M. G., and Hanson, P. W., "Comparison of Supercritical and Conventional Wing Flutter Characteristics," NASA TM X-72837, May 1976.
- ³Ashley, H., "Role of Shocks in the Sub-Transonic Flutter Phenomenon," *Journal of Aircraft*, Vol. 17, No. 3, 1980, pp. 187-197.
- ⁴Guruswamy, G. P., and Goorjian, P. M., "Unsteady Transonic Flow Simulation on a Full-Span Wing-Body Configuration," AIAA Paper 87-1240, Jun. 1987.
- ⁵Kutler, P. and Gross, A. R., "CFD in Design: A Government Perspective," AIAA Paper 89-0094, Jan. 1989.
- ⁶Goldhammer, M. I., and Rubbert, P. E., "CFD in Design: An Airframe Perspective," AIAA Paper 89-0092, Jan. 1989.
- ⁷Nakamichi, J., "Calculations of Unsteady Navier-Stokes Equations Around an Oscillating 3-Wing Using Moving Grid System," AIAA Paper 87-1158, Jun. 1987.
- ⁸Srinivasan, G. R., and McCroskey, W. J., "Navier-Stokes Calculations of Hovering Rotor Flow Fields," *Journal of Aircraft*, Vol. 25, No. 10, 1988, pp. 865-874.
- ⁹Guruswamy, G., "Numerical Simulation of Vortical Flows on Flexible Wings," AIAA Paper 89-0537, Jan. 1989.
- ¹⁰Guruswamy, G., "Vortical Flow Computations on Swept Flexible Wings Using Navier-Stokes Equations," AIAA Paper 89-1183, Apr. 1989.
- ¹¹Flores, J., and Chaderjian, N. M., "A Zonal Navier-Stokes Methodology for Flow Simulation About a Complete Aircraft," *Journal of Aircraft*, Vol. 27, No. 7, 1990, pp. 583-590.
- ¹²Pulliam, T. H., and Chaussee, D. S., "A Diagonal Form of An Implicit Approximate-Factorization Algorithm," *Journal of Computational Physics*, Vol. 39, No. 2, 1981, pp. 347-363.
- ¹³Beam, R. M., and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," *Journal of Computational Physics*, Vol. 22, No. 1, 1976, pp. 87-110.
- ¹⁴Baldwin, B. S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flow," AIAA Paper 78-257, Jan. 1978.
- ¹⁵Chaderjian, N. M., "Transonic Navier-Stokes Wing Solutions Using a Zonal Grid Approach: Part 2. High Angle-of-Attack Simulation," AGARD CP 412, *Applications of Computational Fluid Dy-*

namics in Aeronautics, Paper 30B, Nov. 1986; also NASA TM 88248, April 1986.

¹⁶Pulliam, T. H., and Steger, J. L., "Implicit Finite Difference Simulations of Three-Dimensional Compressible Flow," *AIAA Journal*, Vol. 18, No. 2, 1980, p. 159.

¹⁷Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations," AIAA Paper 85-0438, Jan. 1985.

¹⁸Srinivasan, G. R., Chyu, W. J., and Steger, J. L., "Computation of Simple Three-Dimensional Wing-Vortex Interaction in Transonic Flow," AIAA Paper 81-1206, Jun. 1981.

¹⁹Flores, J., "Convergence Acceleration for a Three-Dimensional Euler/Navier-Stokes Zonal Approach," *AIAA Journal*, Vol. 24, No. 9, 1986, pp. 1441-1442.

²⁰Chakravarthy, S. R., "Relaxation Methods for Unfactored Im-

plicit Upwind Schemes," AIAA Paper 84-0165, Jan. 1984.

²¹Flores, J., Chaderjian, N. M., and Sorenson, R. L., "Simulation of Transonic Viscous Flow Over Fighter-Like Configuration Including Inlet," AIAA Paper 87-1199, June 1987.

²²Luh, R. C. C., "Surface Grid Generation for Complex Three-Dimensional Geometries," NASA TM 101046, Oct. 1988.

²³Sorenson, R. L., "The 3DGRAPE Book: Theory, Users' Manual, Examples," NASA TM 102224, July 1989.

²⁴Vinokur, M., "On One-Dimensional Stretching Functions for Finite Difference Calculations," NASA CR 3313, Oct. 1980.

²⁵Mabey, D. G., Welsh, B. L., and Pyne, C. R., "A Summary of Measurements of Steady and Oscillatory Pressures on a Rectangular Wing," *Aeronautical Journal of the Royal Aeronautical Society*, Vol. 92, No. 911, Jan. 1988, pp. 10-28.

Recommended Reading from the AIAA Education Series

Critical Technologies for National Defense

J. S. Przemieniecki, Editor-In-Chief
Prepared by Air Force Institute of Technology (AFIT)

Critical Technologies for National Defense discusses the underlying physical and engineering principles governing the development of our future defense systems as they relate to the 20 critical technologies that have been identified in the 1990 DoD Critical Technologies Plan. Physical and Engineering Principles, Description of Technology, and Impact on Future Weapon Systems are discussed for each critical technology.

This text is recommended reading for senior managers who direct the development of defense and weapon systems. Topics include: Computational Fluid Dynamics; Simulation and Modeling; Composite Materials; Signature Control; Software Producibility; Biotechnology Materials and Processes; and more.

1991, 318 pp, illus, Hardcover

ISBN 1-56347-009-8

AIAA Members \$36.50 • Nonmembers \$46.95

Order #: 09-8 (830)

Place your order today! Call 1-800/682-AIAA



American Institute of Aeronautics and Astronautics

Publications Customer Service, 9 Jay Gould Ct., P.O. Box 753, Waldorf, MD 20604
Phone 301/645-5643, Dept. 415, FAX 301/843-0159

Sales Tax: CA residents, 8.25%; DC, 6%. For shipping and handling add \$4.75 for 1-4 books (call for rates for higher quantities). Orders under \$50.00 must be prepaid. Please allow 4 weeks for delivery. Prices are subject to change without notice. Returns will be accepted within 15 days.