JOURNAL OF AIRCRAFT
Vol. 47, No. 1, January–February 2010

Parallel Multigrid Algorithm for Aeroelasticity Simulations

Paul G. A. Cizmas,* Joaquin I. Gargoloff,† and Thomas W. Strganac*
Texas A&M University, College Station, Texas 77843-3141
and
Philip S. Beran‡
U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio 45433-7531

DOI: 10.2514/1.40201

This paper presents the development of a multigrid parallel algorithm for a nonlinear aeroelastic analysis. The aeroelastic model consists of 1) a nonlinear structural model that captures in-plane, out-of-plane, and torsional couplings; 2) an unsteady viscous aerodynamic model that captures compressible flow effects for transonic flows with shock/boundary-layer interaction; and 3) a solution methodology that assures a tightly coupled solution of the nonlinear structure and the fluid flow, including a consistent geometric interface between the highly deforming structure and the flowfield. A domain-decomposition parallel computation algorithm based on a message-passing interface was developed for the flow solver. A three-level multigrid algorithm was implemented in the flow solver to further reduce the computational time. A grid generation and deformation algorithm was developed concurrently with the flow solver in order to improve the efficiency of the computation. The grid deformation methodology kept the mesh topology unchanged as the structure deformed. Consequently, it was not necessary for either the parallel computation or the multigrid algorithm to update their communication pointers while the structure deformed. The validation of the numerical solver was done using experimental results of the F-5 wing. The aeroelastic solver was then used to assess the effect of structural nonlinearities on the aeroelastic response of the heavy Goland wing.

I. Introduction

IN RECENT years, studies of nonlinear fluid–structure interactions have been motivated by evidence that there are adverse aeroelastic responses attributed to system nonlinearities. For example, limit-cycle oscillations (LCOs) occur in nonlinear aeroelastic systems and remain a persistent problem on fighter aircraft with store configurations. Nonlinear phenomena such as LCOs have been observed on fighter aircraft such as the F-5, F-15 STOL, F-16, F-111, and F/A-18 [1–3]. LCO has been observed, however, not only on fighter aircraft but also on a generic business jet wind-tunnel flutter model [4] and on the B-2 bomber [5]. In the latter case, residual pitch oscillations, a persistent small-amplitude oscillation involving wing-bending interaction with rigid-body pitching, have been found during flight tests.

Such LCOs are unacceptable since aircraft performance, aircraft certification, mission capability, and human-factor issues such as pilot fatigue are adversely affected. Unfortunately, the ability to fully analyze and optimally design air vehicles with inherent nonlinear features is limited by the computational cost. The frequencies associated with wing flutter are much smaller than those found normally in turbomachinery. These lower frequencies require a long integration time, sometimes of the order of tens of seconds, to properly capture the physics of the problem. Furthermore, and as a consequence of this limitation, the physics of nonlinear responses are not fully understood. For example, while there is little disagreement in the flutter engineering community that LCO arises from the nonlinear structural and/or aerodynamic forces, there is significant disagreement as to which of these sources is the major contribution to the phenomenon.

Most of the computational effort in a high-fidelity aeroelastic simulation is dedicated to flow simulation. Several aerodynamic models that use the Euler/Navier–Stokes equations have been used to date in aeroelastic applications [6–14]. The computational cost and turnaround time of the time-marching aerodynamic models tend to be prohibitive, even when parallel processing is being used [15,16].

The high-fidelity aeroelastic simulations model the flow motion using the Navier–Stokes equations. These flow solvers are either structured or unstructured, depending on the approach taken to discretize the computational domain. Structured solvers are usually less complex and more efficient than unstructured solvers. Unstructured solvers have the advantage of being more flexible in handling arbitrary geometries. Structured Navier–Stokes flow solvers include the Euler/Navier–Stokes 3D Aero-Elasticity (ENS3DAE) code [6] and the Computational Fluids Laboratory 3-Dimensional (CFL3D) flow solver [8]. Unstructured Navier–Stokes solvers include the U.S. Air Force Air Vehicles Unstructured Solver (AVUS) [17].

The objective of this paper is to present the methodology for generating a nonlinear aeroelastic analysis code that consists of 1) a nonlinear structural model, 2) a viscous unsteady aerodynamic model, and 3) a tightly coupled solution methodology, including a consistent geometric interface between the highly deforming structure and the flowfield. The first part of this paper introduces the aeroelastic model, presenting the details of the aerodynamic and structural models and their aeroelastic coupling. The second part presents the numerical method, including the grid generation and deformation algorithms, the flow solver, the parallel strategy employed, and the multigrid implementation. The paper concludes with the results for the F-5 wing and the heavy Goland wing.

II. Aeroelastic Model

The first part of this section presents the aerodynamic model, which solved the Reynolds-averaged Navier–Stokes equations, and a two-equation eddy-viscosity turbulence model. The second part describes the nonlinear structural model that modeled the wing as a nonlinear beam. The third part describes the coupling of the aerodynamic and structural models.

Presented as Paper 330 at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 7–9 January 2007; received 1 August 2008; revision received 29 July 2009; accepted for publication 11 August 2009.

*Professor, Department of Aerospace Engineering, TAMU 3141. Associate Fellow AIAA.

†Graduate Research Assistant, Department of Aerospace Engineering, TAMU 3141. Member AIAA.

‡Principal Research Aerospace Engineer, Building 146, 2210 Eighth Street. Associate Fellow AIAA.

A. Aerodynamic Model

The flow around the deforming wing was modeled as unsteady, compressible, and viscous using the mass, momentum, and energy conservation equations. These equations are collectively known as the Navier–Stokes equations:

$$\frac{\partial}{\partial t}\int_V \boldsymbol{Q}\cdot \mathrm{d}V + \oint_S (\mathbf{F}_{\text{conv}} - \mathbf{F}_{\text{vis}})\cdot \mathrm{d}S = \int_V \mathbf{G}\cdot \mathrm{d}V \tag{1}$$

where $\mathbf{Q}$ is the state vector of conservative variables, $\mathbf{F}_{\text{conv}}$ is the vector of convective fluxes, $\mathbf{F}_{\text{vis}}$ is the vector of viscous fluxes, and $\mathbf{G}$ is the vector of source terms. The state vector of conservative variables is $\mathbf{Q} = (\rho, \rho u, \rho v, \rho w, \rho E)^T$, where ρ is the density of the fluid; u, v and w are the x, y, and z components of the velocity; and E is the total energy of the fluid particle. The vector of convective fluxes is

$$\mathbf{F}_{\text{conv}} = (\rho V, \rho u V + n_x p, \rho v V + n_y p, \rho w V + n_z p, \rho H V)^T$$

where V is the contravariant velocity and H is the total enthalpy. The vector of viscous fluxes is

$$\mathbf{F}_{\text{vis}} = (0, f_x, f_y, f_z, u_k f_k - q_n)^T$$

where f_x, f_y, and f_z are the viscous stresses in the x, y, and z directions, and q_n is the heat conduction in the fluid.

The turbulence effects were modeled by using the two-equation eddy-viscosity shear stress transport model [18]. The time-dependent integral form of these equations was expressed in a vectorial form similar to Eq. (1). The state vector of turbulent conservative variables is $\mathbf{Q}_T = (\rho k, \rho\omega)^T$, where k is the turbulence kinetic energy and ω is the specific dissipation rate.

B. Structural Model

The wing shown in Fig. 1 was modeled as a nonlinear beam that could bend about the y and z axes and twist about the ξ axis. The (x, y, z) axes are inertial and the (ξ, η, ζ) axes are the principal axes of the beam's cross section at position s. The governing equations contained structural coupling terms and included both quadratic and cubic nonlinearities due to curvature and inertia [19].

The equations of motion for in-plane bending w, out-of-plane bending v, and torsion ϕ response were written as

$$m\ddot{w} - j_\eta \ddot{w}'' + D_\eta w^{IV} = G_w^{*\prime} + F_{A_w} \tag{2}$$

$$m\ddot{v} - me\ddot{\phi} - j_\zeta \ddot{v}'' + D_\zeta v^{IV} = G_v^{*\prime} + F_{A_v} \tag{3}$$

$$j_\xi \ddot{\phi} - me\ddot{v} - D_\xi \phi'' = G_\phi + M_A \tag{4}$$

where m is the mass of the wing per unit span; j_ξ, j_η, and j_ζ are the principal mass moments of inertia of the beam per unit span; D_ξ, D_η, and D_ζ are the beam principal stiffnesses; e is the center of gravity offset from the elastic axis; F_{A_w} and F_{A_v} are the aerodynamic forces per unit span; M_A is the aerodynamic moment per unit span; and G_v^*, G_w^*, and G_ϕ are the structural nonlinear components:

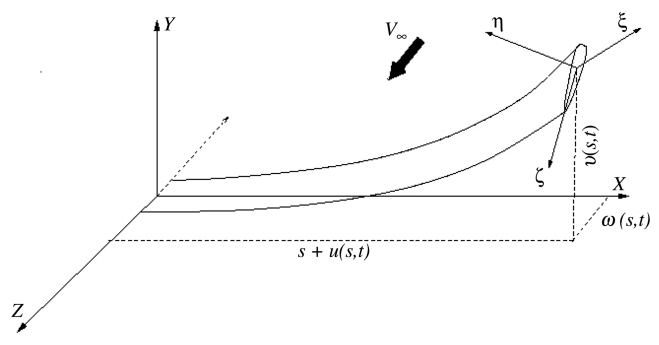

Fig. 1 Coordinate systems for the wing structural model.

$$\begin{aligned} G_v^* = {} & -D_\xi(\phi' + v''w')w'' - (D_\eta - D_\zeta)(2\phi\phi' v'' + \phi^2 v''' \\ & - \phi' w'' - \phi w''' - v''w'w'' - v'w''w'' - v'w'w''') \\ & - v'(D_\zeta v''^2 + D_\eta w''^2) + j_\xi(\dot{\phi} + \dot{v}'w')\dot{w}' \\ & + (j_\eta - j_\zeta)(2\phi\dot{\phi}\dot{v}' + \phi^2\ddot{v}' - \dot{\phi}\dot{w}' - \phi\ddot{w}' - \dot{v}'w'\dot{w}' \\ & - v'\dot{w}'\dot{w}' - v'w'\ddot{w}') + v'(j_\zeta \dot{v}'^2 + j_\eta \dot{w}'^2) - D_\zeta v'^2 v''' \\ & - D_\eta w'w'''v' + j_\zeta \ddot{v}'v'^2 + j_\eta \ddot{w}'w'v' + v'^2 w' M_A \\ & - v'\int_L^y \left\{\frac{1}{2}m\frac{\partial^2}{\partial t^2}\int_0^y (v'^2 + w'^2)\,\mathrm{d}y\right\}\mathrm{d}y + w'M_A \end{aligned} \tag{5}$$

$$\begin{aligned} G_w^* = {} & D_\xi(\phi' + v''w')v'' + (D_\eta - D_\zeta)(\phi' v'' + \phi v''' + 2\phi\phi' w'' \\ & + \phi^2 w''') - w'(D_\eta w''^2 + D_\zeta v''^2) - j_\xi(\dot{\phi} + \dot{v}'w')\dot{v}' \\ & - (j_\eta - j_\zeta)(\dot{\phi}\dot{v}' + \phi\ddot{v}' + 2\phi\dot{\phi}\dot{w}' + \phi^2\ddot{w}') + w'(j_\eta \dot{w}'^2 + j_\zeta \dot{v}'^2) \\ & - D_\zeta w'v'v''' - D_\eta w'^2 w''' + j_\zeta \ddot{v}'v'w' + j_\eta \ddot{w}'w'^2 + v'w'^2 M_A \\ & - w'\int_L^y \left\{\frac{1}{2}m\frac{\partial^2}{\partial t^2}\int_0^y (v'^2 + w'^2)\,\mathrm{d}y\right\}\mathrm{d}y \end{aligned} \tag{6}$$

$$\begin{aligned} G_\phi = {} & D_\xi(w''v'' + w'v''') - (D_\eta - D_\zeta)[(v''^2 - w''^2)\phi - v''w''] \\ & - j_\xi(\dot{w}'\dot{v}' + w'\ddot{v}') + (j_\eta - j_\zeta)[(\dot{v}'^2 - \dot{w}'^2)\phi - \dot{v}'\dot{w}'] \end{aligned} \tag{7}$$

G_v^* and G_w^* were obtained by rearranging the terms of G_v and G_w given in [19], Eqs. (11), to match the grouping of Eqs. (2) and (3). G_ϕ was obtained by rearranging equation (11d) of [19] to match the grouping of Eq. (4).

The terms of the equations of motion were expanded in Taylor series, and nonlinearities up to third order were retained. All three displacement variables (v, w, and ϕ) were considered of the same order. It was assumed that the static equilibrium of the beam corresponded to the situation where the axes ξ and x were coincident and the axes η and ζ were parallel to y and z, respectively. Longitudinal extension of the wing was not permitted; that is, the equations were derived with an inextensionality constraint along the span.

C. Coupling of Aerodynamic and Structural Models

The coupling of aerodynamic and structural models was done using a tightly coupled solution [20] that allowed the two models to communicate during every time step. The aerodynamic nonlinear loads were generated by integrating the pressure and shear stresses over the wing surface. The pressure values were obtained from the unsteady flow solver. The aerodynamic loads were then passed to the structural solver and were used to calculate the wing deformation. It was assumed herein that the wing crosswise deformation was much smaller than the out-of-plane and torsional deformations. This assumption simplified the communication between the aerodynamic and structural models because it allowed specifying the location of the deforming wing by the position of the elastic axis and the leading edge along the span, as shown in Fig. 2. Consequently, only interpolation along the spanwise coordinate s was needed to transfer information between the structural and aerodynamic grids.

The coordinates of the new position of the wing were passed to the grid deformation algorithm, which updated the grid of the flow solver. The flow solution was then calculated and the aerodynamic loads were generated for a new wing position.

III. Numerical Method

This section presents the numerical implementation of the aeroelastic solver. The first part describes the methodology for the grid generation and deformation algorithm. The second part of the section describes the spatial and temporal discretization of the flow solver. The parallel algorithm implementation of the aeroelastic solver is presented next, followed by the description of the multigrid implementation.

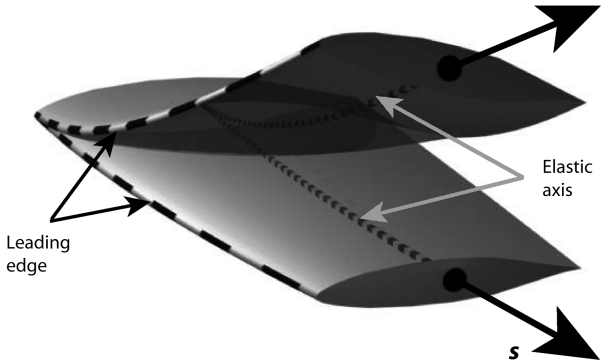

Fig. 2 Deforming wing location specified by loci of elastic axis and leading edge along the span.

A. Grid Generation and Deformation Algorithm

The computational domain was discretized using a hybrid grid, which consisted of hexahedra and triangular prisms. The computational domain was divided into layers that were topologically identical. The topologically identical nodes of adjacent layers were interconnected to generate the volume elements that were either triangular prisms or hexahedra. Each layer had a structured O-grid around the boundary of the structure and an unstructured grid at the exterior of the O-grid. The O-grid allowed good control over the mesh size near the structure and permitted clustering cells in the direction normal to the structure to properly capture boundary-layer effects. The unstructured grid was flexible in filling the rest of the domain. A detail of the grid seen along the x axis (Fig. 3) shows the structured and unstructured grids. The computational mesh deformed according to the structure displacement. Although in the Results section the structure was represented by different types of wings, the grid generation and deformation method was not limited to wings only.

Although the computational grid deformed as the wing bent and twisted under aerodynamic loads, the grid connectivity remained unchanged. To maintain a good quality grid as the wing deformed, the mesh deforming algorithm was required to generate a moving grid that was both perpendicular to the wing surface and to the outer boundary of the computational domain [21].

The algorithm for grid deformation was applied in two steps: 1) grid translation and 2) grid rotation about the three axes. The O-grid layers were translated without being deformed, since herein it was assumed that the wing cross section did not deform. This limitation can be easily removed once a more detailed structural model is implemented. The unstructured grid was deformed in the y–z plane using a spring-analogy technique [22]. The nodes of each layer were also translated in the x direction according to the wing deformation.

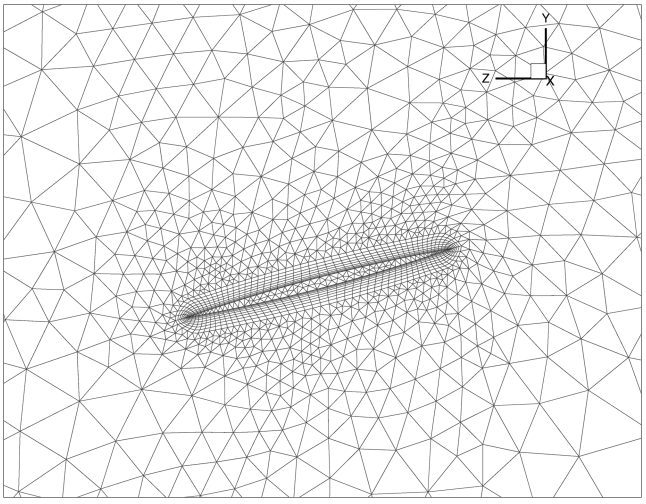

Fig. 3 Detail of a lateral view of the Goland wing grid.

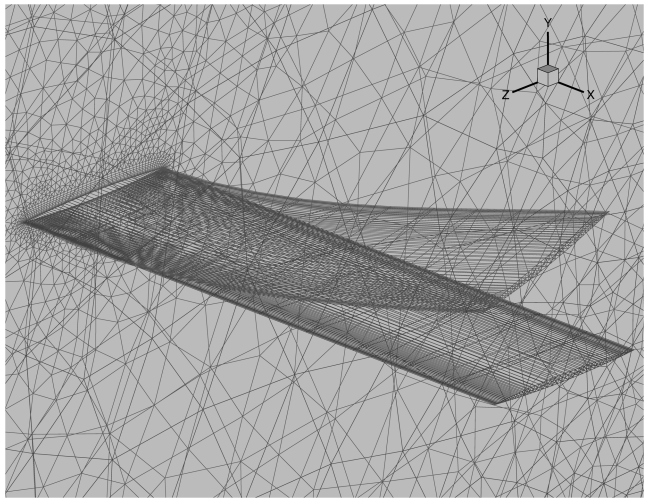

Fig. 4 Deformed and undeformed grids of the Goland wing.

The grid was rotated and stretched about the x, y, and z axes. The numerical values of the rotation angles θ_x, θ_y, and θ_z were calculated using data provided by the structural solver [21]. Figure 4 shows the undeformed and deformed grids for the Goland wing [23,24]. In this case, the deformation at the tip was approximately 20% of the span length, illustrating that the grid deformation algorithm is robust and can be applied to large deformations. It was shown in [21] that this algorithm can be applied to even larger wing-tip deformations, up to 60% of the wing semispan, without regridding. Consequently, this novel grid generation and deformation algorithm not only that allowed the reduction of the computational cost of grid generation by a factor of 3.6 or more [21], but it also maintained the grid topology unchanged. Therefore, the parallel algorithm and the multigrid algorithm described in Secs. III.A and III.D were simplified as if the structure did not deform.

B. Spatial and Temporal Discretization

The governing equations were solved using a finite volume method implemented for an unstructured grid [25]. The cell-averaged variables were stored at the nodes of the grid: that is, the vertices of the cells. The governing equations were discretized using dual meshes as control volumes. The median dual mesh was adopted herein because of its flexibility to handle unstructured mixed meshes [26].

An edge-based data structure was used for the discretization of the governing equations (1), due to its flexibility and efficiency in handling different element types. The numerical scheme, however, was node-based and, consequently, the solution was obtained at the nodes of the mesh. The surface integral of the inviscid and viscous fluxes was approximated as a sum over the faces of each control volume. The source terms were calculated using the control-volume-averaged solution and the derivatives of flow variables at the cell centroid.

Equation (1) was rewritten in a semidiscrete form as

$$\frac{\partial(Q_i \cdot V_i)}{\partial t} = \frac{\partial Q_i}{\partial t} V_i + \frac{\partial V_i}{\partial t} Q_i = -\oint_{S_i} \mathbf{F} \cdot \mathbf{n} \cdot \mathrm{d}S + E_i \cdot V_i \quad (8)$$

where Q_i denotes the volume-averaged flow variable Q over the volume V_i. The surface integral term on the right-hand side of this equation was approximated as

$$\oint_S \mathbf{F} \cdot \mathbf{n} \cdot \mathrm{d}S = \sum_{j=k(i)} (\mathbf{F}_{\mathrm{inv}} - \mathbf{F}_{\mathrm{vis}}) \cdot S_{ij} \quad (9)$$

where $k(i)$ is the set of vertices adjacent to node i, $\mathbf{F}_{\mathrm{inv}} - \mathbf{F}_{\mathrm{vis}}$ is the flux normal to the dual-mesh cell face, and S_{ij} is the cell face surface

area. Both the surface area and the flux were associated with the dual-mesh face that was associated with the edge (i, j).

Equation (8) and the turbulence model equations were written for a grid node i as $(\partial Q_i/\partial t)V_i = R_i$, where the residual R_i was

$$R_i = -\oint_{S_i} \mathbf{F} \cdot \mathbf{n} \cdot \mathrm{d}S + E_i \cdot V_i - \frac{\partial V_i}{\partial t} Q_i \tag{10}$$

The term $\partial V_i/\partial t$ is the time evolution of the cell volume and relates to the deformation of the grid. This term was calculated using the geometric conservation law [27], which relates the time derivative of the volume to the motion of the faces of the cell:

$$\frac{\partial V_i}{\partial t} = \oint_{S_i} \mathbf{V}_S \cdot \mathbf{n} \cdot \mathrm{d}S = \sum_{j=k(i)} \mathbf{V}_S \cdot \mathbf{n}_{ij} \cdot S_{ij} \tag{11}$$

where $\mathbf{V}_S$ denotes the velocity vector corresponding to the center of the face S_{ij}, for which the unitary normal vector is $\mathbf{n}_{ij}$. The integral was solved using the same edge-based numerical scheme used to solve the inviscid fluxes. The inviscid fluxes were computed using the upwind method with Roe's approximate Riemann solver and Harten's entropy fix. The gradients of the viscous fluxes at vertices were computed using the least-squares method with QR decomposition. The accuracy of the averaged gradients at edges was improved by using directional derivatives. Piecewise linear reconstruction was used to obtain second-order spatial accuracy. A first-order spatial accuracy option was also available, which could have been used to avoid divergence at simulation startup. None of the cases presented herein, however, used the first-order spatial accuracy option for startup.

A multistage explicit time-integration algorithm was used to calculate the unsteady solution. A residual smoothing algorithm was used to increase the integration time step [28]. Implicit residual smoothing added an implicit flavor to the explicit method by blending the residuals of the neighboring nodes. Residual smoothing allowed the use of larger time steps, which was especially needed for low-speed flows and sonic conditions.

C. Parallel Algorithm

To reduce the turnaround time, the flow solver was parallelized using the domain-decomposition approach. The message-passing interface standard libraries were used for the interprocessor communication. As described in the grid generation section, the computational domain was divided into topologically identical layers, which simplified the parallel communication. In addition, the grid deformation algorithm was designed so that the connectivity of the grid did not change during wing deformation.

Following the domain-decomposition approach, the grid was divided into several subgrids, and one processor was allocated for each subgrid. Each subgrid could include one or more layers, depending on the number of processors available. For the cases investigated in the Results section, which were all wings, the grid was divided along the wing span. Figure 5 shows the subgrids $i-1$ to $i+1$. The arrows indicate the communication paths. The nodes in the subgrids were divided in two categories: active and ghost nodes. The active nodes, marked as closed circles, were the nodes where the numerical computation took place. The ghost nodes, marked as open circles, were only used to compute the edge-based fluxes and the solution gradients. Each subgrid consisted of n layers of active nodes (layers 1 to n) and two layers of ghost nodes, layers 0 and $n+1$. The subscripts of the layers indicate the subgrid number. As shown in Fig. 5, the first two layers (0 and 1) of a subgrid i coincided with the last two layers (n and $n+1$) of the previous subgrid $i-1$.

The state vector at the ghost nodes was updated at each time step by the neighboring processor. The information traveled from active nodes to ghost nodes, as indicated by the arrows in Fig. 5. The active nodes of layers 1 and n were used in the communication process. As the structure deformed, each active node from the layers 1 and n communicated with only the same ghost node at each time step, as opposed to communicating with different ghost nodes. This was possible because the grid deformation algorithm did not change the mesh topology. The fact that each active node from layers 1 and n communicated with only one ghost node increased the parallelization efficiency, since it reduced the communication effort. The load per processor was balanced to achieve the maximum speedup. The load balance was achieved by evenly splitting the total number of layers of the grid among the available processors.

Table 1 summarizes the efficiency of the parallel computation. These results were obtained for the heavy Goland wing [23] at Mach = 0.09. The computational grid had 64 slices and each slice had 2316 nodes. Runs with up to 16 processors were completed on a 128-processor (1.3 GHz Intel Itanium) SGI Altix 3700 computer at the Texas A&M Supercomputing Center. Table 1 shows a small superlinear speedup for two and four processors. The efficiency reduced once the number of processors increased to eight and higher. In addition, runs with 16 and 32 processors were completed on a 2068-processor (2.4 GHz AMD Opteron) Cray XT3 at the Pittsburgh Supercomputing Center.

D. Multigrid Technique

The parallel algorithm reduced the turnaround time by dividing the computing effort over several processors. The number of operations slightly increased due to the processor communication, but the overall effect of parallel computation was a reduction of the wall-clock time. To further reduce the computational time, a multigrid technique was implemented in the flow solver algorithm [29]. The multigrid technique solved the governing equations on a series of successively coarser grids, accelerating the convergence of the solution on the finer grid while maintaining the overall accuracy of the flow solver. Therefore, multigrid methods achieved a reduction in the computational time by reducing the number of iterations necessary to obtain a converged solution.

The multigrid technique is based on the solution of the governing equations on a series of successively coarser grids that reduce the high-frequency components of the solution error and accelerate the convergence of the solution on the finer grid. The solution on the coarser grid is driven by the residuals of the finer grid, therefore

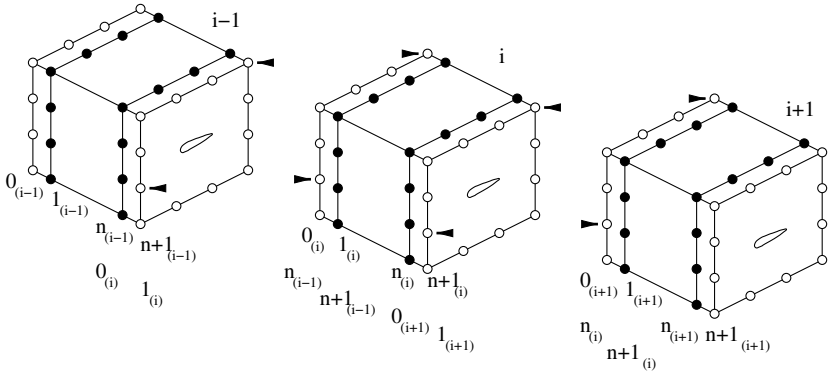

Fig. 5 Schematic view of the parallel implementation: active nodes (closed circles) and ghost nodes (open circles).

Table 1 Efficiency of parallel computation

Number of processors	Wall time, s	Efficiency, %	Computer
1	18,495	100.00	SGI Altix 3700
2	8,963	103.17	SGI Altix 3700
4	4,585	100.85	SGI Altix 3700
8	2,356	98.13	SGI Altix 3700
16	1,244	92.92	SGI Altix 3700
1	9,026	100.00	Cray XT3
16	623	90.55	Cray XT3
32	380	74.23	Cray XT3

maintaining the overall accuracy of the finer grid. The correction obtained for the coarse grid is then interpolated to the finer grid [30].

The three grids for the multigrid solver were generated using grid refinement. This technique starts from a coarse grid and generates finer grids by element subdivision. New nodes are inserted at the midpoint of the existing edges. All of the triangular and quadrilateral elements are subdivided into four elements for the following generation [31]. Figure 6 shows two levels of the multigrid refinement technique. The coarse grid is shown in black, and the finer grid is shown in gray. Note that each triangular and quadrilateral element of the grid is subdivided into four new elements.

If the boundaries of the computational domain are not planar, placing new nodes at the center of the parent edge will result in a faceted representation of the original geometry. To avoid this, the new nodes must be moved from the center of the parent edge to the true shape of the boundary of the domain [32]. Figure 7 shows the elements near the boundary. Note that the new nodes are not located at the midpoint of the parent edges, but instead lay on the true shape of the boundary. Consequently, a better representation of the geometry is obtained as the grid is refined.

The validation of the multigrid implementation was done using a rectangular wing with a NACA0012 airfoil. The Mach number was 0.3 and the angle of attack was 6.776° [33]. The wing span was 5 m and the chord was 1 m. The flow was assumed to be two-dimensional; therefore, the mesh had only two layers of nodes in the transversal dimension. The lateral boundary condition used was the no-penetration condition. The inlet and outlet boundaries were located 12 chords away from the airfoil, and the upper and lower boundaries were located 9 chords away from the airfoil.

The coarse mesh had 64 nodes around the airfoil, 16 layers of nodes for the O-grid around the airfoil, 16 nodes at the inlet and outlet boundaries, and 20 nodes at the upper and lower boundaries. The medium and fine meshes were obtained by grid refinement. Each quadrilateral and triangular cell was subdivided in four cells from the coarse to the medium mesh and again subdivided in four from the medium to the fine mesh. Therefore, the fine mesh had 256 nodes around the airfoil, 64 layers of nodes for the O-grid around the airfoil, 64 nodes at the inlet and outlet boundaries, and 80 nodes at the upper and lower boundaries.

Figure 8 shows the grid upstream of the airfoil leading edge. The grid includes the O-grid that surrounds the airfoil surface and the unstructured mesh that covers the rest of the domain. In this figure the three levels of multigrid meshes are present. The coarse mesh is plotted in thick black lines, the medium mesh with thin black lines, and the fine mesh with thin gray lines.

To verify the implementation of the multigrid turbulent flow solver, the results of a single-level fine grid were compared against a three-level multigrid with coarse, medium, and fine meshes. The residuals of the Navier–Stokes equations (10), the maximum Mach number, and the lift coefficient were compared for the two cases.

Figure 9 shows the variation of the maximum values of the residuals vs the iteration number for the single-level grid and multigrid solvers. Although the flow solver is three-dimensional, the flow was two-dimensional. For this reason, the errors of the ρw component were much smaller than the other components of the state vector. The maximum values of the residuals decreased faster for the multigrid solver compared with the single-level grid solver. The slope of the $\log(\rho u)$ residual between iterations 10,000 and 20,000 was -0.84×10^{-4} for the single-level grid solver and -1.56×10^{-4} for the multigrid solver. Consequently, on this iterations interval, the

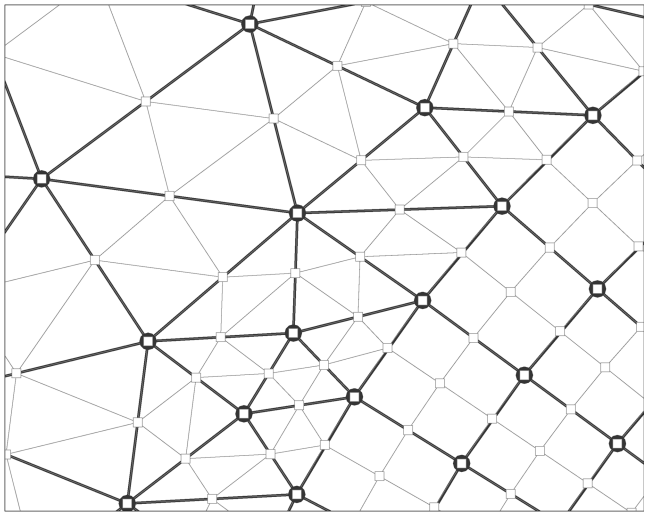

Fig. 6 Grid refinement for triangular and quadrilateral elements.

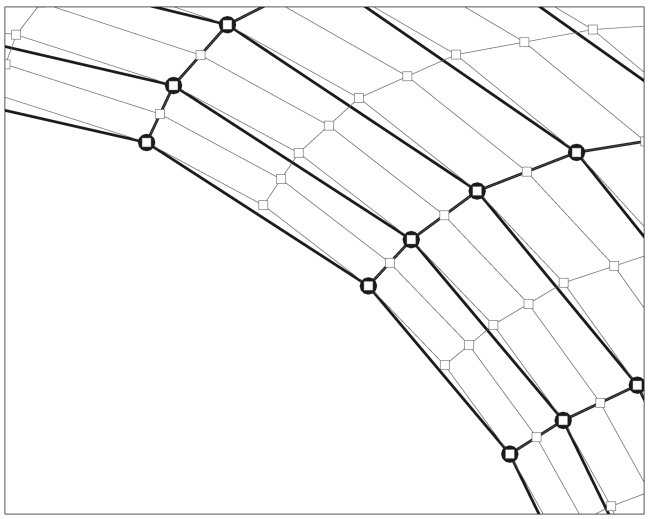

Fig. 7 Grid refinement for quadrilateral elements near boundary.

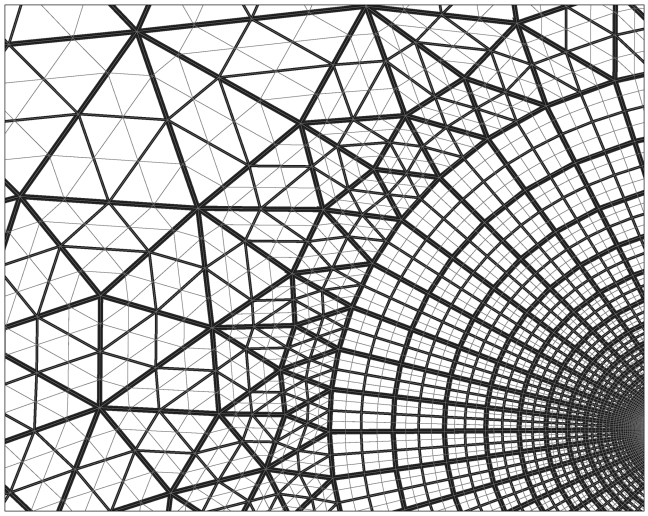

Fig. 8 Three levels of multigrid meshes: coarse (thick black), medium (thin black), and fine (thin gray).

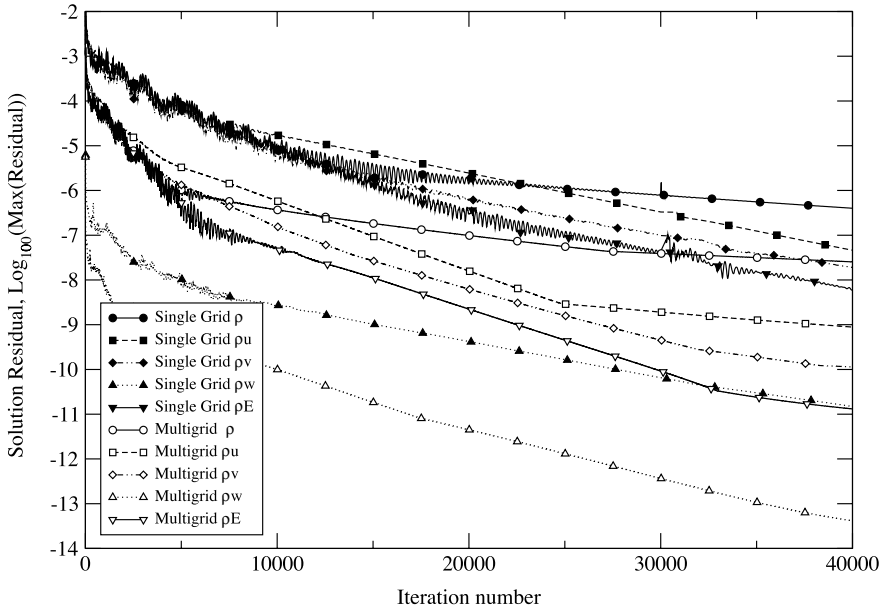

Fig. 9 Maximum residuals for single single-level grid and multigrid flow solvers.

multigrid solver reduced the residuals 1.9 times faster than the single-level grid solver. The second-order multigrid solver needed 8500 iterations to reduce the maximum error below 10^{-6}, and the single-level grid solver needed 27,650 iterations.

To evaluate the difference between the multigrid solution and the single-level grid solution, two variables have been compared on the airfoil surface: the pressure coefficient C_p and the wall shear stress τ_w. The average and maximum difference values were computed using the following formulas [34]:

$$\phi_{\max} = \max_{i\epsilon[1,N]} \frac{|\phi_i^{\text{single-level grid}} - \phi_i^{\text{multigrid}}|}{\phi_{\max} - \phi_{\min}}$$

$$\phi_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\phi_i^{\text{single-level grid}} - \phi_i^{\text{multigrid}}|}{\phi_{\max} - \phi_{\min}}$$

where ϕ is either C_p or τ_w, N is the number of nodes around the airfoil, and $\phi_{\max}$ and $\phi_{\min}$ are the maximum and minimum values of ϕ on the airfoil surface. Table 2 shows the maximum and average difference between the single-level grid solution and the multigrid solution for the pressure coefficient and the wall shear stress. These results indicate that there is a good match between the single-level grid and the multigrid solutions.

Figures 10 and 11 show the time history of the maximum Mach number and the lift coefficient. For both variables, Mach number and lift coefficient, the multigrid solution converged faster than the single-level grid solution. The first four digits of the lift coefficient, $C_L = 0.7257$, and of the maximum Mach number, $M = 0.5967$, were identical for the single-level grid and multigrid solutions. This indicated that the multigrid algorithm was correctly implemented and, consequently, no accuracy loss occurred while using the multigrid solver. It should be noted that in the multigrid method, the residual of the fine mesh governed the residuals of the medium and coarse meshes, and therefore the spatial accuracy of the fine grid was maintained.

Table 2 Average and maximum difference between the single-level grid solution and the multigrid solution

Variable	Average difference	Maximum difference
C_p	7.137E − 6	4.339E − 5
τ_w	1.106E − 4	9.928E − 4

IV. Results

This section starts with a numerical investigation of the forced pitching of the F-5 wing. The numerical results are validated by a comparison against experimental data. Another validation against experimental data, which used the nonlinear aeroelastic test apparatus wing, was presented in [35] and is not included herein. In a second example, the numerical simulation was used to assess the effect of the structural nonlinearities on the aeroelastic response of the heavy Goland wing.

A. F-5 Wing Undergoing Forced Pitching Motion

The experimental results available for the F-5 wing were used to assess the validity of the unsteady aerodynamics part of the aeroelastic solver. The F-5 wing was used for validation because detailed geometry description and steady/unsteady date were available. In addition, the F-5 wing is a challenging configuration because of the thin, drooped leading edge on a highly tapered wing. A 1:4.5 scaled version of the original F-5 wing was used for wind-tunnel experimental measurements [36]. The scaled F-5 wing had a semispan of 0.6476 m, a root chord of 0.6396 m, and a tip chord of 0.18 m. The taper ratio was 0.281. The cross section of the wing had a modified NACA65A004.8 airfoil [37]. Figure 12 shows the planform and the surface mesh.

The wing had a forced pitching motion about the elastic axis, which was located at 50% of the wing root chord. The pitching motion was periodic and had a frequency of 20 Hz and an amplitude of 0.11° [36]. The reduced frequency based on the semichord at midspan was 0.126. The flow conditions in the wind tunnel were Mach number = 0.6, static pressure $p = 100,000$ Pa, static temperature $T = 288.15$ K, and 0° angle of attack.

The computational domain had the inlet and outlet boundaries located 12 chords away from the wing. The upper and lower boundaries were located nine chords away from the wing. Two different grids were used for this numerical simulation, with 64 and 128 nodes around the wing. Wall boundary conditions were specified on the wing surface. No-penetration conditions were imposed at the upper boundary, lower boundary, and the lateral boundaries of the domain. The initial condition was constant freestream applied to all cells in the domain. Approximately eight cycles were necessary for the transient component to decay and the flow to become periodic.

The numerical results were compared against experimental data [36] and another set of numerical results reported by Mello and Sankar [38]. Figures 13 and 14 show the pressure coefficient C_p for

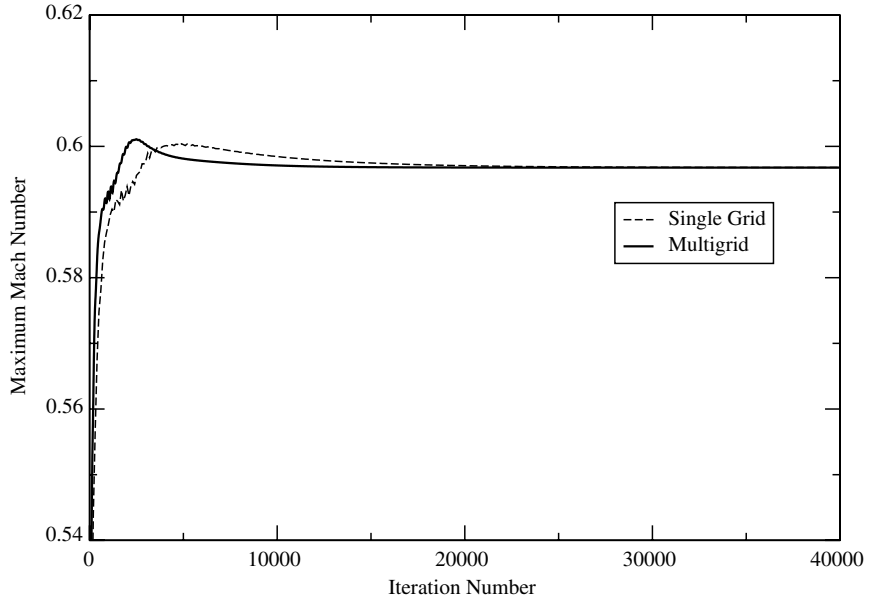

Fig. 10 Maximum Mach number for single-level grid and multigrid flow solvers.

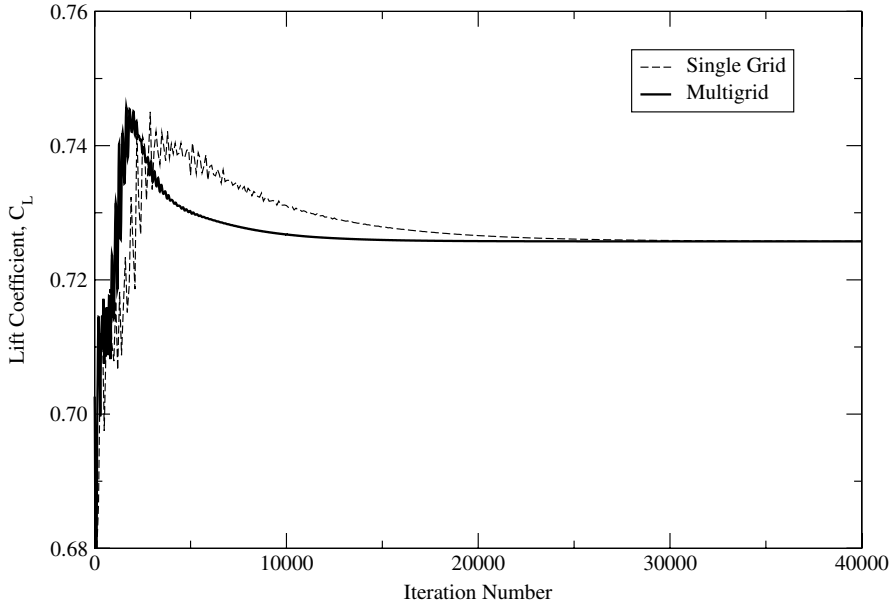

Fig. 11 Lift on NACA0012 airfoil for single-level grid and multigrid flow solvers.

the two different meshes compared against the experimental data [36] and the numerical results of Mello and Sankar [38]. These results were calculated at Secs. II and V, located at 35.2 and 72.1% of the wing span [36]. Both the 64- and 128-node solutions agree well with the experimental results for the real components of the C_p. The numerical solution appears to be grid-independent, since the 64- and 128-node solutions almost overlap.

For the imaginary components, the agreement with the experimental data is fair. Note that the range of the ordinate values of the imaginary component is smaller than that of the real component. For this reason, the differences between different sets of results appear amplified for the imaginary components compared with the real components. Note also that, as expected, the 128-node solution appears closer to the experimental data than the 64-node solution.

For completeness, we included in Figs. 13 and 14 the numerical results reported by Mello and Sankar [38]. These numerical results were obtained using a hybrid Navier–Stokes/potential method solver. The Navier–Stokes solver was used for the region adjacent to the wing, where the viscous effects are important. Figures 13 and 14 show that, at least for the real component of the pressure coefficient, the flow solver described herein produces results that are closer to the experimental than the hybrid Navier–Stokes/Potential Method solver. One reason for this can be the fact that Mello and Sankar used the Baldwin–Lomax algebraic turbulence model and the flow solver described herein used a better turbulence model, the shear stress transport model [18].

B. Heavy Goland Wing

The heavy Goland wing [23] is similar to the original Goland wing [39] but is considerably heavier to ensure applicability in the transonic region [40,41]. The parameters of the heavy Goland wing (which include airfoil shape, mass properties, and wing stiffness) are presented in [23] and will not be repeated in here. The heavy Goland

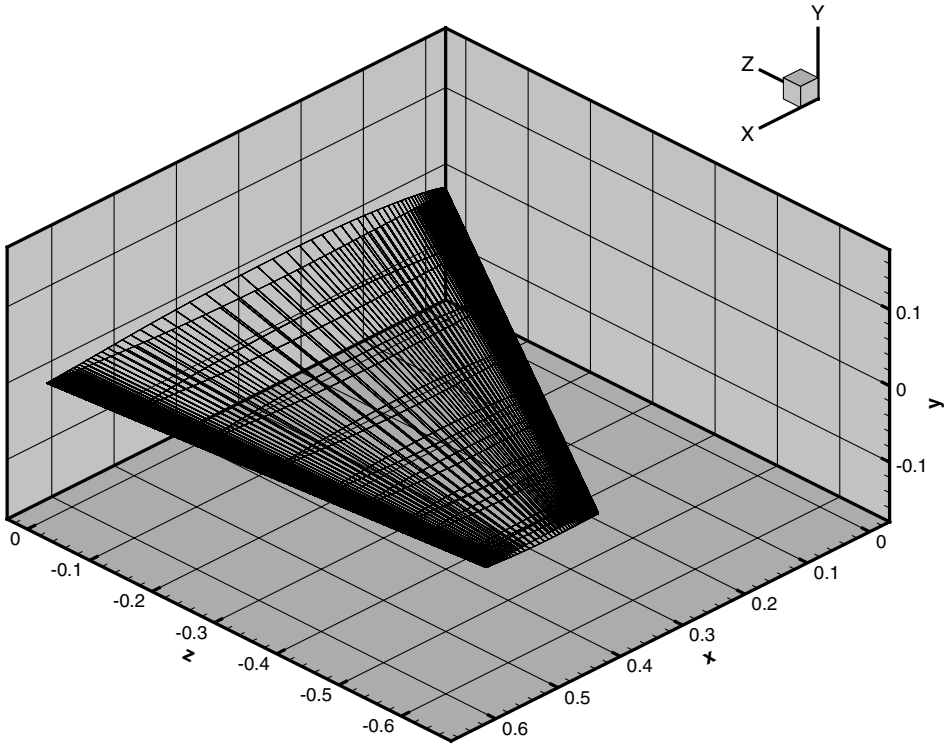

Fig. 12 Isometric projection of the F-5 wing and wing surface mesh.

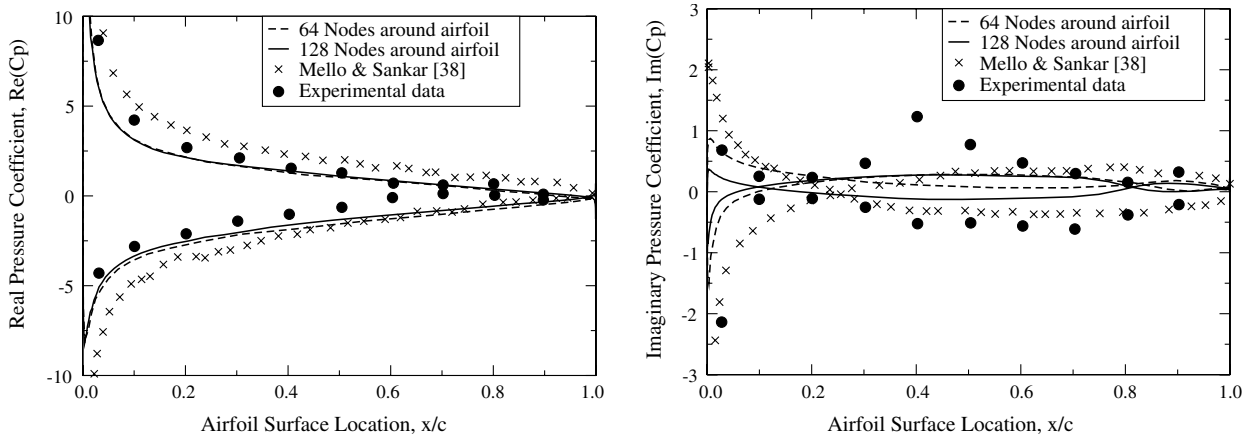

Fig. 13 Real and imaginary components of pressure coefficients on the F-5 wing at station 2 (35.2% span) for Mach = 0.6 and frequency = 20 Hz.

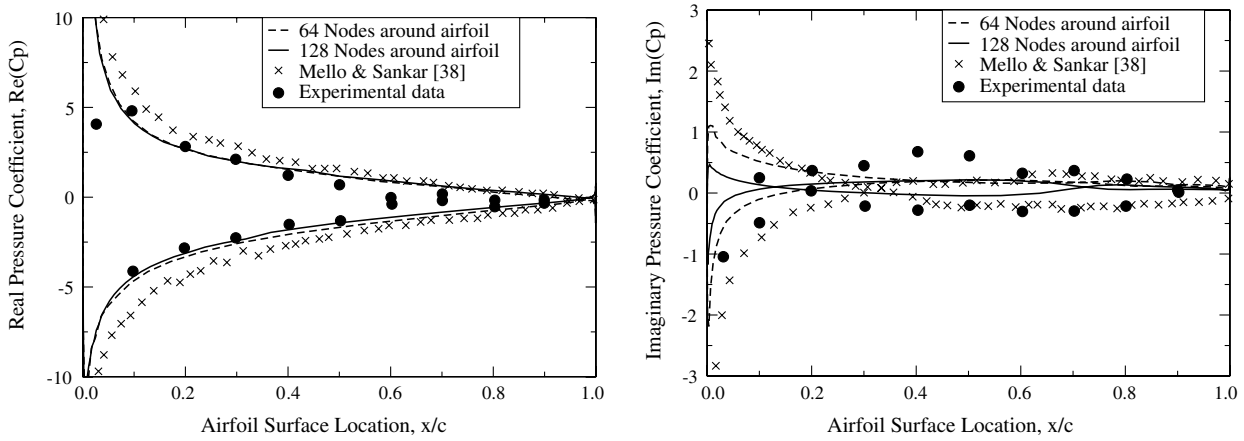

Fig. 14 Real and imaginary components of pressure coefficients on the F-5 wing at station 5 (72.1% span) for Mach = 0.6 and frequency = 20 Hz.

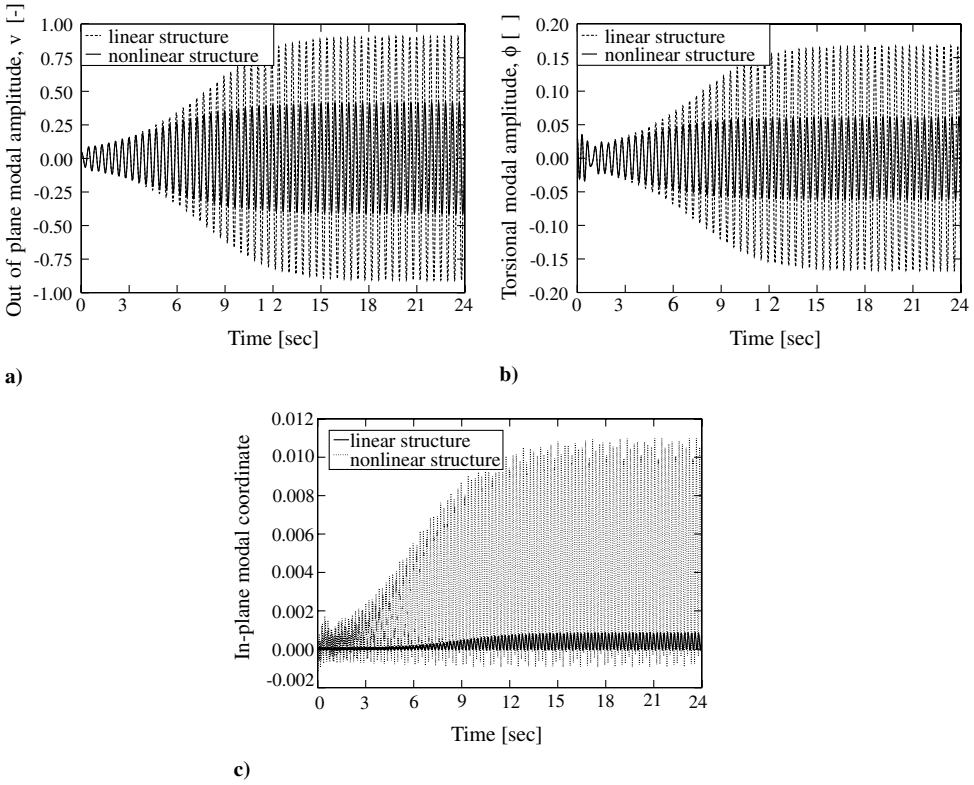

Fig. 15 Modal amplitudes of linear and nonlinear structural model: a) out-of-plane bending, b) torsional, and c) in-plane bending.

wing was used herein to compare the results obtained using linear and nonlinear structural models. These results underscore the importance of using a nonlinear model.

An analysis of the heavy Goland wing using a nonlinear structural model and a simple aerodynamics approach was presented in [12]. Herein, the same nonlinear structural model was coupled with the nonlinear aerodynamic model described in Sec. II.A. The structural model included the mechanics of both the traditional out-of-plane and torsional components plus an additional in-plane mode of motion. This additional mode and the presence of nonlinear effects led to remarkable nonlinear dynamic interactions.

The O-grid around the Goland wing had 97 grid points along the surface and 25 grid points normal to the surface, in each layer. There were 11 layers of nodes along the wing span and five layers past the wing tip. The total number of nodes in the O-grid and the unstructured grid around it was 60,376. The total number of cells was 76,440, out of which 34,560 cells were hexahedra in the O-grid and 41,880 were triangular prisms in the unstructured grid. The code was run on a MacPro computer with two 2.66 GHz dual-core Intel Xeon processors and 3 GB memory. The CPU time was 3.7 s/iteration.

The aeroelastic analysis of the heavy Goland wing was performed for a Mach number of 0.7 and an angle of attach of 0°. Figure 15 shows the out-of-plane bending, torsional, and in-plane bending modal amplitudes. Modal amplitudes were smaller when the nonlinear terms were taken into account. This could indicate that using a linear structural model is a conservative approach; that is, the true (nonlinear) deformation is smaller than the deformation predicted by the linear model. This conclusion was invalidated, however, by results that modeled in-plane deformation, shown in Fig. 15c.

While using a linear structural model, there was no coupling between the modes. The in-plane deformation was positive and oscillated about a nonzero equilibrium. When nonlinear terms were taken into account, the deformation was approximately 1 order of magnitude higher than the deformation of the linear structural model. The amplification of the in-plane deformation was caused by the $(D_\eta - D_\zeta)(\phi' v'' + \phi v''')$ term of the structural nonlinear component G_w^* (6). This term represents the coupling between pitching and out-of-plane bending.

V. Conclusions

This paper presented a multigrid parallel algorithm for a nonlinear aeroelastic analysis. The aeroelastic model consisted of 1) an unsteady viscous aerodynamic model that captured compressible flow effects for transonic flows with shock/boundary-layer interaction; 2) a nonlinear structural model that captured in-plane, out-of-plane, and torsional couplings; and 3) a solution methodology that assured a tightly coupled solution of the nonlinear structure and fluid flow, including a consistent geometric interface between the highly deforming structure and flowfield.

The aeroelastic solver fully integrated a nonlinear structural model and an unsteady, compressible, viscous flow model. The flow was modeled using the Reynolds-averaged Navier–Stokes equations with a two-equation eddy-viscosity turbulence model. A domain-decomposition parallel algorithm based on message-passing interface was developed for the flow solver. A novel grid generation and deformation algorithm was developed for the flow solver. This grid generation and deformation algorithm allowed large deformation, such as wing-tip deformations up to 60% of wing semispan, without regridding. Consequently, not only was the computational cost of generating the grid as the structure deformed reduced by a factor of 3.6 or more, but the topology of the grid also did not change. As a result, the parallel algorithm and the multigrid algorithm were simplified as if the structure did not deform. A three-level multigrid algorithm was implemented in the flow solver to further reduce the computational time. For the case presented herein, the multigrid algorithm reduced the computational time by approximately a factor of 2.

The validation of the numerical solver was done using experimental results of the F-5 wing. A good agreement was obtained between the numerical simulation and the experimental data for the F-5 wing. The aeroelastic solver was then used to assess the effect of structural nonlinearities on the aeroelastic response of the heavy Goland wing. The numerical results for the heavy Goland wing demonstrated the importance of using nonlinear structural models. When the nonlinear terms were included, the in-plane wing deformation was approximately 1 order of magnitude larger than the in-plane deformation that omitted nonlinear terms.

Acknowledgments

This work was sponsored by the U.S. Air Force Office of Scientific Research under grant no. FA9550-04-1-0174. The authors are thankful to the Pittsburgh Supercomputing Center and the Texas A&M University Supercomputing Facility for making the computing resources available.

References

[1] Norton, W. J., "Limit Cycle Oscillation and Flight Flutter Testing," *Proceedings of the 21st Annual Symposium*, Society of Flight Test Engineers, Lancaster, CA, 1990, pp. 3.4.4–3.4.12.

[2] Denegri, C. M. J., "Limit Cycle Oscillation Flight Test Results of a Fighter with External Stores," *Journal of Aircraft*, Vol. 37, No. 5, 2000, pp. 761–769.
doi:10.2514/2.2696

[3] Bunton, R. W., and Denegri, C. M. J., "Limit Cycle Oscillation Characteristics of Fighter Aircraft," *Journal of Aircraft*, Vol. 37, No. 5, 2000, pp. 916–918.

[4] Edwards, J. W., "Calculated Viscous and Scale Effects on Transonic Aeroelasticity," AGARD Rept. R-822, Neuilly-sur-Seine, France, March 1998.

[5] Jacobson, S. B., Britt, R. T., Dreim, D. R., and Kelly, P. D., "Residual Pitch Oscillation (RPO) Flight Test and Analysis on the B-2 Bomber," *39th AIAA Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 98-1805, Long Beach, CA, April 1998.

[6] Schuster, D. M., Vadyak, J., and Atta, E. H., "Fligth Loads Prediction Methods for Aircraft Euler/Navier–Stokes Aeroelastic Method (ENS3DAE), Version 1," Wright Research & Development Center, TR WRDC-TR-89-3104, Wright-Patterson AFB, OH, 1989.

[7] Lee-Rausch, E. M., and Batina, J. T., "Wing Flutter Computations Using an Aerodynamic Model Based on the Navier–Stokes Equations," *Journal of Aircraft*, Vol. 33, No. 6, Nov.–Dec. 1996, pp. 1139–1147.
doi:10.2514/3.47068

[8] Krist, S. L., Biedron, R. T., and Rumsey, C. L., "CFL3D User's Manual (Version 5.0)," NASA TM-1998-208444, 1998.

[9] Melville, R., "Nonlinear Simulation of F-16 Aeroelastic Instability," 39th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2001-570, Reno, NV, Jan. 2001.

[10] Geuzaine, P., Brown, G., Harris, C., and Farhat, C., "Aeroelastic Dynamic Analysis of a Full F-16 Configuration for Various Flight Conditions," *AIAA Journal*, Vol. 41, No. 3, March 2003, pp. 363–371.
doi:10.2514/2.1975

[11] Thomas, J. P., Dowell, E. H., Hall, K. C., and Denegri, C. M., "Further Investigation of Modeling Limit Cycle Oscillation Behaviour of the F-16 Fighter Using a Harmonic Balance Approach," 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA Paper 2005-1917, Austin, TX, April 2005.

[12] Strganac, T. W., Cizmas, P. G., Nichkawde, C., Gargoloff, J., and Beran, P. S., "Aeroelastic Analysis for Future Air Vehicle Concepts Using a Fully Nonlinear Methodology," AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA Paper 2005-2171, Austin, TX, 2005, pp. 4575–4608.

[13] Seber, G., and Bendiksen, O. O., "Nonlinear Flutter Calculations Using Finite Elements in a Large Deformation Direct Eulerian-Lagrangian Formulation," 46thAIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA Paper 2005-1856, Austin, TX, April 2005.

[14] Gordnier, R. E., and Visbal, M. R., "High-Fidelity Computational Simulation of Nonlinear Fluid-Structure Interactions," *The Aeronautical Journal*, Vol. 109, No. 1097, July 2005, pp. 301–312.

[15] Farhat, C., and Lesoinne, M., "Two Efficient Staggered Algorithms for the Serial and Parallel Solution of Three-Dimensional Nonlinear Transient Aeroelastic Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 182, Nos. 3–4, 2000, pp. 499–515.
doi:10.1016/S0045-7825(99)00206-6

[16] Gargoloff, J. I., Cizmas, P. G. A., Strganac, T. W., and Beran, P. S., "Parallel Algorithm for Fully Nonlinear Aeroelastic Analysis," 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA Paper 2006-2073, Newport, RI, May 2006.

[17] Hoke, C., and Schwabacher, G., "Aerodynamic Analysis of Complex Missile Configurations Using AVUS (Air Vehicles Unstructured Solver)," Applied Aerodynamics Conference, AIAA Paper 2004-5452, Providence, RI, Aug. 2004.

[18] Menter, F. R., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.
doi:10.2514/3.12149

[19] da Silva, M. R. M. C., and Glynn, C. C., "Nonlinear Flexural-Flexural-Torsional Dynamics of Inextensional Beams—1: Equations of Motion," *Journal of Structural Mechanics*, Vol. 6, No. 4, Sept. 1978, pp. 437–448.
doi:10.1080/03601217808907348

[20] Smith, M. J., Schuster, D. M., Huttsell, L., and Buxton, B., "Development of an Euler/Navier–Stokes Aeroelastic Method for Three-dimensional Vehicles with Multiple Flexible Surfaces," AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA Paper 1996-1513, Salt Lake City, UT, April 1996, pp. 1733–1746.

[21] Gargoloff, J. I., and Cizmas, P. G. A., "Mesh Generation and Deformation Algorithm for Aeroelastic Simulations," 45th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2007-556, Reno, NV, Jan. 2007.

[22] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, Aug. 1990, pp. 1381–1388.
doi:10.2514/3.25229

[23] Eastep, F. E., and Olsen, J. J., "Transonic Flutter Analysis of a Rectangular Wing with Conventional Airfoil Sections," *AIAA Journal*, Vol. 18, No. 10, 1980, pp. 1159–1164.
doi:10.2514/3.50866

[24] Snyder, R. D., Scott, J. N., Khot, N. S., Beran, P. S., and Zweber, J. V., "Predictions of Store-Induced Limit-Cycle Oscillations Using Euler and Navier–Stokes Fluid Dynamics," 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, AIAA Paper 2003-1727, Norfolk, VA, April 2003.

[25] Han, Z., and Cizmas, P. G. A., "Prediction of Axial Thrust Load in Centrifugal Compressors," *International Journal of Turbo & Jet-Engines*, Vol. 20, No. 1, Jan. 2003, pp. 1–16.

[26] Barth, T. J., "A 3-D Upwind Euler Solver for Unstructured Meshes," 10th AIAA Computational Fluid Dynamics Conference, AIAA Paper 91-1548-CP, Honolulu, HI, June 1991.

[27] Thomas, P. D., and Lombard, C. K., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, Oct. 1979, pp. 1030–1037.
doi:10.2514/3.61273

[28] Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft," AIAA Paper 86-0103, 1986.

[29] Alonso, J., Martinelli, L., and Jameson, A., "Multigrid Unsteady Navier–Stokes Calculations with Aeroelastic Applications," 33rd Aerospace Sciences Meeting and Exhibit, AIAA Paper 1995-0048, Jan. 1995.

[30] Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation*, Vol. 13, Nos. 3–4, 1983, pp. 327–355.
doi:10.1016/0096-3003(83)90019-X

[31] Holmes, D. G., and Connell, S. D., "Solution of the 2D Navier–Stokes Equations on Unstructured Adaptive Grids," 9th AIAA Computational Fluid Dynamics Conference, AIAA Paper 89-1932, Buffalo, NY, June 1989.

[32] Connell, S. D., and Holmes, D. G., "Three-Dimensional Unstructured Adaptive Multigrid Scheme for the Euler Equations," *AIAA Journal*, Vol. 32, No. 8, Aug. 1994, pp. 1626–1632.
doi:10.2514/3.12152

[33] Harris, C. D., "Two-Dimensional Aerodynamic Characteristics of the NACA0012 Airfoil at the Langley 8-Foot Transonic Pressure Tunnel," NASA, TM 81927, 1981.

[34] Cizmas, P. G. A., and Palacios, A., "Proper Orthogonal Decomposition of Turbine Rotor-Stator Interaction," *Journal of Propulsion and Power*, Vol. 19, No. 2, March–April 2003, pp. 268–281.
doi:10.2514/2.6108

[35] Gargoloff, J. I., Cizmas, P. G. A., Strganac, T. W., and Beran, P. S., "A Parallel Multigrid Algorithm for Aeroelasticity Simulations," 45th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2007-330, Reno, NV, Jan. 2007.

[36] Tijdeman, H., van Nunen, J. W. G., Kraan, A. N., Persoon, A. J., Poestkoke, R., Roos, R., Schippers, P., and Siebert, C. M., "Transonic Wind Tunnel Tests on an Oscillating Wing with External Stores. Part 2: The Clean Wing," TR AFFDL-TR-78-194, National Aerospace Lab., Amsterdam, March 1979.

[37] Persoon, A. J., Roos, R., and Schippers, P., "Transonic and Low Supersonic Wind-tunnel Tests on a Wing with Inboard Control Surface," National Aerospace Lab., TR AFWAL-TR-80-3146, Amsterdam, Dec. 1980.

[38] Mello, O. A. F., and Sankar, L. N., "An Improved Hybrid Navier–Stokes/Potential Method for Computation of Unsteady Compressible Flows," 26th AIAA Fluid Dynamics Conference, AIAA Paper 95-2160, San Diego, CA, 1995.

[39] Goland, M., "The Flutter of a Uniform Cantilever Wing," *Journal of Applied Mechanics*, Vol. 12, No. 4, Dec. 1945, pp. 197–208.

[40] Parker, G. H., Maple, R. C., and Beran, P. S., "Computational Aeroelastic Analysis of Store-Induced Limit-Cycle Oscillation," *Journal of Aircraft*, Vol. 44, No. 1, Jan.–Feb. 2007, pp. 48–59. doi:10.2514/1.21051

[41] Woodgate, M. A., and Badcock, K. J., "Fast Prediction of Transonic Aeroelastic Stability and Limit Cycles," *AIAA Journal*, Vol. 45, No. 6, June 2007, pp. 1370–1381. doi:10.2514/1.25604