Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Fluids xxx (2005) xxx–xxx
computers & fluids
www.elsevier.com/locate/compfluid

Flutter prediction by an Euler method on non-moving Cartesian grids with gridless boundary conditions

D.J. Kirshman, F. Liu *

Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA 92697-3975, United States

Received 22 September 2004; received in revised form 28 February 2005; accepted 12 April 2005

Abstract

A method is presented for the prediction of transonic flutter by the Euler equations on a stationary Cartesian mesh. Local grid refinement is established through a series of embedded meshes, and a gridless method is implemented for the treatment of surface boundary conditions. For steady flows, the gridless method applies surface boundary conditions using a weighted average of the flow properties within a cloud of nodes in the vicinity of the surface. The weighting is established with shape functions derived using a least-squares fitting of the surrounding nodal cloud. For unsteady calculations, a perturbation of the shape functions is incorporated to account for a fluctuating surface normal direction. The nature of the method provides for efficient and accurate solution of transient flow problems in which surface deflections are small (i.e. flutter calculations) without the need for a deforming mesh. Although small deviations in angle of attack are considered, the mean angle of attack can be large. Results indicate good agreement with available experimental data for unsteady flow, and with computational results addressing flutter of the Isogai wing model obtained using traditional moving mesh algorithms.

1. Introduction

In this work, the time-accurate solution of a moving body in transonic flow is considered. In particular, aeroelastic simulations associated with oscillating airfoils are addressed. Computations addressing such phenomena have been solved using various techniques and levels of complexity. Methods based on the transonic small-disturbance equations are popular due to their economical implementation [1,2]. However, such methods are limited to irrotational flows absent of strong shocks and nonlinear effects. Moreover, a dominant factor in the study of transonic flutter is shock oscillations, which must be accounted for to adequately predict stability [3]. More importantly, the phase lag of the shock motion with respect to the oscillation of the airfoil must be accurately predicted. In order to treat such effects, methods incorporating the Euler or Navier–Stokes equations are required, and have been successfully implemented [4–18]. In such analyses, the typical approach is to consider a movable or deformable mesh in which regeneration or displacement of grid cells is required at every time step. Such a procedure can be costly and unreliable, especially for Navier–Stokes simulations. Development of an efficient and robust moving grid methodology remains a challenge in the field of computational aeroelasticity [17].

With respect to reducing the labor associated with grid generation, the use of Cartesian grids for fluid dynamic simulations is popular with many researchers [19–27]. Cartesian grid solutions have also been considered in moving mesh simulations [28,29]. The advantages are numerous, including simple and efficient mesh generation, superior implementation of high order discretization schemes, minimal phase error associated with shock-capturing calculations, and an absence of issues

* Corresponding author. Tel.: +1 949 824 1750.
E-mail address: fliu@uci.edu (F. Liu).

0045-7930/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2005.04.004

Nomenclature

c	acoustic speed
C_h	damping coefficient for plunging motion
C_α	damping coefficient for pitching motion
E	total energy
$\mathbf{f}$	convective flux vector in x-direction
f	component of flux vector in x-direction
f^h	gridless approximation of flux variable
$\mathbf{g}$	convective flux vector in y-direction
g	component of flux vector in y-direction
h	wing displacement in plunging motion
H	total enthalpy
i, j	nodal indices
l	pseudo-time level index
L	sectional aerodynamic lift
m	mass per unit span
M_{ea}	sectional aerodynamic moment about elastic axis
n	time level index
p	static pressure
$\mathbf{Q}_{\mathrm{F}}$	multi-grid forcing function
$\mathbf{q}$	wing model displacement vector
r_α	radius of gyration per half chord
R	surface radius of curvature
$\mathbf{R}$	residual vector
$\overline{R}$	ideal gas constant
t	time
t^*	pseudo-time
T	temperature
$\mathbf{u}$	vector of conserved variables
u, v	Cartesian velocity components
V_{F}	flutter speed index
V_{n}	surface normal velocity
V_∞	free-stream velocity
x, y	Cartesian space coordinates
α	wing displacement in pitching motion
α_0	magnitude of prescribed pitching oscillation
α_k	Runge–Kutta stage coefficients for kth stage
α_{m}	mean angle of attack
Δx	local grid spacing
ϕ_n	gridless shape function at node n
γ	ratio of specific heats
κ_{c}	reduced frequency
ρ	density
σ	Courant number
τ	non-dimensional structural time
ω_h	uncoupled natural frequency of plunging motion
ω_a	uncoupled natural frequency of pitching motion

associated with mesh skewness and distortion. The obvious drawback of the Cartesian approach is the difficulty in the implementation of solid wall boundary conditions. Such issues include the requirement for excessive mesh refinement near curved boundaries, and problematic implementation in cases where the geometry under consideration is "thin" compared to the local mesh spacing.

In this work, the gridless boundary condition approach developed by Kirshman and Liu [27] for steady flow calculations is extended to the computation of unsteady flows on non-moving Cartesian grids using a perturbation of the gridless boundary condition. Gao et al. [30–32] also developed a small-perturbation boundary condition method for the Euler equations on non-moving Cartesian grids. In their work, the solid wall boundary conditions for an airfoil are approximated by a first-order expansion on the airfoil mean line in a similar fashion to the classical small-disturbance potential flow method. Gao et al. show that their method provides good accuracy for both steady and unsteady problems involving even relatively thick airfoils and moderately large angles of attack. However, the method is formally limited to thin airfoils and suffers from difficulties associated with a singularity at the blunt leading edge. The gridless boundary condition method developed in Ref. [27] for steady flows, on the other hand, treats the wall boundary condition of an arbitrary shape accurately without the assumption of thin geometry, and there are no problems associated with the leading-edge singularity. For cases in which the unsteady motion or deformation of the flow boundary is small with respect to the mean position, as is the case for flutter simulations, the same small-perturbation approximation may be made on the unsteady part of the gridless boundary condition, providing a very general and accurate method for computing unsteady flows over arbitrary shapes on non-moving Cartesian grids. As will be discussed below, since the gridless boundary condition method makes use of a set of shape functions over a cloud of gridless nodal points, the unsteady perturbation of the boundary conditions can be easily incorporated into a perturbation of the shape functions without changing either the location or the configuration of the original cloud of gridless nodal points.

In the following sections, the basic numerical method based on flux-vector splitting and the steady gridless boundary conditions are first reviewed. The extension of the gridless boundary condition method to the treatment of moving boundaries is then discussed in detail. Before considering the prediction of transonic flutter, two preliminary studies are considered to establish the

accuracy and applicability of the perturbed boundary condition approach. First, the effectiveness of incorporating angle of attack variations using perturbations in the surface normal is established by comparison to steady, body-fitted solutions using Jameson's FLO52 code. It is demonstrated that the effects a varying angle of attack can be accurately incorporated into the perturbation boundary condition without moving the grid or changing the far field boundary condition. Application of the method to a transient simulation of an oscillating airfoil is then presented in which results are shown to compare well with the experimental data of Landon [33]. Finally, the ability of the method to predict wing flutter is then established by comparison with moving mesh simulations of Alonso and Jameson [34] and Liu et al. [11].

2. Governing equations

In this work, the steady two-dimensional flow of an inviscid, compressible gas is considered. Such flows are governed by the Euler equations, which provide for the conservation of mass, momentum, and energy, and are given by

$$\frac{\partial \mathbf{u}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{g}}{\partial y}=0 \tag{1}$$

where

$$\mathbf{u}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix},\quad \mathbf{f}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uH\end{bmatrix},\quad \mathbf{g}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vH\end{bmatrix} \tag{2}$$

For an ideal gas, the total energy and total enthalpy can be written, respectively, as

$$E=\frac{p}{\rho(\gamma-1)}+\frac{1}{2}(u^2+v^2) \tag{3}$$

$$H=E+\frac{p}{\rho} \tag{4}$$

Additionally, the equation of state is given by

$$p=\rho\overline{R}T \tag{5}$$

3. Flow field discretization

3.1. Field node discretization

The entire flow domain is discretized using a purely Cartesian mesh, which is generated independently of the body. In order to solve for the flow field, a finite difference scheme using Van Leer flux-vector splitting is performed [35]. In this scheme, the convective flux vectors **f** and **g** are decomposed into "upwind" and "downwind" components such that the governing equations (1) are re-written as

$$\frac{\partial \mathbf{u}}{\partial t}+\frac{\partial \mathbf{f}^+}{\partial x}+\frac{\partial \mathbf{f}^-}{\partial x}+\frac{\partial \mathbf{g}^+}{\partial y}+\frac{\partial \mathbf{g}^-}{\partial y}=0 \tag{6}$$

where

$$\begin{aligned}\mathbf{f}&=\mathbf{f}^+ +\mathbf{f}^-\\ \mathbf{g}&=\mathbf{g}^+ +\mathbf{g}^-\end{aligned} \tag{7}$$

where the positive and negative superscripts indicate the portion of the total flux that travel in the positive and negative coordinate directions, respectively. For supersonic flow, there is no splitting and the entire flux quantity travels in the downstream direction. For subsonic flow (in a given coordinate direction), the flux vectors are given by

$$\mathbf{f}^{\pm}=\pm\frac{\rho}{4c(u\pm c)^2}\begin{bmatrix}1\\ \frac{1}{\gamma}[u(\gamma-1)\pm 2c]\\ v\\ \frac{[(\gamma-1)u\pm 2c]^2}{(\gamma^2-1)}+\frac{u^2+v^2}{2}\end{bmatrix} \tag{8}$$

$$\mathbf{g}^{\pm}=\pm\frac{\rho}{4c(v\pm c)^2}\begin{bmatrix}1\\ u\\ \frac{1}{\gamma}[v(\gamma-1)\pm 2c]\\ \frac{[(\gamma-1)v\pm 2c]^2}{(\gamma^2-1)}+\frac{u^2+v^2}{2}\end{bmatrix} \tag{9}$$

Considering the grid location defined by the nodal indices i and j in the x- and y-directions, the semi-discrete form of Eq. (6) can be written in terms of a first-order difference and a limited second-order correction as

$$\begin{aligned}\frac{\mathrm{d}\mathbf{u}_{i,j}}{\mathrm{d}t}=&-\frac{1}{2}\left[\left(2+\Psi\left(r^+_{i+\frac{1}{2}}\right)\right)\delta^+_x\mathbf{f}^+_{i,j}-\Psi\left(r^+_{i-\frac{3}{2}}\right)\delta^+_x\mathbf{f}^+_{i-1,j}\right]\\ &-\frac{1}{2}\left[\left(2+\Psi\left(r^-_{i-\frac{1}{2}}\right)\right)\delta^-_x\mathbf{f}^-_{i,j}-\Psi\left(r^-_{i+\frac{3}{2}}\right)\delta^-_x\mathbf{f}^-_{i+1,j}\right]\\ &-\frac{1}{2}\left[\left(2+\Psi\left(r^+_{j+\frac{1}{2}}\right)\right)\delta^+_y\mathbf{g}^+_{i,j}-\Psi\left(r^+_{j-\frac{3}{2}}\right)\delta^+_y\mathbf{g}^+_{i,j-1}\right]\\ &-\frac{1}{2}\left[\left(2+\Psi\left(r^-_{j-\frac{1}{2}}\right)\right)\delta^-_y\mathbf{g}^-_{i,j}-\Psi\left(r^-_{j+\frac{3}{2}}\right)\delta^-_y\mathbf{g}^-_{i,j+1}\right]\end{aligned} \tag{10}$$

where the δ are first-order upwind fluxes

$$\delta^+_x f^+_{i,j}=\frac{f^+_{i,j}-f^+_{i-1,j}}{\Delta x},\quad \delta^-_x f^-_{i,j}=\frac{f^-_{i+1,j}-f^-_{i,j}}{\Delta x} \tag{11}$$

and similarly for the y-direction. The flux limiter, $\Psi(r)$ in the above is implemented to suppress oscillations of the second-order solution in high gradient regions (i.e. shocks). In the results presented here, the standard minmod limiter is incorporated.

3.2. Gridless node discretization

A gridless treatment is used to apply the surface boundary conditions along the body geometry, and to evolve the solution at Cartesian mesh nodes in close proximity to the surface (i.e. those that do not have a complete computational stencil to apply Eq. (10)). Implementation of the gridless method requires the definition of nodal shape functions whose construction is achieved by means of a least-squares fit of surrounding Cartesian mesh and surface nodes. Their derivation is presented in detail in Ref. [27], and is not repeated here.

Once shape functions are derived, an approximated conservative flux variable f^h at any gridless point located at (x_p, y_p) (either on the body or in the field) can be written in terms of N surrounding cloud of nodes, and a shape function ϕ using

$$f^h(x_p, y_p) = \sum_{n=1}^{N} \phi_n(x_p, y_p) f_n \tag{12}$$

The spatial derivative of the flux f^h in the kth direction ($k = 1, 2$) can be written in terms of a derivative shape function as

$$\frac{\partial f^h(x_p, y_p)}{\partial x_k} = \sum_{n=1}^{N} \frac{\partial \phi_n(x_p, y_p)}{\partial x_k} f_n \tag{13}$$

It is pointed out that the shape functions have the following property

$$\sum_{n=1}^{N} \phi_n = 1, \quad \sum_{n=1}^{N} \frac{\partial \phi_n}{\partial x_k} = 0 \tag{14}$$

Note that, as discussed in Ref. [27], the N fitted cloud nodes are selected from the appropriate upwind direction consistent with the split flux component under consideration.

For Cartesian field nodes near the body designated as "gridless", flux discretization in (10) for directions having an incomplete stencil are substituted with expressions based on the shape function representation

$$\begin{aligned} \frac{\partial \mathbf{f}^{\pm}}{\partial x} &= \sum_{n=1}^{N_x^{\mp}} \frac{\partial \phi_n^{\pm}}{\partial x} \mathbf{f}_n^{\pm} \\ \frac{\partial \mathbf{g}^{\pm}}{\partial y} &= \sum_{n=1}^{N_y^{\mp}} \frac{\partial \phi_n^{\pm}}{\partial y} \mathbf{g}_n^{\pm} \end{aligned} \tag{15}$$

For surface nodes, the boundary condition is specified using the normal gradient of flow properties at the wall. Namely, one can write

$$\begin{aligned} &\frac{\partial \rho}{\partial \tilde{y}} = 0 \\ &\frac{\partial \tilde{u}}{\partial \tilde{y}} = 0 \\ &\tilde{v} = V_{\mathrm{n}} \\ &\frac{\partial p}{\partial \tilde{y}} = \frac{\rho \tilde{u}^2}{R} \end{aligned} \tag{16}$$

where V_{n} is the surface normal velocity at node p based on the motion of the body. In Eqs. (16), the first two relations are reflection conditions, and the last equation represents a balance between the pressure in the fluid and the centrifugal force associated with the fluid motion along a curved path defined by the local surface radius of curvature, R. The tildes indicate that the velocity components are in a local $(\tilde{x}, \tilde{y})$ coordinate system, where $\tilde{x}$ and $\tilde{y}$ are in the local tangential and normal directions of the surface, respectively. The surface boundary conditions of Eqs. (16) are written in terms of gridless shape functions at any surface node p as

$$\begin{aligned} \rho_p &= -\frac{1}{\partial \phi_p^- / \partial \tilde{y}} \left(\sum_{\substack{n=1 \\ n \neq p}}^{N_y^+} \frac{\partial \phi_n^-}{\partial \tilde{y}} \rho_n \right) \\ \tilde{u}_p &= -\frac{1}{\partial \phi_p^- / \partial \tilde{y}} \left(\sum_{\substack{n=1 \\ n \neq p}}^{N_y^+} \frac{\partial \phi_n^-}{\partial \tilde{y}} \tilde{u}_n \right) \\ \tilde{v}_p &= V_{\mathrm{n}} \\ p_p &= \frac{1}{\partial \phi_p^- / \partial \tilde{y}} \left(\frac{\rho_p \tilde{u}_p^2}{R_p} - \sum_{\substack{n=1 \\ n \neq p}}^{N_y^+} \frac{\partial \phi_n^-}{\partial \tilde{y}} p_n \right) \end{aligned} \tag{17}$$

The above equations are evaluated subsequent to the field nodes after each time step. Also, evaluation of the shape functions at the various gridless nodes as a pre-processing activity precludes the requirement for least-squares fitting at every time step, allowing for efficient implementation of the scheme.

In forming the shape functions for the gridless field nodes, three points are fit in each flux direction using a polynomial basis given by [1, x, y]. For wall points, six points are fit using a six-term basis, given by [1, x, y, xy, x^2, y^2].

3.3. A perturbation method for the gridless boundary conditions

In the analyses considered here, a time-accurate solution of a moving (oscillating) body is addressed. The boundary conditions listed in Eq. (17) must be applied at every instant with the instantaneous values of the shape functions and flow quantities in an exact implementation. The local surface normal direction and the surface normal velocity V_{n} can be calculated exactly based on the known motion of the airfoil surface independent of the use of a moving or non-moving computational grid. The changes in the derivative of the shape functions that appear in Eq. (17) reflect the motion of the grid due to the motion of the airfoil surface. Because the displacement of the airfoil from its mean position is assumed to be small in a flutter simulation,

a perturbation of the derivative of the gridless shape functions can be used in an approximate numerical boundary treatment without the requirement for a deforming mesh.

In such an approximation, one must establish the manner in which the local normal derivatives of the gridless shape functions, as they appear in Eq. (17), vary due to the motion of the airfoil. As a first consideration, it is noted that the most influential effect is due to the change in the orientation of the airfoil surface normal. The effects of translational displacement are not as significant. Thus, in considering a perturbation boundary condition for small displacements, it can be assumed that the relative distances of the various nodes are essentially fixed, and the dominant factor in the perturbation is the change in relative angle between the various nodes.

Recall that for a surface node p, the boundary condition is specified using shape functions established at surrounding N nodes. Writing the derivative shape function for one of these surrounding nodes, n, in the abbreviated notation ϕ'_n, one can consider a perturbation of the derivative shape function given by

$$\hat{\phi}'_n = \phi'_n + \frac{\partial \phi'_n}{\partial \theta}\Delta\theta \tag{18}$$

where $\Delta\theta$ represents a small angular displacement of the airfoil surface normal at p, and $\hat{\phi}'_n$ is the perturbed value of the normal derivative of the shape function associated with the displacement.

For small displacements, if one assumes that $\partial\phi'_n/\partial\theta$ is constant, Eq. (18) relates a linear variation of derivative shape function with angular separation between p and n. It is interesting to note that the least-squares shape functions exhibit such nearly-linear behavior for small deflections. For instance, consider the six-point gridless cloud associated with the surface node represented in Fig. 1 where each node in the cloud is given a reference number, node 1 being the surface node p. If one were to calculate the least-squares shape functions

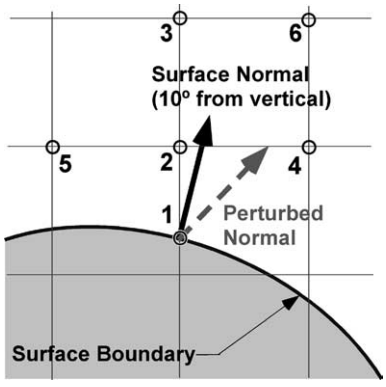

Fig. 1. Example of surface node gridless cloud with perturbed surface normal.

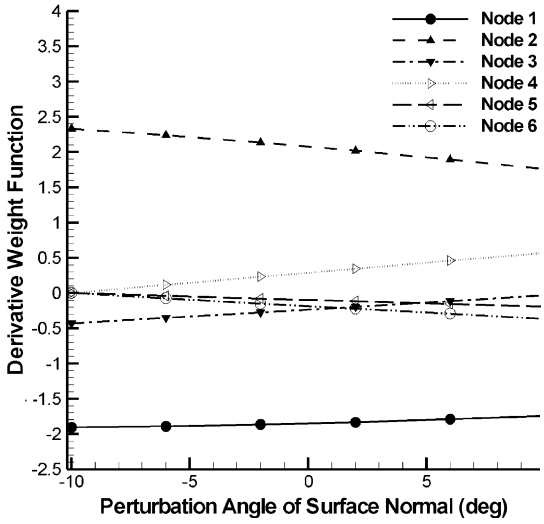

Fig. 2. Variation of derivative shape function with perturbation angle.

at the various nodes for a $\pm 10°$ deviation in the surface normal, the variation in the gridless shape functions for the different nodes is nearly linear, as shown in Fig. 2. As a result, a constant value of the $\partial\phi'_n/\partial\theta$ can be established as a pre-processing activity using a difference of the nominal values and a second value based on the maximum anticipated angular deviation of the surface normal. New shape functions for the various nodes can then be quickly established for various surface orientations throughout the evolution of the solution.

It is noted that the only approximation made in the above perturbation method is in the evaluation of the derivative of the shape functions. The surface normal directions and the surface normal velocities at each surface point are accurately represented. This perturbation method is similar in concept to that used in Gao et al. [32] in a pure Cartesian grid method, but the present method is not restricted to thin airfoils and does not suffer from the leading-edge singularity associated with the mean chord approximation used in Ref. [32].

4. Temporal discretization

Temporal discretization is achieved using a second order accurate implicit backwards difference formulation based on the work of Jameson [36]. For a stationary mesh, one has at node j for the vector of conserved variables, $\mathbf{u}_j$, at various physical time levels, n

$$\frac{3}{2\Delta t}\mathbf{u}_j^{n+1} - \frac{2}{\Delta t}\mathbf{u}_j^{n} + \frac{1}{2\Delta t}\mathbf{u}_j^{n-1} = -\mathbf{R}(\mathbf{u}_j^{n+1}) \tag{19}$$

where the residual at node j, $\mathbf{R}\left(\mathbf{u}_j^{n+1}\right)$, is a discretized representation of the spatial derivative (e.g. right-hand side of Eq. (10)). Eq. (19) can be solved in an iterative fashion by considering a steady-state Runge–Kutta time-marching formulation in pseudo-time using multi-

grid acceleration and local time stepping [37]. Namely, one forms an unsteady contribution to the residual and rewrites (19) as

$$\frac{d\mathbf{u}}{dt^*} = -\mathbf{R}^*(\mathbf{u}_j) \tag{20}$$

where

$$\mathbf{R}^*(\mathbf{u}_j) = \mathbf{R}(\mathbf{u}_j) + \frac{3}{2\Delta t}\mathbf{u}_j - \frac{2}{\Delta t}\mathbf{u}_j^n + \frac{1}{2\Delta t}\mathbf{u}_j^{n-1} \tag{21}$$

Within a given physical time level, n, the Runge–Kutta time marching in pseudo-time is written

$$\begin{aligned}
\mathbf{u}_j^{(0)} &= (\mathbf{u}_j^*)^l \\
\mathbf{u}_j^{(1)} &= \mathbf{u}_j^{(0)} - \alpha_1 \Delta t_j^* \mathbf{R}^*(\mathbf{u}_j^{(0)}) \\
\mathbf{u}_j^{(2)} &= \mathbf{u}_j^{(0)} - \alpha_2 \Delta t_j^* \mathbf{R}^*(\mathbf{u}_j^{(1)}) \\
&\vdots \\
(\mathbf{u}_j^*)^{l+1} &= \mathbf{u}_j^{(0)} - \alpha_k \Delta t_j^* \mathbf{R}^*(\mathbf{u}_j^{(m-1)})
\end{aligned} \tag{22}$$

where Δt^* is the local allowable pseudo-time step at node j. The superscript $*$ indicates intermediate values within each pseudo-time. The time-marching scheme represented by Eq. (22) takes the solution from pseudo-time level l to $l+1$, in which multi-grid and local time stepping (in pseudo-time) are used to accelerate the convergence of the solution. Iterations in pseudo-time are continued until the residual is sufficiently reduced such that values of the conserved variables at pseudo-time level $l+1$ approximate those at physical time level $n+1$. Using this approach, the physical time step can be chosen based on the physical time scales of the moving body, without restrictions typically associated with explicit time-marching methods (i.e. propagation of acoustic signals). A four-stage scheme is considered here using coefficients recommended by Van Leer et al. [38], given as

$$\alpha_k = \{0.0833, 0.2069, 0.4265, 1.0000\} \tag{23}$$

Although the physical time step may be selected based on the time varying characteristics of the physical system, the time marching in pseudo-time is conducted in an explicit fashion and remains subject to constraints imposed by the CFL condition. For a constant grid spacing Δx in both Cartesian coordinate directions, the CFL number is defined by

$$\sigma = \frac{\Delta t}{\Delta x}[(|u| + c) + (|v| + c)] \tag{24}$$

For evolution of the gridless fluxes at node p, the CFL number is taken to be

$$\sigma = \frac{1}{2}\frac{a\Delta t}{\Delta x}\left(\phi'_p + \sum_{n\neq p}^{N} |\phi'_n|\right) \tag{25}$$

as derived in Ref. [39] where a is the magnitude of the net wave propagation velocity.

It is noted that in using the multi-stage scheme of Eq. (22) in a dual time stepping approach, instabilities can occur if the physical time step, Δt, is on the same order (or smaller) than the pseudo-time step, Δt^* [40]. This fact imposes an additional constraint on the magnitude of the pseudo-time step.

5. Fluid/structure dynamic coupling

In order to consider aeroelastic phenomena, the two-degree-of-freedom wing model of Isogai [41,42] is considered. This model is represented in Fig. 3, in which an airfoil of chord length $2b$ is shown. The model consists of a damped spring–mass system in which the pitching motion represents twisting of the wing, and the plunging motion represents bending of the wing along the span. For details of the structural model, the reader is referred to Ref. [34,11].

In short, the motion of the mechanical system is decoupled into a generalized displacement vector $(z_{1,i}, z_{2,i})$ for each of the two vibrational modes of freedom, i. The solution vector $\mathbf{z}_i$, for each ith vibrational mode, is incorporated into a Runge–Kutta time stepping scheme using a second-order discretization similar to the flow field in Eq. (19). Namely, one can write

$$\begin{aligned}
\frac{3}{2\Delta\tau}z_{1,i}^{n+1} - \frac{2}{\Delta\tau}z_{1,i}^{n} + \frac{1}{2\Delta\tau}z_{1,i}^{n-1} &= -\mathbf{R}(z_{1,i}^{n+1}, Q_i^{n+1}) \\
\frac{3}{2\Delta\tau}z_{2,i}^{n+1} - \frac{2}{\Delta\tau}z_{2,i}^{n} + \frac{1}{2\Delta\tau}z_{2,i}^{n-1} &= -\mathbf{R}(z_{2,i}^{n+1}, Q_i^{n+1})
\end{aligned} \tag{26}$$

where the residual is written not only in terms of z, but also in terms of a scaled aerodynamic forcing function, Q_i. For the exact form of Q_i, the reader is referred to Ref. [34,11], in which the time-marching scheme was originally formulated. The above can be reformulated into a pseudo-time formulation, similar to the flow field Eqs. (20) and (21) by

$$\begin{aligned}
\frac{dz_{1,i}}{d\tau^*} &= -\mathbf{R}^*_{1,i}(z^*_{1,i}, Q^*_{1,i}) \\
\frac{dz_{2,i}}{d\tau^*} &= -\mathbf{R}^*_{2,i}(z^*_{2,i}, Q^*_{2,i})
\end{aligned} \tag{27}$$

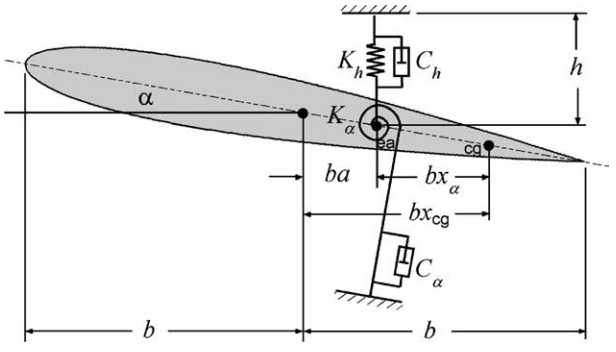

Fig. 3. Two-dimensional wing flutter model.

where

$$\mathbf{R}^*_{1,i}(z^*_{1,i}, Q^*_i) = \mathbf{R}_{1,i}(z^*_{1,i}, Q^*_i) + \frac{3}{2\Delta\tau} z^*_{1,i} - \frac{2}{\Delta\tau t} z^n_{1,i} + \frac{1}{2\Delta\tau} z^{n-1}_{1,i}$$
$$\mathbf{R}^*_{2,i}(z^*_{2,i}, Q^*_i) = \mathbf{R}_{2,i}(z^*_{2,i}, Q^*_i) + \frac{3}{2\Delta\tau} z^*_{2,i} - \frac{2}{\Delta\tau} z^n_{2,i} + \frac{1}{2\Delta\tau} z^{n-1}_{2,i} \quad (28)$$

Eqs. (20), (21), (27) and (28) enable the coupling of the flow field and mechanical response of the wing model by providing one single system of equations in pseudo-time. Upon iterating to steady state in pseudo-time, the system is fully time accurate at any given physical time without any lag between the fluid dynamic and structural dynamic equations. Since the equations governing the structural response converge much faster, however, it is more efficient to perform several iterations of the flow equations in pseudo-time (typically 5 or 10) per iteration of the equations governing the structural response.

6. Results and discussion

6.1. Steady flow results

Before attempting time-accurate simulations using the perturbation boundary condition, the effectiveness of the treatment for steady flow simulations is first investigated. Namely, the effectiveness of the method in approximating angle of attack effects is established. In this pursuit, comparisons are made for simulation of flow about the NACA 0012 airfoil at Mach 0.8 for angles of attack varying between $\pm 6°$ using the FLO52 code of Jameson. In these comparisons, FLO52 simulations are performed with the angle of attack prescribed as a far field boundary condition. In the Cartesian mesh simulations, the effects of angle of attack are completely imposed by the perturbation boundary condition.

In order to establish the differential term $\partial\phi'_n/\partial\theta$ in Eq. (18), gridless shape functions are evaluated for the direction of the mean surface normal, as well as that of a positive 5° deflection, and forming the difference of the two values. It is noted that there are actually two approximations to the gridless boundary condition being assessed here. One is that the surface geometry can be represented by the pseudo-normal. The second is that the differential term is constant for the range of deflection angles considered ($\pm 6°$). Although one would expect lift and moment coefficients to be anti-symmetric with angle of attack, it is important to present results for both positive and negative angles attack cases to verify the assumption of linearity of the shape function variation with pitch angle.

The computational mesh used for the steady flow Cartesian mesh simulation is presented in Fig. 4 in which the finest resolution is 0.01 of the chord length and there are 10,418 total cells. Considering a free-stream Mach number of 0.8, Fig. 5 presents the predicted lift, drag, and moment coefficients as compared to the FLO52 simulations. As shown, good agreement with FLO52 is demonstrated for lift and drag coefficients, and the perturbation method retains symmetry between the positive and negative angle of attack positions. Reasonable, though not as accurate results are obtained for the moment coefficient. For all force coefficients, the discrepancy grows with simulated angle of attack as expected since the perturbation of the surface normal becomes more unrealistic. The surface pressure coefficient distribution is presented in Fig. 6. Excellent agreement with FLO52 is indicated, particularly since all angle of attack effects on the Cartesian mesh simulation were generated using the perturbed boundary conditions.

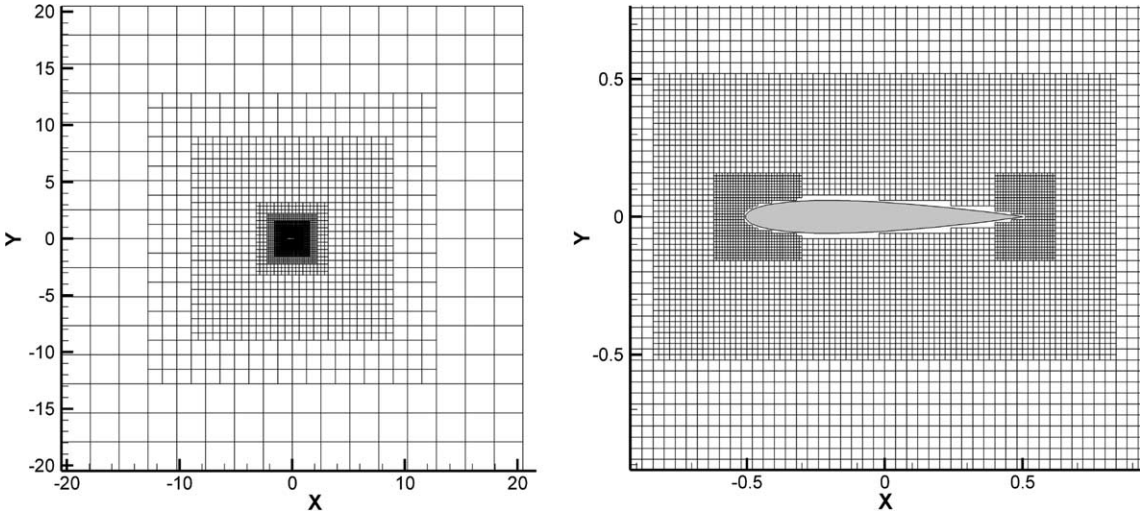

Fig. 4. Cartesian mesh for steady flow perturbation boundary condition study.

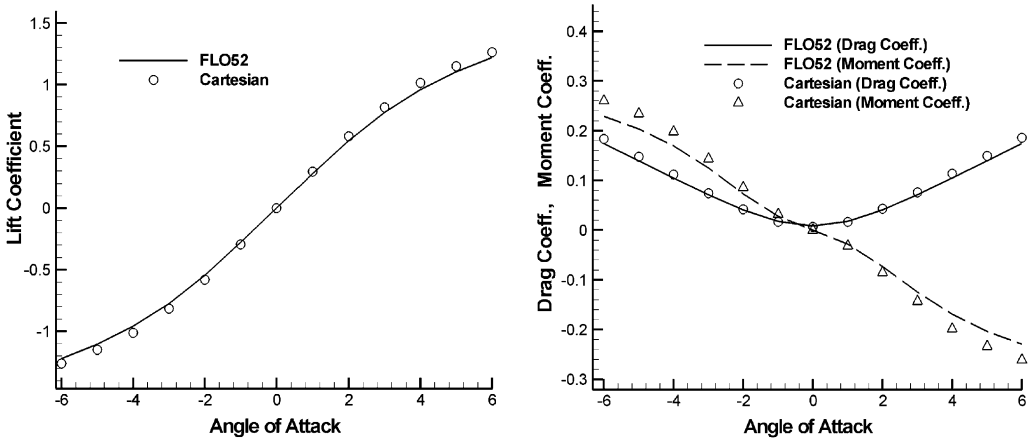

Fig. 5. Force coefficients using perturbed surface normal to establish angle of attack.

6.2. Unsteady flow results

Next, the application of the boundary treatment to the prediction of unsteady flows is presented. Here, a simulation using the perturbed gridless boundary condition is compared to the experimental results of Landon [33]. This simulation considers the prescribed oscillating motion of the NACA 0012 airfoil in which the instantaneous motion is given as a function of time by

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t) \tag{29}$$

For this analysis, the mean angle of attack, α_m, is prescribed by the far field boundary condition, and perturbation of the surface normal will be used to account for the oscillating pitch of the airfoil.

The particular experimental results being considered here is that of AGARD CT Case 5. The case considers the NACA 0012 airfoil with a free-stream Mach number of 0.755, and a mean angle of attack of $\alpha_m = 0.016°$. The amplitude of the oscillation is given as $\alpha_0 = 2.51°$, and the reduced frequency is given as $\kappa_c = 0.0814$ where

$$\kappa_c = \frac{\omega_\alpha b}{V_\infty} \tag{30}$$

The computational mesh for this simulation is identical to that used for the steady-state simulation as shown in Fig. 4. The unsteady simulation is initiated from a steady flow solution with the mean angle of attack prescribed as a far field condition. In performing the unsteady solution, one must give consideration to the size of the physical time step. In these results, sufficient temporal independence was achieved by dividing the period of the airfoil oscillation into 32 steps to establish the physical time step. Furthermore, 50 iterations (in pseudo-time) per physical time step were found to be sufficient for convergence in pseudo-time. On a Pentium III 800 MHz, each of the pseudo-time iterations required approximately 1.0 CPU seconds, so that one complete period of the airfoil was simulated in roughly 2.5 min.

The lift and moment coefficient of the oscillating airfoil versus structural time, $\tau = \omega_\alpha t$, is presented in Fig. 7. As shown, the solution becomes periodic after roughly one oscillation. Upon attainment of periodic motion, the surface pressure coefficient is compared to experimental data for various phase angles of the oscillation in Figs. 8 and 9. For phase angles from 0° to 180°, shown in Fig. 8, the airfoil has a positive angle of attack. For phase angles between 180° and 350°, shown in Fig. 9, the angle of attack is negative. As shown, the calculated pressure coefficients agree well with the experimental data.

A more succinct method of comparing the predictions with the experimental data is by means of the variation in lift and moment coefficient versus angle of attack. This comparison is presented in Fig. 10 in which reasonable agreement is shown. A final comparison with experimental data is made based on the Fourier decomposition of the time variation of surface pressure coefficient. A Fourier analysis of the pressure variation over the surface of the airfoil is important to gauge whether the method is capturing the correct phase lag of the shock with respect to the oscillation of the airfoil. The degree of phase lag plays a dominant role in the manifestation of flutter. The real and imaginary components of the first three Fourier modes are presented in Fig. 11. Comparisons for the first mode, which is the dominating contribution, are in excellent agreement as shown for both the real and imaginary components, indicating that the phase lag is being properly calculated.

6.3. Simulation of transonic flutter

The application of the method to the prediction of transonic flutter is now considered. In particular, the wing model of Isogai [41,42] (Case A) as represented by Fig. 3 is addressed. Here, the flutter boundary for a wing section incorporating the NACA 64A0010 airfoil

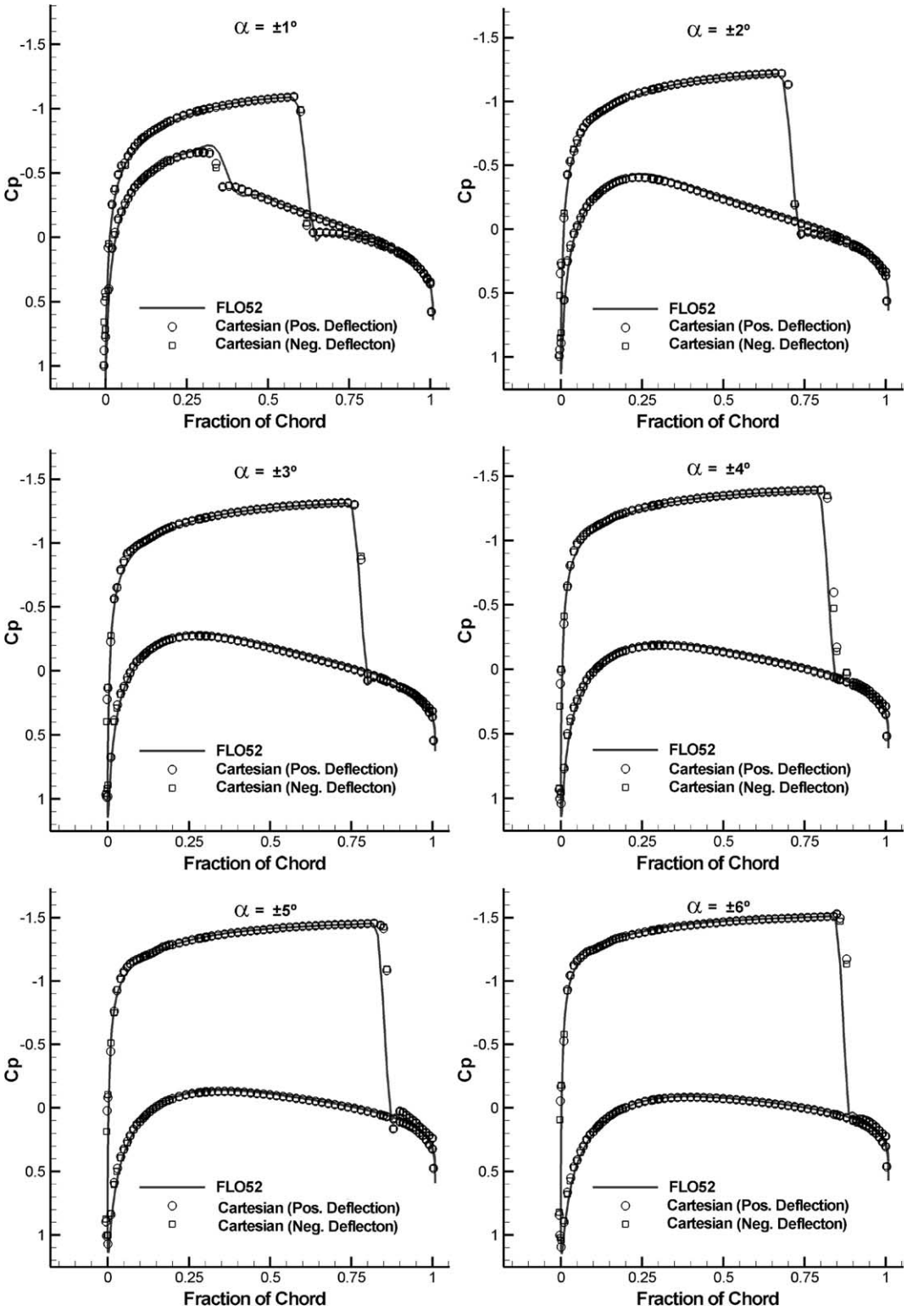

Fig. 6. Pressure coefficients using perturbed surface normal to establish AOA.

is studied. The flutter boundary is established using an iterative process in which aeroelastic simulations are performed for a given Mach number and various values of the flutter speed index, V_F, given by

$$V_F = \frac{V_\infty}{b\omega_\alpha\sqrt{\mu}} \tag{31}$$

where μ is the mass ratio, taken here to be 60. Starting from an initial steady solution, the model is "kicked"

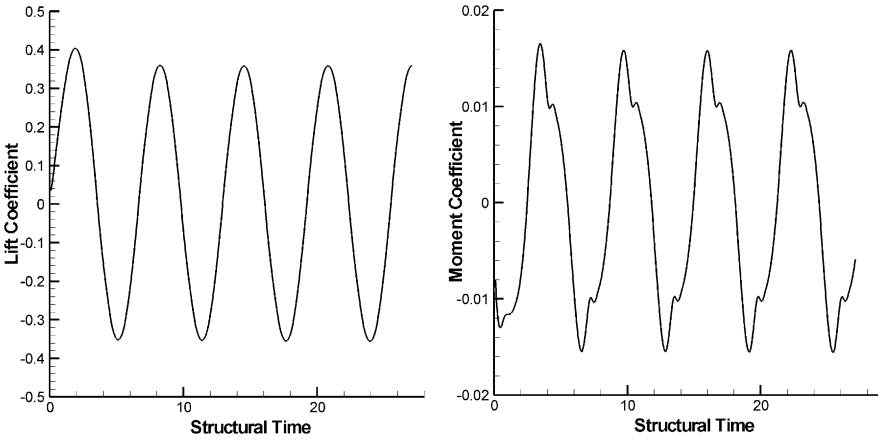

Fig. 7. Variation of lift and moment coefficient with structural time for prescribed oscillation of NACA 0012 airfoil.

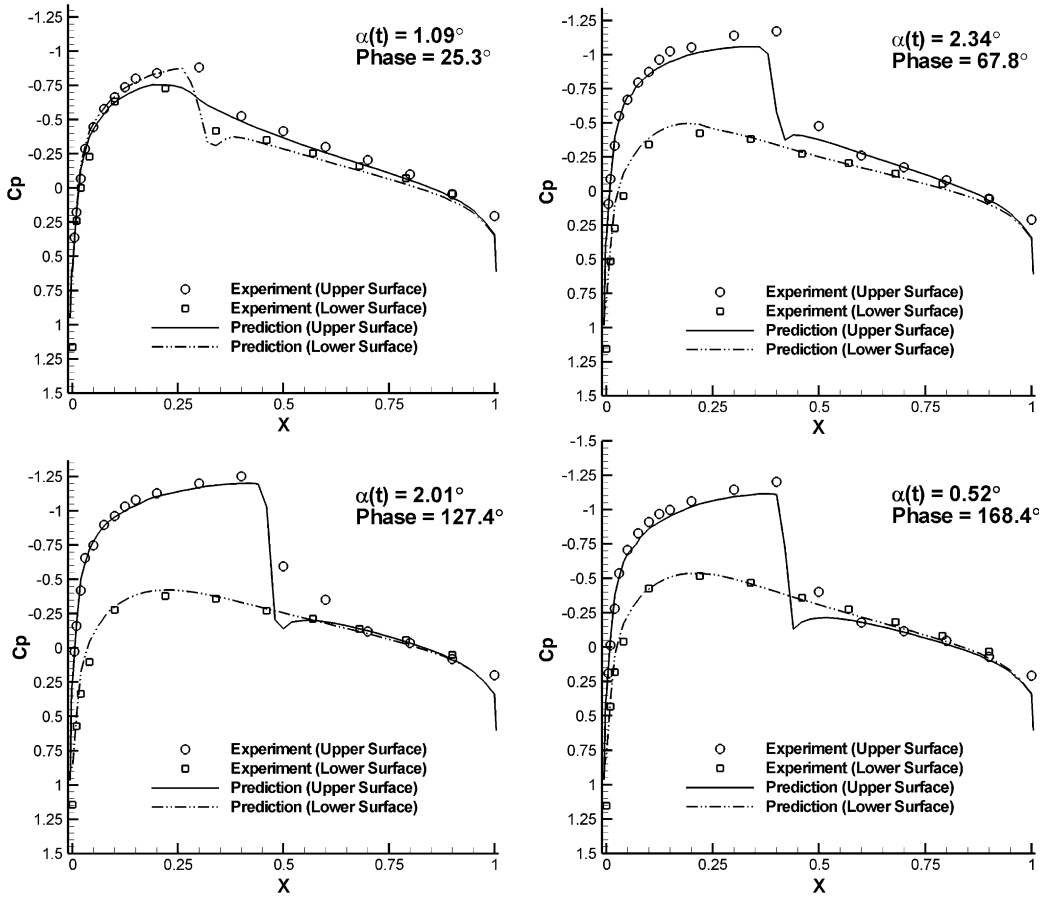

Fig. 8. Comparison of surface pressure coefficient with experimental data for prescribed oscillation of NACA airfoil (positive angles of attack).

with a small disturbance in pitching motion given as $d\tau/d\alpha = 0.01$. If the resulting motion grows in an unbounded fashion with time, the system is considered unstable and prone to aeroelastic flutter. If the disturbances are damped with time, the system is stable, and flutter does not occur. If the system continues to oscillate with constant amplitude, the system is neutrally stable and the flutter boundary is established.

The computational mesh for the aeroelastic simulation is presented in Fig. 12, which has 10,351 cells and a finest resolution of 0.01 chord lengths. In an effort to compare with the results of Alonso and Jameson [34] and Liu et al. [11], in which moving mesh simulations were considered (e.g. [43]), six different Mach numbers are considered for evaluation of the flutter boundary. These are Mach 0.75, 0.8, 0.825, 0.85, 0.875, and 0.9.

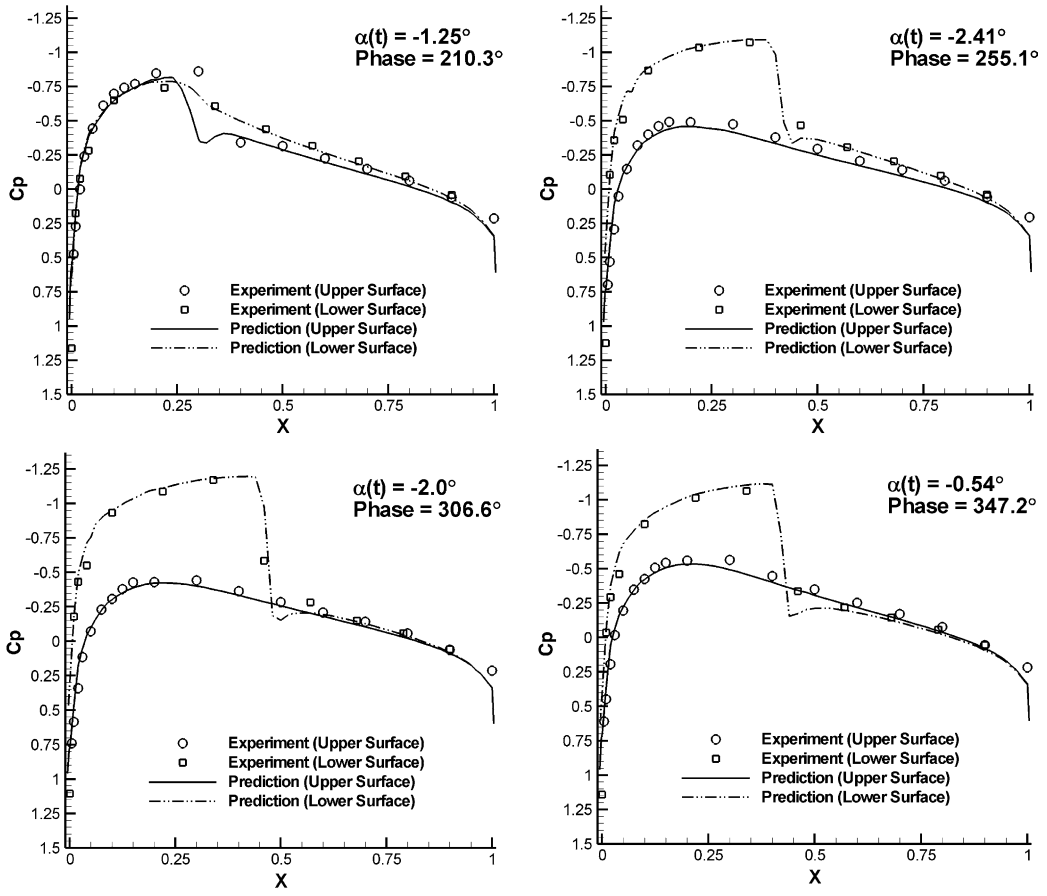

Fig. 9. Comparison of surface pressure coefficient with experimental data for prescribed oscillation of NACA airfoil (negative angles of attack).

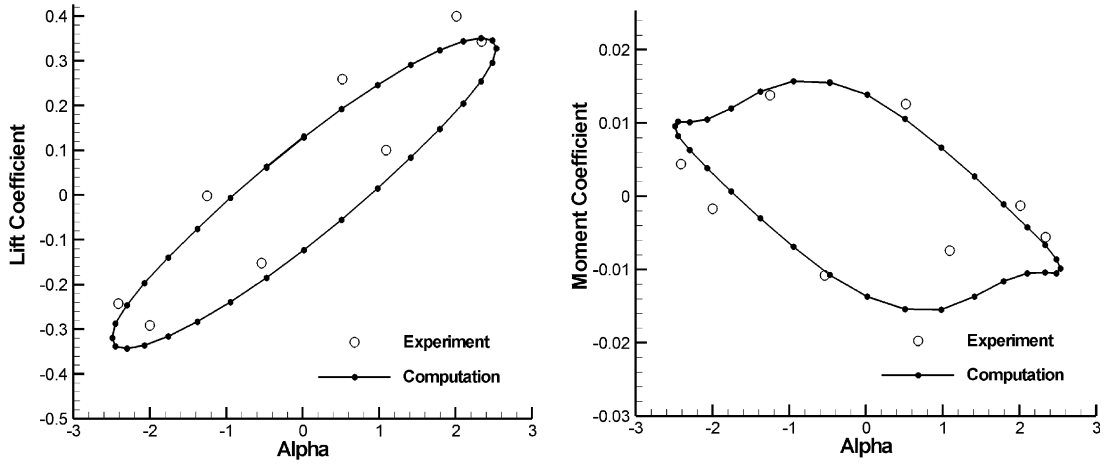

Fig. 10. Comparison of computed lift coefficient with experimental data for prescribed oscillation of NACAA 0012 airfoil.

As an example of the iterative process, the behavior of the system with flutter speed index at Mach 0.85 is provided in Fig. 13. As shown, the method predicts a neutral point at a flutter speed index of 0.5. Furthermore, the system is shown to be unstable for larger values of V_F, and stable for smaller values.

As discussed by Alonso and Jameson [34] and Gao et al. [32], the system under consideration exhibits multiple flutter points in the vicinity of Mach 0.875 such that an inflection is present in the flutter boundary. Namely, once the lower stability point is crossed, the system may again become stable as the flutter speed index is increased when a second neutral point is crossed. Upon further increase of the flutters speed index, a third neutral point is crossed and the system again becomes unstable. Hall et al. [44] predicted similar multiple flutter

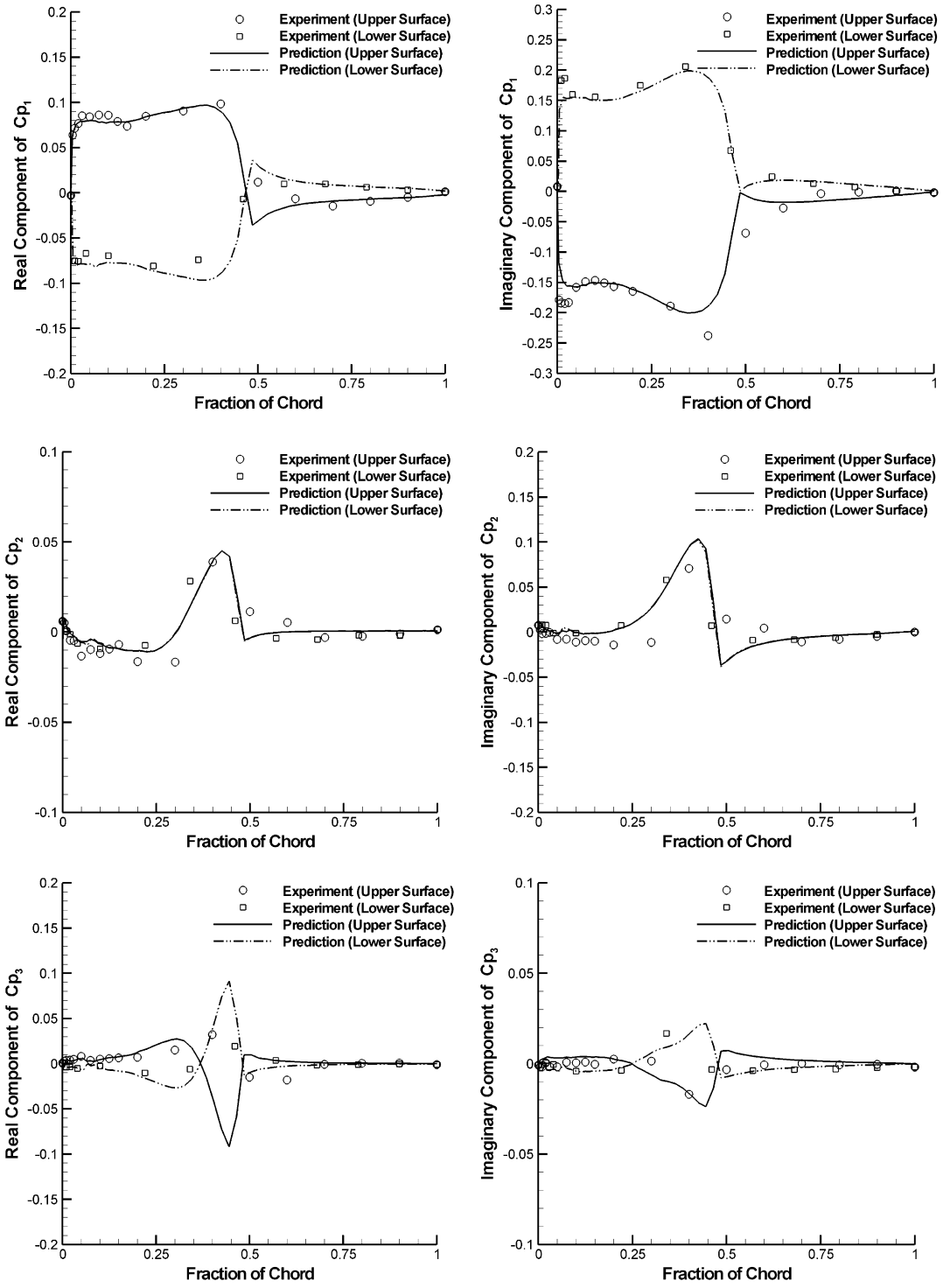

Fig. 11. Comparison of first three Fourier modes with experimental data of oscillating NACA airfoil.

points by tracing the root locus of the least stable eigenvalue of the Isagai aeroelastic system using a reduced-order model for the aerodynamics. Establishment of the lower, middle and upper flutter boundary points at Mach 0.875 by the present time-domain method is presented in Figs. 14–16, respectively. As shown in the vicinity of the middle neutral point (i.e. Fig. 15), the system increases in stability as the flutter speed index is increased.

In performing aeroelastic simulations for all of the Mach numbers under consideration, a plot of the flutter boundary is established and presented in Fig. 17, as compared to the moving mesh simulation of Alonso and Jameson [34] and Liu et al. [11]. As shown, results

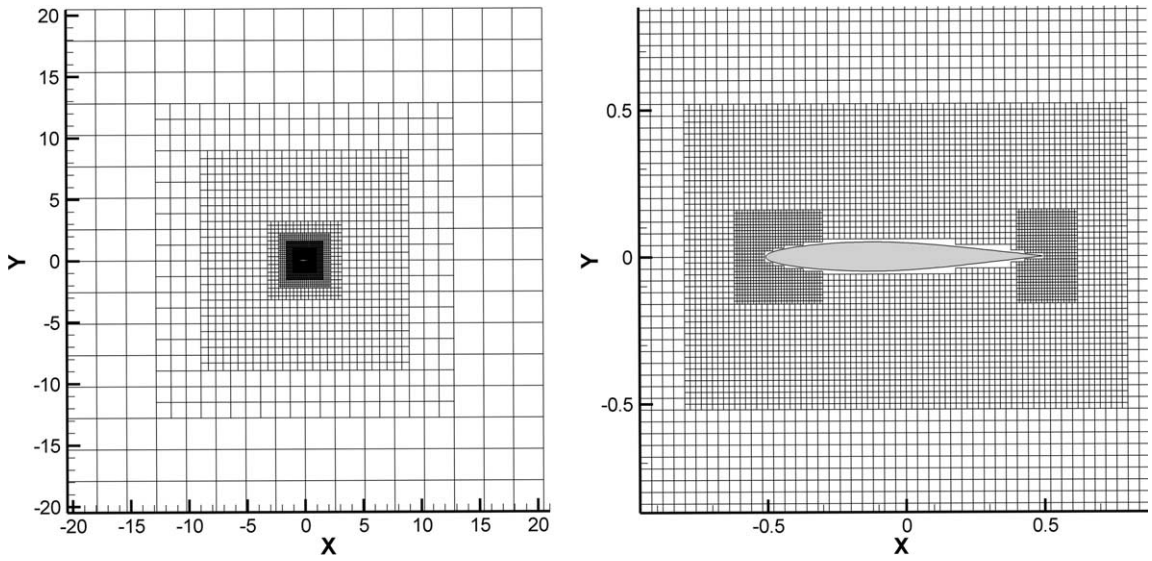

Fig. 12. Cartesian mesh for aeroelastic simulation of NACA 64A010.

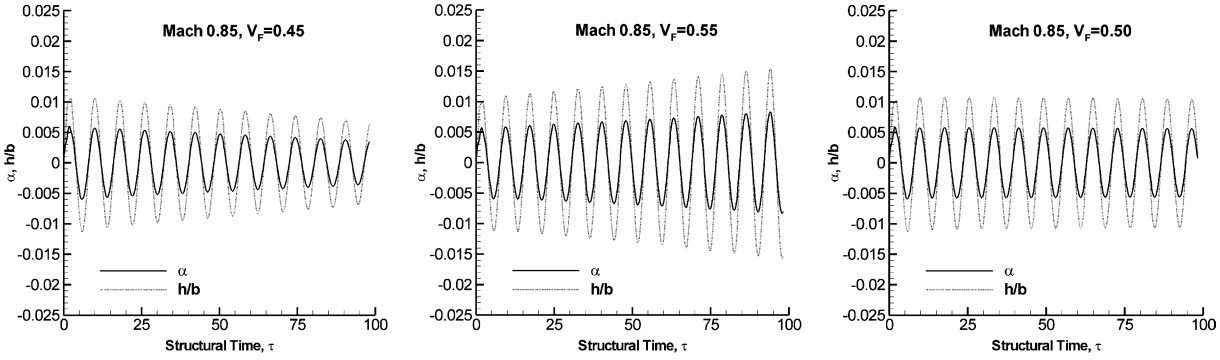

Fig. 13. Sensitivity of NACA 64A010 wing model to flutter at Mach 0.85.

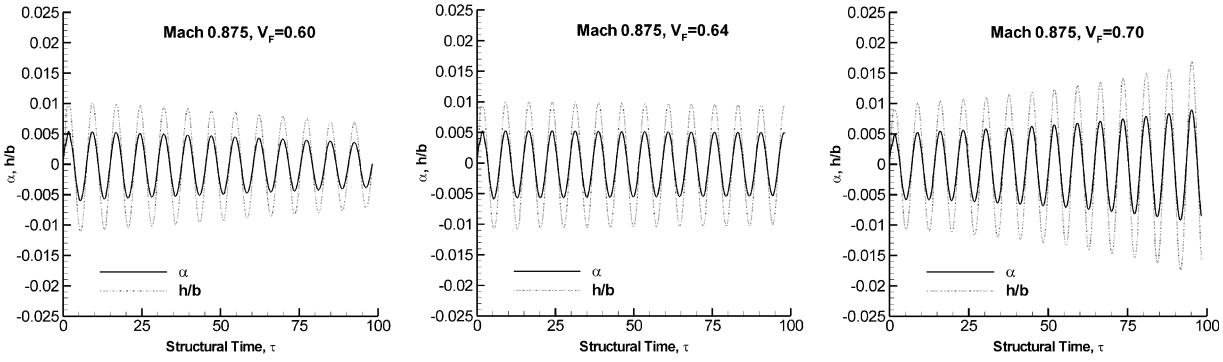

Fig. 14. Sensitivity of NACA 64A010 wing model to flutter at Mach 0.875.

are in excellent agreement, especially at the lower Mach numbers. Also shown in Fig. 17 is a best estimate curve-fit of the flutter boundary in which the presence of an inflection point between Mach 0.85 and Mach 0.9 is depicted.

Lastly, a prediction is made for the existence of limit cycle oscillation (LCO) at Mach 0.75 and a flutter speed index of 1.33, as originally reported by Kousen and Bendiksen [45], and verified by Alonso and Jameson [34]. Using the present approach, the predicted occurrence of LCO is presented in Fig. 18. As shown, LCO is exhibited after many oscillations of the system, as is defined by an initial divergence followed by a limiting of the amplitude at large structural time, τ. As a result, it is

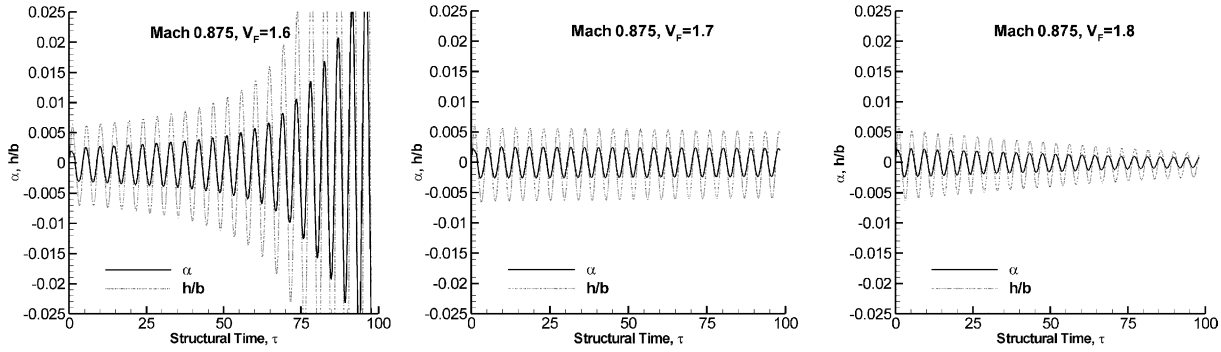

Fig. 15. Sensitivity of NACA 64A010 wing model to flutter at Mach 0.875.

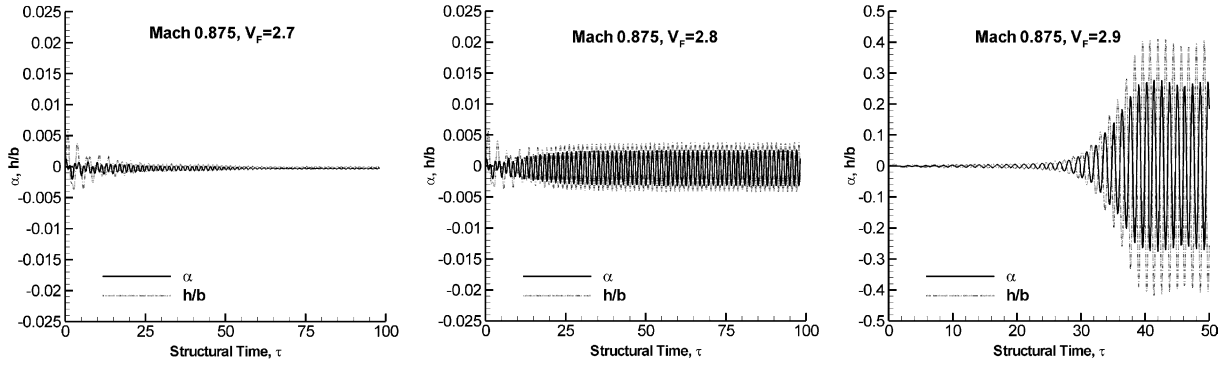

Fig. 16. Sensitivity of NACA 64A010 wing model to flutter at Mach 0.875.

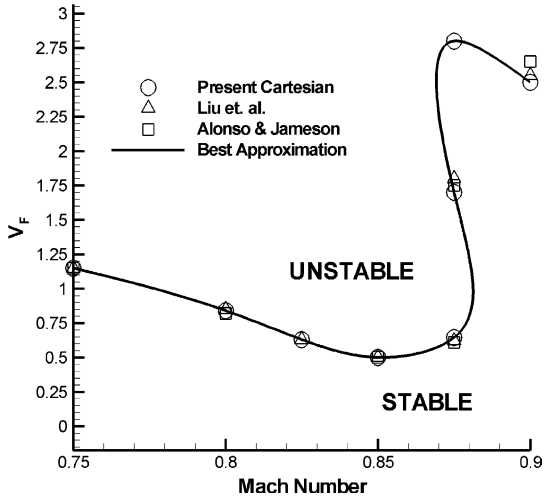

Fig. 17. Comparison of predicted NACA 64A10 flutter boundary.

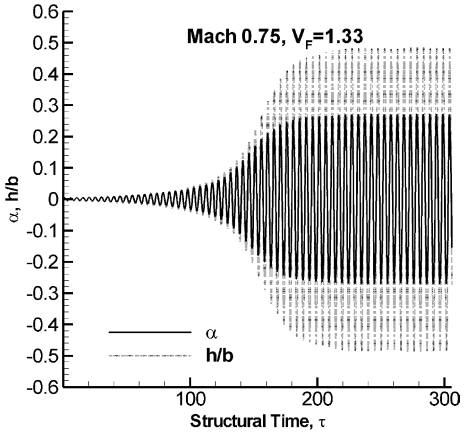

Fig. 18. Example of NACA 64A010 wing model limit cycle oscillation at Mach 0.75, $V_F = 1.33$.

shown that LCO is computationally expensive to predict due to the large number of time steps that must be simulated. Here, LCO is exhibited at a structural time of roughly 220.

It should be noted that the time step associated with these calculations was based on 1/32 of the natural period of the airfoil structural system. For the case at Mach 0.9, the time step was reduced by a factor of three due to the higher frequency of the oscillation. Similarly, for the upper neutral point, the time step was reduced by a factor of two compared to that based on the natural period. These higher flutter frequencies are associated with an anti-symmetric motion of the system with respect to pitching and plunging motion as exhibited by the higher speed indices, a phenomena that was also found by a reduced-order modeling in Ref. [44].

The current implementation of the gridless boundary condition method is limited to the Euler equations only. Viscous effects can be accounted for efficiently in an

aeroelastic simulation by using a boundary-layer approach as was demonstrated by Edwards [2]. The gridless boundary condition method is in principle equally applicable to the solution of the Navier–Stokes equations. However, problems regarding grid resolution of the Cartesian grid may arise for thin boundary layers. Proper local grid refinement strategies must be implemented.

7. Conclusions

A method for computing time-accurate Euler solutions of an oscillating airfoil was presented. Of particular interest is the ability of the method to accurately perform aeroelastic simulations without the need for a deforming mesh. The use of a patched embedded Cartesian mesh and a gridless boundary condition allows for efficient setup and execution of simulations. The method exhibits excellent agreement with experimental data for the prescribed motion of an oscillating airfoil. Furthermore, comparisons with moving mesh simulations indicate that the method provides for comparable accuracy of aeroelastic applications using a fixed mesh, greatly reducing computational requirements for such simulations.

References

[1] Batina JT. A finite-difference approximate-factorization algorithm for solution of the unsteady transonic small-disturbance equation. NASA TP 3129, January 1992.
[2] Edwards J. Transonic shock oscillations calculated with a new interactive boundary layer coupling method. AIAA Paper 93-0777, 1993.
[3] Ashely H. Role of shocks in the sub-transonic flutter phenomena. J Aircraft 1980;17:187–97.
[4] Lee-Rausch EM, Batina JT. Wing flutter boundary prediction using unsteady Euler aerodynamic method. J Aircraft 1995;32(2):416–22.
[5] Lee-Rausch EM, Batina JT. Wing flutter computations using an aerodynamic model based on the Navier–Stokes equations. J Aircraft 1996;33(6):1139–47.
[6] Byun C, Guruswamy GP. Aeroelastic computation on wing-body-control configurations on parallel computers. J Aircraft 1998;35(2):288–94.
[7] Goodwin SA, Weed RA, Sankar LN, Raj P. Toward cost-effective aeroelastic analysis on advanced parallel computing systems. J Aircraft 1999;36(4):710–5.
[8] Gordnier RE, Melville RB. Numerical simulation of limit-cycle oscillations of a cropped delta wing using the full Navier–Stokes equations. Int J Comput Fluid Dyn 2001;14(3):211–24.
[9] Raveh DE, Levy Y, Karpel M. Efficient aeroelastic analysis using computational unsteady aerodynamics. J Aircraft 2001;38(3):547–56.
[10] Schewe G, Knipfer A, Mai H, Dietz G. Nonlinear effects in transonic flutter. In: International forum on aeroelasticity and structural dynamics, Madrid, 5–7 June 2001.
[11] Liu F, Cai J, Zhu Y, Wong ASF, Tsai HM. Calculation of wing flutter by a coupled fluid-structure method. J Aircraft 2001;38(2):332–4.
[12] Tang L, Bartels R, Chen P, Liu DD. Numerical investigation of transonic limit oscillations of a 2-D supercritical wing. AIAA Paper 2001-1290, 2001.
[13] Beran P, Silva W. Reduced-order modeling: new approaches for computational physics. AIAA Paper 2001-0853, January 2001.
[14] Geuzaine P, Brown G, Harris C, Farhat C. Aeroelastic dynamic analysis of a full F-16 configuration for various flight conditions. AIAA J 2003;41(3):363–71.
[15] Harris TM, Huttsell LJ. Aeroelasticity research at Wright–Patterson air force base (Wright field) from 1953–1993. J Aircraft 2003;40(5):813–9.
[16] Yurkovich R. Status of unsteady aerodynamic prediction for flutter of high-performance aircraft. J Aircraft 2003;40(5):832–42.
[17] Schuster DM, Liu DD, Huttsell LJ. Computational aeroelasticity: success, progress, challenge. J Aircraft 2003;40(5):843–56.
[18] Dowell E, Edwards J, Strganac T. Nonlinear aeroelasticity. J Aircraft 2003;40(5):857–74.
[19] Pember RB, Bell JB, et al. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. J Comput Phys 1995;120:278–304.
[20] Melton JE, Berger MJ, Aftosmis MJ, Wong MD. 3D applications of a Cartesian grid Euler method. AIAA Paper 95-0853, 1995.
[21] Forrer H, Jeltsch R. A higher order boundary treatment for Cartesian-grid methods. J Comput Phys 1998;140:259–77.
[22] Lahur PR, Nakamura Y. A new method for thin body problem in Cartesian grid generation. AIAA Paper 99-0919, 1999.
[23] Aftosmis MJ, Berger MJ, Adomavicius G. A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries. In: 38th aerospace sciences meeting and exhibit. AIAA Paper 2000-0808, January 2000.
[24] Dadone A, Grossman B. An immersed body methodology for inviscid flows on Cartesian grids. AIAA Paper 2002-1059, January 2002.
[25] Kirshman DJ, Liu F. Gridless boundary condition treatment for a non-body-conforming mesh. In: 32nd AIAA fluid mechanics conference and exhibit. AIAA Paper 2002-3285, June 2002.
[26] Kirshman DJ, Liu F. Cartesian grid solution of the Euler equations using a gridless boundary condition treatment. In: 16th AIAA computational fluid dynamics conference. AIAA Paper 2003-3974, June 2003.
[27] Kirshman DJ, Liu F. A gridless boundary condition method for the solution of the Euler equations on embedded Cartesian meshes with multigrid. J Comput Phys 2004;201(1):119–47.
[28] Lahur PR, Nakamura Y. Simulation of flow around moving 3D body on unstructured Cartesian grid. AIAA Paper 2001-2605, June 2001.
[29] Murman SM, Aftosmis MJ, Berger MJ. Implicit approaches for moving boundaries in a 3-D Cartesian method. In: 41st Aerospace sciences meeting. AIAA Paper 2003-1119, January 2003.
[30] Gao C, Luo S, Liu F, Schuster DM. Calculation of unsteady transonic flow by an Euler method with small perturbation boundary conditions. In: 41st Aerospace sciences meeting and exhibit. AIAA Paper 2003-1267, January 2003.
[31] Gao C, Luo S, Liu F, Schuster DM. Calculation of airfoil flutter by an Euler method with approximate boundary conditions. In: 16th AIAA computational fluid dynamics conference. AIAA Paper 2003-3830, June 2003.
[32] Gao C, Yang S, Luo S, Liu F, Schuster DM. Calculation of airfoil flutter by an Euler method with approximate boundary conditions. AIAA J 2005;43(2):295–305.
[33] Landon RH. NACA 0012 oscillating and transient pitching, Compendium of Unsteady Aerodynamic Measurements, Data Set 3. AGARD Report R-702, August 1982.
[34] Alonso JJ, Jameson A. Fully implicit time-marching aeroelastic solutions. AIAA Paper 94-0056, January 1994.

[35] Van Leer B. Flux-vector splitting for the Euler equations. Proc 8th international conference on numerical methods in fluid dynamics. Springer-Verlag; 1982. p. 507–12.
[36] Jameson A. Time dependent calculations using multi-grid with application to unsteady flows past airfoils and wings. In: AIAA 10th computational fluid dynamics conference. AIAA Paper 91-1956, June 1991.
[37] Jameson A, Schmidt W, Turkel E. Numerical solutions of the Euler equations by finite volume methods using Runge–Kutta time-stepping. In: AIAA 14th fluid and plasma dynamics conference. AIAA Paper 81-1259, June 1981.
[38] Van Leer B, Tai C-H, Powell KG. Design of optimally smoothing multi-stage schemes for the Euler equations. AIAA Paper 89-1933, 1989.
[39] Kirshman DJ. Calculation of steady and unsteady transonic flow using a Cartesian mesh and gridless boundary conditions with application to aeroelasticity. PhD Thesis, University of California, Irvine, 2003.
[40] Arnone A, Liou MS, Povinelli LA. Multigrid time-accurate integration of Navier–Stokes equations. AIAA Paper 93-3361, 1993.
[41] Isogai K. On the transonic-dip mechanism of flutter of a sweptback wing. AIAA J 1979;17(7):793–5.
[42] Isogai K. On the transonic-dip mechanism of flutter of a sweptback wing: Part II. AIAA J 1981;19(7):1240–2.
[43] Tsai HM, Wong ASF, Cai J, Zhu Y, Liu F. Unsteady flow calculations with a parallel multiblock moving mesh algorithm. AIAA J 2001;39(6):1021–9.
[44] Hall KC, Thomas JP, Dowell EH. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. AIAA J 2000;38(10).
[45] Kousen KA, Bendiksen OO. Limit cycle phenomena in computational transonic aeroelasticity. J Aircraft 1994;31(6):1257–63.