

# Higher-Order Time-Integration Schemes for Aeroelastic Applications on Unstructured Meshes

Zhi Yang\* and Dimitri J. Mavriplis†  
University of Wyoming, Laramie, Wyoming 82071

DOI: 10.2514/1.22847

Efficient techniques for computational aeroelasticity on unstructured meshes are investigated. These include the formulation of a fourth-order implicit Runge–Kutta time-integration scheme, a third-order backward-difference time-integration scheme, and a second-order backward-difference time-integration scheme for the coupled system of flow and structural equations. Strong coupling between the flow solution and structural equations is maintained by fully converging the fully coupled system at each time step or stage. The time-integration schemes are constructed such that discrete conservation is maintained in the flow solution in the presence of dynamically deforming meshes, by verifying the discrete geometric conservation law. Efficient multigrid solution techniques are devised for solving both the governing flow equations and the mesh motion equations, using the same agglomerated coarse levels for both problems. The flutter boundary prediction of the AGARD 445.6 wing is used to demonstrate the accuracy and efficiency of these techniques as well as the sensitivity of flutter boundary prediction to time-step size and temporal errors.

## Nomenclature

$C$	=	damping matrix
$F(t)$	=	aerodynamic loading
$\mathbf{F}(U)$	=	convective fluxes
$\mathbf{G}(U)$	=	viscous fluxes
$K$	=	stiffness matrix
$M$	=	mass matrix
$\mathbf{n}$	=	interface normal
$q$	=	generalized coordinates
$\mathbf{R}$	=	discrete convective fluxes
$\mathfrak{R}$	=	nonlinear residual
$\mathbf{S}$	=	discrete viscous fluxes
$U$	=	vector of conserved quantities
$V$	=	volume
$w$	=	displacement
$\dot{\mathbf{x}}$	=	interface velocity
$\alpha$	=	coefficients of time-stepping schemes
$\partial\Omega(t)$	=	interface of control volume
$\xi$	=	modal damping
$\phi$	=	comparison functions
$\Omega(t)$	=	control volume
$\omega$	=	natural frequency

## I. Introduction

**A**EROELASTICITY involves the interaction between the fluid flow and the flexible structure of an aerospace vehicle. Aeroelastic problems include panel flutter, vertical tail buffet, fluid–blade interaction in rotorcraft, and active wings. Aeroelasticity plays an important role in the design of modern aircraft, because of the need to predict performance degradation, or even structural failure, resulting from strong fluid–structure interaction. Computational aeroelasticity, which combines computational fluid dynamics (CFD)

and computational structural dynamics (CSD), provides an efficient method for predicting and understanding aeroelastic phenomena.

Unstructured mesh approaches have become well established for steady-state flow simulations, due to the flexibility they afford for dealing with complex geometries. For unsteady flows with moving boundaries, such as aeroelastic problems, implicit time-integration strategies are required for the efficient solution of the flow equations; at the same time, robust mesh-deformation techniques are necessary for maintaining a suitable discretization of the evolving computational domain. To develop an efficient unsteady-flow simulation capability, both mesh deformation and flow solution aspects must be considered, as well as the interaction between these two phenomena. In previous work [1], we investigated the use of fast multigrid solvers for high-order implicit time integration of unsteady flows on static meshes. In [2], we extended our investigation to the simultaneous development of robust and efficient unstructured mesh-deformation techniques, as well as efficient higher-order time-integration strategies, and investigated the requirements for maintaining high time accuracy in the presence of deforming meshes.

For unsteady-flow simulations, computational time remains an important issue. When small temporal errors are desired, higher-order time integration (higher than second order) has been shown to be more efficient than low-order time integration. Bijl et. al. [3] investigated and compared higher-order implicit Runge–Kutta schemes and backward-difference schemes on structured grids, whereas Jothiprasad et al. [1] showed how a fourth-order Runge–Kutta scheme (IRK64) outperforms second-order backward-difference formulas (BDF2) on unstructured grids using a multigrid algorithm for solving the implicit system arising at each time step. In [2], a third-order backward-difference scheme (BDF3) was also shown to be competitive for time-dependent dynamic mesh problems. In this work, we investigate the application of BDF2, BDF3, and IRK64 schemes for computational aeroelasticity.

When dynamic meshes are used, the mesh velocities and other parameters related to geometry need to be considered carefully so that the errors introduced by the deformation of the mesh do not degrade the formal accuracy of the flow simulation. The discrete geometric conservation law (DGCL) provides a guideline on how to evaluate these quantities. First-order and second-order time-accurate and geometrically conservative schemes were presented and discussed in [4,5], respectively. Guillard and Farhat [6] have proved that to obtain at least first-order time accuracy, a DGCL condition must be satisfied. For higher-order time-integration schemes such as BDF3 and IRK64, a DGCL strategy that preserves the design order of the scheme must be explicitly constructed, as shown in [2,7–9].

Presented as Paper 441 at the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 9–12 January 2006; received 30 January 2006; revision received 11 August 2006; accepted for publication 12 August 2006. Copyright © 2006 by D. J. Mavriplis. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code \$10.00 in correspondence with the CCC.

\*Postdoctoral Research Associate, Department of Mechanical Engineering; zyang@uwyo.edu. Member AIAA.

†Professor, Department of Mechanical Engineering; mavriplis@uwyo.edu. Associate Fellow AIAA.

In addition to preserving time accuracy, efficient solution strategies must be employed to avoid excessive computational times for long time-integration problems. Nonlinear and linear multigrid methods have been investigated previously for steady and unsteady-flow simulations using unstructured meshes [10–12]. In [2,7,8] the use of a linear unstructured agglomeration multigrid method was shown to provide the most efficient strategy for integrating the nonlinear flow equations in time.

Several mesh-deformation strategies, such as the tension spring analogy [13,14], the torsion spring analogy [15,16], and the linear elasticity analogy [17,18], have been successfully demonstrated in the literature. In previous work [2], we investigated various methods for computing deformed-mesh configurations and found the linear elasticity approach to be the most robust approach, particularly when a variable modulus of elasticity is incorporated. However, for simple or smooth surface deformations, the basic spring analogy mesh motion technique is often sufficient to obtain smooth and valid deformed-mesh configurations. In both cases, the resulting mesh motion equations can be solved very efficiently using the same agglomeration multigrid approach used to solve the flow equations [2].

In the following sections, we first outline the governing equations and the base flow solver. We then discuss our choice and implementation of high-order time-integration schemes, including the fourth-order implicit Runge–Kutta scheme (IRK64), as well as the second and third-order backward-difference schemes (BDF2 and BDF3). This is followed by a presentation of the geometric conservation law (GCL), which must be verified by all of these schemes. The spring analogy mesh-deformation strategy is then briefly described. The computational structural dynamic equations are presented next, followed by a description of the algorithm employed for solving the fully coupled CFD and CSD problem at each time step or stage. Finally, a brief description of the agglomeration multigrid solver is given, as well as the application of this method to solve the fully coupled aeroelastic problem, including relative timings for the various components of the solver. These techniques are then used to compute the dynamic response and flutter boundary of the AGARD 445.6, in order to illustrate the performance of these methods and the sensitivity of flutter boundary prediction methods to numerical temporal errors.

## II. Flow Solver

### A. Governing Equations in Arbitrary Lagrangian–Eulerian Form and Base Flow Solver

The Navier–Stokes equations in conservative form can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot [\mathbf{F}(\mathbf{U}) + \mathbf{G}(\mathbf{U})] = 0 \quad (1)$$

Integrating over a (moving) control volume  $\Omega(t)$ , we obtain

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} [\mathbf{F}(\mathbf{U}) \cdot \mathbf{n}] dS + \int_{\partial\Omega(t)} [\mathbf{G}(\mathbf{U}) \cdot \mathbf{n}] dS = 0 \quad (2)$$

Using the differential identity

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV = \int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} \mathbf{U}(\dot{\mathbf{x}} \cdot \mathbf{n}) dS \quad (3)$$

Eq. (2) becomes

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV + \int_{\partial\Omega(t)} [\mathbf{F}(\mathbf{U}) - \mathbf{U}\dot{\mathbf{x}}] \cdot \mathbf{n} dS + \int_{\partial\Omega(t)} \mathbf{G}(\mathbf{U}) \cdot \mathbf{n} dS \\ = 0 \end{aligned} \quad (4)$$

Considering  $\mathbf{U}$  as cell-averaged quantities, these equations are discretized in space as

$$\frac{\partial}{\partial t} (V\mathbf{U}) + \mathbf{R}[\mathbf{U}, \dot{\mathbf{x}}(t), \mathbf{n}(t)] + \mathbf{S}[\mathbf{U}, \mathbf{n}(t)] = 0 \quad (5)$$

where

$$\mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}, \mathbf{n}) = \int_{\partial\Omega(t)} [\mathbf{F}(\mathbf{U}) - \dot{\mathbf{x}}\mathbf{U}] \cdot \mathbf{n} dS$$

represents the discrete convective fluxes in arbitrary Lagrangian–Eulerian form, and  $\mathbf{S}(\mathbf{U}, \mathbf{n})$  represents the discrete viscous fluxes. In the discrete form,  $\dot{\mathbf{x}}(t)$  and  $\mathbf{n}(t)$  now represent the time-varying velocities and surface normals of the control-volume boundary faces.

The Navier–Stokes equations are discretized by a central-difference finite volume scheme with additional matrix-based artificial dissipation on hybrid meshes that may include triangular and quadrilateral elements in two dimensions, or tetrahedra, pyramids, prisms, and hexahedra in three dimensions. Second-order accuracy is achieved using a two-pass construction of the artificial dissipation operator, which corresponds to an undivided biharmonic operator. A single unifying edge-based data structure is used in the flow solver for all types of elements. For multigrid calculations, a first-order-accurate discretization is employed for the convective terms on coarse grid levels [19,20].

### B. Higher-Order Time Integration and the Discrete Geometric Conservation Law

For unsteady-flow simulations, a fully implicit time-integration strategy is most often adopted, using either multistep backward-difference formulas (BDF) or multistage implicit Runge–Kutta (IRK) schemes. Although first-order (BDF1) and second-order (BDF2) backward-difference schemes are A-stable, higher-order multistep BDF schemes (beyond second order) are not A-stable. However, the unstable region of the BDF3 scheme is very small and this scheme has often been used successfully for unsteady-flow simulations [2]. Multistage implicit Runge–Kutta schemes of high order that are A-stable and L-stable can easily be constructed. However, multiple nonlinear problems need to be solved at each time step using IRK schemes, which makes these more expensive alternatives to the BDF schemes [3]. The application of higher-order IRK schemes for dynamic unstructured mesh problems has been investigated in [7–9]. In the current paper, our discussion will be limited to the two backward-difference schemes BDF2 and BDF3 and a six-stage, fourth-order-accurate IRK scheme (IRK64).

Equation (5) can be discretized in time, using the general formula for a  $k$ -step backward-difference scheme, as [1,3] (dropping the viscous flux terms for simplicity)

$$\alpha_1(V\mathbf{U})^{n+1} + \sum_{i=0}^{1-k} \alpha_i(V\mathbf{U})^{n+i} = \Delta t \mathbf{R}[(V\mathbf{U})^{n+1}, t^{n+1}] \quad (6)$$

For the BDF2 scheme,  $k = 2$  and the coefficients are given as  $\alpha_1 = \frac{3}{2}$ ,  $\alpha_0 = -2$ , and  $\alpha_{-1} = \frac{1}{2}$ , whereas for the BDF3 scheme,  $k = 3$  and the coefficients take on the values  $\alpha_1 = \frac{11}{6}$ ,  $\alpha_0 = -3$ ,  $\alpha_{-1} = \frac{3}{2}$ , and  $\alpha_{-2} = -\frac{1}{3}$ .

By defining a nonlinear residual [1]

$$\begin{aligned} \mathfrak{R}^{n+1}(\mathbf{U}) \equiv \mathfrak{R}(\mathbf{U}^{n+1}) \equiv \alpha_1(V\mathbf{U})^{n+1} - \sum_{i=0}^{k-1} \alpha_i(V\mathbf{U})^{n-i} \\ - \Delta t \mathbf{R}[(V\mathbf{U})^{n+1}, t^{n+1}] \end{aligned} \quad (7)$$

the solution of Eq. (6) can be obtained by solving the nonlinear problem  $\mathfrak{R}^{n+1}(\mathbf{U}) = 0$  at each time step.

A fourth-order time-accurate implicit Runge–Kutta scheme (IRK64) that obeys the discrete geometric conservation law is also employed in this paper. The general form for an  $s$ -stage implicit Runge–Kutta scheme applied to Eq. (5) is given as

$$(VU)^k = (VU)^n + \Delta t \sum_{j=1}^s a_{kj} \mathbf{R}[(VU)^j, t_n + c_j \Delta t] \quad (8)$$

$$k = 1, \dots, s$$

$$(VU)^{n+1} = (VU)^n + \Delta t \sum_{j=1}^s b_j \mathbf{R}[(VU)^j, t_n + c_j \Delta t] \quad (9)$$

where the  $n$  and  $n + 1$  superscripts on the dependent variable refer to the values at the beginning and end of the current time step, and the  $k$  and  $j$  superscripts refer to stage values computed by the Runge–Kutta scheme within the current time step. In this paper, we will concentrate on the explicit first-stage, single diagonal coefficient, diagonally implicit class of RK schemes (ESDIRK). The coefficients for the considered schemes can be represented by the Butcher tableau, in which the numerical values of the coefficients for the scheme used in this work are given in [1,3]. These schemes are characterized by a lower triangular form of the coefficient table, thus resulting in a single implicit solve at each individual stage. The

$$c_k = \sum_{j=1}^k a_{kj} \quad k = 1, \dots, s$$

coefficients denote the corresponding point in time for each individual stage. The first stage is explicit ( $a_{1j} = 0$ ), and the last stage coefficients are such that  $a_{kj} = b_j$  and  $j = 1, \dots, s$ , which omits the requirement of evaluating Eq. (9), because the value of  $(VU)^{n+1}$  at the new time step is given by the last stage value  $(VU)^{k=s}$ .

To apply the multistep BDF schemes or the multistage IRK scheme in the presence of dynamic unstructured meshes, the so-called geometric conservation law should be satisfied to avoid degrading the formal accuracy of the scheme. The GCL has also been shown to be a necessary condition for guaranteeing nonlinear stability [21], which makes GCL-compliant schemes important for aeroelasticity simulations that may involve long time integration. The original statement of the geometric conservation law was introduced by Thomas and Lombard [22] for structured meshes. The discrete geometric conservation law requires that the state  $\mathbf{U} = \text{const}$  be an exact solution of Eq. (5). In this case, we have  $S(\mathbf{U}, \mathbf{n}) = 0$ , because the viscous fluxes are based on gradients of  $\mathbf{U}$ . Additionally, we have

$$\int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \mathbf{U}\dot{\mathbf{x}}) \cdot \mathbf{n} \, dS = \mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}, \mathbf{n}) = -\mathbf{U}\bar{\mathbf{R}}(\dot{\mathbf{x}}, \mathbf{n}) \quad (10)$$

because the integral of the convective fluxes  $\mathbf{F}(\mathbf{U})$  around a closed control volume must be zero for constant  $\mathbf{U}$ , for any spatially conservative scheme, with  $\bar{\mathbf{R}}$  referring to the discretization of the second term in the preceding boundary integral. Thus, the GCL can be stated in semidiscrete form as

$$\frac{\partial V}{\partial t} - \bar{\mathbf{R}}[\dot{\mathbf{x}}(t), \mathbf{n}(t)] = 0 \quad (11)$$

The geometric conservation law must be satisfied in the discrete form, or so-called discrete geometric conservation law. Lesoinne and Farhat [4] presented a time-accurate BDF1 that obeys the discrete geometric conservative law, whereas Koobus and Farhat [23] derived a time-accurate BDF2 that obeys the discrete conservation law. To construct the DGCL for the BDF3 scheme, Eq. (6) is rewritten as

$$\sum_{i=1}^{-2} \alpha_i (VU)^{n+i} = \Delta t \mathbf{R}[(VU)^{n+1}, t^{n+1}, \dot{\mathbf{x}}(t), \mathbf{n}(t)] \quad (12)$$

where  $\dot{\mathbf{x}}(t)$  and  $\mathbf{n}(t)$  represent the velocity and the normal vector of a control-volume boundary face, which are evaluated in such a manner that the preceding equation is satisfied exactly in the presence of uniform flow, as determined by the DGCL. A full derivation of the DGCL for BDF3 can be found in [2].

**Table 1 Butcher tableau for the ESDIRK class of RK schemes with number of stages;  $s = 6$**

$c_1 = 0$	0	0	0	0	0	0
$c_2$	$a_{21}$	$a_{66}$	0	0	0	0
$c_3$	$a_{31}$	$a_{32}$	$a_{66}$	0	0	0
$c_4$	$a_{41}$	$a_{42}$	$a_{43}$	$a_{66}$	0	0
$c_5$	$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$	$a_{66}$	0
$c_6 = 1$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$a_{66}$
$\mathbf{w}^{n+1}$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$a_{66}$

Similarly to the derivation for BDF schemes, a nonlinear residual for each stage of the IRK scheme is defined as

$$\mathfrak{R}^{n+1}(\mathbf{U}) \equiv \mathfrak{R}(\mathbf{U}^{n+1}) \equiv (VU)^k - (VU)^n \Delta t \sum_{j=1}^k \alpha_{kj} \mathbf{R}(\mathbf{U}^k, \mathbf{x}^k, \dot{\mathbf{x}}^k) \quad (13)$$

$$k = 1, \dots, s$$

with the corresponding values  $\mathbf{x}^k = \mathbf{x}(t + c_k \Delta t)$  and  $\dot{\mathbf{x}}^k = \dot{\mathbf{x}}(t + c_k \Delta t) \cdot \mathbf{n}(t + c_k \Delta t)$ , with the values of  $c_k$  being given by the RK scheme. To satisfy the DGCL at each stage, the  $\dot{\mathbf{x}}^k$  can be computed by solving the system:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{x}}^1 \\ \dot{\mathbf{x}}^2 \\ \dot{\mathbf{x}}^3 \\ \dot{\mathbf{x}}^4 \end{pmatrix} = \frac{1}{\Delta t} \begin{pmatrix} \Delta t \dot{\mathbf{x}}^{(1)} \\ V^2 - V^n \\ V^3 - V^n \\ V^4 - V^n \end{pmatrix} \quad (14)$$

where the explicit first stage  $\alpha_{11} = 0$  (cf. Table 1) corresponds to equating  $\dot{\mathbf{x}}^1$  with the value determined at the end of the previous time step. Because the IRK64 scheme is an SDIRK scheme (single coefficient diagonally implicit RK), the matrix of  $\alpha_{kj}$  coefficients is of lower triangular form, and Eq. (14) is easily solved by forward substitution. A full derivation of the DGCL for IRK schemes can be found in [7–9].

At each time step or RK stage, Eqs. (7) or (13) are solved using the agglomeration multigrid technique described subsequently.

### III. Mesh-Moving Strategies

The mesh-deformation strategy used in this work is based on the spring analogy approach, which has been used extensively in the past for aeroelastic calculations (see, for example, [13]). Each edge of the mesh is represented by a spring in which the stiffness is related to the length of the edge. The governing equations are closely related to a simple Laplace equation, as the displacements in each coordinate direction become decoupled and are governed by the equations:

$$(\Delta x_i)_m = \frac{\sum_j k_{ij} [(x_j)^{n+1}_m - (x_j)^n_m]}{\sum_j k_{ij}} \quad (15)$$

where  $m = 1, 2, 3$ ,

$$k_{ij} = \frac{1}{l_{ij}^p}$$

and

$$l_{ij} = \left( \sum_{m=1}^3 [(x_i)_m - (x_j)_m]^2 \right)^{\frac{1}{2}}$$

The parameter  $p$  usually is set to two, as recommended in [14]. Although these equations are relatively simple to solve, this approach tends to produce deformed meshes with collapsed or negative cell volumes, especially for large deformations in complex geometries. However, for the types of deformations considered in this work, the spring analogy was found to be sufficient for

generating smooth and valid meshes at all time steps. In previous work [2], we have advocated the use of linear elasticity methods for more demanding mesh motion cases and plan to incorporate these techniques into the aeroelasticity framework in the near future.

#### IV. Structure Equation

The governing equation for the structure of a typical wing under aerodynamic loading can be written as

$$[M]\{\ddot{w}\} + [C]\{\dot{w}\} + [K]\{w\} = \{F(t)\} \quad (16)$$

where  $[M]$ ,  $[C]$ , and  $[K]$  represent the mass matrix, damping matrix, and stiffness matrix, respectively;  $\{F(t)\}$  is the aerodynamic loading; and  $w$  is the displacement. Using modal analysis, the dependent variables are expanded in terms of the natural free vibration modes as

$$\{w(x, t)\} = \sum_{i=1}^N \mathbf{q}_i(t) \{\phi_i(x)\} \quad (17)$$

where  $\mathbf{q}_i(t)$  and  $\{\phi_i(x)\}$  are generalized coordinates and comparison functions satisfying the free vibration modes, respectively. The comparison functions are obtained by solving the eigenproblem of the free vibration problem, using only the first  $N$  modes of the expansion. Substituting Eq. (17) into Eq. (16) and then multiplying by  $\phi^T$ , we obtain [24,25]

$$\ddot{\mathbf{q}} + [\xi]\dot{\mathbf{q}} + [\omega]\mathbf{q} = \mathbf{Q} \quad (18)$$

where

$$[\xi] = \phi^T[C]\phi, \quad [\omega] = \phi^T[K]\phi, \quad \mathbf{Q} = \phi^T\{F\}$$

Because  $[\omega]$  and  $[\xi]$  are diagonal matrices and their diagonal terms are  $\omega_i$  and  $2\xi_i\omega_i$ , respectively, the coupled system of Eq. (18) can be rewritten as a set of uncoupled scalar equations:

$$\ddot{q}_i + 2\xi_i\omega_i\dot{q}_i + \omega_i^2 q_i = Q_i \quad 1 = 1, 2, \dots, N \quad (19)$$

where  $\xi_i$  and  $\omega_i$  are the modal damping parameter and the natural frequency for the  $i$ th mode, respectively. To convert Eq. (19) to a first-order system, let

$$y_{1i} = q_i, \quad y_{2i} = \dot{q}_i$$

then, Eq. (19) can be rewritten as

$$\dot{\mathbf{Y}}_i = \mathbf{A}_i \mathbf{Y}_i + \mathbf{B}_i \quad (20)$$

where

$$\mathbf{Y}_i = \begin{Bmatrix} y_{1i} \\ y_{2i} \end{Bmatrix}, \quad \mathbf{A}_i = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\xi_i\omega_i \end{bmatrix}, \quad \mathbf{B}_i = \begin{Bmatrix} 0 \\ Q_i \end{Bmatrix} \quad (21)$$

The implicit time-integration schemes (BDF or IRK) that are used to solve the flow equations are used to solve Eq. (20), as well to obtain  $q$  and  $w$ . The general form for a backward-difference scheme for Eq. (20) can be written as

$$\alpha_1 Y_i^{n+1} + \sum_{i=0}^{1-k} \alpha_i Y_i^{n+i} = \Delta t \left( \mathbf{A}_i Y_i^{n+1} + \mathbf{B}_i^{n+1} \right) \quad (22)$$

and implicit Runge–Kutta schemes for Eq. (20) can be written as

$$Y_i^k = Y_i^n + \Delta t \sum_{j=1}^s a_{kj} \left[ \mathbf{A}_i Y_i^j + \mathbf{B}_i(Y_i^j, t_n + c_j \Delta t) \right] \quad k = 1, \dots, s \quad (23)$$

$$Y_i^{n+1} = Y_i^n + \Delta t \sum_{j=1}^s b_j \left[ \mathbf{A}_i Y_i^j + \mathbf{B}_i(Y_i^j, t_n + c_j \Delta t) \right] \quad (24)$$

#### V. Coupling Algorithm for CFD and CSD, Method of Solution

Computational aeroelasticity is a coupled fluid–structure problem that requires the exchange of forces and displacements between the flow and the structure, as well as the simultaneous solution of the CFD and CSD equations. In the general case, the boundary surfaces of the mesh used in the flow computation are different from those used in the structure computation. Many researchers have presented interface algorithms that ensure energy-conserving force transfers and smooth displacement transfers between CFD and CSD surface meshes. Sadeghi et al [26] provided a good comparison of the infinite-plate spline interpolation (IPS) [27] method, the constant-volume tetrahedron method, and the inverse boundary element method. The IPS method is based on the equilibrium equation of an infinite thin plate. According to Harder and Desmarais [27], the displacement of a plate surface can be written as

$$w(x, y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^N F_i r_i^2 \ln(r_i^2) \quad (25)$$

where  $r_i^2 = (x - x_i)^2 + (y - y_i)^2$  and the coefficients  $a_0$ ,  $a_1$ ,  $a_2$ , and  $F_i$  are obtained by solving

$$\sum_{i=1}^N F_i = 0, \quad \sum_{i=1}^N x_i F_i = 0, \quad \sum_{i=1}^N y_i F_i = 0 \quad (26)$$

$$w_i = a_0 + a_1 x_i + a_2 y_i + \sum_{j=1}^N F_j r_{ij}^2 \ln(r_{ij}^2)$$

where  $r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$ , which corresponds to matching the displacements on the CSD and CFD meshes at the  $i = 1, \dots, N$  prescribed points. The advantage of IPS is that the connections between the  $N$  prescribed points are not required.

A conservative force and energy transfer method has also been given by Farhat et al. [28] and is illustrated in Fig. 1a. The vertices of the CFD mesh are first projected to the corresponding enclosing element of the structural mesh. The natural coordinates  $\xi$  and  $\eta$  of this projection point in each enclosing element are computed, and the displacements originally defined at the vertices of the CSD mesh are transferred to the CFD mesh vertices as

$$w_{aj} = \sum_{i=1}^{i_e} N_i(\xi_j, \eta_j) w_{si} \quad (27)$$

where  $w_{aj}$  and  $w_{si}$  are displacements of the CFD and CSD mesh vertices, respectively, and  $N_i(\xi_j, \eta_j)$  are the CSD mesh element shape functions.

It should be noted that the infinite-plate spline interpolation (IPS) is a two-dimensional method. However, the test case used in this work is the AGARD 445.6 wing, which is a thin wing. The interface of the CFD mesh and the CSD mesh can be approximated by a plane [26]. Both methods are used in this paper, and no effect on the final results could be observed between the two methods.

At each physical time step or each stage of the IRK scheme, the combined system of fluid and structural equations must be solved in a fully coupled manner. This requires the use of a strongly coupled implicit method [24,29] and is necessary to eliminate any time lag between the flow solution and the structure solution at each physical time step. Although loosely coupled solution schemes can be used for aeroelastic simulations, our goal is to study the temporal accuracy of the proposed schemes, and we use a fully coupled approach to eliminate any other possible sources of error that may contaminate our accuracy study. Figure 1b depicts the strategy for simultaneously

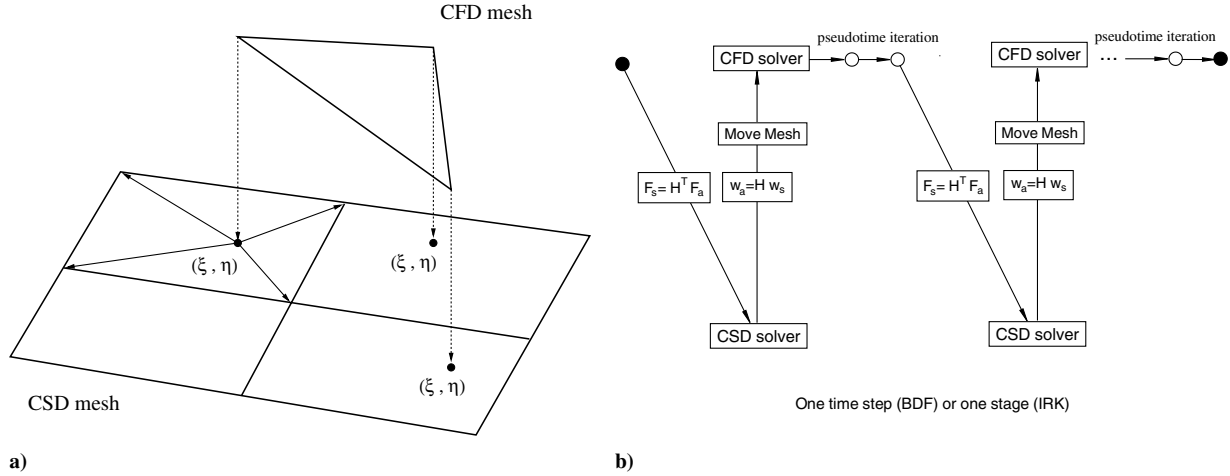


Fig. 1 Illustrations of algorithms for a) mesh displacement transfer [28] and b) strong flow-structural equation coupling.

converging the fully coupled system. At each physical time step of a BDF scheme or at each stage of the IRK scheme, the following steps are taken:

- 1) Calculate the aerodynamic loading and transfer the loading to the structure mesh.
- 2) Solve the structural equations [cf. Eq. (22) or Eqs. (23) and (24)].
- 3) Transfer displacements to the interface of the CFD mesh.
- 4) Solve the mesh motion [Eq. (15)].
- 5) Solve the flow equations in pseudotime.
- 6) If appreciable flowfield changes occur, repeat steps 1–5 until fully converged.

The entire procedure operates in a parallel computing environment, except for step 2, in which the modal structural solver is executed on the master node of the parallel computer, because this step requires negligible computational time compared with the other tasks.

The transfer of displacements from the CSD mesh to the CFD mesh in step 3 proceeds as determined by Eqs. (25) or (27) and may be written as

$$w_a = H w_s \quad (28)$$

where  $w_a$  is the interface displacement of the CFD mesh,  $w_s$  is the interface displacement of the CSD mesh, and  $H$  is the transfer operator, which is determined by Eqs. (25) or (27).

The transfer of the aerodynamic loading in step 1 can then be written as

$$F_s = H^T F_a \quad (29)$$

where  $F_s$  represents the CSD interface forces,  $F_a$  represents the CFD

interface forces, and the force transfer operator  $H^T$  is taken as the transpose of the operator used for displacement transfers. This approach has been shown to result in a transfer mechanism that is energy-conserving [28].

In the fully coupled solution algorithm, overall convergence depends on the number of outer iterations, as well as the number of inner iterations for the flow solver and mesh motion equations in pseudotime. The inner pseudotime solution is performed using the agglomeration multigrid algorithm described in the next section.

## VI. Acceleration Strategies and Computational Cost

Agglomeration multigrid methods are used to accelerate the solution of the nonlinear problem arising at each time step of the unsteady-flow equations, as well as for solving the mesh motion equations. Agglomeration multigrid was originally developed as a steady-state solver for the Euler and Navier–Stokes equations on unstructured grids [10–12,20]. The idea of multigrid is to accelerate the solution on a fine grid by iteratively computing corrections to the fine grid problem on coarser grid levels for which the cost of the iterations is lower and the global error components are more easily reduced. Figures 2a and 2b show an example of the agglomeration multigrid procedure. Figure 2a shows the fine grid, whereas Fig. 2b shows the second-level coarse grid, on which each coarse cell is formed as a combination of several fine cells.

For highly stretched meshes, the effectiveness of the multigrid approach degrades due to the anisotropic stiffness induced by the grid stretching. To relax this stiffness, an implicit line solution technique was introduced in [30,31]. The lines are constructed along the strong coupling direction in the mesh, as shown by the example depicted in Fig. 2c for an unstructured mesh on a NACA0012 airfoil. In these regions, a block tridiagonal algorithm is used to solve all

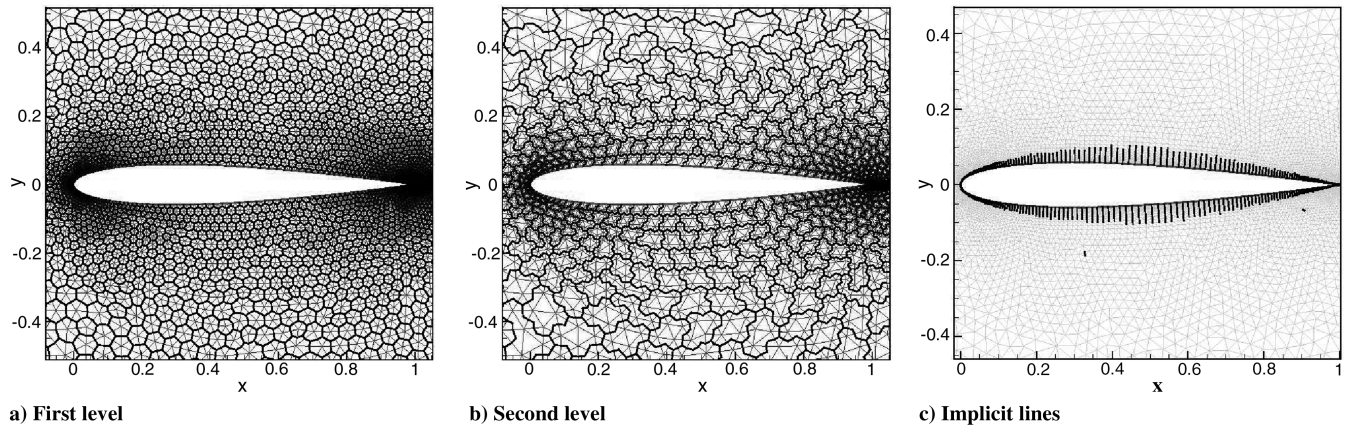


Fig. 2 Illustration of agglomeration multigrid and implicit line construction.

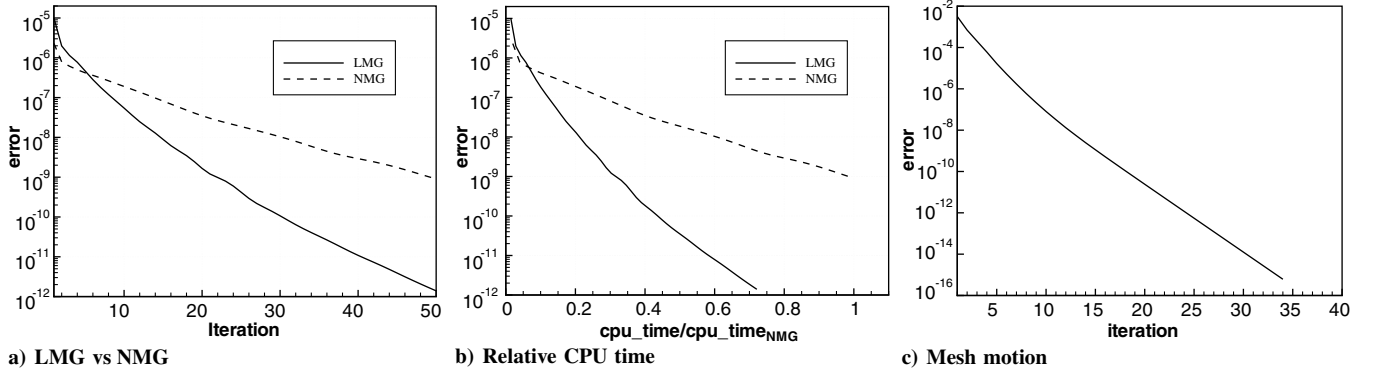


Fig. 3 Convergence history of mesh motion; LMG and NMG.

quantities along each line implicitly, thus replacing the simple explicit approach on all grid levels. The same line structures and coarse agglomerated levels are used for solving both the mesh motion equations and the implicit time-dependent flow equations. However, for the nonlinear flow equations, either a nonlinear multigrid (NMG) approach may be employed or a linear multigrid method (LMG) may be used to drive an outer approximate Newton iteration. Figure 3 illustrates the convergence history obtained at a particular BDF2 time step of the AGARD 445.6 wing test case described next, using the nonlinear and linear multigrid approaches, respectively, for a Mach number of  $M = 1.141$ , both in terms of nonlinear iterations and in terms of CPU time. For the linear multigrid case, three linear multigrid cycles are used for each outer nonlinear Newton update, whereas a single nonlinear multigrid cycle constitutes a nonlinear iteration in the case of the NMG solver. This difference accounts for the faster convergence of the LMG solver in terms of iterations. The linear multigrid approach is also found to be substantially more economical in terms of CPU time than the nonlinear multigrid solver, requiring about one third of the overall CPU time of the nonlinear multigrid method to achieve the same level of convergence. This is largely due to the lower CPU time required by the linear multigrid cycles, as documented in [2]. Similar results are observed for the convergence of the implicit system of flow equations that occur at each stage of the IRK scheme. Therefore, the linear multigrid method is used exclusively in the computations described in the remainder of this paper. Figure 3c shows the convergence history of the mesh motion solver for the same test case at the same time step. These equations can be converged to machine precision in less than 40 multigrid cycles.

For aeroelastic computations, full convergence of the flow equations and mesh motion equations at each time step is not required. In fact, the optimal level of convergence in pseudotime for dual-iteration strategy implicit time-integration schemes remains an area of research. For the computational aeroelastic problem, the important convergence criterion is based on the fully coupled system of flow and structural equations. In the present work, the fully coupled system is partially converged by using three outer CFD/CSD coupling iterations, as depicted in Fig. 1b, with each coupling iteration performing 10 nonlinear iterations on the flow equations, whereas the mesh motion equations are converged to  $10^{-10}$  at each coupling iteration, requiring a variable number (but less than 20) of multigrid cycles. This strategy results in a minimum of two orders of magnitude reduction in the fully coupled system residuals.

Table 2 shows the relative CPU time of the flow solver and the mesh motion solver for BDF2 and IRK64 within a single coupled CFD/CSD time step. All values are normalized by the CPU time of

the BDF2 flow solver. The CPU time of the mesh motion solver varies from 24 to 36% of the CPU time of the corresponding flow solver. The CPU time of IRK64 for one time step is about 4.7 times that of the CPU time of BDF2, because the IRK64 scheme requires the solution of five implicit systems (one for each stage, except for the first stage). Note that the mesh motion equations are close to fully converged at each coupling iteration to avoid any possible errors from these terms, and the time required by the mesh motion solver could likely be reduced without any adverse effect on the overall solution by easing this convergence tolerance. Additionally, the CPU time of the CSD solver is negligible, because the modal analysis used to solve the CSD equations requires a trivial amount of CPU time.

## VII. Results and Discussion

### A. Time-Accuracy Validation for Dynamic Unstructured Mesh Problem

The time accuracy of the BDF2 and IRK64 implicit time-integration schemes is examined for a simplified test case involving prescribed motion. A three-dimensional inviscid flow test case is constructed by forced twisting of an ONERA M6 wing, at a Mach number of 0.755 and an incidence of 0.016 deg. A periodic pitching motion is prescribed at the quarter-chord point of the wing tip, with a reduced frequency of 0.1628 and an amplitude of 2.51 deg. The wing root is held fixed and a linear variation of the pitching motion is prescribed between the wing tip and root, thus resulting in a twisting motion. A fully tetrahedral mesh (shown in Fig. 4a) is used for this case, containing 53,961 vertices. Mesh deformation is computed at each time step or stage by solving the spring analogy equations subject to a fixed outer-boundary condition. The computed lift coefficient vs time is shown in Fig. 4b for the BDF2 and the IRK64 scheme, using 8, 16, 32, or 64 time steps per period. Good temporal accuracy for the IRK64 scheme is observed using as few as eight time steps per period. Figure 4c illustrates the computed temporal error as a function of the time step for both schemes, at the time  $t = 54$ , using a highly resolved IRK64 solution with a time step of 0.3375 (128 time steps per period) as the reference solution, in which the error is computed as the RMS difference of all flow variables between the computed solution and the reference solution at the final time. The slope of the error curve achieved for the BDF scheme is 2, which matches the design accuracy of the scheme, whereas the IRK64 scheme results in an error curve slope of 3.3, which is slightly lower than the design value of 4, for this fourth-order-accurate scheme. Because we are uniquely concerned with the quantification of temporal error in this study, we omit any comparison with experimental data, which necessarily brings in effects due to spatial and modeling (inviscid flow) errors.

### B. Flutter Boundary Prediction for AGARD 445.6 Wing

The AGARD 445.6 wing test case [32], which was tested in the transonic dynamics tunnel at NASA Langley Research Center, has been widely used as a benchmark for evaluating aeroelastic simulations [29,33,34]. The wing is a semispan model that has a

Table 2 Relative CPU time for flow solver and mesh motion solver

	Flow	Mesh	Total
BDF2	1.00	0.358	1.358
IRK64	5.16	1.243	6.400

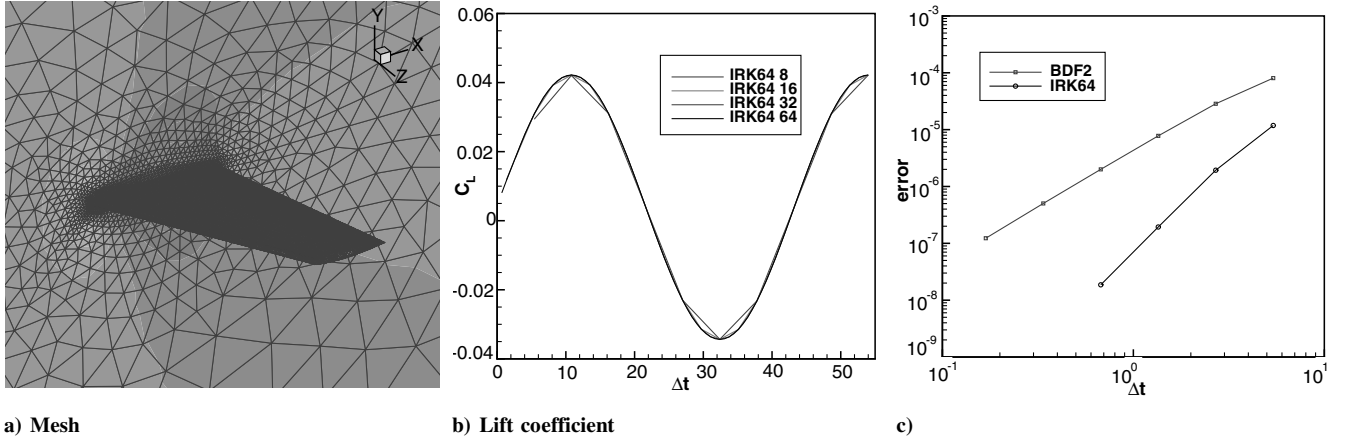


Fig. 4 Flow solution for BDF2 and IRK64 schemes for a three-dimensional twisting M6 wing.

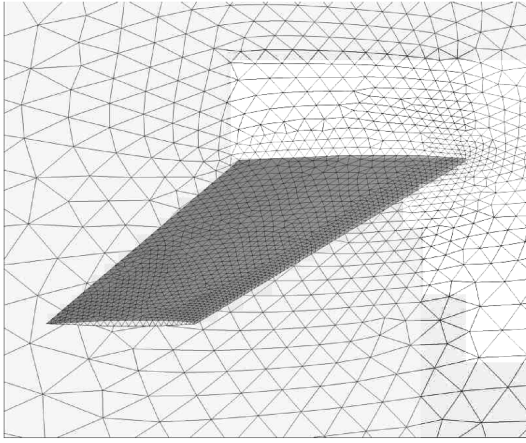


Fig. 5 Meshes for AGARD wing 445.6.

quarter-chord sweep angle of 45\_deg, a panel aspect ratio of 1.65, a taper ratio of 0.66, and a NACA 65A004 airfoil cross section.

The wing flutter speed index is defined as

$$V_f = \left( \frac{U_\infty}{b\omega_\alpha\sqrt{\mu}} \right)_{\text{neutral}} \quad (30)$$

where  $U_\infty$  is the velocity of the freestream,  $b$  is the half chord at the wing root,  $\omega_\alpha$  is the natural circular frequency of the first uncoupled torsion mode,  $\mu = m/(\rho v)$  is the wing mass ratio,  $m$  is the mass of the wing,  $v$  is the volume related to the wing, and  $\rho$  is the density of the fluid. To find a flutter point at one freestream Mach number  $M_\infty$ , the Mach number and corresponding freestream velocity  $U_\infty$  are kept fixed. The fluid density  $\rho$  and corresponding dynamic pressure  $\rho U_\infty^2/2$  are adjusted and the wing transients are calculated. If the transients of the generalized coordinates do not grow or decay, a neutral point or a flutter boundary point is obtained. To find the flutter boundary, the structure responses are analyzed using the modal identification technique of Bennett and Desmarais [35]. A nonlinear least-squares curve fit method in the form of Eq. (31) is used to fit the response curves:

$$q_i(t) = a_0 + \sum_{j=1}^n e^{\sigma_j t} [a_j \cos(\omega_j t) + b_j \sin(\omega_j t)] \quad i = 1, 2, 3 \quad (31)$$

where  $\sigma_j$  and  $\omega_j$  are the damping and frequency, respectively.

The IRK64 and BDF3 schemes are used to investigate the effect of higher-order time integration on the wing flutter boundary predictions. BDF2 is also used as a baseline reference scheme. The CFD mesh used in this paper is shown in Fig. 5. The mesh consists of 40,460 nodes and 224,531 tetrahedral elements. Inviscid flow simulation is performed on this mesh, and the first four modes are used in the modal analysis of the structural computations. These modes correspond to the first four modes of the weakened model 3 [32], which are identified as the first bending, first torsional, second bending, and second torsional modes, as shown in Fig. 6.

To demonstrate the difference between the low-order scheme and the higher-order schemes, the time history of the wing displacements computed using BDF2, BDF3, and IRK64 at different Mach numbers are examined in detail, next.

Figure 7 depicts the first, second, third, and fourth modal dynamic responses at a speed index of 0.4459 and a Mach number of  $M = 0.499$  for BDF2, BDF3, and IRK64. The time step varies from  $\Delta t = 0.0375$  to  $\Delta t = 1.8$ , with the largest time step corresponding to approximately  $\frac{1}{3}$  of the second mode period. The IRK scheme uses the largest time steps ( $\Delta t = 1.8$  and  $\Delta t = 0.6$ ), whereas the lower-order schemes use a range of smaller time steps. All four mode displacements show a decaying response. The details of the response shown in Figs. 7b, 7d, 7f, and 7h clearly show the BDF2 and BDF3 results converging to the IRK64 results as the time-step size is decreased. For a given time-step size, the BDF3 results are closer to the IRK64 results than those of the BDF2 scheme, except for the fourth mode at  $\Delta t = 0.6$ . The growing response for BDF3 at  $\Delta t = 0.6$  [Fig. 7g] shows the BDF3 scheme is unstable at this largest time step. Note that the response of the higher modes is more sensitive to the time-step size for the lower-order BDF schemes, whereas the fourth-order IRK scheme displays relatively little sensitivity to the time-step size, even for the higher modes.

Figures 8a and 8b provide a comparison of the temporal error vs the time-step size and the relative CPU time for the various time-integration schemes. The error is computed based on the first mode

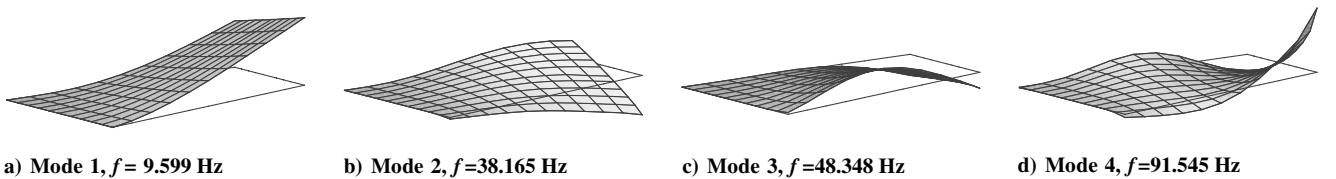
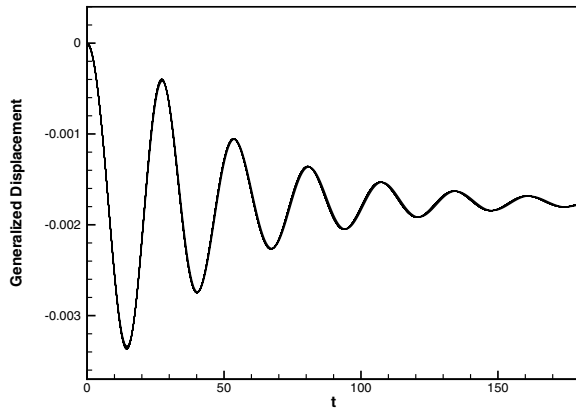
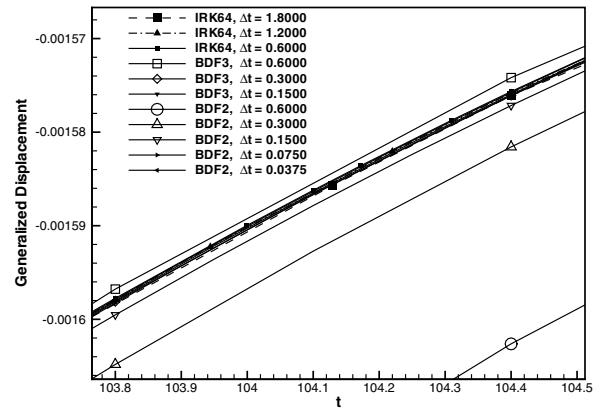


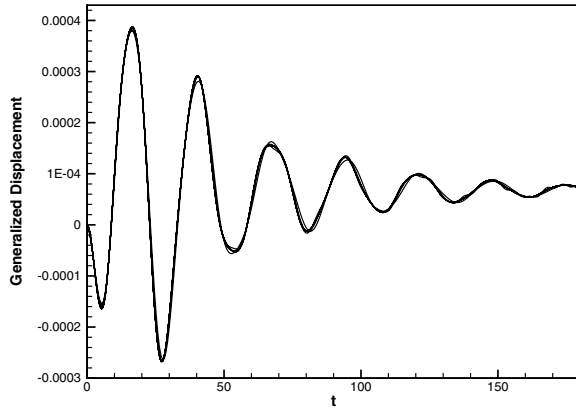
Fig. 6 Modal deflection for AGARD wing 445.6; weakened model 3.



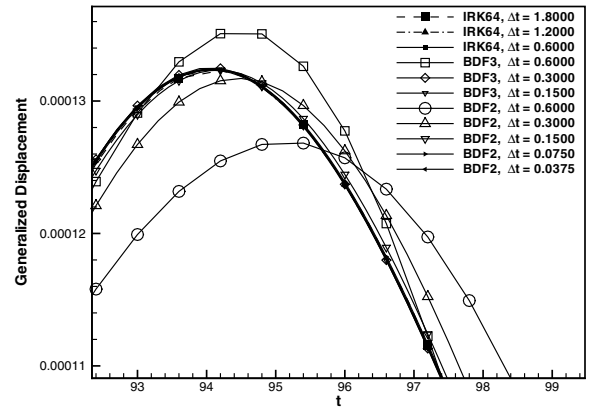
a) First



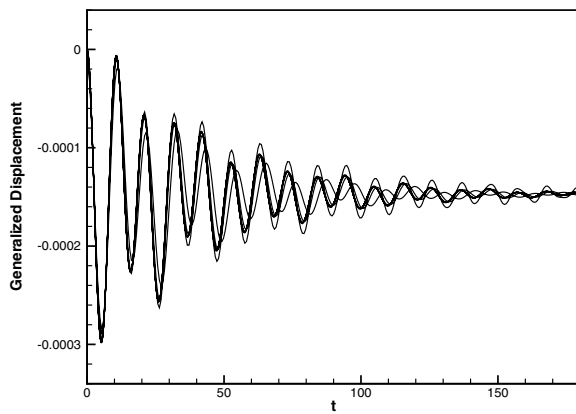
b) Zoom in (first)



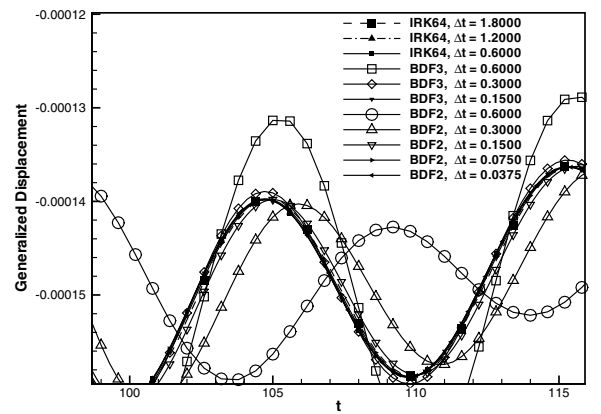
c) Second



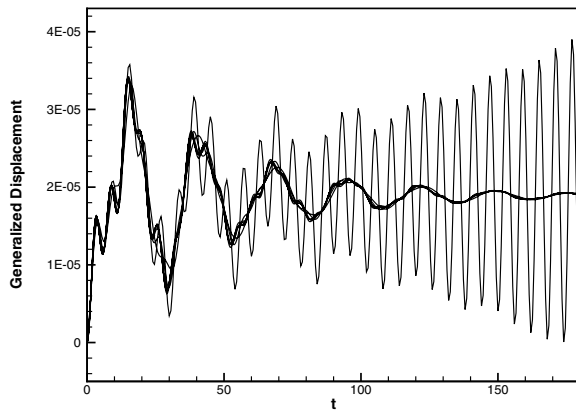
d) Zoom in (second)



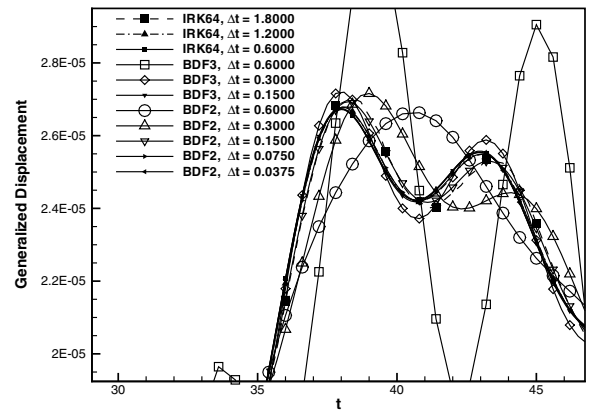
e) Third



f) Zoom in (third)



g) Fourth



h) Zoom in (fourth)

Fig. 7 Time history of the first four modes;  $M = 0.499$ ; speed index of 0.4459; stable response.



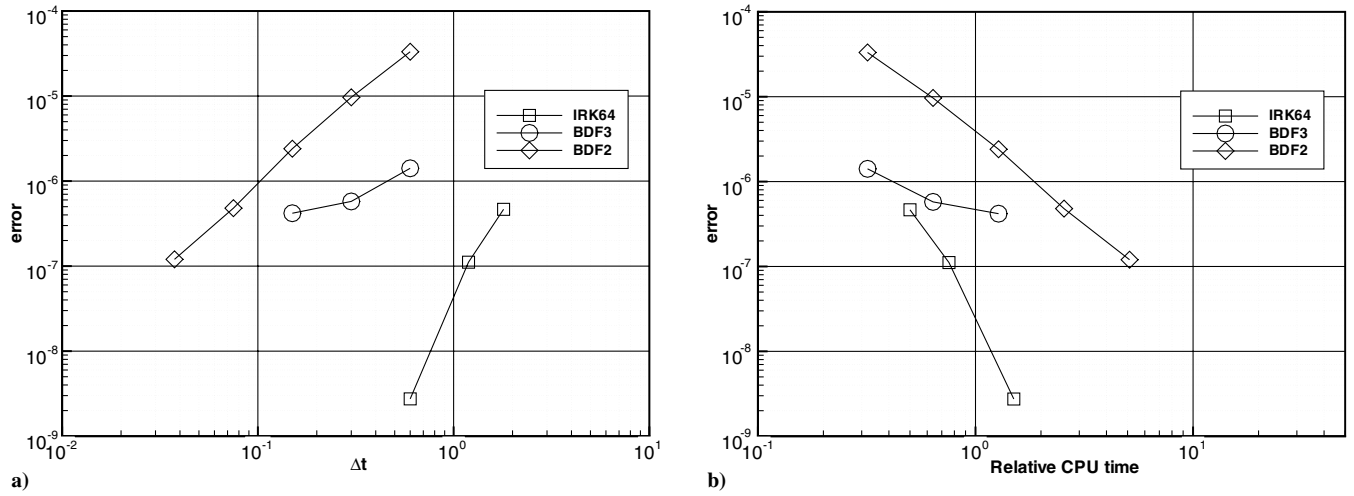


Fig. 8 Error vs  $\Delta t$  and CPU time;  $M = 0.499$ ; speed index of 0.4459; stable response.

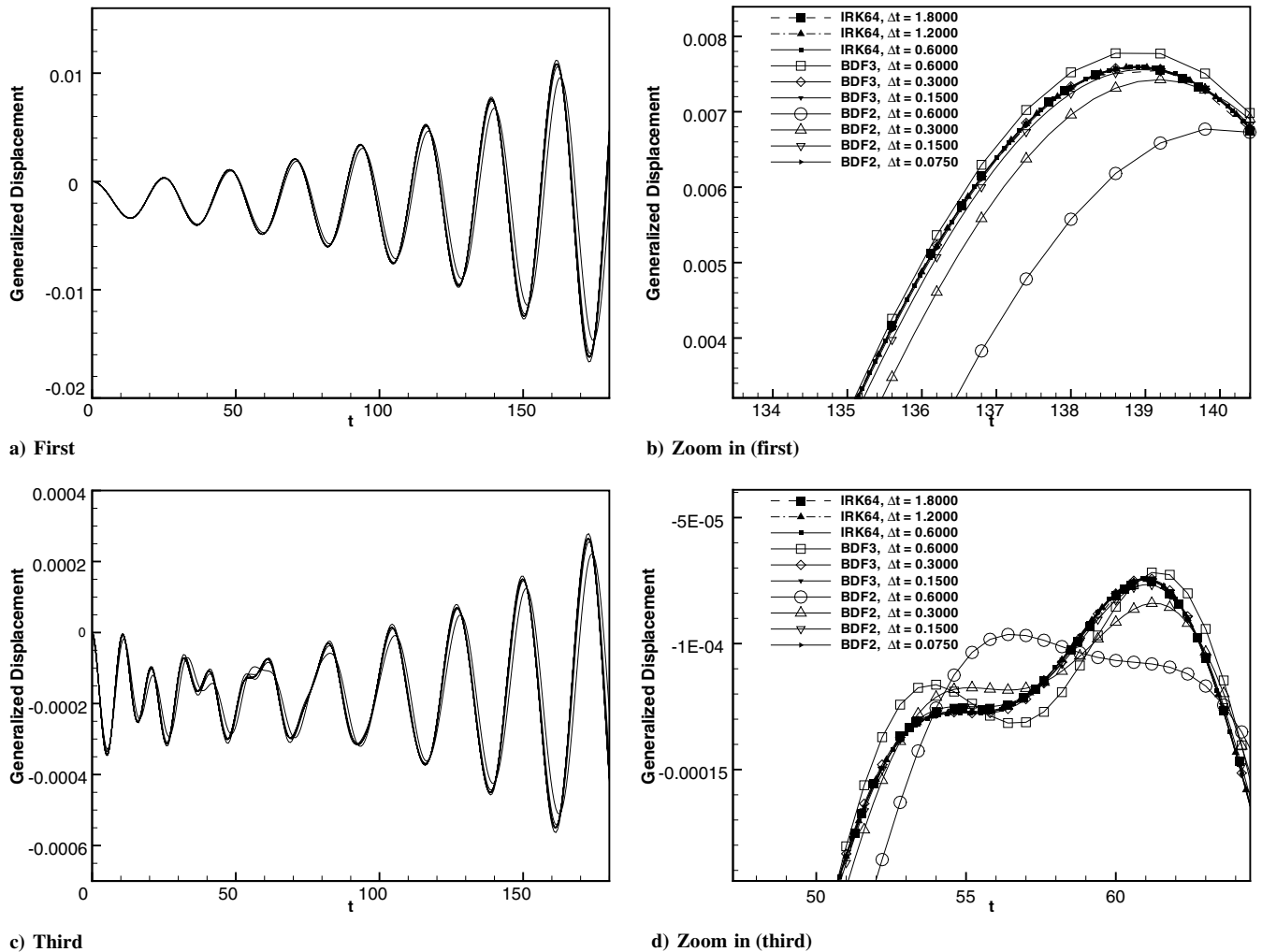


Fig. 9 Time history of the first and the third mode;  $M = 0.499$ ; speed index of 0.49; unstable response.

response, using results computed with a smaller time step ( $\Delta t = 0.3$ ) on the IRK64 scheme as the reference "exact" solution. The BDF2 scheme delivers second-order accuracy based on the slope of the error curve, whereas BDF3 does not recover its design accuracy, although it delivers higher absolute accuracy at any given time step compared with the lower-order scheme. The IRK64 scheme is more than third-order accurate, but falls somewhat short of the fourth-order design accuracy. Nevertheless, this scheme delivers the highest

absolute accuracy for any given time step. Figure 8b shows IRK64 to be up to five times faster in terms of CPU time compared with the BDF2 scheme for small error tolerances of the order of  $5 \times 10^{-7}$ .

Figure 9 shows the first and the third modal dynamic responses at a speed index of 0.49 and a Mach number of  $M = 0.499$  for BDF2, BDF3, and IRK64. The time steps vary from  $\Delta t = 0.075$  to  $\Delta t = 1.8$ . The first mode is the dominating mode and the first mode

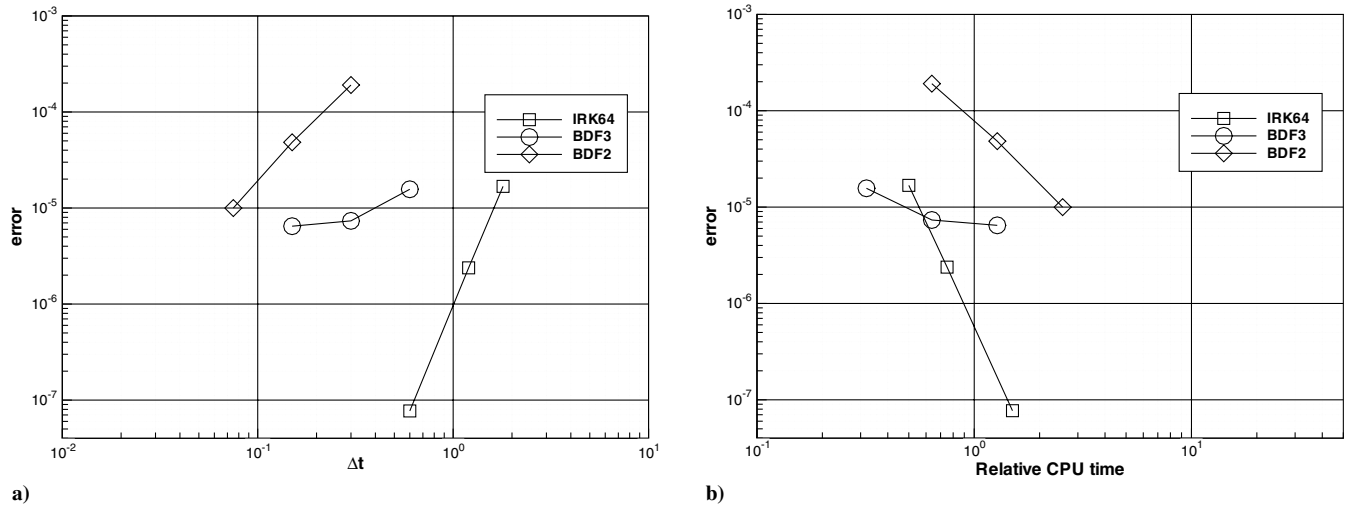


Fig. 10 Error vs  $\Delta t$  and CPU time;  $M = 0.499$ ; speed index of 0.49; unstable response.

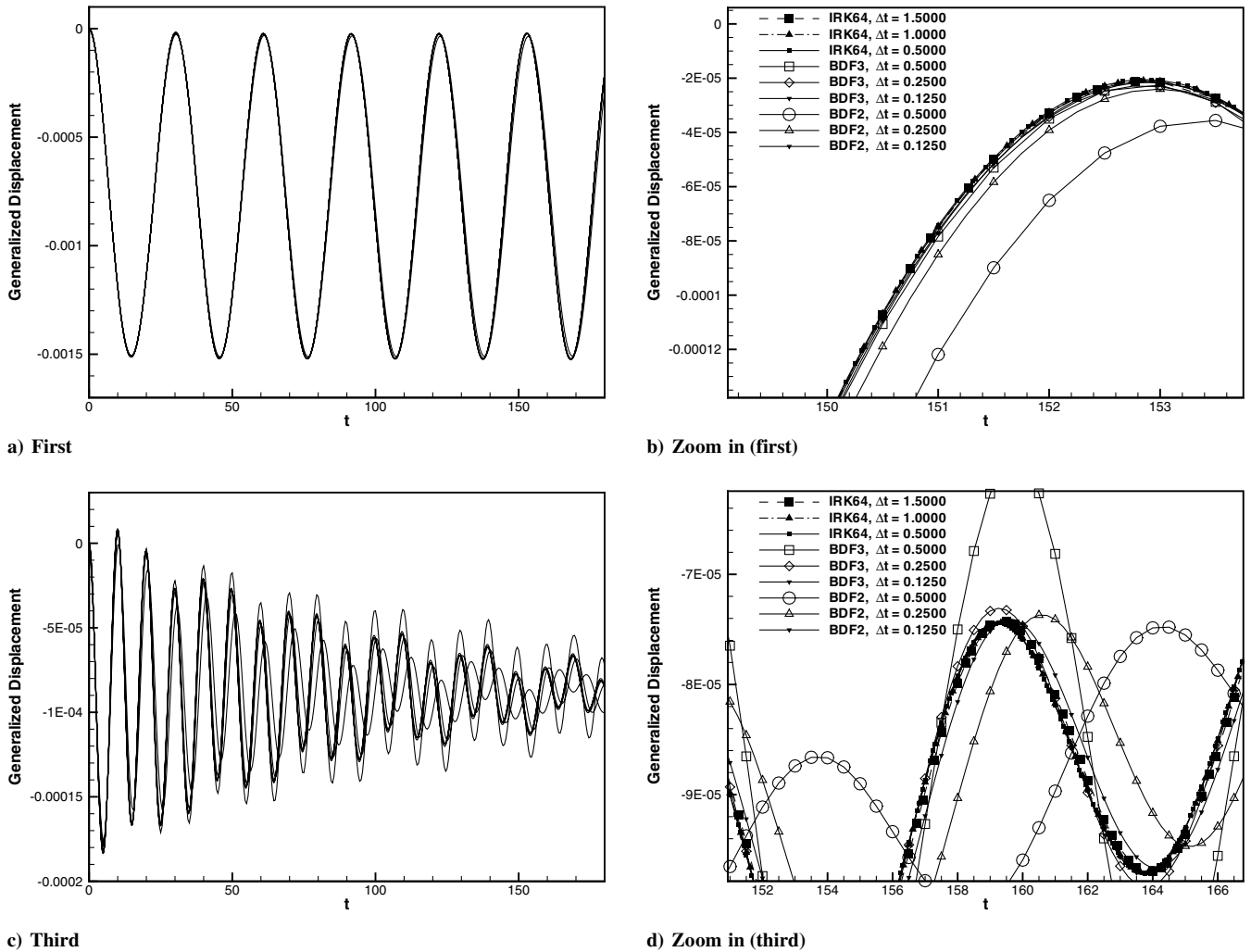


Fig. 11 Time history of the first and the third mode;  $M = 0.901$ ; speed index of 0.37; neutral response.

response is growing from the beginning. The oscillating amplitude of the third mode decreases before  $t = 50$  and then increases, which indicates that energy is transferred from other modes. The growing of both the first and the third mode displacements shows an unstable response. The details in the zoomed Figs. 9b and 9d show that the BDF2 and BDF3 schemes converge to the IRK64 results as the time step is decreased. At the same  $\Delta t$ , the results of the BDF3 scheme are

closer to the IRK64 results than those of BDF2. The error curves of Figs. 10a and 10b, obtained in the same manner as those described previously, show second-order accuracy for the BDF2 scheme and better than third-order accuracy for the IRK64 scheme, whereas the BDF3 scheme is seen to lag its design accuracy. Again, the IRK64 scheme is seen to be up to five times faster than the BDF2 scheme for small error tolerances.

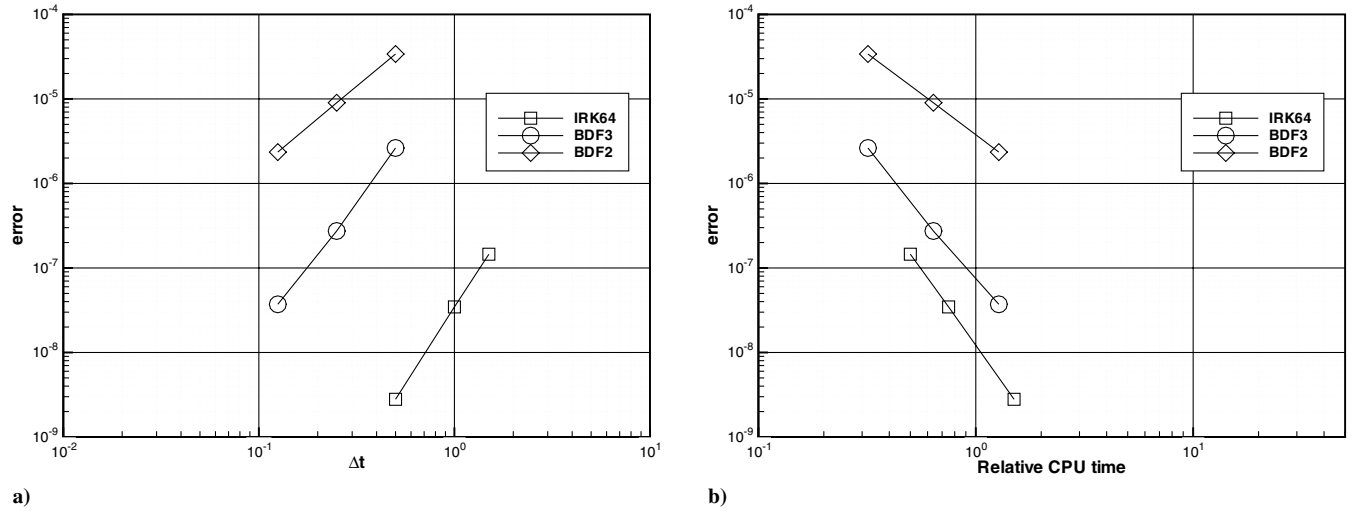


Fig. 12 Error vs  $\Delta t$  and CPU time;  $M = 0.901$ ; speed index of 0.37; neutral response.

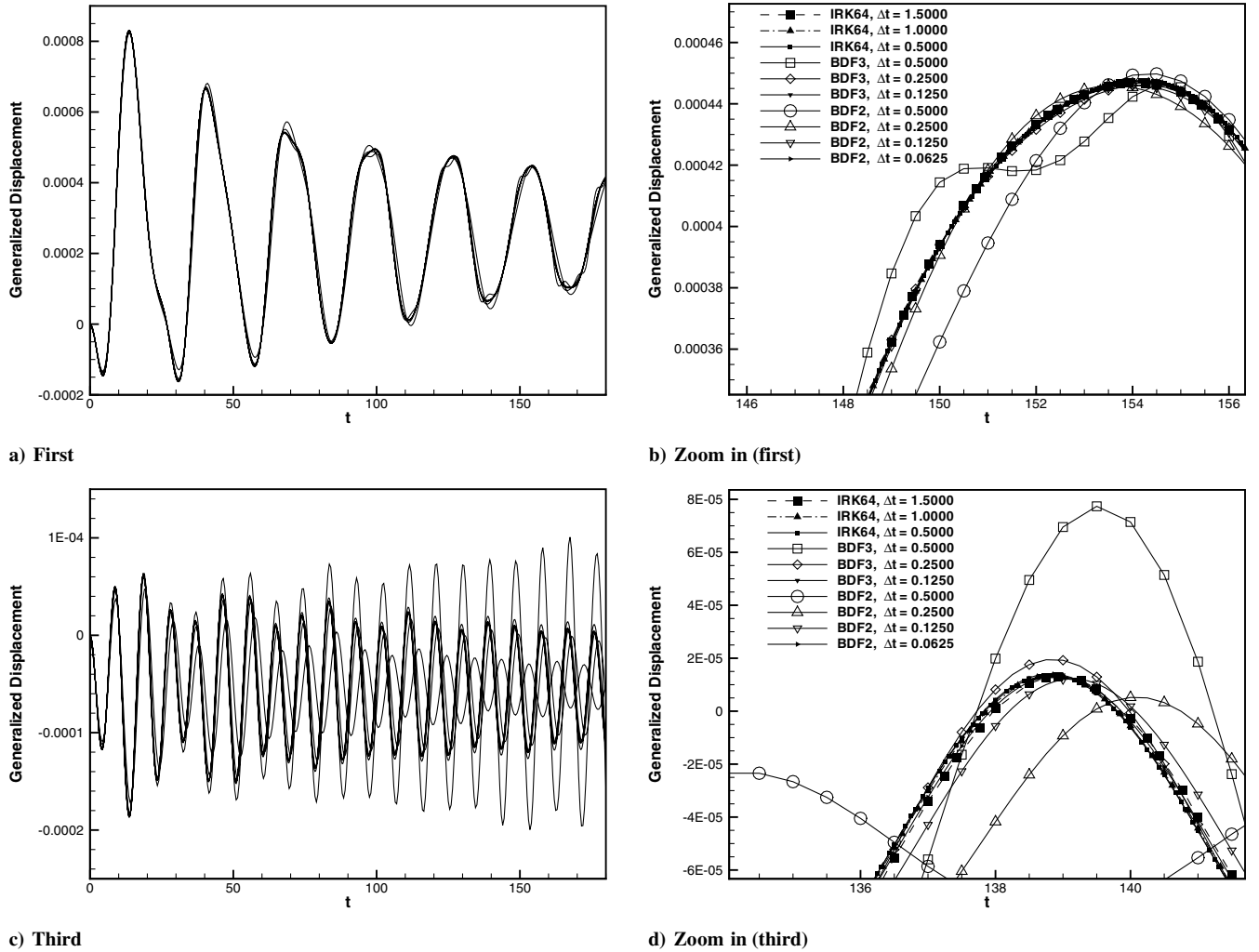


Fig. 13 Time history of the first mode;  $M = 1.141$ ; speed index of 0.442; stable response.

Figure 11 shows the first and the third modal dynamic response at a speed index of 0.37 and a Mach number of  $M = 0.901$  for BDF2, BDF3, and IRK64. The time step varies from  $\Delta t = 0.125$  to  $\Delta t = 1.5$ , which is about  $\frac{1}{8}$  of the second mode period. The first mode shows a neutral response. As shown previously, the BDF2 and BDF3 results converge to the IRK64 results as the time step is reduced, and the IRK64 results remain the most efficient, particularly for high temporal accuracy cases, as seen from Fig. 12b.

Figure 13 shows the first and the third modal dynamic response at a speed index of 0.442 and a Mach number of  $M = 1.141$  for BDF2, BDF3, and IRK64. The time step varies from  $\Delta t = 0.03125$  to  $\Delta t = 1.5$ , which is about  $\frac{1}{8}$  of the second mode period. Both the first and the third mode responses are decaying. The blowup in Figs. 13b and 13d show that the BDF2 and BDF3 schemes converge to the IRK64 results as the time step decreases, but Fig. 13c shows the BDF3 scheme going unstable for larger time-step values. The

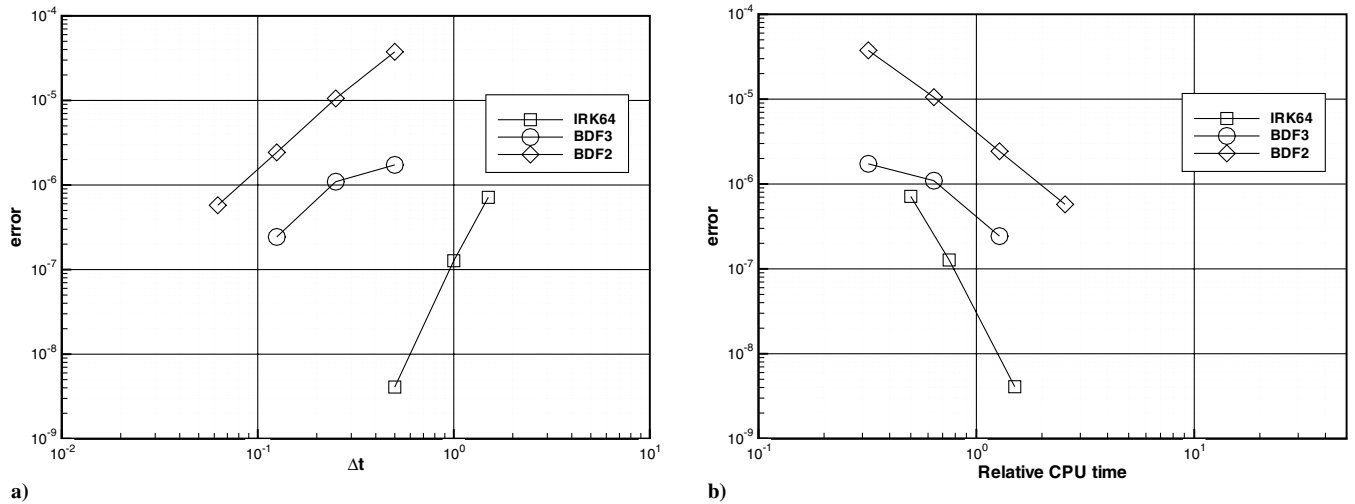


Fig. 14 Error vs  $\Delta t$  and CPU time;  $M = 1.141$ ; speed index of 0.442; stable response.

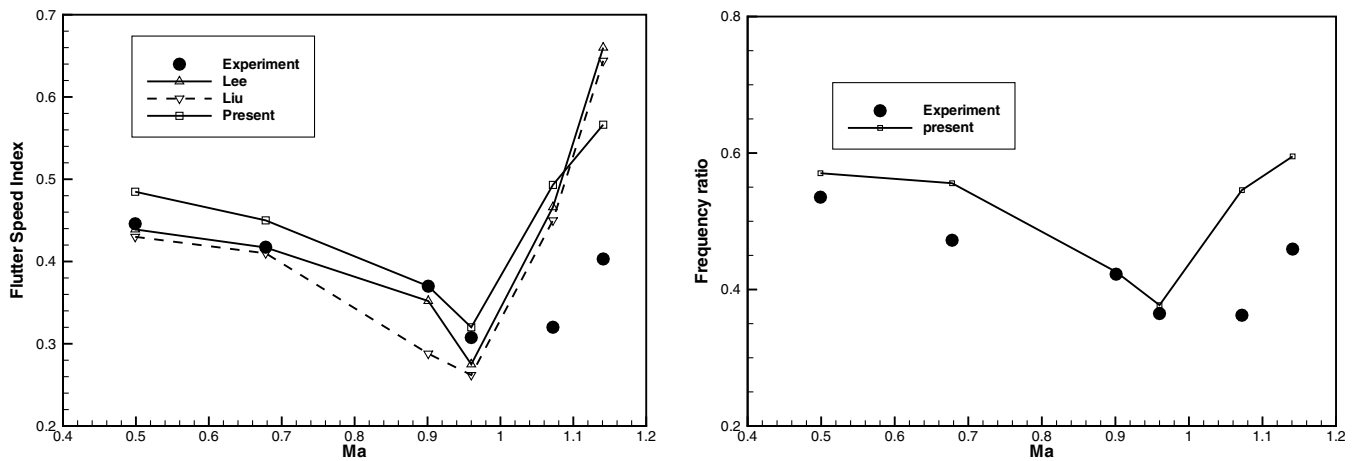


Fig. 15 Comparison of the flutter prediction with experimental data.

superior efficiency of the IRK64 scheme is again documented in the accuracy curves of Fig. 14.

Figure 15 shows the comparison of the flutter speed index and frequency ratio  $\omega/\omega_a$  by the BDF and IRK schemes with other researchers' computational results, as well as the corresponding experimental values. Fairly good agreement with experimental data is obtained in the subsonic region for the IRK and BDF schemes. In most cases, the results of these schemes are indistinguishable in predicting the flutter boundary, using the smallest time step for each scheme. However, the IRK64 scheme has the potential for significantly reducing the computational time required to predict the flutter boundary for a given accuracy level through the use of larger time steps. For the supersonic region, all methods show substantial underestimation of the flutter boundary, which is consistent with previously reported computational results [29,34]. The mesh used in this study is relatively coarse and the resulting spatial error, as well as the use of an inviscid flow model, may be responsible for these discrepancies. However, the inviscid coarse mesh simulations are well suited for studying the temporal error of the various time-integration schemes in detail, because the effects of temporal and spatial error are treated separately in our approach.

## VIII. Conclusions

The principal elements required for the efficient solution of time-dependent aeroelasticity simulations have been developed and validated. These elements include the formulation of the DGCL of high-order BDF and IRK time-integration schemes and the development of a robust mesh motion solver and a fast linear

multigrid solver for the flow equations. These have been applied to a standard aeroelastic test case in order to validate their accuracy and efficiency. The time histories of the wing deflection computed using the BDF2, BDF3, and IRK64 schemes have been compared and the IRK64 scheme shows superior overall efficiency. The responses of the higher modes are most sensitive to the temporal error and, thus, to time-step sizes and the order of the time-integration scheme. Although the BDF3 scheme offers high temporal accuracy at modest computational cost, the tendency of this scheme to go unstable for large time-step sizes may limit its use, particularly when higher structural modes are included. Although one may suspect the absence of unconditional stability as the cause of the BDF3 instabilities, the precise cause of this behavior remains to be determined. For cases in which many higher modes are considered, the use of high-order time-integration schemes such as IRK64 yields stable and accurate results and may prove to be essential. Future work will include larger test cases involving more structural modes and viscous flow simulations.

## Acknowledgment

This work was supported in part by grant NNL04AA63G from the NASA Langley Research center.

## References

- [1] Jothiprasad, G., Mavriplis, D. J., and Caughey, D. A., "Higher-Order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Meshes," *Journal of Computational Physics*, Vol. 191, No. 2, 2003, pp. 542–566.

- [2] Yang, Z., and Mavriplis, D. J., "Unstructured Dynamic Meshes with Higher-Order Time Integration Schemes for the Unsteady Navier-Stokes Equations," AIAA Paper 2005-1222, Jan. 2005.
- [3] Bijl, H., Carpenter, M. H., Vatsa, V. N., and Kennedy, C. A., "Implicit Time Integration Schemes for the Unsteady Incompressible Navier-Stokes Equations: Laminar Flow," *Journal of Computational Physics*, Vol. 179, No. 1, 2002, pp. 313-329.
- [4] Lesoinne, M., and Farhat, C., "Geometric Conservation Laws for Flow Problems with Moving Boundaries and Deformable Meshes, and Their Impact on Aeroelastic Computations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 134, Nos. 1-2, 1996, pp. 71-90.
- [5] Koobus, B., and Farhat, C., "Second-Order Time-Accurate and Geometrically Conservative Implicit Schemes for Flow Computations on Unstructured Dynamic Meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 170, No. 1-2, 1999, pp. 103-129.
- [6] Guillard, H., and Farhat, C., "On the Significance of the Geometric Conservation Law for Flow Computations on Moving Meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, Nos. 11-12, 2000, pp. 1467-1482.
- [7] Mavriplis, D. J., and Yang, Z., "Achieving Higher-Order Time Accuracy for Dynamic Unstructured Mesh Fluid Flow Simulations: Role of the GCL," AIAA Paper 2005-5114, June 2005.
- [8] Mavriplis, D. J. and Yang, Z., "Construction of the Discrete Geometric Conservation Law for High-Order Time Accurate Simulations on Dynamic Meshes," *Journal of Computational Physics*, Vol. 213, No. 2, 2006, pp. 557-573.
- [9] Van Zuijlen, A., and Bijl, H., "High Order Time Integration For Fluid-Structure Interaction on Moving Meshes," AIAA Paper 2005-5247, June 2005.
- [10] Mavriplis, D. J., "Multigrid Techniques for Unstructured Meshes," Lecture Series, von Kármán Institute for Fluid Dynamics No. 1995-02, 1995.
- [11] Mavriplis, D. J., "On Convergence Acceleration Techniques for Unstructured Meshes," AIAA Paper 1998-2966, June 1998.
- [12] Mavriplis, D. J., "An Assessment of Linear Versus Non-Linear Multigrid Methods for Unstructured Mesh Solvers," *Journal of Computational Physics*, Vol. 175, No. 1, 2002, pp. 302-325.
- [13] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381-1288.
- [14] Venkatakrishnan, V., and Mavriplis, D. J., "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 127, No. 2, 1996, pp. 380-397.
- [15] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M., "Torsional Spring for Two-Dimensional Dynamic Unstructured Fluid Meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 163, No. 1-4, 1998, pp. 231-245.
- [16] Murayama, M., Nakahashi, K., and Matsushima, K., "Unstructured Dynamic Mesh for Large Movement and Deformation," AIAA Paper 2002-0122, Jan. 2002.
- [17] Baker, T., and Cavallo, P. A., "Dynamic Adaptation for Deforming Tetrahedral Meshes," AIAA Paper 1999-3253, June 1999.
- [18] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155-1163.
- [19] Mavriplis, D. J., and Venkatakrishnan, V., "A Unified Multigrid Solver for the Navier-Stokes Equations on Mixed Element Meshes," NASA CR-198183, July 1995.
- [20] Mavriplis, D. J., and Pirzadeh, S., "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *Journal of Aircraft*, Vol. 36, No. 6, Dec. 1999, pp. 987-998.
- [21] Farhat, C., Geuzaine, P., and Crandmont, C., "The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids," *Journal of Computational Physics*, Vol. 174, No. 2, 2001, pp. 669-694.
- [22] Thomas, P. D., and Lombard, C. K., "Geometric Conservation Law and its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030-1037.
- [23] Koobus, B., and Farhat, C., "On the Implicit Time Integration of Semi-Discrete Viscous Fluxes on Unstructured Dynamic Meshes," *International Journal for Numerical Methods in Fluids*, Vol. 29, No. 8, 1999, pp. 975-996.
- [24] Alonso, J. J., and Jameson, A., "Fully-Implicit Time-Marching Aeroelastic Solutions," AIAA Paper 1994-0056, Jan. 1994.
- [25] Kamakoti, R., Shyy, W., Thakur, S., and Sankar, B., "Time Dependent RANS Computation for an Aeroelastic Wing," AIAA Paper 2004-0886, Jan. 2004.
- [26] Sadeghi, M., Liu, F., Lai, K., and Tsai, H., "Application of Three-Dimensional Interface for Data Transfer in Aeroelastic Computations," AIAA Paper 2004-5376, Aug. 2004.
- [27] Harder, R. L., and Desmarais, R. N., "Interpolation Using Surface Splines," *Journal of Aircraft*, Vol. 9, No. 2, 1972, pp. 189-191.
- [28] Farhat, C., Lesoinne, M., and LeTallec, P., "Load and Motion Transfer Algorithms for Fluid/Structure Interaction Problems with Non-Matching Discrete Interfaces: Momentum and Energy Conservation, Optimal Discretization and Application to Aeroelasticity," *Computer Methods in Applied Mechanics and Engineering*, Vol. 157, No. 1-2, 1998, pp. 95-114.
- [29] Liu, F., Cai, J., Zhu, Y., Tsai, H., and Wong, A., "Calculation of Wing Flutter by a Coupled Fluid-Structure Method," *Journal of Aircraft*, Vol. 38, No. 2, 2001, pp. 334-342.
- [30] Mavriplis, D. J., "Directional Agglomeration Multigrid Techniques for High-Reynolds Number Viscous Flow," *AIAA Journal*, Vol. 27, No. 10, 1999, pp. 1222-1230.
- [31] Mavriplis, D. J., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," *Journal of Computational Physics*, Vol. 145, No. 1, Sept. 1998, pp. 141-165.
- [32] Yates, E. C. J., "AGARD Standard Aeroelastic Configurations for Dynamic Response, 1: Wing 445.6," NASA TM 100492, 1987.
- [33] Bennett, R. M., Batina, J. T., and Cunningham, H. J., "Wing-Flutter Calculations with the CAP-TSD Unsteady Transonic Small-Disturbance Program," *Journal of Aircraft*, Vol. 26, No. 9, 1989, pp. 876-882.
- [34] Lee-Rausch, E. M., and Batina, J. T., "Calculation of AGARD Wing 445.6 Flutter Using Navier-Stokes Aerodynamics," AIAA Paper 1993-3476, 1993.
- [35] Bennett, R. M., and Desmarais, R. N., "Curve Fitting of Aeroelastic Transient Response Data with Exponential Functions," NASA SP 415, May 1975.

E. Livne  
Associate Editor