applied sciences

MDPI

Article

An Effective Arbitrary Lagrangian-Eulerian-Lattice Boltzmann Flux Solver Integrated with the Mode Superposition Method for Flutter Prediction

Tianchi Gong [1,2], Feng Wang [3] and Yan Wang [1,2,*]

[1] State Key Laboratory of Mechanics and Control for Aerospace Structures, Nanjing University of Aeronautics and Astronautics and Astronautics, Yudao Street 29, Nanjing 210016, China

[2] College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Yudao Street 29, Nanjing 210016, China

[3] Key Lab of High Speed and High Reynolds, Aerodynamic Research Institute of AVIC, Shenyang 110034, China

[*] Correspondence: aerowangy@nuaa.edu.cn

Abstract: An arbitrary Lagrangian-Eulerian lattice Boltzmann flux solver (ALE-LBFS) coupled with the mode superposition method is proposed in this work and applied to study two- and three-dimensional flutter phenomenon on dynamic unstructured meshes. The ALE-LBFS is applied to predict the flow field by using the vertex-centered finite volume method with an implicit dual time-stepping method. The convective fluxes are evaluated by using lattice Boltzmann solutions of the non-free D1Q4 lattice model and the viscous fluxes are obtained directly. Additional fluxes due to mesh motion are calculated directly by using local conservative variables and mesh velocity. The mode superposition method is used to solve for the dynamic response of solid structures. The exchange of aerodynamic forces and structural motions is achieved through interpolation with the radial basis function. The flow solver and the structural solver are tightly coupled so that the restriction on the physical time step can be removed. In addition, geometric conservation law (GCL) is also applied to guarantee conservation laws. The proposed method is tested through a series of simulations about moving boundaries and fluid–structure interaction problems in 2D and 3D. The present results show good consistency against the experiments and numerical simulations obtained from the literature. It is also shown that the proposed method not only can effectively predict the flutter boundaries in both 2D and 3D cases but can also accurately capture the transonic dip phenomenon. The tight coupling of the ALE-LBFS and the mode superposition method presents an effective and powerful tool for flutter prediction and can be applied to many essential aeronautical problems.

Keywords: flutter prediction; ALE-LBFS; mode superposition method; limit cycle oscillation; transonic dip

check for updates

Citation: Gong, T.; Wang, F.; Wang, Y. An Effective Arbitrary Lagrangian-Eulerian-Lattice Boltzmann Flux Solver Integrated with the Mode Superposition Method for Flutter Prediction. *Appl. Sci.* **2024**, *14*, 3939. https://doi.org/10.3390/app14093939

Academic Editor: Maria Grazia De Giorgi

Received: 16 April 2024
Revised: 30 April 2024
Accepted: 30 April 2024
Published: 5 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In aeronautical applications, aerodynamic flutter represents a formidable danger to aircraft, characterized by its ability to enlarge the small amplitude oscillations swiftly and result in catastrophic consequences. This phenomenon originates from the intricate interplay of aerodynamic, inertial, and elastic forces, and it typically occurs in the transonic flow regime, resulting in the structure engaging in undamped oscillations with constant or increasing amplitude. Both numerical simulations [1,2] and experimental studies [3,4] have been carried out to study the transonic flutter phenomenon and found that the flutter boundary often shifts with the Mach number. Due to the development of high-performance computing techniques, a variety of efficient effective computational fluid dynamics/computational solid dynamics (CFD/CSD) solvers [5–9] have been developed.

Among the various CFD/CSD methods, the arbitrary Lagrangian-Eulerian method (ALE) [10–12] distinguishes itself by maintaining mesh fitted to the solid structure all the time during the structural deformation, a vital element for accurately obtaining the aerodynamic forces from the surrounding flow field with enhanced resolution near the boundaries. ALE empowers mesh nodes near moving or deforming boundaries to evolve with the structure, while those farther away remain fixed in the Eulerian description. ALE furnishes physical meshes with arbitrary velocities independent of fluid motion, eliminating the need to remap Lagrangian solutions to adjusted meshes. This not only enhances accuracy but also significantly streamlines mesh computations, reducing computational overhead. In terms of solutions on the moving mesh, both Euler equations and Navier–Stokes (NS) equations with suitable turbulence models have worked as fluid solvers. Consequently, the numerical schemes for the discretization of the convective terms should be adapted to the dynamic meshes by considering the local velocity of the grids, which may be not trivial considering the requirement of low dissipation and high stability for transonic and supersonic flows intensive shock waves. Lee-Rausch and Batina [13,14] have numerically discovered the transonic dip phenomenon by coupling unsteady Euler and/or Navier–Stokes equations with structural mode equations, separately. Comparing these two solvers, they further present the important role of viscosity in flutter prediction. Yuan et al. [15] apply the Roe flux difference splitting scheme to discretize the Euler/NS equations and tightly couple them with structural dynamic equations. By the finite element method, the aircraft wing deforms numerically, governed by the 3D elasticity equations. They succeeded in predicting the aeroelastic instability of the rigid airfoil and elastic AGARD 445.6 wing. For wide engineering applications, the ALE method is adapted and involves new techniques together with various other numerical methodologies for the prediction of flutter, including frequency-domain techniques [16], deep learning [17], and model order reduction [18–20]. Although some improvements are achieved, most of them still depend on solving the NS equations by evaluating the fluxes in a macroscopic way.

In fact, with the rising tendency of complex flow environments and novel materials for aircraft manufacturing, it is feasible and superior to obtain the macroscopic equations of fluid mechanics from the mesoscopic point of view. In the merit of simple implementation and parallelism, the lattice Boltzmann method (LBM) [21–23] turns out to be a good choice in the domains of multiphase flow [24], microscale flow [25], thermal flow [26], and turbulence [27]. The LBM is established on the kinetic theory and describes the flow as discrete particle distribution functions associated with a set of discrete velocities statistically. The particles propagate over time and tend to relax toward a local equilibrium state. The density distribution of discrete particles evolves on simple rules of the collision and the subsequent streaming. Its simplicity and efficiency make it an attractive alternative to traditional CFD methods [28–31]. Cheylan et al. [29] couple the LBM with the immersed boundary method for moving objects. The shear layer and near-wall effects are both predicted in the high Reynolds number flow. Meldi et al. [22] apply the ALE approach to the LBM to simulate the flow–structure interactions. Validated by a moving square cylinder in a Poiseuille flow, it is confirmed that the ALE-LBM framework succeeds in capturing the flow phenomena. For robustness, Bhadauria et al. [19] introduce a two-population lattice Boltzmann formulation to describe the complex flows. Under the ALE scheme, they accurately capture the dynamic behavior of multiple cylinders and simulate transonic flutter over an airfoil. Coincidentally, Saadat et al. [21] investigate the unsteady flows over forced pitching airfoils in transonic regions and even at high speeds. The moving boundary is described by the ALE technique.

Stemming from lattice Boltzmann equations, a lattice Boltzmann flux solver (LBFS) [32–34] has been proposed by Shu and his colleagues recently. The LBFS abandons the restriction on the uniform meshes and endows itself with a flexible alternative for problems with complex geometries, no matter structured or unstructured meshes [35–38]. Moreover, within the framework of LBFS, the boundary and initial conditions are treated in a manner identical to finite-volume NS solvers. Thanks to these aforementioned advantages, the applications of lattice Boltzmann flux solver with various lattice Boltzmann models have been accessible for multiphase flow [39], thermal flow [40], and compressible flow [41,42]. The LBFS coupled with the immersed boundary method has also been extended for the study of fluid–structure interaction problems with large deformation on the simple Cartesian mesh. For example, Wang et al. apply IBM-LBFS to simulate the flow-induced vibration of single and multiple hyperelastic beams behind a cylinder [43] and falling freely solid objects [44,45]. Liu et al. [46] solve 3D FSI problems of complex multiple flexible structures. However, it is rare to combine the LBFS with the ALE technique on unstructured dynamic meshes for fluid–structure interactions in aeronautical applications, such as airfoil and wing flutter in compressible flows, which motivates the present work.

In this study, an ALE-LBFS coupled with the mode superposition method is developed and applied to study both 2D and 3D transonic flutter phenomenon on dynamic unstructured meshes. To simulate the transonic flow field on the dynamic meshes, the vertex-centered ALE-LBFS with the non-free parameter lattice model D1Q4 [47] is presented. With the aid of reconstructed variables, the lattice Boltzmann solutions are obtained locally and then employed to evaluate the fluxes across the interfaces. As the general application of ALE to other problems, an additional velocity accounting for mesh motion and morph is involved in the convective term of the governing equations. The mode superposition method is applied to solve for the structural motion caused by the action of the aerodynamic, structural, and inertial forces. Structural mode equations are obligatory for the dynamic response of 2D airfoil and 3D wing, which project the generalized physical domain onto the mode space. The first few mode equations are accurate enough to recover the real structural dynamics so as to reduce the computational burden. The exchange of information between the ALE-LBFS and the mode superposition method is achieved by interpolations based on spline matrices using radial basis functions (RBF). A fully implicit dual time-step method accelerates the computation. In the pseudo-time step, the artificial mass term induced by geometric conservation law (GCL) is eliminated with the controlled order of temporal error, as proposed by Biedron [48].

The rest of this paper is organized as follows: Section 2 states the principles involved in the flutter prediction, including ALE kinematic description, ALE formulation of LBFS, geometric conservation law, structural solver, interface mapping, and the coupling method. Section 3 validates the proposed method by simulating a variety of 2D and 3D FSI problems. Section 4 presents the conclusions.

2. Methodology

2.1. Governing Equations of the Aeroelastic System

The aeroelastics is one of coupled fluid–structure interaction problems. As introduced in the reference [49], its general governing equations of the aeroelastic system include three fields to account for the evolution of fluid, structural, and computational meshes, respectively. Their coupled partial differential equations are described as follows:

$$\left.\frac{\partial(J\mathbf{w})}{\partial t}\right|_{\xi} + J\nabla_x \cdot \left(F(\mathbf{w}) - \frac{\partial x}{\partial t}\mathbf{w}\right) = J\nabla_x \cdot R(\mathbf{w}) \tag{1}$$

$$\rho_s \frac{\partial^2 \mathbf{u}_s}{\partial t^2} - div\left(\sigma_s\left(\varepsilon_s(\mathbf{u}_s), \frac{\partial \varepsilon_s}{\partial t}(\mathbf{u}_s)\right)\right) = b \tag{2}$$

$$\widetilde{\rho}\frac{\partial^2 x}{\partial t^2} - div(\widetilde{\sigma}(\widetilde{\varepsilon}(x))) = 0 \tag{3}$$

Equation (1) is the ALE conservative form of the Navier–Stokes equations where $x(t)$ is the time-dependent displacement of a fluid grid point, and ξ, as well as $J = \det(dx/d\xi)$, denotes its position in a reference coordinate. Vector $\mathbf{w}$ is the convective variable in the fluid domain, and F and R represent the convective and diffusive ALE fluxes. Depending on the type of considered physical problem, such as compressibility and viscosity, diverse fluid governing equations are developed. Limited by the transonic flutter problem, the viscous Newtonian flow in the compressible regime is governed by Navier–Stokes equations. If necessary, the large eddy simulation or RANS is employed to augment the compressible

Navier–Stokes equations by solving transport equations of additional turbulent variables. Rather, the unsteady aerodynamic forces are computed by the compressible Euler equations if viscosity is neglected.

In the elastodynamic Equation (2), $\mathbf{u}_s$ and ρ_s denote the displacement and density of the structure, $\boldsymbol{\sigma}_s$ and $\boldsymbol{\varepsilon}_s$ are the tensors for stress and strain, and b indicates the body forces. By the finite element method, Equation (2) is solved and is capable of expressing the nonlinearity of structure with satisfactory accuracy. In fact, under the assumption of small displacement amplitudes, the harmonic motion method is a common alternative to alleviate the intensive computation of the structure equations. Modal structural equations are adopted here to achieve satisfactory accuracy with acceptable computational burden.

Equation (3) represents the dynamics of moving computational meshes. Similar to the elastodynamic equation, it is treated as a pseudo-structural system. Since the technology is out of interest in this work, the inverse-volume spring method is applied directly to treat the mesh deformation.

Figure 1 displays that the statement of boundary conditions at the dynamic interface is necessary, which couples tightly the aforementioned Equations (1)–(3) and closes the complete governing equations of the aeroelastic system. First, Equations (1) and (3) are directly coupled. Then, under the supposition that u_f, p, $\boldsymbol{\sigma}_f$ represent the ALE displacement, pressure, and viscous stress tensor, respectively, the constraint conditions at the interface between fluid and structure are expressed as

$$\boldsymbol{\sigma}_s \cdot \mathbf{n} = -p\mathbf{n} + \boldsymbol{\sigma}_f \cdot \mathbf{n},\ on\ \Gamma \tag{4}$$

$$\frac{\partial u_s}{\partial t} = \frac{\partial u_f}{\partial t},\ on\ \Gamma \tag{5}$$

where Γ denotes the fluid–structure interface boundary, and $\mathbf{n}$ is the corresponding normal vector. On the interface boundary Γ, Equation (4) describes the tractions equilibrium between the structure and fluid domain. Additionally, the compatibility of velocity among the structure and fluid is guaranteed by Equation (5).

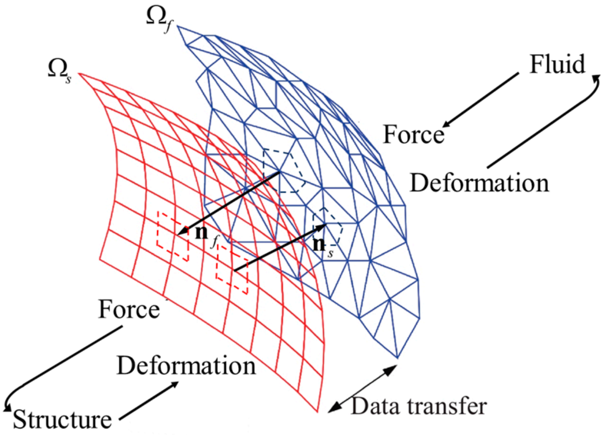


Figure 1. Schematic of an interface boundary where fluid (Ω_f) and solid domain (Ω_s) meet and their outward normal vectors n_f and n_s point in opposite directions.

In conjunction with continuity constraint, the following equations describe the coupling between the structure and dynamic mesh

$$x = u_s,\ on\ \Gamma \tag{6}$$

$$\frac{\partial x}{\partial t} = \frac{\partial u_s}{\partial t},\ on\ \Gamma \tag{7}$$

2.2. ALE-LBFS for Fluid Solver

2.2.1. Macroscopic Conservation Laws

As one of the core components of flutter analysis, the lattice Boltzmann flux solver (LBFS) acts as a fluid solver in the ALE formulation. LBFS is a novel numerical solver developed from lattice Boltzmann equations. It involves the evaluation of flux at the interface and obeys laws of conservation for mass, momentum, and energy. In a Cartesian coordinate system of unstructured meshes, Equation (1) is rewritten, and the unsteady ALE-LBFS for a finite control volume is

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F}_c - \nabla \cdot \mathbf{F}_g - \nabla \cdot \mathbf{F}_v = 0 \tag{8}$$

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{bmatrix}, \tag{9}$$

$$\mathbf{F}_c = \begin{bmatrix} \rho u_n \\ \rho \mathbf{u} u_n + p\mathbf{n} \\ (\rho E + p) u_n \end{bmatrix}, \mathbf{F}_g = \begin{bmatrix} \rho u_{g,n} \\ \rho \mathbf{u} u_{g,n} \\ \rho E u_{g,n} \end{bmatrix} \text{ and } \mathbf{F}_v = \begin{bmatrix} 0 \\ \boldsymbol{\tau} \cdot \mathbf{n} \\ \boldsymbol{\Theta} \cdot \mathbf{n} \end{bmatrix} \tag{10}$$

where the vector $\mathbf{W}$ is the conservative variable, $\mathbf{F}_c$ denotes the convective flux, $\mathbf{F}_g$ indicates the additional flux contributed by moving mesh, and $\mathbf{F}_v$ is the viscous flux. For a detailed description, it is necessary to state that ρ is density, E is the total energy in unit mass. $\mathbf{u}$ and $\mathbf{u}_g$ are the fluid velocity vector and the mesh velocity vector. The subscript n in the u_n and $u_{g,n}$ indicates the normal velocity component. $\mathbf{n}$ is the unit normal vector. Aside from static pressure p, unit tensor $\mathbf{I}$, and time t, tensor $\boldsymbol{\tau}$ is the viscous stress as common and $\boldsymbol{\Theta}$ describes the contribution made by other physical effects, such as heat conduction.

Adapting spatial discretization with a vertex-centered finite-volume scheme, Equation (8) is semi-discrete as

$$\frac{d\mathbf{W}_i}{dt} = -\frac{1}{d\Omega_i} \sum_{k=1}^{N_f} \left(\mathbf{F}_{ck} - \mathbf{F}_{gk} - \mathbf{F}_{vk} \right) S_k \tag{11}$$

where $\mathbf{W}_i$ is the vector of aforementioned conservative variables specified for the i-th control volume, and $\mathbf{F}_{ck}$ and $\mathbf{F}_{vk}$ subscripted with k are the vectors of convective and viscous flux across the kth interface of control volume for index i, respectively. $\mathbf{F}_{gk}$ is the additional flux contributed by mesh motion on the kth interface. It is common to say that S_k denotes the area of the kth interface, let alone cell volume Ω_i and the number of the interface N_f.

Figure 2 presents a schematic diagram for the control volume of a median-dual cell-vertex scheme. The interfaces of a control volume are made up of the lines for 2D or faces for 3D linked by the midpoint of the edge and element center around the considered grid node. Contrary to the cell-centered approach, $\mathbf{W}$ is stored at the grid node together with mesh velocity $\mathbf{u}_g$. As a consequence, the solutions of governing equations and mesh velocity are directly computed and obtained at the nodes at the same time. Unlike the conventional LBFS built on a cell-centered scheme treating dynamic meshes where it has to integrate and average grid velocity from grid nodes to the cell center, it is novel and simple to establish LBFS with a cell-vertex scheme.

To solve Equation (11), it needs to separately evaluate the convective, viscous, and additional ALE flux according to Equation (10). First, the non-free parameter D1Q4 model [42,47] is responsible for building connections between local fluid properties and lattice velocity and density distribution. Figure 3 shows the distribution of discrete lattice particles. In the assumption of the 1D model, the lattice particles transport in lines. Derived from the conservation of moment, the mathematic formulation of D1Q4 comes out as

$$g_1 = \frac{\rho\left(-d_1 d_2^2 - d_2^2 u + d_1 u^2 + d_1 c^2 + u^3 + 3uc^2\right)}{2d_1\left(d_1^2 - d_2^2\right)}, \tag{12}$$

$$g_2 = \frac{\rho\left(-d_1d_2^2 + d_2^2u + d_1u^2 + d_1c^2 - u^3 - 3uc^2\right)}{2d_1\left(d_1^2 - d_2^2\right)}, \tag{13}$$

$$g_3 = \frac{\rho\left(d_1^2d_2 + d_1^2u - d_2u^2 - d_2c^2 - u^3 - 3uc^2\right)}{2d_2\left(d_1^2 - d_2^2\right)}, \tag{14}$$

$$g_4 = \frac{\rho\left(d_1^2d_2 - d_1^2u - d_2u^2 - d_2c^2 + u^3 + 3uc^2\right)}{2d_2\left(d_1^2 - d_2^2\right)} \tag{15}$$

with

$$d_1 = \sqrt{u^2 + 3c^2 - \sqrt{4u^2c^2 + 6c^4}}, \tag{16}$$

$$d_2 = \sqrt{u^2 + 3c^2 + \sqrt{4u^2c^2 + 6c^4}} \tag{17}$$

where g_i and d_α are the equilibrium distribution function and lattice velocity depending on the state of α-direction. c represents the peculiar velocity of particles defined as $c = \sqrt{p/\rho}$. At the angle of the interface, the lattice particle is moving across or far away from it with corresponding lattice velocity and distribution. It is essential to transform the conservative variables to the local frame specified for each interface. $\mathbf{u}_\tau$ is the tangential component of velocity and has the formulations as

$$\mathbf{u}_\tau = \mathbf{u} - u_n\mathbf{n} \tag{18}$$

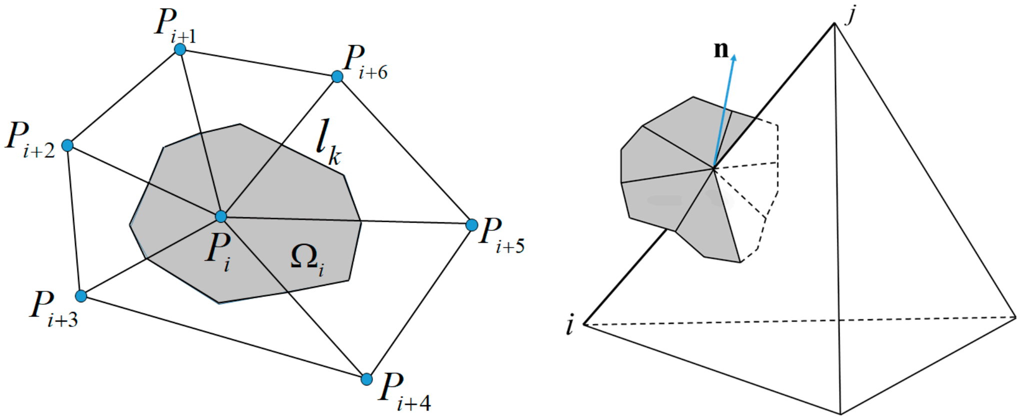


Figure 2. Control volume of a median-dual cell-vertex scheme for (**left**) 2D and (**right**) 3D.

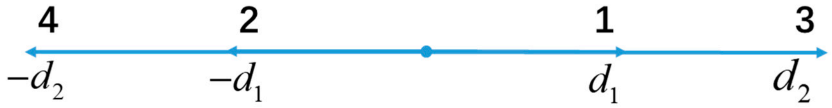


Figure 3. Distribution of discrete lattice velocities in 1D model.

Substituting Equation (18) into Equations (9) and (10), the variables of interest can be rewritten in the local Cartesian coordinate with $\mathbf{u}_\tau$ and u_n as

$$\mathbf{W} = \bar{\mathbf{W}} + \hat{\mathbf{W}} = \begin{bmatrix} \rho \\ \rho u_n\mathbf{n} \\ \rho\left(\frac{u_n^2}{2} + e\right) \end{bmatrix} + \begin{bmatrix} \rho \\ \rho\mathbf{u}_\tau \\ \rho\frac{\|\mathbf{u}_\tau\|^2}{2} \end{bmatrix} \tag{19}$$

$$\mathbf{F}_c = \bar{\mathbf{F}}_c + \hat{\mathbf{F}}_c = \begin{bmatrix} \rho u_n \\ (\rho u_n^2 + p)\mathbf{n} \\ \left(\rho\left(\frac{u_n^2}{2} + e\right) + p\right)u_n \end{bmatrix} + \begin{bmatrix} \rho u_n \\ \rho\mathbf{u}_\tau u_n \\ \rho u_n\frac{\|\mathbf{u}_\tau\|^2}{2} \end{bmatrix} \tag{20}$$

Furthermore, LBFS evaluates the normal component of $\mathbf{W}$ and $\mathbf{F}_c$ using the lattice Boltzmann solution reconstructed locally. Using the distribution functions $f_\alpha(\mathbf{r}, t)$, the components contributed by normal velocity have the following formulation

$$\bar{\mathbf{W}} = \sum_{\alpha=1}^{N} \boldsymbol{\varphi}_\alpha f_\alpha(\mathbf{r}, t) \tag{21}$$

and

$$\bar{\mathbf{F}}_c = \sum_{\alpha=1}^{N} \xi_\alpha \boldsymbol{\varphi}_\alpha f_\alpha(\mathbf{r}, t) \tag{22}$$

where ξ_α is the particle velocity in the α-direction, i.e., $\xi_1 = d_1, \xi_2 = -d_1, \xi_3 = d_2, \xi_4 = -d_2$. With the potential energy of particles $e_p = [1 - (\gamma - 1)/2]e$, $\boldsymbol{\varphi}_\alpha$ stands for the moments

$$\boldsymbol{\varphi}_\alpha = \begin{bmatrix} 1 \\ \xi_\alpha \\ \frac{\xi_\alpha^2}{2} + e_p \end{bmatrix} \tag{23}$$

Within the D1Q4 compressible LBFS, the evaluation of convective flux depends on $f_\alpha(\mathbf{r}, t)$, among which the equilibrium part acts as the inviscid flux while the latter is responsible for numerical dissipation. Forced on $f_\alpha(\mathbf{0}, t)$ at the cell interface, its formulation is as

$$f_\alpha(\mathbf{0}, t) = g_\alpha(\mathbf{0}, t) + f_\alpha^{neq}(\mathbf{0}, t) \tag{24}$$

Following the expansion of the Chapman–Enskog analysis, the nonequilibrium part $f_\alpha^{neq}(\mathbf{0}, t)$ truncated by the first term is discretized by the finite difference scheme as

$$f_\alpha^{neq}(\mathbf{0}, t) = -\frac{\tau}{\delta t}(g_\alpha(\mathbf{0}, t) - g_\alpha(-\xi_\alpha \delta t, t - \delta t)) \tag{25}$$

Substituting Equation (25) into Equation (22), the flux $\bar{\mathbf{F}}_c$ at the interface can be calculated by

$$\bar{\mathbf{F}}_c = \sum_{\alpha=1}^{N} \xi_\alpha \boldsymbol{\varphi}_\alpha g_\alpha(0, t) - \frac{\tau}{\delta t} \sum_{\alpha=1}^{N} \xi_\alpha \boldsymbol{\varphi}_\alpha (g_\alpha(0, t) - g_\alpha(-\xi_\alpha \delta t, t - \delta t)) \tag{26}$$

where τ is the collision time and δt is the streaming time step. N is the number of velocities in the lattice model and is equal to 4 for the D1Q4 lattice model.

From Equation (26), it is apparent that $\bar{\mathbf{F}}_c$ is determined by the density distribution function at the cell interface $g_\alpha(\mathbf{0}, t)$ as well as that streaming from the neighboring location $g_\alpha(-\xi_\alpha \delta t, t - \delta t)$. In fact, this coincides with the conventional convective scheme at the interface of the cell control volume, as depicted in Figure 4. Let $\mathbf{F}_c^I$ and $\mathbf{F}_c^{II}$ be

$$\mathbf{F}_c^I = \sum_{i=1}^{N} \xi_\alpha \boldsymbol{\varphi}_\alpha g_\alpha(0, t) \tag{27}$$

$$\mathbf{F}_c^{II} = \sum_{i=1}^{N} \xi_\alpha \boldsymbol{\varphi}_\alpha g_\alpha(-\xi_\alpha \delta t, t - \delta t) \tag{28}$$

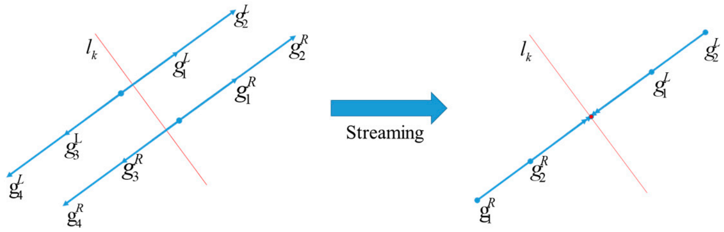


Figure 4. Streaming process in the D1Q4 model.

In order to evaluate $\mathbf{F}_c^I$, $g_\alpha(0,t)$ can be obtained using Equations (12)–(15) by the conservative variable $\bar{\mathbf{W}}$ at the interface.

From the lattice structure in Figure 4, an upwind scheme is applied to evaluate $g_\alpha(-\xi_\alpha \delta t, t-\delta t)$ by reconstructed variables on the left and right side of the interface, that is,

$$g_\alpha(-\xi_\alpha \delta t, t-\delta t) = \begin{cases} g_\alpha^L \; if \; \alpha = 1,3 \\ g_\alpha^R \; if \; \alpha = 2,4 \end{cases} \tag{29}$$

Thus, $\bar{\mathbf{W}}$ can be computed by

$$\bar{\mathbf{W}} = \sum_{\alpha=1}^{N} \boldsymbol{\varphi}_\alpha g_\alpha(0,t) = \sum_{\alpha=1,3} \boldsymbol{\varphi}_\alpha g_\alpha^L + \sum_{\alpha=2,4} \boldsymbol{\varphi}_\alpha g_\alpha^R \tag{30}$$

where g_α^L and g_α^R are computed from Equations (12)–(15). Once the evaluation of $\bar{\mathbf{W}}$ is completed, $g_\alpha(0,t)$ can be calculated by Equations (12)–(15) and then used to calculate $\mathbf{F}_c^I$ in Equation (27). Within the calculation of $\mathbf{F}_c^{II}$ in Equation (28), $g_\alpha(-\xi_\alpha \delta t, t-\delta t)$ can be determined according to the streaming discretion. Then,

$$\mathbf{F}_c^{II} = \sum_{\alpha=1}^{N} \xi_\alpha \boldsymbol{\varphi}_\alpha g_\alpha(-\xi_\alpha \delta t, t-\delta t) = \sum_{\alpha=1,3} \xi_\alpha \boldsymbol{\varphi}_\alpha g_\alpha^L + \sum_{\alpha=2,4} \xi_\alpha \boldsymbol{\varphi}_\alpha g_\alpha^R \tag{31}$$

To complete the evaluation of $\mathbf{F}_c$ in Equation (20), the terms related to $\rho u_n \mathbf{u}_\tau$ and $\rho u_n \|\mathbf{u}_\tau\|^2$ are reconstructed locally as

$$\rho u_n \mathbf{u}_\tau = \sum_{\alpha=1,3} \xi_\alpha g_\alpha^L \mathbf{u}_\tau^L + \sum_{\alpha=2,4} \xi_\alpha g_\alpha^R \mathbf{u}_\tau^R \tag{32}$$

$$\rho u_n \|\mathbf{u}_\tau\|^2 = \sum_{\alpha=1,3} \xi_\alpha g_\alpha^L \left\|\mathbf{u}_\tau^L\right\|^2 + \sum_{\alpha=2,4} \xi_\alpha g_\alpha^R \left\|\mathbf{u}_\tau^R\right\|^2 \tag{33}$$

Substituting Equations (27), (28), (32) and (33) into Equation (20), the final convective flux $\mathbf{F}_c$ is obtained.

The term $\mathbf{F}_{gk}$ is calculated by local conservative variables and mesh velocity is reconstructed by the arithmetic average method. Other terms with regard to temporal discretization and viscous flux $\mathbf{F}_v$ are all directly applied in the same way as the conventional finite volume ALE-NS solver, along with the boundary and initial conditions. In viscous flow, the unsteady RANS with SA turbulence model is used to simulate the flow field.

2.2.2. Geometric Conservation Law

It is worth noting that the GCL should be obeyed in the simulation of unsteady flows with dynamic meshes. Supposing that conservative variables $\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$, the mathematical form of GCL is recovered as

$$\frac{\partial V}{\partial t} = \int_S \mathbf{U}_g \cdot \mathbf{n} dS \tag{34}$$

Since introduced by Thomas and Lombard [50], GCL has accounted for alleviating the numerical instability induced by discretization errors in the treatment of grid velocity. The Geometric Conservation Law may be discretized temporally by implicit dual time-stepping as

$$\frac{V^{p+1} - V^p}{\Delta \tau} + \frac{1}{\Delta t}\left(\phi_p V^p + \phi_n V^n + \cdots\right) = -\frac{1}{V}\left(R_{GCL}^p + \frac{\partial R_{GCL}^p}{\partial V} \Delta V^p\right) \tag{35}$$

The term R^p_{GCL} in Equation (35) is evaluated as follows

$$R^p_{GCL} = \oint_{\partial V} \mathbf{U}_g \cdot \mathbf{n} dS \cong \sum_{k=1}^{N_f} U_{gk} dS_k \tag{36}$$

Taking advantage of surface integral with the midpoint rule, it yields a local error in the order of $\Delta h \Delta S \sim \Delta h^3$ where Δh is a characteristic dimension of the interface. The accuracy of mesh velocity is in the same order as that of temporal discretization of Equation (11).

In order to control temporal errors, one can estimate the difference in residual contribution at each time step. It is evaluated by approximations of time derivatives in sequent levels. The norm of temporal error is selected as the exit criterion to terminate the subiteration loop of the dual time-stepping procedure. Such a strategy makes temporal accuracy uniform among all time steps and eliminates the prior work of selecting iterative steps or residual reduction.

2.3. Modal Solver for Structural Field

Flutter is a classic topic of the aeroelastic problem, which is the dynamic response of inertia force, aerodynamic force, and elastic force. Without loss of generality, its motion can be described in the following governing equations

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F} \tag{37}$$

where the matrices **M**, **C**, **K** denote mass, dampness, and stiffness. $\ddot{\mathbf{q}}$, $\dot{\mathbf{q}}$, and $\mathbf{q}$ are the acceleration, velocity, and displacement, respectively. **F** is the force imposed on the structure. In order to solve Equation (37), the mode analysis is employed to decouple it into an eigenvalue problem. The first N modes are taken into consideration and are enough to approximate a dynamic system. It is given by

$$\mathbf{q} = \sum_{r=1}^{N} \eta_r \mathbf{\Phi}_r \tag{38}$$

where $\mathbf{\Phi}_r$ is the r-th eigenvector of the generalized eigenproblem, and η_r is the corresponding normal coordinate. Truncated by the first N mode frequency, the displacement vector is determined by the coordinate space of those N eigenmodes. The displacement vector can be decomposed as

$$\mathbf{q} = \mathbf{\Phi}\boldsymbol{\eta} \tag{39}$$

Premultipling Equation (37) by Φ^T, the generalized equations are formed as

$$\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi}\ddot{\boldsymbol{\eta}} + \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi}\dot{\boldsymbol{\eta}} + \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}\boldsymbol{\eta} = \mathbf{\Phi}^T\mathbf{F} \tag{40}$$

It is notable that $\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi}$ is an identity matrix. For a simplified and straight description, Equation (40) can be rewritten as

$$\ddot{\eta}_i + 2\zeta_i\omega_i\dot{\eta}_i + \omega_i^2\eta_i = Q_i \tag{41}$$

$$Q_i = \mathbf{\Phi}_i^T\mathbf{F} \tag{42}$$

$$2\zeta_i\omega_i = \Phi_i^T\mathbf{C}\Phi_i \tag{43}$$

$$\omega_i^2 = \Phi_i^T\mathbf{K}\Phi_i \tag{44}$$

where ζ_i and ω_i are the modal damping and natural frequency of the i-th mode, respectively.

2.4. Fluid-Solid Coupling

In coupling Equations (8) and (37), two challenges arise. The former originates from the question of how often solutions of the fluid and structure solver are computed within one physical time step and how the relevant kinematic and dynamic quantities are exchanged. Depending on the frequency of the fluid–solid interaction, the coupling strategies can be distinguished as the loose-coupling method and the tight-coupling method. For the former, the coupling loop marches and is fixed per time step. The coupling constraints are weakly enforced in each time step because data only communicate after each discrete time instance. Thus, it is hard to guarantee the balance of forces and displacements between the fluid and solid domains at the interface.

The favorite tight-coupling method relieves the restriction of the physical time step, which is small enough so that the coupling accuracy is guaranteed. Tight-coupling method exchanges data between fluid and structure several times within each physical time step, as described in Figure 5. These operations, which are equivalent to pre-estimation and correction iterations, eliminate time step delay and improve temporal accuracy.

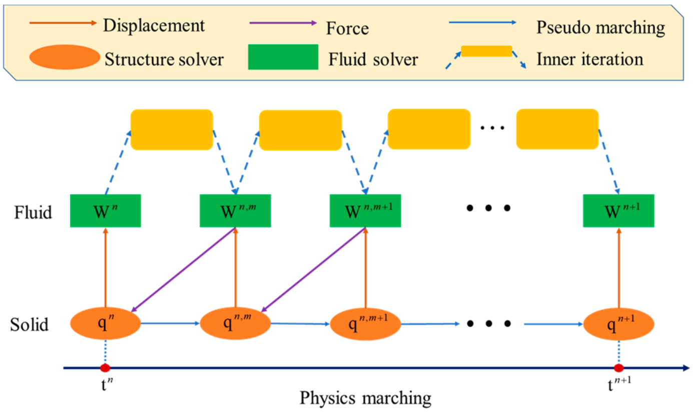


Figure 5. Computational flowchart for the tight-coupling method.

In Figure 5, the computational flowchart for the tight-coupling method displays the progress of one physical time step from t^n to t^{n+1}. In the wake of structural response yielding from structure solver q^n at the time step t^n, the new mesh is regenerated and grid velocity is calculated according to the updated structural positions. After solving fluid governing equations, the new flow state is obtained and the aerodynamic forces acting on the solid are solved. Then, the next position of the dynamic object is determined from the solutions of structural equations. For matching mesh, it is natural and easy to realize the above displacements and force transfer. However, fluid and structure meshes do not necessarily coincide in a node-to-node manner at the interface in most FSI problems. Typically, the fluid domain is discretized with finer levels of mesh resolution compared with the solid meshes. It constitutes the second challenge to couple the fluid and solid solver and means that some mapping between the fluid and solid solver has to be applied to interpolate necessary information between them. The general data transformation is in the form of

$$\mathbf{h} = \mathbf{G}\mathbf{x} \tag{45}$$

$$\mathbf{F}_s = \mathbf{G}^T\mathbf{F}_a \tag{46}$$

where matrix **x** represents the displacement for all elements of structural grids while **h** represents the displacement for all vertexes accounting for the deformable boundary. **G** is the connection matrix that relates and interpolates the elements on the structural grids to the aerodynamic grids. $\mathbf{G}^T$ is the transposed matrix interpolating force load on the grid of the structure from loading on the aerodynamic grids. These mappings are guaranteed by the principle of virtual work.

Here, the radial basis function (RBF) establishes the bridge to realize the communication of interface displacement and loading force between the fluid solver and the structure solver. The interpolation function adapts the radial basis function corrected with a linear mapping related to the interpolation location as

$$w(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z + \sum_{i=1}^{N} \lambda_i r_i(x, y, z) \tag{47}$$

where $r_i(x, y, z)$ is the connection matrix relating the interpolating point (x, y, z) with the i-th source point (x_i, y_i, z_i). Its mathematic expression can be defined as

$$r_i(x, y, z) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \tag{48}$$

For interpolation using the function $w(x, y, z)$, it is necessary to obtain all coefficients in Equation (47) first. With constraints for coefficient λ_i,

$$\sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \lambda_i x_i = \sum_{i=1}^{N} \lambda_i y_i = \sum_{i=1}^{N} \lambda_i z_i = 0 \tag{49}$$

a matrix mapping as stated in Yuan's work [15] is constructed

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ w_1 \\ \cdots \\ w_N \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & x_1 & \cdots & x_N \\ 0 & 0 & 0 & 0 & y_1 & \cdots & y_N \\ 0 & 0 & 0 & 0 & z_1 & \cdots & z_N \\ 1 & x_1 & y_1 & z_1 & 0 & \cdots & r_{1N} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_N & y_N & z_N & r_{N1} & \cdots & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \lambda_1 \\ \cdots \\ \lambda_N \end{bmatrix} = \mathbf{KC} \tag{50}$$

Once a_0, a_1, a_2, a_3 and F_i are known, the value at any point is accessible with the aid of function $w(x, y, z)$. Figure 6 exhibits the data transformation of displacement and load between fluid and structure with Equations (45), (46), and (50).

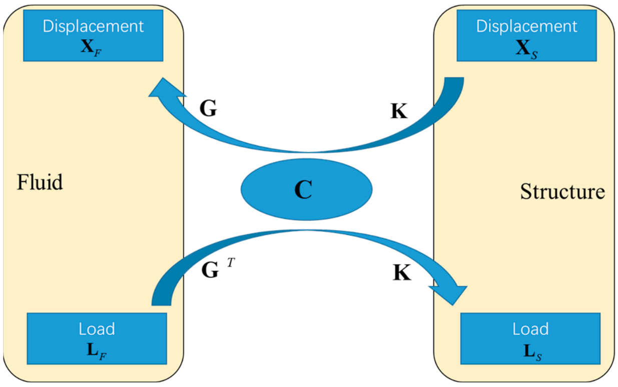


Figure 6. Displacement and force transfer between fluid and structure solver.

2.5. Computational Sequence

Figure 7 displays the complete solution process with the tight-coupling method. The procedure is as follows

1. In the very first time step, start with the prior setting of physical states and impose grid velocity with zero to avoid fictitious velocities.
2. Interpolate the positions of aerodynamic mesh via Equation (45) from the structural positions. Meanwhile, compute grid velocities if required.
3. Run the fluid solver in Equation (8). Owing to the vertex-centered finite volume method, obtain and store the surface aerodynamic forces at the fluid nodes.

4. Judge the type of structural motion. If the structural motion is specified, impose the displacement field on the structural nodes directly. Otherwise,
   (i) Interpolate the forces onto the structural nodes via Equation (46).
   (ii) Run the structural solver in Equation (37) to update the positions.
5. Check the convergence on the structural displacements by evaluating the MSE of the incremental structural displacements.
6. Once convergence is achieved, the nest structural displacements can be calculated. Otherwise, relax the structural solution with an Aitken relaxation.
7. Repeat the steps from 2 to 6 until the maximum of physical time steps is reached.

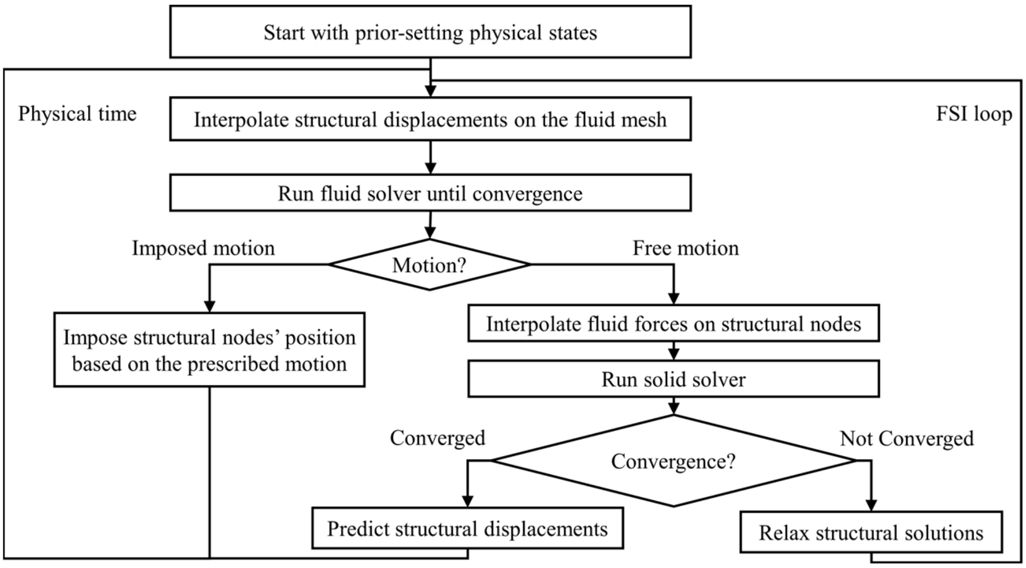

Start with prior-setting physical states
Physical time
Interpolate structural displacements on the fluid mesh
FSI loop
Run fluid solver until convergence
Imposed motion
Motion?
Free motion
Impose structural nodes' position based on the prescribed motion
Interpolate fluid forces on structural nodes
Run solid solver
Converged
Convergence?
Not Converged
Predict structural displacements
Relax structural solutions

Figure 7. Flowchart of the solution process for the FSI problem.

3. Numerical Validation

Since LBFS has the capability of simulating inviscid and viscid flow on steady meshes with comparable performance to other numerical methods, it is interesting to combine LBFS with ALE for unsteady flow on unstructured dynamic meshes. First, the performance of ALE-LBFS on dynamic meshes is tested by fluid-solid coupling with forced motion. In the case of forced pitching of the NACA 0012 airfoil, ALE-LBFS performs well in the 2D problems. The cases of uCRM-13.5 in harmonic motion and forced pitching of the BSCW wing are responsible for validating the ability in 3D problems. Furthermore, flutter boundaries for both NACA 64A010 airfoil and AGARD 445.6 wing are predicted. Their motions of structure respond to the aerodynamic forces.

3.1. FSI with Forced Motion

3.1.1. Forced Pitching of Airfoil

The pitching NACA0012 airfoil in subsonic flow is considered to test the performance of ALE-LBFS. The airfoil is forced to pitch around its pivot axis sinusoidally, on which occasion the angle of attack is modified with the specified motion function depending on time as

$$\alpha(t) = \alpha_m + \alpha_o \sin(\omega t) \tag{51}$$

where α_m means the averaged angle of attack and α_o is the oscillation amplitude.

$$k_c = \frac{\omega c}{2U_\infty} \tag{52}$$

ω is the angular frequency, along with airfoil chord c and free-stream velocity U_∞. Detailed conditions for NACA 0012 are listed in Table 1. Among them, $a = 0.25$ indicates that the pivot axis lies a quarter chord away from the leading edge of the airfoil.

Table 1. Conditions for NACA 0012 airfoil.

Test Case	Ma	$\alpha_m/(°)$	$\alpha_o/(°)$	k_c	Re	a
NACA0012	0.6	2.89	2.41	0.0808	4.8×10^6	0.25

Compared with numerical results obtained by Yang [51], the performance of ALE-LBFS is validated by the forced pithing of the NACA 0012 airfoil. Figure 8 offers a comparison of C_L and C_m with numerical and experimental results. Its aerodynamic coefficients vary over a pitch cycle and coincide with experimental results. Figure 9 plots various snapshots of pressure coefficients distributed on the surface of NACA 0012. They correspond to four phase angles in one pitch cycle, namely, $\phi = 45°, 90°, 135°, 180°$. Calculated by Equation (51), their angles of attack are equal to $\alpha = 4.59°, 5.30°, 4.59°, 2.89°$, respectively. The four subplots indicate that ALE-LBFS preserves satisfactory performance for simulating the forced pitch of the NACA 0012 airfoil in moving meshes.

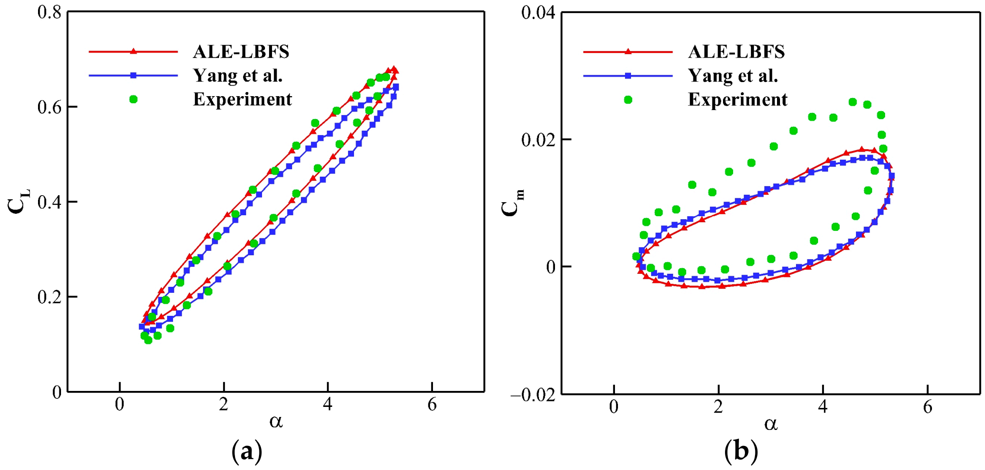


Figure 8. Comparison of C_L and C_m for NACA 0012 airfoil with the numerical and experimental results. (**a**) Coefficient of lift. (**b**) Coefficient of moment.

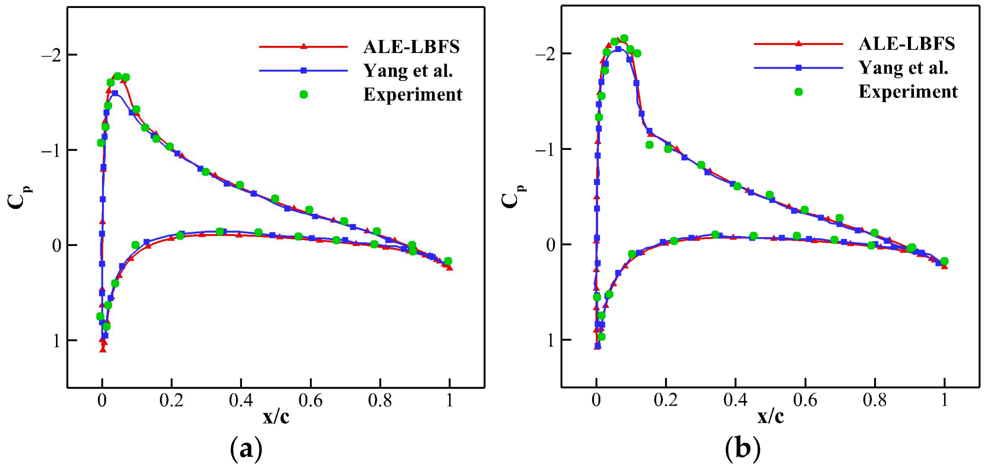


Figure 9. *Cont.*

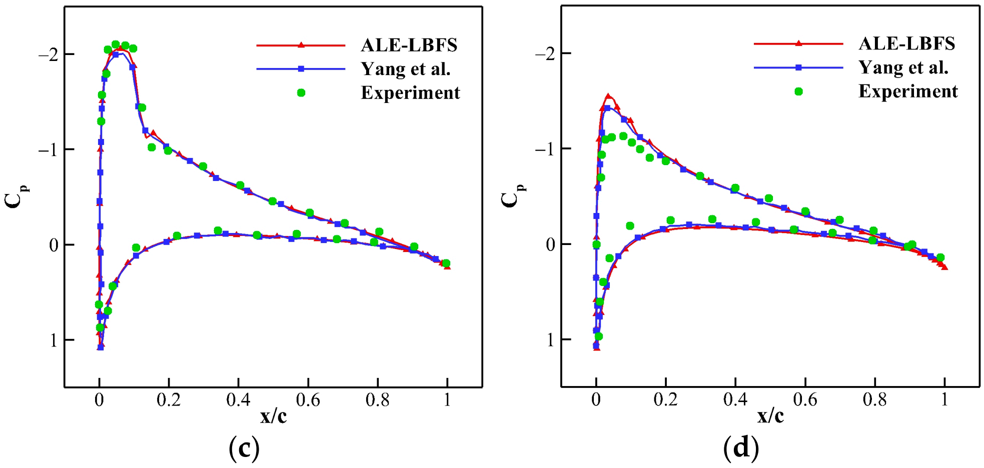


Figure 9. Comparison of pressure coefficients distributed on NACA 0012 airfoil for four different snapshots in one pitch cycle. (**a**) $\alpha = 4.59^\circ, \phi = 45^\circ$. (**b**) $\alpha = 5.30^\circ, \phi = 90^\circ$. (**c**) $\alpha = 4.59^\circ, \phi = 135^\circ$. (**d**) $\alpha = 2.89^\circ, \phi = 180^\circ$.

3.1.2. uCRM-13.5 Wing in Harmonic Motion

The ALE-LBFS is further applied to the uCRM-13.5 wing in harmonic motion. A schematic planform of the uCRM-13.5 [52] is plotted in Figure 10. The semi-span of interest b is 35.996 m and the length of the root chord c_{root} is 12.893 m. The Young's modulus of structural material is 6.9×10^{10} Pa, with Poisson's ratio of 0.3 and density of 2700 kg/m^3.

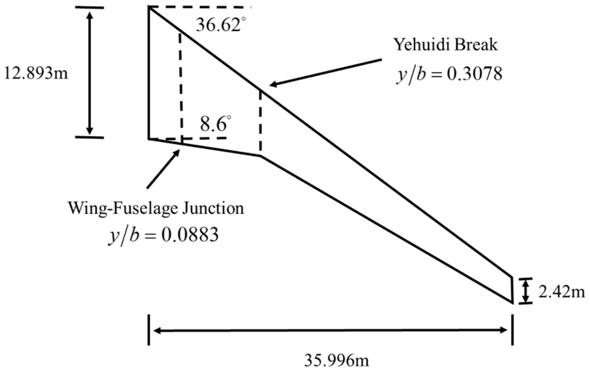


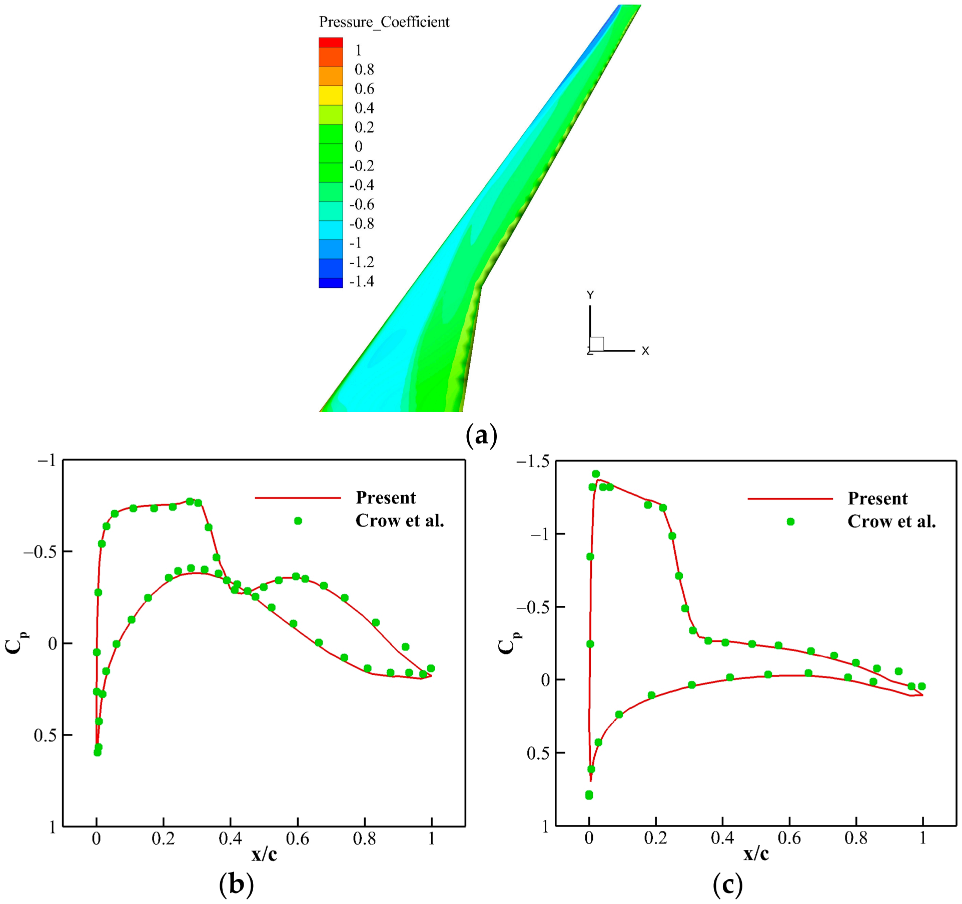
Figure 10. Schematic diagram of the uCRM-13.5 wing.

In order to analyze the calculated pressure coefficient distribution accurately, the harmonic motion is forced on each mode of the uCRM-13.5 wing, respectively. Thus, ahead of harmonic simulations, the first four modes are extracted from the wing structure by a modal approach. Here, we specify that the harmonic motion obeys the following dynamic function

$$f = \eta \sin(\omega t) \tag{53}$$

where η is the factor of amplitude and ω is the natural frequency calculated from reduced frequency $k_c = \omega c_{root}/V$. In the harmonic simulations, the reduced frequency is selected as $k_c = 0.1$ and η is equal to 6.

There exist two complete cycles of harmonic motion in each mode. Measuring the steady-state C_p at the specified spanwise slices, namely, $y/b = 0.35$ and $y/b = 0.89$, their results are presented in Figure 11. Figures 12 and 13 show how C_p changes in the unsteady harmonic motion. In each figure, four plots are shown, which correspond to the first four respective modes. Three curves are drawn for the maximum, middle, and minimum points in the second cycle of the harmonic motion with their phase angles Φ being 90°, 180°, and 270°.

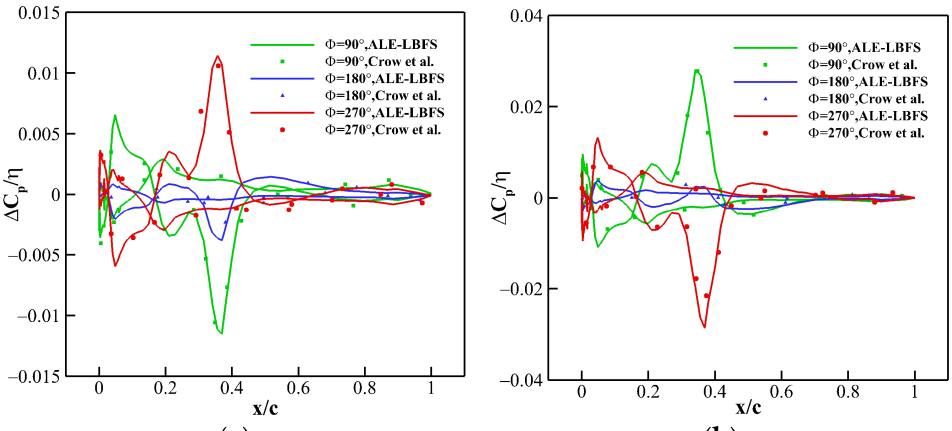


(**a**)

(**b**) (**c**)

Figure 11. Distribution on the wing surface in a steady state. (**a**) Pressure coefficient contour. (**b**) Pressure coefficient distribution along $y/b = 0.35$. (**c**) Pressure coefficient distribution along $y/b = 0.89$.

Normalized by the maximum amplitude, the discrepancy ΔC_p in each curve between the time-dependent C_p and the steady-state C_p is displayed. Similar to the results by Crow et al. [53], it can be seen that C_p at the phase angle of 180° is close to the steady state compared with the other curves. For $\Phi = 90°$ and $\Phi = 180°$, the two plots exhibit a nearly symmetrical relationship with each other. This could be accounted for by the nonsymmetrical airfoil of uCRM-13.5. Although the pressure difference between $\Phi = 90°$ and $\Phi = 180°$ is not equal to zero, their magnitude values are nearly close. This concurs with the idea that pressure is linearly dependent on the displacement.



(**a**) (**b**)

Figure 12. *Cont.*

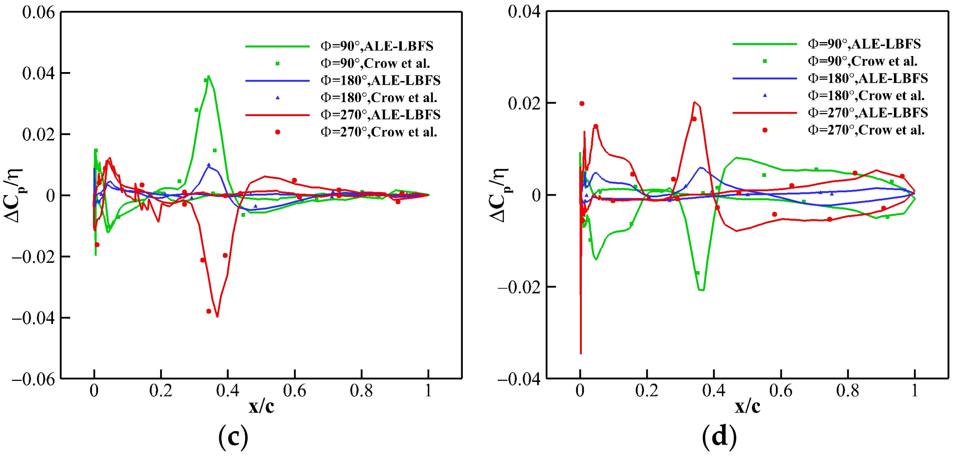


Figure 12. Time-dependent C_p for the wing at $y/b = 0.35$. (**a**) Mode 1, frequency 0.505 Hz. (**b**) Mode 2, frequency 1.670 Hz. (**c**) Mode 3, frequency 2.733 Hz. (**d**) Mode 4, frequency 3.843 Hz.

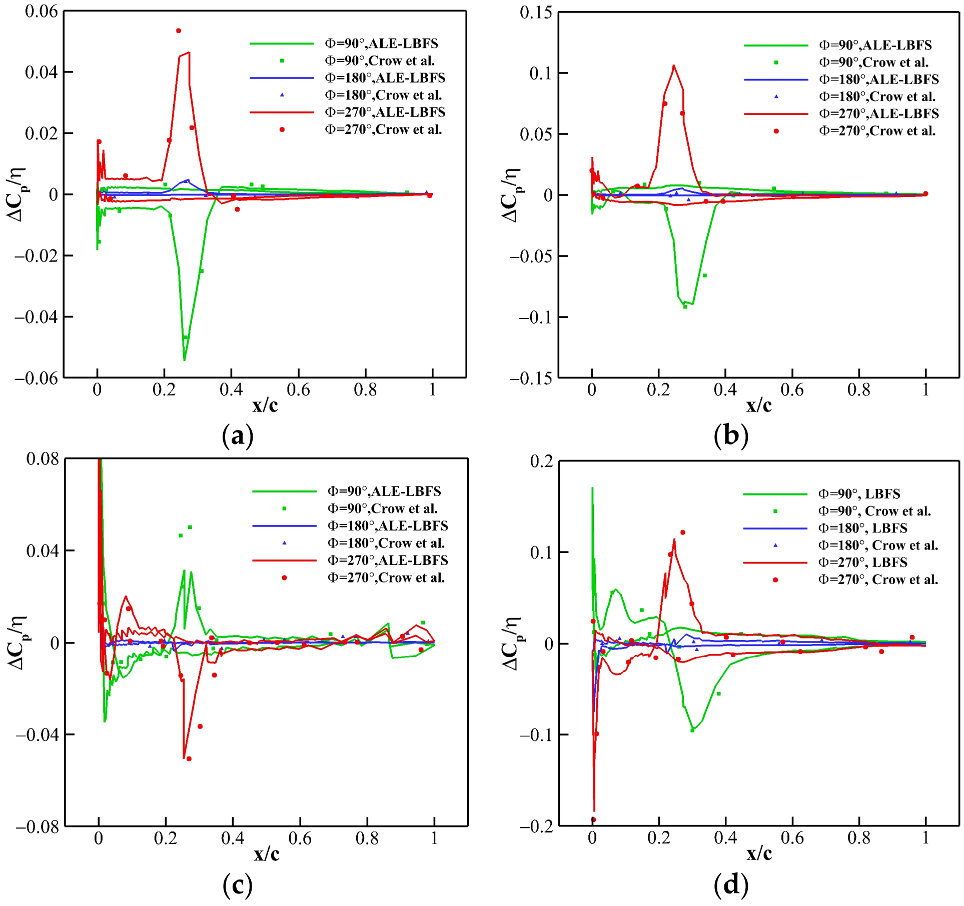


Figure 13. Time-dependent C_p for the wing at $y/b = 0.89$. (**a**) Mode 1, frequency 0.505 Hz. (**b**) Mode 2, frequency 1.670 Hz. (**c**) Mode 3, frequency 2.733 Hz. (**d**) Mode 4, frequency 3.843 Hz.

3.1.3. Forced Pitching of the BSCW Wing

The test case of the BSCW wing is accessible in the Aeroelastic Prediction Workshop for forced oscillations. Supposing that the wing itself is rigid, the structural motion is contributed by the pitch apparatus. The BSCW wing is forced to pitch with Reynolds number Re $= 4.56 \times 10^6$ based on the chord length c = 0.4064 m. The dynamic simulation is tested at a Mach number Ma = 0.7 and a mean angle of attack of 3°. With an oscillation frequency f = 10 Hz and a pitching amplitude of 1 degree, the pitching axis is located at x/c = 0.3.

Figure 14 plots the time-averaged surface C_p distributed on the pitching wing, along with the mean pressure coefficient extracted from the wing span at 60%. The result is compared with the experimental data from the workshop, as well as the numerical results [54,55]. During the pitching motion, a shock wave forms at the leading edge of the top surface. The present result obtained by ALE-LBFS has good agreement with experimental data.

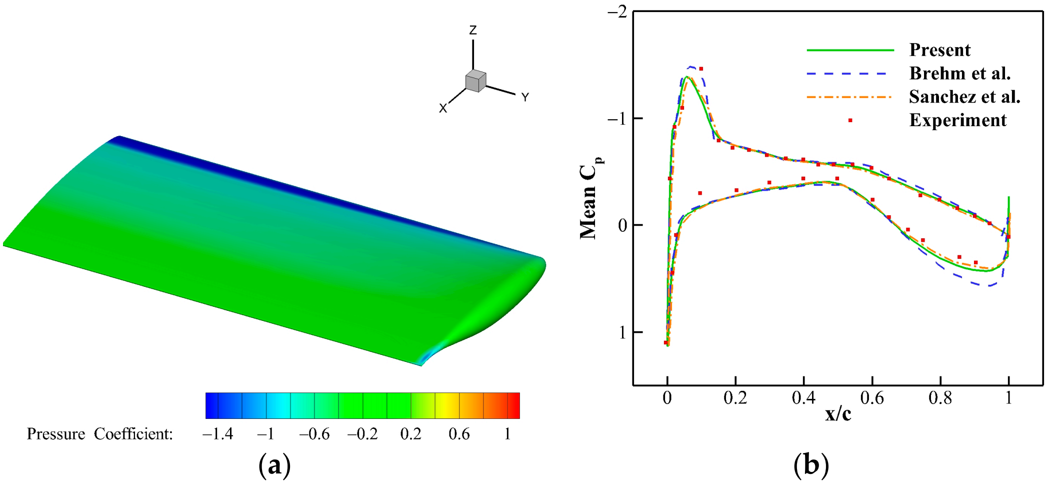


Figure 14. Pressure distribution on 3D pitching wing. (**a**) Mean pressure distribution. (**b**) Mean pressure coefficient at 60% wing span.

For investigating the unsteady aerodynamics, the transfer function is computed by the C_p and the pitch angle. The variation of the amplitude and phase is shown in Figure 15. From the subplot of amplitude, it can be seen that the present ALE-LBFS predicts two peaks on the upper side. Both of them are close to the leading edge. It is visible that some discrepancies between numerical and experimental results exist on the upper surface. These mismatches are common in other numerical work because some local deformations of the wing are ignored in numerical simulation due to the assumption of rigid motion. Compared with Brehm's work, the present result enables the recovery of the formation of the two peaks. Because of the shock wave, the second peak is higher than the first one. The influence of the shock wave is also visible in the phase subplot. Meanwhile, differences arise near the trailing edge. Their magnitudes of phase are close to 180.

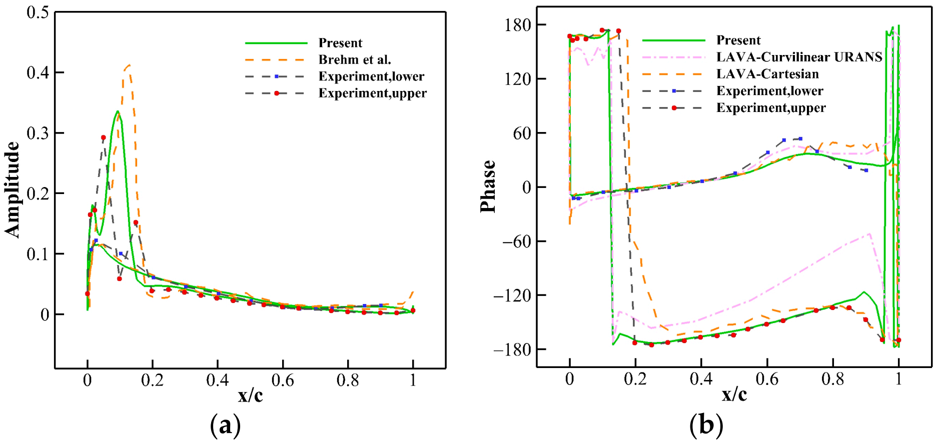


Figure 15. The variation in terms of the amplitude and phase when the forced oscillation frequency f = 10 Hz. (**a**) Amplitude. (**b**) Phase.

3.2. FSI with Free Motion

In this section, flutter prediction and analysis are conducted on the NACA64A010 airfoil and AGARD 445.6 wing. Figure 16 presents their configurations, respectively. For the sake of problem description, exemplified by the typical section of the wing model, AGARG 445.6 wing in 3D problem is viable to analogy.

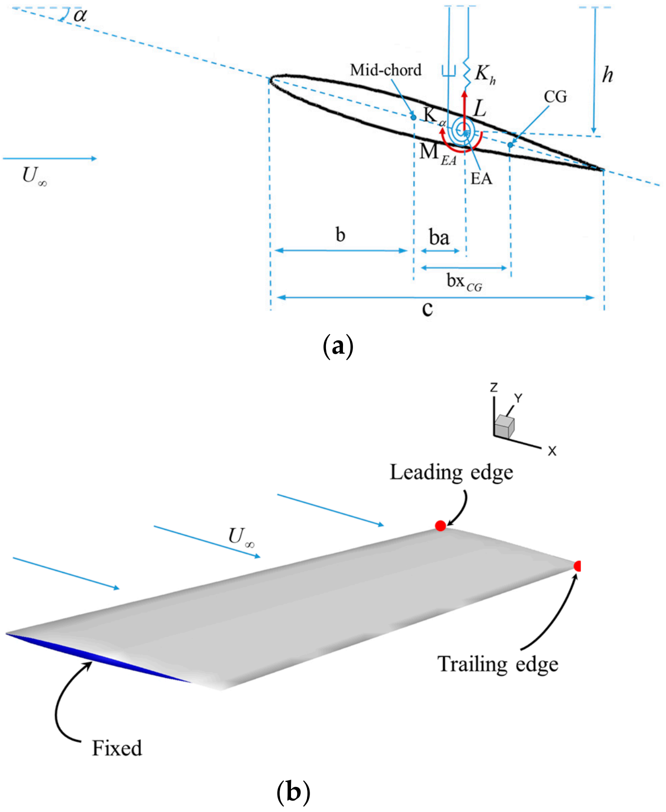


Figure 16. Schematic of (**a**) the typical section wing model and (**b**) AGARD 445.6 wing.

In light of Figure 16a, the model contains two degrees of freedom, referred to as pitching α and plunging h. The motions are governed by

$$\begin{aligned} m\ddot{h} + S_\alpha \ddot{\alpha} + K_h h &= -L \\ S_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + K_\alpha \alpha &= M_{EA} \end{aligned} \tag{54}$$

where L and M_{EA} denote the lift and moment about the elastic axis, respectively. K_h and K_α are blending and torsional spring stiffness. m represents the mass per unit span, S_α denotes the static moment around the elastic axis, and I_α indicates the rotational moment of inertia. With dimensionless time reduced by the uncoupled natural frequency of the torsional spring, $\tau = \omega_\alpha t$, the equation of motion is rewritten as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F} \tag{55}$$

where

$$\mathbf{M} = \begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix}, \tag{56}$$

$$\mathbf{K} = \begin{bmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix} \tag{57}$$

are the dimensionless mass and stiffness matrices,

$$\mathbf{F} = \frac{1}{\pi\mu k_c^2}\begin{bmatrix}-C_l\\2C_m\end{bmatrix} = \frac{U^{*2}}{\pi\mu}\begin{bmatrix}-C_l\\2C_m\end{bmatrix}, \tag{58}$$

$$\mathbf{q} = \begin{bmatrix}\frac{h}{b}\\\alpha\end{bmatrix} \tag{59}$$

states the combination of loads and displacements. C_l and C_m are the coefficients of lift and moments about the elastic axis. $x_\alpha = S_\alpha/b$ is static unbalance, $\omega_h = \sqrt{K_h/m}$ is the uncoupled natural frequency in plunge, $\omega_\alpha = \sqrt{K_\alpha/I_\alpha}$ is the uncoupled natural frequency in pitch. $r_\alpha^2 = I_\alpha/\left(mb^2\right)$ is the squared radius of gyration, U^* is the reduced velocity defined as $U_\infty/(\omega_\alpha b)$, $k_c = \omega_\alpha b/U_\infty$ is the reduced frequency, and $\mu = m/(\rho_\infty v)$ is the mass ratio. ρ_∞ is endowed with the density of flow and v is the volume coefficient.

As a matter of fact, a dimensionless parameter, flutter velocity V_f^*, is put forth to evaluate the critical value. When local velocity exceeds that, a small disturbance is going to amplify.

$$V_f^* = \frac{U_\infty}{b\omega_\alpha\sqrt{\mu}} \tag{60}$$

The function of V_f^* depends on a Mach number, so there exists a flutter boundary differing the situations into stable and unstable conditions. From a practical point of view, good prediction about flutter boundary plays an important role in the transonic region. The indicator V_f^* measures the flutter point in quantity. When V_f^* is lower than the critical value, the oscillation is damped. As V_f^* increases, the neutrally stable point is reached. When V_f^* exceeds the critical value, the amplitude of oscillation diverges, even inducing collapse. To capture the flutter boundary, a series of simulations are performed by varying the V_f^*. In a standard rule, various runs are needed until a zero damping response is yielded. However, the process of operation in this work is simplified under the guidance of the literature [56]. Owing to their accessible flutter boundary, for each Mach number, only several V_f^* spanning beside the sensible and sensitive regions are simulated, with an increment $\Delta V_f^* = 0.1$. The intermediate value of two V_f^* is viewed as the critical point if their responses differ—namely, one is stable while the other is not. For higher accuracy, the increment ΔV_f^* would be halved and the aforementioned process is repeated until the expected accuracy is received.

3.2.1. Flutter of NACA 64A010 Airfoil

The two free-degree structural systems are in free motion in response to the aerodynamic forces, namely, pitch and plunge. The structural parameters for this case are $x_\alpha = 1.8$, $r_\alpha^2 = 3.48$, $\omega_h = \omega_\alpha = 100\ \text{rad/sec}$, and $\mu = 60$. $a = -2.0$ indicates that the pivot axis lies on a semi-chord ahead of the leading edge of the NACA 64A010 airfoil.

The common practice [57,58] is to force the airfoil sinusoidally in pitch around the elastic axis until three full cycles are completed with the frequency ω close to the first mode of vibration. It is notable that the angle of attack is set as 1° for an obvious initial disturbance.

Figure 17 compares the flutter boundaries between the present work calculated by ALE-LBFS and previous references [59–61]. It is apparent that the current algorithm predicts the "transonic dip" phenomenon well. In addition to the flutter boundary in viscid flow, the ALE-LBFS also predicts the inviscid flutter boundary in the Euler simulations. The bottom of the dip is placed at the speed index of $V_f^* = 0.75$ at a Mach number $Ma_\infty = 0.825$ while it is $V_f^* = 0.55$ in the inviscid flow. As can be seen in Figure 17, viscosity is vital when predicting the flutter. The inviscid and viscous simulations agree well with each other until the bottom of the transonic dip. Due to the consideration of viscosity, the critical flutter velocities are lifted in the region of Ma = 0.8–0.9 and the S shape flutter boundary in inviscid simulations disappears. For higher Mach numbers, the predicted viscous flutter boundary has visible discrepancies compared with other references, which is caused by the use of numerical schemes [15]. Figure 18 presents several responses in different pairs of a Mach number and flutter speed velocity, namely, damped response, divergent response, and limit cycle oscillation (LCO). Figure 19 depicts four instantaneous snapshots in a circle of LCO and their phase angles correspond to $\omega = 0, 90°, 180°, 270°$. Forced by shocks moving along the surface of the airfoil, the structure pitches and plunges in a periodic process.

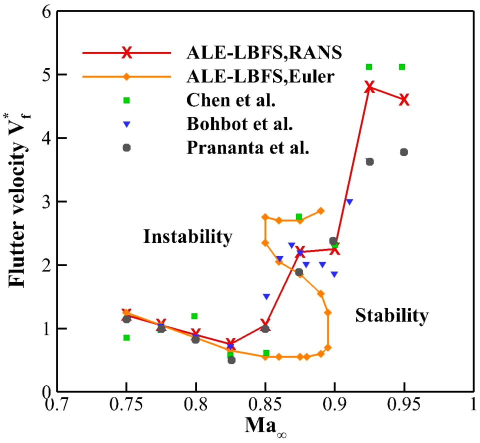
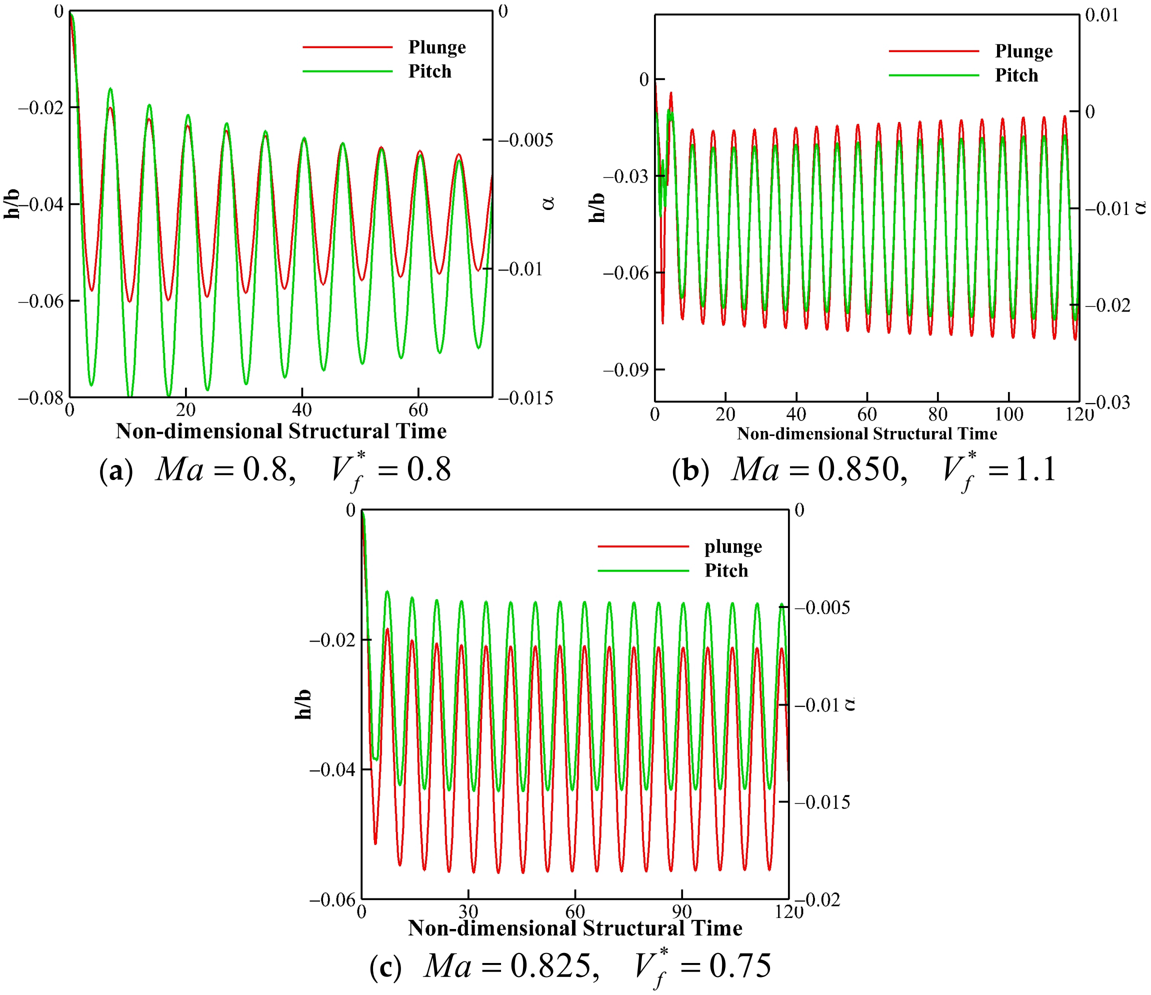


Figure 17. Flutter speed index V_f^* as a function of a free-stream Mach number for NACA 64A010.



(**a**) $Ma = 0.8, \quad V_f^* = 0.8$

(**b**) $Ma = 0.850, \quad V_f^* = 1.1$

(**c**) $Ma = 0.825, \quad V_f^* = 0.75$

Figure 18. History of plunge and pitch for (**a**) damped response; (**b**) divergent response; (**c**) limit cycle oscillation (LCO).

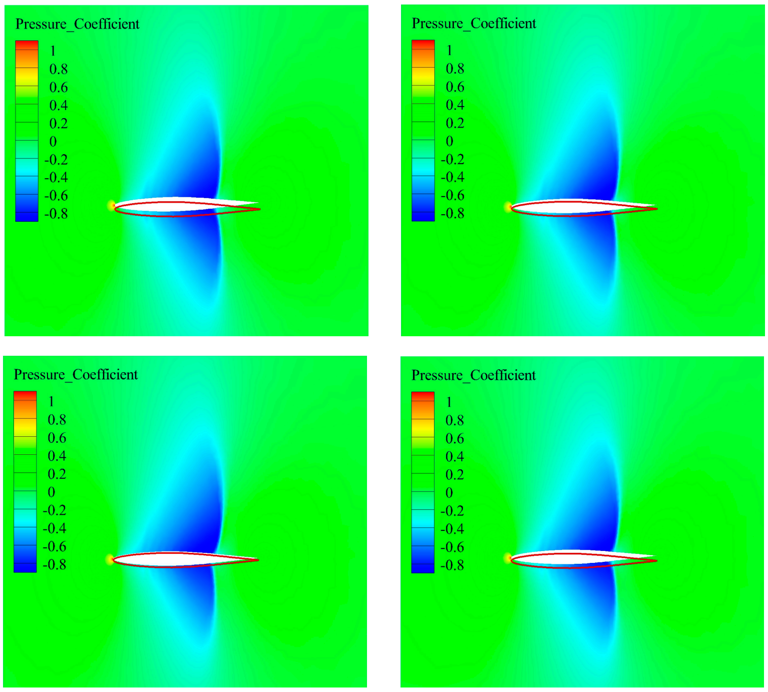


Figure 19. Pressure coefficient contours at four snapshots in a circle of LCO when Ma = 0.85. The red solid line represents the initial contour of the airfoil.

3.2.2. Flutter of the AGARG 445.6 Wing

AGARD 445.6 is a classical wing model for flutter analysis. Yates et al. [62] has conducted mode analysis on AGARD 445.6 for two kinds of models. This section adapts the AGARD 445.6 model as the object for studying. Figure 20 depicts the geometry of the AGARD 445.6 wing and its airfoil. This wing takes NACA 65A004 as its airfoil and extends it with no twist or curvature along its quarter-chord sweep angle (45°). The corresponding tip and root chords are 0.559 m and 0.368 m, respectively, while the semi-span is 0.762 m. The material of the wing used in Yates' experiment is laminated mahogany, which has distinguished physical properties in different directions. Based on the experimental data in Yates' work and previous works by Silva [63], the material properties of the wing model, such as elasticity modulus E, Poisson's ratio ν, shear modulus G and density ρ, are summarized and listed in Table 2.

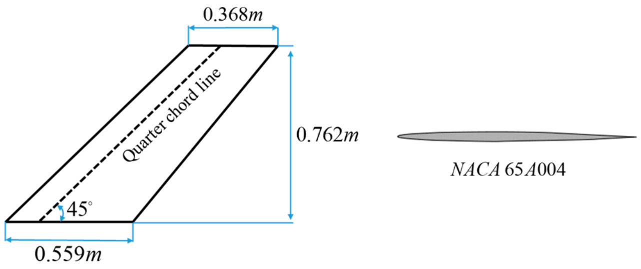


Figure 20. Geometry of AGARD 445.6 and its NACA 65A004 airfoil.

Table 2. The material properties in each direction for AGARD 445.6 wing.

Material Properties	Value
E_{11}	3.2456144 GPa
$E_{22}E_{33}$	0.416 GPa
$\nu_{12}\nu_{23}\nu_{23}$	0.31
$G_{12}G_{23}G_{12}$	0.411942 GPa
ρ	$381.98\ \mathrm{kg/m^3}$

When it comes to solving the structural dynamic response, it is first necessary to extract its eigenvalues and eigenvectors as described in Equation (38). With the aid of finite element structural analysis software Patran/Nastran, mode analysis is conducted on models of the wing entity. Several simplifications of the structural model, shell model, and plate with constant thickness are analyzed as well. Their meshes are exhibited in Figure 21. Table 3 compares the first six mode frequencies with numerical and experimental results obtained by Yates. It suggests that contrasted with the experiment, a good agreement for the wing entity is obtained. Due to the same principle assumption involved in the analysis of shell and thin plate, they perform similarly but have differences in the 3rd, 5th, and 6th modes in comparison with results obtained from experiments and numerical simulations. Therefore, the eigenvalues and eigenvectors with the first six modes for the wing entity are exploited in the following flutter analysis. Figure 22 shows the first four mode shapes along with their deformation depicted by black lines.

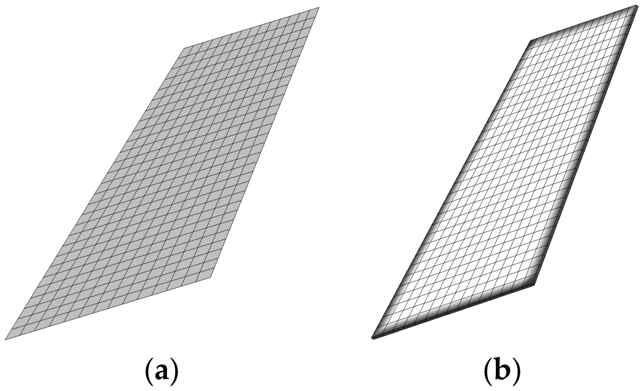


Figure 21. Mesh for (**a**) shell model and (**b**) plate model with constant thickness.

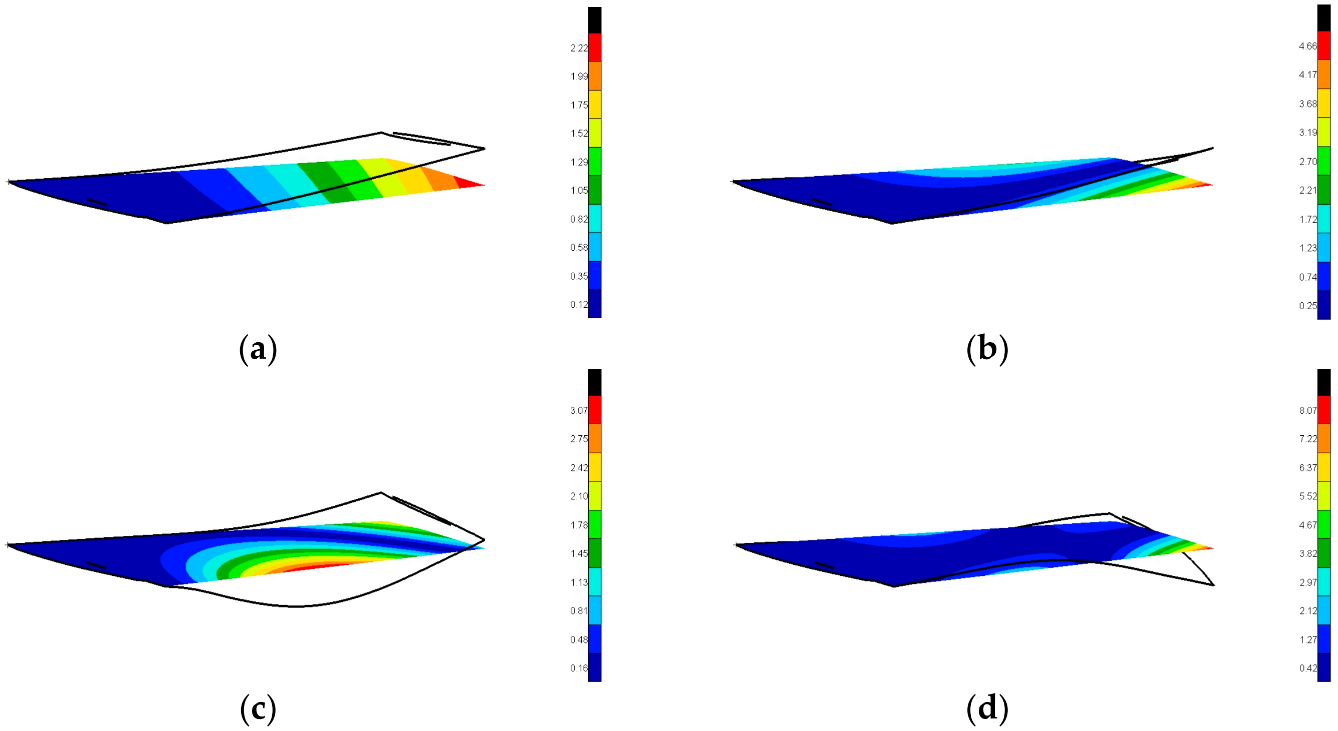


Figure 22. The contour of mode shape and deformation diagram for the first four mode frequencies. (**a**) $f_1 = 9.6224$. (**b**) $f_2 = 40.925$. (**c**) $f_3 = 49.863$. (**d**) $f_4 = 98.796$.

Then, tight coupling connects the fluid and structural solver. Within each physical time iteration, the fluid and structural solver interact with each other for twenty inner time iterations. As soon as the inner iterations are up to twenty or the residual is beneath the tolerance, the loop ends and marches physical time. Second-order dual time-stepping with 10 internal iterations is responsible for time discretization in the simulation of the flow field. In advance of flutter taking place, steady numerical simulation is solved with time step $\Delta t = 0.001$ and angle of attack $\alpha = 0$. Then, this perturbation of the steady-state solution is used as an initial condition for the unsteady state.

Table 3. Comparison of mode frequency.

Modes	Experiment	Numerical	Wing Entity	Shell	Plate with Thickness
1	9.6	9.5992	9.6224	9.5733	9.6
2	38.1	38.1650	40.925	35.736	35.778
3	50.7	48.3482	49.863	58.446	58.515
4	98.5	91.5448	98.796	97.663	97.724
5	/	118.1132	124.37	159.65	159.42
6	/	140.200	147.55	161.96	164.28

In the definition of flutter velocity V_f^*, b is the length of the semi-root chord and ω_α is the first torsional frequency of the wing. The time history of generalized displacements is an indicator to determine the flutter boundary. The same as the flutter analysis in NACA64A010, only if V_f^* approaches the critical point, the generalized displacements are neutral. When V_f^* exceeds the critical point, the generalized displacements diverge. Otherwise, they behave with a damped response.

Figure 23 shows the divergent responses of the first three generalized displacements corresponding to $Ma = 0.96$ and $V_f^* = 0.292$. Through Equation (39), its displacements in the x-, y-, and z-directions are easy to compute as shown in Figure 24. It describes the displacements at the probe of the leading edge and trailing edge, which are drawn in Figure 16b. The displacements in the x- and y-direction can be ignored compared with those in the z-direction, though all of them are in divergence. At the same time, the distinction of z-direction amplitudes between the leading edge and the trailing edge reflects that the torsional motion exists. Judging from the dynamic motion of z-direction displacement, its frequency is 14.5Hz, computed by the fast Fourier transform, which approaches the first-order bending. In other words, current flutter, dominated by bending deformation, is caused by the first-order bending mode.

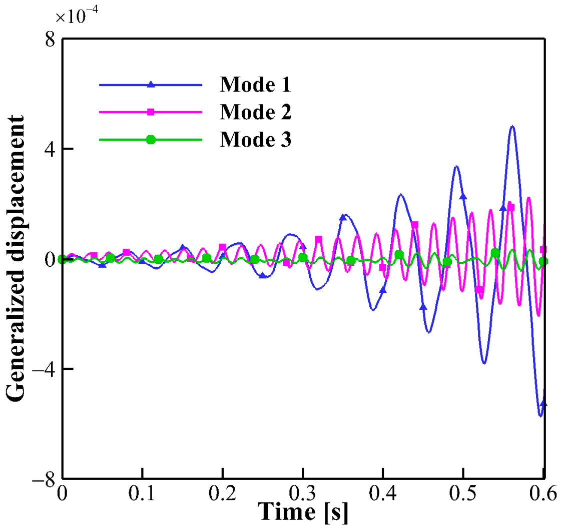


Figure 23. Generalized displacement for the first three modes for Ma = 0.96 and $V_f^* = 0.292$.

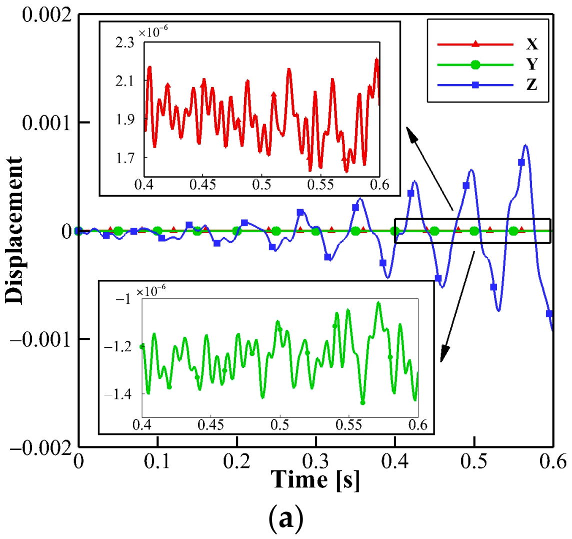


Figure 24. Displacement in x-, y-, z-directions at the probe of (**a**) the leading edge and (**b**) the trailing edge for Ma = 0.96 and $V_f^* = 0.292$.

In order to depict the flutter boundary along a Mach number, it is necessary to run several simulations for the specified Mach number until the neutral response is achieved. The practice of identifying critical onset flutter is similar to the studies performed by Chawdhury where a detailed flutter analysis is conducted with various wind velocities [8,9]. Here, flutter velocity V_f^* varies through the fixed value of density and manual adjustment of pressure. After repeating the aforementioned process for different Mach numbers, the flutter boundary is obtained, as shown in Figure 25. It compares the present result with the work of others [62,64–66], including experimental and numerical data. It is found that a good agreement with Yates' experiment is obtained.

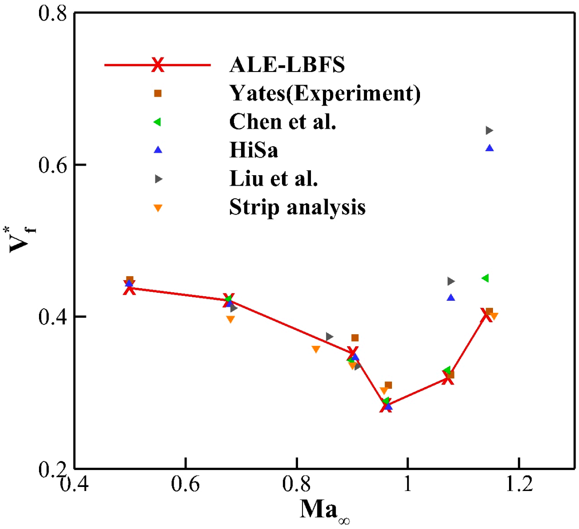


Figure 25. Flutter speed index V_f^* as a function of a free-stream Mach number for AGARD 445.6 wing.

In the subsonic flow, it is considered that the classical bending-torsion coupling flutter takes place due to the inertia coupling between the bending and torsion of the elastic wing. The additional aerodynamic forces yielding from flexural and torsional deformation will excite the wing. The wing becomes an energy conversion switch, which converts the energy of the incoming flow into energy with complex vibration properties, resulting in the occurrence of wing flutter. In the initial stage shown in Figure 26, there exists a slight phase difference in the z-direction displacement between the leading edge and the trailing edge. It indicates that the torsional deformation invites the flutter. Then, the phases are synchronized by aerodynamic instability in a hurry. The mode of flutter is transited into bending-torsion coupling vibration.

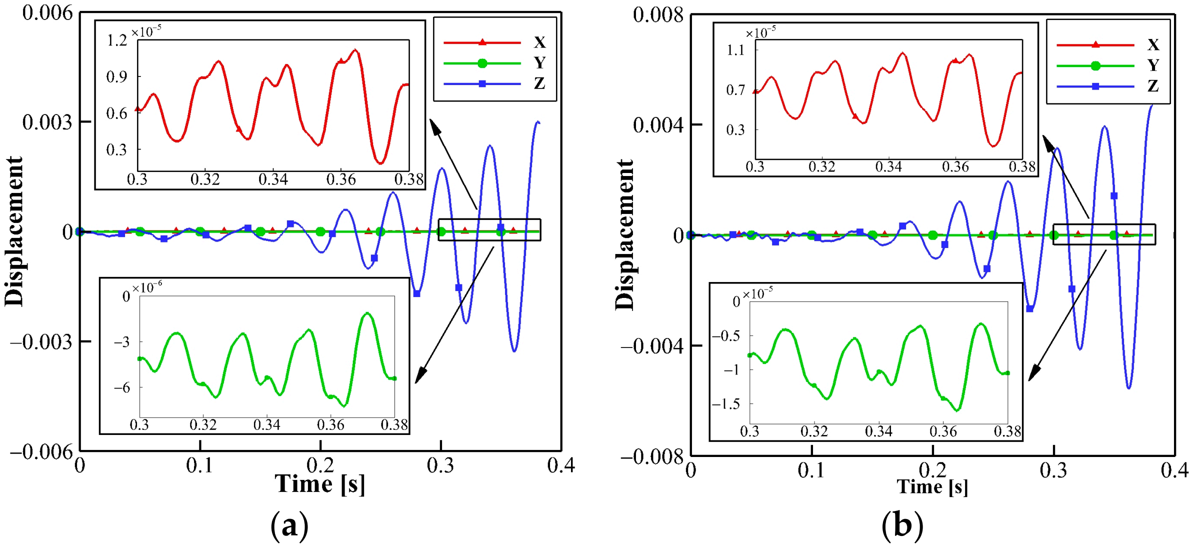


Figure 26. Displacement in x-, y-, z-directions at the probe of (**a**) the leading edge and (**b**) the trailing edge for Ma = 0.499 and $V_f^* = 0.44$.

The transonic dip occurs in the region of Ma = 0.9–1.1. For a Mach number $Ma = 0.96$, the flutter velocity V_f^* is equal to 0.283. In fact, with the Mach number falling into the transonic region, unstable shock waves occur and move back and forth on the surface of the wing, further inducing a changeable aerodynamic force. Therefore, it is inferred that the complex flow contributes to the dip of the flutter boundary as depicted in Figure 25. Meanwhile, for supersonic situations, the predicted flutter velocity is prone to the experiment results obtained by Yates. It is clarified that the obvious discrepancy from the numerical results obtained by others is due to the viscous effect.

4. Conclusions

In this paper, a tightly coupled ALE-LBFS with the mode decomposition method is presented to predict the 2D and 3D flutter phenomenon in compressible flows. The vertex-centered ALE-LBFS is applied to the unstructured mesh to solve for the flow field and the mode decomposition method is applied for the solution of structural deformations. The exchange of information between the flow field and solid field at the interface is carried out by using the RBF interpolation. A tightly coupled temporal scheme is used to advance the governing equations of the system. The performances of the proposed method are summarized as follows.

(a) The ALE-LBFS on unstructured dynamic mesh achieves good accuracy in the cases of the 2D forced pitching of the NACA0012 airfoil, the 3D uCRM-13.5 wing, and the BSCW wing with harmonic motions. When tightly coupled with a structure solver, it is qualified to solve aeroelastic problems, regardless of whether the dynamic of the structure is self-executed or controlled manually. The dynamic changes in pressure coefficient distribution C_p around the wing surface are obtained. Compared with the published results, it achieves good agreement.

(b) The ALE-LBFS coupled with the mode decomposition method has been successfully predicting the flutter phenomenon of the 2D airfoils and 3D wings. The so-called transonic dip captured the critical flutter velocities at different Mach numbers compared with the published data. The damped response, divergent response, and limit cycle oscillation (LCO) are demonstrated as well.

Witnessing the satisfactory performance, it seems that the ALE-LBFS coupled with the mode decomposition method presented in this work has a high potential and can be used as a powerful alternative to analyze flutter with more complex flow situations and structural configurations. It is a reliable and accurate numerical tool and lays the methodical and technical foundation for active flutter control as well.

Author Contributions: Conceptualization, Y.W. and T.G.; methodology, Y.W., T.G. and F.W.; software, Y.W. and T.G.; validation, Y.W. and T.G.; formal analysis, Y.W. and T.G.; investigation, T.G.; resources, Y.W. and T.G.; data curation, Y.W. and T.G.; writing—original draft preparation, T.G.; writing—review and editing, Y.W. and F.W.; visualization, Y.W. and T.G.; supervision, Y.W.; funding acquisition, Y.W. and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant Nos. 12272178), Aeronautical Science Foundation of China (Grant No. 20220012052004), Key Laboratory of Computational Aerodynamics, AVIC Aerodynamics Research Institute, and the Spring Sunshine Program of the Ministry of Education of China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

Acknowledgments: This work is also partially supported by the High-Performance Computing Platform of Nanjing University of Aeronautics and Astronautics.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Karnick, P.T.; Venkatraman, K. Shock–boundary layer interaction and energetics in transonic flutter. *J. Fluid Mech.* **2017**, *832*, 212–240. [CrossRef]
2. Bendiksen, O.O. Nonlinear mode interactions and period-tripling flutter in transonic flow. *J. Fluids Struct.* **2004**, *19*, 591–606. [CrossRef]
3. Schewe, G.; Mai, H. Experiments on transonic limit-cycle-flutter of a flexible swept wing. *J. Fluids Struct.* **2019**, *84*, 153–170. [CrossRef]
4. Schewe, G.; Mai, H.; Dietz, G. Nonlinear effects in transonic flutter with emphasis on manifestations of limit cycle oscillations. *J. Fluids Struct.* **2003**, *18*, 3–22. [CrossRef]
5. Ilie, M. A fully-coupled CFD/CSD computational approach for aeroelastic studies of helicopter blade-vortex interaction. *Appl. Math. Comput.* **2019**, *347*, 122–142. [CrossRef]
6. Huang, C.; Huang, J.; Song, X.; Zheng, G. Aeroelastic Simulation Using CFD/CSD Coupling Based on Precise Integration Method. *Int. J. Aeronaut. Space Sci.* **2020**, *21*, 750–767. [CrossRef]
7. Klimchenko, V.; Baeder, J.D. CFD/CSD Study of Interactional Aerodynamics of a Coaxial Compound Helicopter in High-Speed Forward Flight. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2020. [CrossRef]
8. Chawdhury, S.; Morgenthal, G. Numerical simulations of aeroelastic instabilities to optimize the performance of flutter-based electromagnetic energy harvesters. *J. Intell. Mater. Syst. Struct.* **2018**, *29*, 479–495. [CrossRef]
9. Chawdhury, S.; Morgenthal, G. A partitioned solver to simulate large-displacement fluid–structure interaction of thin plate systems for vibration energy harvesting. *Comput. Struct.* **2019**, *224*, 106110. [CrossRef]
10. Hirt, C.W.; Amsden, A.A.; Cook, J.L. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.* **1974**, *14*, 227–253. [CrossRef]
11. Su, X.; Cao, Y.; Zhao, Y. An unstructured mesh arbitrary Lagrangian-Eulerian unsteady incompressible flow solver and its application to insect flight aerodynamics. *Phys. Fluids* **2016**, *28*, 061901. [CrossRef]
12. Jin, P.; Deng, X.; Xiao, F. An ALE formulation for compressible flows based on multi-moment finite volume method. *Eng. Appl. Comput. Fluid Mech.* **2018**, *12*, 791–809. [CrossRef]
13. Lee-Rausch, E.M.; Batina, J.T. Wing flutter boundary prediction using unsteady Euler aerodynamic method. *J. Aircr.* **1995**, *32*, 416–422. [CrossRef]
14. Lee-Rausch, E.M.; Batina, J.T. Wing flutter computations using an aerodynamic model based on the Navier-Stokes equations. *J. Aircr.* **1996**, *33*, 1139–1147. [CrossRef]

15. Yuan, W.; Sandhu, R.; Poirel, D. Fully Coupled Aeroelastic Analyses of Wing Flutter towards Application to Complex Aircraft Configurations. *J. Aerosp. Eng.* **2021**, *34*, 04020117. [CrossRef]
16. Simiriotis, N.; Palacios, R. A flutter prediction framework in the open-source SU2 suite. In Proceedings of the AIAA Scitech 2022 Forum, San Diego, CA, USA, 3–7 January 2022; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2022. [CrossRef]
17. Wang, Y.-R.; Wang, Y.-J. Flutter speed prediction by using deep learning. *Adv. Mech. Eng.* **2021**, *13*, 16878140211062275. [CrossRef]
18. Yang, Z.; Huang, R.; Liu, H.; Zhao, Y.; Hu, H. An improved nonlinear reduced-order modeling for transonic aeroelastic systems. *J. Fluids Struct.* **2020**, *94*, 102926. [CrossRef]
19. Lowe, B.; Zingg, D. Flutter Prediction using Reduced-Order Modeling. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2020. [CrossRef]
20. Bombardieri, R.; Cavallaro, R.; de Teresa, J.L.S.; Karpel, M. Nonlinear Aeroelasticity: A CFD-based Adaptive Methodology for Flutter Prediction. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019. [CrossRef]
21. Chen, S.; Doolen, G.D. Lattice boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.* **1998**, *30*, 329–364. [CrossRef]
22. Guo, Z.; Shu, C. *Lattice Boltzmann Method and Its Application in Engineering*; World Scientific: Singapore, 2013.
23. Aidun, C.K.; Clausen, J.R. Lattice-Boltzmann Method for Complex Flows. *Annu. Rev. Fluid Mech.* **2010**, *42*, 439–472. [CrossRef]
24. Inamuro, T.; Ogata, T.; Tajima, S.; Konishi, N. A lattice Boltzmann method for incompressible two-phase flows with large density differences. *J. Comput. Phys.* **2004**, *198*, 628–644. [CrossRef]
25. Lim, C.Y.; Shu, C.; Niu, X.D.; Chew, Y.T. Application of lattice Boltzmann method to simulate microchannel flows. *Phys. Fluids* **2002**, *14*, 2299–2308. [CrossRef]
26. Wang, Y.; Shu, C.; Teo, C.J. Thermal lattice Boltzmann flux solver and its application for simulation of incompressible thermal flows. *Comput. Fluids* **2014**, *94*, 98–111. [CrossRef]
27. Yu, H.; Girimaji, S.S.; Luo, L.-S. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *J. Comput. Phys.* **2005**, *209*, 599–616. [CrossRef]
28. Bhadauria, A.; Dorschner, B.; Karlin, I. Lattice Boltzmann method for fluid–structure interaction in compressible flow. *Phys. Fluids* **2021**, *33*, 106111. [CrossRef]
29. Cheylan, I.; Favier, J.; Sagaut, P. Immersed boundary conditions for moving objects in turbulent flows with the lattice-Boltzmann method. *Phys. Fluids* **2021**, *33*, 095101. [CrossRef]
30. Saadat, M.H.; Karlin, I.V. Arbitrary Lagrangian–Eulerian formulation of lattice Boltzmann model for compressible flows on unstructured moving meshes. *Phys. Fluids* **2020**, *32*, 046105. [CrossRef]
31. Meldi, M.; Vergnault, E.; Sagaut, P. An arbitrary Lagrangian–Eulerian approach for the simulation of immersed moving solids with Lattice Boltzmann Method. *J. Comput. Phys.* **2013**, *235*, 182–198. [CrossRef]
32. Shu, C.; Wang, Y.; Yang, L.M.; Wu, J. Lattice Boltzmann Flux Solver: An Efficient Approach for Numerical Simulation of Fluid Flows. *Trans. Nanjing Univ. Aeronaut. Astronaut.* **2014**, *31*, 1–15.
33. Shu, C.; Wang, Y.; Teo, C.J.; Wu, J. Development of Lattice Boltzmann Flux Solver for Simulation of Incompressible Flows. *Adv. Appl. Math. Mech.* **2014**, *6*, 436–460. [CrossRef]
34. Wang, Y.; Yang, L.; Shu, C. From Lattice Boltzmann Method to Lattice Boltzmann Flux Solver. *Entropy* **2015**, *17*, 7713–7735. [CrossRef]
35. Liu, Y.Y.; Shu, C.; Zhang, H.W.; Yang, L.M. A high order least square-based finite difference-finite volume method with lattice Boltzmann flux solver for simulation of incompressible flows on unstructured grids. *J. Comput. Phys.* **2020**, *401*, 109019. [CrossRef]
36. Walsh, B.; Boyle, F.J. A preconditioned lattice Boltzmann flux solver for steady flows on unstructured hexahedral grids. *Comput. Fluids* **2020**, *210*, 104634. [CrossRef]
37. Wu, Q.-F.; Shu, C.; Wang, Y.; Yang, L.-M. An effective lattice Boltzmann flux solver on arbitrarily unstructured meshes. *Mod. Phys. Lett. B* **2018**, *32*, 1840012. [CrossRef]
38. Li, W.; Luo, L.-S. An implicit block LU-SGS finite-volume lattice-Boltzmann scheme for steady flows on arbitrary unstructured meshes. *J. Comput. Phys.* **2016**, *327*, 503–518. [CrossRef]
39. Yang, L.; Shu, C.; Chen, Z.; Wang, Y.; Hou, G. A simplified lattice Boltzmann flux solver for multiphase flows with large density ratio. *Int. J. Numer. Methods Fluids* **2021**, *93*, 1895–1912. [CrossRef]
40. Wang, Y.; Shu, C.; Teo, C.J.; Wu, J.; Yang, L. Three-Dimensional Lattice Boltzmann Flux Solver and Its Applications to Incompressible Isothermal and Thermal Flows. *Commun. Comput. Phys.* **2015**, *18*, 593–620. [CrossRef]
41. Yang, L.M.; Shu, C.; Wu, J. A Hybrid Lattice Boltzmann Flux Solver for Simulation of Viscous Compressible Flows. *Adv. Appl. Math. Mech.* **2016**, *8*, 887–910. [CrossRef]
42. Chen, J.; Yang, D.; Chen, Q.; Sun, J.; Wang, Y. A rotated lattice Boltzmann flux solver with improved stability for the simulation of compressible flows with intense shock waves at high Mach number. *Comput. Math. Appl.* **2023**, *132*, 18–31. [CrossRef]
43. Wang, Y.; Shu, C.; Teo, C.J.; Wu, J. An immersed boundary-lattice Boltzmann flux solver and its applications to fluid–structure interaction problems. *J. Fluids Struct.* **2015**, *54*, 440–465. [CrossRef]
44. Wang, Y.; Shu, C.; Yang, L.M.; Sun, Y. On the immersed boundary-lattice Boltzmann simulations of incompressible flows with freely moving objects. *Int. J. Numer. Methods Fluids* **2017**, *83*, 331–350. [CrossRef]
45. Wang, Y.; Shu, C.; Teo, C.J.; Yang, L.M. Numerical study on the freely falling plate: Effects of density ratio and thickness-to-length ratio. *Phys. Fluids* **2016**, *28*, 103603. [CrossRef]

46. Liu, F.; Liu, G.; Shu, C. Fluid–structure interaction simulation based on immersed boundary-lattice Boltzmann flux solver and absolute nodal coordinate formula. *Phys. Fluids* **2020**, *32*, 047109. [CrossRef]
47. Yang, L.M.; Shu, C.; Wu, J. A moment conservation-based non-free parameter compressible lattice Boltzmann model and its application for flux evaluation at cell interface. *Comput. Fluids* **2013**, *79*, 190–199. [CrossRef]
48. Biedron, R.T.; Thomas, J.L.; Thomas, J.L. Recent Enhancements To The FUN3D Flow Solver For Moving-Mesh Applications. In Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL, USA, 5–8 January 2009; Available online: https://ntrs.nasa.gov/citations/20090007605 (accessed on 18 July 2022).
49. Farhat, C.; Geuzaine, P.; Brown, G. Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Comput. Fluids* **2003**, *32*, 3–29. [CrossRef]
50. Thomas, P.D.; Lombard, C.K. Geometric Conservation Law and Its Application to Flow Computations on Moving Grids. *AIAA J.* **1979**, *17*, 1030–1037. [CrossRef]
51. Yang, X.; Cheng, S.; Yang, A.; Sun, G. Spectral Method for Numerical Simulation of Unsteady Viscous Flow over Oscillating Airfoil and Wing. *ACTA Aeronaut. Astronaut. Sin.* **2013**, *34*, 787. [CrossRef]
52. Brooks, T.R.; Kenway, G.K.W.; Martins, J.R.R.A. Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings. *AIAA J.* **2018**, *56*, 2840–2855. [CrossRef]
53. Crow, B.T.; Nagawkar, J.R.; Leifsson, L.T.; Thelen, A.S. Development of an Open-source Flutter Prediction Framework for the Common Research Model Wing. In Proceedings of the AIAA Scitech 2021 Forum, Virtual Event, 11–15, 19–21 January 2021; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2021. [CrossRef]
54. Brehm, C.; Barad, M.F.; Kiris, C.C. An Immersed Boundary Method for Solving the Compressible Navier-Stokes Equations with Fluid-Structure Interaction. In Proceedings of the 34th AIAA Applied Aerodynamics Conference, Washington, DC, USA, 13–17 June 2016; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2016. [CrossRef]
55. Sanchez, R.; Kline, H.L.; Thomas, D.; Variyar, A.; Righi, M.; Economon, T.D.; Alonso, J.J.; Palacios, R.; Dimitriadis, G.; Terrapon, V. Assessment of the fluid-structure interaction capabilities for aeronautical applications of the open-source solver SU2. In Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2016), Crete Island, Greece, 5–10 June 2016; Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA) Greece: Athens, Greece, 2016; pp. 1498–1529. [CrossRef]
56. Alonso, J.; Jameson, A. Fully-implicit time-marching aeroelastic solutions. In Proceedings of the 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 10–13 January 1994; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 1994. [CrossRef]
57. Chen, X.; Zha, G.; Hu, Z.; Yang, M.-T. Flutter Prediction Based on Fully Coupled Fluid-Structural Interactions. In Proceedings of the 9th National Turbine Engine High Cycle Fatigue Conference, Pinehurst, NC, USA, 16–19 March 2004; pp. 16–19.
58. Sanchez, R.; Palacios, R.; Economon, T.D.; Kline, H.L.; Alonso, J.J.; Palacios, F. Towards a Fluid-Structure Interaction Solver for Problems with Large Deformations within the Open-Source SU2 Suite. In Proceedings of the 57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, San Diego, CA, USA, 4–8 January 2016; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2016. [CrossRef]
59. Chen, X.; Zha, G.-C.; Hu, Z. Numerical Simulation of Flow Induced Vibration Based on Fully Coupled Fluid-Structural Interactions. In Proceedings of the 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, OR, USA, 28 June–1 July 2004; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2004. [CrossRef]
60. Bohbot, J. Time Domain Analysis of Two DOF Airfoil Flutter Using an Euler/Turbulent Navier-Stokes Implicit Solver. In Proceedings of the International Forum on Aeroelasticity and Structural Dynamics, Madrid, Spain, 1–5 January 2001; pp. 75–86. Available online: https://cir.nii.ac.jp/crid/1570009750059002496 (accessed on 11 June 2023).
61. Blom, F.J.; Leyland, P. Analysis of Fluid-Structure Interaction by Means of Dynamic Unstructured Meshes. *J. Fluids Eng.* **1998**, *120*, 792–798. [CrossRef]
62. Yates, E.C. *AGARD Standard Aeroelastic Configurations for Dynamic Response. Candidate Configuration I.-Wing 445.6*; Tech Rep AGARD Report No.765; AGARD: Neuilly sur Seine, France, 1987.
63. Silva, P.A.S.F.; de Morais, M.V.G. Fluid Structure Interaction on AGARD 445.6 wing at Mach 0.9. *Rev. Interdiscip. Pesqui. Eng.* **2016**, *2*, 224–235. [CrossRef]
64. Liu, F.; Cai, J.; Zhu, Y.; Tsai, H.M.; Wong, A.S.F. Calculation of Wing Flutter by a Coupled Fluid-Structure Method. *J. Aircr.* **2001**, *38*, 334–342. [CrossRef]
65. Chen, X.; Zha, G.-C.; Yang, M.-T. Numerical simulation of 3-D wing flutter with fully coupled fluid–structural interaction. *Comput. Fluids* **2007**, *36*, 856–867. [CrossRef]
66. Pahlavanloo, P. *Dynamic Aeroelastic Simulation of the AGARD 445.6 Wing Using Edge*; Defence and Security, Systems and Technology, Swedish Defence Research Agency (FOI): Stockholm, Sweden, 2007.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.