A Coupled Newton-Krylov Time Spectral Solver for Flutter Prediction

Sicheng He,* Eirikur Jonsson,†
Charles A. Mader,‡ Joaquim R. R. A. Martins§
University of Michigan, Department of Aerospace Engineering, Ann Arbor, MI

Flutter methods can be broadly categorized into low and high fidelity methods. Low fidelity methods in general lack the ability to capture accurately the strong nonlinearities in the transonic flow regime; researchers have used Harmonic Balance (HB) method (a frequency domain method equivalent with time-spectral (TS) method modeled in time domain) to capture the flutter onset point. However, the previous solution method is hindered by scalability: $O(N_{\text{CSD}})$ CFD evaluations needed for each Newton step, and a lack of robustness due to the nature of Newton–Raphson method. In this paper, we propose a preconditioned coupled Jacobian-free Newton–Krylov (CNK) method for TS flutter equations. By resolving the coupled system directly, the $O(N_{\text{CSD}})$ CFD evaluations are avoided. With a line search method CNK is expected to be more robust. A block-Jacobian preconditioning method is used in this work. CNK is compared with and validated by time-accurate (TA) flutter boundary results from both literature and our new implementation.

I. Introduction

High-fidelity computational modeling and optimization of complex engineering systems has the potential to allow engineers to produce more efficient designs and to reduce the occurrence of unforeseen late stage design modifications. In particular, for transonic wing design, the simultaneous optimization of both the aerodynamic design and the internal structural sizing can yield significant fuel burn savings. However, when conducting optimization on an aircraft, all of the relevant physics must be represented in the optimization problem, otherwise the result generated by the optimization procedure may not be meaningful or physical. As an example, previous optimization results obtained by Kenway et al. [1, 2, 3], without flutter constraints produced optimized wings with large aspect ratio as shown in Fig. 1. Such configurations are prone to flutter, which calls into question the usefulness of the result. Therefore, in order to generate more reasonable designs, we seek to compute the flutter characteristics of the aircraft. The main focus of the paper is to propose an efficient and robust solution methodology for finding the flutter onset. This method is a preconditioned, Jacobian-free, coupled Newton–Krylov method. It directly deals with all CFD, CSD, flutter velocity index V_f and flutter frequency variables without black box computation. In one coupled Newton solution procedure, the residual of the coupled system is driven to zero and the flutter velocity index V_f is found. In order to demonstrate the validity of the proposed approach, we compare the results of a simple two degree of freedom airfoil case with both time-accurate computational fluid dynamics (CFD) solutions and experimental results.

II. Background

Since flutter is a certification-critical phenomenon, it is important to be able to predict it accurately. Conservative design approaches may lead to overly-stiff and hence high mass designs, while unconstrained optimization approaches, such as those shown in Fig. 1 may lead to overly-flexible wings that may cause problems when certifying the aircraft. Further, it is currently not unusual for flutter issues to be identified only at the final design and flight testing stages, at which point design changes are extremly costly. Accurate flutter prediction methods will therefore lead directly to significant cost savings.

The current standard for flutter prediction in industry are methods based on panel codes, doublet-lattice method (DLM) [4], and linearized transonic small disturbance (TSD) equations [5, 6]. The CAP-TSD code developed for analysis of flutter of realistic aircrafts in transonic flow uses a time-accurate approximate factorization (AF) algorithm for solution of the unsteady transonic small-disturbance potential equation. Batina et al. [7, 5] used the AF method

*PhD Candidate, AIAA student member
†PhD Candidtae, AIAA student member
‡Research Investigator, AIAA Senior Member
§Professor, AIAA Associate Fellow

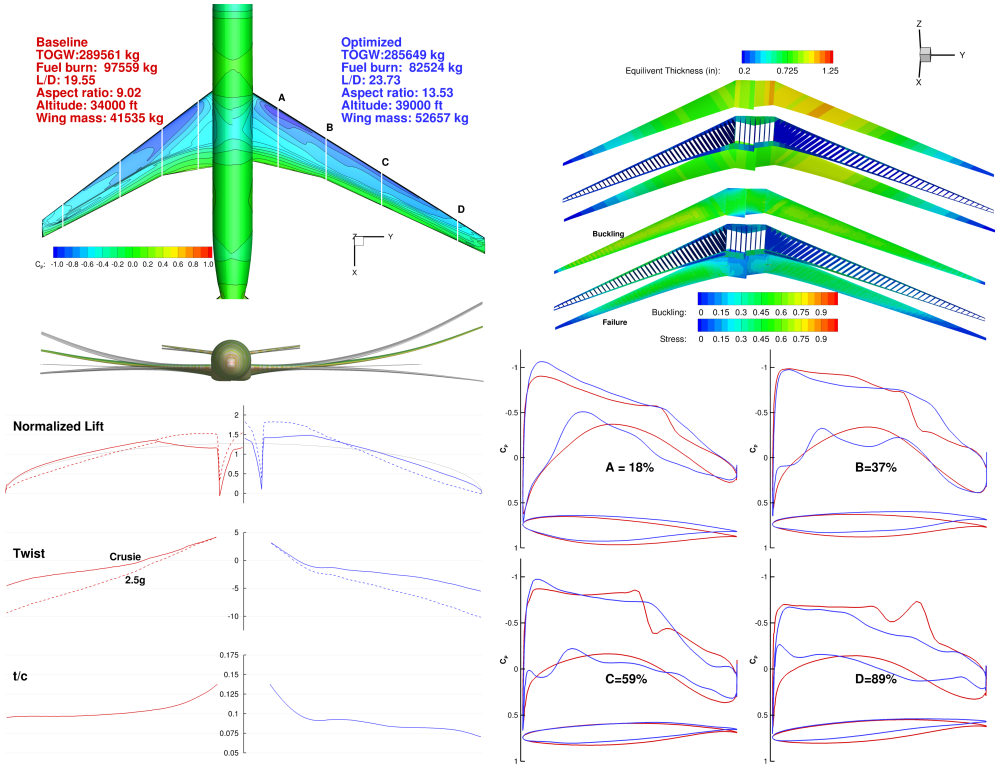


Figure 1. Aerostructural optimization result [3]: C_p and planform comparison with initial design (upper left); equivalent thickness distribution, stress and buckling KS failure criteria (upper right); comparison of initial and optimized lift distributions, twist distributions and thickness to chord ratio (t/c) (lower left); four airfoils with corresponding C_p distributions (lower right). (notice the increased span ratio)

which involves a Newton linearization, coupled with an internal iteration technique. Using this approach a solution is obtained in several hundred time steps. The DLM has gained remarkable success and is found in commercial software products [8, 9] and has become the aeroelasticians method of choice in industry.

One major limit of some of those low-fidelity linear aerodynamic methods is that they are unable to predict the occurrence of shock waves in the transonic flow. A consequence of this is that the prediction of the flutter boundaries can become inaccurate. In the transonic region there is a significant reduction in the flutter speed, called the *transonic dip (or flutter bucket)*. The bottom of the dip defines the minimum velocity at which flutter can occur across the flight envelope. Corrections using wind tunnel experimental data can however be applied to panel method aerodynamic influence coefficients (AIC). Unfortunately, for aerostructural design optimization this data is unavailable. Although other methods such as full potential flow or the Euler equations are able to predict shock waves in the flow, they still fail to accurately predict the flutter boundary. In order to predict onset of flutter accurately viscous effects are necessary as shown and summarized by several authors [10, 11, 6].

In the research community, the standard method for predicting a wing's flutter boundary is to analyze the wing with a time-accurate coupled CFD-CSD solver similar as proposed by Liu et al. [12]. However, these methods incur a high computational cost since hundreds if not thousands of time steps are often required to simulate the flutter motion. This high computational cost makes the full time-accurate method ill-suited to optimization. Like in many periodic problems, much of the computational time is spent resolving the decay of the initial transients in the unsteady problem [13]. Fortunately, in problems where the periodic steady state is of primary interest, time-periodic simulation methods such as the Harmonic Balance (HB) presented by Hall et al. [14], Time-Spectral by Gopinath and Jameson [15] or the Non-Linear Frequency Domain (NLFD) method by McMullen [13] can all be used to accelerate the solution process.

The first of these time-periodic methods was the Harmonic Balance method proposed by Hall et al. [14]. It was originally designed for rotating system, but has been demonstrated by Hall et al. [14] to work efficiently for computations of a variety of periodic systems. The basic idea of time-periodic methods is to represent all the state variables in the system with a Fourier series. This allows the time-dependent problem to be transformed into a series of coupled steady state problems. Several similar methods have been developed, including the nonlinear frequency domain method by McMullen [16] and the time-spectral method by Gopinath and Jameson [15]. While this class of

methods was originally developed in the frequency domain, more recent versions of these methods, such as the work of Gopinath et al. [15] have reformulated the spectral methods completely in the time domain.

Work has also been done to extend these spectral methods to solve aeroelastic problems. For example, Kachra and Nadarajah [17] extended the NLFD method to simulate a two dimensional model of a swept-wing in a loosely coupled manner. In this case they solve structure and aerodynamics equations separately and exchange the interface data for every N cycles. Choi and Datta [18] have also done time-spectral aeroelastic simulation of a three-dimensional helicopter rotor.

There also has been significant effort put to prediction of the flutter boundary using spectral methods [17, 16, 19]. Based on a physical intuition, Thomas et al. present both equations to capture flutter phenomenon and a solution methodology with Newton–Raphson method in [20] .With their method, the flutter velocity index is a state variable to be solved rather than an external parameter used as an input for a general unsteady solver. That gives the method an advantage of being automatic which is critical in MDO. However, the method is hindered by scalability: $O(N_{\text{CSD}})$ CFD evaluations are needed for each Newton step. The root cause of this limitation is that the CFD module is treated as a black box. In addition, the method is also lack of robustness due to the nature of Newton–Raphson method used to solve for the flutter velocity index.

To overcome these shortcomings, we propose a preconditioned, Jacobian-free, coupled Newton–Krylov method which resolves both CSD and CFD equations without any black box limitations. Since all the states are directly resolved, the $O(n_{\text{CSD}})$ CFD evaluations are gone, making the method more efficient. The use of Krylov iterative methods guarantees fast solution for each linear equation encountered in Newton iterations. W While the Jacobian-free nature of the method further cuts the computational and memory cost of the method by eliminating the need to evalutate and store the full Jacobian. The robustness of the solver is further augmented through the use of line search procedures implemented in PETSc [21]. The proposed method in this work is an extension of the steady aerostructural solution methodology implemented by Kenway et al. [22]. The main contribution of the paper is this solution methodology for identifying flutter onset efficiently and robustly.

III. Methodology

There are several techniques and components necessary to enable high-fidelity flutter computations. In the following section we outline the major characteristics of these components. Then we present the theory necessary for predicting flutter and finally we explain the solution methodology.

1. Aerodynamics Model: CFD Solver

The CFD code used for this work is ADflow (extended from SUmb [23]) a parallel, finite-volume, cell-centered, multi-block solver, which solves the Euler and the Reynolds averaged Navier-Stokes (RANS) equations in either steady, unsteady or time-spectral (TS) modes. For unsteady applications, the second-order implicit backwards difference formula (BDF2), time integration scheme is used. For deforming grids, an arbitrary Lagrangian Eulerian (ALE) formulation satisfying the Geometric Conservation Law (GCL)[24] is also implemented. In this work, we limit ourselves to Euler equations although present formulation can be directly applied using the RANS equations.

1.1. Time-spectral (TS) method for CFD

As we mentioned above, we use a TS method to accelerate the computation of the flutter boundaries. The TS method is well-established in CFD [15]. The method converts an unsteady CFD problem into a series of time-coupled steady state problems. The equations generated by this set of coupled, steady-state problems have two additional parameters: the largest time period considered T, and the number of time instances, or points n, to be solved within that period. If we write the time-dependent residual as $\mathcal{A}(\zeta(t)) = 0$, the TS form is $\mathcal{A}(\zeta^n, T) = 0$, where ζ^n represents the state variables for all time instances, and so its size is n times that of a steady-state solution. The residual form of TS CFD equations can be written in the following form, adapted from Gopinath and Jameson [15]:

$$\mathcal{A}_{\text{TS}} := D(\omega)\zeta^n + R(\zeta^n) = 0, \tag{1.1}$$

where $D(\omega)$ is a $n \times n$ matrix. It represents the temporal spectral interpolation of the time derivative term and is a function of the angular velocity ω. $R(\zeta^n)$ is the spatial residual term and ζ^n are the states from all time-instances. A mix of D3ADI and RK, approximate Newton method and Newton-Krylov method are used to drive the residual above to zero. This mix of solvers allows for robust start up and rapid terminal convergence of the flow solver.

To capture flutter onset point, we also need the flutter velocity index, V_f, as a variable. This affects the CFD equations through the boundary condition (or grid velocity: if the air is set to be at rest while the airfoil with the grid is in motion). For reasons mentioned in the following section, the grid nodal coordinates also affect the results. A more general form of the CFD equations can thus be written as:

$$\mathcal{A}_{\text{TS}}(V_f, \omega, \zeta^n, X_V^n) = 0, \tag{1.2}$$

where X_V^n is the grid volume nodal coordinates for all time-instances.

1.2. Time-spectral method for grid velocity calculation

For a dynamic mesh CFD solution, the relative velocity is needed for flux calculation:

$$V_x = v_x - v_{x,g}, \tag{1.3}$$
$$V_y = v_y - v_{y,g}, \tag{1.4}$$
$$V_z = v_z - v_{z,g}, \tag{1.5}$$

where V_x, V_y and V_z are relative velocities, v_x, v_y and v_z are the absolute velocities and $v_{x,g}, v_{y,g}$ and $v_{z,g}$ are the grid surface center velocities. We solve for the grid velocity by spectral interpolation. The grid motion can be approximated by a sum of harmonic functions (Only the x coordinate is shown here, but the y and z coordinates have the same form.):

$$x_g \approx \sum_{k=-(n-1)/2}^{(n-1)/2} \hat{x}_k e^{\frac{2\pi k i t}{T}}. \tag{1.6}$$

Here we assume that n, the total time-instance number, is odd. x_g is the x coordinate of some node from the grid. $\hat{x}_k$ are their Fourier series coefficients. Using the approach used for the approximation of the temporal derivative term in TS CFD [15], we have:

$$\dot{x}_g^n \approx D(\omega) x_g^n, \tag{1.7}$$

where $\dot{x}_g^n$ is the true grid nodal velocities for all time-instances and x_g^n are the grid nodal coordinates for all time-instances. The grid surface center velocities $v_{x,g}, v_{y,g}$ and $v_{z,g}$, are approximated by averaging of the nodal velocities. Although the geometric conservative law [24] is not enforced here, the numerical results in IV. Sec.5 validate this simplification for the flutter boundary calculation.

2. Structural Model: Two-dimensional Wing Section Model

In this work, we use the NACA 64A010 two-dimensional wing section model as described in [25]. A schematic of the configuration is shown in Fig. 2:

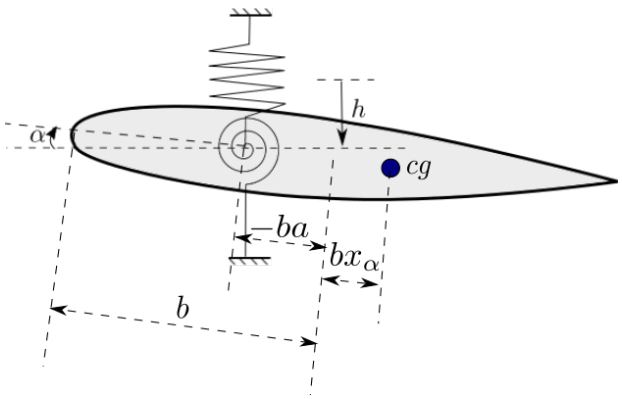


Figure 2. Typical section wing model

The computational structural dynamics (CSD) equation for this model is repeated here for completeness:

$$\begin{pmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{pmatrix} \begin{pmatrix} \frac{\ddot{h}}{b} \\ \ddot{\alpha} \end{pmatrix} + \begin{pmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{pmatrix} \begin{pmatrix} \frac{h}{b} \\ \alpha \end{pmatrix} = \frac{V_f^2}{\pi} \begin{pmatrix} -C_l \\ 2C_m \end{pmatrix}, \tag{2.1}$$

where x_α is dimensionless static unbalance; r_α is dimensionless section moment of inertia about the elastic axis, or the radius of gyration; ω_h, ω_α are uncoupled natural frequencies of typical section in plunge and pitch respectively; h/b is dimensionless plunging motion and α is the pitching motion; V_f is the flutter velocity index defined as $U_f/b\omega_\alpha\sqrt{\mu}$ where μ is airfoil mass ratio and U_f is the speed of main stream. Numerical values used in this study are given in Table 1. The elastic axis location is $a = -2$, indicating that it lies ahead of the airfoil.

Table 1. Airfoil structural properties.

Parameter		Value
Static unbalance	$x_\alpha = S_\alpha/mb$	1.8
Radius of gyration	$r_\alpha^2 = I_\alpha/mb^2$	3.48
Plunging natural frequency	ω_h	100.0
Pitching natural frequency	ω_α	100.0
Frequency ratio	ω_h/ω_α	1.0
Mass ratio	$\mu = m/\pi\rho_\infty b^2$	60

Table 1 equation can be succinctly written as:

$$M\ddot{u} + Ku = f, \tag{2.2}$$

where

$$M := \begin{pmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{pmatrix}, K := \begin{pmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{pmatrix}, u := \begin{pmatrix} \frac{h}{b} \\ \alpha \end{pmatrix}, f := \frac{V_f^2}{\pi}\begin{pmatrix} -C_l \\ 2C_m \end{pmatrix}. \tag{2.3}$$

2.1. Time-spectral method for structural dynamics

Besides the CFD, the TS method can also be applied to CSD. The key step is to apply the spectral interpolation to solve for the derivatives and permute the displacement history for the convenience of the spectral interpolation step. A 3 time-instances example is shown here and it is straight forward to extend to the case with a different number of time-instances. Consider,

$$\begin{pmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{pmatrix}\begin{pmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{pmatrix} + \begin{pmatrix} K & 0 & 0 \\ 0 & K & 0 \\ 0 & 0 & K \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{V_f^2}{\pi}\begin{pmatrix} \bar{f}_1 \\ \bar{f}_2 \\ \bar{f}_3 \end{pmatrix}, \tag{2.4}$$

where the subscript denotes the time instance (e.g. u_2 indicates displacement from the second instance) and $\bar{f}_i$ is defined as $(-C_{l,i}, 2C_{m,i})^T$ for the i^{th} time instance. Permute the history vector

$$Q\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = Q\begin{pmatrix} \frac{h_1}{b} \\ \alpha_1 \\ \frac{h_2}{b} \\ \alpha_2 \\ \frac{h_3}{b} \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \frac{h_1}{b} \\ \frac{h_2}{b} \\ \frac{h_3}{b} \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.5}$$

With the permutation matrix Q, we conduct spectral interpolation twice to get the second derivative (notice e.g. for the pitching mode: $\ddot{\alpha}^n \approx D(\omega)(\dot{\alpha}^n) \approx D(\omega)^2\alpha^n$), we have:

$$\begin{pmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{pmatrix} Q^T \begin{pmatrix} D(\omega)^2 & 0 \\ 0 & D(\omega)^2 \end{pmatrix} Q \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} K & 0 & 0 \\ 0 & K & 0 \\ 0 & 0 & K \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{V_f^2}{\pi}\begin{pmatrix} \bar{f}_1 \\ \bar{f}_2 \\ \bar{f}_3 \end{pmatrix}. \tag{2.6}$$

This is our time-spectral structural dynamics equation. We denote it as:

$$\mathcal{S}_{\text{TS}}(V_f, \omega, u^n, \bar{f}^n) := \boldsymbol{M}\boldsymbol{D}_Q(\omega)u^n + \boldsymbol{K}u^n - \frac{V_f^2}{\pi}\bar{f}^n = 0 \tag{2.7}$$

where $V_f^2/\pi \bar{f}^n$ denotes the load from CFD for all time instances, u^n denotes the displacement for all the time instances and

$$\boldsymbol{M} := \begin{pmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{pmatrix}, \quad \boldsymbol{D}_Q(\omega) := Q^T \begin{pmatrix} D(\omega)^2 & 0 \\ 0 & D(\omega)^2 \end{pmatrix} Q, \quad \boldsymbol{K} := \begin{pmatrix} K & 0 & 0 \\ 0 & K & 0 \\ 0 & 0 & K \end{pmatrix}. \tag{2.8}$$

3. Mesh Deformation

Mesh quality is important in order to provide reliable results. In an aeroelastic computation, the geometry is altered as the structure deforms and the mesh is required to adjust appropriately. To ensure the quality of the deformed mesh, we use an analytic inverse distance method similar to the one described by Luke et al. [26]. With this method, the displacements of the CFD volume mesh are a combination of all surface deformations, weighted by the inverse of the distance to each surface node. The computational cost of a naive implementation of this method scales with the number of surface nodes. However, with a suitable fast spatial search algorithm and multipole-like expansion of the summation, the cost can be reduced to $\mathcal{O}(\log N)$. A careful implementation of the $\log N$ algorithm is fast and robust enough for use in aerostructural optimization. For a typical aerostructural analysis, the mesh movement scheme requires only 2-3% of the total solution time. With the mesh deformation algorithm, we have:

$$X_V = \mathcal{G}(u) \tag{3.1}$$

where u is the structural displacement and will affect the surface coordinates X_S; in turn X_S will determine the volume coordinates X_V. With this and the spectral interpolated grid velocity, we finally can rewrite the CFD residual form in terms of the structural displacement:

$$\mathcal{A}_{\text{TS}}(V_f, \omega, \zeta^n, u^n) = 0. \tag{3.2}$$

IV. Flutter Analysis

1. Time Spectral Flutter Equation

In flutter onset point prediction, besides CFD and CSD states, there are two additional unknowns: the flutter velocity index V_f and the frequency ω. In total, we have $2 + N_{CSD} + N_{CFD}$ where N_{CSD} is the number of CSD states and N_{CFD} are the number of CFD states. However, we only have $N_{CSD} + N_{CFD}$ equations if we just consider the CFD and CSD equations, and therefore two more equations are needed to close the system. Now we present TS flutter equation and its physical interpretation. The TS flutter equation is based on HB flutter equation from Thomas et al. [20].

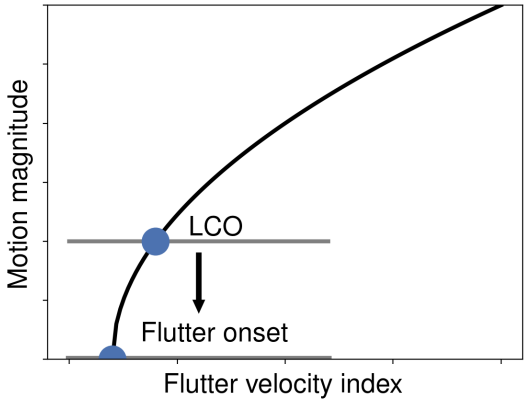


Figure 3. LCO curve

Without any additional constraints, all points from Fig. 3 are feasible solutions satisfying both CSD and CFD equations. By adding a constraint for the magnitude of the motion, the solution is limited to one point on the curve. This leaves the solution the freedom to shift in phase. Therefore, an equation constraining the phase is added, providing the required number of equations to fix a unique solution.

Here we show the derivation of these two equations constraining the magnitude and phase of the first pitching mode. Suppose the states from different time instances are defined as $\alpha_1, \alpha_2, ..., \alpha_N$. At first, we define the expansion of the variable $\alpha(t)$ over a harmonic basis:

$$\alpha(t) \approx c_0 + c_1 e^{i\frac{2\pi}{T}t} + c_2 e^{i\frac{4\pi}{T}t} + ... + c_{-2} e^{-i\frac{4\pi}{T}t} + c_{-1} e^{-i\frac{2\pi}{T}t}. \tag{1.1}$$

The first pitching mode is given as:

$$\begin{aligned}
\alpha_{\text{1st mode}} &= c_1 e^{i\frac{2\pi}{T}t} + c_{-1} e^{-i\frac{2\pi}{T}t} \\
&= (c_{1,r} + ic_{1,i}) \left(\cos\left(\frac{2\pi}{T}t\right) + i \sin\left(\frac{2\pi}{T}t\right) \right) + (c_{-1,r} + ic_{-1,i}) \left(\cos\left(-\frac{2\pi}{T}t\right) + i \sin\left(-\frac{2\pi}{T}t\right) \right) \\
&= (c_{1,r} + c_{-1,r}) \cos\left(\frac{2\pi}{T}t\right) + (-c_{1,i} + c_{-1,i}) \sin\left(\frac{2\pi}{T}t\right) + \text{pure imaginary number} \\
&= C_c \cos\left(\frac{2\pi}{T}t\right) + C_s \sin\left(\frac{2\pi}{T}t\right) + \text{pure imaginary number}
\end{aligned} \tag{1.2}$$

Dropping the pure imaginary number, the motion is governed by:

$$Re(\alpha_{\text{1st mode}}) = |\alpha_{\text{1st mode}}| \sin\left(\frac{2\pi}{T}t + \phi\right), \tag{1.3}$$

where magnitude and the phase of the 1st pitching mode are defined as:

$$\begin{aligned}
|\alpha_{\text{1st mode}}| &:= \sqrt{C_c^2 + C_s^2}, \\
\phi &:= \sin^{-1}\left(\frac{C_c}{\sqrt{C_c^2 + C_s^2}}\right).
\end{aligned} \tag{1.4}$$

Given c_i, the magnitude and the motion can be calculated from the equations above. The c_i coefficients are calculated by fast-Fourier-transform (FFT) with the pitching history as the input:

$$[c_0, c_1, c_2, ..., c_{-2}, c_{-1}] = \frac{1}{n} FFT(\alpha_1, \alpha_2, ..., \alpha_n). \tag{1.5}$$

Finally the residual of the two constraints are written out as following:

$$\begin{aligned}
\mathcal{R}_{\text{motion, magnitude}} &:= |\alpha_{\text{1st mode}}| - \epsilon_0, \\
\mathcal{R}_{\text{motion, phase}} &:= \phi - \phi_0,
\end{aligned} \tag{1.6}$$

where ϵ_0, ϕ_0 are prescribed small motion magnitude and its phase respectively.

Finally, we define TS flutter residual form as the aggregation of the motion equations residual, CSD equation residual and CFD equation residual:

$$\mathcal{R}(q) := \begin{pmatrix} \mathcal{R}_{\text{motion, magnitude}} \\ \mathcal{R}_{\text{motion, phase}} \\ \mathcal{S}_{\text{TS}} \\ \mathcal{A}_{\text{TS}} \end{pmatrix}, \tag{1.7}$$

where the state variables q is defined as:

$$q := \begin{pmatrix} V_f \\ \omega_f \\ u^n \\ \zeta^n \end{pmatrix}. \tag{1.8}$$

2. Coupled Newton-Krylov Solver for the Time Spectral Flutter Equation

Now we present the solution methodology for the TS flutter equation, Eqn. Eq. (1.7). The method used is a preconditioned coupled Jacobian-Free Newton-Krylov method. Key ideas of the method is explained briefly here. Equation (1.7) is solved using Newton's method, which results in the following linear system:

$$\begin{aligned}
J\Delta q &= -\mathcal{R}(q_k), \\
q_{k+1} &= q_k + \alpha \Delta q,
\end{aligned} \tag{2.1}$$

where J is the Jacobian, $\frac{\partial \mathcal{R}(q)}{\partial q}|_{q=q_k}$, Δq is the increment step, q_k and q_{k+1} are the current states and the states from next step, and α is a positive step size determined by either line search or trust region methods. Each increment step, Δq, is solved iteratively with a Krylov method, i.e. by minimizing the residual $J\Delta q + \mathcal{R}(q_k)$ norm in the span of $\{\mathcal{R}(q_k), J\mathcal{R}(q_k), ..., J^{m-1}\mathcal{R}(q_k)\}$, where m is the Krylov subspace size. The most computational demanding steps of this process are those related with matrix vector products, i.e. Jv. Instead of evaluating all the terms in the Jacobian, saving it explicitly and directly applying matrix vector product, we apply the following approximation, which is more economic in terms of both computational time and memory:

$$Jv \approx \frac{\mathcal{R}(q_k + \epsilon v) - \mathcal{R}(q_k)}{\epsilon}. \tag{2.2}$$

The details for Jacobian-Free Newton Krylov method can be found in [27]. Solving the TS flutter equation has now been reduced to residual evaluation. The residual evaluation is described in Algorithm 1:

Algorithm 1 Coupled nonlinear residual computation

function $\mathcal{R}(V_f, \omega_f, u^n, \zeta^n)$
  $X_S^n \leftarrow T u^n + X_J$ ▷ Transfer displacements
  $X_V^n \leftarrow \mathcal{W}(X_S^n)$ ▷ Deform volume mesh to match surface
  $\mathcal{A}_{\text{TS}} \leftarrow \mathcal{A}_{\text{TS}}(V_f, \omega_f, \zeta^n, X_V^n)$ ▷ Evaluate CFD residuals
  $f_A^n \leftarrow f_A^n(\zeta^n, X_S^n)$ ▷ Evaluate aerodynamics forces
  $\bar{f}^n \leftarrow T^T f_A^n$ ▷ Transfer forces
  $\mathcal{S}_{\text{TS}} \leftarrow \mathcal{S}_{\text{TS}}(V_f, \omega_f, u^n, \bar{f}^n)$ ▷ Evaluate CSD residuals
  $\mathcal{R}_{\text{motion, magnitude}} \leftarrow \mathcal{R}_{\text{motion, magnitude}}(u^n)$ ▷ Evaluate prescribed motion magnitude residual
  $\mathcal{R}_{\text{motion, phase}} \leftarrow \mathcal{R}_{\text{motion, phase}}(u^n)$ ▷ Evaluate prescribed motion phase residual
  $\mathcal{R} \leftarrow (\mathcal{R}_{\text{motion, magnitude}}, \mathcal{R}_{\text{motion, phase}}, \mathcal{S}_{\text{TS}}, \mathcal{A}_{\text{TS}})$ ▷ Combine residuals
  return $\mathcal{R}$
end function

We implement the solver through PETSc [21]. We use FGMRES to solve the linear system for each Newton step and select the cubic line search option. A relatively large subspace iteration number of 150 to 300 are used to enhance the robustness of the solver with a penalty on the solution speed. This solution method is an extension of Kenway et al. [22] which is a steady state aeroelastic solver.

2.1. Preconditioner

When solving Eq. (2.1), one critical requirement for good performance of an iterative method is that the eigenvalues of J be close with each other. To guarantee that, we need to carefully design a preconditioner. Similar to the previous steady aerostructural work, we use a block-Jacobi preconditioner. We implement a right preconditioner:

$$\begin{aligned} (JP^{-1})\Delta y &= -\mathcal{R}(q_k), \\ P^{-1}\Delta y &= \Delta q, \end{aligned} \tag{2.3}$$

where P is the preconditioner. To be more specific, the second equation is expanded as:

$$\begin{pmatrix} P^{-1}_{motion,CSD} & 0 \\ 0 & P^{-1}_{CFD} \end{pmatrix} \begin{pmatrix} \Delta y_{V_f,\omega,CSD} \\ \Delta y_{CFD} \end{pmatrix} = \begin{pmatrix} \Delta q_{V_f,\omega,CSD} \\ \Delta q_{CFD} \end{pmatrix}. \tag{2.4}$$

The preconditioner already implemented for CFD solver P^{-1}_{CFD} [1] (solution of CFD with a smaller stencil) was reused and a new preconditioner for the motion equations and the CSD equations are set up. Denoted as $P^{-1}_{motion,CSD}$, a direct inversion of the Jacobian is used as the preconditioner for the motion equations and the CSD equations. With relatively small size of the 2D case, a direct factorization is reasonable. However, with a larger structure, such as that required for a full 3D wing box, more careful research should be devoted to the preconditioner design. The resulting preconditioner is:

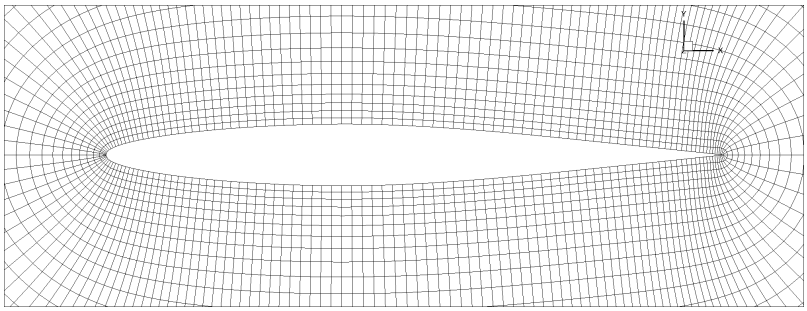
$$P^{-1}_{motion,CSD} = \begin{pmatrix} 0 & 0 & \frac{\partial |\alpha_{\text{1st mode}}|}{\partial u^n} \\ 0 & 0 & \frac{\partial \phi}{\partial u^n} \\ -\frac{2V_f}{\pi} f^n & \boldsymbol{M}\frac{d\boldsymbol{D}_Q}{d\omega} u^n & \boldsymbol{M}\boldsymbol{D}_Q + \boldsymbol{K} \end{pmatrix}^{-1}, \tag{2.5}$$

where all the terms are defined in

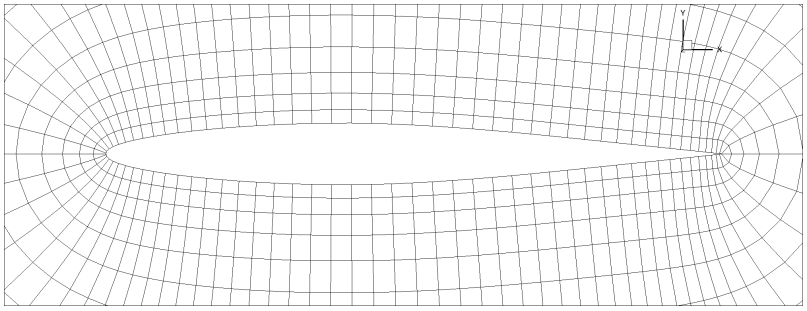
V. Results

1. Mesh convergence study

The mesh in this study has only one cell (plus ghost cells) in the spanwise direction, with symmetry planes on both sides to simulate two-dimensional flow. The fine mesh is a O-mesh topology with 192 cells around the airfoil and 64 cells from the airfoil to the far field and the flow field radius is about 90 times of the chord length. The coarse mesh has 96 cells around and 32 cells from airfoil to the far field and it has the same computational domain radius as the fine mesh. The near field meshes are shown in Fig. 4a and Fig. 4b respectively.



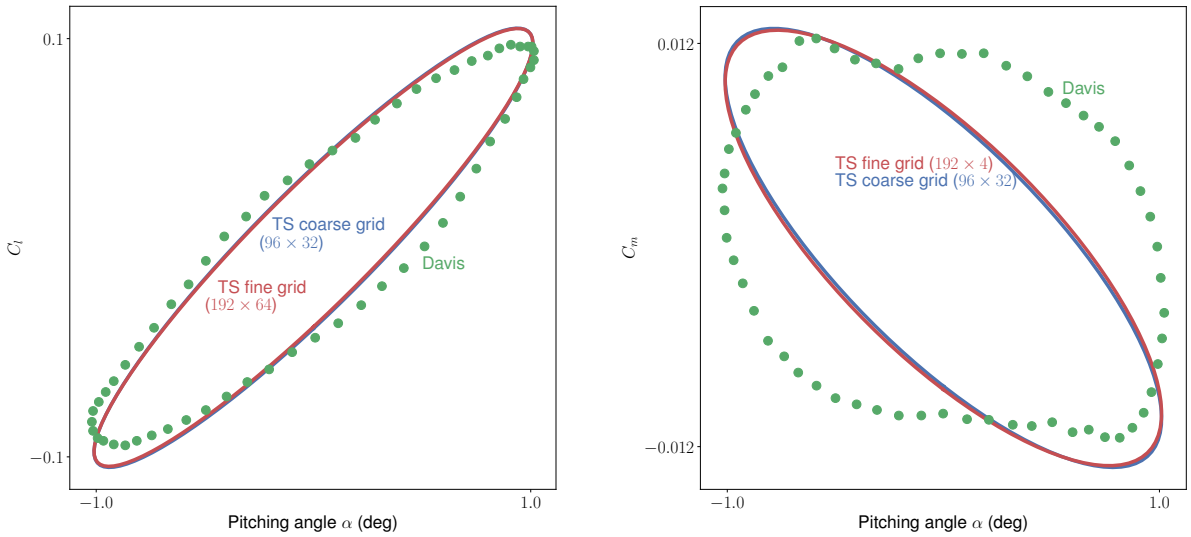
a) Fine mesh (192×64)



b) Coarse mesh (96×64)

Figure 4. NACA 64A010 mesh

In this benchmark case, we apply the same set up as Davis [28] for a 2D picthing NACA 640A10 airfoil. The CFD governing equation is the Euler equations. Figure 5 shows computed C_l and C_m hysteresis compared to the experimental results.



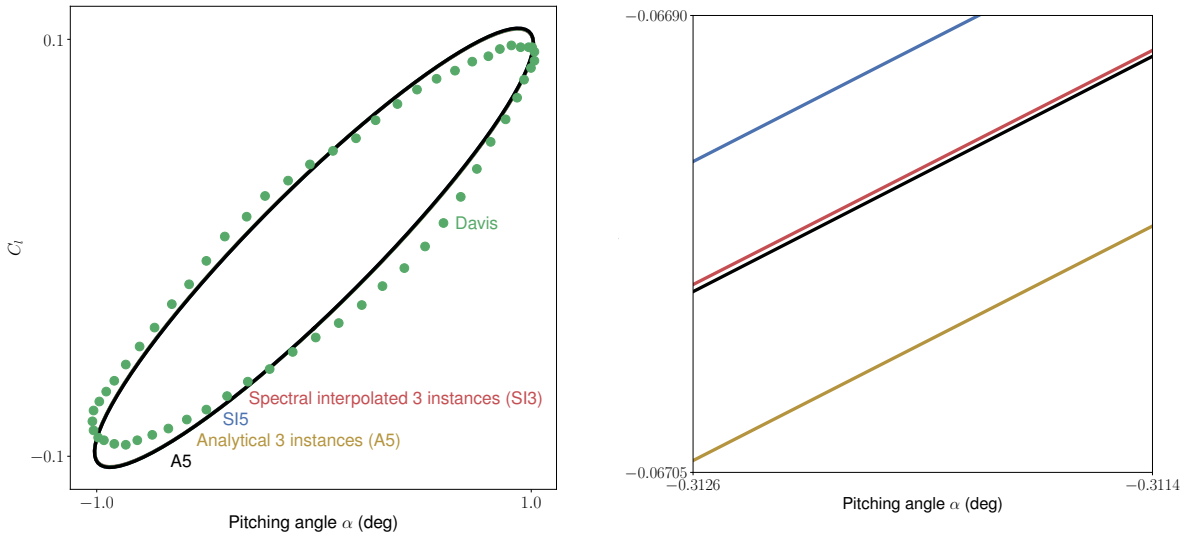
a) C_l **vs. Pitching angle** **b)** C_m **vs. Pitching angle**

Figure 5. NACA 64A010 mesh convergences study compared to results of Davis [28]

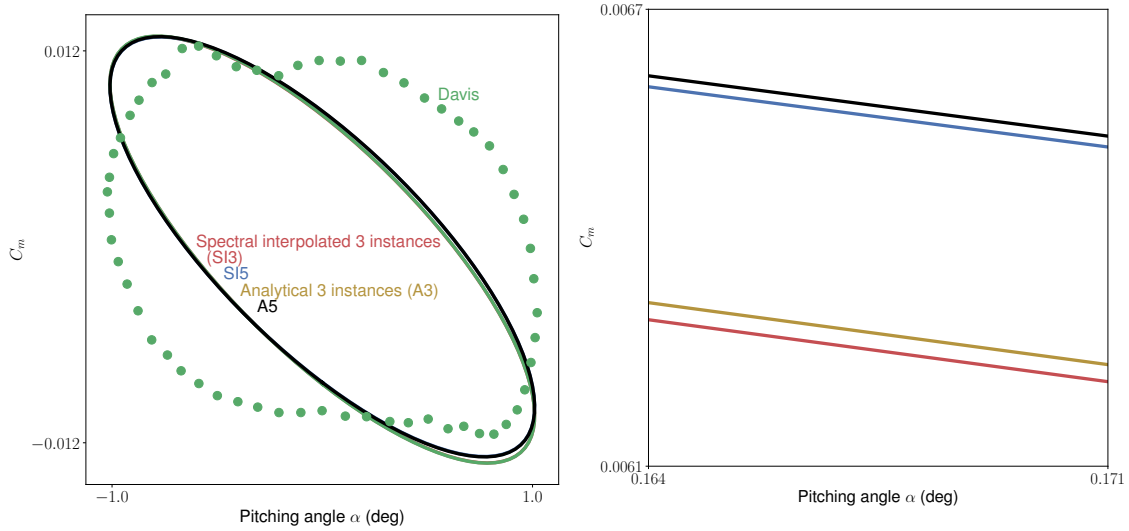
It can be seen that the results for the fine and coarse meshes are almost on top of each other. Numerical results match well with the experimental results as well. It also has a similar trend with McMullen's NLFD method [13].

2. Spectral interpolated grid velocity validation

In this benchmark case, we test our TS solver with warped mesh and spectral interpolated grid velocity. The spectral interpolated results are compared with results whose grid velocity is calculated analytically. We also consider the effect of number of time instances on the results. 3 and 5 time-instances are considered. Again, we apply the same setting as [28]. The results are shown in Fig.6.



a) C_l **vs. Pitching angle** **b)** C_l **vs. Pitching angle zoom in**



c) C_m **vs. Pitching angle** **d)** C_m **vs. Pitching angle zoom in**

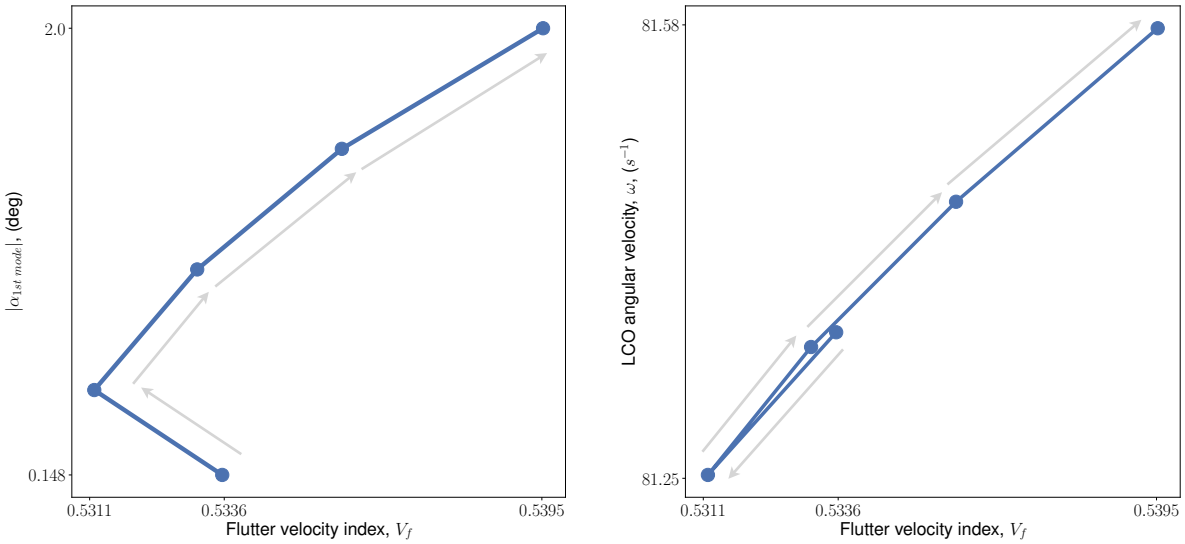
Figure 6. NACA 64A010 time instance number convergence results and spectral interpolation results

It can be seen that the C_l and C_m hysteresis which are calculated from spectral interpolating the grid velocity and applying the analytical grid velocity are similar under the same time instance numbers. There is also some visible discrepancy between the C_m hysteresis between the results with 3 and 5 instances. While the C_l hysteresis results are similar for simulations with difference time instance numbers. That may be attributed to the fact that C_m contains some higher frequency signal which can be resolved only if more time instances added. But in general all the results match well and a similar trend can be found from McMullen's thesis [13].

With the close match and for convenience, if not specified, all the results from the following sections are with spectral interpolated grid velocity, 3 time instances and coarse mesh.

3. LCO prediction

We apply our TS flutter method to predict multiple LCOs under different motion magnitude. The $|\alpha_{\text{1st mode}}|$ vs. V_f relation and ω vs. V_f relations are studied here.



a) $|\alpha_{\textbf{1st mode}}|$ **vs.** V_f **b)** ω **vs.** V_f

Figure 7. LCO responses under various V_f at $M = 0.85$ (the states q from the tails of the gray arrows are used to initialized the states q of the heads of the gray arrows)

From the V_f-$|\alpha_{1\text{st mode}}|$ relation, shown in Fig. 7a, it is observed that the airfoil undergoes a subcritical response – a decrease of V_f will give a increase in the LCO magnitude but subsequently an increment in V_f will result in yet another increase in LCO motion magnitude. Since the change of V_f is relatively small (less than 0.01 about 2% of V_f), the response is almost a supercritical one. A more severe subcritical response may cause so-called "hard flutter" where the structure will jump to a LCO with large magnitude from a steady-state. The frequency remains almost the same in the whole process as shown in Fig. 7b. So by solving the flutter equations with different prescribed pitching motion $|\alpha_{1\text{st mode}}|$, a $|\alpha_{1\text{st mode}}|$ vs. $|V_f|$ curve is obtained. With that curve,the supercritical/subcritical responses, which are critical for aircraft design, could be captured by the new proposed method.

4. Time accurate (TA) and TS verification

We validate our time-spectral flutter analysis result by comparing them to a time-accurate analysis. Here we use a partitioned loosely coupled scheme. The CFD solver's time-accurate analysis uses BDF2 time-integration, an implicit second order accurate scheme. The structural equations are integrated in time using the second order accurate Newmark-β [29] time-integration scheme. To ensure a second-order time-accurate analysis of the coupled aeroelastic system we apply corrector/predictor schemes [30, 31]. Multiple time step sizes are used in order to ensure a refined enough time step is used. The parameters are shown in Table 2. One aeroelastic response with $N = 9000$ per period at $M = 0.85$ and $V_f = 0.5331$ is shown in Fig. 8. A detailed TA vs TS comparison with different time step sizes is presented in Fig. 9.

Table 2. Parameters for TA simulation under $M = 0.85$ $V_f = 0.5331$

Steps per period	Step size, Δt, (sec)	Total step	Total time, T, (sec)
90	$8.727e^{-4}$	720	0.628
450	$1.745e^{-4}$	3600	0.628
4500	$1.745e^{-5}$	36000	0.628
9000	$8.727e^{-6}$	72000	0.628

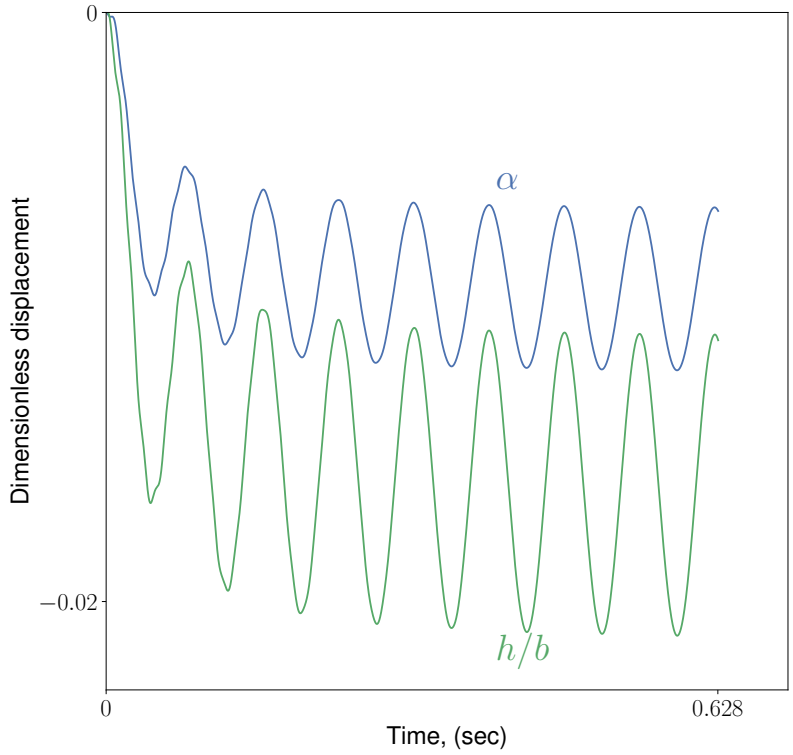


Figure 8. Displacement history of TA with $N = 9000$ per period

It can be seen that the transient response has been damped out and finally the periodic response is observed.

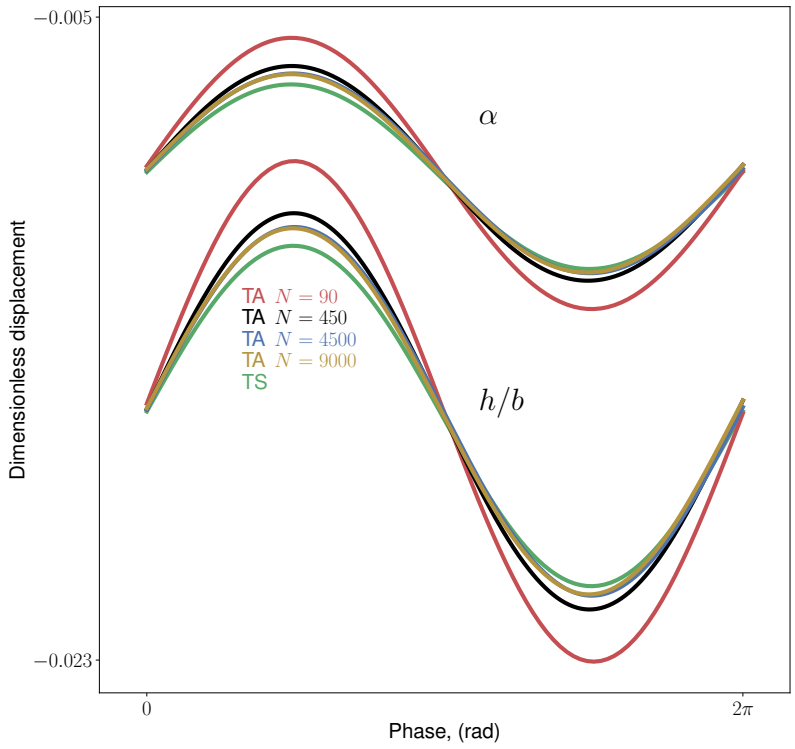


Figure 9. Comparison between time-accurate and time-spectral method at $M = 0.85, V_f = 0.5331$ ($N = 90$ indicates about 90 steps per period)

The time-spectral and the time-accurate match reasonably well – about 5% for the magnitude in plunging and pitching between TA $N = 9000$ and TS cases. It is worth mentioning how we got the TS flutter velocity here. After initializing with an almost zero displacement history, u^n (with small disturbance in case of division by zero in Eq. (1.2))) and an initial guess of $V_f = 0.51$, we are able to obtain the solution shown Fig. 9 in about 15 min. By contrast, for the TA result with 450 steps per period, with 3600 iterations to damp the transient response and get a convergent result, it took about 43 min to run. Thus, TS flutter solution method is almost 3 times faster than one TA analysis, not including the fact that several TA analysis are needed to fully determine the flutter boundary.

5. Flutter prediction benchmark case: Two-dimensional airfoil NACA 64A010

In this test, the flutter boundary for TS method is sequentially determined starting from $M = 0.75$. Each higher Mach number solution is solved with the neighboring lower Mach number solution as an initial guess. There are jumps of V_f about 25%, but the solver is still able converge the result which is partly attributed to the line search methods used in solving Eq. (2.1). In numerical tests, it is also found that a larger size of Krylov space also helps to converge difficult transonic case but with a price on solution time. More careful study on the robustness and efficiency trade-off is recommended. In order to verify the TA flutter boundary, a bracketing approach using a series of time-accurate analysis is performed by varying the flutter velocity index, V_f under the same Mach number, M, until the damping of the convergent results is within ± 0.05. Damping values of the non-linear time-accurate history is computed using an Autoregressive Moving Average method (ARMA) [32, 33]. The results from the proposed method as well as the time-accurate validation results are shown in Fig. 10. The four sets of results match well for lower Mach number, but there is a larger difference as Mach number getting close to 0.9 between the two methods found in literature and the TS and TA results. The agreement between the TS and TA results is however excellent. The trends are similar for all four methods and they all predicted some flutter dip. The difference may be caused by the different meshes used in those studies and the limited number of time-instances used with TS method.

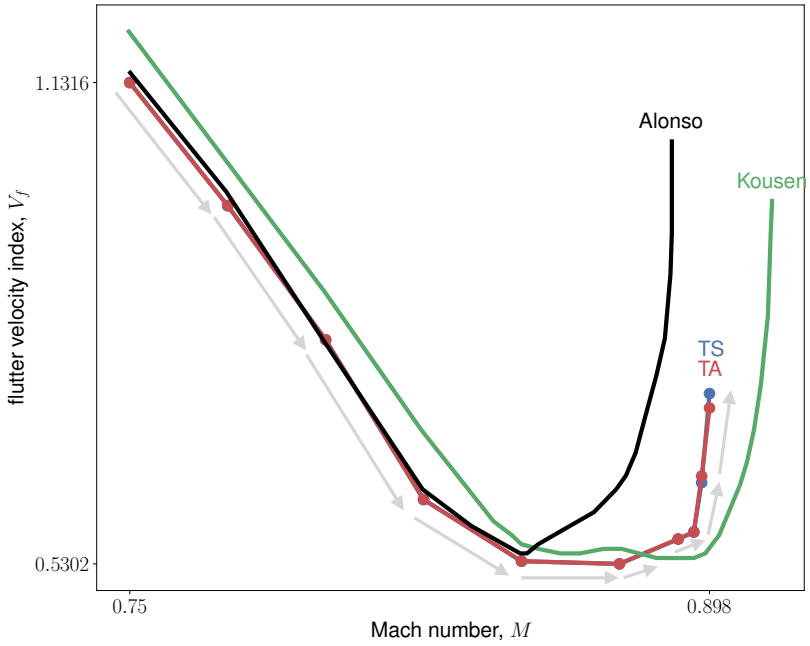
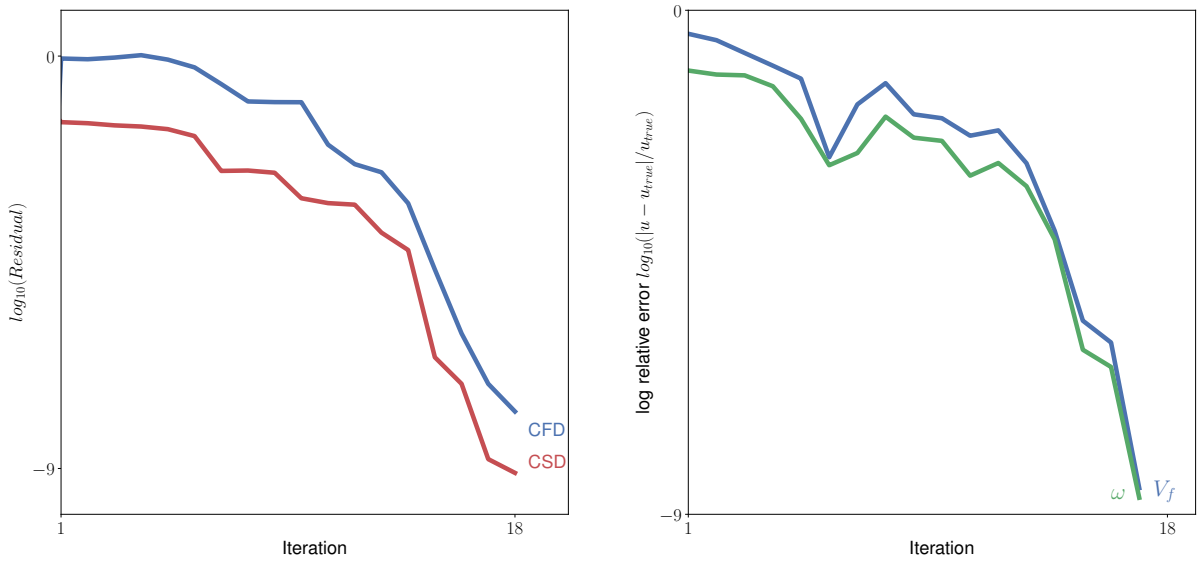


Figure 10. NACA 64A010 flutter onset boundary (the states q from the tails of the gray arrows are used to initialized the states q of the heads of the gray arrows; the flutter boundary here is approximated by a *LCO* with 0.148° pitching magnitude.)

5.1. Convergence study

In Fig. 11, we show the convergence history from solving for the flutter boundary at $M = 0.825$ initialized with states from the flutter boundary at $M = 0.8$.

As Fig. 11a shows, the residual drops slowly at the beginning. Towards the end, the reduction is faster which is a typical behavior for Newton's method – although in this study we did not observe quadratic convergence. In Fig. 11b, the history of relative error of V_f and ω is presented. Here we pick the V_f and ω from the last iteration as the "true" results. The trend is similar with that of the residual history. It is observed that despite of a 25% difference between



a) Residual history **b) V_f error history**

Figure 11. Convergence history of solving $M = 0.825$ flutter boundary with an initialization of $M = 0.8$ solution

the initial and true V_f, the CNK method is able to reduce the residual by over 7 orders in 18 Newton steps, validating the robustness and efficiency of the method.

VI. Conclusion

In this work, we present a high-fidelity flutter onset prediction methodology. A derivation of the time-spectral (TS) flutter equations which characterize the flutter onset point are shown. A coupled Jacobian Free Newton-Krylov method has been proposed and has been shown to be both robust and efficient. By implementing the method in the PETSc environment and leveraging the line search module, the method is more robust than previously used Newton-Raphson methods. By solving the coupled system together and removing all black box operations, the computationally expensive $O(N_{\text{CSD}})$ CFD evaluations for each Newton step are avoided, improving efficincy. Finally, the newly proposed method has been compared with and validated by the time-accurate (TA) method. The method has applications in flutter boundary prediction and can also be used to capture the subcritical/supercritical LCO responses.

VII. Acknowledgement

The first author wants to thank Gaetan Kenway who has been working in our MDO lab for sharing some fundamental ideas for this project. The authors also want to thank Daning Huang from our department for his critical contribution of implementing ALE (arbitrary Lagrangian-Eulerian) method in our ADflow code and the first author is also thankful for many insightful discussions on aeroelasticity.

References

[1] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "A Scalable Parallel Approach for High-Fidelity Aerostructural Analysis and Optimization," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, HI, April 2012, AIAA 2012-1922.

[2] Kenway, G. K. W. and Martins, J. R. R. A., "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, Vol. 51, No. 1, January 2014, pp. 144–160. doi:10.2514/1.C032150.

[3] Kenway, G. W. K. and Martins, J. R. R. A., "High-fidelity aerostructural optimization considering buffet onset," *Proceedings of the 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Dallas, TX, June 2015, AIAA 2015-2790.

[4] Albano, E. and Rodden, W. P., "A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows." *AIAA Journal*, Vol. 7, No. 2, Feburary 1969, pp. 279–285. doi:10.2514/3.5086.

[5] Batina, J. T., Seidel, D. A., Bland, S. R., and Bennett, R. M., "Unsteady transonic flow calculations for realistic aircraft configurations," *Journal of Aircraft*, Vol. 26, No. 1, 1989, pp. 21–28.

[6] Gibbons, M. D., "Aeroelastic calculations using CFD for a typical business jet model," 1996.

[7] Batina, J. T., Bennett, R. M., Seidel, D. A., Cunningham, H. J., and Bland, S. R., "Recent advances in transonic computational aeroelasticity," *Computers & Structures*, Vol. 30, No. 1, 1988, pp. 29–37.

[8] MSC Software Corp., *MSC.Nastran Reference Manual*, 2001.

[9] NEILL, D. J., JOHNSON, E. H., and CANFIELD, R., "ASTROS - A multidisciplinary automated structural design tool," *Journal of Aircraft*, Vol. 27, No. 12, dec 1990, pp. 1021–1027. doi:10.2514/3.45976.

[10] Schuster, D. M., Liu, D. D., and Huttsell, L. J., "Computational aeroelasticity: success, progress, challenge," *Journal of Aircraft*, Vol. 40, No. 5, 2003, pp. 843–856.

[11] Bennett, R. M. and Edwards, J. W., "An overview of recent developments in computational aeroelasticity," *AIAA paper*, , No. 98-2421, 1998.

[12] Liu, F., Cai, J., Zhu, Y., Tsai, H., and F. Wong, A., "Calculation of wing flutter by a coupled fluid-structure method," *Journal of Aircraft*, Vol. 38, No. 2, 2001, pp. 334–342.

[13] McMullen, M., *The Application of Non-Linear Frequency Domain Methods to the Euler and Navier-Stokes Equations*, Ph.D. thesis, Stanford University, March 2003.

[14] Hall, K. C., Thomas, J. P., and Clark, W. S., "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, Vol. 40, No. 5, 2015/06/01 2002, pp. 879–886. doi:10.2514/2.1754.

[15] Gopinath, A. and Jameson, A., "Time spectral method for periodic unsteady computations over two-and three-dimensional bodies," 2005.

[16] McMullen, M. and Jameson, A., "The Computational Efficiency of Non-Linear Frequency Domain Methods," *Journal of Computational Physics*, Vol. 212, No. 2, 2006, pp. 637–661.

[17] Kachra, F. and Nadarajah, S. K., "Aeroelastic Solutions Using the Nonlinear Frequency-Domain Method," *AIAA Journal*, Vol. 46, 2008, pp. 9.

[18] Choi, S. and Datta, A., "CFD Prediction of Rotor Loads using Time-Spectral Method and Exact Fluid-Structure Interface," *26th AIAA Applied Aerodynamics Conference*, Honolulu, HI, June 2008. doi:10.2514/6.2008-7325.

[19] Mundis, N. and Mavriplis, D., "Quasi-periodic Time Spectral Method for Aeroelastic Flutter Analysis," 2015/06/01 2013.

[20] Thomas, J. P., Dowell, E. H., and Hall, K. C., "Nonlinear Inviscid Aerodynamic Effects on Transonic Divergence, Flutter, and Limit-Cycle Oscillations," *AIAA Journal*, Vol. 40, No. 4, apr 2002, pp. 638–646. doi:10.2514/2.1720.

[21] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., *Efficient Management of Parallelism in Object Oriented Numerical Software Libraries*, Birkhäuser Press, 1997, pp. 163–202.

[22] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, May 2014, pp. 935–951. doi:10.2514/1.J052255.

[23] van der Weide, E., Gopinath, A., and Jameson, A., "Turbomachinery Applications with the Time Spectral Method," *35th AIAA Fluid Dynamics Conference and Exhibit*, American Institute of Aeronautics and Astronautics, jun 2005. doi:10.2514/6.2005-4905.

[24] 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, oct 1979, pp. 1030–1037. doi:10.2514/3.61273.

[25] Alonso, J. J. and Jameson, A., "Fully-Implicit Time-Marching Aeroelastic Solutions," *32nd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 1994. doi:10.2514/6.1994-56.

[26] Luke, E., Collins, E., and Blades, E., "A Fast Mesh Deformation Method Using Explicit Interpolation," *Journal of Computational Physics*, Vol. 231, No. 2, Jan. 2012, pp. 586–601. doi:10.1016/j.jcp.2011.09.021.

[27] Knoll, D. A. and Keyes, D. E., "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, Vol. 193, No. 2, 2004, pp. 357–397.

[28] Davis, S., "NACA 64A010 (NASA Ames Model) Oscillatory Pitching," Tech. rep., NASA, Ames, CA, 1982, Report 702, AGARD,Dataset 2.

[29] Bathe, K.-J., *Finite element procedures*, Prentice Hall, Pearson Education, Inc., Upper Saddle River, NJ, 2006.

[30] Farhat, C., van der Zee, K. G., and Geuzaine, P., "Provably Second-Order Time- Accurate Loosely-Coupled Solution Algorithms for Transient Nonlinear Computational Aeroelasticity," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 17, 2006, pp. 1973–2001.

[31] Miller, B., Crowell, A. R., and McNamara, J. J., "Loosely Coupled Time-Marching of Fluid-Thermal-Structural Interactions," *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Boston, MA, April 2013. doi:doi:10.2514/6.2013-1666.

[32] PAK, C.-G. and FRIEDMANN, P., "New time-domain technique for flutter boundary identification," *Dynamics Specialists Conference*, American Institute of Aeronautics and Astronautics, apr 1992. doi:10.2514/6.1992-2102.

[33] McNamara, J. J. and Friedmann, P. P., "Flutter Boundary Identification for Time-Domain Computational Aeroelasticity," *AIAA Journal*, Vol. 45, No. 7, July 2007, pp. 1546–1555. doi:10.2514/1.26706.