

Buffet Alleviation via Linear Stability Adjoint

Rohit Kanchi*

University of Tennessee, Knoxville, United States

Sicheng He†

University of Tennessee, Knoxville, United States

Eirikur Jonsson‡

University of Michigan, Ann Arbor, Michigan, United States

Joaquim R. R. A. Martins§

University of Michigan, Ann Arbor, Michigan, United States

Transonic buffet typically happens when the object is subjected to oscillatory or fluctuating aerodynamic forces. Buffeting can lead to vibrations, oscillations, and structural stresses, which can affect the stability and performance of the object. Thus, it is critical to avoid buffet in the flight envelope. However, the state-of-the-art buffet onset criteria, such as the 0.1° criterion and the separation sensor based method, are not based on first principal physics and thus can either cause overly conservative or infeasible designs. The linear stability analysis predict buffet onset by extracting the eigenvalue information around the steady-state solution. One bottleneck to include linear stability analysis is the difficulty to efficiently differentiate this constraint with a large number of design variable. In this paper, we propose to use multi-scale coupled adjoint method to efficiently compute the sensitivity of linear stability constraint with a large number of design variables. We will demonstrate the proposed method with an airfoil test case where we suppress the buffet onset.

I. Nomenclature

\mathbf{x} = vector of design variables

$f(\mathbf{x})$ = generic objective function

$g(\mathbf{x})$ = generic constraint

II. Introduction

The occurrence of transonic buffet arises from the separation of airflow induced by shock waves, typically found at the base of these waves. As the lift coefficient or Mach number increases, the intensity of shock waves amplifies, leading to the gradual development of buffet. The point at which buffet first manifests is referred to as buffet onset. Buffet is undesirable due to its adverse effects on aircraft control, passenger comfort, and structural integrity. Regulatory bodies such as the Federal Aviation Administration (FAA) and the European Union Aviation Safety Agency (EASA) mandate a 30% lift margin from the cruise operating condition to the buffet-onset boundary for commercial aircraft. This safety margin allows aircraft to navigate and handle turbulence effectively. Consequently, quantifying buffet onset and incorporating corresponding constraints in aerodynamic shape optimization becomes imperative.

It is essential to accurate and efficient predict buffet onset and incorporate this information in aircraft design. There are four methods to simulate buffet: (1). Time accurate method, (2). Time spectral method (or harmonic

*PhD Candidate, Department of Mechanical, Aerospace, and Biomedical Engineering, AIAA Member.

†Assistant Professor, Department of Mechanical, Aerospace, and Biomedical Engineering, AIAA Member.

‡Postdoc, Department of Aerospace Engineering, AIAA Member.

§Professor, Department of Aerospace Engineering, AIAA Fellow.

balance method), (3). Linear stability analysis, and (4). Heuristic methods. We review the methods and the status of design optimization using these methods. Examining buffet directly often requires employing high-fidelity, time accurate simulations utilizing computational fluid dynamics (CFD), such as unsteady Reynolds-averaged Navier–Stokes (URANS) [3–5], detached-eddy simulations [6–8], and large-eddy simulations [9,10]. However, these methods prove excessively computationally demanding when applied to aircraft shape optimization tasks, as simulations may need to be executed numerous times, even for a single-point optimization problem. In efforts to mitigate computational expenses, researchers have been exploring alternative approaches that assess buffet onset through steady CFD simulations.

Table 1 Comparison of different buffet simulation methods.

Method	First principal	Cost	Onset tracking	Subcritical response tracking
Time Accurate Method	✓	Very high	✗	✗
Time Spectral Method	✓	High	✓	✓
Linear Stability Method	✓	Medium	✗	✗
Heuristic	✗	Low	✓	✗

Buffet has been simulated using the time accurate simulation [1]. It was demonstrated that the URANS-based simulation can predict buffet onset with high accuracy. However, the computational cost of URANS is high and to conduct design optimization requires the development of unsteady adjoint, which can require significant development effort and cast high memory cost. Besides, buffet is a Hopf bifurcation problem and the stability of the bifurcation can also be an important design characteristics. The URANS-based methods are not able to predict an unstable bifurcation (or subcritical bifurcation).

Time spectral method was used to simulate buffet [2, 3]. The time spectral method is more efficient than the time accurate method in simulating periodic problem. In addition, it can also be used to predict subcritical response [4–6]. The equation is solved using Newton method which requires an initial solution that is in the convergent domain of the Newton solver. The initial solution can be obtained by solving a linear stability equation [4]. Thus, its computational cost is usually higher than the linear stability equation-based solver.

In the industry, an alternative heuristic approach to buffet-onset analysis relies solely on global aerodynamic coefficients, such as the lift-curve break method [7]. This method can be implemented using the $\Delta\alpha \approx 0.1$ deg formulation, where the buffet-onset point is estimated by the intersection of the lift curve and an auxiliary line formed by offsetting the linear portion of the lift curve to the right by 0.1 deg. Despite being computationally more economical than the URANS-based method, it still incurs a substantial cost, requiring a few RANS simulations for buffet onset evaluation. Li et al. [7] reduces the cost by directly relating the buffer onset with the surface pressure distribution. However, the heuristic methods are not based on first principle and with complex geometry they may not predict the buffet onset accurately.

Recently, the linear stability equation is solved to simulate the buffet onset [8]. The algorithm is shown to be able to capture the buffet onset for a realistic aircraft case. As mentioned above, the computational cost of linear stability analysis is expected to be lower than time accurate and time spectral methods. Although the linear stability method is not able to capture the subcritical buffet, by including higher order coefficients (second and third order Jacobian matrix), the subcritical response can be captured [9]. Due to its low computational cost, we select linear stability method to simulate buffet onset. A detailed comparison of different methods is given in table 1.

One major bottleneck for the adoption of large scale design optimization using the linear stability analysis is the efficient sensitivity computation. Marquet et al. [10] and Mettot et al. [11] proposed to compute the derivative of the buffet, a linear stability problem caused by the interaction of boundary layer and shock waves, with respect to a local forcing term. They proposed a sequential adjoint that computes the derivative in two stages: First, compute the derivative of the eigenvalue with respect to the Jacobian matrix entries. Then, compute the derivative of the nonlinear equation at the equilibrium point. The proposed algorithm scales well with the number of design variables. However, the mathematical derivation uses two simplifications: First, the design variable (the forcing term) only affects the underlying nonlinear system. A more general design variable that affect both the underlying nonlinear system and the eigenvalue problem can not be handled by this approach. Second, the function of interest is limited to eigenvalues. In

some applications, such as the laminar-turbulent transition flow modeling, the eigenvector (mode shape) derivatives is required [12]. Thus, we propose to extend the methods to include general design variables and functions of interest.

Alternatively, the coupled adjoint equation [13] can be used to compute the derivative of general design variables and functions of interest. This method combines all the individual coupled equations into a system of equations, and a global adjoint equation is constructed to compute the derivatives. However, blindly applying the coupled adjoint method will result in a high computational cost. Recently, Bouillé et al. [14] proposed using the continuous coupled adjoint to solve the derivative extending the results of their previous paper [15]. In this paper, we combine the idea of the coupled adjoint method and the sequential adjoint method in the following way: The coupled adjoint method provides the global adjoint equation that can be used to compute the derivative for general design variables and functions of interest. Then, leveraging the feed-forward structure of the residuals, we use an approach similar to the sequential adjoint to solve two individual adjoint equations to obtain a solution to the coupled adjoint equation. To extend the method to eigenvector derivative computation, we apply the general adjoint equation developed by He et al. [16].

The contributions of this paper are summarized as follows: 1) We extend the result of Marquet et al. [10] and Mettot et al. [11] to consider arbitrary design variables; 2) In addition to the eigenvalue derivative, the eigenvector derivative is also taken into account. We apply a more general eigenvalue problem adjoint proposed by He et al. [16] that can compute both the eigenvalue and eigenvector derivatives in a scalable way; 3) We propose an algorithmic differentiation (AD) and complex-step [17] mixed method to compute the derivative of the Jacobian-vector product. This combination gives machine-precision accuracy and is superior to the less accurate finite differences-based method found in the literature.

The paper is organized as follows. In Section III, we present the residual form for the equilibrium point solution and linear stability analysis. Next, in Section IV, we show the proposed derivative computation method based on the coupled adjoint method. Then, we present our previous heuristic method-based design optimization results in Section V. We verify the proposed derivative computation method with finite differences and also conduct linear stability-constrained optimization using several dynamical system examples in Section VI. Finally, preliminary results and tasks summarized in Section VII and ?? concludes the paper.

III. Linear stability analysis

In this section, we describe the analysis of both the underlying nonlinear dynamical system equations (the *bottom-level* problem) and the linear stability equation of the tangential dynamical system (the *top-level* problem) evaluated at a solution of the bottom-level problem. Then we combine these two blocks to obtain the governing equations for stability analysis. Finally, we present the forms of functions that can be evaluated by this method.

A. Bottom-level

We consider problems of the form

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{r}(\mathbf{w}; \mathbf{x}) = 0, \quad (1)$$

where $\mathbf{r} \in \mathbb{R}^n$ is the residual, $\mathbf{w} \in \mathbb{R}^n$ is the state variable, $\mathbf{x} \in \mathbb{R}^{n_x}$ is the design variable, n is the problem size, and n_x is the design variable dimension. The semi-colon in the residual form $\mathbf{r}(\mathbf{w}; \mathbf{x})$ distinguishes state variables from the design variables and other parameters. For an equilibrium point, \mathbf{w}_0 , we have

$$\mathbf{r}(\mathbf{w}_0; \mathbf{x}) = 0, \quad (2)$$

where \mathbf{w}_0 is no longer a function of time.

B. Top-level

The equilibrium point solution Eq. (2) is usually sufficient for engineering application. However, there are problems that we must consider the time derivative, e.g., the buffet onset problem encountered in aircraft wing design. Specifically, in some problems, in addition to the equilibrium point performance, we need to consider the stability of the equilibrium point. This is captured by the eigenvalue of the following eigenvalue problem.

First, we assume that the solution takes an infinitesimal perturbation around the equilibrium point \mathbf{w}_0 , so we can write

$$\mathbf{w} = \mathbf{w}_0 + \epsilon \mathbf{q} e^{\lambda t} + O(\epsilon^2), \quad (3)$$

where $\mathbf{q} \in \mathbb{C}^n$ is a normalized eigenvector, $\lambda \in \mathbb{C}$ is the eigenvalue, and ϵ is the perturbation magnitude. The eigenvalue and the eigenvector are from the Jacobian of the dynamical system Eq. (1) at the equilibrium point, \mathbf{w}_0 . There are multiple eigenvalue and eigenvector pairs available. Because the real part of the eigenvalues determines the stability of the linearized dynamical system, We choose the one whose eigenvalue has the largest real part.

The eigenvector is normalized for both its magnitude and phases

$$\begin{aligned} \mathbf{q}^* \mathbf{q} &= 1, \\ \mathbf{e}_k^\top \mathbf{q} &= 0, \end{aligned} \quad (4)$$

where “*” is the complex adjoint operator and k is the index of the entry with the largest magnitude, i.e.,

$$k = \operatorname{argmax}_l \|\mathbf{q}[l]\|_2, \quad (5)$$

where $\mathbf{q}[l]$ denotes the l^{th} entry of \mathbf{q} . The vector \mathbf{e}^k has all zero entries besides the k^{th} entry that is equal to 1. The normalization condition is not unique.

Inserting Eq. (3) into Eq. (1) and linearizing about the equilibrium point, neglecting high order terms for ϵ , and using the fact that $\mathbf{r}(\mathbf{w}_0; \mathbf{x}) = 0$, we obtain,

$$\frac{\partial \mathbf{r}}{\partial \mathbf{w}_0}(\mathbf{w}_0; \mathbf{x}) \mathbf{q} = \lambda \mathbf{q}. \quad (6)$$

Thus, the information passed from the bottom to the top level is the Jacobian matrix,

$$\mathbf{J} = \frac{\partial \mathbf{r}}{\partial \mathbf{w}_0}(\mathbf{w}_0; \mathbf{x}), \quad (7)$$

where $\mathbf{J} \in \mathbb{R}^{n \times n}$.

We can decompose the complex equations Eqs. (6) and (4) into two real equations, namely the real and imaginary components of the original equations. The resulting system of equations can then be written in terms of real numbers only for the top-level problem as

$$\hat{\mathbf{r}}(\mathbf{v}; \mathbf{w}_0, \mathbf{x}) = \hat{\mathbf{r}}(\mathbf{v}; \mathbf{J}(\mathbf{w}_0; \mathbf{x})) = 0, \quad (8)$$

where the residual, $\hat{\mathbf{r}}$, and the state variable, \mathbf{v} , are defined as

$$\hat{\mathbf{r}}(\mathbf{v}; \mathbf{w}_0, \mathbf{x}) = \begin{bmatrix} \hat{\mathbf{r}}_r \\ \hat{\mathbf{r}}_i \\ \hat{\mathbf{r}}_m \\ \hat{\mathbf{r}}_p \end{bmatrix} = \begin{bmatrix} \mathbf{J}\mathbf{q}_r - \lambda_r \mathbf{q}_r + \lambda_i \mathbf{q}_i \\ \mathbf{J}\mathbf{q}_i - \lambda_r \mathbf{q}_i - \lambda_i \mathbf{q}_r \\ \mathbf{q}_r^\top \mathbf{q}_r + \mathbf{q}_i^\top \mathbf{q}_i - 1 \\ \mathbf{e}_k^\top \mathbf{q}_i \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_i \\ \lambda_r \\ \lambda_i \end{bmatrix}. \quad (9)$$

The subscripts r and i denote the real and imaginary parts of the eigenvalue equation, respectively; the subscripts m and p denote the magnitude and the phase residual, respectively. More details on the eigenvalue problem setup are provided by He et al. [16].

For the solution to be locally stable, we need to enforce that

$$\lambda_r < 0. \quad (10)$$

For more details about the linear stability criterion, we refer the reader to Section 3.3 of the textbook by Slotine and Li [18]. Recall that the eigenvalue with the largest real part is the one we consider here. Computing this eigenvalue efficiently is still a challenging problem. This is because most iterative solvers (e.g., the Lanczos method) can only efficiently compute the eigenvalue with the largest magnitude that is not necessarily the one with the largest real part. The solution to this special eigenvalue problem is not the focus of the paper; there are various other references that detail methods for computing this eigenvalue [8, 19, 20]. Also, the equation does not track the Hopf bifurcation onset point but rather only gives information about whether the solution is stable. For the Hopf onset point solution method, see the paper by Roose and Hlaváček [21].

C. Governing equation for the linear stability analysis

Combining the bottom and the top-level residual forms, we have the governing equation for the linear stability problem

$$\begin{aligned}\mathbf{r}(\mathbf{w}_0; \mathbf{x}) &= 0, \\ \hat{\mathbf{r}}(\mathbf{v}; \mathbf{w}_0; \mathbf{x}) &= 0.\end{aligned}\tag{11}$$

There is no feedback from the top level to the bottom level. Later, we will leverage this observation to avoid a coupled adjoint solution.

D. Function of interest evaluation

The formulation is general because we can select any function of interest f . We define the general form of f that we can evaluate and compute derivatives for as

$$f = f(\mathbf{v}, \mathbf{w}_0; \mathbf{x}).\tag{12}$$

Specifically, when computing the linear stability, the real part of the eigenvalue is of interest; that is,

$$f = \lambda_r.\tag{13}$$

Further, in some problems, we need to evaluate and compute the derivative of a function with respect to the eigenvectors. For example, we may need to compute the weighted eigenvector derivative where the function of interest is defined as

$$f = \tilde{\mathbf{q}}_r^T \mathbf{q}_r + \tilde{\mathbf{q}}_i^T \mathbf{q}_i,\tag{14}$$

where $\tilde{\mathbf{q}}_r$ and $\tilde{\mathbf{q}}_i \in \mathbb{R}^n$ are constant weight vectors. In the aerodynamic shape optimization with the laminar-turbulent flow transition model, the eigenvector needs to be differentiated in this manner.

In the derivation presented in this paper, the general definition of the function of interest is used. The associated partial derivatives need to be computed based on the underlying function of interest.

E. Algorithm

The procedures discussed in this section are presented in Fig. 1 using an extended design structure matrix (XDSM) format [22]. The algorithm is defined in Algorithm 1. For the bottom-level and top-level equation solutions, the user has the freedom to choose their own solvers.

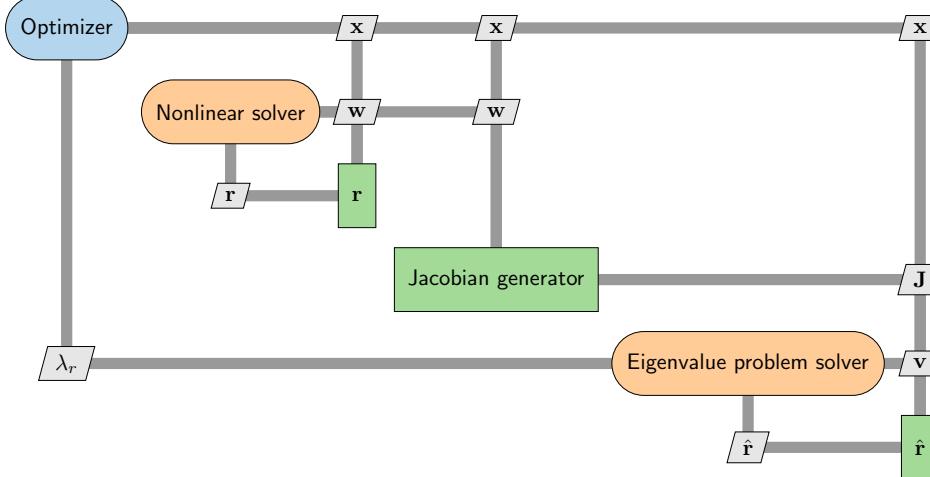


Fig. 1 XDSM for linear stability optimization.

Algorithm 1 Linear stability analysis

```

1: function  $\mathbf{f}_{\text{stab}}(\mathbf{x})$ 
2:    $\mathbf{w}_0 \leftarrow \mathbf{r}(\mathbf{w}_0; \mathbf{x}) = 0$                                  $\triangleright$  Bottom-level state variable solution.
3:    $\mathbf{J} = \frac{\partial \mathbf{r}}{\partial \mathbf{w}_0}(\mathbf{w}_0; \mathbf{x})$                           $\triangleright$  Construct Jacobian.
4:    $\mathbf{v} \leftarrow \hat{\mathbf{f}}(\mathbf{v}; \mathbf{J}(\mathbf{w}_0; \mathbf{x})) = 0$                        $\triangleright$  Top-level state variable solution.
5:    $\lambda_r \leftarrow \mathbf{v}$                                           $\triangleright$  Extract real part of the eigenvalue.
6:   return  $\lambda_r$ 
7: end function

```

IV. Linear stability adjoint sensitivity

This section discusses how to differentiate the linear stability measure, i.e., the maximum real part of all the eigenvalues, to obtain the derivative.

A. Coupled adjoint method

To compute the derivative, we propose using the adjoint method due to its favorable scaling with many design variables. Since we have two sets of equations, we need to solve the coupled adjoint equation. The total derivative equation and the coupled adjoint equation are written as follows:

$$\begin{aligned} \frac{df}{dx} &= \frac{\partial f}{\partial x} - \begin{bmatrix} \psi_r^\top & \psi_{\hat{r}}^\top \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial x} \\ \frac{\partial \hat{\mathbf{f}}}{\partial x} \end{bmatrix}, \\ \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial w_0}^\top & \frac{\partial \hat{\mathbf{f}}} {\partial w_0}^\top \end{bmatrix} \begin{bmatrix} \psi_r \\ \psi_{\hat{r}} \end{bmatrix} &= \begin{bmatrix} \frac{\partial f}{\partial w_0}^\top \\ \frac{\partial \hat{f}}{\partial v}^\top \end{bmatrix}, \end{aligned} \quad (15)$$

where the second equation is the coupled adjoint equation, and $\psi_r, \psi_{\hat{r}}$ are the adjoint vectors. As mentioned in the previous section, the stability problem has a special feature that the bottom-level residual form does not depend on the top-level state variables, that is,

$$\frac{\partial \mathbf{r}^\top}{\partial \mathbf{v}} = 0. \quad (16)$$

Thus, we can apply back-substitution by solving the second-row equation from Eq. (15) first and then solve the equation of the first row. Then, we have the top level adjoint equation

$$\psi_{\hat{r}} = \left(\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{v}} \right)^{-\top} \frac{\partial f}{\partial \mathbf{v}}, \quad (17)$$

and the bottom-level adjoint equation

$$\psi_r = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{w}_0} \right)^{-\top} \left(\frac{\partial f}{\partial \mathbf{w}_0}^\top - \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{w}_0}^\top \psi_{\hat{r}} \right). \quad (18)$$

By reducing the solution process into two separated adjoint equations, we avoid having to solve a larger coupled adjoint equation directly. The benefit of this approach is that the individual adjoint equation algorithms and solvers are more well-developed than those of the coupled adjoint equation. In addition, the two segregated adjoint equations are about a third and two thirds the size of the coupled one, respectively, making them cheaper to solve.

B. Top level

The top-level adjoint equation is defined by Eq. (17) where the coefficient matrix is found to be

$$\frac{\partial \hat{\mathbf{r}}^\top}{\partial \mathbf{v}} = \begin{bmatrix} \mathbf{J} - \lambda_r \mathbf{I} & \lambda_i \mathbf{I} & -\mathbf{q}_r & \mathbf{q}_i \\ -\lambda_i \mathbf{I} & \mathbf{J} - \lambda_r \mathbf{I} & -\mathbf{q}_i & -\mathbf{q}_r \\ 2\mathbf{q}_r^\top & 2\mathbf{q}_i^\top & 0 & 0 \\ 0 & \mathbf{e}_k^\top & 0 & 0 \end{bmatrix}^\top. \quad (19)$$

When the real part of the eigenvalue, λ_r , is taken as the function of interest Eq. (13), the term of the top level adjoint equation, $\partial f / \partial \mathbf{v}$ is defined as

$$\frac{\partial f}{\partial \mathbf{v}} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}. \quad (20)$$

Otherwise, if a function of the eigenvector is used, such as Eq. (14), the partial derivative is set to be

$$\frac{\partial f}{\partial \mathbf{v}} = \begin{bmatrix} \tilde{\mathbf{q}}_r^\top & \tilde{\mathbf{q}}_i^\top & 0 & 0 \end{bmatrix}. \quad (21)$$

For a general case, we have to solve Eq. (17). However, when the function of interest is constructed from the real or an imaginary part of the eigenvalue or eigenvector, we have an analytic solution for this equation (for detailed derivation, see [16]).

For instance, when the function of interest is defined as the real part of the eigenvalue, i.e., $f = \lambda_r$, then the adjoint solution is

$$\boldsymbol{\psi} = \begin{bmatrix} \mathbf{u}_r^{(1)\top} & \mathbf{u}_i^{(1)\top} & 0 & 0 \end{bmatrix}^\top, \quad (22)$$

where $\mathbf{u}^{(1)}$ is the left eigenvector satisfying the following normalization condition

$$\mathbf{J}^\top \mathbf{u}^{(1)} = \lambda^* \mathbf{u}^{(1)}, \quad \mathbf{u}^{(1)*} \mathbf{q} = -1. \quad (23)$$

Similarly, when the function of interest is defined as the imaginary part of the eigenvalue, i.e., $f = \lambda_i$. It shows up in problems where certain frequencies need to be avoided (see [23]). The adjoint solution is

$$\boldsymbol{\psi} = \begin{bmatrix} \mathbf{u}_r^{(2)\top} & \mathbf{u}_i^{(2)\top} & 0 & 0 \end{bmatrix}^\top, \quad (24)$$

where the $\mathbf{u}^{(2)}$ is the left eigenvector that satisfies the following normalization condition

$$\mathbf{J}^\top \mathbf{u}^{(2)} = \lambda^* \mathbf{u}^{(2)}, \quad \mathbf{u}^{(2)*} \mathbf{q} = i. \quad (25)$$

This can be verified by substituting Eqs. (22) and (24) into Eq. (17). The eigenvectors $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}$ are the same eigenvector, but due to the different normalization conditions, they take different forms, hence the difference in naming.

When we compute the stability derivative, we recommend solving for the left eigenvector and using the analytic solution for the adjoint instead of solving Eq. (17). This is because we can apply the same method to solve the left eigenvalue problem.

C. Bottom level

The bottom-level adjoint equation is defined by Eq. (18). We detail each term in the following discussion. Although, for both Eq. (13) and Eq. (14), the first term $\partial f / \partial \mathbf{w}_0$ is zero, but for a more general case it could be nonzero, and thus has to be retained. The second term, $(\partial \hat{\mathbf{r}} / \partial \mathbf{w}_0)^\top \boldsymbol{\psi}_{\hat{\mathbf{r}}}$ can be expanded as follows

$$\frac{\partial \hat{\mathbf{r}}^\top}{\partial \mathbf{w}_0} \boldsymbol{\psi}_{\hat{\mathbf{r}}} = \frac{\partial \mathbf{q}_r^\top \mathbf{J}^\top \boldsymbol{\psi}_{\hat{\mathbf{r}},r}}{\partial \mathbf{w}_0} + \frac{\partial \mathbf{q}_i^\top \mathbf{J}^\top \boldsymbol{\psi}_{\hat{\mathbf{r}},i}}{\partial \mathbf{w}_0}, \quad (26)$$

where $\boldsymbol{\psi}_{\hat{\mathbf{r}},r} \in \mathbb{R}^n$ and $\boldsymbol{\psi}_{\hat{\mathbf{r}},i} \in \mathbb{R}^n$ are sub-vector of the vector $\boldsymbol{\psi}_{\hat{\mathbf{r}}}$. More specifically,

$$\boldsymbol{\psi}_{\hat{\mathbf{r}}} = \begin{bmatrix} \boldsymbol{\psi}_{\hat{\mathbf{r}},r}^\top & \boldsymbol{\psi}_{\hat{\mathbf{r}},i}^\top & \boldsymbol{\psi}_{\hat{\mathbf{r}},m}^\top & \boldsymbol{\psi}_{\hat{\mathbf{r}},p}^\top \end{bmatrix}^\top, \quad (27)$$

where $\psi_{\hat{\mathbf{r}},m} \in \mathbb{R}$ and $\psi_{\hat{\mathbf{r}},p} \in \mathbb{R}$ are two scalars related to the magnitude and phase residuals of Eq. (4).

To compute Eq. (26) we need a method that can compute the derivative of the pattern

$$\frac{\partial \mathbf{r}_2^\top \mathbf{J}^\top \mathbf{r}_1}{\partial \mathbf{w}}. \quad (28)$$

We propose two methods to compute it.

For the first proposed method, we use reverse algorithmic differentiation (RAD) and finite differences. RAD is a form of AD that differentiate a program by accumulating the derivative from the outputs backward to the inputs [24]. In the proposed method, the residual form needs to be differentiated using RAD. For more details about RAD [25, Sec. 6.6]. The result is

$$(\text{FDRAD}) : \frac{\partial \mathbf{r}_2^\top \mathbf{J}^\top \mathbf{r}_1}{\partial \mathbf{w}} \approx \left(\frac{\frac{\partial \mathbf{r}(\mathbf{w}_0 + h\mathbf{r}_2)^\top}{\partial \mathbf{w}} \mathbf{r}_1 - \frac{\partial \mathbf{r}(\mathbf{w}_0)^\top}{\partial \mathbf{w}} \mathbf{r}_1}{h} \right)^\top. \quad (29)$$

We denote it as FDRAD for the forward difference of a RAD formula.

The terms $(\partial \mathbf{r}(\mathbf{w}_0 + h\mathbf{r}_2)/\partial \mathbf{w})^\top \mathbf{r}_1$ and $(\partial \mathbf{r}(\mathbf{w}_0)/\partial \mathbf{w})^\top \mathbf{r}_1$ can be computed using RAD. Besides the forward difference, we can use the central difference as well. The central difference formula is

$$(\text{CDRAD}) : \frac{\partial \mathbf{r}_2^\top \mathbf{J}^\top \mathbf{r}_1}{\partial \mathbf{w}} \approx \left(\frac{\frac{\partial \mathbf{r}(\mathbf{w}_0 + h\mathbf{r}_2)^\top}{\partial \mathbf{w}} \mathbf{r}_1 - \frac{\partial \mathbf{r}(\mathbf{w}_0 - h\mathbf{r}_2)^\top}{\partial \mathbf{w}} \mathbf{r}_1}{2h} \right)^\top. \quad (30)$$

We denote it as CDRAD for the central difference of a RAD formula.

For the second proposed method, we use RAD and complex-step. We propose using the following formula to compute the second-order derivatives

$$(\text{CSRAD}) : \frac{\partial \mathbf{r}_2^\top \mathbf{J}^\top \mathbf{r}_1}{\partial \mathbf{w}} = \frac{\text{Im} \left(\frac{\partial \mathbf{r}(\mathbf{w}_0 + i\mathbf{h}\mathbf{r}_2)^\top}{\partial \mathbf{w}} \mathbf{r}_1 \right)}{h}, \quad (31)$$

where a machine precision can be achieved by selecting a small h , e.g. $h = 10^{-200}$. Thus, the “ \approx ” is replaced by “ $=$ ”. We denote it as CSRAD for the complex step of a RAD formula.

CSRAD is much more accurate than CDRAD and FDRAD. However, implementing a PDE solver may require more development effort because the differentiated code needs to be complexified. We compare the Eqs. (29), (30) and (31) in Section VI.A.1.

D. Algorithm

We summarize and organize the previously discussed operations into Algorithm 2. In Algorithm 2, we assume $f = \lambda_r$ ommit zero terms.

Algorithm 2 Local stability derivative using the block back-substitution method.

```

1: function  $\mathbf{g}_{\text{stab}}(\mathbf{x})$ 
2:    $\frac{\partial \lambda_r}{\partial \mathbf{v}} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\top$  ▷ Set the RHS for the top-level adjoint equation.
3:    $\boldsymbol{\psi}_{\hat{\mathbf{r}}} \leftarrow \frac{\partial \hat{\mathbf{r}}^\top}{\partial \mathbf{v}} \boldsymbol{\psi}_{\hat{\mathbf{r}}} = \frac{\partial \lambda_r}{\partial \mathbf{v}}$  ▷ Solve the top-level adjoint by direct solution Eq. (17) or analytic solution Eq. (22).
4:    $\mathbf{b} \leftarrow \frac{\partial \hat{\mathbf{r}}^\top}{\partial \mathbf{w}_0} \boldsymbol{\psi}_{\hat{\mathbf{r}}}$  ▷ Evaluate the partial derivative using one of Eqs. (29), (30), and (31).
5:    $\boldsymbol{\psi}_{\mathbf{r}} \leftarrow \frac{\partial \mathbf{r}^\top}{\partial \mathbf{w}_0} \boldsymbol{\psi}_{\hat{\mathbf{r}}} = \mathbf{b}$  ▷ Solve the bottom-level adjoint Eq. (18).
6:    $\frac{d\lambda_r}{dx} \rightarrow -\boldsymbol{\psi}_{\mathbf{r}}^\top \frac{\partial \mathbf{r}}{\partial x} - \boldsymbol{\psi}_{\hat{\mathbf{r}}}^\top \frac{\partial \hat{\mathbf{r}}}{\partial x}$  ▷ Compute the total derivative.
7:   return  $\frac{d\lambda_r}{dx}$ 
8: end function

```

From Algorithm 2, we find that no matter how many design variables we have, we only need to solve the coupled adjoint equation once. For more details about a comparison of different optimization strategies, such as using gradient-free and gradient-based together with the finite differences, we refer the reader to the paper by [26] in the context of PDE-constrained optimization.

One advantage of the block back-substitution method is that it requires much lower implementation effort than the direct solution method. This is because the coupled system is never formed explicitly, and existing implementation of the top-level eigenvalue adjoint equation solver and the bottom-level adjoint equation solver can be applied directly.

Besides the more straightforward implementation, the block back-substitution method is also more efficient than the direct solution method. For large-scale PDE-based problems, the adjoint equations are high-dimensional and usually sparse. In this case, we may never form the coefficient matrices explicitly and use a Jacobian-free approach to solve the adjoint equations. An iterative solver, such as Krylov subspace methods, can be used together with a preconditioner to solve such equations. One example is the aerostructural optimization coupled-adjoint solver proposed by Kenway et al. [13]. More general coupled-adjoint formulations are also possible with the OpenMDAO framework [27], which has been used in many applications, including PDE-constrained problems [28]. A well-constructed preconditioner is essential for the efficiency of the overall solution. The preconditioner for a coupled problem can be constructed by putting the segregated preconditioners on the diagonal. This diagonal preconditioner may work to some extent but is likely not comparable to a full preconditioner. More computational studies need to be conducted for the adjoint solvers in the future to validate the discussion.

V. Previous results

In our previous paper [7], we studied buffet using a data-driven heuristic based on the $\Delta\alpha = 0.1^\circ$ method. The analysis result is shown in Fig. 2. The choice of investigated wing sections significantly influences the estimated buffet factor. We show this effect in Fig. 2, where the CRM wing is analyzed using RANS ($M = 0.85$, $Re = 5.0 \times 10^6$, and $\alpha = 4.0^\circ$). The left and right contours show the pressure and friction coefficients, respectively. The shock wave is shown as the orange domain in the left panel. Seven wing sections (locations indicated by the white slice lines) between the wingtip and the wing root are investigated. Distributions of the pressure coefficients and friction coefficients across the wing sections are shown in the bottom panel, and the data-driven model evaluates the buffet factors of the sectional shapes. From Fig. 2, we find the flow separations are mostly located outboard of the Yehudi break. Thus, it may be reasonable to choose wing sections from this region for this case. Nevertheless, the appropriate wing sections would be case-dependent, and we find that the Mach number has a significant impact.

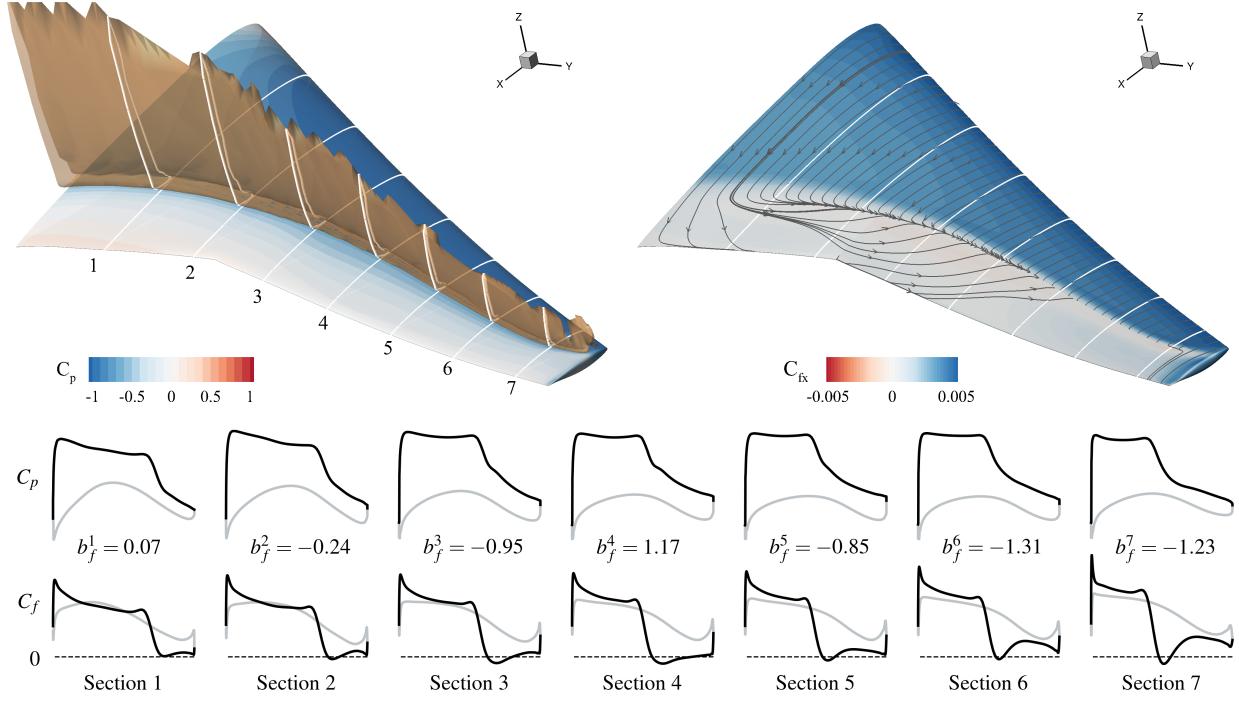


Fig. 2 Buffet analysis of wing sections using the data-driven method.

With the optimization effectiveness demonstration in the previous case study, we now apply the EGO framework to this case. We perform two optimizations with the buffet constraints imposed by the data-driven method ($\beta^{\text{GMM}} \leq 0.0$) and the Kenway–Martins method ($\chi \leq 4\%$), respectively. The optimized results are shown in Fig. 3, where the buffet boundaries of the optimized wings are computed using the $\Delta\alpha = 0.1^\circ$ method. As explained previously, the solved buffet onset points may vary due to different selections of linear portions of lift curves, so the boundaries are shown as shades in Fig. 3. The data-driven method appropriately imposes the buffet constraints, and the boundary of the optimized wing almost crosses the two buffet points. However, the Kenway–Martins method is conservative, leading to a wing with a higher buffet boundary than required. The influence of this overstatement is clearly negative: larger cruise drag and more complex shock waves on the wing surface. As an accurate alternative of the $\Delta\alpha = 0.1^\circ$ method, the presented data-driven model provides an efficient way to impose buffet onset constraints in wing shape design optimization.

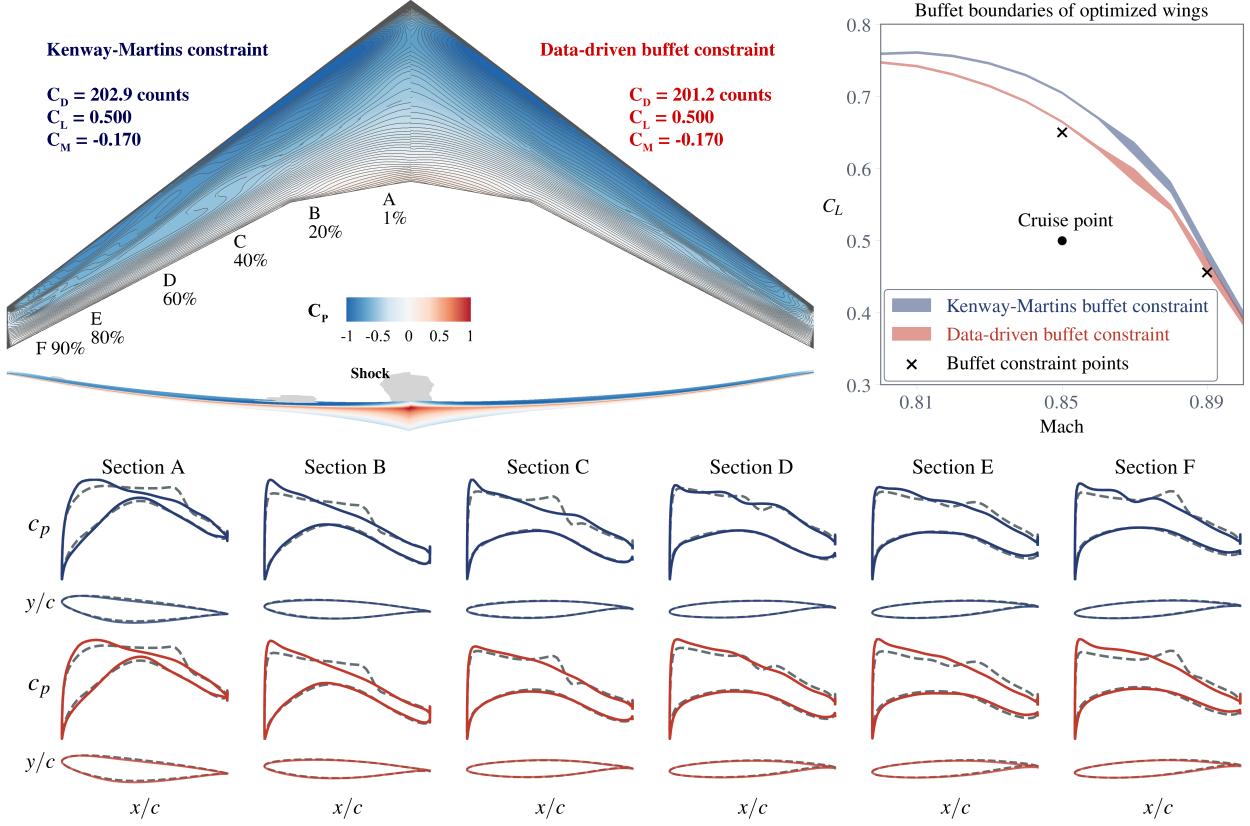


Fig. 3 Wing shape optimization results show that the data-driven method is more effective to impose buffet onset constraints.

VI. Results for the multi-scale coupled adjoint for dynamical system

We verified the proposed multi-scale coupled adjoint method with a simple algebraic example. The problem is relevant because similar to buffet, under certain parameters, Hopf bifurcation will be triggered for this case. We conduct design optimization to suppress this instability.

A. Verification

In this section, we verify the proposed second order partial derivative computation method in Section VI.A.1, and then we verify the adjoint-based total derivative computation method in Section VI.A.2.

1. Partial derivative verification

In this section, we verify FDRAD, CDRAD, and CSRAD methods Eqs. (29), (30), and (31) proposed in Section IV.C. The following dynamical system considered is,

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} = \begin{bmatrix} 0.3w_1 - w_2 + w_1^3 - 0.2e^{w_1} - 0.1 \\ w_1 + 0.2w_2 + w_2^3 - w_1w_2^3 \end{bmatrix}, \quad (32)$$

where the state variables are written as,

$$\mathbf{w} = [w_1 \quad w_2]^\top. \quad (33)$$

The dynamical system is a manufactured test example that will become linearly unstable under some combination of design variables.

The vector \mathbf{r}_1 and \mathbf{r}_2 are randomly generated as

$$\mathbf{r}_1 = \begin{bmatrix} 0.54881350 \\ 0.71518937 \end{bmatrix}, \quad \mathbf{r}_2 = \begin{bmatrix} 0.60276338 \\ 0.54488318 \end{bmatrix}. \quad (34)$$

The result is plotted in Fig. 4. As anticipated, the proposed CSRAD formula is accurate to machine precision, resulting in a more accurate total derivative computation than the finite differences-based FDRAD and CDRAD formulas.

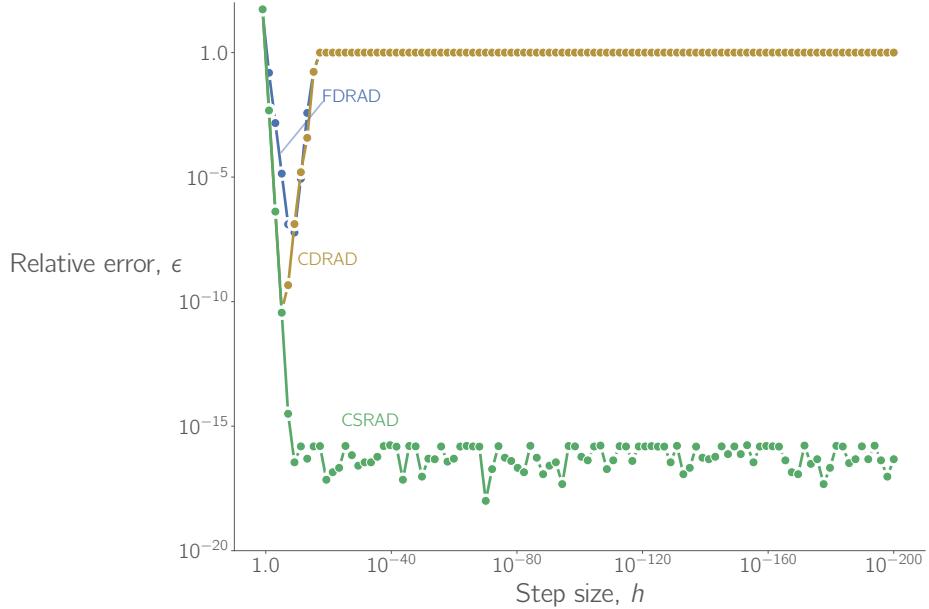


Fig. 4 In contrast to the finite differences methods FDRAD and CDRAD Eq. (29) and (30), the complex-step method CSRAD Eq. (31) is not subject to subtractive cancellation errors.

2. Total derivative verification

We consider the following dynamical system

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} = \begin{bmatrix} (x_1 - 1.2x_2^2)w_1 - w_2 + (2x_2 - 1)w_1^3 - 0.1 \\ w_1 + (x_1 - 1)w_2 + w_2^3 \end{bmatrix}, \quad (35)$$

where the state variables are

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^\top, \quad (36)$$

and the design variables are

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top. \quad (37)$$

The objective function is set to

$$f(\mathbf{w}_0, \mathbf{x}) = 0.3(1 - x_1)^2 + 0.5(x_2 - 0.5)^2 + w_{0,1}^2 + 3w_{0,2}. \quad (38)$$

The objective function is constructed to include terms from both the state variables and the design variables.

We test the stability derivative computed using Algorithm 2. The finite differences use a step size of 10^{-6} at an arbitrarily selected point

$$x_1 = 0.3, \quad x_2 = 0.7. \quad (39)$$

The finite-difference and adjoint results are listed in Table 2. The derivative computed using the adjoint method has a less than 10^{-6} relative difference compared to the finite-difference approximation. This result demonstrates the accuracy of the proposed algorithm Algorithm 2.

Table 2 Comparison of the stability derivative, $d\lambda_r/dx$, computed using the adjoint method and finite differences.

x	Finite differences	Adjoint	Relative difference
x_1	1.01453326	1.014533 <u>17</u>	8.42×10^{-8}
x_2	-0.85409190	-0.854091 <u>62</u>	3.36×10^{-7}

As mentioned before, the function of interest can take a more general form. Here we set f to be the function defined as

$$f_{\text{loss}} = \|\mathbf{q}_r - \mathbf{q}_{r,\text{tgt}}\|_2^2 + \|\mathbf{q}_i - \mathbf{q}_{i,\text{tgt}}\|_2^2, \quad (40)$$

where $\mathbf{q}_{r,\text{tgt}}$ and $\mathbf{q}_{i,\text{tgt}}$ are the real and imaginary parts of the targeted eigenvector. For this case, the function of interest, f , is a penalized eigenvector loss function.

The target eigenvector is

$$\mathbf{q}_{r,\text{tgt}} = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ 0 \end{bmatrix}, \quad \mathbf{q}_{i,\text{tgt}} = \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix}. \quad (41)$$

The total derivatives computed by finite differences and proposed algorithm are listed in Table 3. Like the linear stability measure derivative, the loss function derivative is accurate, with a relative difference ranging from 10^{-6} to 10^{-7} .

Table 3 Comparison of the loss function derivative, df_{loss}/dx , computed using the adjoint method and finite differences.

x	Finite differences	Adjoint	Relative difference
x_1	0.00280319	0.00280319	4.99×10^{-7}
x_2	0.11593017	0.115930 <u>33</u>	1.35×10^{-6}

B. Optimization

When using a gradient-based optimizer to conduct optimization, the optimizer needs to know both the function value and the function derivatives with respect to the design variables. The function value is obtained using Algorithm 1 and the derivative value is evaluated using Algorithm 2. Multiple steps are required to find the optimum. For more details of the gradient-based optimization, we refer the reader to the textbook by Martins and Ning [25].

1. An algebraic problem

The detailed problem setup is defined in Section VI.A.1. For the baseline design, we set

$$x_1^{(0)} = 0.9, \quad x_2^{(0)} = 0.2. \quad (42)$$

In addition, the stability upper bound is set to $\sigma = -0.1$. We use SNOPT [29] optimizer with a python wrapper named pyOptSparse [30] for this optimization study.

Figure 5 shows the contours of the dominant eigenvalue real part and the objective function, $f(\mathbf{w}_0, \mathbf{x})$. The path the optimizer takes is also shown on both contour plots. In one step, the optimizer finds a feasible solution and remains in the feasible domain until converging to a point where the linear stability constraint is active. The optimized solution is

$$x_1^* = 0.554, \quad x_2^* = 0.536. \quad (43)$$

It takes the optimizer seven major iterations to find the solution.

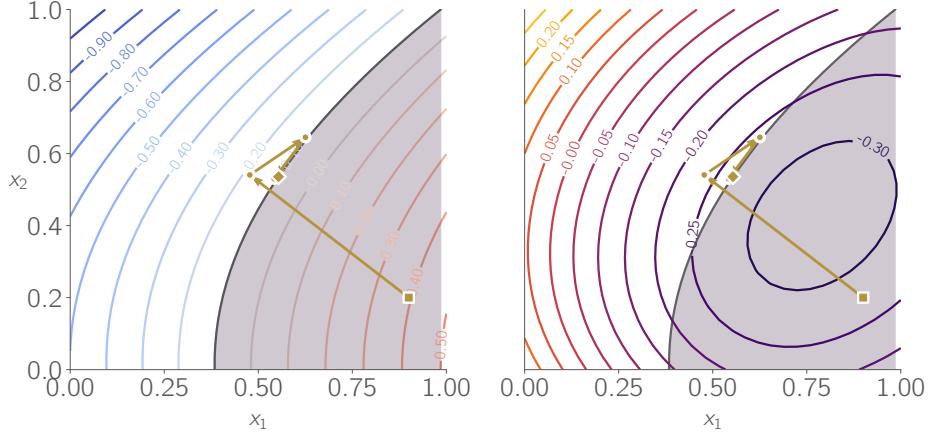


Fig. 5 Contours of the dominant eigenvalue real part (left) and objective function (right). The progress of the optimization is indicated by the arrows. The initial and optimized solutions are denoted by a square and a diamond, respectively. The infeasible region is shaded.

VII. Preliminary results for global linear stability theory (LST)

ADflow has not been previously used for LST. Therefore, we have used the canonical cylinder flow vortex shedding case to benchmark our results against literature. In this section, we present the LST for **ADflow** and results for the unsteady cylinder vortex shedding and steady base flow LST derived eigenfrequencies.

The governing equations for Navier–Stokes solved in **ADflow** can be summarized as

$$\mathbf{r}_k(\mathbf{w}) = 0, \quad (44)$$

where \mathbf{w} is the state vector and \mathbf{r}_k is the residual. For steady state base flow solution $\bar{\mathbf{w}}$, we have for a small increment in time

$$\frac{d\mathbf{w}}{dt} = -\mathbf{M}^{-1}\mathbf{r}_k(\mathbf{w}), \quad (45)$$

where \mathbf{M} is the mass matrix from the FVM solver. All terms on RHS are treated implicitly for the Newton’s method. In **ADflow**, the mass matrix is invoked into the residual by dividing by cell volumes. Therefore, the equation becomes

$$\frac{d\mathbf{w}}{dt} = -\mathbf{r}(\mathbf{w}), \quad (46)$$

where $\mathbf{r} = \mathbf{M}^{-1}\mathbf{r}_k$. Now, this can be expanded as

$$\begin{aligned} \frac{d(\bar{\mathbf{w}} + \delta\mathbf{w})}{dt} &= -\mathbf{r}(\bar{\mathbf{w}} + \delta\mathbf{w}), \\ \implies \frac{d\delta\mathbf{w}}{dt} &= -\left(\mathbf{r}(\bar{\mathbf{w}}) + \frac{\partial\mathbf{r}}{\partial\mathbf{w}}|_{\bar{\mathbf{w}}}\right), \\ \implies \frac{d\delta\mathbf{w}}{dt} &= -\mathbf{J}\delta\mathbf{w}, \end{aligned} \quad (47)$$

where we linearized and neglected the higher order terms. Assuming a modal ansatz $\delta\mathbf{w} = \Phi e^{\lambda t}$, and substituting that in the above equation, we get

$$-\mathbf{J}\Phi = \lambda\Phi. \quad (48)$$

This is the final equation which is an eigenvalue problem. Solving this equation gives us the eigenvalues. The eigenvalue can be written in the following manner

$$\lambda = \lambda_r + i\lambda_i, \quad (49)$$

where λ_r and λ_i are the real and complex-valued parts of the eigenvalue. For unstable flow exhibiting the Hopf bifurcation and entering into saturated limit-cycle oscillations, we typically see a complex conjugate pair of eigenvalues with a positive real part [10, 31].

It is a fact that **ADflow** non-dimensionalizes the governing equations. The eigenvalues sought are therefore non-dimensional. To re-dimensionalize the eigenvalues to help compare with literature, we must scale them by the factor

$$\lambda_s = \frac{\lambda}{t_{ref}}, \quad (50)$$

where λ_s is the scaled eigenvalue and

$$t_{ref} = L_{ref} \frac{1}{\sqrt{RT}}, \quad (51)$$

where R is the gas constant 287 J/kgK, T is the reference temperature 300 K and L_{ref} is set to a value of 1 m. λ_s is in rad/s. To convert it into physical frequency, we divide it by 2π .

A structured grid was generated for a cylinder of diameter 1 m. The grid is shown below in the first sub-figure (left) in Figure 6. The domain was discretized with 385 elements in both the circumferential and radial directions. A boundary layer grid near to the cylinder wall was made, as shown in the second sub-figure (right) in Figure 6. The off-wall spacing of the first cell normal to the cylinder is $\Delta r = 10^{-5}$ m. The grid is similar to that used in literature [32, 33].

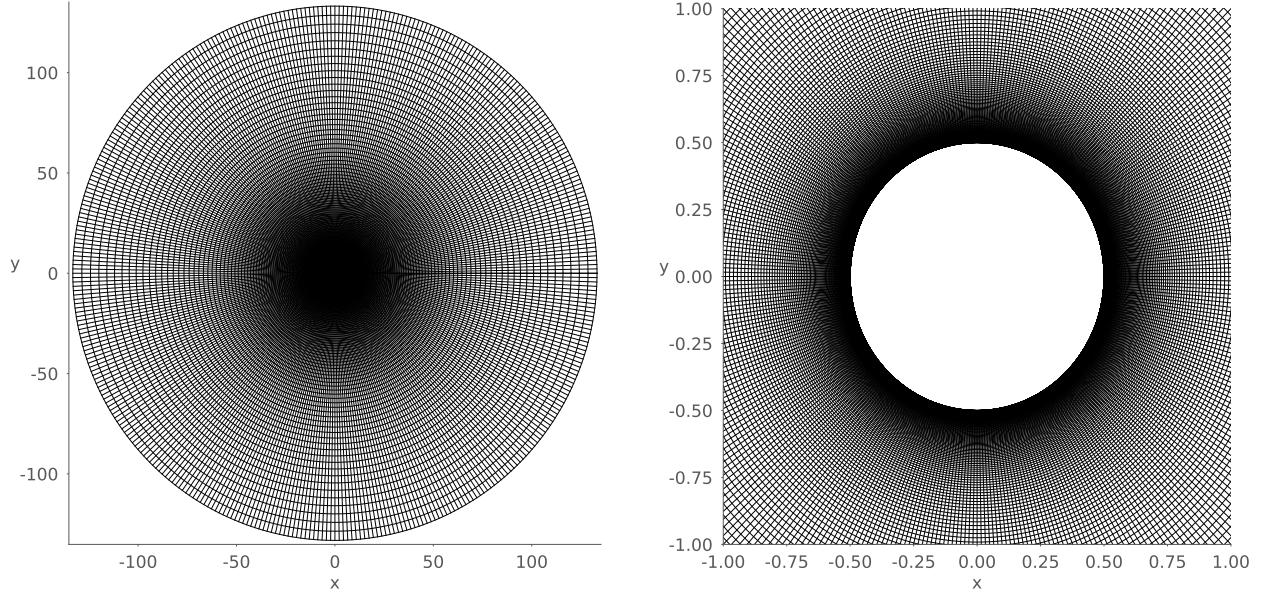


Fig. 6 The full structured grid for the cylinder flow (left) and the grid near the cylinder wall region zoomed in (right). A boundary layer grid is made with the first cell normal to the cylinder wall $\Delta r = 10^{-5}$ m. The cylinder diameter is set to 1 m.

Unsteady simulations were run with the laminar Navier Stokes model. The freestream conditions are Mach number of 0.1, Reynolds number in the range 46–120 and temperature of 300 K. A snapshot of the vortex shedding is shown in Figure 7, for a Reynolds number of 100. A classic Von–Karman vortex street is seen behind the cylinder and the flow is in a saturated limit cycle, as seen in the lift–coefficient C_L vs time plot in Figure 7. A small perturbation was made around the base flow, by rotating the cylinder five degrees in one second of physical time. This pushes the system into the saturated limit cycle oscillations.

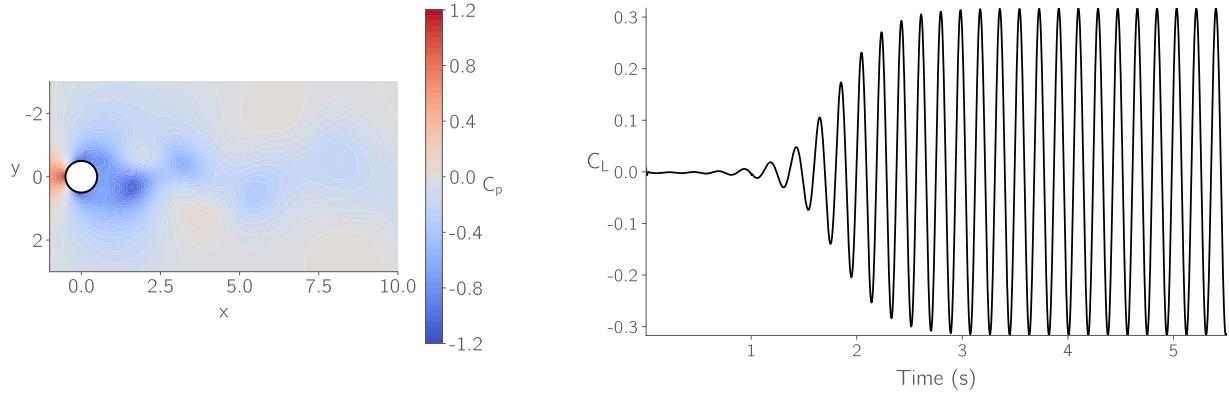


Fig. 7 Unsteady vortex shedding at Reynolds number 100 (left). The Von–Karman vortex street is seen formed inside the saturated limit cycle. The unsteady lift coefficient vs time plot is presented (right). Initial perturbations from the cylinder rotation about base flow grow and saturate into the limit cycle oscillations. The unsteady shedding frequency derived Strouhal number is computed to be 0.163 for Reynolds number 100.

The Strouhal number is defined as

$$St = \frac{f}{LU}, \quad (52)$$

where $L = L_{ref} = 1\text{m}$, $U = 34.718 \text{ m/s}$ and f is the vortex shedding frequency in Hz. In our simulations, we measure this frequency as the frequency of one complete oscillation of the lift-coefficient when the flow has entered into a fully saturated limit cycle.

A plot of the Strouhal number vs Reynolds number is shown in Figure 8 below. We compare our results with Williamson's results [34]. Our results match appreciably with the Hopf-bifurcation diagram from literature.

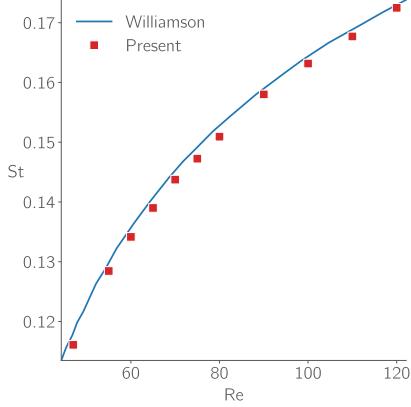


Fig. 8 A plot of the Strouhal numbers evaluated at different Reynolds numbers for the cylinder vortex shedding case. A comparison is made against Williamson's results [34] and the data agrees appreciably.

Finally, we present a plot of the eigenfrequencies from our LST with ADflow in Figure 9. We compare our results with Marquet et al [10]. Our results match to almost machine precision with their results. It should also be noted that the first unstable eigenpair was observed at the critical Reynolds number 46.85, which is very close to Marquet et al [10] who reported the critical Reynolds number to be 46.8 ± 0.05 . We also present the modes \hat{u} and \hat{v} , the x and y velocity modes in Figure 10. Complex eigenmodes can be scaled and rotated in the complex plane for the same eigenvalues [35] and therefore are not expected to always match with the results from other solvers. The difference in the modes is typically a result of normalization factor used, or whether the mode was at all normalized to begin with. Nevertheless, the mode shapes match appreciably.

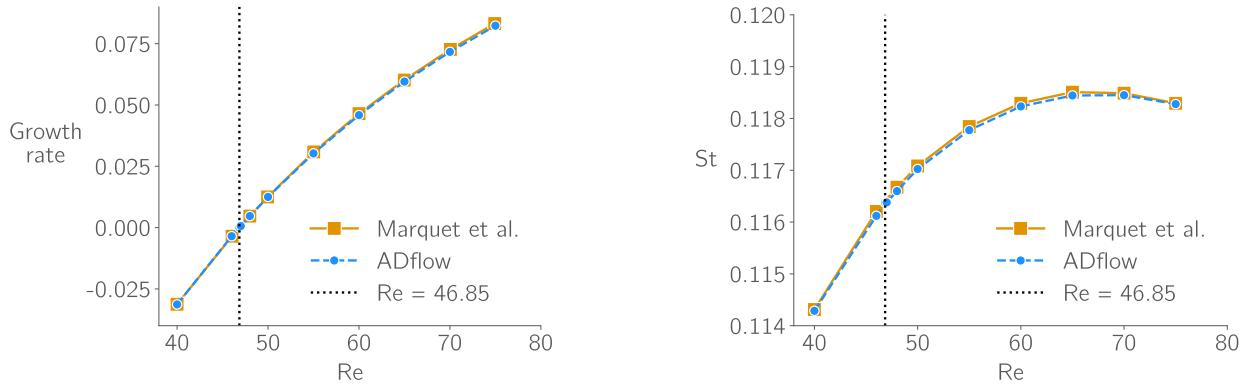


Fig. 9 The growth rates (left) and the eigenfrequency derived Strouhal numbers (right) for the base flow computed at different Reynolds numbers. Results from ADflow are benchmarked against those of Marquet et al. [10], which seem to match appreciably. The critical Reynolds number is found to be 46.85.

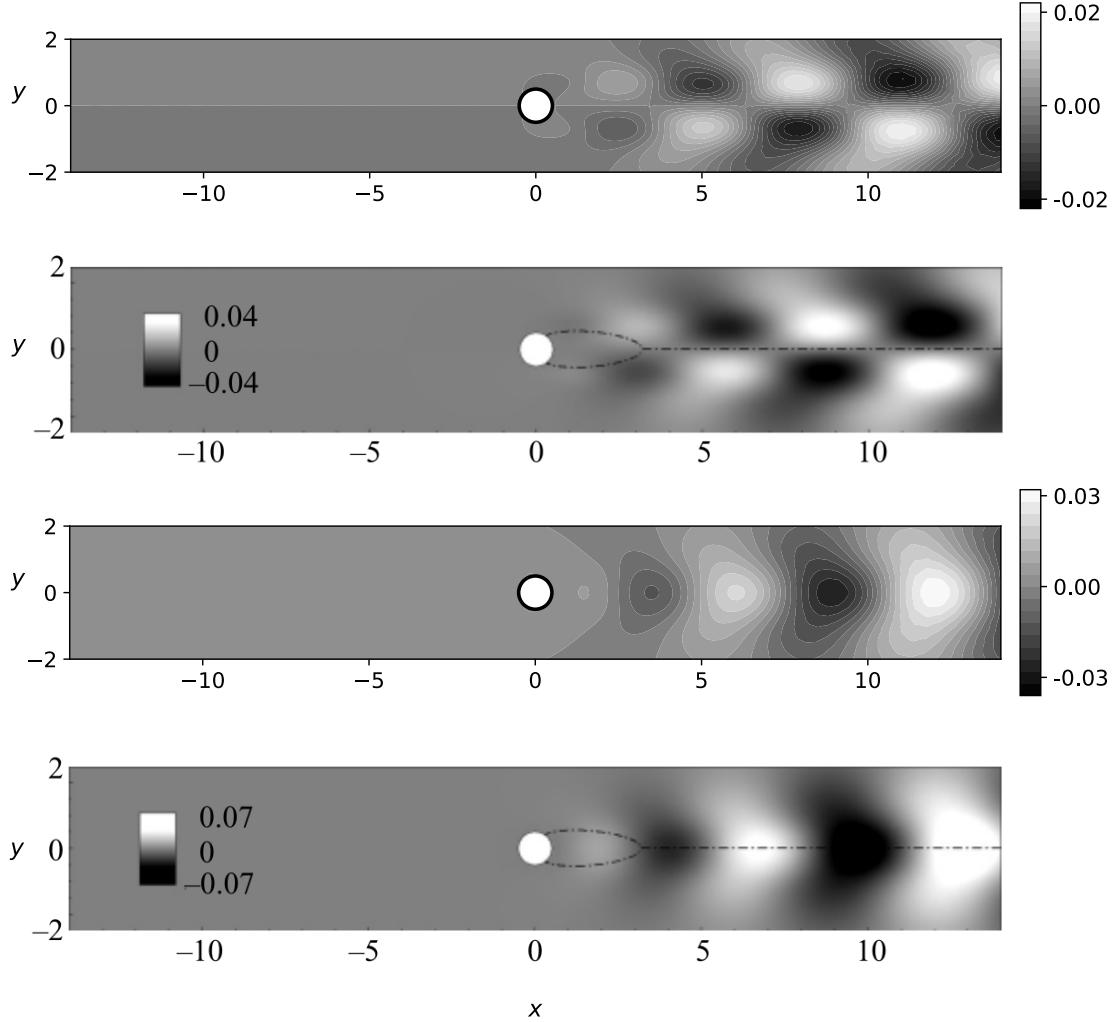


Fig. 10 Unstable eigenmodes from ADflow (from top) are shown in the first and third figures for \hat{u} and \hat{v} respectively, at the critical Reynolds number 46.85 evaluated in the current study. The results from Marquet et al. [10] are shown in the second and fourth figures for \hat{u} and \hat{v} respectively, at the critical Reynolds number 46.8 evaluated in their study.

All simulations and results were converged to tolerance ($< 10^{-12}$), including the eigensolver used—**SLEPc**, using the shift–invert strategy. The complex part of eigenfrequencies (presented as Strouhal number) do not match with the unsteady Strouhal numbers as shown in Figures 8 and 9. This is a classic case of mean flow and base flow being different, leading to different unstable unsteady frequencies and base flow eigenfrequencies [36, 37].

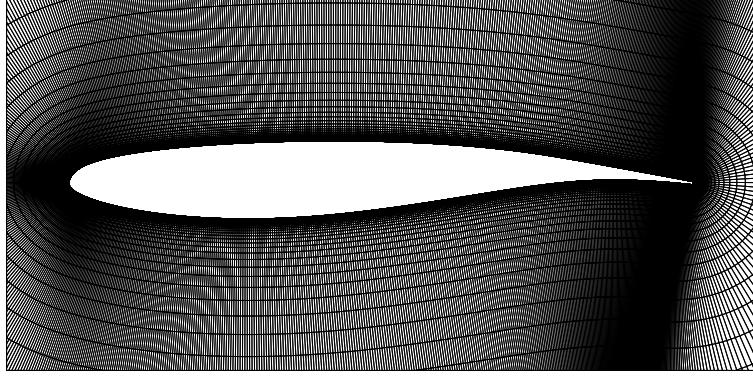


Fig. 11 The full structured grid for the OAT15A transonic airfoil zoomed in on the airfoil wall. A boundary layer grid is made with the first cell normal to the cylinder wall $\Delta r = 10^{-6}$ m. The airfoil chord is set to 1 m.

Next, we present the simulation for the transonic buffet 2-D case of OAT15A airfoil [38] and present the grid in Figure 11. The airfoil chord is 1m and We simulated it for Reynolds number 3.2×10^6 , Mach number of 0.73 and angle of attack 4° . The unsteady results are shown in Figures 12 and 13, where we present the full unsteady buffet cycle and density contours over one-limit cycle and the unsteady lift coefficient vs time plot.

It is observed that with a cold-start for the simulation, the system readily jumps into a limit-cycle oscillation. The Strouhal number for this setup is computed to be 0.074, which is close to the value reported by Sartor et al [39]. LST was performed on the transonic buffet case by simulating at same conditions for steady base flow. The base flow and the unstable eigenmode are presented in Figure 14. The unstable non-dimensional eigenvalues for this base flow are $0.1 \pm 0.445j$. The Strouhal number from this eigenfrequency is computed to be 0.045, which slightly under predicts the eigenfrequency computed by Sartor et al [39].

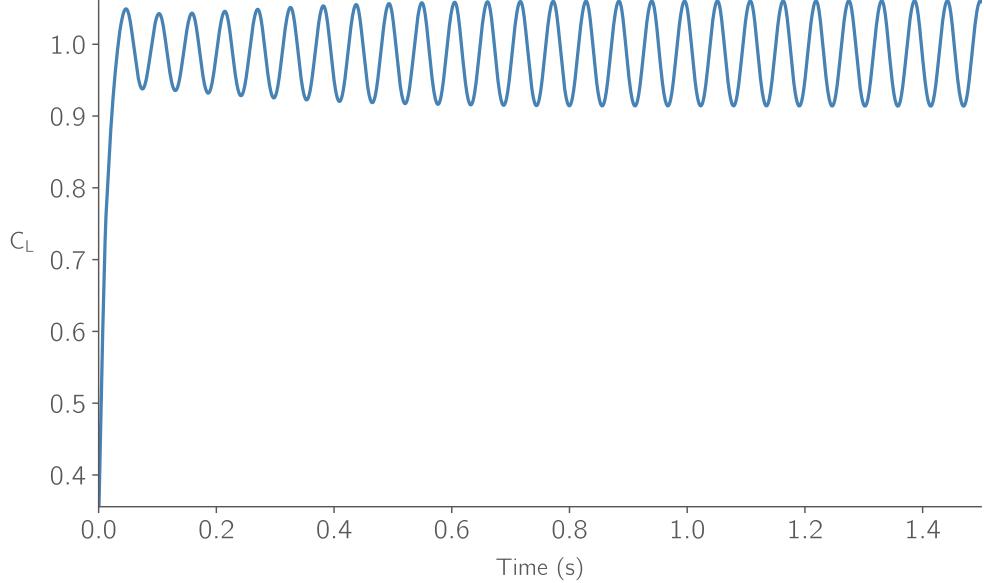


Fig. 12 The unsteady transonic buffetting oscillations presented for OAT15A airfoil at Mach number 0.73, Reynolds number 3.2×10^6 angle of attack 4.0° . The system readily enters into the limit cycle from cold-started simulation. The Strouhal number is computed to be 0.074.

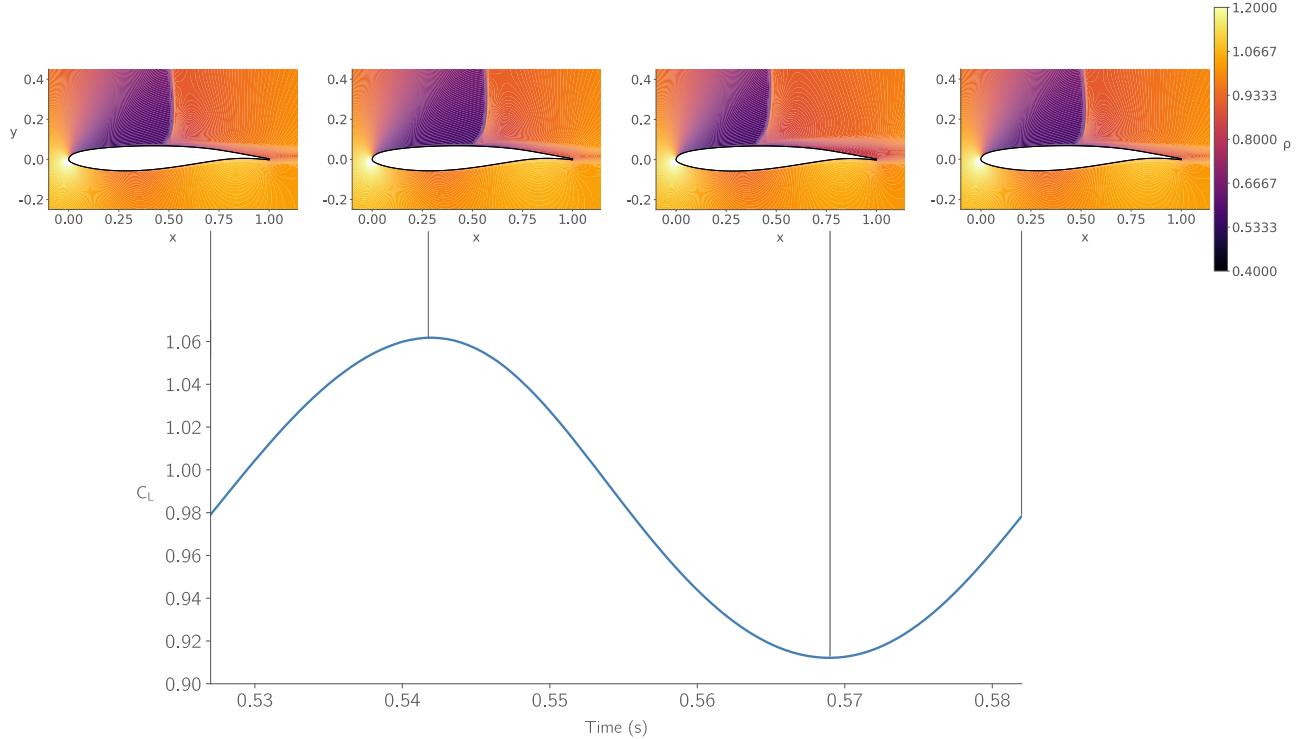


Fig. 13 The unsteady simulation density contours over one cycle in the saturated limit cycle oscillations for the OAT15A transonic airfoil. Freestream conditions are Mach number 0.73, Reynolds number 3.2×10^6 angle of attack 4.0° . Four plots have been made, each at distinct physical times on the limit cycle, indicated by vertical solid black lines from the sinusoidal wave. The Strouhal number for this unsteady transonic buffet is 0.074.

It is a fact that the OAT15A transonic buffet phenomenon is weakly non-linear [39–41]. This means that the mean and base flows are almost identical, leading to the same Strouhal numbers computed from the unsteady limit cycle frequency and the base flow's LST derived eigenfrequency. Our current focus is to investigate this with ADflow and benchmark the LST results for transonic buffet. The unstable eigenmode for this pair of eigenvalues is shown in Figure 14. The shock region right in front of the shock, aft shock, shock foot and the separation bubble appear in the unstable eigenmode. Of all the regions mentioned, the shock by itself dominates over the other flow structures of the eigenmode in magnitude. These flow structures form part of the leading instability mechanism for transonic buffet.

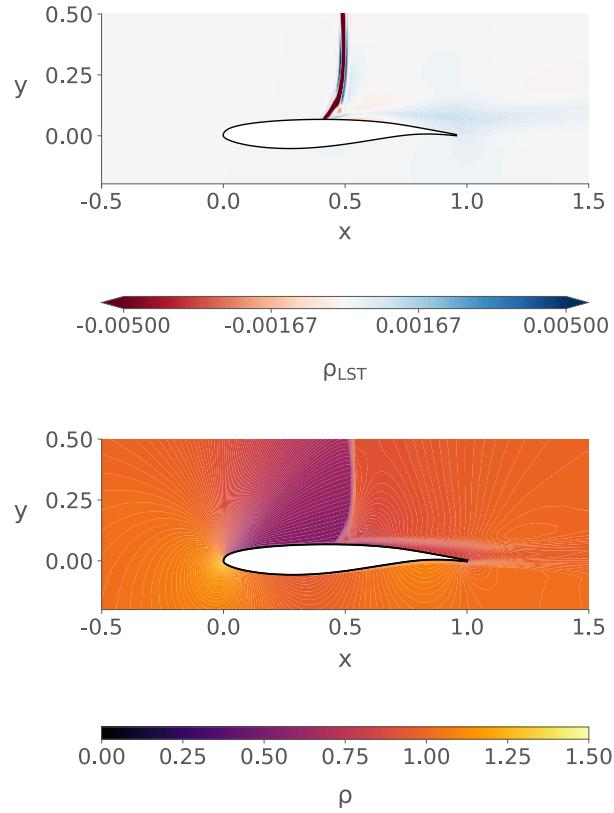


Fig. 14 The density contour for base flow of OAT15A transonic airfoil for Mach number 0.73, Reynolds number 3.2×10^6 angle of attack 4.0° (bottom) and its corresponding unstable eigenmode (top). The regions of the shock, shock foot and separation bubble appear in the unstable eigenmode, indicating the leading flow structures in the instability phenomenon.

References

- [1] Church, K. E. M., and Liu, X., “Cost-Effective Robust Stabilization and Bifurcation Suppression,” *SIAM Journal on Control and Optimization*, Vol. 57, No. 3, 2019, pp. 2240–2268. doi:[10.1137/18M1213142](https://doi.org/10.1137/18M1213142).
- [2] Plante, F., Dandois, J., and Laurendeau, É., “ZDES and URANS Simulations of 3D Transonic Buffet Over Infinite Swept Wings,” *Progress in Hybrid RANS-LES Modelling*, Springer International Publishing, 2019, pp. 259–269. doi:[10.1007/978-3-030-27607-2_21](https://doi.org/10.1007/978-3-030-27607-2_21).
- [3] Thomas, J., and Dowell, E. H., “Methodology for Numerically Stabilizing a Harmonic Balance Based Aeroelastic Solution Approach,” *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, 2020.
- [4] 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, 2002, pp. 638–646. doi:[10.2514/2.1720](https://doi.org/10.2514/2.1720), URL <http://dx.doi.org/10.2514/2.1720>.
- [5] Li, H., and Ekici, K., “A Novel Approach for Flutter Prediction of Pitch–Plunge Airfoils Using an Efficient One-Shot Method,” *Journal of Fluids and Structures*, Vol. 82, 2018, pp. 651–671. doi:[10.1016/j.jfluidstructs.2018.08.012](https://doi.org/10.1016/j.jfluidstructs.2018.08.012), URL <https://doi.org/10.1016/j.jfluidstructs.2018.08.012>.
- [6] He, S., Jonsson, E., Mader, C. A., and Martins, J. R. R. A., “Coupled Newton–Krylov Time-Spectral Solver for Flutter and Limit Cycle Oscillation Prediction,” *AIAA Journal*, Vol. 59, No. 6, 2021, pp. 2214–2232. doi:[10.2514/1.J059224](https://doi.org/10.2514/1.J059224).
- [7] Li, J., He, S., Zhang, M., Martins, J. R. R. A., and Khoo, B. C., “Physics-Based Data-Driven Buffet-Onset Constraint for Aerodynamic Shape Optimization,” *AIAA Journal*, Vol. 60, No. 8, 2022, pp. 4775–4788. doi:[10.2514/1.J061519](https://doi.org/10.2514/1.J061519).
- [8] Timme, S., “Global instability of wing shock-buffet onset,” *Journal of Fluid Mechanics*, Vol. 885, 2020, p. A37. doi:[10.1017/jfm.2019.1001](https://doi.org/10.1017/jfm.2019.1001).
- [9] Kuznetsov, Y. A., *Elements of Applied Bifurcation Theory*, Springer New York, 2004. doi:[10.1007/978-1-4757-3978-7](https://doi.org/10.1007/978-1-4757-3978-7).
- [10] MARQUET, O., SIPP, D., and JACQUIN, L., “Sensitivity analysis and passive control of cylinder flow,” *Journal of Fluid Mechanics*, Vol. 615, 2008, pp. 221–252. doi:[10.1017/s0022112008003662](https://doi.org/10.1017/s0022112008003662).
- [11] Mettot, C., Renac, F., and Sipp, D., “Computation of Eigenvalue Sensitivity to Base Flow Modifications in a Discrete Framework: Application to Open-Loop Control,” *Journal of Computational Physics*, Vol. 269, 2014, pp. 234–258. doi:[10.1016/j.jcp.2014.03.022](https://doi.org/10.1016/j.jcp.2014.03.022).
- [12] Shi, Y., Mader, C. A., He, S., Halila, G. L. O., and Martins, J. R. R. A., “Natural Laminar-Flow Airfoil Optimization Design Using a Discrete Adjoint Approach,” *AIAA Journal*, Vol. 58, No. 11, 2020, pp. 4702–4722. doi:[10.2514/1.J058944](https://doi.org/10.2514/1.J058944).
- [13] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations,” *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951. doi:[10.2514/1.J052255](https://doi.org/10.2514/1.J052255).
- [14] Boullié, N., Farrell, P. E., and Rognes, M. E., “Optimal Control of Hopf Bifurcations,” , 2022. doi:[10.48550/arXiv.2201.11684](https://arxiv.org/abs/2201.11684), URL <https://arxiv.org/abs/2201.11684>.
- [15] Boullié, N., Farrell, P. E., and Paganini, A., “Control of Bifurcation Structures Using Shape Optimization,” *SIAM Journal on Scientific Computing*, Vol. 44, No. 1, 2022, pp. A57–A76. doi:[10.1137/21M1418708](https://doi.org/10.1137/21M1418708).
- [16] He, S., Shi, Y., Jonsson, E., and Martins, J. R. R. A., “Eigenvalue Problem Derivatives Computation for a Complex Matrix Using the Adjoint Method,” *Mechanical Systems and Signal Processing*, Vol. 185, 2023, p. 109717. doi:[10.1016/j.ymssp.2022.109717](https://doi.org/10.1016/j.ymssp.2022.109717).
- [17] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., “The Complex-Step Derivative Approximation,” *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, 2003, pp. 245–262. doi:[10.1145/838250.838251](https://doi.org/10.1145/838250.838251).
- [18] Slotine, J.-J. E., and Li, W., *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, NJ, 1991.
- [19] Elman, H. C., Meerbergen, K., Spence, A., and Wu, M., “Lyapunov Inverse Iteration for Identifying Hopf Bifurcations in Models of Incompressible Flow,” *SIAM Journal on Scientific Computing*, Vol. 34, No. 3, 2012, pp. A1584–A1606. doi:[10.1137/110827600](https://doi.org/10.1137/110827600).

- [20] Timme, S., Badcock, K. J., Wu, M., and Spence, A., “Lyapunov Inverse Iteration for Stability Analysis Using Computational Fluid Dynamics,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, American Institute of Aeronautics and Astronautics, 2012. doi:[10.2514/6.2012-1547](https://doi.org/10.2514/6.2012-1547).
- [21] Roose, D., and Hlavaček, V., “A Direct Method for the Computation of Hopf Bifurcation Points,” *SIAM Journal on Applied Mathematics*, Vol. 45, No. 6, 1985, pp. 879–894. doi:[10.1137/0145053](https://doi.org/10.1137/0145053).
- [22] Lambe, A. B., and Martins, J. R. R. A., “Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes,” *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 273–284. doi:[10.1007/s00158-012-0763-y](https://doi.org/10.1007/s00158-012-0763-y).
- [23] Zhu, Y., Wang, Y., Zhang, X., and Kang, Z., “A New Form of Forbidden Frequency Band Constraint for Dynamic Topology Optimization,” *Structural and Multidisciplinary Optimization*, Vol. 65, No. 4, 2022. doi:[10.1007/s00158-022-03220-1](https://doi.org/10.1007/s00158-022-03220-1).
- [24] Linnainmaa, S., “Taylor Expansion of the Accumulated Rounding Error,” *BIT*, Vol. 16, No. 2, 1976, pp. 146–160. doi:[10.1007/BF01931367](https://doi.org/10.1007/BF01931367).
- [25] Martins, J. R. R. A., and Ning, A., *Engineering Design Optimization*, Cambridge University Press, Cambridge, UK, 2022. doi:[10.1017/9781108980647](https://doi.org/10.1017/9781108980647), URL <https://mdobook.github.io>.
- [26] Lyu, Z., Xu, Z., and Martins, J. R. R. A., “Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization,” *Proceedings of the 8th International Conference on Computational Fluid Dynamics*, Chengdu, Sichuan, China, 2014. ICCFD8-2014-0203.
- [27] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., “OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization,” *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. doi:[10.1007/s00158-019-02211-z](https://doi.org/10.1007/s00158-019-02211-z).
- [28] Yildirim, A., Gray, J. S., Mader, C. A., and Martins, J. R. R. A., “Coupled Aeropropulsive Design Optimization of a Podded Electric Propulsor,” *AIAA Aviation Forum*, 2021. doi:[10.2514/6.2021-3032](https://doi.org/10.2514/6.2021-3032).
- [29] Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP algorithm for large-scale constrained optimization,” *SIAM Journal of Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:[10.1137/S1052623499350013](https://doi.org/10.1137/S1052623499350013).
- [30] Wu, N., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R. R. A., “pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems,” *Journal of Open Source Software*, Vol. 5, No. 54, 2020, p. 2564. doi:[10.21105/joss.02564](https://doi.org/10.21105/joss.02564).
- [31] Sahin, M., and Owens, R. G., “A numerical investigation of wall effects up to high blockage ratios on two-dimensional flow past a confined circular cylinder,” *Physics of Fluids*, Vol. 16, No. 5, 2004, pp. 1305–1320. doi:[10.1063/1.1668285](https://doi.org/10.1063/1.1668285).
- [32] Rajani, B. N., Kandasamy, A., and Majumdar, S., “Numerical Simulation of Laminar Flow Past a Circular Cylinder,” *Applied Mathematical Modelling*, Vol. 33, No. 3, 2009, pp. 1228–1247. doi:[10.1016/j.apm.2008.02.019](https://doi.org/10.1016/j.apm.2008.02.019).
- [33] Franke, R., Rodi, W., and Schönung, B., “Numerical Calculation of Laminar Vortex-Shedding Flow Past Cylinders,” *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 35, 1990, pp. 237–257. doi:[10.1016/0167-6105\(90\)90219-3](https://doi.org/10.1016/0167-6105(90)90219-3).
- [34] Williamson, C. H. K., “Vortex Dynamics in the Cylinder Wake,” *Annual Review of Fluid Mechanics*, Vol. 28, No. 1, 1996, pp. 477–539. doi:[10.1146/annurev.fl.28.010196.000240](https://doi.org/10.1146/annurev.fl.28.010196.000240).
- [35] He, S., Shi, Y., Jonsson, E., and Martins, J. R. R. A., “Eigenvalue problem derivatives computation for a complex matrix using the adjoint method,” *Mechanical Systems and Signal Processing*, Vol. 185, 2023, p. 109717. doi:[10.1016/j.ymssp.2022.109717](https://doi.org/10.1016/j.ymssp.2022.109717).
- [36] Barkley, D., “Linear analysis of the cylinder wake mean flow,” *Europhysics Letters (EPL)*, Vol. 75, No. 5, 2006, pp. 750–756. doi:[10.1209/epl/i2006-10168-7](https://doi.org/10.1209/epl/i2006-10168-7).
- [37] SIPP, D., and LEBEDEV, A., “Global stability of base and mean flows: a general approach and its applications to cylinder and open cavity flows,” *Journal of Fluid Mechanics*, Vol. 593, 2007, pp. 333–358. doi:[10.1017/s0022112007008907](https://doi.org/10.1017/s0022112007008907).
- [38] Jacquin, L., Molton, P., Deck, S., Maury, B., and Soulevant, D., “Experimental Study of Shock Oscillation over a Transonic Supercritical Profile,” *AIAA Journal*, Vol. 47, No. 9, 2009, pp. 1985–1994. doi:[10.2514/1.30190](https://doi.org/10.2514/1.30190).

- [39] Sartor, F., Mettot, C., and Sipp, D., “Stability, Receptivity, and Sensitivity Analyses of Buffeting Transonic Flow over a Profile,” *AIAA Journal*, Vol. 53, No. 7, 2015, pp. 1980–1993. doi:[10.2514/1.j053588](https://doi.org/10.2514/1.j053588).
- [40] CROUCH, J. D., GARBARUK, A., MAGIDOV, D., and TRAVIN, A., “Origin of transonic buffet on aerofoils,” *Journal of Fluid Mechanics*, Vol. 628, 2009, pp. 357–369. doi:[10.1017/s0022112009006673](https://doi.org/10.1017/s0022112009006673).
- [41] Crouch, J., Ahrabi, B., and Kamenetskiy, D., “Weakly nonlinear behaviour of transonic buffet on airfoils,” *Journal of Fluid Mechanics*, Vol. 999, 2024. doi:[10.1017/jfm.2024.499](https://doi.org/10.1017/jfm.2024.499).