This is a preprint of the following article, which is available from http://mdolab.engin.umich.edu
Y. Shi, Charles. A. Mader, S. He, G. L. O. Halila and J. R. R. A. Martins. Natural Laminar-Flow Airfoil Optimization Design Using a Discrete Adjoint Approach. *AIAA Journal*, 2020 (In press)
The published article may differ from this preprint and is available at:
https://doi.org/10.2514/1.J058944

Natural Laminar-Flow Airfoil Optimization Design Using a Discrete Adjoint Approach

Yayun Shi, Charles A. Mader, Sicheng He, Gustavo L. O. Halila
and Joaquim R. R. A. Martins
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109

Abstract

Natural laminar-flow wings are one of the most promising technologies for reducing fuel burn and emissions for commercial aviation. However, there is a lack of tools for performing shape optimization of wings based on computational fluid dynamics considering laminar-to-turbulent transition. To address this need, we develop a discrete adjoint-based optimization framework where transition is modeled. The core of this framework is a Reynolds-averaged Navier–Stokes solver that is coupled with a simplified e^N method to predict Tollmien–Schlichting and laminar separation-induced transition that consists of a laminar boundary-layer code and a database method for flow stability analysis. The transition prediction is integrated with a Spalart–Allmaras turbulence model through a smoothed intermittency function, which makes it suitable for gradient-based optimization. A coupled-adjoint approach that uses transpose Jacobian-vector products derived via automatic differentiation computes the transition prediction derivatives. Lift-constrained drag minimizations of airfoils for a single-point design and a multipoint design problem are performed. The results show that the optimizer successfully reduces the drag coefficient by increasing the extent of laminar flow. The multipoint optimization formulation produces an airfoil with a significant amount of laminar flow that is maintained at several flight conditions. The proposed methods make it possible to perform aerodynamic shape optimization considering laminar-to-turbulent transition in airfoil optimization.

1 Introduction

Aircraft designers are constantly looking for ways to reduce drag as a means to reduce aircraft fuel burn. One of the most promising technologies available to address this need is to develop aircraft with more laminar flow over the aircraft's surface [1]. The

friction drag for a laminar boundary layer is 90% lower than for a turbulent one. Since friction drag accounts for up to 50% of the total aircraft drag in cruise [2], there is a significant potential for drag reduction through increased laminar flow.

While laminar-flow control techniques are being developed, natural laminar flow (NLF) is a far less complex solution for increasing the extent of laminar flow and has been already implemented in a few aircraft designs. The Piaggio P180 can maintain 50% and 30% laminar region on the wing and the fuselage, respectively, at the cruise condition [3]. With NLF, the drag is reduced by 20–25% compared with the conventional tractor turboprops. The HondaJet is designed with a favorable pressure gradient around the front 43% chord on the wing and gains 40% chord laminar region [4]. Boeing has also achieved NLF on the 787 nacelle and the 737 Max winglet [5, 6] [1], and Airbus has investigated laminar flow on a modified A340 in the Breakthrough Laminar Aircraft Demonstrator (BLADE) project by designing a wing tip with a transonic laminar profile and a real internal primary structure [2].

Because laminar-to-turbulent transition is driven by multiple physical phenomena, it is important to develop accurate, robust, and efficient computational fluid dynamics (CFD)-based prediction tools [7–9]. In addition, it is desirable to develop design optimization approaches that yield shapes that exhibit significant amounts of laminar flow [10, 11].

The laminar-to-turbulent transition in the boundary layer is the result of a sequence of complex phenomena. These phenomena depend on Reynolds number, pressure gradient, disturbance environment, and configuration surface roughness [12–15], which makes developing a rigorous model of the transition process challenging. As a result, industrial CFD applications focus on the main transition mechanisms, such as Tollmien–Schlichting (TS) waves, crossflow (CF) vortices, separation-induced transition, and attachment-line transition (ALT). Focusing on these key mechanisms has allowed the development of several useful transition prediction models [8, 16–19].

Reynolds-averaged Navier–Stokes (RANS) turbulence modeling approaches are the most widely used tools in industrial applications because they are a good compromise between accuracy and computational cost. Thus, many RANS-based transition prediction methods have been developed [8, 18–21]. There are two main types of transition prediction methods: transport equation modeling, and e^N-based methods.

The transport equation modeling concept is solely based on local parameters, making the approach compatible with modern CFD solvers. The most prominent transport equation model for transition prediction is the correlation-based equation $\left(\gamma - \widetilde{Re}_{\theta t}\right)$ proposed by Langtry and Menter [8]. The original model evaluates streamwise transition, including natural, bypass, and separation-induced transition mechanisms. To capture CF-induced transition, other researchers have proposed modifications to the Langtry–Menter method [22–24]. However, the Langtry–Menter method and its modifications do not consider the physically relevant upstream flow history, which limits its application in industry [7].

[1] http://www.boeing.com/commercial/737max/737-max-winglets [retrieved July, 2018]

[2] https://www.airbus.com/newsroom/press-releases/en/2017/09/airbus_-_blade_-laminar-flow-wing-demonstrator-makes-first-fligh.html [retrieved May, 2019]

As an alternative, Coder and Maughmer [7] proposed the amplification factor transport (AFT) model. The AFT model uses the approximate envelope method (e^N-based) [25] and takes the envelope amplification factor as a scalar variable, which makes the approach appealing for use in parallel CFD applications. Coder [20] improved this model to be an entirely localized and Galilean invariant by introducing a new local shape factor. Unfortunately, both the original and extended AFT models can only predict transition induced by the TS waves, whereas CF-triggered transition, which is relevant for swept wings, is not included in the formulation.

In contrast to the transport equation models, e^N-based transition methods typically couple RANS codes with external transition modules. The RANS code predicts the flow based on an assumed transition location, and the transition model uses the RANS flow solution to estimate the transition location. Thus, an e^N-based transition framework requires multiple components and iteration between the RANS solver and the transition module. Both TS waves and CF vortices can be predicted by e^N-based methods, and these methods have been used in various industrial applications [26–29]. These methods require boundary-layer information (velocity profile, temperature profile, pressure profile) and stability analysis.

The boundary-layer flow information is obtained from the flow solutions by one of two means: 1) from a boundary layer well resolved by the RANS solver, where the mean flow profiles are extracted directly from the field solutions; or 2) the surface pressure distribution from the RANS solution is imposed as the boundary conditions for a boundary-layer code to generate mean flow profiles. Directly resolving the boundary layer with the RANS solver is compatible with three-dimensional configurations, but it needs a finer mesh, especially in the surface wall-normal direction. On the other hand, if the flow three-dimensionality is not strong, the wall pressure distribution can be used under a conical wing hypothesis as an input for the boundary-layer code to compute the boundary-layer velocity profiles and the corresponding integral parameters.

Possible flow stability analysis approaches for the e^N method include linear stability theory (LST) [12, 18, 19], a database method based on LST [30, 31], and linear parabolized stability equations [32]. The LST method is derived from the linearized unsteady Navier–Stokes equations and neglects non-local and non-parallel effects. The LST approach is time consuming and demands specific knowledge, as well as user intervention.

Motivated by the need for a simple, robust, accurate, and fast prediction tool, Perraud et al. [31] developed a database method. The general idea for this simplified method is to use the exact LST method to establish the analytical relationship between the disturbance amplification rate and boundary-layer parameters. With this relationship, we can obtain the amplification rate by finding the stability results in the database directly. The computational time is substantially reduced compared to exact LST computation.

Bégou et al. [33] also used the database method with a RANS solver. This method was successfully applied to the ONERA-D airfoil in incompressible conditions and an industrial laminar airfoil at transonic conditions. Good agreements between experimental data and the numerical transition predictions were obtained, which demonstrated the accuracy of the database transition prediction method.

Different combinations of boundary-layer model and stability analysis methods are possible: RANS velocity field with a database method, boundary-layer code with LST, and a boundary-layer code with a database method. For instance, the TAU unstructured CFD solver includes RANS field velocities, the database method, and LST code as options to account for transition to turbulence effects [18, 19]. Shi et al. [28] implemented an automatic transition prediction tool based on the e^N method in the open-source CFD solver ADflow [3] [34, 35].

In recent years, gradient-based optimization techniques have been used for CFD-based aircraft shape optimization with large numbers of design variables, thanks to efficient gradient computation via the adjoint method [35–41]. To this end, the adjoint method has been implemented in several large-scale CFD solvers [35, 41–44]. Because the extent of laminar flow is strongly dependent on the local surface shape, extending the region of laminar flow requires a large number of design parameters. Therefore, an adjoint-based approach is necessary to extend the laminar-flow region and to minimize the drag. This type of optimization is based on either a transport equation model or an e^N-based method. Zhang et al. [45] added the $\gamma - \widetilde{Re}_{\theta t}$ model into a CFD framework and used a continuous adjoint approach to perform optimization in a turbomachinery application. This adjoint-based framework considers linearized transition and a frozen $\gamma - \widetilde{Re}_{\theta t}$ model.

Lyu et al. [46] compared aerodynamic shape optimization results with both frozen-turbulence and full-turbulence adjoint derivatives and concluded that the full-turbulence results were more accurate than those obtained with the frozen-turbulence assumption for optimizations with fully-turbulent simulations. The frozen-turbulence approach neglects the turbulence contribution to the main flow in the adjoint system, while the frozen-transition method neglects the transition properties in the main flow. Therefore, the frozen-transition modeling is also likely to affect the accuracy of the derivatives computations.

Khayatzadeh and Nadarajah [47] implemented the $\gamma - \widetilde{Re}_{\theta t}$ model into a two-dimensional optimization framework using a discrete adjoint method, where the derivatives of the turbulence model and transition model were obtained by hand differentiation. This study did not show detailed derivative accuracy verification by either complex step or finite difference.

Lee and Jameson [48, 49] coupled an adjoint-based RANS solver with an e^N database method for NLF shape optimization design. In the coupled system, they did not consider the gradients of the transition module and focus on eliminating the shock wave to reduce the drag. This causes the optimized wing to have an even shorter laminar region than the initial configuration, meaning that the full drag reduction potential of a laminar-flow wing is not realized.

More recently, Driver and Zingg [50] coupled a two-dimensional Newton–Krylov (NK) RANS solver with a discrete adjoint to the MSES solver [51], which includes laminar-to-turbulent transition capabilities. The derivatives of the objective function with respect to the transition locations and the derivatives of the transition locations with respect to design variables were computed using finite difference method in MSES.

[3] https://github.com/mdolab/adflow [retrieved September, 2020]

They then used this optimization approach to maximize airfoil lift. Rashad and Zingg [10] continued to develop this laminar flow design capability [50] . Based on the original adjoint equations, they augmented the transition adjoint formulation. This produces a coupled-adjoint equation involving the two-state RANS-transition formulation. The gradients were verified by various gradient computation techniques and good agreement was reported. The optimization results show that this approach is a robust laminar flow design tool for airfoils.

In this work, we implement a laminar-to-turbulent optimization framework based on ADflow [35](see footnote [4]): an open-source RANS CFD solver that has been used extensively for aerodynamic shape optimization [40, 52–54]. The transition prediction is based on the e^N method [18, 19, 27]. Specifically, we use a boundary-layer code with a database method (simplified e^N) for its accuracy and efficiency. We extend the laminar flow prediction capability developed in Ref. [28] and apply it to a full set of airfoil optimizations.

The present work differs from previous efforts in that the coupled-adjoint implementation avoids any finite difference approximation, which improves accuracy and computational efficiency. To further improve the efficiency of the coupled-adjoint system for laminar-to-turbulent flow, we compute the derivatives in a Jacobian-free fashion with reverse-mode automatic differentiation (AD), which results in more accurate derivatives and efficient solution [35]. Another novel aspect is that we have formulated the transition model such that it yields smooth functions, making it more suitable for gradient-based optimization.

This paper is structured as follows. We start by introducing the transition prediction tool and the corresponding verification in Sec. 2. We then describe the coupled-adjoint equation solution and verification in Sec. 3. Airfoil optimization results are shown in Sec. 4 and summarized in Sec. 5.

2 Transition Prediction Methodology and Verification

The RANS-e^N transition prediction tool used in this work consists of the ADflow RANS solver [34, 35] and an e^N method. ADflow is a second-order finite volume CFD solver that solves for compressible flows over three-dimensional configurations, and it can handle structured multiblock and overset meshes [55, 56]. ADflow has several turbulence models, but for this study, we use the Spalart–Allmaras (SA) model exclusively [57]. The transition prediction module uses an e^N method and includes a laminar boundary-layer equation solution and the transition criteria. We use an intermittency function to model the laminar-to-turbulent transition process and couple the e^N method with the RANS solver in a robust iterative procedure. The transition prediction framework can predict TS waves and laminar separation-induced transition.

[4] https://github.com/mdolab/adflow [retrieved September, 2020]

2.1 Laminar Boundary-Layer Solver

Solving the laminar boundary-layer equations is an efficient and accurate way of computing the boundary-layer characteristics for two-dimensional geometries and high-aspect-ratio three-dimensional wings [58]. We develop a quasi-three-dimensional laminar boundary-layer solver under the conical assumption ($\partial p/\partial r = 0$, where the r is the distance along the generator from a conveniently placed origin) [59]. The laminar boundary-layer system includes the continuity, momentum, and energy equations, forming a system of parabolic partial differential equations [60].

We apply the Keller-box method to reformulate the nonlinear boundary-layer equations. Then, we use a second-order central-difference scheme for the normal-direction discretization and a second-order backward finite difference for the streamwise discretization. The nonlinear differential equations are solved using Newton's method [61], where each Newton step is solved with the block-elimination method in this finite difference solver.

The input parameters (mean flow) for the boundary-layer code include: Mach number M, pressure coefficient distribution C_p from the RANS solver, sweep angle Λ, Reynolds number Re, and the cross-sectional airfoil shape (x, y) at specified span-wise position z. The nonlinear laminar boundary-layer simulation residuals are

$$\boldsymbol{\mathcal{B}}(\bar{\boldsymbol{q}}) = 0 \tag{1}$$

where $\bar{\boldsymbol{q}}$ stands for the mean flow variables components $(\bar{u}, \bar{v}, \bar{w})$, pressure $\bar{p}$, and density $\bar{\rho}$.

We define a vector $\boldsymbol{d}_{\text{bl}}$ consisting of the displacement thickness Reynolds number Re_{δ_1}, momentum thickness Reynolds number Re_{δ_2}, shape factor H_{12}, local Reynolds number Re_x, density ρ_e, Mach number M_e, and viscosity coefficient μ_e at the boundary-layer edge. This vector is required by the flow stability analysis and is expressed as

$$\boldsymbol{d}_{\text{bl}} = F_{\boldsymbol{\mathcal{B}}}(\bar{\boldsymbol{q}}, C_{p_s}; \boldsymbol{x}_s) \tag{2}$$

where C_{p_s} is the cross-sectional profile pressure coefficient distribution, which is extracted from the RANS flow solution; and $\boldsymbol{x}_s$ stands for the airfoil surface coordinates.

2.2 Transition Criteria

On a swept wing, transition to the turbulent flow regime can be achieved through four transition mechanisms: ALT at the leading edge, TS waves in the streamwise direction, CF vortices in the direction normal to the potential velocity, and separation-induced transition [12]. Since we focus on airfoil optimization in this study, we only consider the TS wave and laminar separation transition mechanisms.

2.2.1 Simplified e^N Transition Prediction Method

Based on our LST code, we implement the database method proposed by Arnal [30] and improved by Perraud et al. [31]. The database method idea comes from the results obtained by studying the stability of a large number of three-dimensional velocity

profiles obtained from self-similar Falkner–Skan–Cooke (FSC) profiles [30]. The study considers the relationship between two variables: the displacement thickness Reynolds number Re_{δ_1}, and the dimensionless spatial amplification rate α_i ($\alpha_i = \alpha_i^* \delta_1$, where α_i^* is the spatial amplification rate) for a given reduced frequency F. The α_i versus Re_{δ_1} curve can be represented by two inverted parabolas, as shown in Fig. 1. Using this representation, α_i is computed directly with Re_{δ_1}. Thus, the database method dramatically reduces the computational cost compared to an exact LST computation, which would require solving the Orr–Sommerfeld equation.

The representation of $\alpha_i(Re_{\delta_1})$ using two inverted parabolas is derived as follows. For simplicity, we define the amplification rate as

$$\alpha_i = -\sigma = -\max(\sigma_I, \sigma_V) \tag{3}$$

where σ_I and σ_V represent the inviscid and viscous amplification rates defined as

$$\sigma_I(Re_{\delta_1}) = \sigma_{M_I}\left(1 - \left(\frac{Re_{\delta_1} - R_{M_I}}{R_{K_I} - R_{M_I}}\right)^2\right), \sigma_V(Re_{\delta_1}) = \sigma_{M_V}\left(1 - \left(\frac{Re_{\delta_1} - R_{M_V}}{R_{K_V} - R_{M_V}}\right)^2\right) \tag{4}$$

The amplified region is determined by R_{0_I}, R_{1_I} or R_{0_V}, R_{1_V}, whereas R_{M_I} and R_{M_V} define the peaks of the parabolas. When $\sigma \leqslant 0$, there is no amplification for the given condition case. Taking the maximum of these two parabolas results in σ, whose curve is highlighted in Fig. 1.

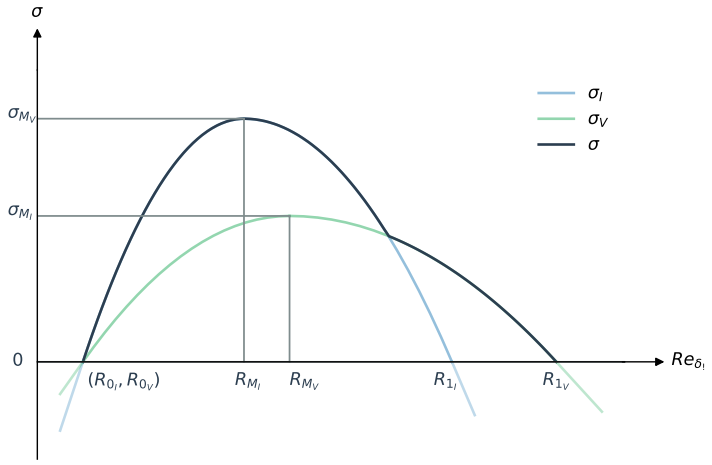


Figure 1: Parabolic model of the amplification rate for both the inviscid and viscous case [31].

To evaluate Eqs. (3) and (4), we need values for R_K and R_M. R_K is defined as:

$$R_{K_I} = \begin{cases} R_{0_I} & \text{if } Re_{\delta_1} < R_{M_I} \\ R_{1_I} & \text{if } Re_{\delta_1} > R_{M_I} \end{cases}, \quad R_{K_V} = \begin{cases} R_{0_V} & \text{if } Re_{\delta_1} < R_{M_V} \\ R_{1_V} & \text{if } Re_{\delta_1} > R_{M_V} \end{cases} \tag{5}$$

where subscript V denotes the viscous value at low speed and subscript I stands for the inviscid value at higher velocities or in strong adverse pressure gradients. The stable

region ($\sigma < 0$) is added by extrapolating the parabolas with a tangent line at the zero crossing point, as shown in Fig. 1.

The parameters σ_M, R_M, R_0, and R_1 vary with F; and they are approximated by the following relationships:

$$\begin{bmatrix} R_{M_I}(F) \\ R_{1_I}(F) \\ R_{0_I}(F) \\ \sigma_{M_I}(F) \end{bmatrix} = \begin{bmatrix} k_{1_I} F^{E_{1_I}} \\ k_{2_I} F^{E_{2_I}} \\ R_{M_V}\left(1 + C(F_c - F)\right) \\ B_I \left(1 - \frac{F}{F_{0_I}}\right) \end{bmatrix}, \quad \begin{bmatrix} R_{M_V}(F) \\ R_{1_V}(F) \\ R_{0_V}(F) \\ \sigma_{M_V}(F) \end{bmatrix} = \begin{bmatrix} k_{1_V} F^{E_{1_V}} \\ k_{2_V} F^{E_{2_V}} \\ R_{M_V}\left(1 + C(F_c - F)\right) \\ \min\left(B_V \left(1 - \frac{F}{F_{0_V}}\right), \sigma_{M_{V_{\max}}}\right) \end{bmatrix} \tag{6}$$

The Reynolds number bounds (R_{M_I}, R_{1_I}, R_{M_V}, R_{1_V}) for the unstable region and σ_{M_I}, σ_{M_V} are related to the reduced frequency F, which we define as

$$F = \frac{2\pi f^* \nu}{U_e^2} = \frac{\omega^* \nu \delta_1}{U_e \delta_1} = \omega / Re_{\delta_1} \tag{7}$$

where $\omega^* = 2\pi f^* / U_e = \omega / \delta_1$.

To build an estimate for α_i from Eqs. (3–5) we need estimates for the following 15 parameters:

$$\begin{aligned} &[B(M_e, H_{12}), F_0(M_e, H_{12}), k_1(M_e, H_{12}), E_1(M_e, H_{12}), k_2(M_e, H_{12}), E_2(M_e, H_{12})]_{I,V}\,, \\ &C(M_e, H_{12}), F_c(M_e, H_{12}), \sigma_{M_{V_{\max}}}(M_e, H_{12}) \end{aligned} \tag{8}$$

These parameters are dependent on the base flow quantities: Mach number M_e, and the pressure gradient related parameter H_{12}. A data fit for these 15 parameters is generated using FSC velocity profiles and our LST code using the following procedure [25, 30, 31, 33]:

1) Solve the FSC ordinary differential equations [62] and generate numerous similarity velocity profiles for different H_{12} values.

2) Run the exact LST code at specified M_e, H_{12}, and F; and compute the amplification rate α_i.

3) For each specified M_e and H_{12}, compute R_0, R_1, R_M, and σ_M versus α_i and Re_{δ_1} at different frequencies.

4) Use least-squares fitting to obtain the 15 parameters with different F at specified M_e and H_{12}.

5) Establish the two-entry lookup table for different M_e and H_{12} with these 15 parameters.

The parameters σ_M, R_M, R_0, and R_1 include both viscous and inviscid criteria. The two-entry lookup table is valid for $2.22 \leqslant H_{12} \leqslant 4.023$ and $0 \leqslant M_e \leqslant 1.3$. Decelerated flow ($H_{12} > 2.59$) is considered by using the inviscid criterion.

We now describe the procedure to compute the transition onset flow location, x_{tr}. For a given flow, we extract M_e and H_{12} with the laminar boundary code for calculating α_i. We then integrate α_i in the streamwise direction for N and obtain the maximum N-factor, N_{TS}. The transition location is determined by the linear interpolation

$$x_{\text{tr}} = \frac{N_{\text{TScrit}} - N_{\text{TS}_L}}{N_{\text{TS}_R} - N_{\text{TS}_L}}(x_R - x_L) + x_L \tag{9}$$

where x_L and x_R are, respectively, the coordinates of the left and right neighbors of the position where the critical N-factor is reached, and N_{TS_L} and N_{TS_R} are the corresponding N_{TS} values. Once the data fits underlying the method are generated, no specialized knowledge is required to use the database method. This makes the database method more efficient and convenient to use for adjoint-based optimization compared to the LST approach.

In the remainder of this paper, all transition locations triggered by TS waves are computed with the database method. The LST method is only used to generate training data and to benchmark the database method.

2.2.2 Separation-induced Transition Criterion

The laminar boundary-layer code terminates at the laminar separation position. When N_{TS} does not reach the critical N factor upstream of the separation point, the transition location is set to coincide with this separation position, i.e., $x_{\text{tr}} = x_{\text{sep}}$, where x_{sep} is the separation point. However, in reality, there is a laminar bubble downstream of the separation point and the laminar region is extended downstream. Taking the laminar bubble into consideration, Mayle [63] proposed a correction to predict the separation-induced transition more accurately. The separation-induced transition location is defined as

$$x_{\text{tr}} = x_{\text{sep}} + S_b Re_{\delta_2}^{0.7} \frac{\mu_e}{\rho_e U_e} \tag{10}$$

Depending on the separation topology, there can be a short laminar separation bubble or a long laminar separation bubble. For the short bubble ($Re_{\delta_1} > 450$), S_b is 300, whereas for for the long bubble ($Re_{\delta_1} \leq 450$), S_b is 1000. In our applications, we mainly apply the short bubble correlation.

2.2.3 Intermittency Function

Dhawan and Narasimha [64] proposed the intermittency concept, which is used to describe the flow fluctuation between laminar and turbulent regimes. This concept means that the flow becomes intermittent and alternates in the transition process. To study the physical nature of such flow, the fraction of time in which the flow remains turbulent during the transition phase is represented by the intermittency function [64, 65],

$$g(x) = \begin{cases} 1 - e^{-0.413\left(\frac{3.36(x-x_{\text{tr}})}{l_{\text{tr}}}\right)^2} & \text{if } x \geq x_{\text{tr}}, \\ 0 & \text{if } x < x_{\text{tr}} \end{cases} \tag{11}$$

where l_{tr} is the transition length. The intermittency function affects the production term of the turbulent model to predict transition in the RANS solver. The transition lengths induced by TS waves and induced by separation are determined by,

$$l_{\text{tr}_{\text{TS}}} = 2.3\sqrt{\left(\frac{\rho_e U_e}{\mu_e}\right)}(\delta_1)^{1.5}, \quad l_{\text{tr}_{\text{sep}}} = 700 Re_{\delta_2}^{0.7} \frac{\mu_e}{\rho_e U_e} \tag{12}$$

respectively [65]. The corresponding values are calculated at the transition point or the separation point.

2.3 Transition Prediction Framework

The e^N transition is modeled by the boundary layer solver and the database method. These two components constitute the transition simulation algorithm (TSA) module. The TSA module is coupled with the RANS solver using an iterative procedure in a two-field formulation. We show the coupling of these and other components of the transition prediction framework in Fig. 2 [66].

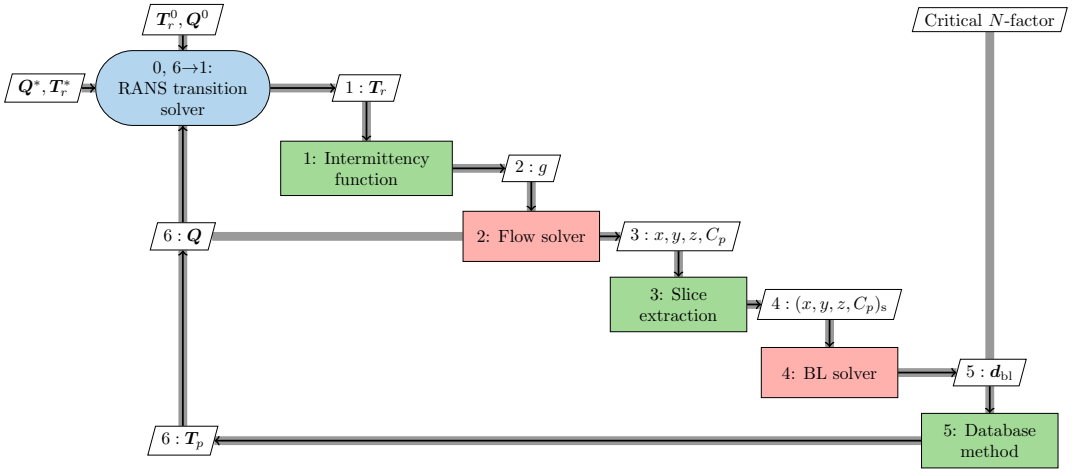


Figure 2: Extended design-structure matrix [66] for the transition prediction framework. Red blocks are iterative solvers, whereas green boxes represent explicit functions.

ADflow solves the flow field with the following form

$$\boldsymbol{\mathcal{A}}(\boldsymbol{Q}) = 0 \tag{13}$$

where $\boldsymbol{Q}$ is the vector of flow state variables. ADflow computes the solution of the RANS equations for a fixed transition location. We can either start the RANS solution from a fully turbulent boundary-layer computation or with the fixed transition positions. In this paper, we always start from a fully turbulent solution. The RANS equation residuals ($\boldsymbol{\mathcal{A}}$) include the inviscid, viscous, and artificial dissipation terms.

The database method and criterion for separation-induced transition for computing transition locations and lengths is defined as

$$\boldsymbol{T}_p = F_{\boldsymbol{\mathcal{T}}}(\boldsymbol{d}_{\text{bl}}; \boldsymbol{x}_s) \tag{14}$$

$\boldsymbol{T}_p$ is directly determined by the boundary-layer information contained in $\boldsymbol{d}_{\text{bl}}$. In the iterative procedure, the state variables $\boldsymbol{T}_r$ is updated using

$$\boldsymbol{T}_r^{(k)} = \boldsymbol{T}_r^{(k-1)} + \theta(\boldsymbol{T}_p^{(k-1)} - \boldsymbol{T}_r^{(k-1)}), \tag{15}$$

where $\boldsymbol{T}_p$ is the vector of the predicted transition locations and θ is the under-relaxation factor. Then, the transition module residual function is written as

$$\boldsymbol{\mathcal{L}}(\boldsymbol{T}_r) = \boldsymbol{T}_r - \boldsymbol{T}_p. \tag{16}$$

Thus, the whole transition module solution is treated as a laminar problem and is written as

$$\boldsymbol{\mathcal{L}}(\boldsymbol{T}_r) = 0 \tag{17}$$

After solving the RANS equations, the cross-sectional values are obtained from the flow solution field by extracting two-dimensional (2-D) slices from the three-dimensional (3-D) solution ("Slice extraction" in Fig 2) and the cross-section flow solution is transferred to the transition simulation. The results in this paper are pseudo-2-D solutions. Two-dimensional solutions are computed using single-cell-wide 3-D grids with parallel symmetry planes. As a result, we still need to extract slices from the 3-D solution for the boundary layer solver.

With the laminar problem formulation, we can write the governing equations as a function of the flow states $\boldsymbol{Q}$, transition results $\boldsymbol{T}_{\text{r}}$, and design variables $\boldsymbol{X}$. We write the combined residual equations from the RANS and the transition disciplines as

$$\boldsymbol{\mathcal{R}} = \begin{bmatrix} \boldsymbol{\mathcal{A}}(\boldsymbol{Q}, \boldsymbol{T}_r; \boldsymbol{X}) \\ \boldsymbol{\mathcal{L}}(\boldsymbol{T}_r, \boldsymbol{Q}; \boldsymbol{X}) \end{bmatrix} = 0 \tag{18}$$

The transition prediction based on the RANS-e^N framework finds a solution, $(\boldsymbol{Q}, \boldsymbol{T}_r)$, that satisfies the coupled residual equations. We solve the coupled transition prediction (18) using the nonlinear block Gauss–Seidel (NLBGS) method. In this approach, the aerodynamic simulation is partially converged, with $\epsilon_{\boldsymbol{\mathcal{A}}} \leqslant 10^{-8}$, allowing an approximation of the pressure coefficient C_p distribution to be computed.

The C_p distribution is transferred to the transition module, where the boundary-layer states $\boldsymbol{d}_{\text{bl}}$ are computed using Eqs. (1) and (2). The transition locations x_{tr} for the two types of transition are computed using Eqs. (9) and (10). The corresponding transition lengths l_{tr} are computed using Eq. (12).

The transition locations and transition lengths are passed back to the ADflow solver, where the intermittency function (11) is computed. A new aerodynamic solution is converged using the new intermittency function values, and this procedure is repeated until a certain tolerance is met ($\epsilon_{\boldsymbol{\mathcal{A}}} \leq 10^{-8}$).

The RANS solver and the transition module are coupled using the intermittency function as a factor in a SA turbulence model source term used in ADflow. The modified equation is

$$\frac{D\widetilde{\nu}}{Dt} = g\left\{c_{b1}\widetilde{S}\widetilde{\nu}(1-f_{t2}) - \left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\widetilde{\nu}}{d}\right)^2\right\} + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu+\widetilde{\nu})\right)\frac{\partial\widetilde{\nu}}{\partial x_j} + c_{b2}\frac{\partial\widetilde{\nu}}{\partial x_j}\frac{\partial\widetilde{\nu}}{\partial x_j}\right] \tag{19}$$

When the flow is laminar, g is zero and the production term of the turbulence model turns off; see Eq.(11). This process is defined as a modeling of transitional flow regions that is used for updating the transition locations and transition lengths in the RANS solver. Descriptions of the constants c_{b1}, c_{b2} and $c_{\omega 1}$ and Eq. (19) is available in the literature [57]. The full NLBGS solution procedure is detailed in Algorithm 1.

Algorithm 1 Transition prediction using a nonlinear block Gauss–Seidel solver

Given: $\boldsymbol{Q}^{(0)}$, $\boldsymbol{T}_r^{(0)}$, $k_{\max}$
$\boldsymbol{x}_s \leftarrow \boldsymbol{x}_{\text{surf}}$ ▷ Extract the cross-sectional coordinates
for $k \leftarrow 1, k_{\max}$ **do**
  $g^{(k)} \leftarrow \boldsymbol{T}_r^{(k-1)}$ ▷ Transfer transition locations from TSA to RANS
  $\boldsymbol{\mathcal{A}}^{(k)} \leftarrow \boldsymbol{\mathcal{A}}\left(\boldsymbol{Q}^{(k-1)}, g^{(k)}\right)$ ▷ Evaluate initial RANS residual
  while $\parallel \boldsymbol{\mathcal{A}}(\boldsymbol{Q}_i^{(k-1)}; \boldsymbol{X}) \parallel > \varepsilon_{\boldsymbol{\mathcal{A}}} \parallel \boldsymbol{\mathcal{A}}^{(k)} \parallel$ **do** ▷ RANS partial convergence criterion
    $\boldsymbol{Q}_{i+1}^{(k-1)} \leftarrow \boldsymbol{Q}_i^{(k-1)} + \Delta \boldsymbol{Q}_i$ ▷ Iterate RANS
    $i \leftarrow i + 1$
  end while
  $\boldsymbol{Q}^{(k)} \leftarrow \boldsymbol{Q}^{(k-1)}$ ▷ Updated RANS solution
  $C_p^{(k)} \leftarrow C_p\left(\boldsymbol{Q}^{(k)}, g^{(k)}\right)$ ▷ Evaluate the surface pressure coefficient
  $C_{p_s}{}_{\boldsymbol{\mathcal{A}}}^{(k)} \leftarrow C_p^{(k)}$ ▷ Extract the cross-section pressure coefficient
  $C_{p_s}^{(k)} \leftarrow C_{p_s}{}_{\boldsymbol{\mathcal{A}}}^{(k)}$ ▷ Transfer the pressure coefficient from RANS to TSA
  $\boldsymbol{\mathcal{B}}^{(k)} \leftarrow \boldsymbol{\mathcal{B}}\left(\bar{\boldsymbol{q}}^{(k-1)}, C_{p_s}^{(k)}\right)$ ▷ Evaluate initial boundary-layer residual
  while $\parallel \boldsymbol{\mathcal{B}}(\bar{\boldsymbol{q}}^{(k-1)}, C_{p_s}^{(k)}; \boldsymbol{x}_s) \parallel > \varepsilon_{\boldsymbol{\mathcal{B}}} \parallel \boldsymbol{\mathcal{B}}^{(k)} \parallel$ **do** ▷ Boundary-layer partial convergence criterion
    $\bar{\boldsymbol{q}}_{i+1}^{(k-1)} \leftarrow \bar{\boldsymbol{q}}_i^{(k-1)} + \Delta \bar{\boldsymbol{q}}_i$ ▷ Iterate the boundary-layer
    $i \leftarrow i + 1$
  end while
  $\bar{\boldsymbol{q}}^{(k)} \leftarrow \bar{\boldsymbol{q}}^{(k-1)}$ ▷ Update the boundary-layer solution
  $\boldsymbol{d}_{\text{bl}}^{(k)} \leftarrow F_{\boldsymbol{\mathcal{B}}}(\bar{\boldsymbol{q}}^{(k)}, C_{p_s}^{(k)}; \boldsymbol{x}_s)$ ▷ Evaluate the boundary-layer parameters
  $\boldsymbol{T}_p^{(k)} \leftarrow F_{\boldsymbol{\mathcal{T}}}(\boldsymbol{d}_{\text{bl}}^{k}; \boldsymbol{x}_s)$ ▷ Transition locations computation
  $\Delta \boldsymbol{T}_p^{(k)} \leftarrow \boldsymbol{T}_p^{(k)} - \boldsymbol{T}_p^{(k-1)}$ ▷ Compute the displacement increment
  if $k > 1$ **then**
    $\theta \leftarrow \left(1 - \frac{\left(\Delta \boldsymbol{T}_p^{(k)} - \boldsymbol{T}_p^{(k-1)}\right) \cdot \Delta \boldsymbol{T}_p^{(k)}}{\| \Delta \boldsymbol{T}_p^{(k)} - \Delta \boldsymbol{T}_p^{(k-1)} \|^2}\right)$ ▷ Adapt under-relaxation with Aitken acceleration
  end if
  $\boldsymbol{T}_r^{(k+1)} \leftarrow \boldsymbol{T}_r^{(k)} + \theta \Delta \boldsymbol{T}_p^{(k)}$ ▷ Compute the temporary transition locations
  if $\parallel \boldsymbol{\mathcal{A}}^{(k)} \parallel < \varepsilon_{\boldsymbol{\mathcal{AL}}} \parallel \boldsymbol{\mathcal{A}}^{(1)} \parallel$ **and** $\parallel \boldsymbol{\mathcal{L}}^{(k)} \parallel < \varepsilon_{\boldsymbol{\mathcal{AL}}} \boldsymbol{\mathcal{L}}^{(1)}$ **then** ▷ Transition

prediction convergence criterion

break

end if

end for

Result: $\boldsymbol{T}_r$

2.4 Validation and Verification

We validate and verify the transition framework [28] using the database method on the classic NLF(1)-0416 airfoil [67]. This airfoil is designed to operate at $C_l = 0.4$ and has a maximum thickness ratio $t/c = 0.16$, which leads to favorable stall characteristics.

We validate the transition prediction tool against experimental data for various angles of attack at $Re = 4.0 \times 10^6$ and $M = 0.1$ [67]. The freestream turbulence intensity is 0.15%. According to the correlation by Mack [68], $N_{\mathrm{TScrit}} = 7.2$ at this turbulence level. The simulation requires about 6–10 iterations for the RANS-e^N transition framework to converge. Figure 3 compares the simulation results and experimental data. The experimental lift coefficient is well matched in the linear region, although the simulation slope is slightly larger than the experimental data in Fig. 3(a). This result also has the same trend observed by Coder [69]. The drag polar obtained from our simulation results shows an excellent agreement with experimental data, as shown in Fig. 3(b).

The transition locations are compared in Fig. 3(c). Because the experimental data does not cover the transition results on both the upper and lower surface for all angles of attack, we compare the available transition location regions only. The simulation results we present here indicate that our transition predicting framework is reliable and robust for laminar-to-turbulent transition simulation.

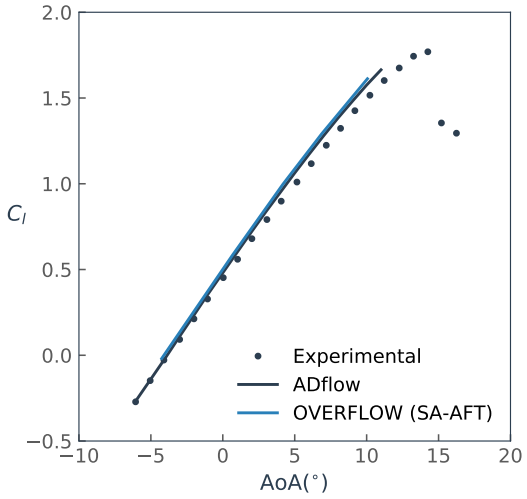
Finally, we show the N-factor distribution for both the database method and exact LST at an angle of attack (AOA) of 4 deg and $C_l = 0.9$. Numerical results results obtained with the database method for different configurations and conditions have been verified by Perraud et al. [31] and by Bégou et al. [33]. Using our code, the database method results agree with the exact LST results, as shown in Fig. 4, which verifies the accuracy of the simplified e^N method.

2.5 Ensuring Smooth Functions

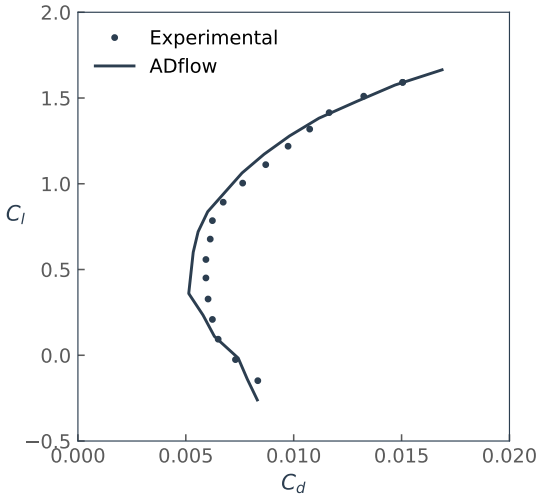
Ensuring that objective and constraint functions are smooth with respect to the design variables is crucial when using gradient-based optimization. To check the smoothness of these functions when considering transition, we select two airfoils and interpolate linearly in the design space between these two airfoils using mode shapes [70]. The two airfoils are analyzed for $M = 0.42$, $Re = 12 \times 10^6$, and AOA $= 1.54°$. Figure 5 shows the lift, drag coefficients and transition locations versus the interpolation factor χ.

We show three sets of results:

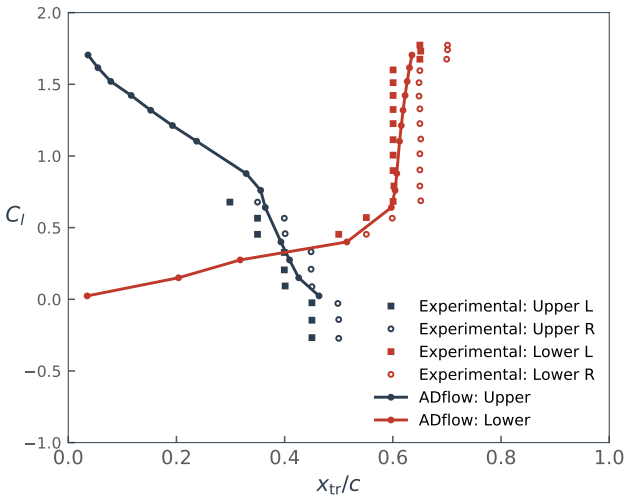
1) The first is the result with no $\boldsymbol{g}$. this is the solution without including the intermittency function (11). The turbulent production terms are switch on at discrete



(a) Lift coefficient versus AOA [69]



(b) Drag polar



(c) Transition locations versus C_l

Figure 3: Transition simulation results comparison for the NLF(1)-0416 airfoil with experimental data [67].

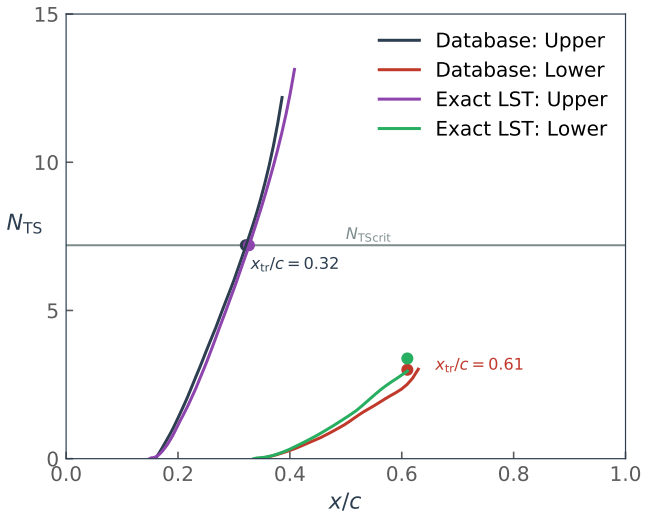


Figure 4: Comparison of database N factor with LST results at AOA= 4°, $C_l = 0.9$.

cell locations downstream of the transition location.

2) The second result is with $\boldsymbol{g}$. This solution is with an intermittency function that is not smoothed. The turbulent production terms are switched on over the length of the transition length, with the start of the transition region being identified by the closest node to the transition location.

3) The third result is smoothed. This solution with is the smoothed intermittency function (11). The turbulent production terms are switched on over the length of the transition region, with the start of the transition region being interpolated to its exact position by interpolating between cells and using a strict transition convergence criterion.

The intermittency function is related to the transition location and transition length through Eq. (11). In the transition simulation procedure, two modules (TSA and ADflow) use interpolation of the transition location. When determining the transition location from the critical N factor, the transition locations are defined by Eq. (9), with the mesh points close to the transition point in the TSA module being used to identify the transition location. Then, the transition location is passed to ADflow, where the discrete mesh locations are used to specify the intermittency. Specifically, the surface is divided into laminar and turbulent regions by the transition location and g is set to zero in the laminar region. Without interpolation, this procedure introduces discontinuities in the design space, since the transition location jumps from cell to cell as the shape of the airfoil changes. From the implementation point of view, code developers may use the grid points when using the intermittency function, which affects the smoothness of the functions, and thus the robustness of gradient-based optimization. Therefore, we show the results "With g".

The computation labeled "With g" uses the mesh points directly, without interpolation, in both the TSA framework and ADflow to compute the transition location

and the intermittency function. The transition convergence tolerance is 10^{-3} in the transition simulation for this case. Figure 5 shows that both with and without intermittency, C_l is smooth but C_d is not. Although, the drag coefficient is smoother with the intermittency function than without intermittency, it still exhibits some noise. The same figure shows that by linearly interpolating the exact transition locations and lengths for intermittency function in ADflow and by using a stricter convergence criteria ($\epsilon_{\mathcal{L}} \leqslant 10^{-7}$) in the TSA module, we are able to smooth the drag coefficient function with respect to the airfoil shape variables. We also show a smooth variation transition locations with respect to the airfoil shape variables in Fig. 5, which in turn ensures the smoothness of the lift and drag coefficient functions with respect to the airfoil shape variables. The smoothed version increases the robustness of the gradient-based optimizations, which made it possible to obtain the results presented in Sec. 4.

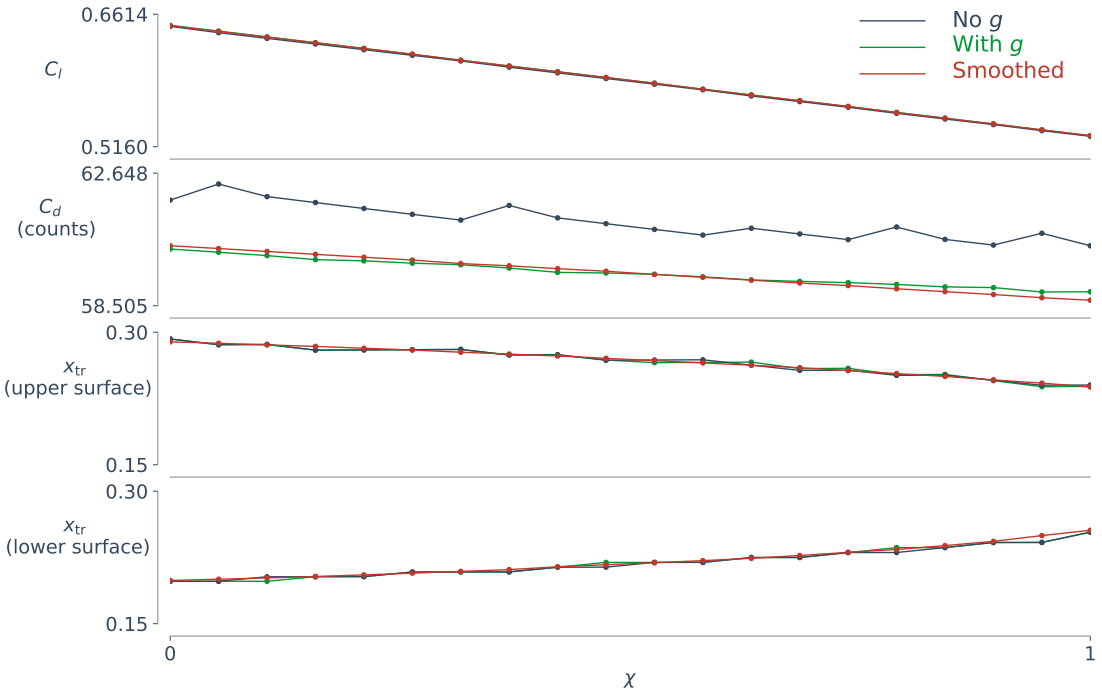


Figure 5: Proposed approach is effective in smoothing drag coefficient with respect to airfoil shape variables (bottom plot), where χ represents the position in a line going from one airfoil ($\chi = 0$) to another ($\chi = 1$).

3 Coupled-Adjoint Derivative Computation

3.1 Coupled-Adjoint Equations

As mentioned in the introduction (Sec. 1), adjoint derivative computation is essential to aerodynamic shape optimization. To account for the effect of laminar-turbulent transition, a coupled adjoint is required to compute the derivatives of the coupled system of equations. The coupled-adjoint approach has been previously implemented

for problems coupling aerodynamics and structures [71, 72] and has been generalized to multiple disciplines [73–75].

We use the discrete adjoint implemented in ADflow to compute the derivatives for the RANS equations [35]. In this section, we describe the additional terms we implemented to compute the derivatives of the system consisting of the RANS equations and the transition model.

The total derivative of the function of interest I is

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{X}} = \frac{\partial I}{\partial \boldsymbol{X}} - \boldsymbol{\psi}^T \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{X}} \tag{20}$$

where $\boldsymbol{X}$ is the vector of design variables, and $\boldsymbol{Q}$ is the vector of state variables. The function of interest, $I = I(\boldsymbol{X}, \boldsymbol{Q})$, is either an objective function such as the drag coefficient, or a constraint, such as the lift coefficient. Using the existing fully turbulent adjoint as a starting point, we augment the sensitivities to include the transition prediction code. To accomplish this, we couple the RANS-based solver, which solves the equation $\boldsymbol{\mathcal{A}}(\boldsymbol{Q}) = 0$, with the simplified e^N transition algorithm, which solve the equation $\boldsymbol{\mathcal{L}}(\boldsymbol{T}_r) = 0$. There are two sets of state variables: the flow state vector, $\boldsymbol{Q}$, and the vector of transition locations, $\boldsymbol{T}_r$.

The coupled residuals, states, and adjoints are

$$\boldsymbol{\mathcal{R}} = \begin{bmatrix} \boldsymbol{\mathcal{A}} \\ \boldsymbol{\mathcal{L}} \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Q} \\ \boldsymbol{T}_r \end{bmatrix}, \quad \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{bmatrix} \tag{21}$$

where the transition residual is $\boldsymbol{\mathcal{L}} = \boldsymbol{T}_r - \boldsymbol{T}_p$ for a given transition location predicted by the simplified e^N method $\boldsymbol{T}_p$. Given the states of the coupled system [Eq. (21)], the coupled total derivative of the function of interest is written as

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{X}} = \frac{\partial I}{\partial \boldsymbol{X}} + \begin{bmatrix} \frac{\partial I}{\partial \boldsymbol{Q}} & \frac{\partial I}{\partial \boldsymbol{T}_r} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{Q}}{\mathrm{d}\boldsymbol{X}} \\ \frac{\mathrm{d}\boldsymbol{T}_r}{\mathrm{d}\boldsymbol{X}} \end{bmatrix} \tag{22}$$

The other total derivative equation that is needed to derive for the adjoint method is the total derivative of the residuals, which must be zero for the governing equations to remain satisfied. The total derivative of the coupled residuals is

$$\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}\boldsymbol{X}} = \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{\mathcal{A}}}{\mathrm{d}\boldsymbol{X}} \\ \frac{\mathrm{d}\boldsymbol{\mathcal{L}}}{\mathrm{d}\boldsymbol{X}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{X}} \\ \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{X}} \end{bmatrix} + \begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{Q}} & \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{T}_r} \\ \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{Q}} & \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{T}_r} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{Q}}{\mathrm{d}\boldsymbol{X}} \\ \frac{\mathrm{d}\boldsymbol{T}_r}{\mathrm{d}\boldsymbol{X}} \end{bmatrix} = 0 \tag{23}$$

Substituting the solution of Eq. (23) into Eq. (22), we get

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{X}} = \frac{\partial I}{\partial \boldsymbol{X}} - \underbrace{\begin{bmatrix} \frac{\partial I}{\partial \boldsymbol{Q}} & \frac{\partial I}{\partial \boldsymbol{T}_r} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{Q}} & \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{T}_r} \\ \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{Q}} & \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{T}_r} \end{bmatrix}^{-1}}_{\boldsymbol{\Psi}^T} \begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{X}} \\ \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{X}} \end{bmatrix} \tag{24}$$

Using the two terms highlighted [Eq. (24)], we can form the coupled-adjoint equations:

$$\begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{Q}} & \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{T}_r} \\ \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{Q}} & \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{T}_r} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial \boldsymbol{Q}} \\ \frac{\partial I}{\partial \boldsymbol{T}_r} \end{bmatrix}^T \tag{25}$$

These equations are solved for the two adjoint vectors, which can then be used in the following equations to compute the total derivatives for the coupled RANS-e^N system:

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{X}} = \frac{\partial I}{\partial \boldsymbol{X}} - \boldsymbol{\psi}^T \left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{X}} \right) - \boldsymbol{\phi}^T \left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{X}} \right) \tag{26}$$

In the next section, we describe how the partial derivatives in the above equations are computed and how the coupled-adjoint equations are solved.

3.2 Computation of Partial Derivatives

To solve the coupled-adjoint equations, we need to derive all the partial derivatives in Eqs. (25) and (26), and to have an efficient implementation of these terms. These are $\partial\boldsymbol{\mathcal{A}}/\partial\boldsymbol{Q}$, $\partial\boldsymbol{\mathcal{A}}/\partial\boldsymbol{T}_r$, $\partial\boldsymbol{\mathcal{A}}/\partial\boldsymbol{X}$, $\partial\boldsymbol{\mathcal{L}}/\partial\boldsymbol{T}_r$, $\partial\boldsymbol{\mathcal{L}}/\partial\boldsymbol{Q}$, and $\partial\boldsymbol{\mathcal{L}}/\partial\boldsymbol{X}$. Since ADflow uses generalized minimal residual method to solve the adjoint equation [35], we only require the matrix-vector products to solve the coupled-adjoint equations (25). None of the partial derivatives in this section are stored explicitly. Instead, we used the Tapenade [76] AD tool in reverse mode to generate the code that computes the partial derivative matrix-vector products.

3.2.1 Aerodynamic Partial Derivatives

The RANS diagonal term, $\partial\boldsymbol{\mathcal{A}}/\partial\boldsymbol{Q}$, represents the derivative of the RANS equation residuals with respect to the RANS equation state variables and is provided by the RANS adjoint implemented in ADflow [35].

The off-diagonal block, $\partial\boldsymbol{\mathcal{A}}/\partial\boldsymbol{T}_r$, is the derivative of the RANS residuals with respect to the laminar transition states and represents the coupling between the two systems. We compute this term using reverse accumulation, starting with the adjoint vector and using the chain rule to propagate the derivative backward as shown in the following:

$$\left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{T}_r} \right)^T \boldsymbol{\psi} = \left(\frac{\partial g}{\partial \boldsymbol{T}_r} \right)^T \underbrace{\left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial g} \right)^T \boldsymbol{\psi}}_{\boldsymbol{\psi}_1} . \tag{27}$$

We first compute $\boldsymbol{\psi}_1$ and then use that result to compute the value for $(\partial g/\partial \boldsymbol{T}_r)^T \boldsymbol{\psi}_1$. The matrix-vector product for the RANS residual partial derivative $\boldsymbol{\psi}_1$ is provided by ADflow and dependent on the modified turbulence model. We compute the partial derivatives of this term by extending the reverse-mode AD in ADflow to include the terms in this computation. The term $(\partial g/\partial \boldsymbol{T}_r)^T \boldsymbol{\psi}_1$ is defined by Eq. (11), and the corresponding code is also differentiated using Tapenade.

The adjoint product of the derivative of the aerodynamic residuals with respect to the design variables is computed using the chain rule

$$\left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{X}} \right)^T \boldsymbol{\psi} = \left(\frac{\partial \boldsymbol{x}_{\text{surf}}}{\partial \boldsymbol{X}} \right)^T \underbrace{\left(\frac{\partial \boldsymbol{x}_v}{\partial \boldsymbol{x}_{\text{surf}}} \right)^T \underbrace{\left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{x}_v} \right)^T \boldsymbol{\psi}}_{\boldsymbol{\psi}_1}}_{\boldsymbol{\psi}_2} \tag{28}$$

The two terms on the left are computed by the geometry module (pyGeo[5]) and the mesh movement module (IDWarp[6]). The right-most term is derived using reverse-mode AD.

3.2.2 Transition Partial Derivatives

The diagonal term, $\partial \boldsymbol{\mathcal{L}} / \partial \boldsymbol{T}_r$, represents the derivatives of the transition residuals with respect to the laminar transition states. This term is an identity matrix of size $N_{\text{tr}} \times N_{\text{tr}}$, where N_{tr} includes upper and lower surface transition locations and lengths, and it is four times the number of spanwise sections N_{s}.

The remaining off-diagonal block, $\partial \boldsymbol{\mathcal{L}} / \partial \boldsymbol{Q}$, represents the derivatives of the transition residuals with respect to the RANS state variables. This term is also obtained using reverse accumulation through the chain rule,

$$\left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{\phi} = \left(\frac{\partial C_p}{\partial \boldsymbol{Q}}\right)^T \underbrace{\left(\frac{\partial C_{p_s}}{\partial C_p}\right)^T \underbrace{\left(\frac{\partial \boldsymbol{d}_{\text{bl}}}{\partial C_{p_s}}\right)^T \underbrace{\left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{d}_{\text{bl}}}\right)^T \boldsymbol{\phi}}_{\boldsymbol{\phi}_1}}_{\boldsymbol{\phi}_2}}_{\boldsymbol{\phi}_3}, \tag{29}$$

where $(\partial \boldsymbol{\mathcal{L}} / \partial \boldsymbol{d}_{\text{bl}})$ is the derivative of the transition location residuals with respect to the boundary-layer parameters, and $\partial \boldsymbol{d}_{\text{bl}} / \partial C_{p_s}$ is the derivative of the boundary-layer parameters with respect to the section profile pressure coefficient distribution. This term is related to the partial differential equations solution for the laminar boundary layer. We compute the $(\partial C_{p_s} / \partial C_p)^T \boldsymbol{\phi}_2$ in ADflow, where $(\partial C_{p_s} / \partial C_p)^T$ represents the partial derivatives of the section profile value with respect to the surface C_p of the airfoil CFD mesh. The term $(\partial C_p / \partial \boldsymbol{Q})^T$ represents the partial derivatives of the surface pressure coefficient with respect to the RANS state variables. Finally, we compute $(\partial C_p / \partial \boldsymbol{Q})^T \boldsymbol{\phi}_3$.

The derivative of the transition residual with respect to the design variables, $\partial \boldsymbol{\mathcal{L}} / \partial \boldsymbol{X}$, is expressed by the chain rule as

$$\left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{X}}\right)^T \boldsymbol{\phi} = \left(\frac{\partial \boldsymbol{x}_{\text{surf}}}{\partial \boldsymbol{X}}\right)^T \underbrace{\left(\frac{\partial \boldsymbol{x}_v}{\partial \boldsymbol{x}_{\text{surf}}}\right)^T \underbrace{\left(\frac{\partial \boldsymbol{x}_s}{\partial \boldsymbol{x}_v}\right)^T \underbrace{\left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{x}_s}\right)^T \boldsymbol{\phi}}_{\boldsymbol{\phi}_1}}_{\boldsymbol{\phi}_2}}_{\boldsymbol{\phi}_3}, \tag{30}$$

where $\boldsymbol{x}_s$ is the geometry coordinates for the section profile. Note that $(\partial \boldsymbol{\mathcal{L}} / \partial \boldsymbol{x}_s)^T$ is obtained in the same way as for $(\partial \boldsymbol{\mathcal{L}} / \partial C_{p_s})^T$. We use the reverse-mode AD in ADflow to compute $(\partial \boldsymbol{x}_s / \partial \boldsymbol{x}_{\text{surf}})^T \boldsymbol{\phi}_1$. The matrix-vector product $(\partial \boldsymbol{x}_{\text{surf}} / \partial \boldsymbol{X})^T \boldsymbol{\phi}_3$ and $(\partial \boldsymbol{x}_v / \partial \boldsymbol{x}_{\text{surf}})^T \boldsymbol{\phi}_2$ are unchanged from the RANS adjoint implementation [35, 55] and

[5] https://github.com/mdolab/pygeo [retrieved September, 2020]
[6] https://github.com/mdolab/idwarp [retrieved September, 2020]

are computed by the geometry module (pyGeo[7]) and the mesh movement module (IDWarp[8]), respectively.

3.2.3 Function of Interest

The functions of interest are the force and moment coefficients (C_l, C_d, and C_m). The transition location is not set as the function interest because we are ultimately interested in objectives and constraints based on the aerodynamic force coefficients. The partial derivatives of the total-derivative function, $\partial I/\partial \boldsymbol{Q}$, are the same as before, whereas $(\partial I/\partial \boldsymbol{T}_r)$ is zero.

3.3 Coupled-adjoint Solution

For the RANS-e^N laminar-to-turbulent transition prediction method, there are two methods available for solving the coupled-adjoint system [Eq. (25)]: the linear block Gauss–Seidel (LBGS) method and the coupled-Krylov method [77]. The linear block Gauss–Seidel method uses a segregated approach, whereas the coupled-Krylov method uses a monolithic approach [72].

3.3.1 Linear Block Gauss–Seidel Method

The LBGS method moves the off-diagonal terms of the RANS-e^N adjoint equations [Eq. (25)] to the right-hand side (RHS), resulting in the lagged system of equations:

$$\begin{aligned} \left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{\psi}^{(k)} &= \left(\frac{\partial I}{\partial \boldsymbol{Q}}\right)^T - \left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{\phi}^{(k-1)} \\ \left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{T}_r}\right)^T \boldsymbol{\phi}^{(k)} &= \left(\frac{\partial I}{\partial \boldsymbol{T}_r}\right)^T - \left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{T}_r}\right)^T \boldsymbol{\psi}^{(k)} \end{aligned} \tag{31}$$

where k is the iteration index. The partial derivative $\partial \boldsymbol{\mathcal{L}}/\partial \boldsymbol{T}_r$ is an identity matrix, and $\partial I/\partial \boldsymbol{T}_r$ is zero. We iterate these two equations in sequence until a target convergence level is reached. Each of these two equations is fully converged to their respective solutions before proceeding to the other equation. We use an under-relaxation factor (Algorithm 2) for the transition update to improve convergence.

Formulated in this fashion, the coupled-adjoint equations now look like a pair of single discipline adjoint equations with augmented right-hand sides. Therefore, we can reuse the existing RANS adjoint solver in ADflow, which is the main advantage of this algorithm. Further, because the diagonal block of the coupled-adjoint system corresponding to the transition computation is an identity matrix, the solution of the transition adjoint reduces to a matrix-vector product with the RANS adjoint vector. The pseudocode for the complete procedure is shown in Algorithm 2.

Algorithm 2 Coupled-adjoint linear block Gauss–Seidel solver

1: **Given**: $\boldsymbol{\psi}^{(0)}$, $\boldsymbol{\phi}^{(0)}$, $k_{\max}$ ▷ $\boldsymbol{\psi}^{(0)}$ and $\boldsymbol{\phi}^{(0)}$ could come from previous solution

2: **for** $k \leftarrow 1, k_{\max}$ **do**

[7] https://github.com/mdolab/pygeo [retrieved September, 2020]

[8] https://github.com/mdolab/idwarp [retrieved September, 2020]

$\boldsymbol{\phi}_2 \leftarrow \left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial C_p}\right)^T \boldsymbol{\phi}^{(k-1)}$ ▷ Transfer the transition adjoint from TSA

$RHS_{\boldsymbol{\mathcal{A}}}^{(k)} \leftarrow \left(\frac{\partial I}{\partial \boldsymbol{Q}}\right)^T - \left(\frac{\partial C_p}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{\phi}_2$ ▷ RANS adjoint RHS

$\left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{\psi}^{(k)} = RHS_{\boldsymbol{\mathcal{A}}}^{(k)}$ ▷ Partially solve RANS adjoint to get $\boldsymbol{\psi}^{(k)}$

$\boldsymbol{\phi}^k \leftarrow -\left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{T}_r}\right)^T \boldsymbol{\psi}^{(k)}$ ▷ Transfer the RANS contribution to transition RHS

$RHS_{\boldsymbol{\mathcal{L}}}^{(k)} \leftarrow \boldsymbol{\phi}^k - \boldsymbol{\phi}^{(k-1)}$ ▷ Compute the transition RHS

$\Delta\boldsymbol{\phi} = RHS_{\boldsymbol{\mathcal{L}}}^{(k)}$ ▷ Compute the TSA adjoint update

$\boldsymbol{\phi}^{(k)} \leftarrow \boldsymbol{\phi}^{(k-1)} + \theta\Delta\boldsymbol{\phi}$ ▷ Under-relaxed transition adjoint update

if $\| RHS_{\boldsymbol{\mathcal{A}}}{}^{(k)} \| < \varepsilon_{\text{LF}} \| RHS_{\boldsymbol{\mathcal{A}}}{}^{(1)} \|$ and $\| RHS_{\boldsymbol{\mathcal{L}}}^{(k)} \| < \varepsilon_{\text{LF}} \| RHS_{\boldsymbol{\mathcal{L}}}^{(1)} \|$ **then**

break ▷ Coupled-adjoint converged

end if

end for

Result: $\boldsymbol{\psi}$, $\boldsymbol{\phi}$

3.3.2 Coupled-Krylov Method

The second method for solving the coupled-adjoint system is the coupled-Krylov method, which is monolithic. Kennedy and Martins [77] and Kenway et al. [72] applied the coupled-Krylov method to aerostructural problems. Kenway et al. [72] benchmarked the coupled-Krylov method against LBGS and concluded that the coupled-Krylov method is more efficient. This motivates the implementation of the coupled-Krylov method for the laminar flow problem.

Because we use reverse-mode AD to compute all of the partial derivatives in the coupled-adjoint system, we can directly compute the transpose matrix-vector products needed by the Krylov method. This allows us to efficiently solve the coupled system without explicitly storing the coupled Jacobian. To ensure that the Krylov solver performs well, we need effective preconditioning for the coupled system solution. To accomplish this, we store the approximate diagonal block $(\partial \boldsymbol{\mathcal{A}}/\partial \boldsymbol{Q})_{\text{PC}}^T$ computed by ADflow and ignore the off-diagonal terms $\partial \boldsymbol{\mathcal{A}}/\partial \boldsymbol{T}_r$ and $\partial \boldsymbol{\mathcal{L}}/\partial \boldsymbol{Q}$. For the transition module diagonal block, the preconditioner is an identity matrix, since that corresponds to the actual values of the Jacobian. These two diagonal block form a block-Jacobi preconditioner for the coupled system. For the RANS block, we reuse the preconditioned Krylov subspace method used for the fully-turbulent adjoint solution. The pseudocode for the transpose matrix-vector product used in the coupled linear solution is shown in Algorithm 3.

Algorithm 3 Coupled-adjoint Krylov linear operator

function MULT($\boldsymbol{z}$) ▷ Compute Jacobian-vector product with $\boldsymbol{z}$

  $(\boldsymbol{z}_{\mathcal{A}}, \boldsymbol{z}_{\mathcal{L}}) \leftarrow \boldsymbol{z}$ ▷ Extract the RANS and transition components

  $\boldsymbol{\psi} \leftarrow \left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{z}_{\mathcal{A}}$ ▷ Evaluate diagonal contributions in parallel

  $\boldsymbol{\phi} \leftarrow \boldsymbol{z}_{\mathcal{L}}$

  $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} + \left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \boldsymbol{Q}}\right)^T \boldsymbol{z}_{\mathcal{L}}$ ▷ Add RANS off-diagonal term

  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \left(\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{T}_r}\right)^T \boldsymbol{z}_{\mathcal{A}}$ ▷ Add off-diagonal term of transition module

  $\boldsymbol{\Psi} \leftarrow (\boldsymbol{\psi}, \boldsymbol{\phi})$ ▷ Combine RANS and transition components

  return $\boldsymbol{\Psi}$

end function

3.4 Gradient Verification and Adjoint Solution Algorithms Comparison

3.4.1 Baseline Solution

We use the RAE 2822 airfoil case to verify the adjoint implementation. The computational mesh for the airfoil has 251 points in the chordwise direction and 121 points in the wall-normal direction. The airfoil geometry, transition locations, and free-form deformation (FFD) control points (which are used as shape design variables) [78] are shown in Fig. 6. We have 18 design variables for this case. The flow conditions for the simulation are $Re = 5.6 \times 10^6$, $M = 0.19$ and AOA $= 0.7498°$. The resulting pressure distribution and predicted transition locations are shown in Fig. 6. The predicted transition location and transition length are $x_{\text{tr}}/c = 0.370$ and $l_{\text{tr}}/c = 0.0485$, respectively, on the upper surface; whereas on the lower surface, the corresponding quantities are $x_{\text{tr}}/c = 0.481$ and $l_{\text{tr}}/c = 0.0604$, respectively. Additionally, the upper surface transition is caused by TS wave amplification, whereas laminar separation triggers transition on the lower surface because N_{TS} is not up to N_{TScrit} until the laminar separation point (see Fig. 6).

3.4.2 Verification of Partial Derivatives

In this paper, we use two approaches to verify the gradient computations: The dot-product test and finite difference derivative estimates. The dot-product test is used to check the consistency between the forward AD and reverse AD codes. For a given code with a vector of inputs $\boldsymbol{Z}$ and outputs $\boldsymbol{U}$, such that $\boldsymbol{U} = \boldsymbol{F}(\boldsymbol{Z})$, the dot-product test is expressed as

$$\dot{\boldsymbol{Z}}^T \overline{\boldsymbol{Z}} = \dot{\boldsymbol{U}}^T \overline{\boldsymbol{U}} \tag{32}$$

where the forward AD result and the reverse AD result are, respectively,

$$\dot{\boldsymbol{U}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Z}} \dot{\boldsymbol{Z}}, \quad \overline{\boldsymbol{Z}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Z}}^T \overline{\boldsymbol{U}}$$

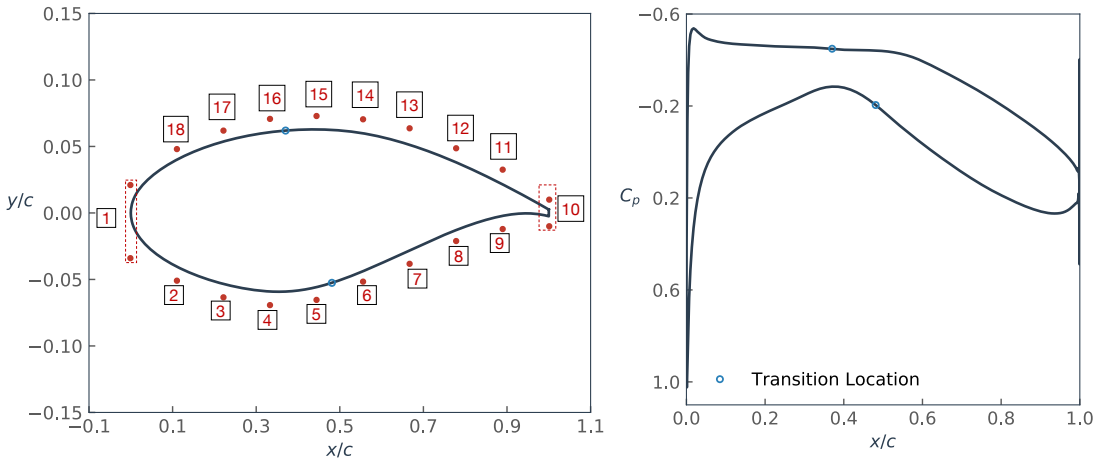


Figure 6: RAE 2822 airfoil geometry, FFD control points, transition location, and pressure distribution.

In the forward derivative computation, we seed a random vector $\dot{\boldsymbol{Z}}$ and we want to compute $\dot{\boldsymbol{U}}$. Then, we generate a random vector $\bar{\boldsymbol{U}}$ and compute the corresponding $\bar{\boldsymbol{Z}}$ with the reverse AD code. Finally, we verify that the left- and right-hand sides of Eq. (32) match to machine precision.

This dot-product test ensures that the AD code is consistent with the original subroutine, and we can perform this test for every AD module in the chain. We used this method to check all the related partial derivatives subroutines: the transition simulation module, the spanwise direction slices subroutine, and the intermittency function subroutine. As shown in Table 1, the forward and reverse modes match each other to least 10 digits.

Table 1: Dot-product verification

Product	$\partial\boldsymbol{\mathcal{L}}/\partial C_{p_s}$	$\partial C_p/\partial C_{p_s}$	$\partial g/\partial \boldsymbol{T}_r$
$\dot{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}}$	2.667966055**014**	34.333164845932	−7.8272538571149
$\dot{\boldsymbol{U}}^T\bar{\boldsymbol{U}}$	2.667966055**389**	34.333164845932	−7.8272538571148

3.4.3 Adjoint Solution Verification

In a second verification step, we compare the directional derivatives of C_l and C_d in a random direction computed with the coupled-Krylov solver for the coupled-adjoint equations with forward finite differences. The forward finite difference formula for a directional derivative is

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{Z}}\dot{\boldsymbol{Z}} \approx \frac{F(\boldsymbol{Z}+h\dot{\boldsymbol{Z}})-F(\boldsymbol{Z})}{h} \tag{33}$$

where $\dot{\boldsymbol{Z}}$ is in a random unit vector, and h is the step size.

In this case, we compute the derivatives of C_l and C_d, and use coupled Krylov to solve the coupled-adjoint equations. Table 2 lists the derivatives computed using the coupled-adjoint method and the finite difference method. We consider eight step sizes, from 10^{-1} to 10^{-8}. Then, we choose the derivatives computed with the optimum step h_{opt}, which minimizes the sum of the truncation and subtractive cancellation errors [79].

The derivatives computed with the coupled-adjoint method differ from the finite difference method with relative errors ranging from 10^{-4} to 10^{-7}, which represents as good as an agreement as can be expected, given the errors inherent in finite difference approximations.

Table 2: Verification of coupled-adjoint gradients

Variable	Coupled adjoint (C_l)	Finite difference (C_l)	Rel. error (C_l)	h_{opt} (C_l)	Coupled adjoint (C_d)	Finite difference (C_d)	Rel. error (C_d)	h_{opt} (C_d)
1	−0.3660854201	−0.3660**934745**	2.20×10^{-5}	10^{-4}	0.0894652885	0.0894**77565**	2.76×10^{-5}	10^{-4}
2	0.3929952735	0.3929**63426**	2.72×10^{-6}	10^{-3}	0.0037564749	0.0037561**439**	-8.81×10^{-5}	10^{-3}
3	0.3982117282	0.3982**32122**	3.73×10^{-6}	10^{-3}	0.0028817686	0.0028816**980**	-2.45×10^{-5}	10^{-3}
4	0.6162273716	0.6162**42770**	-5.02×10^{-6}	10^{-3}	−0.0254760358	−0.025476**2670**	9.08×10^{-6}	10^{-3}
5	0.6709159055	0.6709**32341**	-3.98×10^{-6}	10^{-3}	−0.0279524925	−0.027952**8039**	1.11×10^{-5}	10^{-3}
6	0.4129534277	0.412**9652340**	2.86×10^{-5}	10^{-4}	0.0574631198	0.057**4629440**	-3.06×10^{-6}	10^{-4}
7	1.0951753415	1.0951**531170**	-1.84×10^{-6}	10^{-2}	−0.0091245450	−0.009124**8230**	3.05×10^{-5}	10^{-3}
8	1.7510676993	1.7510**69323**	-4.38×10^{-7}	10^{-3}	0.0065412416	0.006541**2558**	2.17×10^{-6}	10^{-3}
9	5.2954886067	5.2954**79960**	-1.15×10^{-7}	10^{-4}	0.0309890452	0.030989**3164**	8.75×10^{-6}	10^{-2}
10	2.9867870010	2.9867**766199**	-3.48×10^{-6}	10^{-5}	0.1700279389	0.170028**4219**	2.84×10^{-6}	10^{-3}
11	5.7235077212	5.7235**064851**	-2.16×10^{-7}	10^{-5}	0.0231841921	0.023184**3302**	5.96×10^{-6}	10^{-3}
12	1.8297703540	1.829770**1544**	-1.09×10^{-7}	10^{-3}	0.0102940232	0.010294**3202**	2.89×10^{-5}	10^{-2}
13	1.0060945300	1.006094**0367**	-4.90×10^{-7}	10^{-3}	0.0098241461	0.009824**4238**	2.83×10^{-5}	10^{-3}
14	0.5388777840	0.5388**01164**	4.33×10^{-6}	10^{-3}	0.0278680554	0.027867**7631**	-1.05×10^{-5}	10^{-3}
15	0.5306604993	0.530667**0045**	1.23×10^{-5}	10^{-4}	−0.0009847709	−0.000938**038**	-9.82×10^{-4}	10^{-3}
16	1.0219009205	1.021909**7890**	8.68×10^{-6}	10^{-3}	−0.1138294892	−0.113**8315144**	1.78×10^{-5}	10^{-3}
17	0.4243864364	0.424387**5222**	2.56×10^{-6}	10^{-3}	0.0007118109	0.000711**5004**	-4.36×10^{-4}	10^{-3}
18	0.3803439437	0.380340**7838**	-8.31×10^{-6}	10^{-3}	0.0161730351	0.016173**4646**	2.66×10^{-5}	10^{-3}
AOA	0.1150853223	0.115085**3584**	3.15×10^{-7}	10^{-2}	0.0011165774	0.001116**5577**	-1.76×10^{-5}	10^{-2}

3.4.4 Adjoint solution performance comparison

We now compare the computational cost of the two different coupled-adjoint solution methods. The transition prediction solution for this case takes 104.1 s using 15 processors for solving the RANS equations and one processor for solving the transition for a total of 16 CPU cores on an Intel Xeon E5-2630 v4 at 2.2 GHz. We use the adjoint of the C_d objective for the comparison, and we converge the adjoint residual to a relative tolerance of 10^{-8}. The results of are listed in Table 3, and the corresponding residual convergence histories are shown in Fig. 7.

Table 3: Gradient computation time comparison with 19 variables

Method	θ	N_{iter}	Time(s)
Finite difference	–	–	731.0
LBGS	0.5	31	137.3
LBGS	1.0	10	49.2
Coupled Krylov	–	52	42.8

The finite difference method is the most time consuming method: over 12 minutes for 19 design variables. The computational time of LBGS varies, depending on the

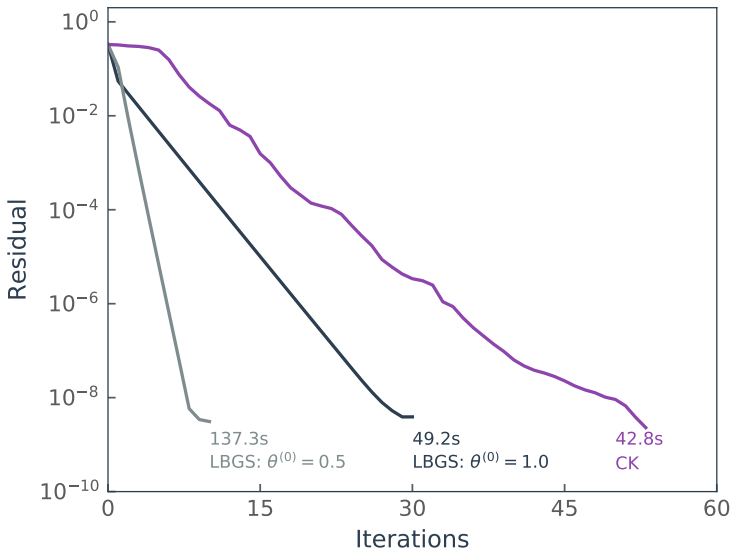


Figure 7: Coupled-adjoint convergence for different solution methods and parameters.

under-relaxation factor value. In this case, using $\theta = 1$ is much faster than 0.5. However, the best under-relaxation factor is case dependent. The coupled-Krylov solver is more efficient than LBGS, using 13% less time than the fastest LGBS solution. The subspace size for the coupled-Krylov solver was set to 20.

4 Optimization Results

To demonstrate our adjoint-based optimization framework for laminar flow design, we conduct single-point and multipoint laminar flow airfoil optimizations. Since TS waves are dominant in low-sweep wings, we select two applications with low sweep: the Cessna 172 SP Skyhawk and the HondaJet. The first is a low-speed and low-Reynolds-number application, whereas the second is a high subsonic speed with moderate-Reynolds-number application. We complete both transition-based and fully turbulent optimizations for both the single- and multipoint cases to assess the impact of the transition estimation on the optimal airfoil result.

The airfoil geometry is parameterized using FFD volumes with the pyGeo software [78] [see footnote [9]]. The FFD approach requires the user to provide a structured volume mesh that contains airfoil geometry of the airfoil (red points in Fig. 6). The FFD volume is represented by a trivariate B-spline, allowing the control points of this volume to manipulate the geometry. The airfoil geometry is mapped parametrically to the FFD B spline. Therefore, as the control points move, the embedded surface mesh deforms according to the B-spline mapping. The FFD control points move in the vertical direction to change the airfoil shape. Once the airfoil surface is deformed, the volume mesh in the CFD solver is updated using the IDwarp tool (see footnote

[9] https://github.com/mdolab/pygeo [retrieved September, 2020]

[10]), which implements a mesh-warping algorithm [80], with automatic differentiation in forward and reverse modes [55].

All the optimizations are performed with SNOPT [81]: a gradient-based optimizer that implements the sequential quadratic programming method. We call SNOPT through the pyOptSparse Python interface [82].[11] A feasibility tolerance of 10^{-6} is reached for all optimizations in this paper, which corresponds to six digits of accuracy for the lift coefficient. An optimality tolerance of 2.5×10^{-4} is satisfied by all results and, for the majority of the results, it is 10^{-5}.

4.1 Single-point Optimizations

The flow conditions are $M = 0.19$ and $Re = 5.6 \times 10^6$, where the lift coefficient is constrained to $C_l = 0.3$. This corresponds to the Cessna 172 SP Skyhawk at a typical cruise condition. We use the RAE 2822 airfoil as the initial design. Since laminar-to-turbulent transition is sensitive to the leading-edge geometry, we arrange the FFD control points to have finer shape control at the leading edge, as shown in Fig. 8.

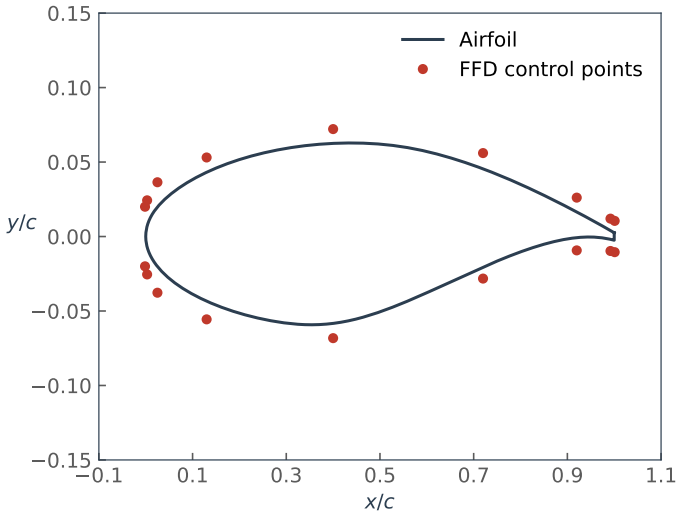


Figure 8: FFD control points for the initial airfoil of the single-point optimization.

The optimization parameters for the single-point optimization problem are listed in Table 4 [83]. The critical N factor for the TS waves N_{TScrit} is set to be 9.0, since the turbulent intensity is about 0.07% [68]. Cross-sectional area and thickness constraints are enforced to ensure that the optimal airfoil remains practical from the structural and manufacturing points of view. The area is constrained to not decrease relative to the initial airfoil ($S/S_{\mathrm{init}} \geqslant 1$), whereas the thickness is constrained such that $t_y \geqslant 0.3 t_{y_{\mathrm{init}}}$ and relative thickness constraints are enforced at 10 positions along the chord.

The initial and optimized results using both laminar-to-turbulent transition (LT) and fully turbulent (FT) models are defined separately. For instance, FT-optimized-

[10] https://github.com/mdolab/idwarp [retrieved September, 2020]
[11] https://github.com/mdolab/pyoptsparse [retrieved, September, 2020]

Table 4: Single-point optimization problem statement

Category	Name	Quantity	Lower	Upper	Units
Objective	C_d	1	–	–	–
Variables	AOA	1	0.0	4.0	degrees
	Shape	16	-0.02	0.02	m
Constraints	C_l	1	0.3	0.3	–
	S/S_{init}	1	1.0	30	m^2
	t_y	10	$0.3t_{y_{\text{init}}}$	–	m

FT means that we take the airfoil optimized using the fully turbulent model and then analyze it with the same fully turbulent model. FT-optimized-LT is the same optimized result, but analyzed using the transition model.

The designed geometry and corresponding pressure distribution from LT optimization are much more beneficial for extending the laminar region. Figure 10(a) shows that compared to the initial LT airfoil, the leading radius of the LT-optimized-LT airfoil is reduced and the position of maximum thickness is shifted aft. As a result, the favorable pressure gradient region (which suppresses TS wave amplification) is extended, delaying the laminar-to-turbulent transition (Fig. 10(b)). The transition location moves from 45.3% to 75.7% chord on the upper surface and from 47.9% to 73.4% chord on the lower surface, which demonstrates the ability of the optimizer to delay transition, and thus reduce the friction drag coefficient.

Consequently, the total drag coefficient, consisting of pressure drag coefficient and friction drag coefficient, is significantly reduced. The difference between the optimized and initial results is given by ΔC_d, which is shown in Table 5. The total drag coefficient of the LT-optimized-LT is reduced by 32.468 counts (52.27%) compared with the initial LT result (Table 5) for the LT optimization. The pressure drag coefficient reduction is 13.385 counts (63.87%) and the friction drag coefficient reduction is 19.084 counts (46.372%).

Table 5: Single-point optimization results

Geometry	Sim. Type	C_d	ΔC_d	C_l	C_m	x_{tr}/c (upper)	x_{tr}/c (lower)	AOA (degrees)
Initial	LT	62.112	–	0.3000	-0.0742	0.453	0.479	0.4499
Initial	FT	99.156	–	0.3000	-0.0702	–	–	0.5975
LT-optimized	LT	29.644	-32.468	0.2999	-0.0687	0.757	0.734	0.6361
FT-optimized	FT	89.708	-9.448	0.2999	-0.0447	–	–	1.1488
FT-optimized	LT	42.553	-19.559	0.2999	-0.0481	0.579	0.555	0.9809

Transition is caused by different mechanisms for the initial-LT airfoil and the LT-optimized-LT airfoil. We show a comparison of the N factor N_{TS} in Fig. 11. For the initial-LT airfoil, TS waves induce laminar-to-turbulent transition on the upper surface, whereas on the lower surface, the critical value for TS transition is not reached. On the lower surface, laminar separation is caused by the negative pressure gradient, which

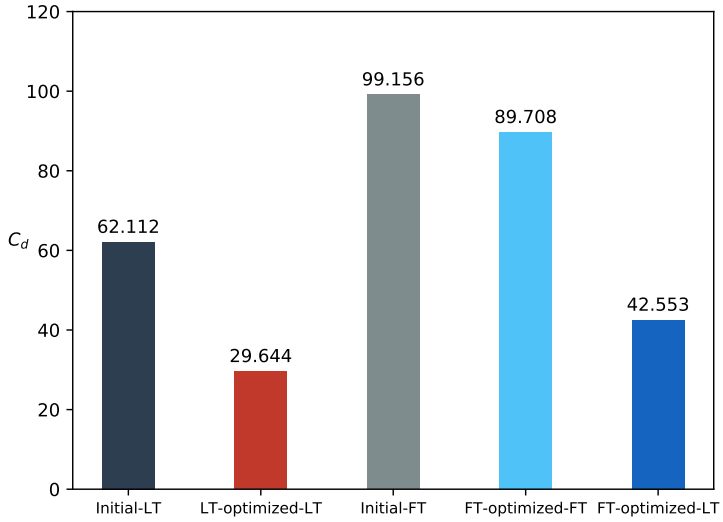
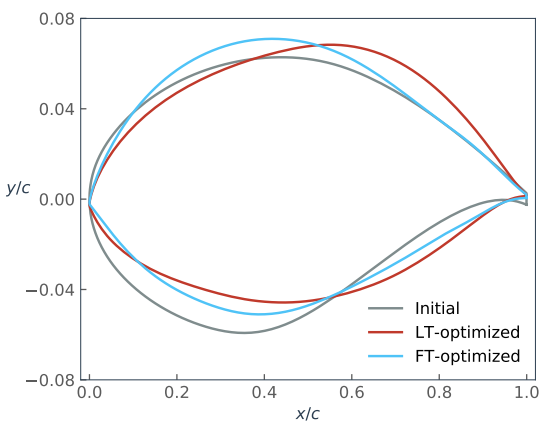
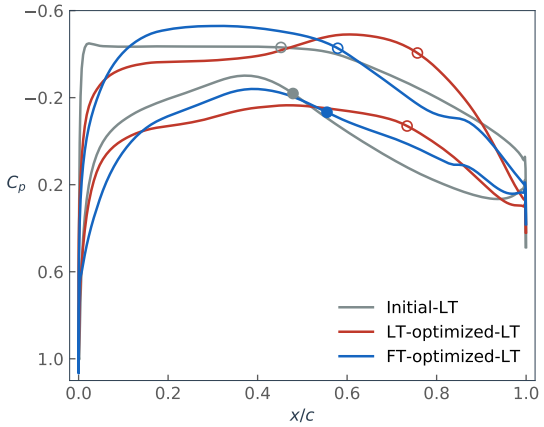


Figure 9: Single-point optimization comparison.



(a) Airfoil shape comparison.



(b) Pressure coefficient distribution comparison.

Figure 10: Single-point airfoils and pressure coefficient distribution results by fully laminar-to-turbulent transition prediction: TS-induced transition (hollow circles), and separation-induced transition (solid circles).

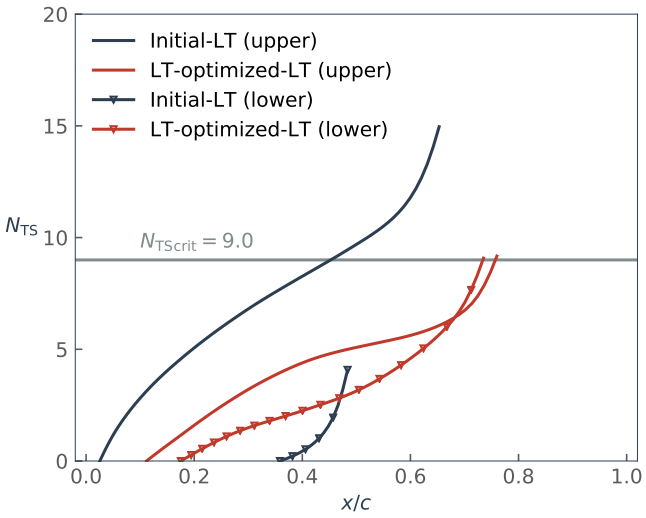


Figure 11: N-factor distribution comparison.

triggers transition. The lower surface N_{TS} curve terminates around the midchord point, where the laminar separation is triggered. For the LT-optimized-LT result, the N factor reaches the critical N factor at the transition point and the TS waves cause transition on both upper and lower surfaces.

The transition location is determined by the critical position and the disturbance amplification trend. For the LT-optimized-LT upper surface, the laminar region is extended by both moving the critical position aft and by lowering the disturbance amplification rate. For the LT-optimized-LT lower surface, the laminar region is extended, owing to the reduction in the disturbance amplification rate. The N-factor results demonstrate that the optimizer successfully explores the mechanisms to suppress boundary layer disturbances.

The entire improvement in the FT-optimized-FT total drag coefficient comes from the 9.448 count reduction in the pressure drag (9.53%), whereas the friction drag coefficient increases. This is expected, since transition is not considered. Figure 10(a) shows that the leading-edge camber of the FT optimized airfoil increases, and Fig. 12 shows that the pressure peak is increased, which reduces the pressure drag. The large difference between the LT and FT optimization results demonstrate the importance of considering transition in drag reduction when the flow conditions allow some laminar flow.

To emphasize the significance of using transition modeling, we compare the initial airfoil analyzed with and without transition (initial-LT and initial-FT, respectively), as well as the airfoil optimized without transition analyzed with and without transition ((FT-optimized-LT and FT-optimized-FT, respectively). Compared with the initial-FT, the total drag of the initial-LT has a lower drag by 37.044 counts (37.36%); and pressure drag coefficient and friction drag coefficient are 6.444 counts and 30.6 counts lower, respectively. The comparison shows the significant difference between the FT and LT simulations. The initial-LT and initial-FT pressure coefficient distributions

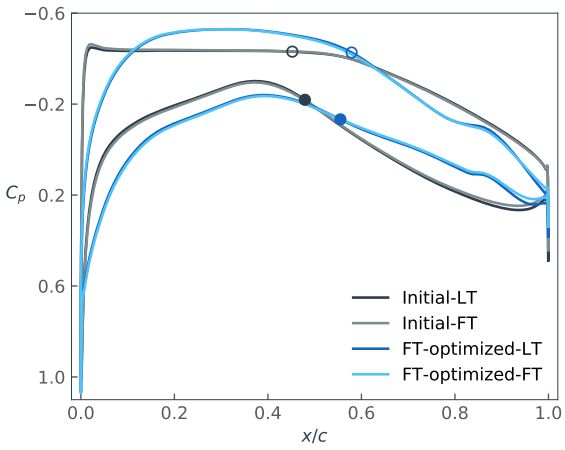


Figure 12: Single-point pressure coefficient distribution comparison by between laminar-to-turbulent transition prediction and fully turbulent: TS-induced transition (hollow circles), and separation-induced transition (solid circles).

(black and gray lines in Fig. 12, respectively) are similar overall, but there are some visible differences at the trailing edge. The C_p distribution differs due to the thinner laminar boundary layer, which results in a pressure drag coefficient that is lower by 6.444 counts. Similarly, the total drag for the FT-optimized-LT case is lower than value of the FT-optimization-FT case by 47.155 counts (52.56%). The C_p distributions are compared in Fig. 12. The difference in the C_p distributions is similar to the initial results, and it results in a lower pressure drag coefficient for LT simulation.

We further motivate the use of transition modeling in optimization by comparing LT and FT optimization with LT results (FT-optimized-LT and LT-optimized-LT). The total drag coefficient of the FT-optimized-LT case is lower than that of the initial-LT case by 19.559 counts (31.49%). The drag reduction of FT-optimized-LT (31.49%) is much smaller than that of the LT-optimized-LT (52.27%). Additionally, the laminar region extension of FT-optimized-LT is much shorter than that of LT-optimized-LT (Table 5 and Fig. 10(b)).

The LT and FT simulations of the optimal shapes emphasize the importance of considering transition for the simulation itself, as well as in the design optimization.

4.2 Multipoint Optimizations

Single point optimization does not yield practical results because the resulting aerodynamic performance is not robust with respect to the flight conditions [53]. This is especially important when considering transition because transition is especially sensitive to changes in the flight conditions. Therefore, we perform multipoint optimization to obtain more practical designs [54].

To demonstrate multipoint optimization, we start with the case proposed by Fujino et al. [4] as a baseline, which is inspired on the HondaJet aircraft. The flight conditions are listed in Table 6 and plotted in $M - C_l$ space (Fig. 13). Point 1 corresponds to the nominal cruise condition, and Point 3 is the nominal climb condition. Two other flight

conditions (Points 2 and 4) are used to reduce the sensitivity of the final optimized shape to perturbations in the flight conditions. Point 4 is included to limit the minimum negative pitching moment coefficient to -0.04 so that the trim-drag penalty at high-altitude and high-Mach-number cruise conditions is not too severe.

Table 6: Design points and weightings for multipoint optimization [4]

Point	Condition	Weights $\mathcal{T}_i$	M	C_l	Reynolds number
1	Cruise	1/2	0.69	0.26	11.7×10^6
2	Cruise	1/6	0.69	0.18	11.7×10^6
3	Climb	1/6	0.31	0.35	13.6×10^6
4	Cruise	1/6	0.70	0.38	7.93×10^6

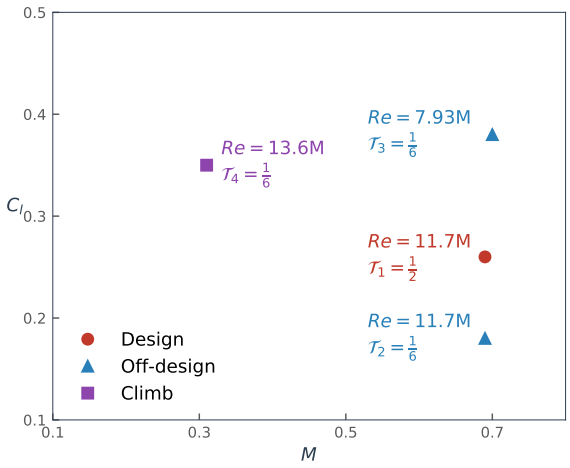


Figure 13: The multipoint optimization flight conditions representing a four-point in $M - C_l$ space.

The optimization objective is the weighted sum of the drag coefficient at these four conditions using the weight values listed in Table 6. The multipoint optimization problem is detailed in Table 7. The area and thickness constraints are the same as for the single-point optimizations ($S/S_{\text{init}} \geqslant 1$ and $t_y \geqslant 0.3t_{y_{\text{init}}}$). The initial geometry (Fig. 14) has typical NLF airfoil properties with a favorable pressure coefficient distribution. The grid of multipoint case is similar to the single-point mesh. The initial airfoil and the corresponding FFD control points are shown in Fig. 14. We use 24 control points, and the leading edge is arranged with finer control points. The TS wave critical factor N_{TScrit} is same with the single point and is 9.0.

As before, the results are defined by different optimization and simulation types, such as LT-optimized-LT, FT-optimized-FT, and etc. ΔC_d in Table 8 is the difference between the optimized and initial drag coefficients for the same flight condition.

The airfoil shape and pressure coefficient results are compared in Fig. 15 and Fig. 16. Compared with the initial-LT results (gray lines in Fig. 16), the suction peak of LT-optimized-LT result (red lines in Fig. 16) decreases, reducing the favorable pressure

Table 7: Multipoint optimization problem statement

Category	Name	Quantity	Lower	Upper	Units
Objective	$\sum_{i=1}^{N} \boldsymbol{\mathcal{T}}_i C_{d_i}$	1	–	–	–
Variables	AOA_i	4			degrees
	Shape	24			m
Constraints	C_{l_i}	4	C_{l_i}	C_{l_i}	–
	$C_{m_{i=4}}$	1	-0.040	0.0	–
	S/S_{init}	1	1.0	30	m^2
	t_y	10	$0.3t_{y_{\text{init}}}$	–	m

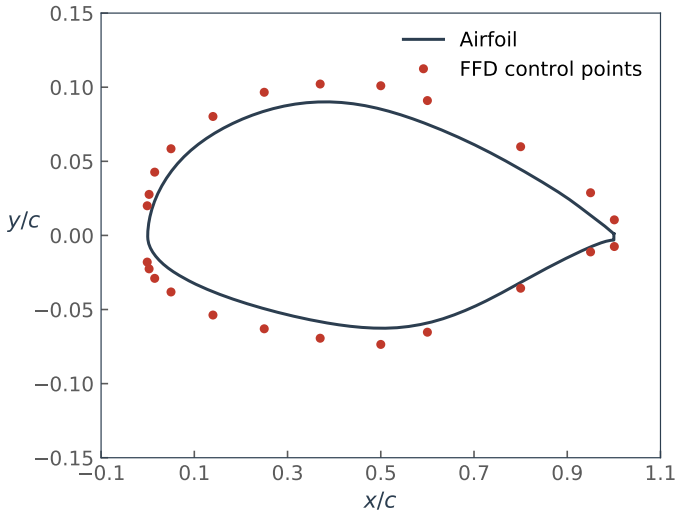


Figure 14: FFD control points for the initial airfoil of multipoint optimization.

gradient on the upper surface for the LT optimization results. There are two main reasons for this change: a lower favorable pressure expands the favorable-gradient region, which delays the pressure recovery position and allows the laminar region to be extended in three out of four operating points; and a lower C_p peak reduces the wave drag substantially at high Mach numbers. When shock waves are present, a less favorable pressure gradient can reduce the shock strength and consequently wave drag is decreased. Therefore, in this type of flow, a tradeoff between laminarization and wave drag control must be performed; and these results represent the optimal tradeoff for this particular case. On the lower surface, the optimized results have a much longer favorable pressure gradient to extend the laminar region.

As a result, the LT optimization successfully reduces the drag coefficient at each flight condition and delays transition in general, and the force and transition location results and comparisons are shown in Table 8 and Fig. 17. For Point 1 in the LT optimization, a 33.2% total drag coefficient reduction is achieved. The transition location is extended from 41.3% chord to 58.3% chord on the upper surface and from 56.1% chord to 70.9% chord on the lower surface. For Point 2, we obtain a total drag

Table 8: Multipoint optimization results

Point	Geometry	Sim. Type	C_d	ΔC_d	C_l	C_m	x_{tr}/c (upper)	x_{tr}/c (lower)	AOA (degrees)
1	Initial	LT	49.263	–	0.2600	−0.0497	0.413	0.583	−0.3919
	Initial	FT	94.614	–	0.2600	−0.0409	–	–	−0.1498
	LT-optimized	LT	32.929	−16.334	0.2599	−0.0262	0.561	0.709	−0.1981
	FT-optimized	FT	91.575	−3.039	0.2599	−0.0329	–	–	0.9447
	FT-optimized	LT	56.003	6.740	0.2600	−0.0384	0.324	0.393	0.7855
2	Initial	LT	60.854	–	0.1800	−0.0489	0.430	0.344	−0.9449
	Initial	FT	93.205	–	0.1800	−0.0400	–	–	−0.6393
	LT-optimized	LT	32.214	−28.640	0.1801	−0.0233	0.563	0.706	−0.6167
	FT-optimized	FT	90.257	−2.948	0.1800	−0.0335	–	–	0.4778
	FT-optimized	LT	54.231	−6.623	0.1800	−0.0390	0.366	0.360	0.3151
3	Initial	LT	47.451	–	0.3500	−0.0441	0.404	0.638	0.8559
	Initial	FT	89.702	–	0.3500	−0.0389	–	–	1.0459
	LT-optimized	LT	42.926	−4.525	0.3499	−0.0244	0.375	0.730	1.1867
	FT-optimized	FT	87.789	−1.913	0.3499	−0.0305	–	–	2.1972
	FT-optimized	LT	61.996	14.545	0.3499	−0.0321	0.130	0.518	2.1686
4	Initial	LT	60.242	–	0.3800	−0.0514	0.473	0.618	0.2977
	Initial	FT	114.480	–	0.3800	−0.0398	–	–	0.6496
	LT-optimized	LT	39.849	−20.393	0.3799	−0.0269	0.503	0.748	0.4030
	FT-optimized	FT	99.646	−14.834	0.3799	−0.0298	–	–	1.6160
	FT-optimized	LT	58.803	−1.439	0.3799	−0.0366	0.308	0.476	1.4196

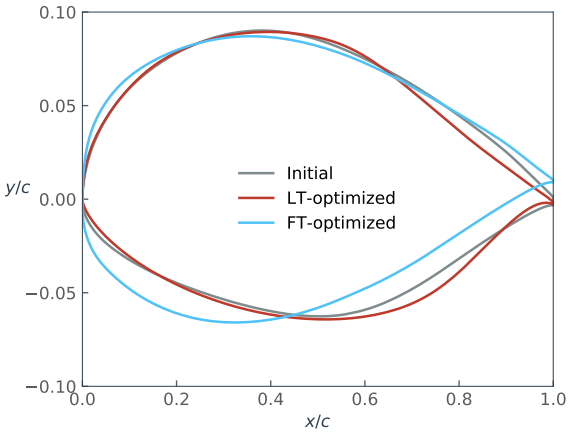


Figure 15: Multipoint airfoil optimization results.

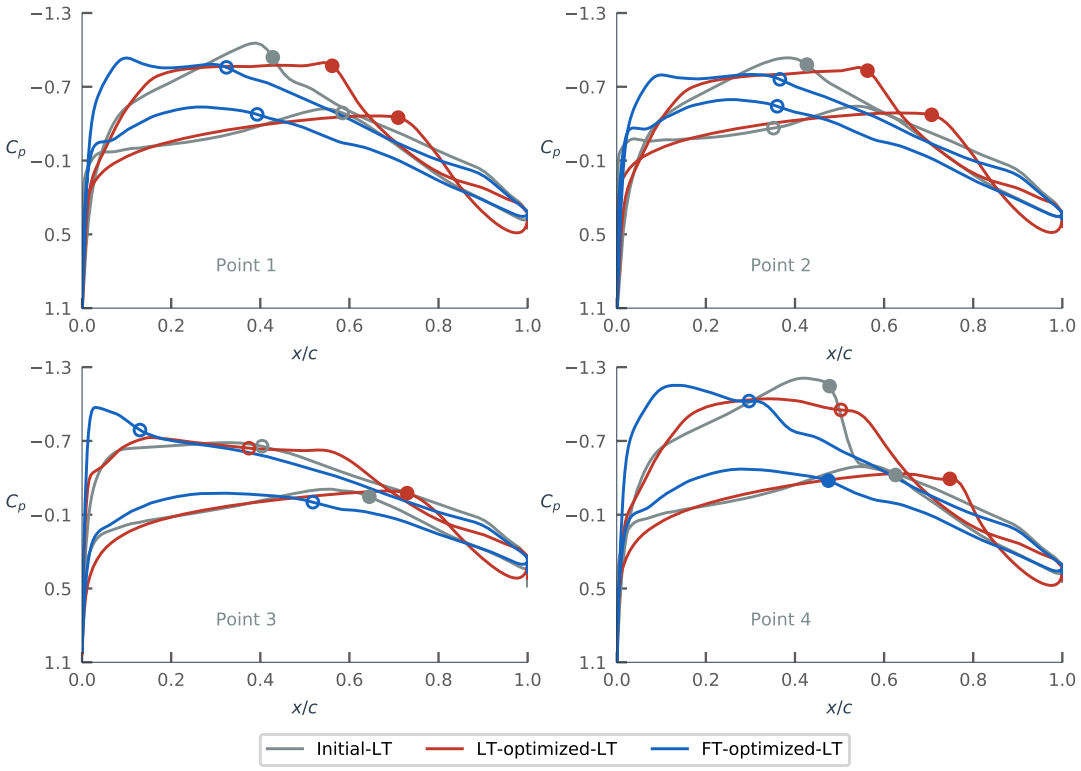
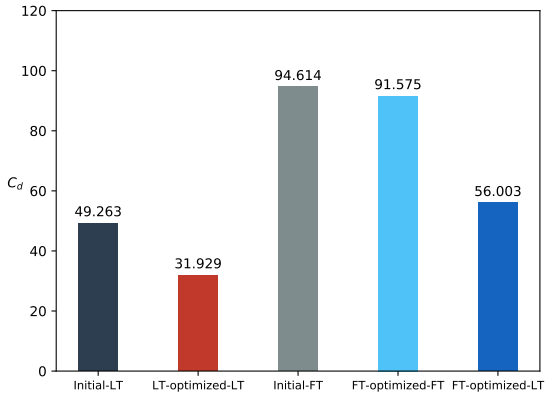
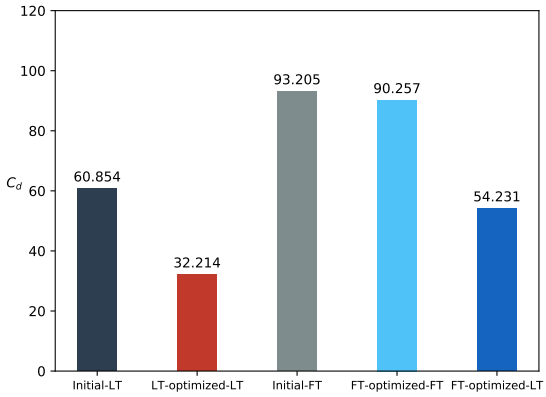


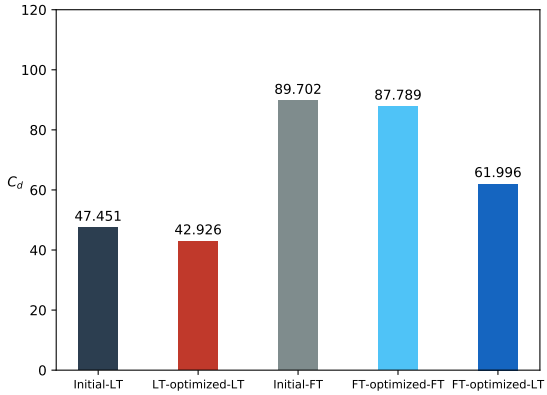
Figure 16: Pressure coefficient distribution predicted by fully laminar-to-turbulent transition simulation at different flight conditions: TS-induced transition (hollow circles), separation-induced transition (solid circles).



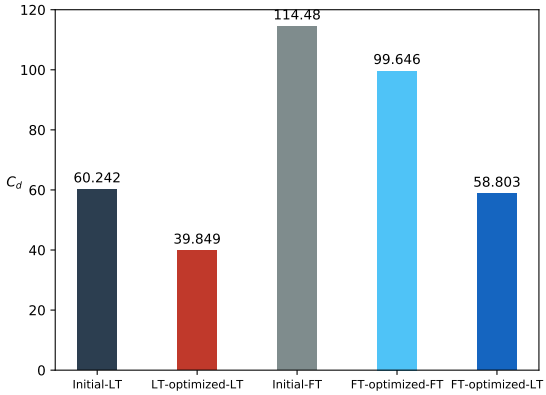
(a) Point 1



(b) Point 2



(c) Point 3



(d) Point 4

Figure 17: Multipoint optimization: simulation comparison results by bar chart.

reduction of 28.64 counts (47.0%). The transition location is extended from 43.0% chord to 56.3% chord on the upper surface and from 34.4% chord to 70.6% chord on the lower surface. For the climb case (Point 3), the LT-optimized-LT solution shows a total drag coefficient reduction of 4.525 counts (9.5%). The laminar region is slightly reduced from 40.4% chord to 37.5% on the upper surface, but laminar region on the lower surface is extended from 63.8% chord to 73%. For Point 4, the shock wave is almost removed after optimization, as shown in Fig. 16, and the transition locations are moved aft on both the upper and lower surfaces. The total drag coefficient is reduced by 20.393 counts (33.85%), where the main contribution is from a pressure drag coefficient reduction.

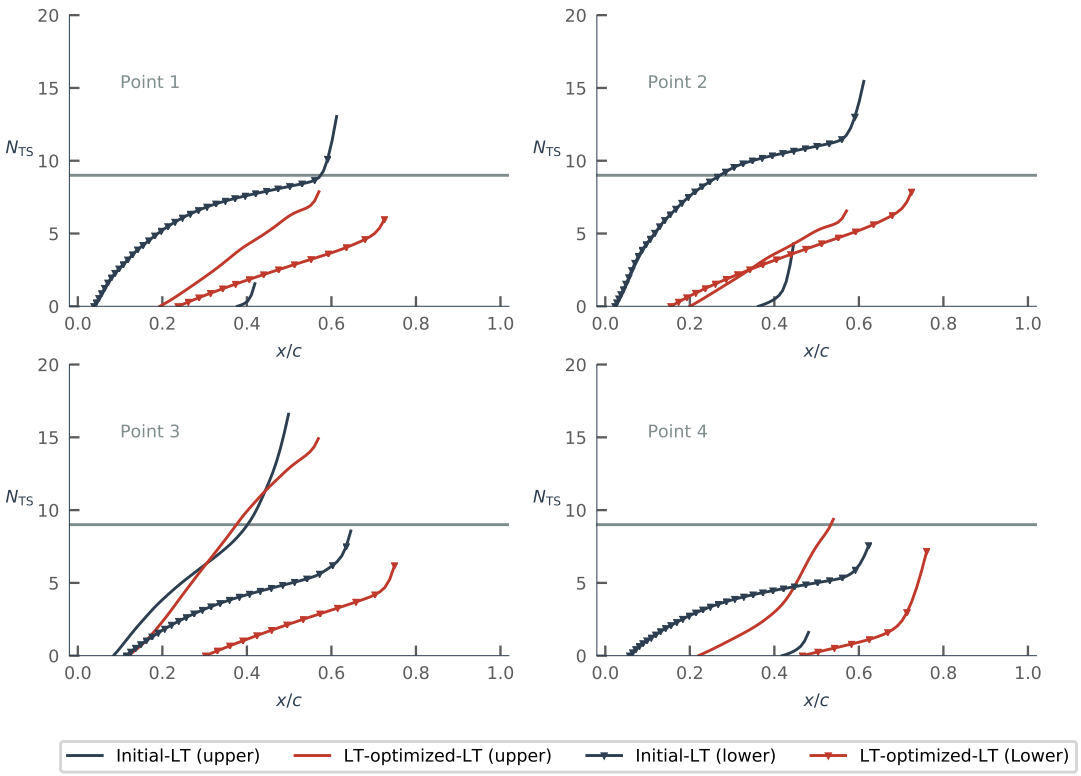


Figure 18: N-factor distribution at different flight conditions. Upper and lower denote upper and lower surfaces, respectively

The N-factor results' comparison in Fig. 18 reveals the different transition mechanisms and how the transition locations are moved fore and aft. For Point 1, laminar separation cause the transition on the upper surface for the initial-LT and LT-optimized-LT cases. The N factor of LT-optimized-LT on upper surface is enlarged but does not reach the critical value; as a result, the laminar separation point moves aft and the laminar region is extended. On the lower surface, TS waves trigger the transition for the initial-LT case, whereas laminar separation occurs for the LT-optimized-LT case. The N factor is suppressed on the lower surface, thanks to a more favorable pressure gradient, as shown in Fig. 16. The transition mechanism for Point 2 is same as for

Point 1. Both the upper surface and lower surface gain longer laminar region. For Point 3, the upper surface N factor of optimized airfoil is slightly enlarged, and the transition location moves upstream. However, the N factor for the LT-optimized-LT case is reduced due to a much more favorable pressure coefficient distribution on the lower surface, as shown in Fig. 16. For Point 4, the optimized upper surface N factor increases to the critical N-factor value at the transition point.

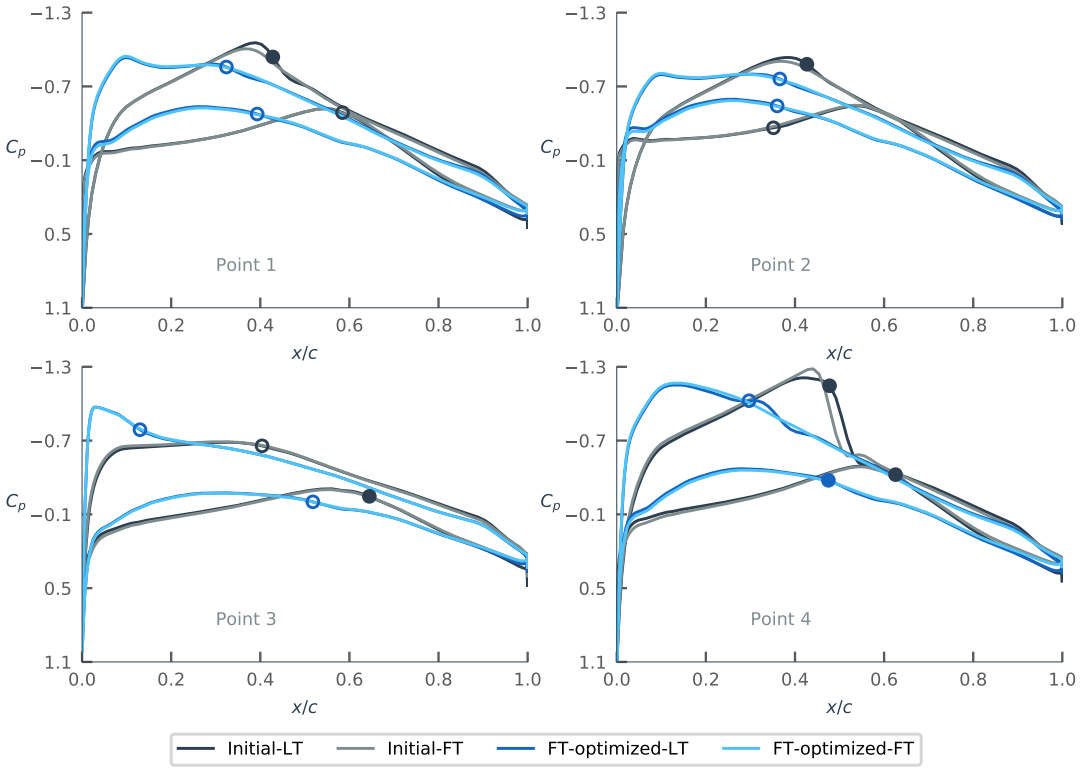


Figure 19: Pressure coefficient distribution comparison between laminar-to-turbulent transition prediction and fully turbulent at different flight conditions: TS-induced transition (hollow circles), separation-induced transition (solid circles).

We compare the LT optimization results with FT optimization results to once again demonstrate the potential for laminar airfoil drag reduction. For the FT optimization results, the total drag reduction is 3.039 counts (3.21%) for Point 1, 2.635 counts (2.83%) for Point 2, 1.913 counts (2.13%) for Point 3, and 14.834 counts (12.96%) for Point 4. For Point 4, since the FT optimization removes the shock wave, the pressure drag coefficient is significantly reduced, as shown in the light blue line in Fig. 19. The drag reduction for the FT optimization is much lower than that of all LT optimization results (Fig. 17). Even for Point 4, the LT optimization reduces the pressure drag coefficient much more than the FT optimization.

We also compare the cases for initial-LT and initial-FT, FT-optimized-LT and FT-optimized-FT to emphasize the importance of considering transition in the CFD simulation process. The general conclusions from this comparison are similar to the

single-point case. The drag coefficient differences are due to a smaller boundary-layer thickness in laminar flow compared to turbulent flow. The LT simulation results have both lower pressure drag coefficient and lower friction drag coefficient than the FT simulation (shown in Table 8). It is worth demonstrating that a dramatic pressure recovery (shock wave for Point 4) shown in Fig. 19 at a higher Mach number corresponds to a large difference in the pressure recovery region, which is also in the transition region.

For this multipoint optimization, the total drag of the FT-optimized-LT is even larger than initial-LT for Point 1 and Point 3 (Table 8 and Fig. 17). The transition locations of FT-optimized-LT solution are upstream relative to those of the initial-LT, except for the lower surface result for Point 2. This demonstrates the importance of considering the transition in the optimization process.

Given that the performance of the LT-optimized airfoil is improved relative to the initial airfoil at every point, we conclude that the multipoint optimization is successful in designing a robust NLF airfoil. Similarly to the single-point optimizations, these multipoint optimizations show that it is important to consider laminar-to-turbulent transition, because optimization without considering transition leads to suboptimal results.

5 Conclusions

A methodology for natural laminar flow simulation, derivative computation, and design optimization is presented. A simplified e^N method is coupled with a RANS solver using an intermittency function for laminar-to-turbulent transition prediction. The e^N method includes a laminar boundary-layer solution and a database method for stability analysis. The block Gauss–Seidel method with Aitken acceleration is applied to solve the coupled RANS-e^N equations. The transition prediction tool is validated by comparing CFD results with experimental data for the NLF(1)-0416 airfoil. All predicted transition locations were verified to match the experimental measurement within experimental error.

Smoothness in the design space is ensured by interpolating the intermittency function over the transition length so that the transition location varies smoothly as the airfoil shape changes. This interpolation ensures that the transition location in the CFD solution does not exhibit abrupt changes between the discrete mesh points, making it amenable to gradient-based optimization.

The derivatives considering the transition modeling are computed by solving the coupled-adjoint for the RANS-e^N equations. Reverse AD is used to generate the code that computes the required partial derivatives. Two methods are used for solving the coupled-adjoint system: a linear block Gauss–Seidel method, and a fully coupled-Krylov method. The coupled-Krylov method method is 13% faster than the linear block Gauss–Seidel method. The derivative computation time for one objective with respect to 19 design variables is about 41.1% of the flow solution cost. The computed coupled derivatives are verified using two approaches: the dot-product test, and the finite difference method. Since all of the augmented subroutines agree to machine

precision for the dot-product tests, the AD code is consistent. A good agreement with finite difference derivative approximations is demonstrated for the derivatives, with an agreement of five digits or more. Therefore, the coupled-adjoint-based system is shown to be accurate and efficient.

The developed methodology is used to perform the gradient-based shape optimization of an airfoil for both the laminar-to-turbulent transition case and for the fully-turbulent case. The single-point laminar airfoil optimization reduces the drag coefficient by 52.27%, owing to an extension of the laminar region to 75.7% chord on the upper surface and 73.4% chord on the lower surface. The multipoint laminar optimization at transonic conditions successfully trades friction drag and pressure drag, extending the transition locations and reducing the shock wave drag. As a result, we achieve a drag coefficient reductions of 33.2% at $C_l = 0.26$, 47.0% at $Cl = 0.18$, 9.5% at $C_l = 0.35$, and 33.85% at $C_l = 0.38$, demonstrating a more practical design.

These applications demonstrate the practicality of CFD adjoint-based aerodynamic optimization as a useful NLF design tool and highlight the benefits of laminar drag reduction. With the same optimization problem, the fully turbulent case yields significantly lower reductions than the ones achieved in the laminar transition case (9.53% drag reduction for the single-point optimization; and a reductions of 3.21% at $C_l = 0.26$, 2.83% at $Cl = 0.18$, 2.13% at $C_l = 0.35$, and 12.96% at $C_l = 0.38$ for the multipoint case). These results stress the significant advantages of NLF airfoil for drag reduction. Also, the laminar-to-turbulent transition simulation results of the fully-turbulent airfoil, which are closer to the real physics compared with fully turbulent results, show that the airfoil optimized assuming fully turbulent flow produced even higher drag than the initial airfoil when simulated considering laminar-to-turbulent transition. This highlights the importance of considering transition when optimizing in the relevant flow regimes.

References

[1] Campbell, R. L., and Lynde, M. N., "Building a Practical Natural Laminar Flow Design Capability," *35th AIAA Applied Aerodynamics Conference*, 2017, pp. 1–21. doi:10.2514/6.2017-3059.

[2] Cella, U., Quagliarella, D., Donelli, R., and Imperatore, B., "Design and Test of the UW-5006 Transonic Natural-Laminar-Flow Wing," *Journal of Aircraft*, Vol. 47, No. 3, 2010, pp. 783–795. doi:10.2514/1.40932.

[3] Roberto dePompeis, P. C., and Martini, P. S., "Development and Certification Flight Test on the Piaggio P.180 Avanti Aircraft: A General Overview," Technical Note SAE Technical Paper 911003, SAE International, 1991. doi:10.4271/911003.

[4] Fujino, M., Yoshizaki, Y., and Kawamura, Y., "Natural-Laminar-Flow Airfoil Development for a Lightweight Business Jet," *Journal of Aircraft*, Vol. 40, No. 4, 2003, pp. 609–615. doi:10.2514/2.3145.

[5] Crouch, J., "Modeling Transition Physics for Laminar Flow Control," , No. June, 2008, pp. 1–9. doi:10.2514/6.2008-3832.

[6] Eppink, J. L., "The Effect of Forward-Facing Steps on Stationary Crossflow Instability Growth and Breakdown," *AIAA Aerospace Sciences Meeting*, 2018. doi:10.2514/6.2018-0817.

[7] Coder, J. G., and Maughmer, M. D., "Computational Fluid Dynamics Compatible Transition Modeling Using an Amplification Factor Transport Equation," *AIAA Journal*, Vol. 52, No. 11, 2014, pp. 2506–2512. doi:10.2514/1.J052905.

[8] Langtry, R. B., and Menter, F. R., "Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes," *AIAA Journal*, Vol. 47, No. 12, 2009, pp. 2894–2906. doi:10.2514/1.42362.

[9] Krumbein, A., "En Transition Prediction for 3d Wing Configurations Using Database Methods and a Local, Linear Stability Code," *Aerospace Science and Technology*, Vol. 12, No. 8, 2008, pp. 592–598. doi:10.1016/j.ast.2008.01.005.

[10] Rashad, R., and Zingg, D. W., "Aerodynamic Shape Optimization for Natural Laminar Flow Using a Discrete-Adjoint Approach," *AIAA Journal*, Vol. 54, No. 11, 2016, pp. 3321–3337. doi:10.2514/1.J054940.

[11] Lee, D. S., Fahey, D. W., Forster, P. M., Newton, P. J., Wit, R. C., Lim, L. L., Owen, B., and Sausen, R., "Aviation and global climate change in the 21st century," *Atmospheric Environment*, Vol. 43, No. 22–23, 2009, pp. 3520–3537. doi:10.1016/j.atmosenv.2009.04.024.

[12] Arnal, D., and Casalis, G., "Laminar-Turbulent Transition Prediction in Three-Dimensional Flows," *Progress in Aerospace Sciences*, Vol. 36, No. 2, 2000, pp. 173–191. doi:10.1016/S0376-0421(00)00002-6.

[13] Fu, S., and Wang, L., "RANS Modeling of High-Speed Aerodynamic Flow Transition with Consideration of Stability Theory," *Progress in Aerospace Sciences*, Vol. 58, 2013, pp. 36–59. doi:10.1016/j.paerosci.2012.08.004.

[14] Vermeersch, O., Yoshida, K., Ueda, Y., and Arnal, D., "Natural Laminar Flow Wing for Supersonic Conditions: Wind Tunnel Experiments, Flight Test and Stability Computations," *Progress in Aerospace Sciences*, Vol. 79, 2015, pp. 64–91. doi:10.1016/j.paerosci.2015.07.003.

[15] Krishnan, K. S., Bertram, O., and Seibel, O., "Review of Hybrid Laminar Flow Control Systems," *Progress in Aerospace Sciences*, Vol. 93, No. February, 2017, pp. 24–52. doi:10.1016/j.paerosci.2017.05.005.

[16] Mayer, C. S., Von Terzi, D. A., and Fasel, H. F., "Direct Numerical Simulation of Complete Transition to Turbulence via Oblique Breakdown at Mach 3," *Journal of Fluid Mechanics*, Vol. 674, 2011, pp. 5–42. doi:10.1017/S0022112010005094.

[17] Ducros, F., Comte, P., and Lesieur, M., "Large-Eddy Simulation of Transition to Turbulence in a Boundary Layer Developing Spatially over a Flat Plate," *Journal of Fluid Mechanics*, Vol. 326, 1996, pp. 1–36. doi:10.1017/S0022112096008221.

[18] Krumbein, A., Krimmelbein, N., and Schrauf, G., "Automatic Transition Prediction in Hybrid Flow Solver, Part 1: Methodology and Sensitivities," *Journal of Aircraft*, Vol. 46, No. 4, 2009, pp. 1176–1190. doi:10.2514/1.39736.

[19] Krumbein, A., Krimmelbein, N., and Schrauf, G., "Automatic Transition Prediction in Hybrid Flow Solver, Part 2: Practical Application," *Journal of Aircraft*, Vol. 46, No. 4, 2009, pp. 1191–1199. doi:10.2514/1.39738.

[20] Coder, J. G., "Enhancement of the Amplification Factor Transport Transition Modeling Framework," *55th AIAA Aerospace Sciences Meeting*, 2017, pp. 1–14. doi:10.2514/6.2017-1709.

[21] Xu, J., Bai, J., Qiao, L., and Zhang, Y., "Development of a Computational Fluid Dynamics Compatible Mathematical Model for Boundary Layer Transitional Flows in Low-Disturbance Environment," *Aerospace Science and Technology*, Vol. 86, 2019, pp. 487–496. doi:10.1016/j.ast.2019.01.036.

[22] Grabe, C., and Krumbein, A., "Correlation-Based Transition Transport Modeling for Three-Dimensional Aerodynamic Configurations," *Journal of Aircraft*, Vol. 50, No. 5, 2013. doi:10.2514/1.C032063.

[23] Langtry, R., "Extending the Gamma-Rethetat Correlation Based Transition Model for Crossflow Effects (Invited)," 2015, pp. 1–12. doi:10.2514/6.2015-2474.

[24] Choi, J. H., and Kwon, O. J., "Enhancement of a Correlation-Based Transition Turbulence Model for Simulating Crossflow Instability," *AIAA Journal*, Vol. 53, No. 10, 2015, pp. 3063–3072. doi:10.2514/1.J053887.

[25] Drela, M., and Giles, M. B., "Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils," *Journal of Aircraft*, Vol. 25, No. 10, 1987, pp. 1347–1355.

[26] Krimmelbein, N., Krumbein, A., and Grabe, C., "Validation of Transition Modeling Techniques for a Simplified Fuselage Configuration," 2018, pp. 1–36. doi:10.2514/6.2018-0030.

[27] Streit, T., Horstmann, K., Schrauf, G., Hein, S., Fey, U., Egami, Y., Perraud, J., El Din, I. S., Cella, U., and Quest, J., "Complementary Numerical and Experimental Data Analysis of the Etw Telfona Pathfinder Wing Transition Tests," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011, pp. 1–16. doi:10.2514/6.2011-881.

[28] Shi, Y., Gross, R., Mader, C. A., and Martins, J. R. R. A., "Transition Prediction Based on Linear Stability Theory with the RANS Solver for Three-Dimensional

Configurations," *Proceedings of the AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, Kissimmee, FL, 2018. doi:10.2514/6.2018-0819.

[29] Shi, R., Liu, L., Long, T., Wu, Y., and Tang, Y., "Filter-based adaptive Kriging method for black-box optimization problems with expensive objective and constraints," *Computer Methods in Applied Mechanics and Engineering*, Vol. 347, 2019, pp. 782–805. doi:10.1016/j.cma.2018.12.026.

[30] Arnal, D., "Transition Prediction In Transonic Flow," *IUTAM Symposium Transsonicum III DFVLR-AVA*, 1988, pp. 253–262. doi:10.2514/6.1998-2824.

[31] Perraud, J., Arnal, D., Casalis, G., Archambaud, J.-P., and Donelli, R., "Automatic Transition Predictions Using Simplified Methods," *AIAA Journal*, Vol. 47, No. 11, 2009, pp. 2676–2684. doi:10.2514/1.42990.

[32] Halila, G. L. O., Chen, G., Shi, Y., Fidkowski, K. J., Martins, J. R. R. A., and de Mendonça, M. T., "High-Reynolds Number Transitional Flow Simulation via Parabolized Stability Equations with an Adaptive RANS Solver," *Aerospace Science and Technology*, Vol. 91, 2019, pp. 321–336. doi:10.1016/j.ast.2019.05.018.

[33] Bégou, G., Deniau, H., Vermeersch, O., and Casalis, G., "Database Approach for Laminar-Turbulent Transition Prediction: Navier–Stokes Compatible Reformulation," *AIAA Journal*, Vol. 55, No. 11, 2017, pp. 3648–3660. doi:10.2514/1.j056018.

[34] Yildirim, A., Kenway, G. K. W., Mader, C. A., and Martins, J. R. R. A., "A Jacobian-free approximate Newton–Krylov startup strategy for RANS simulations," *Journal of Computational Physics*, Vol. 397, 2019, p. 108741. doi:10.1016/j.jcp.2019.06.018.

[35] Kenway, G. K. W., Mader, C. A., He, P., and Martins, J. R. R. A., "Effective Adjoint Approaches for Computational Fluid Dynamics," *Progress in Aerospace Sciences*, Vol. 110, 2019, p. 100542. doi:10.1016/j.paerosci.2019.05.002.

[36] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/BF01061285.

[37] Mavriplis, D. J., "Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes," *AIAA Journal*, Vol. 45, No. 4, 2007, pp. 741–750. doi:10.2514/1.22743.

[38] Mader, C. A., Martins, J. R. R. A., Alonso, J. J., and van der Weide, E., "ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers," *AIAA Journal*, Vol. 46, No. 4, 2008, pp. 863–873. doi:10.2514/1.29123.

[39] Brezillon, J., and Dwight, R. P., "Applications of a discrete viscous adjoint method for aerodynamic shape optimisation of 3D configurations," *CEAS Aeronautical Journal*, Vol. 3, No. 1, 2012, pp. 25–34. doi:10.1007/s13272-011-0038-0.

[40] Mader, C. A., and Martins, J. R. R. A., "Computing Stability Derivatives and their Gradients for Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 52, No. 11, 2014, pp. 2533–2546. doi:10.2514/1.J052922.

[41] Albring, T., Sagebaum, M., and Gauger, N. R., *New Results in Numerical and Experimental Fluid Mechanics X: Contributions to the 19th STAB/DGLR Symposium Munich, Germany, 2014*, Springer International Publishing, Cham, 2016, Chaps. A Consistent and Robust Discrete Adjoint Solver for the SU2 Framework—Validation and Application, pp. 77–86. doi:10.1007/978-3-319-27279-5_7.

[42] Nielsen, E. J., and Diskin, B., "Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," *AIAA Journal*, Vol. 51, No. 6, 2013, pp. 1355–1373. doi:10.2514/1.J051859.

[43] He, P., Mader, C. A., Martins, J. R. R. A., and Maki, K. J., "An Aerodynamic Design Optimization Framework Using a Discrete Adjoint Approach with OpenFOAM," *Computers & Fluids*, Vol. 168, 2018, pp. 285–303. doi:10.1016/j.compfluid.2018.04.012.

[44] Müller, J.-D., Mykhaskiv, O., and Hückelheim, J., "STAMPS: a Finite-Volume Solver Framework for Adjoint Codes Derived with Source-Transformation AD," *AIAA Multidisciplinary Analysis and Optimization Conference*, 2018. doi:10.2514/6.2018-2928.

[45] Zhang, P., Lu, J., Wang, Z., Song, L., and Feng, Z., "Adjoint-Based Optimization Method with Linearized Sst Turbulence Model and a Frozen Gamma-Theta Transition Model Approach for Turbomachinery Design," *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, American Society of Mechanical Engineers, 2015, pp. 19–34. doi:doi:10.1115/GT2015-42582.

[46] Lyu, Z., Kenway, G. K., Paige, C., and Martins, J. R. R. A., "Automatic Differentiation Adjoint of the Reynolds-Averaged Navier–Stokes Equations with a Turbulence Model," *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 2013. doi:10.2514/6.2013-2581.

[47] Khayatzadeh, P., and Nadarajah, S., "Aerodynamic Shape Optimization of Natural Laminar Flow (NLF) airfoils," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2012, pp. 1–15. doi:10.2514/6.2012-61.

[48] Lee, J.-D., and Jameson, A., "Natural-Laminar-Flow Airfoil and Wing Design by Adjoint Method and Automatic Transition Prediction," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, 2009. doi:10.2514/6.2009-897.

[49] Lee, J.-D., and Jameson, A., "NLF Wing Design by Adjoint Method and Automatic Transition Prediction," Ph.D. thesis, Stanford University, Stanford, CA, USA, 2009.

[50] Driver, J., and Zingg, D. W., "Numerical Aerodynamic Optimization Incorporating Laminar-Turbulent Transition Prediction," *AIAA Journal*, Vol. 45, No. 8, 2007, pp. 1810–1818. doi:10.2514/1.23569.

[51] Drela, M., "Design and Optimization Method for Multi-Element Airfoils Aerospace Design Conference," *Proceedings of the AIAA/AHS/ASEE Aerospace Design Conference*, Irvine, CA, 1993. doi:10.2514/6.1993-969.

[52] Lyu, Z., and Martins, J. R. R. A., "Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft," *Journal of Aircraft*, Vol. 51, No. 5, 2014, pp. 1604–1617. doi:10.2514/1.C032491.

[53] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968–985. doi:10.2514/1.J053318.

[54] Kenway, G. K. W., and Martins, J. R. R. A., "Multipoint Aerodynamic Shape Optimization Investigations of the Common Research Model Wing," *AIAA Journal*, Vol. 54, No. 1, 2016, pp. 113–128. doi:10.2514/1.J054154.

[55] Secco, N. R., Jasa, J. P., Kenway, G. K. W., and Martins, J. R. R. A., "Component-based Geometry Manipulation for Aerodynamic Shape Optimization with Overset Meshes," *AIAA Journal*, Vol. 56, No. 9, 2018, pp. 3667–3679. doi:10.2514/1.J056550.

[56] Kenway, G. K. W., Secco, N., Martins, J. R. R. A., Mishra, A., and Duraisamy, K., "An Efficient Parallel Overset Method for Aerodynamic Shape Optimization," *Proceedings of the 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, Grapevine, TX, 2017. doi:10.2514/6.2017-0357.

[57] Spalart, P., and Allmaras, S., "A One-Equation Turbulence Model for Aerodynamic Flows," *30th Aerospace Sciences Meeting and Exhibit*, 1992. doi:10.2514/6.1992-439.

[58] Liao, W., Malik, M. R., Lee-Rausch, E. M., Li, F., Nielsen, E. J., Buning, P. G., Choudhari, M., and Chang, C.-L., "Boundary-Layer Stability Analysis of the Mean Flows Obtained Using Unstructured Grids," *Journal of Aircraft*, Vol. 52, No. 1, 2015, pp. 49–63. doi:10.2514/1.C032583.

[59] Kaups, K., and Cebeci, T., "Compressible laminar boundary layers with suction on swept and tapered wings," *Journal of Aircraft*, Vol. 14, No. 7, 1977, pp. 661–667. doi:10.2514/3.44614.

[60] Cebeci, T., *Stability and Transition: Theory and Application. Efficient Numerical Methods with Computer Programs*, Heidelberg, Springer-Verlag Berlin Heidelberg, 2004., Berlin, 2004.

[61] Golbabai, A., and Javidi, M., "Newton-Like Iterative Methods for Solving System of Non-Linear Equations," *Applied Mathematics and Computation*, Vol. 192, No. 2, 2007, pp. 546–551. doi:10.1016/j.amc.2007.03.035.

[62] Falkneb, V. M., and Skan, S. W., "Solutions of the boundary-layer equations," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Vol. 12, No. 80, 1931, pp. 865–896. doi:10.1080/14786443109461870.

[63] Mayle, R. E., "The Role of Laminar-Turbulent Transition in Gas Turbine Engines," *Journal of Turbomachinery*, Vol. 113, No. 4, 1991, pp. 1–28. doi:10.1115/1.2929110.

[64] Dhawan, S., and Narasimha, R., "Some Properties of Boundary Layer Flow during the Transition from Laminar to Turbulent Motion," *Journal of Fluid Mechanics*, Vol. 3, No. 4, 1958, pp. 418–436. doi:10.1017/S0022112058000094.

[65] Krumbein, A., "Transitional Flow Modeling and Application to High-Lift Multi-Element Airfoil Configurations," *Journal of Aircraft*, Vol. 40, No. 4, 2003, pp. 786–794. doi:10.2514/2.3159.

[66] 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, 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.

[67] Somers, D. M., "Design and Experimental Results for a Flapped Natural-Laminar-Flow Airfoil for General Aviation Applications," Technical Report NASA-TP-1865, NASA Langley Research Center, Hampton, VA, United States, 1981.

[68] Mack, L. M., "Boundary-Layer Stability Theory. In: Special course on stability and transition laminar flow," *Special Course on Stability and Transition of Laminar Flow*, 1984. AGARD Report No. 709.

[69] Coder, J. G., "Development of a CFD-Compatible Transition Model Based on Linear Stability Theory," Ph.D. thesis, The Pennsylvania State University, State College, PA, 2014.

[70] Li, J., Bouhlel, M. A., and Martins, J. R. R. A., "Data-based Approach for Fast Airfoil Analysis and Optimization," *AIAA Journal*, Vol. 57, No. 2, 2019, pp. 581–596. doi:10.2514/1.J057129.

[71] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 33–62. doi:10.1023/B:OPTE.0000048536.47956.62.

[72] 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, 2014, pp. 935–951. doi:10.2514/1.J052255.

[73] Martins, J. R. R. A., and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2582–2599. doi:10.2514/1.J052184.

[74] Hwang, J. T., and Martins, J. R. R. A., "A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives," *ACM Transactions on Mathematical Software*, Vol. 44, No. 4, 2018, p. Article 37. doi:10.1145/3182393.

[75] Gray, J. S., and Martins, J. R. R. A., "Coupled Aeropropulsive Design Optimization of a Boundary-Layer Ingestion Propulsor," *The Aeronautical Journal*, Vol. 123, No. 1259, 2019, pp. 121–137. doi:10.1017/aer.2018.120.

[76] Hascoët, L., and Pascual, V., "TAPENADE 2.1 User's Guide," Technical report 300, INRIA, 2004. URL `https://hal.inria.fr/inria-00069880/document`.

[77] Kennedy, G. J., and Martins, J. R. R. A., "A parallel aerostructural optimization framework for aircraft design studies," *Structural and Multidisciplinary Optimization*, Vol. 50, No. 6, 2014, pp. 1079–1101. doi:10.1007/s00158-014-1108-9.

[78] Kenway, G. K., Kennedy, G. J., and Martins, J. R. R. A., "A CAD-Free Approach to High-Fidelity Aerostructural Optimization," *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, 2010. doi:10.2514/6.2010-9231.

[79] 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.

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

[81] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. doi:10.1137/S0036144504446096.

[82] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, 2012, pp. 101–118. doi:10.1007/s00158-011-0666-3.

[83] Rashad, R., "High-Fidelity Aerodynamic Design Optimization for Natural Laminar Flow," Ph.D. thesis, University of Toronto, Toronto, ON, Canada, 2016.