

Data-based Approach for Fast Airfoil Analysis and Optimization

Jichao Li Mohamed A. Bouhlel Joaquim R. R. A. Martins

Abstract Airfoils are of great importance in aerodynamic design, and various tools have been developed to evaluate and optimize their performance. Existing tools are usually either accurate or efficient, but not both. This paper presents a tool that can analyze airfoils in both subsonic and transonic regimes in about one hundredth of a second, and optimize airfoil shapes in a few seconds. We use camber and thickness mode shapes derived from over one thousand existing airfoils to parameterize the airfoil shape, which reduces the number of design variables. More than one hundred thousand RANS evaluations associated with different airfoils and flow conditions are used to train a surrogate model that combines gradient-enhanced kriging, partial least squares, and mixture of experts. These surrogate models provide fast aerodynamic analysis and gradient computation, which are coupled with a gradient-based optimizer to perform rapid airfoil shape design optimization. When comparing the surrogate-based optimization with optimization based on direct RANS evaluations, the largest differences in minimum C_d are 0.04 counts for subsonic cases and 2.5 counts for transonic cases. This approach opens the door for interactive airfoil analysis and design optimization using any modern computer or mobile device.

Nomenclature

\mathbf{a}	=	mode coefficients
\mathbf{A}	=	snapshot matrix of airfoils
C_d	=	drag coefficient
C_l	=	lift coefficient
C_m	=	pitching moment coefficient
d	=	dimension of surrogate model inputs
h	=	number of principal components of the partial least squares (PLS)
$k(.,.)$	=	kernel of surrogates
K	=	number of clusters or local experts
m	=	number of airfoils in the database
M	=	Mach number
Re	=	Reynolds number
$\mathbf{w}^{(l)}$	=	PLS coefficients
x	=	x coordinates of airfoils
y	=	y coordinates of airfoils
\hat{y}_k	=	output of local experts
\hat{y}	=	output of mixture of experts
α	=	angle of attack
ε_p	=	projection error in geometry representation of existing airfoils
π_k	=	mixing proportion of local experts
Φ	=	matrix of airfoil modes

Subscripts

c = airfoil camber line

ct = joint camber and thickness
 f = full airfoil
 t = airfoil thickness

1 Introduction

Airfoil design plays a fundamental role in aircraft design. Components, such as wings, horizontal and vertical stabilizers in fixed-wing aircraft, and rotor blades in helicopters, are designed using airfoil-shaped cross sections. A wide variety of airfoils have been developed for different applications. Airfoils used to be designed exclusively based on wind tunnel experiments, which are reliable but expensive and time-consuming. In the last several decades, airfoils are increasingly designed using computational aerodynamics, sometimes with the aid of numerical optimization to automatically find the shape.

There are several kinds of numerical models that can be used for airfoil aerodynamic analysis, including potential flow theory, Euler equations, and Reynolds-averaged Navier–Stokes (RANS) equations. Aerodynamic analysis using potential flow theory can be performed rapidly with panel methods, but these do not consider viscous or compressibility effects. To achieve higher accuracy, potential flow theory can be coupled with a boundary layer model to model the viscosity, as done in the XFOIL [1] software, which is widely used in the analysis of subsonic airfoils. However, higher fidelity models, such as RANS, are required in the transonic regime. Despite the rapid increase in computational power, aerodynamic analysis using these methods is still time-consuming, especially when many design iterations are required.

Airfoil design optimization requires a choice of shape parameterization to control the geometry. Generally, a parameterization method that requires few parameters and provides a high accuracy in the airfoil geometry is desirable. Masters et al. [2] compared seven different airfoil parameterization schemes by considering the geometric shape recovery of over 2000 airfoils. In their comparison, they included class-shape transformations [3], Hicks–Henne bump functions [4], singular-value decomposition (airfoil modes) [5, 6, 7, 8], B-splines, radial basis function domain elements, Bézier surfaces, and the parameterized sections method. They concluded that the airfoil modes parameterization was the most efficient approach. Robinson and Keane [9] derived airfoil modes from a family of supercritical airfoils and demonstrated that only a few of these modes are required to approximate the aerodynamic behavior of the original airfoils. Based on these studies, we adopt a parameterization approach based on the identification of airfoil modes shapes, since a smaller number of parameters are desirable in the construction of a more accurate surrogate model.

There are two broad categories of optimization methods: gradient-free [10, 11] and gradient-based [12, 13, 14] methods. Gradient-free methods usually need to evaluate thousands of different shapes when performing airfoil optimization, and thus the overall computational cost is prohibitive when using RANS. A common way to handle this issue is to use gradient-free methods together with surrogate models [15, 16, 17]. This type of method works well for low-dimensional cases, but surrogate models tend to scale poorly as the number of dimensions increases—the so called *curse of dimensionality*. Moreover, these surrogate models are generally designed for the optimization of a specific airfoil and trained in a perturbation subspace around an initial airfoil under a given objective and some constraints. They are not generic, and they are not suitable for the design of other airfoils. Therefore, a generic and accurate surrogate for fast aerodynamic analysis and optimization is of great interest. Other recent work explored the performance of a convolutional neural network method in the prediction of the lift coefficient of various airfoil shapes in diverse flow conditions [18]; however, this model was not tested with the drag and pitching moment coefficients.

On the other hand, gradient-based optimization methods require fewer iterations to converge to optimal airfoils. Using the adjoint method [12, 19, 20], the computational cost to computing gradients is dramatically reduced compared to the finite-difference or complex-step methods [21]. Usually, a gradient-based airfoil optimization with an efficient adjoint solver converges to the optimal solution using dozens to hundreds of aerodynamic analyses. While RANS-based airfoil analysis can be done in minutes using 16 processors in a high-performance parallel computer, and shape optimization can be performed within one or two hours, a much faster analysis would be desirable for a more interactive design process that can be done with much more modest hardware.

To address the need of fast interactive airfoil analysis and design optimization, we adopt a data-based approach that provides accurate results for airfoils in both subsonic and transonic flow regimes. The new contributions of this work relative to previous surrogate-based airfoil optimization work are: (1) the design space considered is much larger, (2) high accuracy is achieved for a much wider range in the design space. We perform a singular-value decomposition (SVD) modal extraction method to derive mode shapes from existing airfoils to control the shape of airfoils in each

flow regime. The modes for the subsonic regime are based on 1172 existing airfoils taken from the University of Illinois Urbana-Champaign (UIUC) airfoil database [22], while the modes for the transonic regime are based on the NASA SC(2) airfoils. Unlike approaches that use full-airfoil modes [2, 5, 6, 7, 8], we compute separate camber and thickness modes to parameterize the airfoils because they are more suitable for optimization problems with thickness constraints. The design space contains all the existing airfoils that are used for the decomposition, and we use a strategy to exclude abnormal airfoils that would never be desirable. Then, we create two databases for the subsonic and transonic regimes, respectively. These two databases contain more than one hundred thousand RANS samples with different shapes and flow conditions generated using an adaptive approach with an incremental Latin hypercube sampling (LHS) method to fill the subsonic and transonic design spaces.

We use these samples to train a recently developed surrogate model, which combines gradient-enhanced kriging, partial least squares (GE-KPLS), and mixture of experts [23]. The gradients for the RANS evaluations are computed using an adjoint method, which contributes towards the overall efficiency and accuracy of the surrogate model construction. Two global models composed of dozens of GE-KPLS models are constructed for the subsonic and transonic flow regimes. These surrogate models provide fast aerodynamic analysis and gradient computation. To perform rapid airfoil shape design optimization, these models are coupled with an efficient gradient-based optimizer. We demonstrate that accurate constrained airfoil design optimization can be performed under two seconds.

The rest of this paper is organized as follows: First, we introduce the derivation of airfoil mode shapes in Section 2. Then, we describe how we define the geometry design space in Section 3, and how the samples in our CFD databases are selected and evaluated in Section 4. We then introduce the data-driven approach in Section 5, including the surrogate model and the mixture of experts method. Finally, we verify this approach by comparing the surrogate model results to CFD analysis in Section 6 and to CFD-based optimization in Section 7. We end with the conclusions in Section 8.

2 Airfoil Shape Parameterization

We find that working in a design space defined by mode shapes as opposed to using control points works better when creating surrogate models. One of the reasons for this is that fewer parameters are required by the mode shape representation to meet a required geometry accuracy in the airfoil shape [2]. In addition, the information from the airfoil mode shapes of existing airfoils provides a way to exclude abnormal airfoils that would never be useful in practice, like for example, airfoils with excessive oscillations in their shape.

We use SVD on the airfoil databases to find the modes that parameterize the airfoil shape. To cover the largest possible number of airfoils, we first investigate airfoils in the UIUC airfoil database and obtain 1172 airfoils after some selection and smoothing operations. The detailed procedure for this smoothing and trailing edge treatment is given in the Appendix. We derive mode shapes from these UIUC airfoils to control airfoil shapes in the subsonic regime because most of them are designed for use in the subsonic range. Airfoils designed for the transonic regime are typically supercritical airfoils that have different characteristics from those designed for the subsonic regime, so we derive specific modes from 21 supercritical airfoils (NASA SC(2) series) to parameterize airfoils in the transonic regime.

2.1 Mode Shape Derivation

We derive airfoil mode shapes through SVD using an airfoil database of m airfoils that share the same x coordinates. These modes can be found by either looking at variations about a shape, or by considering the shape directly. We use the direct approach, which is more intuitive.

Assuming that each airfoil is represented by n surface points (x_i, y_i) , $i = 1, \dots, n$, the y coordinates for each of the m airfoils in the database are assembled in a snapshot matrix as

$$\mathbf{A} = \begin{bmatrix} y_{11} & y_{21} & \dots & y_{m1} \\ y_{12} & y_{22} & \dots & y_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1n} & y_{2n} & \dots & y_{mn} \end{bmatrix}. \quad (1)$$

Performing SVD on \mathbf{A} , we obtain

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

where columns in \mathbf{U} are full-airfoil modes, $\mathbf{\Sigma}$ is a $n \times m$ diagonal matrix, and \mathbf{V} is a matrix whose columns are the right-singular vectors. The diagonal entries in $\mathbf{\Sigma}$ are the singular values of \mathbf{A} in descending order, and thus the full-airfoil modes in \mathbf{U} are sorted by their importance. Not all the modes are required, so we use $\mathbf{\Phi}$ to represent the first n_f full-airfoil modes that we want to consider. For a new airfoil, the y coordinate values can be expressed by modes $\mathbf{\Phi}$ and their coefficients \mathbf{a}_{full} as

$$\mathbf{y} = \mathbf{\Phi} \mathbf{a}_{\text{full}}. \quad (2)$$

Fig. 1 shows the first 20 full-airfoil modes that are derived from the 1172 UIUC airfoils. Some modes, such as modes 3, 8, and 10, represent mostly camber, while others include both camber and thickness information. However, we cannot attribute any design meaning in general to these modes.

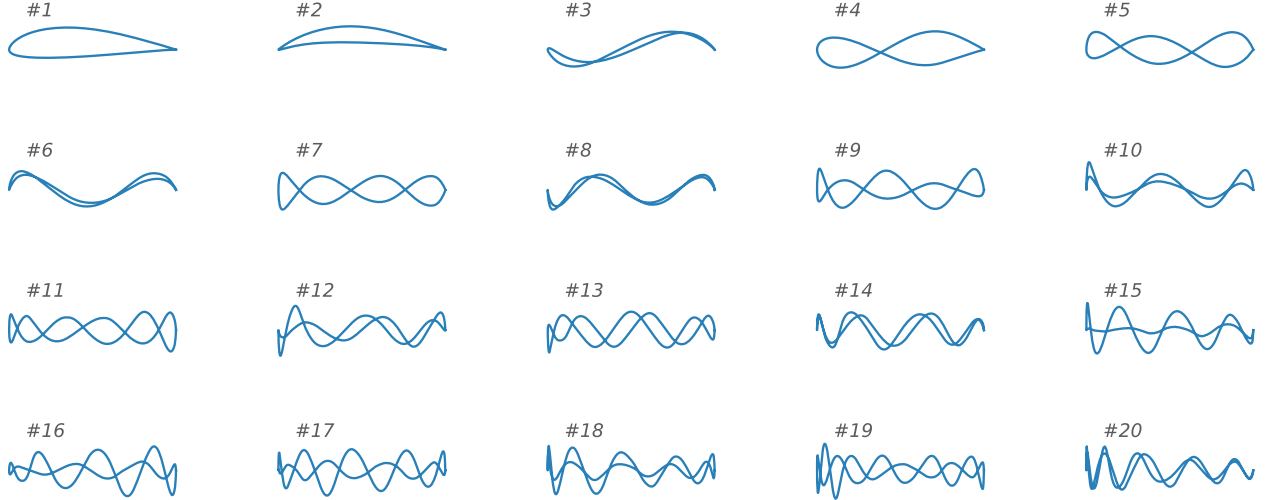


Fig. 1 First 20 full-airfoil mode shapes derived from 1172 UIUC airfoils.

An alternative approach to representing an airfoil is to use the y coordinates of the camber and thickness lines. Using the same airfoil database, the camber and thickness line of each airfoil can be represented by the coordinates of the camber and thickness lines, (x_i, y_{c_i}, y_{t_i}) , where $i = 1, \dots, n/2$. Similarly to the full-airfoil mode approach, the y coordinate values of the camber and thickness lines are assembled using snapshot matrices

$$\mathbf{A}_c = \begin{bmatrix} y_{c11} & y_{c21} & \cdots & y_{cm1} \\ y_{c12} & y_{c22} & \cdots & y_{cm2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{c1n_h} & y_{c2n_h} & \cdots & y_{cmn_h} \end{bmatrix}, \quad \mathbf{A}_t = \begin{bmatrix} y_{t11} & y_{t21} & \cdots & y_{tm1} \\ y_{t12} & y_{t22} & \cdots & y_{tm2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{t1n_h} & y_{t2n_h} & \cdots & y_{tmn_h} \end{bmatrix},$$

where $n_h = n/2$. Then, the modes of camber and thickness lines, \mathbf{U}_c and \mathbf{U}_t , are also be obtained by SVD:

$$\mathbf{A}_c = \mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c^T, \quad \mathbf{A}_t = \mathbf{U}_t \mathbf{\Sigma}_t \mathbf{V}_t^T.$$

We use $\mathbf{\Phi}_c$ and $\mathbf{\Phi}_t$ to denote the first n_c camber modes and the first n_t thickness modes, respectively. To find the airfoil coordinates from the modes and coefficients, we can take the camber line and then add and subtract the half thickness to obtain the coordinates of the upper and lower surfaces, respectively. We can write this as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_{\text{upper}} \\ \mathbf{y}_{\text{lower}} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_c & \mathbf{\Phi}_t \\ \mathbf{\Phi}_c & -\mathbf{\Phi}_t \end{bmatrix} \begin{bmatrix} \mathbf{a}_c \\ \frac{1}{2} \mathbf{a}_t \end{bmatrix} = \mathbf{\Phi}_{ct} \mathbf{a}_{ct}. \quad (3)$$

The first five camber and thickness modes derived from UIUC and SC(2) airfoils are shown in Fig. 2.

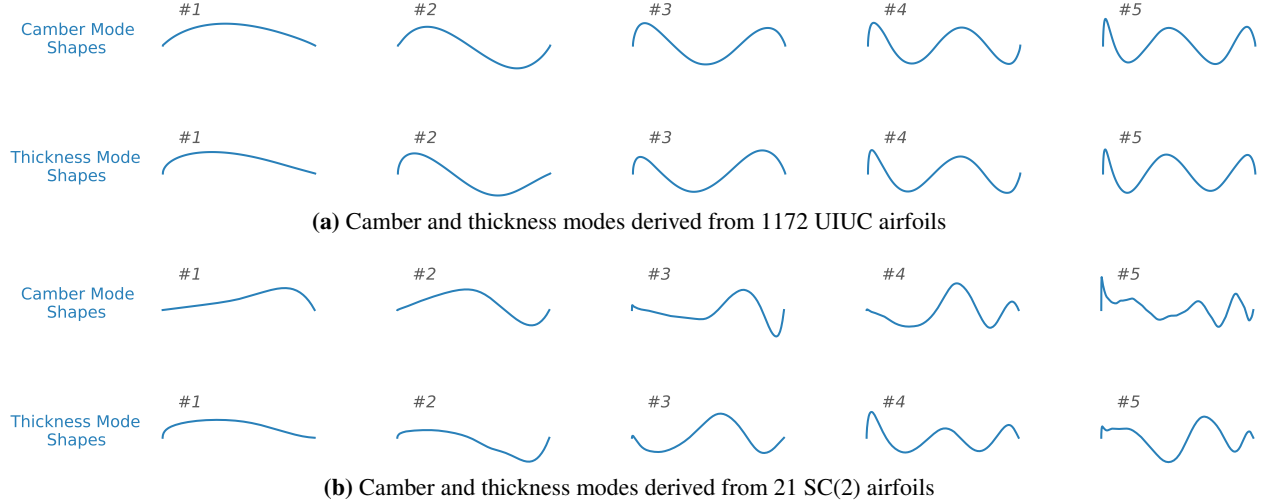


Fig. 2 Mode shapes from different airfoils are distinct.

2.2 Comparison of Full-airfoil Modes and Camber-thickness Modes

The design spaces defined by the two approaches are spanned by Φ (Eq. (2)) and Φ_{ct} (Eq. (3)), respectively. The rank of Φ is n_f , but the rank of Φ_{ct} is not necessarily equal to $n_c + n_t$. However, after many tests, we find that if $n_f = n_c + n_t$, then $\text{rank}(\Phi) = \text{rank}(\Phi_{ct})$. Therefore, the geometry space spanned by the camber-thickness modes is as large as that spanned by the full-airfoil modes.

Nevertheless, the ability of these modes to represent airfoils might be not the same. From the mathematical point of view, full-airfoil modes would be the most accurate in representing the snapshot airfoils. To study the difference in airfoil representation, we define the error ε_p to represent the projection error between the reconstruction airfoils and original airfoils after the preprocessing.

$$\varepsilon_p = \sum_{i=1}^m \frac{\|\mathbf{y}_{\text{orig}}^{(i)} - \mathbf{y}_{\text{mode}}^{(i)}\|_1}{m\|\mathbf{y}_{\text{orig}}^{(i)}\|_1}, \quad (4)$$

which we use to evaluate the ability of the parameterization to accurately represent the 1172 UIUC airfoils in the database. In the above equation, $\mathbf{y}_{\text{mode}}^{(i)}$ represents the airfoil coordinates given by the mode approach of Eqs. (2) and (3), and $\mathbf{y}_{\text{orig}}^{(i)}$ represents the original airfoil coordinates, which are used as the reference here. The error in Eq. (4) reduces as the number of modes is increased, as shown in Fig. 3.

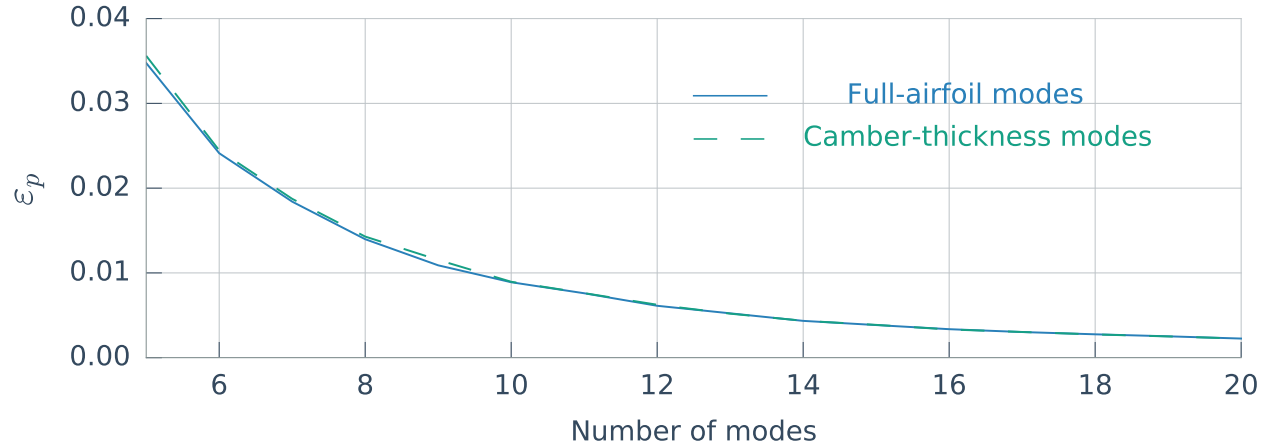


Fig. 3 Accuracy of full-airfoil modes and camber-thickness modes in representing 1172 UIUC airfoils.

For a certain number of modes, there are several possible combinations of n_c and n_t . We find that the highest

accuracy is achieved when

$$\begin{cases} n_c = n_t, & \text{if } n_c + n_t \text{ is even} \\ n_c = n_t - 1, & \text{otherwise} \end{cases} \quad (5)$$

In Fig. 3, the combination of Eq. (5) is used for each total number of modes used in the camber-thickness approach. We can find that both types of modes are equally accurate, especially for 10 modes or more.

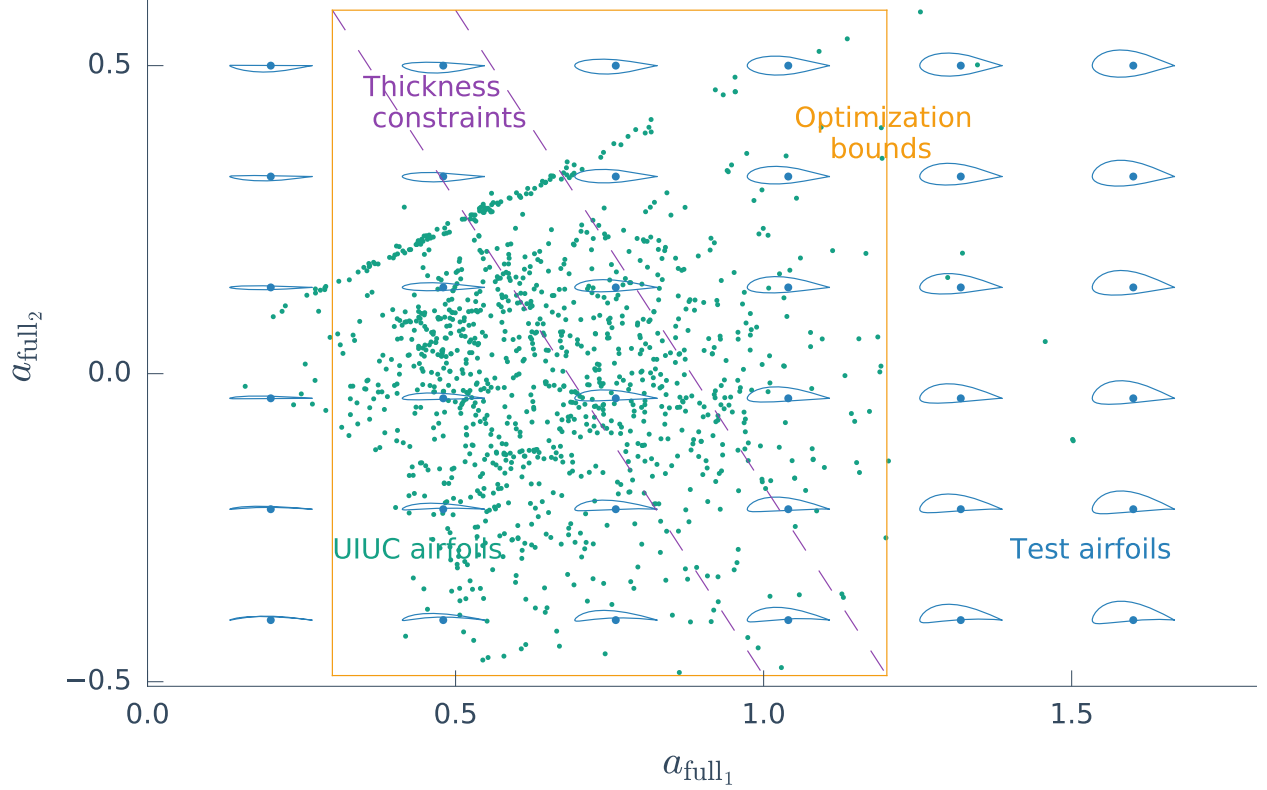


Fig. 4 Optimization bounds of full-airfoil modes satisfying given thickness constraints.

From the designer point of view, using camber and thickness separately is more intuitive and practical. For example, in airfoil optimization, it is common to enforce thickness constraints. In full-airfoil modes method, each mode contains both camber and thickness information, so there is no simple way to bound thicknesses directly. When using camber-thickness modes, there is a direct relationship to thickness and camber, and it is much easier to enforce bounds for a certain problem. As an example, consider a two-mode case shown in Figs. 4 and 5. The thickness increase is not aligned with the mode axes for the full-airfoil method (Fig. 4), and a practical design domain would have large bounds, as shown by the orange box. When using camber-thickness modes, however, we can clearly see in Fig. 5 a more direct relationship between the modes and the corresponding thickness and camber characteristics. This is both more meaningful and helpful because smaller design space bounds simplify the optimization and the surrogate model construction.

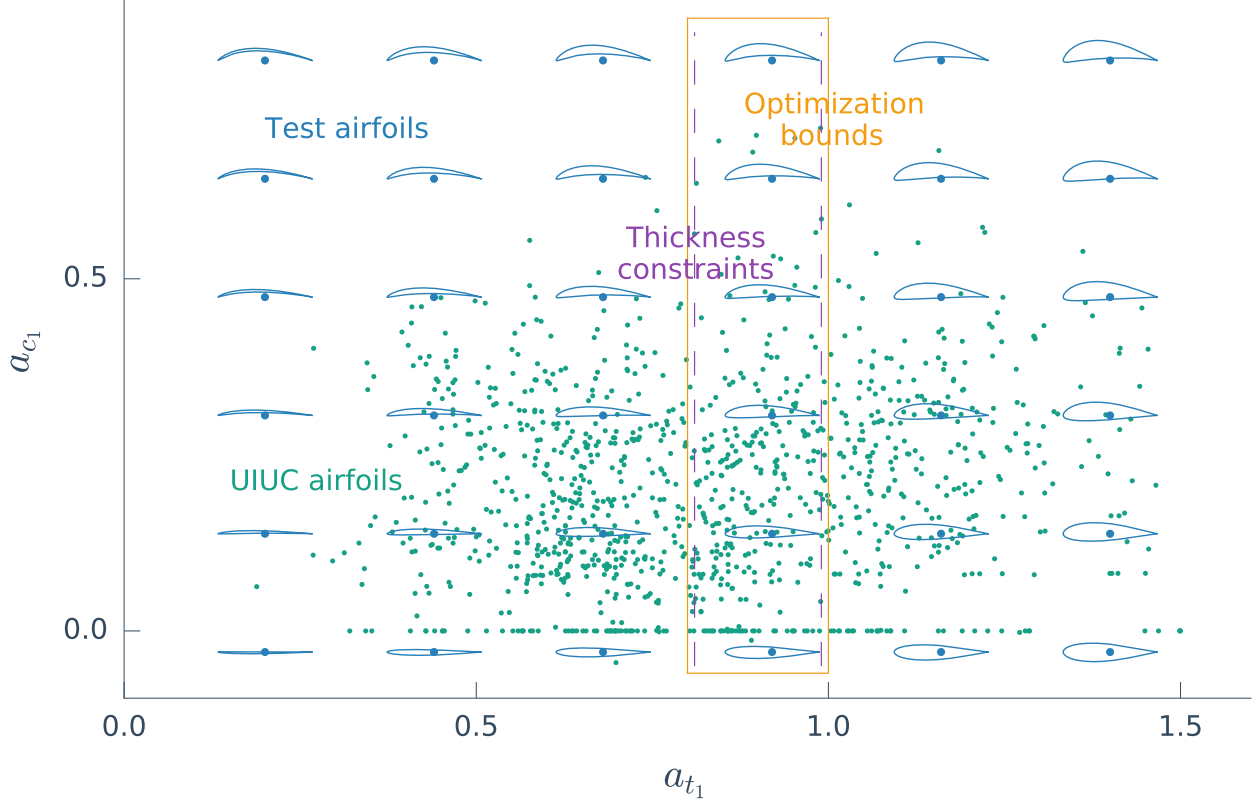


Fig. 5 Optimization bounds of camber-thickness modes satisfying given thickness constraints.

3 Design Space Definition

We now explain how to define the airfoil design space using the modes that we compute in the previous section. There are two main questions: how many modes to use, and what should be the mode amplitude bounds.

3.1 Number of Mode Shapes

There is always a trade-off when selecting the number of geometry parameters in aerodynamic surrogate models. On the one hand, increasing the number of geometry parameters decreases the projection error ε_p in representing airfoils, but on the other, when more geometry parameters are used, it becomes more difficult for surrogate models to represent functions accurately. In the transonic regime, due to nonlinear flow phenomena such as double shock waves (see Fig. 6), it is difficult for surrogate models to capture the aerodynamic profiles. The surrogate model is trained with 1000 LHS samples using the GE-KPLS method. Three camber and four thickness modes derived from UIUC airfoils are used in this test, and the flow conditions are fixed to $M = 0.73$, $\alpha = 1^\circ$, and $Re = 3 \times 10^6$. The error of the surrogate model becomes larger when predicting airfoils with increasingly strong double shock waves.

Airfoils in two design spaces are parameterized by modes derived from UIUC and SC(2) airfoils, and we refer to these modes as *subsonic modes* and *transonic modes*, respectively. We allow a geometry projection error of $\varepsilon_p = 0.5\%$ in both spaces, which guarantees an accurate geometry recovery. To satisfy this condition, seven camber modes with seven thickness modes are required in the subsonic case, and four camber modes with four thickness modes are required in the transonic case. We conduct RANS CFD simulations of 1172 UIUC airfoils in the database and corresponding airfoils parameterized by 14 subsonic modes. The flow condition in this test is $M = 0.5$, $\alpha = 0.0^\circ$, and $Re = 3.0 \times 10^6$. The average error in C_d is only 0.53 counts, showing that the specified projection error ($\varepsilon_p = 0.5\%$) ensures accurate results.

Including some “low-speed” information by adding subsonic airfoils in the derivation of transonic modes can make the optimization of transonic airfoils more robust. However, more transonic modes are needed to meet the requirement on the projection error, and it is more costly to train an accurate model in the transonic regime. The choice of using transonic modes derived from SC(2) airfoils is a compromise between these concerns.

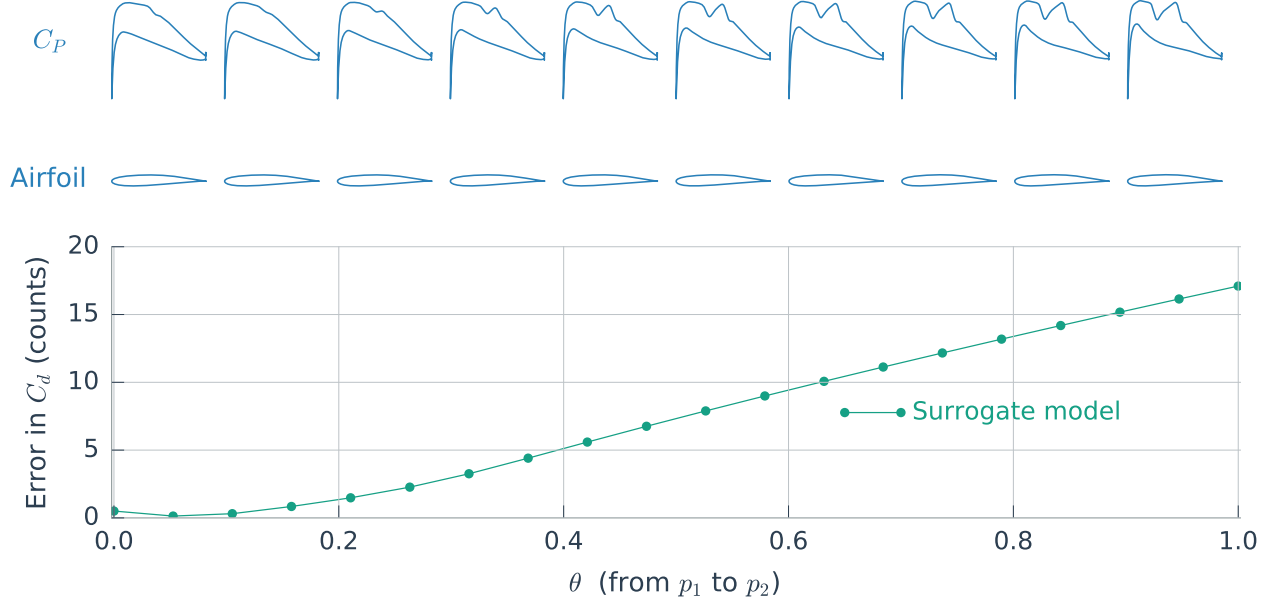


Fig. 6 The performance of a surrogate model in transonic regime along a line between two airfoils, where the intermediate airfoils are linearly interpolated.

As defined, the modes derived previously result in airfoils with sharp trailing edges. However, we want to represent airfoils with blunt trailing edges because all airfoils have such trailing edges in practice due to manufacturing constraints. One could use another variable to control the thickness of the trailing edge, but this would involve one more design variable that would make it more difficult to construct the surrogate models. Instead, we use a correlation between the amplitude of the first thickness mode and the trailing edge thickness. After investigating the SC(2) airfoils, we find that there is a relationship between the thickness of the trailing edge and the coefficient of the first thickness mode, from which we can determine the trailing edge thickness. In Fig. 7, we show how a linear regression equation captures this relationship, where we determine the regression coefficient to be 0.00647, which we approximate to 0.006. The UIUC airfoil database contains different types of airfoils, including the sharp trailing edge ones, and it might not make sense to create a separate regression using UIUC airfoils for the subsonic regime. We use this equation to determine the trailing edge thickness for all airfoils in both regimes. This approach results in airfoils with reasonable trailing edge thickness values without introducing an additional design variable.

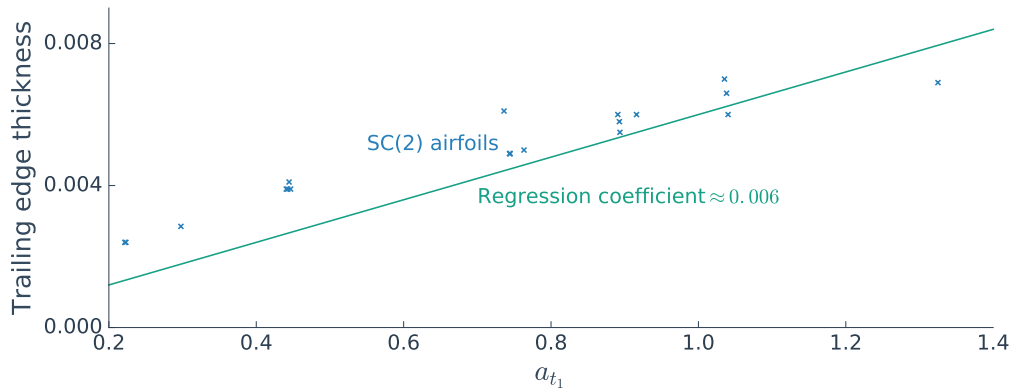


Fig. 7 The relationship between the first thickness mode and trailing edge thickness

3.2 Mode Shape Bounds

We aim to capture as many existing airfoils as possible, but this does not mean that we have to consider all possible geometries. If the bounds of airfoil modes are too large, this could result in airfoil shapes that are undesirable. This

would be unproductive because we would be sacrificing the surrogate model accuracy unnecessarily. In this paper, we adopt a bound definition that considers the relationship between dominant modes (first camber and thickness modes) and higher-order modes. The design space defined by these bounds includes all existing airfoils in the database and leaves out undesirable airfoils.

One could roughly define the bounds of each camber and thickness mode using

$$\begin{cases} a_{c_{\text{lower}}}^{(j)} = \min \mathbf{a}_c^{(j)} \\ a_{c_{\text{upper}}}^{(j)} = \max \mathbf{a}_c^{(j)} \end{cases}, \quad \begin{cases} a_{t_{\text{lower}}}^{(j)} = \min \mathbf{a}_t^{(j)} \\ a_{t_{\text{upper}}}^{(j)} = \max \mathbf{a}_t^{(j)} \end{cases}, \quad (6)$$

where $\mathbf{a}_c^{(j)} \in \mathbb{R}^m$, $j = 1, \dots, n_c$, and $\mathbf{a}_t^{(j)} \in \mathbb{R}^m$, $j = 1, \dots, n_t$ represent the mode coefficients of the m existing airfoils in the database.

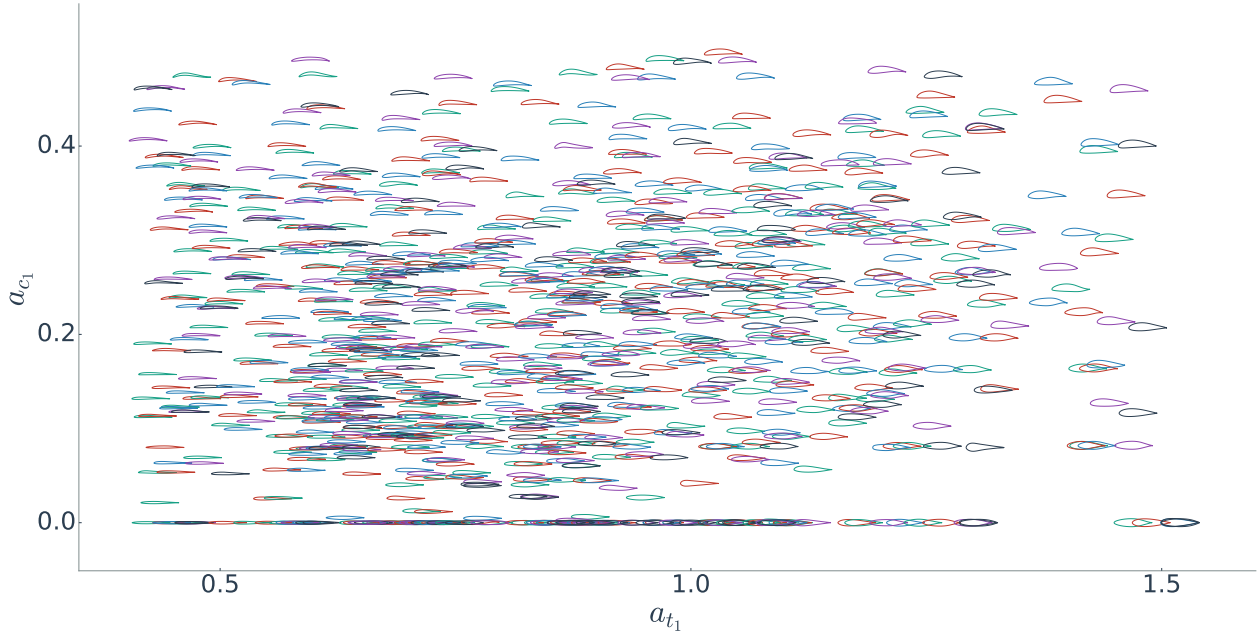


Fig. 8 Distribution of UIUC airfoils with respect to the first thickness and camber modes (a_{t_1} and a_{c_1}).

The bounds determined by Eq. (6) are larger than necessary and the feasible space still includes airfoils that have unrealistic shapes. To eliminate these unrealistic shapes, we start by looking at the first thickness and camber modes. For the UIUC database airfoils, two airfoils have similar shapes if they are close in the a_{t_1} – a_{c_1} plane (shown in Fig. 8). Thus, we can interpolate a 2D function $S(a_{t_1}, a_{c_1})$ with respect to the first thickness and first camber modes to capture the distributions of higher-order modes in existing airfoils, and then add and subtract a margin to this function to define the upper and lower bounds. However, we find that this results in bound functions that are multimodal, which causes difficulties when performing optimization. Therefore, we develop a smoothing procedure to eliminate this multimodality.

For the first camber and thickness modes, we obtain the bounds from Eq. (6), i.e., the lower and upper bounds are the minimum or maximum values of existing airfoils, respectively. For the other modes, the bounds are determined by four steps:

1. Interpolate a surface function $S(a_{t_1}, a_{c_1})$ using the inverse distance weighting method with the data of existing airfoils, generate a regular grid of points in the a_{t_1} – a_{c_1} plane, and compute the surface function value S_{ij} for those grid points.
2. Generate the upper and lower bounds coordinates S_{ij}^{upper} and S_{ij}^{lower} based on S_{ij} , where $S_{ij}^{\text{upper}} = \max(0, S_{ij})$ and $S_{ij}^{\text{lower}} = \min(0, S_{ij})$.

3. Smooth the coordinates using Laplacian smoothing. After each Laplacian smoothing iteration, check each upper bound to see if $S_{ij}^{\text{upper}} \in [\max(0, S_{ij}), \max(0, S_{ij}) + \Delta S]$, and set it to the interval upper bound if it is outside the interval. For the lower bound, do the same measure to ensure $S_{ij}^{\text{lower}} \in [\min(0, S_{ij}), \min(0, S_{ij}) - \Delta S]$. The margin ΔS is set to 10% of the variance range of the corresponding mode in the existing airfoils database; i.e., $\Delta S = 0.1 (\max(S_{ij}) - \min(S_{ij}))$.
4. Interpolate the upper and lower bound surfaces with cubic interpolation method. The inputs are the grid points, and the outputs are S_{ij}^{upper} and S_{ij}^{lower} for the upper and lower bounds, respectively.

This procedure reduces the bounds of the higher-order modes and retains their relationship to the dominant modes. To show this effect, we select two points in the a_{t_1} - a_{c_1} plane, and plot the bounds of the second camber mode (a_{c_2}) along the straight line between these two points (see Fig. 9). These bounds are determined by Eq. (6) and the smoothing procedure above, respectively. Compared with the bounds defined by Eq. (6), the bounds defined by the smoothing procedure are more restrictive. These interpolation bounds exclude undesirable airfoils with abnormal shapes and include all existing airfoils in the database.

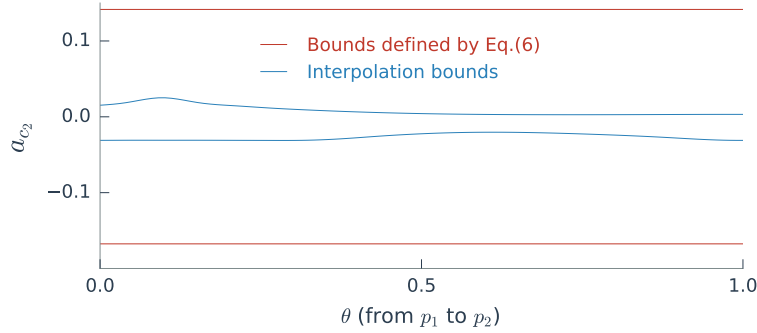


Fig. 9 The bounds produced with the interpolation approach are much tighter than the other approach.

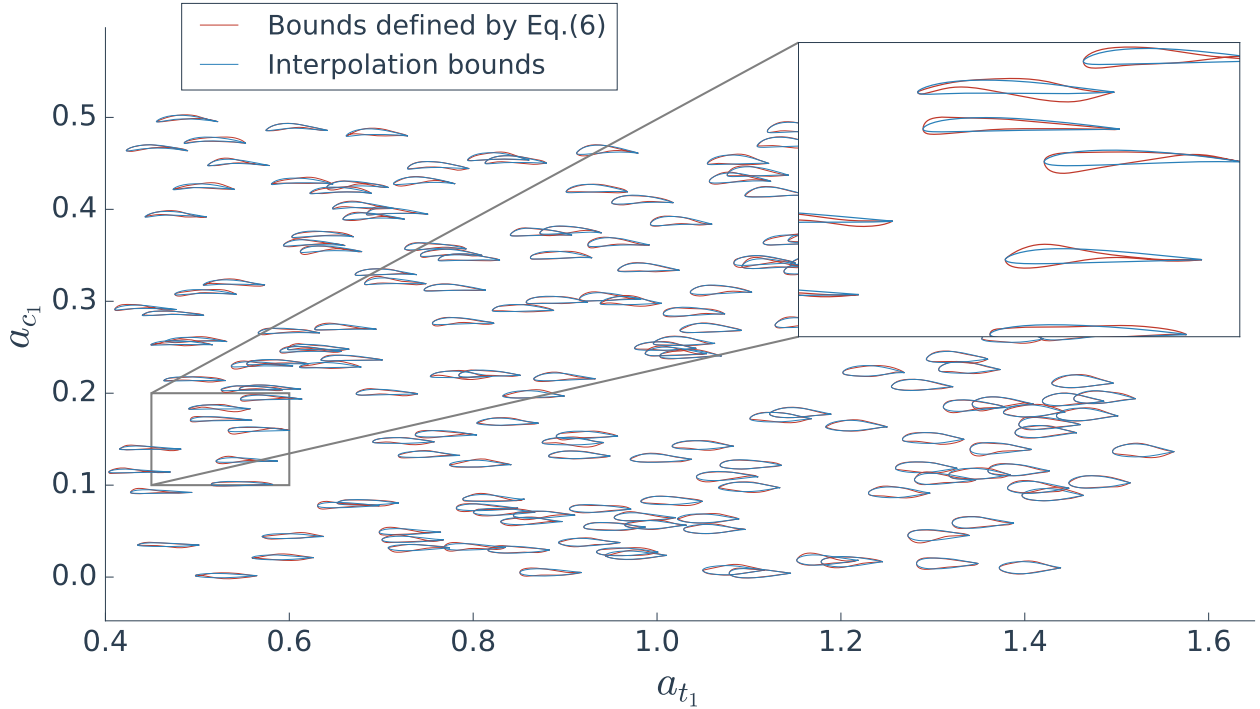


Fig. 10 The interpolation approach filters out undesirable airfoils (red) while producing realistic shapes (blue).

In Fig. 10, we compare the samples generated with different bounds for the modes. This test is conducted with the 14 subsonic modes in the subsonic regime. We first generate 200 normalized points in 14-dimensional space with LHS. Based on the bounds defined in each approach, these sampling points are transferred to corresponding ranges. Then we use the transferred sampling coefficients to generate airfoils and plot them in Fig. 10 based on their values of first camber and thickness modes. We can see that there are some airfoils with strange shapes where the bounds are defined by Eq. (6), while the sampled airfoils are much more reasonable when using our smooth bound strategy.

4 Airfoil Database Sampling

To construct accurate surrogate models, we need to sample a large number of airfoils that adequately cover the design space and evaluate them using CFD. We construct two aerodynamic databases: one for the subsonic regime, and another for the transonic regime. In the subsonic database, the airfoils are controlled by 14 subsonic modes (7 camber and 7 thickness), $\alpha \in [-2.0^\circ, 6.0^\circ]$, and $M \in [0.3, 0.6]$. The flow conditions are set assuming standard atmosphere at an altitude of 10,000 m, so based on the variation of M , Re changes in the range of $[2.5 \times 10^6, 5.1 \times 10^6]$. For the transonic regime, we use 8 transonic modes (4 camber and 4 thickness) to control the airfoil shapes, and $\alpha \in [-1.5^\circ, 4.5^\circ]$, $M \in [0.65, 0.85]$, and $Re \in [5.5 \times 10^6, 7.2 \times 10^6]$. The bounds of α are determined by considering the typical range in the cruise angle of attack of modern aircraft and adding an ample margin.

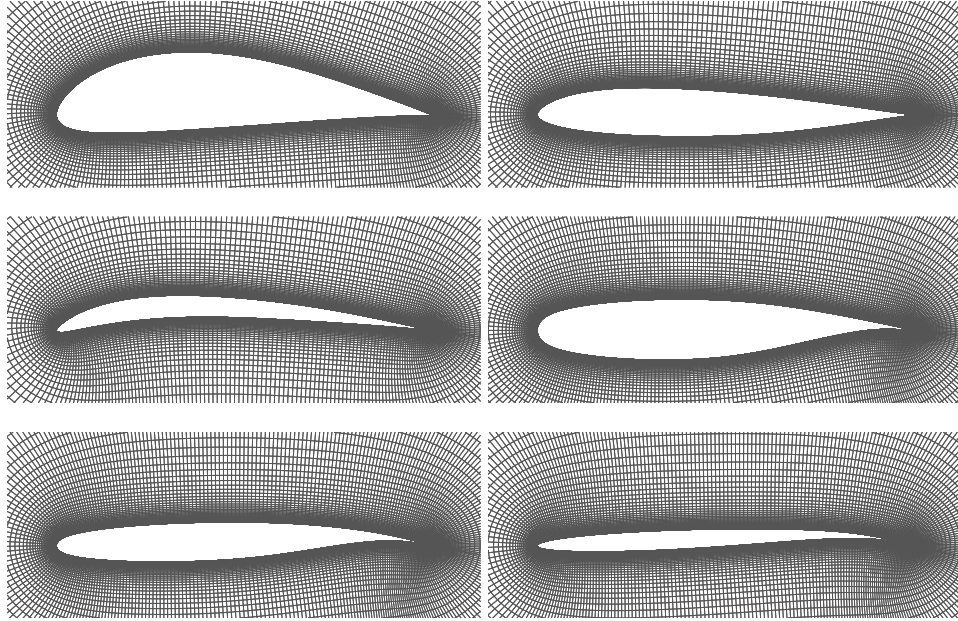


Fig. 11 CFD meshes of some airfoils generated by pyHyp.

Before we sample, we divide the design space into sub-domains evenly with respect to M , α , the first camber mode, and the first thickness mode. We use the LHS method to generate the same number of sampling airfoils in each sub-domain. Since the bounds of higher-order modes are not constant with respect to changes in the two dominant modes, we first generate a normalized coefficient between zero and one for each of these modes, and then transfer it to the range between the lower and upper bounds at the point. In addition, we generate the same number of validating airfoils in each sub-domain using the incremental LHS algorithm based on the existing LHS samples. After running CFD simulations for these sampling and validating airfoils, validation tests are conducted in each sub-domain. If the error for these validating airfoils is not acceptable, we add the validating airfoils to the set of sampling airfoils, then we generate other validating airfoils and run the validation test again.

Table 1: Description of two aerodynamic databases

	Number of variables		Number of samples
	Shape	Flow	
Subsonic database	14 (7 camber and 7 thickness)	2 (M and α)	81,000
Transonic database	8 (4 camber and 4 thickness)	2 (M and α)	32,400

The CFD solver we use is ADflow [24, 25], which solves RANS with a Spalart–Allmaras turbulence model [26]. We use an in-house code (pyHyp) to automatically generate RANS CFD meshes for all airfoils. Fig. 11 shows the meshes that pyHyp generated for some sampling airfoils. The flow solution residuals are converged 15 orders of magnitude lower than the freestream residuals. Since both pyHyp and ADflow can be loaded and used as Python modules, we use Python scripts to generate CFD meshes, run CFD simulations, determine convergence, and collect results automatically. The coefficients of lift C_l , drag C_d , and pitching moment C_m (at the quarter chord point) are obtained after the CFD simulation. The ADflow adjoint solver is also called to compute the gradient with respect to all modes, Mach number, and angle of attack, since the gradient information is required for the construction of the gradient-enhanced surrogate model. More details about the surrogate model approach are given in Section 5. We create two databases: one for the subsonic regime, and the other one for the transonic regime. Table 1 lists the details for the two databases. There are only 0.45% subsonic samples and 2.24% transonic samples whose CFD simulations do not converge. This high success rate is due to the new bounds that exclude most of the undesirable airfoil shapes and is also due to the robust approximate Newton–Krylov implemented in ADflow.

In both design spaces, the CFD library contains about 100,000 samples, while a typical gradient-based airfoil optimization requires only 50 to 100 CFD and adjoint evaluations. Therefore, the computational cost of building the library in this data-based approach is about 1000 to 2000 times that of a typical high-fidelity airfoil optimization.

5 Surrogate Model

5.1 GE-KPLS—Gradient-Enhanced kriging with Partial-Least Squares

Surrogate models, also known as response surfaces, provide a low computational cost alternative to time-consuming aerodynamic models. One of the most popular surrogate models is the kriging model [27, 28, 29, 30]. Kriging has been widely used in airfoil design [6, 10, 31]. Kriging can be extended to utilize gradient information, called gradient-enhanced kriging (GEK), which improves the accuracy of the model for a given number of evaluations. However, GEK is not computationally efficient when the number of inputs, the number of sampling points, or both, are high. This is mainly due to the size of the corresponding correlation matrix, which increases proportionally with both the number of inputs and the number of sampling points. In addition, the correlation matrix becomes ill-conditioned when the sampling points are close to each other. Another difficulty is estimating the hyper-parameters needed in the kriging approach, which involves solving a multimodal optimization problem—maximizing the likelihood function—whose number of variables is proportional to the problem dimension.

To address these issues, we use the gradient-enhanced kriging with partial least squares (GE-KPLS) approach [23]. This method uses the gradient information with a modest increase in the size of the correlation matrix and reduces the number of hyper-parameters. The key idea of GE-KPLS is to generate a set of approximating points around each sampling point using the first order Taylor approximation method. We then apply the PLS method several times, each time on a different number of sampling points with the associated approximating points. Each PLS provides a set of coefficients representing the contribution of each variable near the associated sampling point to the output. Next, we compute the average of all PLS coefficients with respect to each variable to get the global influence to the output. Denoting these coefficients by $\mathbf{w}^{(l)} = [w_1^{(l)}, \dots, w_d^{(l)}]$ for $l = 1, \dots, h$, where d is the dimension of the problem and h is the number of principal components of the PLS, and using the mathematical property that the tensor product of several kernels is a kernel, we finally build the GE-KPLS kernel

$$k_{1:h}(\mathbf{x}, \mathbf{x}') = \prod_{l=1}^h k_l(F_l(\mathbf{x}), F_l(\mathbf{x}')), \quad \forall \mathbf{x}, \mathbf{x}' \in B, \quad (7)$$

where B is a hypercube expressed by the product between the intervals of each direction space, $k_l : B \times B \rightarrow \mathbb{R}$ is an

isotropic stationary kernel, and F_l is a linear map given by

$$\begin{aligned} F_l : B &\longrightarrow B \\ \mathbf{x} &\longmapsto [w_1^{(l)} x_1, \dots, w_d^{(l)} x_d]. \end{aligned} \quad (8)$$

In this paper, we use the Gaussian kernel given by

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{l=1}^h \prod_{i=1}^d \exp \left[-\theta_l (w_i^{(l)} x_i - w_i^{(l)} x'_i)^2 \right], \quad \forall \theta_l \in [0, +\infty], \quad \forall \mathbf{x}, \mathbf{x}' \in B. \quad (9)$$

The GE-KPLS method reduces the number of hyper-parameters to be estimated from d to h , where $h \ll d$, thus drastically decreasing the time to construct the model.

We use a 16-dimensional subsonic problem (14 mode shapes, M , and α) to demonstrate the advantage of GE-KPLS over kriging and GE-kriging methods. To reduce the influence of the particular sampling on the results, we test each model 20 times with different LHS sampling airfoils. Each sample used for the construction of a kriging model requires one CFD evaluation and one adjoint gradient evaluation in the construction of a gradient-enhanced model (GE-kriging and GE-KPLS). The adjoint solution has at most the same cost as a CFD evaluation. To make a fair comparison among all of the surrogate models, we multiply the number of training samples for kriging models by two to account for the adjoint solution cost in the gradient-enhanced models, as shown in Fig. 12. The results shown in the figure are the average of 20 tests in predicting 1000 validating airfoils and demonstrate the advantage of GE-KPLS in accuracy and computation cost over kriging and GE-kriging, respectively.

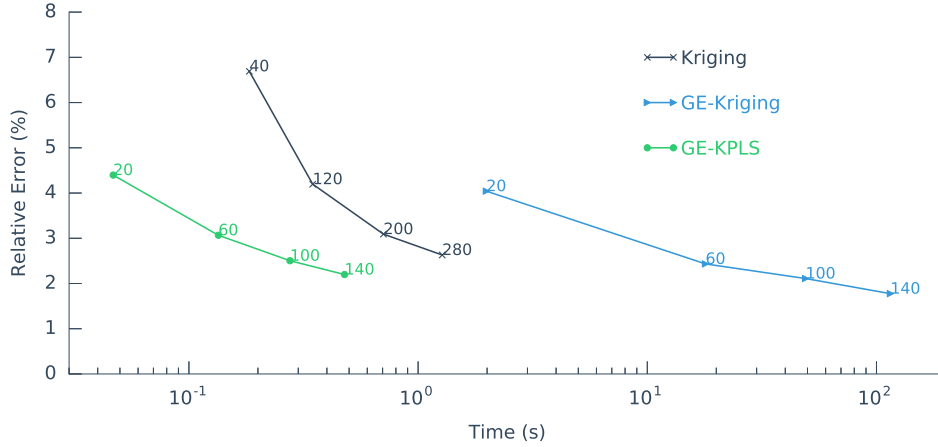


Fig. 12 Comparison of three surrogate models in a subsonic case with 16 parameters, where the label for each point is the cost of the training in terms of CFD evaluations. $M \in [0.4, 0.5]$ and $\alpha \in [1.0^\circ, 2.0^\circ]$.

5.2 GE-KPLS Coupled with Mixture of Experts

GE-KPLS has an obvious advantage in computational cost when the number of sampling points is large. However, it is still difficult for one single GE-KPLS model to handle tens of thousands of sampling points because the kriging correlation matrix becomes ill-conditioned. Therefore, we adopt a mixture of experts strategy.

Mixture of experts (ME) [32] is a method for combining different surrogate models, which is particularly useful for highly nonlinear functions that span a large independent variable space. ME is based on a “divide-and-conquer” approach in which the problem space is divided between a few “experts” (surrogate models). The original ME model was introduced by Jacobs et al. [33], and a cluster-based ME model was proposed by Tang et al. [34]. Typically, the general ME formulation can be expressed as,

$$\begin{aligned} \hat{y}(\mathbf{x}_0) &= \sum_{k=1}^K \pi_k(\mathbf{x}_0) \hat{y}_k(\mathbf{x}_0) \\ \text{with } \pi_k &\in [0.0, 1.0] \text{ and } \sum_{k=1}^K \pi_k = 1.0, \end{aligned} \quad (10)$$

where K is the total number of local experts, \hat{y}_k ($k = 1, \dots, K$) is the k^{th} local surrogate model, and $\pi_k(\mathbf{x}_0)$ is the mixing proportion, which depends on the location of \mathbf{x}_0 in the design space. The output of ME is a weighted combination of K local models.

The construction of a cluster-based ME consists of three steps:

1. Divide the samples into K clusters using an unsupervised clustering algorithm, and train local surrogate models with the samples in each cluster.
2. Use a supervised learning algorithm to obtain cluster posterior probability, which is used to evaluate the mixing proportions.
3. Provide the weighted combination output using Eq. (10).

We adopt the k -means clustering algorithm [35] to do the unsupervised clustering of the samples. To make samples in the same cluster have similar shapes and aerodynamic features, we use the two dominant modes and one aerodynamic function (C_l , C_d , or C_m) as the input of the clustering algorithm. For each aerodynamic function, a three-dimensional k -means clustering is involved in dividing the samples, and different clustering results are generated for different aerodynamic functions to predict. The aerodynamic coefficients are normalized before the clustering process to make them have a similar range with the two modes.

The mixing proportion is typically derived based on the cluster posterior probability, i.e., the probability that the k^{th} cluster is active at \mathbf{x}_0 . We use an approach similar to that of Liem et al. [36], where the regularized Gaussian classifier is the supervised algorithm that provides the cluster posterior probability. The regularized Gaussian classifier is trained by the samples labeled by the clustering results. It only takes the first two modes to train the classification and to predict unknown points. We use the modified “softmax” function to obtain the mixing proportion [36]. The mixing proportion of the k^{th} cluster at point \mathbf{x}_0 is expressed as,

$$\pi_k(\mathbf{x}_0) = \frac{\exp(\omega a_k(\mathbf{x}_0))}{\sum_{j=1}^K \exp(\omega a_j(\mathbf{x}_0))}, \quad (11)$$

where $\omega = 5$ and

$$a_j(\mathbf{x}_0) = \ln \left[p(\mathbf{x}_0 | z_j = 1) p(z_j = 1) \right]. \quad (12)$$

Here, $z_j(\mathbf{x}_0) \in [0, 1.0]$ is an unobservable latent indicator that assigns input points to local clusters, and $\sum_{j=1}^K z_j = 1$.

Fig. 13 shows the mixing proportion of six clusters in the subsonic design space. The clusters are trained by 5000 samples with $M \in [0.4, 0.5]$ and $\alpha \in [0.0^\circ, 2.0^\circ]$. Due to the clustering strategy described previously, different aerodynamic coefficients are associated to different clusters. This figure shows the clustering results for C_l and C_d .

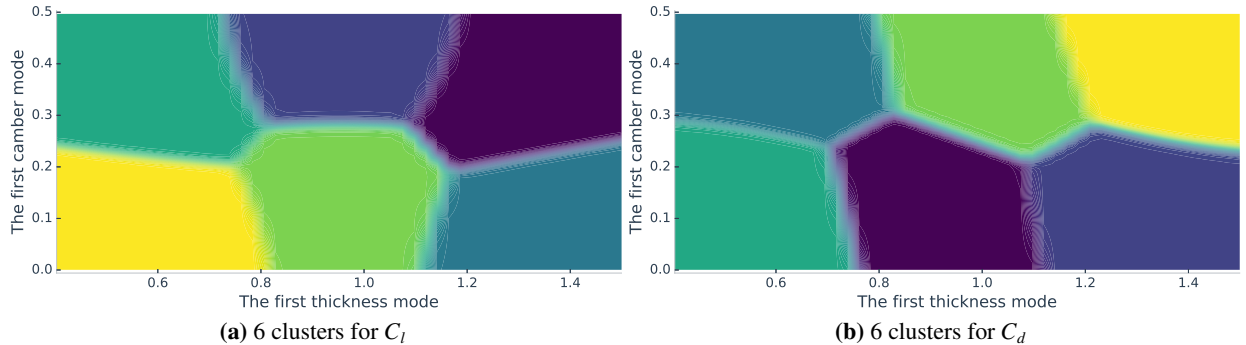


Fig. 13 Mixing proportion π_k plots. Different colors refer to different experts.

The flow condition parameters, M and α , have a distinct influence on the aerodynamic coefficients. For example, two samples with large differences in M or α should not be in the same cluster even if they are close in airfoil shape. In practice, we first split the flow condition domain to divide it into similarly sized regions, and then use k -means clustering in each region. The number of flow condition regions in the subsonic and transonic regimes are 9 and 12,

respectively. The best number of clusters in each part is determined based on the performance of the ME model in predicting validation airfoils in the part. The total number of clusters used in the subsonic and transonic regimes is 108 and 72, respectively.

6 Aerodynamic Analysis

In this section, the surrogate model is validated by performing CFD analyses. To demonstrate the robustness, two different methods are used:

1. Analysis of random airfoils. These airfoils are generated by an incremental LHS sampling method to ensure they are not included in the training database. This test gives us a statistical insight of the surrogate model's performance.
2. Aerodynamic coefficient curves of existing airfoils. The surrogate model is used to plot C_l , C_d , and C_m curves versus α for some existing airfoils. This test aims to show the performance of the surrogate model in the subspace of α .

6.1 Aerodynamic Analysis of Randomly Generated Airfoils

We generate 2745 and 1017 airfoils using an incremental LHS method in the 14-dimensional subsonic airfoil design space and 8-dimensional transonic airfoil design space, respectively. For the subsonic airfoils, the flow conditions are $M = 0.45$ and $\alpha = 2.5^\circ$, while $M = 0.73$ and $\alpha = 1.0^\circ$ for the transonic airfoils. These airfoils are first evaluated by ADflow. Some of these CFD simulations do not converge, since they are randomly generated and might not be suitable for the given Mach number and angle of attack combination. The numbers of subsonic and transonic validating airfoils that converge in ADflow are 2741 and 989, respectively. The converged airfoils are evaluated by the corresponding surrogate model, and the relative error of the surrogate model is computed using

$$\text{Relative Error} = \frac{\|C_{\text{CFD}} - C_{\text{SM}}\|_2}{\|C_{\text{CFD}}\|_2} \times 100\%, \quad (13)$$

where C_{CFD} and C_{SM} are the coefficient predictions of the validating airfoils by ADflow and the surrogate model, respectively.

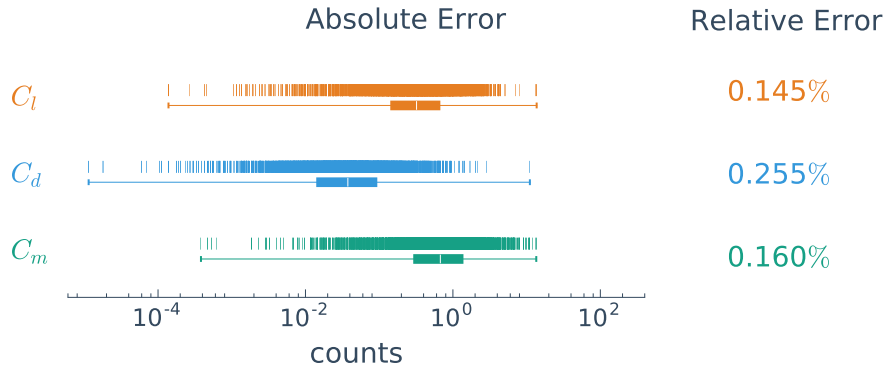


Fig. 14 Validation of the surrogate model for 2741 randomly generated airfoils in the subsonic design space.

Fig. 14 shows the error of the subsonic surrogate model for 2741 subsonic airfoils. On the left side of the figure, the absolute error in counts is shown by a scatter plot and a corresponding box plot. For C_d and C_m , 1 count represents 10^{-4} , while for C_l , 1 count represents 10^{-3} , based on industrial common practice. The scatter plot at the top provides the absolute error for each validating airfoil, and the box plot in the bottom provides the statistics for these absolute errors. The white line in the box plot indicates the median of these errors. For C_l , about 86.6% of the airfoils are predicted with an error of less than 1 count. For C_d and C_m , the percentages of validating airfoils with an error less than 1 count is 99.6% and 63.8%, respectively. There are a few airfoils with larger errors. For example, one airfoil has a C_d error of about 10 counts. Six airfoils have an error in C_l that is greater than 5 counts. Since the absolute values of C_l , C_d , and C_m are different, it is incomplete to use absolute error to compare the performance of the surrogate model.

On the right side of the figure, we provide the total relative error computed by Eq. (13) for the three coefficients, which are all less than 0.3%. We also find that it is more difficult to predict C_d than C_l and C_m in the subsonic regime.

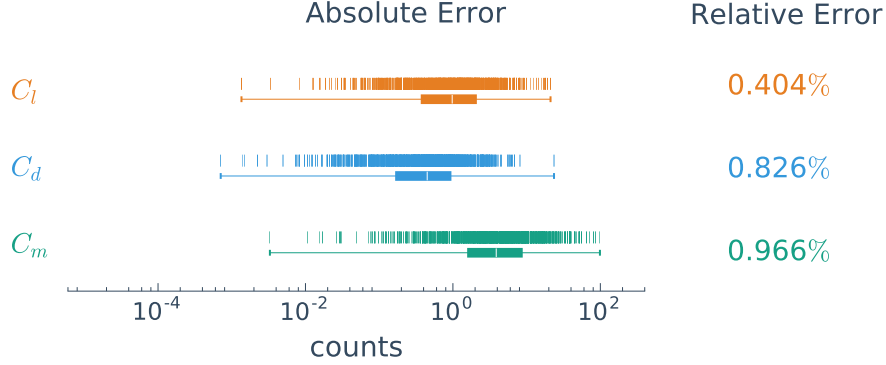


Fig. 15 Validation of the surrogate model for 989 randomly generated airfoils in the transonic design space.

Fig. 15 shows the validation results for 989 supercritical airfoils in the transonic regime. The errors here are larger when compared with those in the subsonic case. This is due to the stronger nonlinearity in the transonic regime. For C_d , the largest error is about 23 counts, and 77% of the airfoils have an error lower than one count. For C_l , there are 14 airfoils with an error that is over 10 counts, and more than 50% of airfoils have an error under one count. For C_m , the percentage of airfoils with an error under 10 counts is 80.3%. The total relative error (13) of C_l , C_d , and C_m increased by a factor of 1.78, 2.23, and 5.03, respectively, compared to the subsonic regime, and the average absolute error among the validating airfoils increases by a factor of 3.2, 8.9, and 6.7. Thus, it can be seen that surrogate models for C_d and C_m are more susceptible to the challenges inherent in the transonic regime. Nevertheless, the performance of the transonic surrogate models remains acceptable.

6.2 Aerodynamic Coefficients Versus Angle of Attack

We use our subsonic surrogate model to validate the variation of the aerodynamic coefficients of three well-known airfoils (NACA4412, ClarkY, and NLF1015) with respect to α at $M = 0.45$. For the transonic regime, we select three supercritical airfoils: SC(2)0404, SC(2)0606, and SC(2)0710, at $M = 0.73$.

In Figs. 16 and 17, we find that the surrogate model is accurate within the given range of α for both subsonic and transonic regimes. The average errors for the subsonic airfoils over this range of α in C_l , C_d , and C_m are 3.4, 0.3, and 9.9 counts, respectively. The corresponding average errors for the supercritical airfoils are 4.0, 4.5, and 14.9 counts, respectively. The largest error in predicting C_d (2.3 counts) in the subsonic regime occurs for the NLF1015 when α is around 3.5° . This is due to an issue with the combination of different local experts in ME. For the subsonic regime, we divide the range of α into three sections in the ME model: $[-2.0^\circ, 0.67^\circ]$, $[0.67^\circ, 3.33^\circ]$, and $[3.33^\circ, 6.0^\circ]$. Similar situations also appear in the transonic regime in predicting the C_l and C_m for the SC(2)0710 airfoil when α is around 2.5° and 0.5° . In the transonic regime, we divide the α domain into $[-1.5^\circ, 0.5^\circ]$, $[0.5^\circ, 2.5^\circ]$, and $[2.5^\circ, 4.5^\circ]$, and the largest errors also occur where the surrogate models are combined. Despite these issues in ME, the surrogate model is accurate.

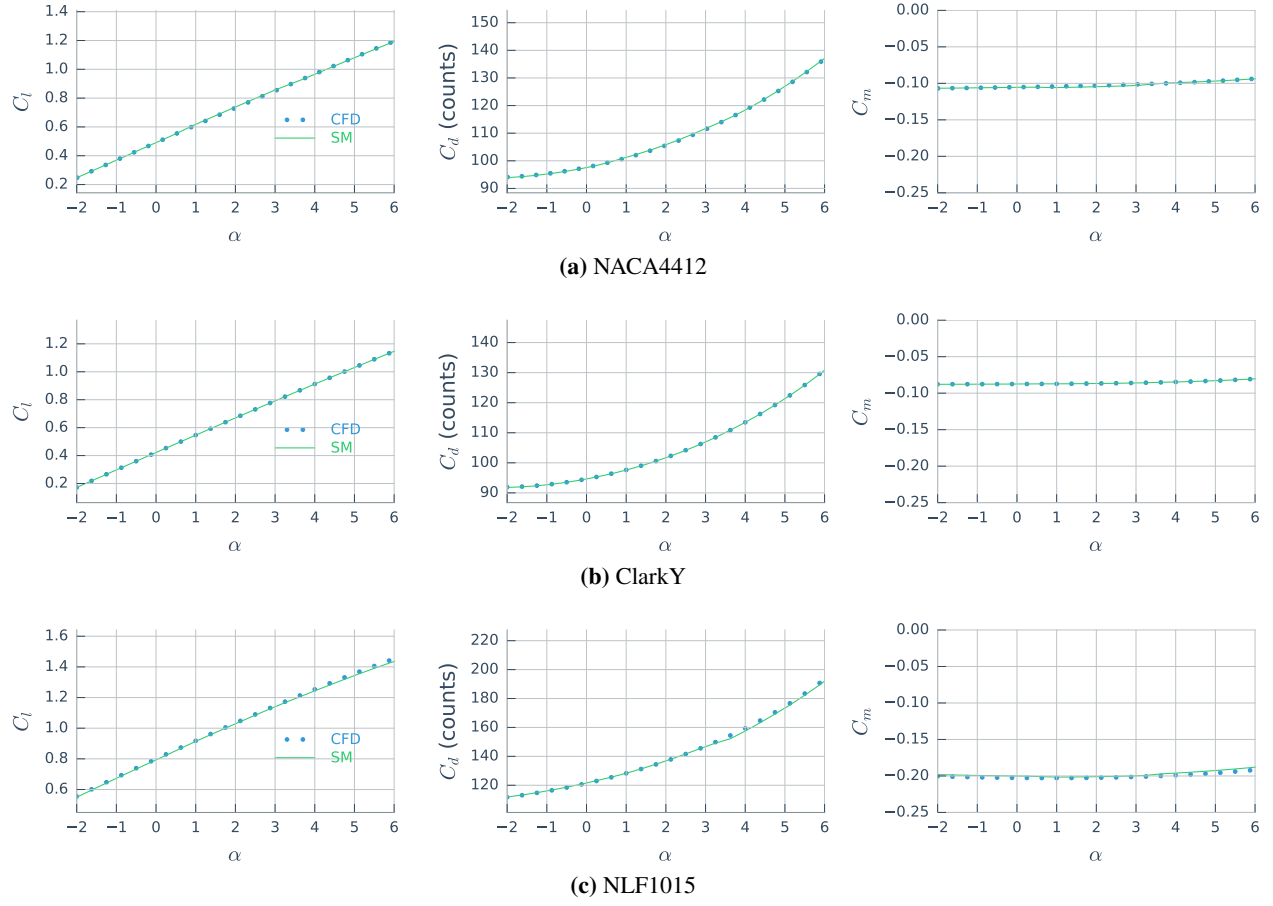


Fig. 16 Aerodynamic coefficients curves obtained by ADflow and the surrogate model in the subsonic regime ($M = 0.45$).

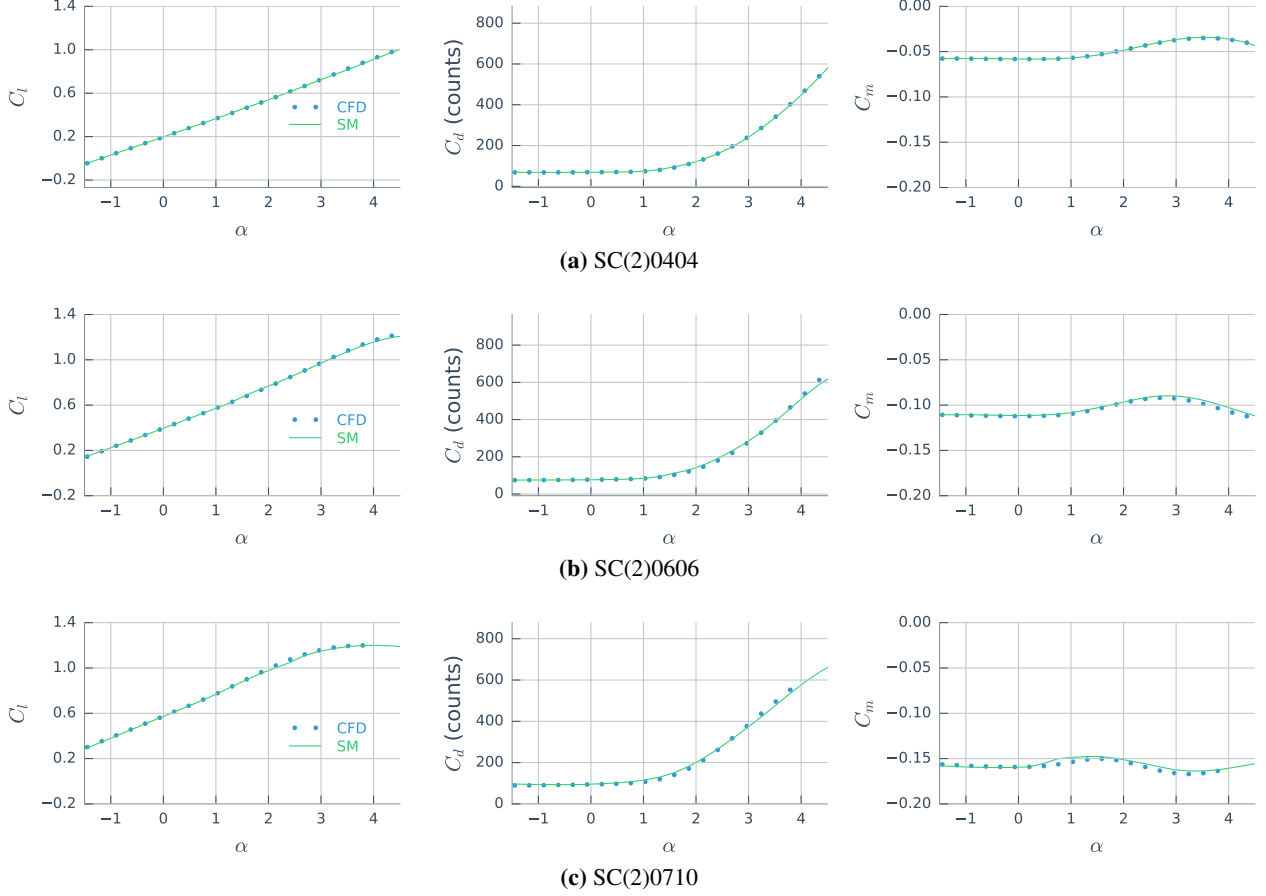


Fig. 17 Aerodynamic coefficients curves obtained by ADflow and the surrogate model in the transonic regime ($M = 0.73$).

7 Airfoil Shape Optimization

7.1 Optimization Method

Gradient-free optimization algorithms like the genetic algorithm converge slowly, requiring tens of thousands of aerodynamic analyses. Even with the fast analysis evaluation provided by a surrogate model, so many analyses would still require a considerable time. Gradient-based optimization algorithms, on the other hand, require less than one hundred aerodynamic analyses in this case. Since our surrogate model framework provides the required gradients, we elect to use a gradient-based optimization algorithm. Although multimodality is a common concern, all evidence so far points towards a unimodal design space in airfoil shape optimization [37, 38]. The multi-start gradient-based optimization we perform below also reinforces this point.

All optimization problems are solved using SNOPT (Sparse Nonlinear OPTimizer) [39], a quasi-Newton gradient-based optimizer that has shown to be efficient in large-scale constrained aerodynamic shape optimization problems [14, 38, 40]. The gradient information of the surrogate model is provided by the built-in functionality in the Surrogate Modeling Toolbox (SMT) [41]. The optimization tool is wrapped using the Python interface pyOpt [42] [43]. All optimization tests are subject to a thickness constraint to ensure practical optimal designs.

In all the following optimization cases, we use camber and thickness modes for the parameterization. In the optimization with the data-based approach, the aerodynamic analyses relies solely on surrogate model evaluations. The lower and upper optimization bounds of each mode are defined by computing the minimum and maximum values of the corresponding existing airfoil database, except for the first thickness mode. Optimization bounds of the first thickness mode are determined by the thickness constraints shown in Fig. 5. As previously described, we define two bound functions for each higher-order mode to exclude undesirable airfoil shapes, so we add them as constraints in the optimization to ensure that the optimal airfoils are within the desired design space range.

7.2 Subsonic Airfoil Optimization

We now perform airfoil design optimization for various lift coefficients using the subsonic surrogate model at $M = 0.45$. To reinforce the idea that the design space is unimodal, all cases are solved using a multi-start strategy using Sobol sampling [44]. All surrogate-based optimizations are compared to optimizations performed via direct CFD evaluations.

The objective is to minimize C_d with respect to the 14 mode shapes and angle of attack, subject to a C_l constraint. The baseline thickness distribution is based on the NACA0012 airfoil. The thickness of the optimized airfoil is constrained to be no less than 90% of the baseline airfoil thickness. We enforce this by applying thickness constraints at five positions along the chord ($x/c = 0.1, 0.3, 0.5, 0.7, 0.9$).

Fig. 18 shows the optimization histories for the surrogate model and ADflow for a single-point $C_l = 0.5$ constraint. The multi-start generates 50 different starting airfoils at different angles of attack. The red lines represent the average values of 50 optimizations in each iteration, while the blue intervals show the minimum and maximum in these 50 values.

We use \mathbf{a}_{ct} to represent the vector of shape design variables. The optimal values are taken to be the average values among the 50 optimization results. In Figs. 18, we see that the difference in C_d between the 50 optimizations is within 10^{-12} , while the difference in C_l is less than 10^{-10} . The average difference in the L2 norm of the design variables is 10^{-6} . So, the results of 50 optimizations are consistent. In Fig. 18, we see that all 50 optimizations converge to the same result, and therefore, there is no evidence of multimodality when using either the ADflow or the surrogate model.

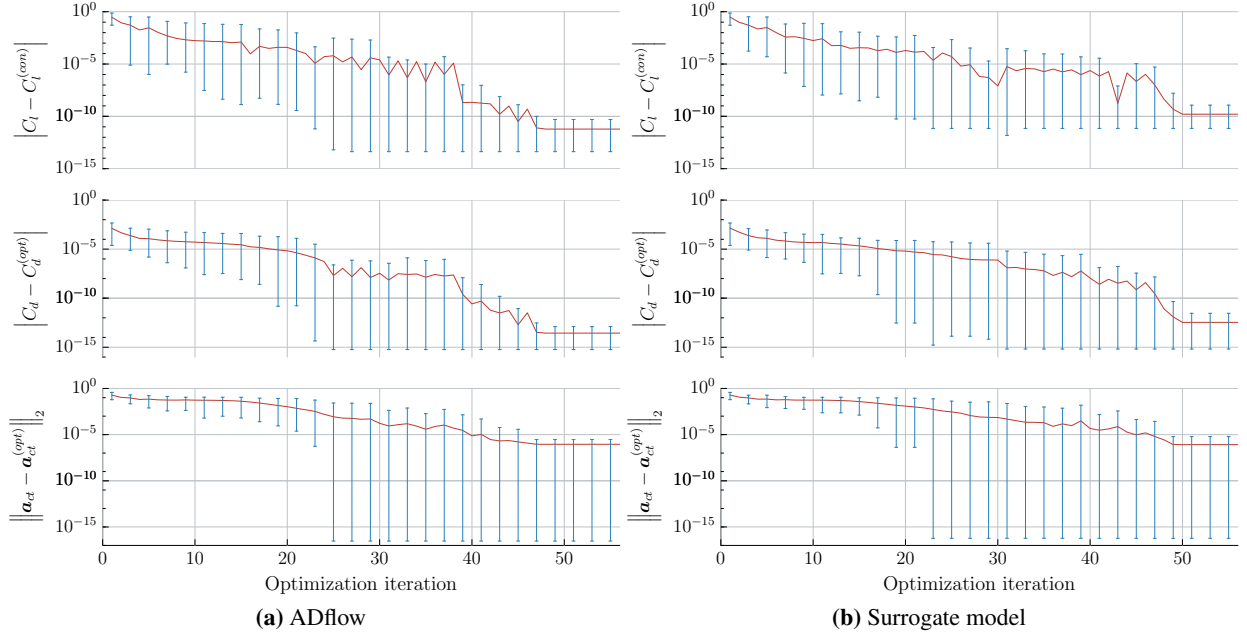


Fig. 18 Convergence history of gradient-based optimization starting from 50 Sobol sampling airfoils with $C_l = 0.5$.

Fig. 19 shows the optimization results using the surrogate model (SM) and direct ADflow evaluation (CFD) for two different lift coefficient constraints: $C_l = 0.5$ and $C_l = 0.75$. The optimized airfoils are plotted on the bottom of each sub-figure. The gray areas behind the airfoils are the result of plotting 1000 airfoil samples taken from the optimization design space, which demonstrates how large the design space is. We analyze the surrogate-optimized airfoil with ADflow, and vice-versa. The numerical results for C_d and C_l are listed in Fig. 19. In addition, the C_p distributions obtained with ADflow are shown on the right-hand side.

For the $C_l = 0.5$ case, the C_d of airfoils optimized with the surrogate model (SM) and ADflow (CFD) agree to within a hundredth of a count, and there is only a slight difference in C_l (within 0.001). For the $C_l = 0.75$ case, the surrogate model has a C_d that is within a tenth of a count. Given the minute differences, these optimizations demonstrate the accuracy and feasibility of the subsonic surrogate model for airfoil design optimization.

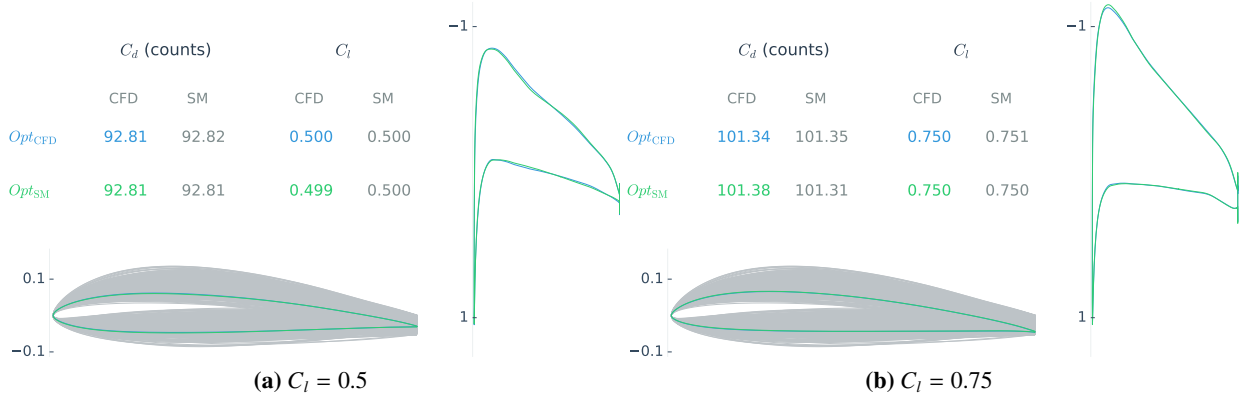


Fig. 19 Optimization results with different C_l constraints for subsonic flight conditions.

7.3 Transonic Airfoil Optimization

We now optimize airfoils for transonic flight conditions. The optimization is performed with respect to the eight mode shapes used to create the transonic surrogate model. The airfoil used for the baseline thickness constraints is the SC(2)0610, using the same positions and values for the five thickness constraints as for the subsonic case. Again, we use multi-start strategy for each optimization and verify that all optimizations consistently converge to the same result. We conduct both single-point and multipoint optimizations, and in addition to the C_l constraint, examine the effect of enforcing a C_m constraint.

7.3.1 Single-point Optimization

For the single-point transonic airfoil optimization, we minimize the drag at $M = 0.72$ and $C_l = 0.82$. We investigate the influence of C_m constraint on the optimization by optimizing both with and without this constraint. Fig. 20 shows the optimization results given by the surrogate model and ADflow with and without the C_m constraint. Overall, the differences in C_d between the CFD-optimized result and the surrogate-model result evaluated with CFD are greater than for the subsonic case, but they are still less than 2 counts. The surrogate model prediction of C_l has an error of 0.01 or less at the optimum.

When minimizing C_d for transonic conditions, we obtain a supercritical airfoil, which tends to use trailing edge camber as an effective way to meet the lift constraint without incurring too much of a drag penalty. However, the side effect is that this causes a large nose-down pitching moment, which is undesirable because it causes a higher trim drag when used in an aircraft configuration. Therefore, we enforce a C_m constraint and analyze the results for two different C_m constraint values: -0.12 and -0.095 . When comparing the results from the optimizations with and without the C_m constraint, we can see that the higher the C_m , the higher the drag, as expected. Thus, there is a trade between drag and moment that can only be fully assessed when designing the full aircraft configuration. However, the surrogate model provides a quick way to assess the trade at the airfoil level that could be used in aircraft-level design studies.

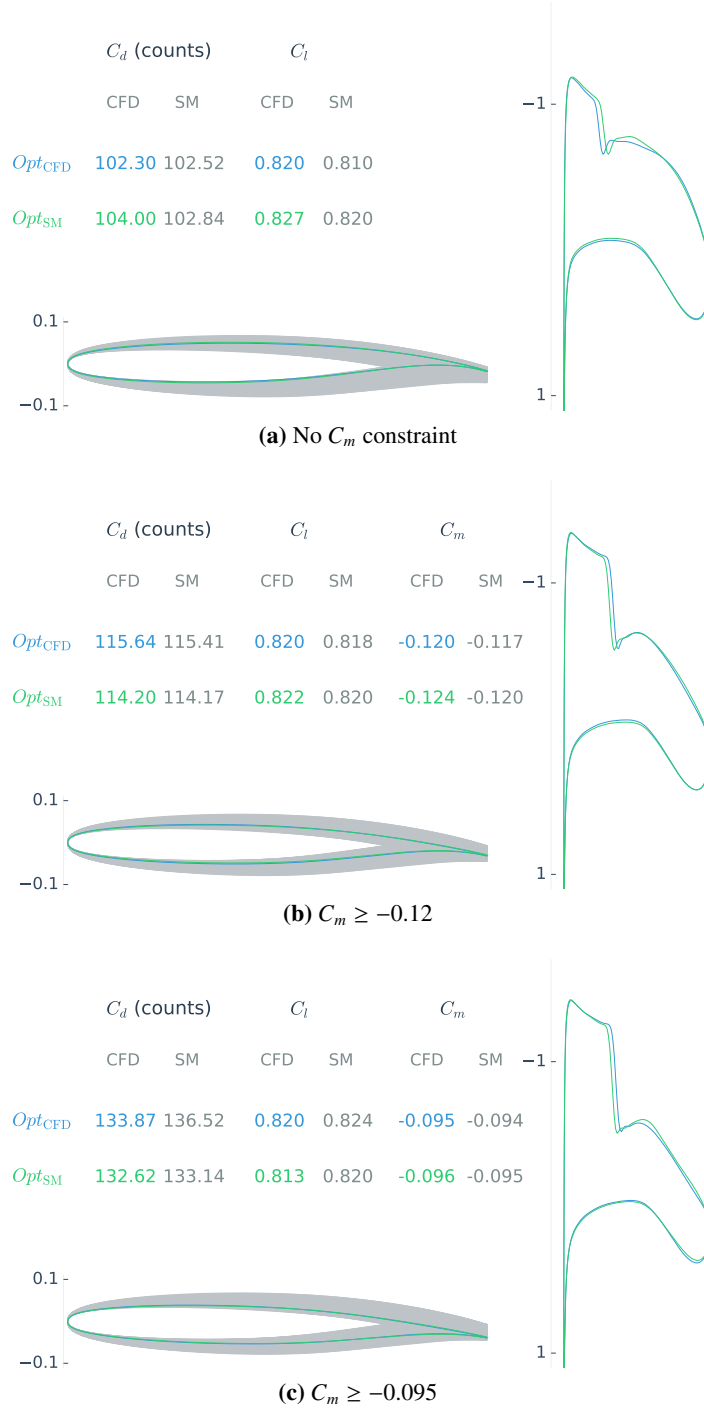


Fig. 20 Single-point optimization results with different C_m constraints in the transonic regime.

7.3.2 Multipoint optimization

Single-point optimization is not practical, since it optimizes the performance at a single flight condition to the detriment of other flight conditions [45]. In practice, aircraft need to operate efficiently at a range of flight conditions due to different weights and altitudes in cruise, as well as in climb and descent. To improve the robustness of the optimized airfoil, we choose the five flight conditions shown in Fig. 21. The optimization is to minimize the average C_d of the airfoil at these five conditions, subject to $C_m \geq -0.12$ for all five conditions. The optimized airfoil is also subject to the same thickness constraints as the single-point optimization case.

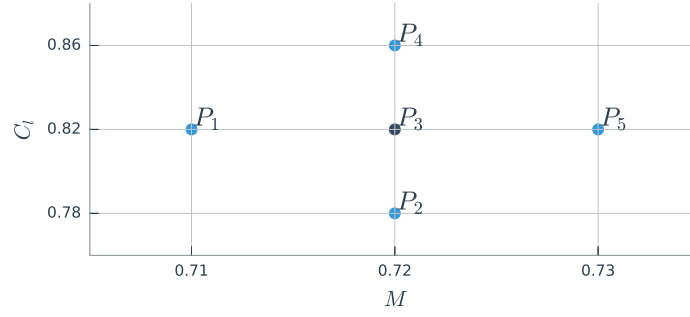


Fig. 21 Multipoint optimization flight conditions

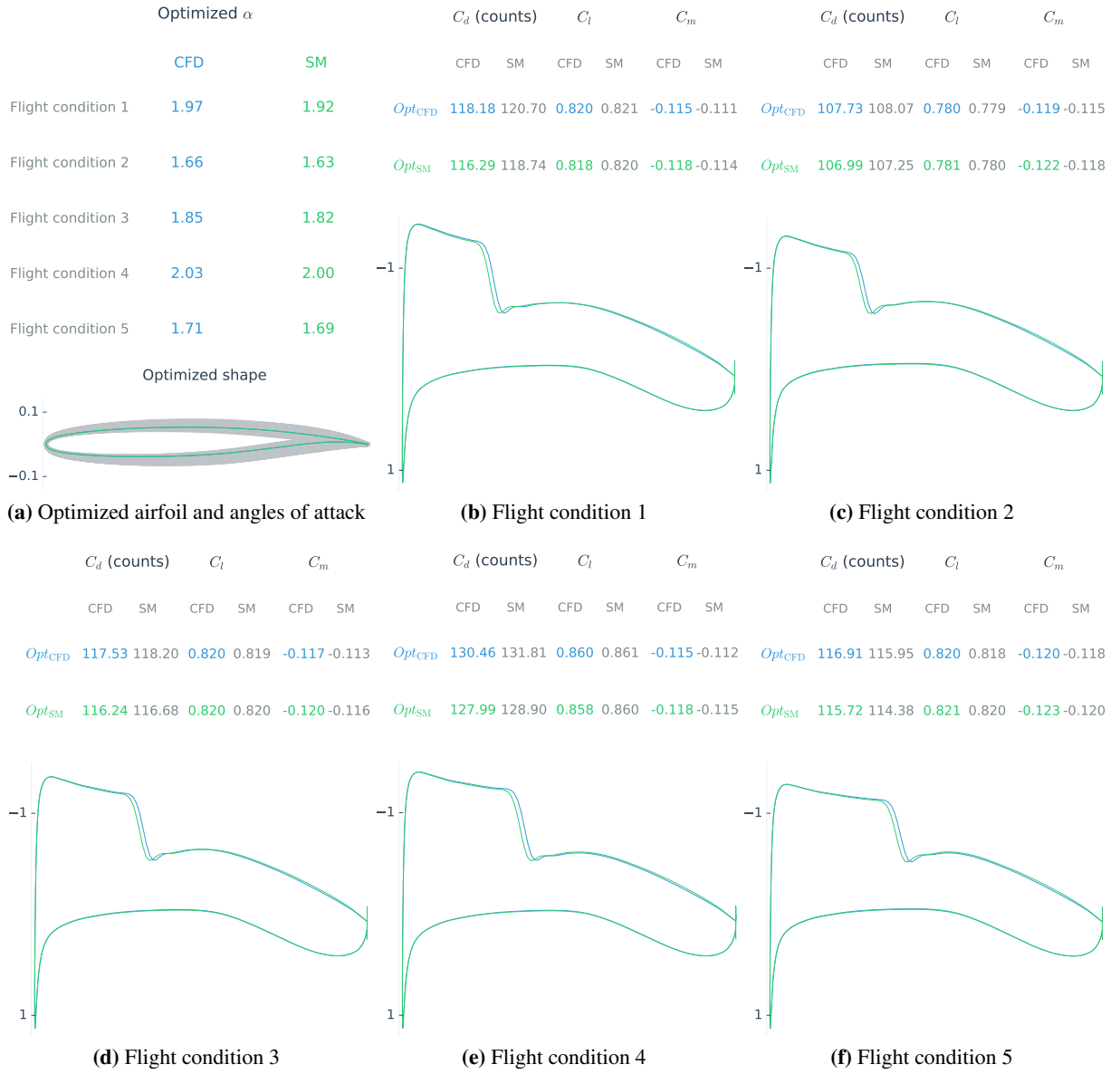


Fig. 22 Multipoint optimization results in the transonic regime

Multipoint optimization with different C_l and C_m constraints for different Mach numbers is more challenging, because it calls the surrogate model many more times (four more times per optimization iteration). As we can see in Fig. 22, the surrogate-optimized result is close to that obtained by ADflow. The optimal airfoil shapes look identical, and the C_p distributions at all five flight conditions are also identical except for the position of the shock, which deviates slightly. The differences in C_d for the surrogate-optimized airfoil are within 2.5 counts compared to ADflow, and the differences in C_l and C_m are within 0.03. This demonstrates that the transonic surrogate model is accurate and reliable enough to handle multipoint airfoil optimization. In the future, we would like to merge the transonic and subsonic surrogate models to enable multipoint optimization problems considering both subsonic and transonic performance.

8 Conclusions

A data-based approach is adopted to provide fast and reliable aerodynamic analysis and design optimization of airfoils. The approach relies on using a database of existing airfoils to determine the mode shapes to represent the airfoil shapes. Camber and thickness modes derived from about 1100 UIUC airfoils and 21 NASA SC(2) supercritical airfoils are computed to parameterize subsonic and transonic airfoils. The geometry design space is wide enough to include all these existing airfoils. The design space is enriched with more than 100,000 generated airfoil samples that are also evaluated using RANS CFD.

Surrogate models are created based on this data by using a mixture of experts strategy coupled to a gradient-enhanced kriging approach. The result is an accurate surrogate model of all possible airfoils that enables drag, lift, and moment coefficients to be computed nearly instantly for any combination of angle of attack, Mach number, and airfoil shape. Together with a gradient-based optimizer, an airfoil design optimization problem can be solved in a few seconds.

A series of C_d minimization problems show that the difference in C_d between airfoils optimized by this surrogate model approach and airfoils optimized directly using RANS CFD is within 0.04 and 2.5 counts in the subsonic and transonic regimes, respectively. The multi-start optimizations performed in this paper reinforce the fact that multi-modality is not a concern in airfoil shape optimization.

The optimization using the proposed approach takes about two seconds using one processor on a personal desktop computer, while the cost of a high-fidelity gradient-based optimization using CFD takes one to two hours using 16 processors in a high-performance parallel computer. This approach makes it possible to perform interactive airfoil analysis and design using any modern computer with accuracy similar to RANS CFD. This data-based approach also opens the door for the analysis and design optimization of other engineering systems that normally require high-fidelity modeling.

Acknowledgments

This work was partially supported by the Air Force Office of Scientific Research (AFOSR) MURI on “Managing multiple information sources of multi-physics systems”, Program Officer Jean-Luc Cambier, Award Number FA9550-15-1-0038. The authors would like to thank other members at the MDO Lab for numerical ideas and suggestions.

References

- [1] Drela, M., “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,” *Lecture Notes in Engineering*, Springer Berlin Heidelberg, 1989, pp. 1–12. doi:[10.1007/978-3-642-84010-4_1](https://doi.org/10.1007/978-3-642-84010-4_1).
- [2] Masters, D. A., Taylor, N. J., Rendall, T. C. S., Allen, C. B., and Poole, D. J., “Geometric Comparison of Aerofoil Shape Parameterization Methods,” *AIAA Journal*, Vol. 55, No. 5, May 2017, pp. 1575–1589. doi:[10.2514/1.j054943](https://doi.org/10.2514/1.j054943).
- [3] Kulfan, B. M., “Universal Parametric Geometry Representation Method,” *Journal of Aircraft*, Vol. 45, No. 1, January 2008, pp. 142–158. doi:[10.2514/1.29958](https://doi.org/10.2514/1.29958).
- [4] Hicks, R. M. and Henne, P. A., “Wing Design by Numerical Optimization,” *Journal of Aircraft*, Vol. 15, No. 7, July 1978, pp. 407–412. doi:[10.2514/3.58379](https://doi.org/10.2514/3.58379).

- [5] Poole, D. J., Allen, C. B., and Rendall, T., “Aerofoil Design Variable Extraction for Aerodynamic Optimization,” *21st AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, June 2013. doi:[10.2514/6.2013-2705](https://doi.org/10.2514/6.2013-2705).
- [6] Toal, D. J. J., Bressloff, N. W., Keane, A. J., and Holden, C. M. E., “Geometric Filtration Using Proper Orthogonal Decomposition for Aerodynamic Design Optimization,” *AIAA Journal*, Vol. 48, No. 5, May 2010, pp. 916–928. doi:[10.2514/1.41420](https://doi.org/10.2514/1.41420).
- [7] Ghoman, S., Wang, Z., Chen, P., and Kapania, R., “A POD-based Reduced Order Design Scheme for Shape Optimization of Air Vehicles,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, American Institute of Aeronautics and Astronautics, April 2012. doi:[10.2514/6.2012-1808](https://doi.org/10.2514/6.2012-1808).
- [8] Poole, D. J., Allen, C. B., and Rendall, T. C. S., “Metric-Based Mathematical Derivation of Efficient Airfoil Design Variables,” *AIAA Journal*, Vol. 53, No. 5, May 2015, pp. 1349–1361. doi:[10.2514/1.j053427](https://doi.org/10.2514/1.j053427).
- [9] Robinson, G. M. and Keane, A. J., “Concise Orthogonal Representation of Supercritical Airfoils,” *Journal of Aircraft*, Vol. 38, No. 3, May 2001, pp. 580–583. doi:[10.2514/2.2803](https://doi.org/10.2514/2.2803).
- [10] Jeong, S., Murayama, M., and Yamamoto, K., “Efficient Optimization Design Method Using Kriging Model,” *Journal of Aircraft*, Vol. 42, No. 2, March 2005, pp. 413–420. doi:[10.2514/1.6386](https://doi.org/10.2514/1.6386).
- [11] Han, Z.-H. and Görtz, S., “Hierarchical Kriging Model for Variable-Fidelity Surrogate Modeling,” *AIAA Journal*, Vol. 50, No. 9, September 2012, pp. 1885–1896. doi:[10.2514/1.j051354](https://doi.org/10.2514/1.j051354).
- [12] Jameson, A., “Aerodynamic design via control theory,” *Journal of Scientific Computing*, Vol. 3, No. 3, September 1988, pp. 233–260. doi:[10.1007/bf01061285](https://doi.org/10.1007/bf01061285).
- [13] Kenway, G. K. W. and Martins, J. R. R. A., “Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration,” *Journal of Aircraft*, Vol. 51, No. 1, January 2014, pp. 144–160. doi:[10.2514/1.C032150](https://doi.org/10.2514/1.C032150).
- [14] Chen, S., Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., “Aerodynamic Shape Optimization of the Common Research Model Wing-Body-Tail Configuration,” *Journal of Aircraft*, Vol. 53, No. 1, January 2016, pp. 276–293. doi:[10.2514/1.C033328](https://doi.org/10.2514/1.C033328).
- [15] Koziel, S., Ciaurri, D. E., and Leifsson, L., “Surrogate-Based Methods,” *Computational Optimization, Methods and Algorithms*, Springer Berlin Heidelberg, 2011, pp. 33–59. doi:[10.1007/978-3-642-20859-1_3](https://doi.org/10.1007/978-3-642-20859-1_3).
- [16] Koziel, S. and Leifsson, L., editors, *Surrogate-Based Modeling and Optimization*, Springer New York, 2013. doi:[10.1007/978-1-4614-7551-4](https://doi.org/10.1007/978-1-4614-7551-4).
- [17] Han, Z.-H., Görtz, S., and Zimmermann, R., “Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function,” *Aerospace Science and Technology*, Vol. 25, No. 1, March 2013, pp. 177–189. doi:[10.1016/j.ast.2012.01.006](https://doi.org/10.1016/j.ast.2012.01.006).
- [18] Zhang, Y., Sung, W. J., and Mavris, D. N., “Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient,” *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, January 2018. doi:[10.2514/6.2018-1903](https://doi.org/10.2514/6.2018-1903).
- [19] Nadarajah, S. and Jameson, A., “A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization,” *Proceedings of the 38th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2000. doi:[10.2514/6.2000-667](https://doi.org/10.2514/6.2000-667).
- [20] 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, April 2008, pp. 863–873. doi:[10.2514/1.29123](https://doi.org/10.2514/1.29123).
- [21] 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, September 2003, pp. 245–262. doi:[10.1145/838250.838251](https://doi.org/10.1145/838250.838251).

- [22] Selig, M. S., “UIUC Airfoil Coordinates Database,” https://m-selig.ae.illinois.edu/ads/coord_database.html.
- [23] Bouhlel, M. A. and Martins, J. R. R. A., “Gradient-enhanced kriging for high-dimensional problems,” *Engineering with Computers*, 2018. doi:[10.1007/s00366-018-0590-x](https://doi.org/10.1007/s00366-018-0590-x), (In press).
- [24] van der Weide, E., Kalitzin, G., Schluter, J., and Alonso, J. J., “Unsteady Turbomachinery Computations Using Massively Parallel Platforms,” *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006. doi:[10.2514/6.2006-421](https://doi.org/10.2514/6.2006-421).
- [25] 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, Jul. 2013. doi:[10.2514/6.2013-2581](https://doi.org/10.2514/6.2013-2581).
- [26] 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](https://doi.org/10.2514/6.1992-439).
- [27] Krige, D. G., “A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand,” *Journal of the Chemical, Metallurgical and Mining Society*, Vol. 52, 1951, pp. 119–139.
- [28] Matheron, G., “Principles of geostatistics,” *Economic Geology*, Vol. 58, No. 8, December 1963, pp. 1246–1266. doi:[10.2113/gsecongeo.58.8.1246](https://doi.org/10.2113/gsecongeo.58.8.1246).
- [29] Cressie, N., “Spatial prediction and ordinary kriging,” *Mathematical Geology*, Vol. 20, No. 4, May 1988, pp. 405–421. doi:[10.1007/bf00892986](https://doi.org/10.1007/bf00892986).
- [30] Sacks, J., Schiller, S. B., and Welch, W. J., “Designs for Computer Experiments,” *Technometrics*, Vol. 31, No. 1, February 1989, pp. 41–47. doi:[10.1080/00401706.1989.10488474](https://doi.org/10.1080/00401706.1989.10488474).
- [31] Boopathy, K. and Rumpfkeil, M. P., “Unified Framework for Training Point Selection and Error Estimation for Surrogate Models,” *AIAA Journal*, Vol. 53, No. 1, January 2015, pp. 215–234. doi:[10.2514/1.j053064](https://doi.org/10.2514/1.j053064).
- [32] Masoudnia, S. and Ebrahimpour, R., “Mixture of experts: a literature survey,” *Artificial Intelligence Review*, Vol. 42, No. 2, May 2012, pp. 275–293. doi:[10.1007/s10462-012-9338-y](https://doi.org/10.1007/s10462-012-9338-y).
- [33] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E., “Adaptive Mixtures of Local Experts,” *Neural Computation*, Vol. 3, No. 1, February 1991, pp. 79–87. doi:[10.1162/neco.1991.3.1.79](https://doi.org/10.1162/neco.1991.3.1.79).
- [34] Tang, B., Heywood, M., and Shepherd, M., “Input partitioning to mixture of experts,” *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN02 (Cat. No.02CH37290)*, IEEE, 2002. doi:[10.1109/ijcnn.2002.1005474](https://doi.org/10.1109/ijcnn.2002.1005474).
- [35] Hartigan, J. A. and Wong, M. A., “Algorithm AS 136: A K-Means Clustering Algorithm,” *Applied Statistics*, Vol. 28, No. 1, 1979, pp. 100. doi:[10.2307/2346830](https://doi.org/10.2307/2346830).
- [36] Liem, R. P., Mader, C. A., and Martins, J. R. R. A., “Surrogate Models and Mixtures of Experts in Aerodynamic Performance Prediction for Aircraft Mission Analysis,” *Aerospace Science and Technology*, Vol. 43, June 2015, pp. 126–151. doi:[10.1016/j.ast.2015.02.019](https://doi.org/10.1016/j.ast.2015.02.019).
- [37] Chernukhin, O. and Zingg, D. W., “Multimodality and Global Optimization in Aerodynamic Design,” *AIAA Journal*, Vol. 51, No. 6, June 2013, pp. 1342–1354. doi:[10.2514/1.j051835](https://doi.org/10.2514/1.j051835).
- [38] Yu, Y., Lyu, Z., Xu, Z., and Martins, J. R. R. A., “On the Influence of Optimization Algorithm and Starting Design on Wing Aerodynamic Shape Optimization,” *Aerospace Science and Technology*, Vol. 75, April 2018, pp. 183–199. doi:[10.1016/j.ast.2018.01.016](https://doi.org/10.1016/j.ast.2018.01.016).
- [39] Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP algorithm for large-scale constrained optimization,” *SIAM Journal of Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:[10.1137/S1052623499350013](https://doi.org/10.1137/S1052623499350013).

- [40] 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, January 2016, pp. 113–128. doi:[10.2514/1.J054154](https://doi.org/10.2514/1.J054154).
- [41] Bouhlel, M. A., “Surrogate Modeling Toolbox,” <https://github.com/SMTorg/smt>.
- [42] 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, January 2012, pp. 101–118. doi:[10.1007/s00158-011-0666-3](https://doi.org/10.1007/s00158-011-0666-3).
- [43] Kenway, G. K. W., “pyOptSparse – PYthon OPTimization (Sparse) Framework,” <https://github.com/mdolab/pyoptsparse>.
- [44] Sobol, I., “On the distribution of points in a cube and the approximate evaluation of integrals,” *USSR Computational Mathematics and Mathematical Physics*, Vol. 7, No. 4, January 1967, pp. 86–112. doi:[10.1016/0041-5553\(67\)90144-9](https://doi.org/10.1016/0041-5553(67)90144-9).
- [45] 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, April 2015, pp. 968–985. doi:[10.2514/1.J053318](https://doi.org/10.2514/1.J053318).

Appendix

The raw UIUC airfoil data is not in a uniform format and is not directly usable for our purposes, so this data requires some preprocessing. For each airfoil, the preprocessing procedure is as follows:

1. Interpolate the airfoil using a cubic B-spline curve.
2. If the airfoil has a blunt trailing edge, make the trailing edge sharp (zero thickness).
3. Define the leading edge by finding the farthest point from the trailing edge on the curve.
4. Normalize the airfoil by making the chord equal to one and rotate the airfoil to make the angle of attack zero.
5. Save the curve into a uniform format sharing the same point distribution scheme. For UIUC airfoils,

$$x_i = \frac{1}{2} \left(\cos \frac{2\pi(i-1)}{250} + 1 \right), \quad i = 1, 2, \dots, 251, \quad (14)$$

while for SC(2) airfoils,

$$x_i = \frac{1}{2} \left(\cos \frac{2\pi(i-1)}{300} + 1 \right), \quad i = 1, 2, \dots, 301. \quad (15)$$

6. Smooth the airfoil using the Laplacian smoothing algorithm below using a tolerance of $\varepsilon = 0.3\%$.

Suppose we have n discrete points $(x_1, y_1^0), \dots, (x_n, y_n^0)$ on the upper or lower surfaces of a given airfoil. We use Laplacian smoothing by modifying the y coordinates as follows:

$$y_i^{k+1} = \rho(x_i) (y_{i-1}^k + y_{i+1}^k) + (1.0 - \rho(x_i)) y_i^k, \quad 1 < i < n, \quad (16)$$

where $\rho(x_i)$ is a weighted function and its value increases (to a maximum of 0.25) with the rise of the curvature at point (x_i, y_i) , and we fix the y coordinate of the leading edge point to zero ($y_{n_f} = 0$) during the smoothing procedure. This is terminated when

$$\frac{y^k - y^0}{y^0} > \varepsilon. \quad (17)$$