Physics-Aware Surrogate-based Optimization with Transfer Mapping Gaussian Processes: for Bio-inspired Flow Tailoring

Payam Ghassemi*, Amir Behjat†, Chen Zeng‡, Sumeet Lulekar§, Rahul Rai¶, and Souma Chowdhury‖
University at Buffalo, Buffalo, NY, 14260

This paper presents a physics-aware surrogate-based approach (aka PhySBO) for computationally efficient optimization of complex systems. This approach is founded on a new hybrid model. This hybrid model combines partial physics and Gaussian process models in a specialized manner that facilitates more generalizable output predictions compared to pure data-driven models (aka surrogate models) and standard low-fidelity-physics/surrogate ensemble models. More specifically, this hybrid modeling approach called OPTMA exploits the potential relationship between the inputs to the partial physics model and the inputs to the full physics model, where this relationship is mapped by a transfer Gaussian process model (GP). The PhySBO method is applied to design surface riblets for bio-inspired passive flow tailoring, where costly CFD simulations are needed to gather high-fidelity samples. In this case, a potential flow solver acting on a 2D airfoil is used as the partial physics model. Here the transfer GP model transforms the inputs from the geometric riblet features on the original 3D ribleted NACA airfoil to geometric and incoming-flow features for the partial physics model. The OPTMA model is estimated to provide 10^5 times reduction in computing time. Results show the proposed hybrid model is twice as accurate and robust (when tested on unseen samples) compared to a pure data-driven model, when the number of training samples is small and the training and test samples come from different distribution. This shows the generalizability capacity of the OPTMA architecture (the median of error is ~0.5%). Based on the OPTMA (hybrid) model, the PhySBO framework is able to converge upon an optimum design with substantial computational efficiency, while providing an optimum that is validated to be 99.73% accurate w.r.t. corresponding high-fidelity estimate.

I. Introduction

A. Physics-Aware Surrogate-based Optimization

Optimizing complex systems often involves computationally expensive simulations (e.g., CFD) to evaluate system behavior and estimate quantities of interest. In many cases, using high-fidelity models or "full physics" simulations in optimization are infeasible, especially in time-sensitive usage contexts like online planning. While surrogate models and surrogate-based optimization [1–4] provides a tractable alternative, in their native form they tend to compromise on the fidelity of the optimization process. In general, purely data-driven low-fidelity surrogate models shows low performance in generalizing and extrapolating [5, 6] - specifically, when the training dataset is limited [7]. More importantly, they do not implicitly guarantee adherence to even basic physics laws guiding the system under study. Due to these inaccuracies, purely data-driven surrogate models often mislead the search process during optimization, leading to sub-optimal or even infeasible solutions. Hybrid modeling architectures (or tuned fidelity modelings) have been proposed to overcome to these aforementioned issues by combining computationally inexpensive partial physics models with purely data-driven models have been proposed to address the aforementioned issues [8, 9]. However, the existing hybrid modeling architectures do not fully address these issues. In this paper, we present a *physics-aware surrogate-based optimization (PhySBO)* method by integrating a physics-aware surrogate model and a particle swarm optimization algorithm [10],

*Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, and AIAA student member.
†Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, and AIAA student member.
‡Ph.D. Student, Department of Mechanical and Aerospace Engineering, and AIAA student member.
§M.S. Student, Department of Mechanical and Aerospace Engineering.
¶Associate Professor, Department of Mechanical and Aerospace Engineering, and AIAA member.
‖Assistant Professor, Department of Mechanical and Aerospace Engineering, and AIAA senior member. Corresponding author. Email: soumacho@buffalo.edu

allowing a fast and reliable optimization. The physics-aware surrogate model is defined as a combination of a surrogate model like an artificial neural network (ANN) or Gaussian Process (GP) and a computationally inexpensive partial physics model to evaluate the design variables. This new PhySBO is used to design ribleted 3D airfoil surfaces for bio-inspired flow tailoring. The remaining portion of this introduction section provides a review of physics-aware hybrid modeling, and converges on the objectives of this paper.

B. Physics-Aware Hybrid Modeling

Data-driven surrogate models are commonly used to provide a tractable and inexpensive approximation of the actual system response in engineering analysis and design activities, e.g., domain exploration, sensitivity analysis, development of empirical models, and optimization. One of the most popular classes of approximation models are surrogate models or metamodels [11], which are purely data-driven models, and are not directly derived from the physics of the system being modeled. Major surrogate modeling methods include polynomial response surfaces [12], Kriging [13, 14], moving least square [15, 16], radial basis functions (RBF) [17], support vector regression (SVR) [18], and artificial neural networks [19]. The data-driven models are frequently shown to provide competitive predictions [20–22]. However, as mentioned earlier, the purely data-driven models are poor at generalizing and extrapolating [5, 6]. In addition, they are very likely to predict a system response which is not aligned with physics laws of the actual system. For addressing these issues, some studies proposed of hybrid modeling architectures (or tuned fidelity models) to combine a computationally-inexpensive partial physics models with purely data-driven models [8, 9].

The hybrid architectures can be classified into two major sub-architectures [23], namely: parallel [23–25], and serial sub-architectures [26–28]. In the parallel sub-architectures, the partial physics model and the pure Data-Driven model receive same inputs and independently predict the system response, then their outputs are fused together to form the final outputs. There exist three main concerns regarding this class of hybrid models. First of all, if the exact point-wise correlation between the partial physics and full physics outputs is weak, the hybrid model is unreliable. Secondly, the parallel hybrid models do not readily facilitate implicit physics adherence characteristics due to the additive nature of the fusion approach. Lastly, the partial physics model, in spite of being computationally frugal, is under-exploited during the training process in most of the existing parallel hybrid models – often only is evaluated w.r.t. the inputs in the sparse full physics sample data. In the second class of hybrid modeling sub-architectures (i.e., serial sub-architectures), two types of arrangement are considered: 1) the data-driven model is cascaded with the partial physics model [26–28]; and 2) the data-driven model is utilized to tune the parameters of the partial physics model [23–25]. In the former approach, the fundamental justification for sequential interaction (e.g., treating partial physics model as an input to the data-driven model) is missing. The latter approach is often limited to parametric deviation contexts, where with a bounded parameter estimation, basic physics laws (e.g., conservation properties) can be expected to be followed in the final predictions.

In this paper, we propose a new hybrid modeling architecture based on a novel input-transfer concept, so-called Opportunistic Physics-mining Transfer Mapping Architecture (OPTMA). Our proposed model, similar to some serial hybrid models, guarantees physical law adherence prediction by virtue of using the partial physics model. In addition, we use a Gaussian process model for the transfer mapping model. Offline training of this transfer model can more effectively exploit the computationally-frugal partial physics model than typical serial or parallel hybrid architectures.

C. Objectives of this Paper

The primary objectives and associated contributions of this paper are given below:

1) Integrate, into particle swarm optimization, a physics-aware surrogate (hybrid) modeling that provides a reliable, physical law adherence prediction in a tractable manner. The central contribution lies in the ability (thus enabled) to estimate a system response using a computationally low cost partial physics model with high accuracy and reliability in the PhySBO framework.
2) Test and compare the proposed physics-aware hybrid modeling method, with standard surrogate modeling method.
3) Apply the new PhySBO method to perform optimization of bio-inspired surface riblets on 3D airfoil sections, to achieve maximum drag reduction. This objective lends itself to the practical contribution of this paper – namely provisions evidence of the effectiveness of the PhySBO method is solving complex optimization problems.

The remainder of the paper is structured as follows. The next section provides an overview of the Physics-aware surrogate-based optimization method, followed by a detailed description of the new hybrid modeling. For studying the effectiveness of the proposed algorithm, a CFD-based problem which is computationally very expensive to simulate is considered. Section III describes the automated riblet design framework and its optimization formulation, and discusses the results. Finally, concluding remarks are given at the end.

II. Physics-aware Surrogate-based Optimization

A. Overview of the PhySBO Framework

In this section, we introduce *Physics-aware surrogate-based optimization (PhySBO)*, which is a particle-based optimization that uses a physics-aware hybrid model during the optimization process to estimate the system response. Our proposed hybrid model is a computationally inexpensive alternative to high-fidelity (or "full physics") model in many applications (especially in time-sensitive applications), where using high-fidelity inside the optimization is not applicable. In addition, the proposed physics-aware hybrid model is more reliable than purely data-driven models; although they show competitive accurate predictions, they are not aware of the law of physics and they are prone to predict a value which is not adherence to physics laws guiding the system. This issue might mislead the search process and generate an infeasible solution. The implementation of the proposed PhySBO involves the six steps, which are illustrated within the PhySBO flowchart shown in Fig. 1. These six steps can be summarized as:

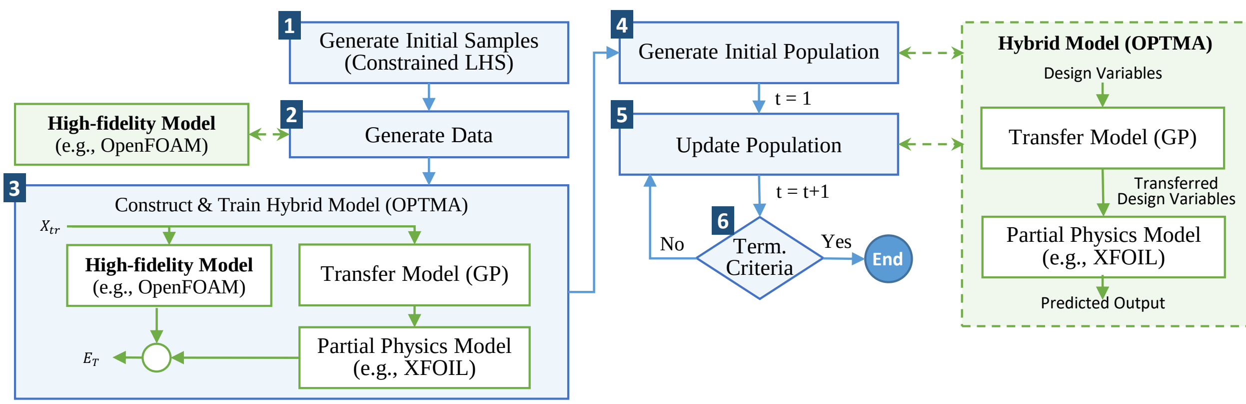


Fig. 1 Physics-aware surrogate-based optimization

Step 1 A set of initial sampling points are generated in the design space using a maximin Latin hypercube sampling (maximin LHS) [29, 30].
Step 2 The system response is calculated by running high-fidelity (full physics) model.
Step 3 Physics-aware hybrid model is constructed and trained. Further details will be given in Subsection II.B.
Step 4 Initial population is generated at $t = 1$, using the trained hybrid model.
Step 5 At every iteration (t) of the heuristic optimization algorithm, the trained hybrid model is used to update the function values of the population, and then set $t = t + 1$. In this paper, particle swarm optimization is the chosen optimization algorithm.
Step 6 The stopping criterion is checked. The following two different methods can be used as the termination criteria: (i) the difference between optimum values of five consecutive iterations is less than a threshold value, and (ii) the maximum allowed number of evaluations of function is reached. If the termination criterion is satisfied, the current optimum (the best global solution in case of PSO) is identified as the final optimum and the optimization process is terminated. Otherwise, go to **Step 5**.

In the next subsection, we talk in details about **Step 3** and then briefly describe **Step 5**.

B. Physics-aware Hybrid Modeling: OPTMA

In this paper, we propose a new input-transfer concept called the *Opportunistic Physics-mining Transfer Mapping Architecture (OPTMA)*. OPTMA is designed on the premise that if the outputs are continuous and bounded for both the partial and full physics, any linearly scaled output value given by the full physics can be produced by the partial physics model, albeit not necessarily in response to the same input vector. With this premise, we conceive the model construction problem to reduce to identifying the transfer mapping between the real input space and a modified input space such that the prediction of the partial physics model operating on the modified input is optimally close to the full physics output given by physical experiments or computationally-expensive simulations. This transfer mapping function is hypothesized to be more generalizable than a supervised direct mapping network when input-input correlations are stronger and simpler compared to output-output relations between the partial and full physics. The transfer mapping can be accomplished using multi-input-multi-output (MIMO) models (e.g., neural network and Gaussian process). Figure 2 shows how the partial physics model can be perceived as a node in both a neural network (NN)-based sequential hybrid architectures and our OPTMA architecture, and how differently they incorporate the partial physics model. In the next subsections, we first define the transfer mapping model, then we elaborate the training process of the OPTMA architecture.

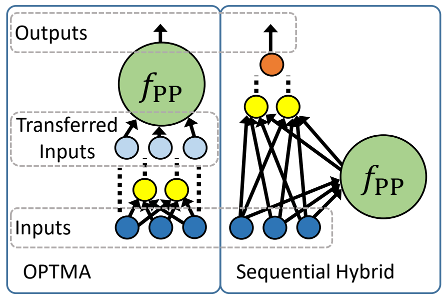


Fig. 2 OPTMA and Sequential Hybrid architectures; here neural network is used as a representative surrogate model; in practice, this could be of other type, e.g., a Gaussian Process model.

C. Definition of Transfer Model

Figure 3 shows the training process of the OPTMA architecture (part (I)), and how the prediction is achieved (part (II)). As shown in this figure, the transfer model ($\bar{M}_T$), the main component of the OPTMA architecture, is mapping input features (X) to transferred inputs (χ_T); i.e., $\bar{M}_T : X \to \chi_T$. Then, the transferred inputs are fed into the partial physics model ($f_{\text{PP}}(\chi_T)$) to generate final outputs (Y_T). An ideal transfer model maps the transferred inputs for the partial physics model such that the partial physics model generates the actual outputs (i.e., $Y_T = Y_{FP}$). Here, the ground truth, Y_{FP}, is the output of the full physics w.r.t. the original input and the estimated output, Y_T, is the output of the partial physics w.r.t. the corresponding transferred input.

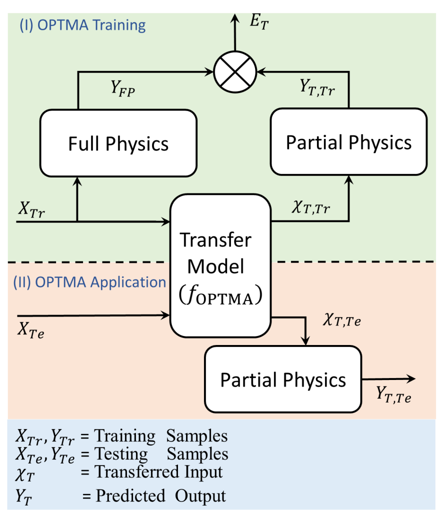


Fig. 3 OPTMA: (I) Training Process, (II) Application

D. Optimization Algorithm: Particle Swarm Optimization

In this paper, we use an advanced implementation of the PSO algorithm called mixed-discrete PSO (MDPSO), developed by [31]. Unlike the standard PSO algorithm [32], MDPSO provides: (i) an explicit diversity preservation capability that mitigates the possibility of premature stagnation of particles, and (ii) an ability to deal with both discrete and continuous design variables. MDPSO has been used to solve a wide variety of highly non-convex (often multimodal) mixed-integer nonlinear programming problems in wind farm design [33] and design of unmanned aerial vehicles [34]. Further description of the MDPSO algorithm can be found in the following paper [31].

III. Application of PhySBO: Optimization of Riblet-based Flow Tailoring

A. Riblet-based Flow Tailoring

Lulekar et al. [35, 36] presented a new approach to represent and optimize surface riblets inspired by passive surface features observed in marine animals, for the purpose of reducing drag coefficients. It has been shown in [35] that it is essential to have an automated riblet design framework to be able to find the optimal design of the riblets for a 3D airfoil. In this riblet-design framework, we are is using a set of open-source tools for running full physics RANS CFD simulation, where each run needs approximately 16 hours of computational time, executed on the UB CCR academic clusters using 2 compute nodes with Intel Xeon Processor E5-2660 (25M Cache, 2.20GHz), 16 cores per node, and 48GB RAM, indicating the expense of one single CFD simulation. The computational cost is significant and it is necessary to overcome this computational cost by minimizing the number of high-fidelity evaluations. For this purpose, we use PhySBO, where the hybrid model (a combination of data-driven model and computationally inexpensive partial physics model (2D CFD)) is trained over high-fidelity 3D CFD or physical experiments, can address this challenge. A brief description of the automated framework is given next.

B. Automated Riblet Design Framework

In order to be able to study and solve the optimization problem using to find the best drag reduction due to the riblets with the minimum human interaction, we developed an automated framework [35], which integrates disparate computational tools in batch mode. The design automation framework developed to solve the above-stated optimization problems, for exploring the aerodynamic benefits of bio-inspired surface riblets, is illustrated in Fig. 4. This computational framework is illustrated in Fig. 4 and it integrates the following major processes: 1) CFD simulations: for end-to-end batch evaluation of the flow behavior with different ribleted 3D airfoil surfaces; 2) design of experiments: for generation of training samples of riblet (CFD) evaluations that satisfy the geometric constraints in Eq. 4; and 3) hybrid surrogate modeling and surrogate-based optimization: to time-efficiently identify riblet shapes with minimum drag coefficient. Each of the major processes are developed either using existing open-source programs/libraries (the CFD components) or our own implementations (the surrogate modeling and optimization components). Further descriptions of the full physics model (including CAD modeling, mesh generation, CFD flow solver, and post processing) can be found in Appendix. In the next, we describe the riblet geometry, the DoE and the optimization formulation, then we talk in details about physics-aware hybrid model used in this problem.

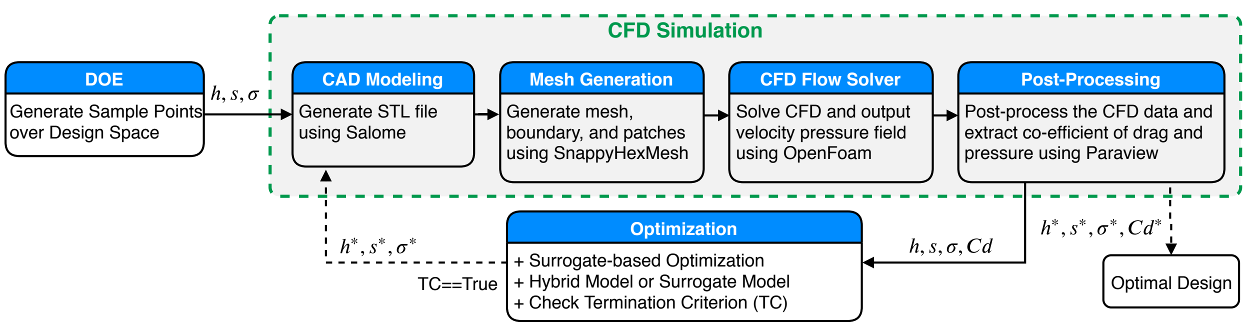


Fig. 4 Overall optimization framework: integrating optimization algorithm with full physics CFD simulation and hybrid model.

C. The Geometry of Riblets

Different type of riblet shapes can be considered. Here, we choose a smooth riblet shape, which can be termed as patterns of Gaussian-shaped ridge lines [35, 36]. The geometry of the Gaussian riblet is parameterized in terms of peak height (h), spacing (s) and the width (σ) of the curve. Thus, any point z, on the ridgeline can be expressed as

$$z(y) = he^{-y^2/\sigma^2} \tag{1}$$

where $-3\sigma < y < 3\sigma$. As illustrated in Fig. 5, the pattern of ridgelines covers the entire top surface of the 3D airfoil section, and the cross-sectional shape/size of the ridge does not change in the streamwise direction in our current implementation. In most of the studies are conducted by changing the protrusion height (h) and the spacing (s) between the riblets to impede the streamwise vortices and their performance is measured based on a specific set of non-dimensional parameters, to account for the change in size of the flow structures like vortex diameter, where h^+ and s^+ is given as follows:

$$h = \frac{h^+\nu}{U_\tau}\sqrt{\frac{2}{C_{d,\text{Barefoil}}}} \tag{2}$$

$$s = \frac{s^+\nu}{U_\tau}\sqrt{\frac{2}{C_{d,\text{Barefoil}}}} \tag{3}$$

Here ν, U_τ and $C_{d,\text{Barefoil}}$ respectively represent the kinematic viscosity, the friction velocity, and the drag coefficient for the airfoil without riblets (Barefoil). In this paper, we set the kinematic viscosity at $\nu = 1.5 \times 10^{-5}$. The drag coefficient, C_d, is computed using CFD simulations (described in Section III.B).

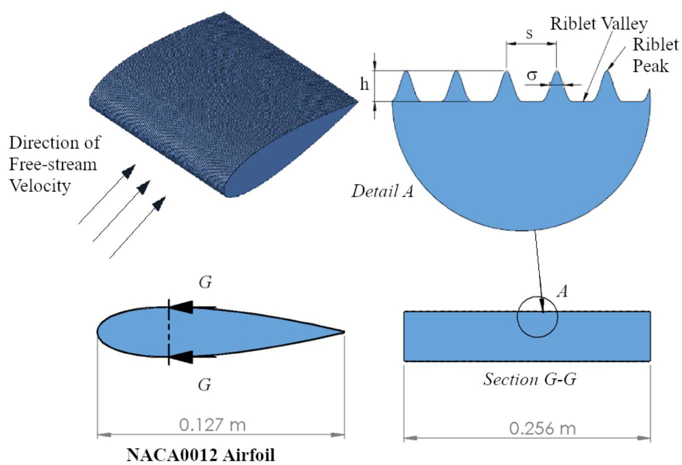


Fig. 5 The shape and orientation of the Gaussian riblets; and the ridgelines on a NACA0012 airfoil section, parallel to the free-stream velocity.

D. Optimization Formulation

It has been shown that the Gaussian riblets offer drag reductions on aerodynamic surfaces such as a 3D airfoil [36]. In this work, we pose it as a parametric shape optimization problem where the overall aerodynamic drag coefficient (C_d) of the ribleted surface is minimized. The problem is defined as finding the optimal riblet parameters (here, h, s, and σ) to minimize the drag coefficient, which leads to drag reduction. To this end, the height, spacing and riblet thickness parameter (σ) are optimized for a single angle of attack ($\alpha = 2°$).

$$
\begin{aligned}
\underset{\mathbf{x}=\{h,s,\sigma\}}{\text{minimize}} \quad & f = C_d(\mathbf{x}) \\
\text{subject to} \quad & g_1(\mathbf{x}):\ 6\sigma - s \le 0 \\
& g_2(\mathbf{x}):\ s - 6h \le 0 \\
& g_3(\mathbf{x}):\ \sigma - 0.6h \le 0 \\
& g_4(\mathbf{x}):\ C_d(\mathbf{x}) - C_d^{\text{thresh}} \le 0
\end{aligned}
\tag{4}
$$

Here $h \in [0.2, 0.6]$, $s \in [0.72, 3.6]$ and $\sigma \in [0.12, 0.46]$, with the dimensions of h, s being in millimeters. The first constraint, $g_1(.)$, is used to prevent adjacent Gaussian ridges from overlapping with each other. The second constraint, $g_2(.)$ in Eq. (4) restricts the inter-ridge spacing to 6 times the height of the ridge. This constraint is motivated by the work of Kennedy et al. [37], which reported that the height to spacing ratio should not be less than 1:6 in order to reduce the burst frequency of the low speed streaks into the boundary layer, thereby reducing the momentum transfer. The third constraint, $g_3(.)$, mitigates the possibility of the Gaussian curve to flatten out, which would otherwise undermine the favorable ability of riblets to impede the cross flow momentum. Based on prior literature [38–43], it was estimated that for favorable drag alteration, the protrusion height of the riblet should lie in the range of $[8, 50]$ wall units. Hence, Eqs. (2)- (3) are used to identify suitable bounds for h, s and σ. The last constraint, $g_4(.)$, is used to eliminate the extrapolation error of each surrogate model. Here, the threshold value of the drag coefficient (C_d^{thresh}) is set at 0.0080.

E. Physics-aware Hybrid Modeling for Riblet Design Problem

As mentioned earlier in this section, a high-fidelity RANS CFD simulation is computationally expensive; each single CFD simulation needs approximately 16 hours of computational time even by executing it on 32-core computing node. On the other hand, the potential flow method is computationally inexpensive; evaluation of each simulation using the partial physics model costs no more than ≈10 seconds – significantly lower than high-fidelity simulation (≈16 hours). The potential flow method is widely have been applied in airfoil and aircraft designs for various applications [44, 45]. These characteristics inspired us to develop a physics-aware hybrid model using a potential flow analysis. Figure 6 shows the overall pipeline for evaluating a single design variable (h, s, σ) using both full physics model (RANS CFD simulation) and our proposed physics-aware hybrid model. As it can be seen from this figure, the hybrid model can be divided into three components, namely: 1) transfer model; 2) airfoil generator (as the first part of the partial physics model); and 3) potential flow tool (the second component of the partial physics model). These components are described in details in next.

Partial Physics Model: In this paper, we create a series of potential flow based airfoil models as the partial physics models for the riblet design problem. The airfoils as the partial physics are smooth NACA 4 digit airfoils (to align with the airfoil characteristics of the baseline airfoil (NACA0012) with variable thickness and curvatures. By adjusting the thickness and curvature of the partial physics airfoil model, plus the flow conditions (Mach number, Reynolds number, and angle of attack), we can mimic the flow characteristics (namely pressure distribution, lift or drag) of the airfoil with riblets.

As shown in Fig. 6, the partial physics model contains two main components and it has 2 transferred inputs: Reynolds number (N_{Re}), and angle of attack (α). The first component is a custom code that generates the airfoil coordinates ([46, 47]) with given thickness and curvature (transfer model's outputs) to be imported into the partial flow tool (i.e., XFOIL). XFOIL [48] is one of the available software tools for performing the potential flow based analysis. Our past experiences [49] suggest that XFOIL gives reliable estimations of lift and drag under low Reynolds number. XFOIL software receives the airfoil coordinates, Mach number, Reynolds number, and angle of attack as input and generates the drag (C_d) coefficient. An example output of XFOIL is depicted in Fig. 7. The transferred inputs are generated by the transfer model, which will be discussed next.

Transfer Model: For this paper, transfer model is defined as two Gaussian process models with three inputs and one output (Fig. 6). The inputs are the original inputs ($\mathbf{x}$): height (h), spacing (s), and width (σ). The outputs of the transfer model produce the transferred inputs $\mathbf{z}$. In this paper, we use a supervised learning approach to learn the transfer model ($M_{\text{transfer}} : \mathbf{x} \rightarrow \mathbf{z}$) because of the colossal difference of the computation time of the Partial Physics and the high fidelity full physics. Here, we identify transfer (intermediate) inputs ($\mathbf{z}$) of the actual partial physics equivalent model through optimization process based on the given original inputs ($\mathbf{x} = [h, s, \sigma]$). These optimized transfer inputs can be used for supervised training of a model, a Gaussian process here, in the next step. For this purpose, we solve the optimization problem explained in below:

$$
\underset{\mathbf{z}=\{N_{\text{Re}},\alpha\}}{\text{minimize}} \quad f = \|(C_d^{\text{FP}}(\mathbf{x}) - C_d^{\text{PP}}(\mathbf{z})\| / C_d^{\text{Barefoil}}
\tag{5}
$$

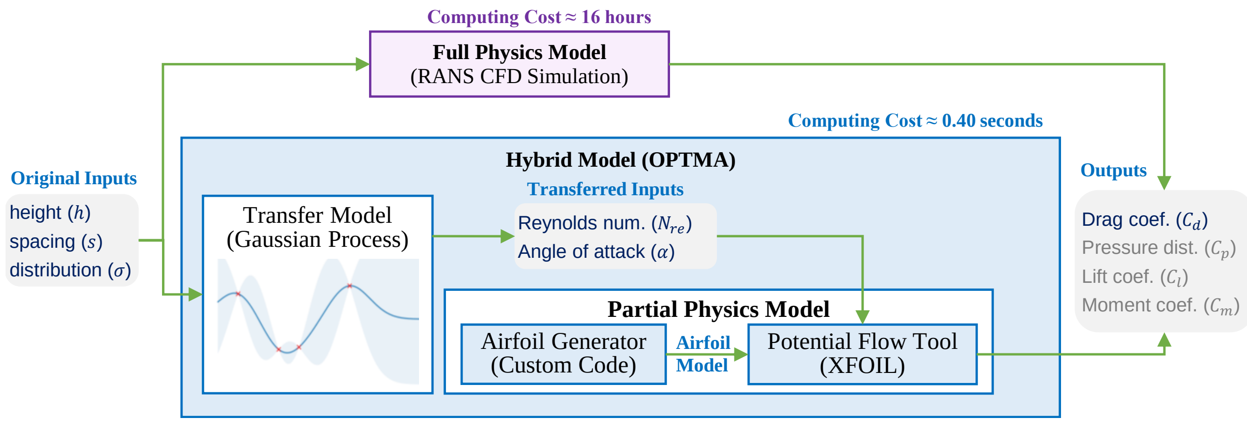


Fig. 6 Physics-aware hybrid model vs. full physics model for riblet design problem. The physics-aware hybrid model is significantly faster than the full physics model, ≈0.40 seconds vs. ≈16 hours

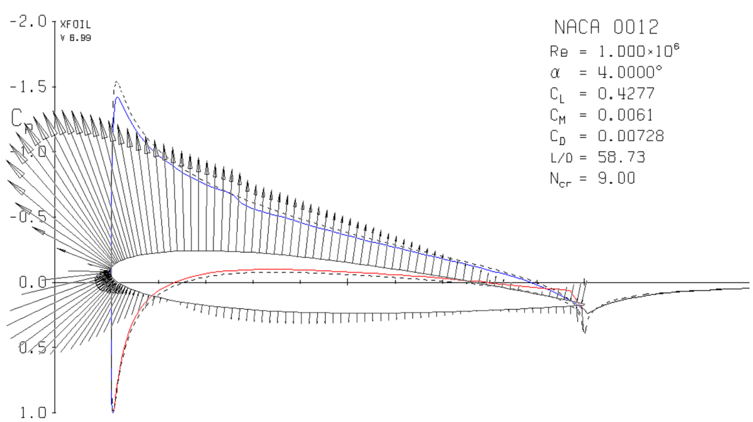


Fig. 7 An example output of XFOIL: estimating the pressure distribution (C_p).

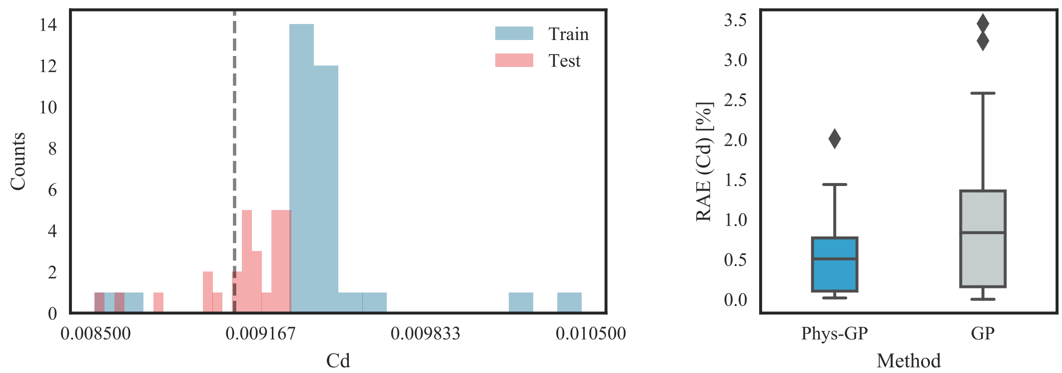
After obtaining the optimal transfer inputs ($\mathbf{z}_i$) for i-th training sample ($\mathbf{z}_i$), we build an intermediate dataset ($\mathcal{D}_{\text{transfer}} = \cup_{i=\{1..N_{\text{train}}\}}\{\mathbf{x}, \mathbf{z}\}$). Then, a GP model ($M_{\text{transfer}}$) is trained using this dataset.

IV. Results and Discussion

A. Results of Modeling

The feasible space of Gaussian riblets are bound by three linear constraints imposed to facilitate favorable flow behavior and prevent geometric conflicts between consecutive riblets. We generate 57 samples only within the feasible region, by adopting an approach called Latin Hypercube Sampling with Inequality Constraints (LHS-IC) [50]. In order to measure the generalizablity power of each modeling technique (i.e., pure data-driven modeling (pure GP) vs. physics-aware hybrid modeling (Phys-GP)), we split the dataset into two training and test dataset such that they have different distributions (Fig. 8(a)). The number of sample points used for training and testing the surrogate models is specified to be $N_{\text{train}} = 32$ and $N_{\text{test}} = 25$, respectively. Here, we are using a GP model with Gaussian kernel for both GP and Phys-GP.

Figure 8(b) demonstrates the model error across test samples (unseen data) in terms of relative absolute error ($\text{RAE} = 100|(C_d^{\text{actual}} - C_d^{\text{predict}})/C_d^{\text{actual}}|$. It can be seen from this figure that the proposed Physics-aware hybrid model (Phys-GP) outperforms the pure data-driven model (GP) in terms of both the median and the variance; the median and maximum errors of the Phys-GP, respectively, are 0.5% and 2% (roughly two times more accurate than the pure data-driven GP method). These results demonstrate the generalizability power of the hybrid model. It should be mentioned that the hybrid model obtains this higher fidelity while keeping its computational cost tractable ($< 0.40s$).



(a) Distribution of training and test dataset

(b) Prediction error (test dataset)

Fig. 8 Distribution of training and test dataset and the prediction error of the surrogate models using the test dataset.

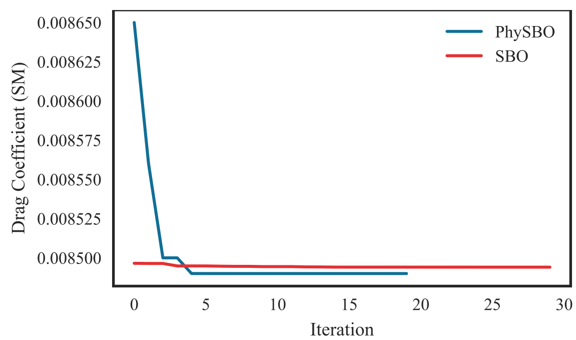


Fig. 9 The convergence history of the optimization processes for the two methods (PhySBO and SBO) with $\alpha = 2°$.

B. Results of Optimization

In this paper, we are using an advanced variation of PSO (MDPSO [31]) with the following settings for both our proposed PhySBO and the standard SBO (SBO) approaches: a population size of PSO is set at $N_{\text{pop}} = 90$, and the maximum iteration is set at $N_{\text{MaxIter}} = 100$. Figure 9 shows the convergence history of the standard SBO and our proposed Phys-SBO approaches for 2° angle of attack. This plot shows the variation in the objective function (drag coefficient) across iterations. In the primary Phys-SBO method, the drag coefficient reduced from 0.008650 to 0.008490 (based on surrogate model evaluations) over the optimization process. This variation is negligible for the SBO case (reduced from 0.008497 to 0.008494).

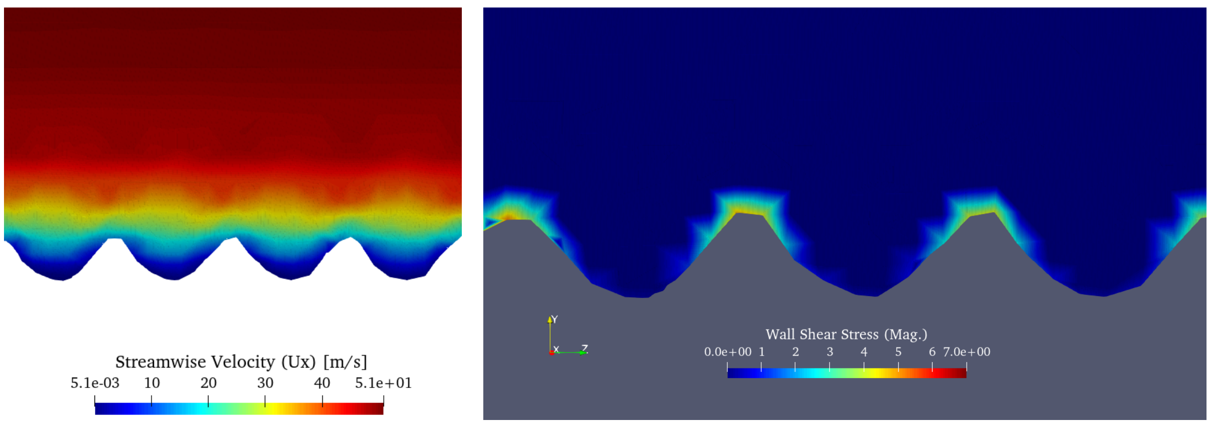
Table 1 summarizes the optimization results, the optimal riblet shapes, and the following information: the drag coefficient values corresponding to the optimized riblet shapes as given by the surrogate model and the high-fidelity CFD model (and associated model error); and the % reduction in drag obtained compared to the bare 3D airfoil, expressed as $RDC = 100(C_{d,\text{Barefoil}} - C_{d,\text{Riblet}})/C_{d,\text{Barefoil}}$ (both values estimated using the high-fidelity CFD model). It can be seen from Table 1 that the proposed Phys-SBO method found slightly better optimum with smaller prediction error at optimum for the drag coefficient. The optimum riblet provides a 6.1% relative drag reduction at the angle of attack 2°.

C. Flow Characteristics of the Optimum Design

In this section, we provide further analysis of the flow phenomena observed with the optimum riblet shape obtained in the previous section. To understand the flow physics of how riblets impede the cross-stream translation of the streamwise vortices, we visualize the flow near the airfoil surface. It can be seen from the near-wall contours of the streamwise velocity (shown in Fig. 10(a)) that the protrusion of the riblets from the surface into the boundary layer creates zones of high-speed and low-speed fluid regions. The low-speed fluid region comes in contact with the riblet

Table 1 Results of optimum riblet design with $\alpha = 2°$; N_{iter}: Number of Iterations to Converge; h: Optimum Height in μm; s: Optimum Spacing in μm; σ: Optimum Width in μm; RAE: Relative Absolute Error in Surrogate Estimated Optimum C_d value; RDC: Reduction in Drag Coefficient compared to airfoil without riblets.

Case	N_{iter}	h	s	σ	$C^*_{d,\text{SM}}$	$C^*_{d,\text{CFD}}$	Error (RAE) [%]	Drag Reduction (RDC) [%]
SBO	35	500.3	970.0	161.3	0.008494	0.008517	0.27	6.0
PhySBO	20	495.4	976.0	159.4	0.008490	0.008510	**0.23**	**6.1**



(a) Streamwise velocity (m/s)

(b) Shear stress (N/m2)

Fig. 10 Contour plots of streamwise velocity and shear stress at $x/c = 0.3$ for the optimal design (obtained by PhySBO) at $\alpha = 2°$.

valleys which constitute a higher portion of the surface area whereas the high-speed fluid comes in contact only with the riblet peaks.

In the ribleted airfoil, there exist different velocity gradients, attributed to low speed and high speed fluid regions attributed to the surface riblets. The velocity gradients can be quantified here by measuring the shear stress (Fig. 10(b)). Riblets demarcate the fluid into low speed and high speed regions, where the shear stress is simultaneously redistributed. In this study, the riblets are aligned with the free-stream velocity and hence, a shift in the wall shear stress distribution can be seen in the spanwise direction. Figure 10(b) shows how the shear stress varies around the riblet peaks and valleys; while a small decrease in boundary layer thickness (Fig. 10(a)) and associated increase in shear stress Fig. 10(b) is observed in the riblet peaks (compared to the bare airfoil), the riblet valleys realize a more dominant increase in boundary layer thickness and associated lowering of shear stress, possibly contributing to the overall reduction in C_d.

V. Concluding Remarks

In this paper, we proposed a novel physics-aware surrogate-based optimization method, the so-called PhySBO, for optimizing complex system in an efficient and reliable manner. The efficiency and the reliablity of the surrogate-based optimization methods are strongly relied on the reliability of the surrogate model that used in the optimization process. The pure data-driven surrogate models can perform poorly if the training dataset (seen samples) and the test dataset (unseen samples) are from different distribution. It is quite often in real-world settings that the training and test samples do not come from the same distributions. One way to address this issue is using physics-aware hybrid techniques that show promising generalization and extrapolation. For this purpose, we integrated a physics-aware surrogate modeling architecture (based on Gaussian process model) into a variation of the particle-swarm optimization (PSO) method. The effectiveness of the proposed approach was evaluated by applying it to a ribleted airfoil design problem. The proposed framework for the ribleted airfoil design incorporated the following components: an automated open-source CFD framework (as full physics model) for generating high-fidelity samples, a GP with Gaussian kernel (as transfer model), a potential flow method (as partial physics model), and a particle-swarm optimization (as optimizer). The results proved the proposed PhySBO method is able to find the optimum design with better performance than a traditional SBO (using pure data-driven model) with higher accuracy. The proposed method also is capable to reduce each flow simulation from 7-16 hours to < 0.40 seconds (provide more than 10^5 times computing time reduction); the time efficiency achieved is readily evident. A drag reduction, e.g., 6.1% when tailoring the height/spacing/width of riblets simultaneously, is accomplished compared to that of a bare 3D airfoil.

Gaussian process models are favored since they provide a measure of uncertainty for their predictions, which is critical for robust optimization. When the GP models are combined with other non-GP models, the uncertainty measurement is not necessarily trivial when the GP model is embedded within a hybrid architecture. Thus, an important direction of future work would be to extend our hybrid model such that it can give a tractable uncertainty measurement.

Acknowledgments

Support from the DARPA Award HR00111890037 from Physics of AI (PAI) program is gratefully acknowledged.

References

[1] Simpson, T., Toropov, V., Balabanov, V., and Viana, F., "Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come-or not," *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2008, p. 5802.

[2] Jin, Y., "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, Vol. 1, No. 2, 2011, pp. 61–70.

[3] Fernández-Godino, M. G., Park, C., Kim, N.-H., and Haftka, R. T., "Review of multi-fidelity models," *arXiv preprint arXiv:1609.07196*, 2016.

[4] Ghassemi, P., Mehmani, A., and Chowdhury, S., "Adaptive in situ model refinement for surrogate-augmented population-based optimization," *Structural and Multidisciplinary Optimization*, 2020. https://doi.org/10.1007/s00158-020-02592-6, URL https://doi.org/10.1007/s00158-020-02592-6.

[5] Haley, P. J., and Soloway, D., "Extrapolation limitations of multilayer feedforward neural networks," *Neural Networks, 1992. IJCNN., International Joint Conference on*, Vol. 4, IEEE, 1992, pp. 25–30.

[6] Neal, R. M., *Bayesian learning for neural networks*, Vol. 118, Springer Science & Business Media, 2012.

[7] Solomatine, D., See, L. M., and Abrahart, R., "Data-driven modelling: concepts, approaches and experiences," *Practical hydroinformatics*, Springer, 2009, pp. 17–30.

[8] Atamuradov, V., Medjaher, K., Dersin, P., Lamoureux, B., and Zerhouni, N., "Prognostics and health management for maintenance practitioners-Review, implementation and tools evaluation," *International Journal of Prognostics and Health Management*, Vol. 8, No. 060, 2017, pp. 1–31.

[9] Mehmani, A., Chowdhury, S., Tong, W., and Messac, A., "Adaptive switching of variable-fidelity models in population-based optimization," *Engineering and Applied Sciences Optimization*, Springer, 2015, pp. 175–205.

[10] Chowdhury, S., Tong, W., Messac, A., and Zhang, J., "A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation," *Structural and Multidisciplinary Optimization*, Vol. 47, No. 3, 2013, pp. 367–388.

[11] Kleijnen, J. P., "A comment on Blanning's "Metamodel for sensitivity analysis: the regression metamodel in simulation"," *Interfaces*, Vol. 5, No. 3, 1975, pp. 21–23.

[12] Jin, R., Chen, W., and Simpson, T. W., "Comparative studies of metamodelling techniques under multiple modelling criteria," *Structural and multidisciplinary optimization*, Vol. 23, No. 1, 2001, pp. 1–13.

[13] Simpson, T., Korte, J., Mauery, T., and Mistree, F., "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, Vol. 39, No. 12, 2001, pp. 2233–2241.

[14] Forrester, A., and Keane, A., "Recent Advances in Surrogate-based Optimization," *Progress in Aerospace Sciences*, Vol. 45, No. 1-3, 2009, pp. 50–79.

[15] Choi, K., Youn, B. D., and Yang, R.-J., "Moving least square method for reliability-based design optimization," *Proc. 4th World Cong. Structural & Multidisciplinary Optimization*, 2001.

[16] Toropov, V. V., Schramm, U., Sahai, A., Jones, R. D., and Zeguer, T., "Design optimization and stochastic analysis based on the moving least squares method," *6th World Congresses of Structural and Multidisciplinary Optimization*, 2005.

[17] Hardy, R. L., "Multiquadric Equations of Topography and Other Irregular Surfaces," *Journal of Geophysical Research*, Vol. 76, 1971, pp. 1905–1915.

[18] Clarke, S. M., Griebsch, J. H., and Simpson, T. W., "Analysis of support vector regression for approximation of complex engineering analyses," *Journal of Mechanical Design*, 2005.

[19] Yegnanarayana, B., *Artificial Neural Networks*, PHI Learning Pvt. Ltd., 2004.

[20] An, D., Kim, N. H., and Choi, J.-H., "Practical options for selecting data-driven or physics-based prognostics algorithms with reviews," *Reliability Engineering & System Safety*, Vol. 133, 2015, pp. 223–236.

[21] Artun, E., "Characterizing interwell connectivity in waterflooded reservoirs using data-driven and reduced-physics models: a comparative study," *Neural Computing and Applications*, Vol. 28, No. 7, 2017, pp. 1729–1743.

[22] Ghassemi, P., Zhu, K., and Chowdhury, S., "Optimal surrogate and neural network modeling for day-ahead forecasting of the hourly energy consumption of university buildings," *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2017, pp. V02BT03A026–V02BT03A026.

[23] Javed, K., "A robust & reliable Data-driven prognostics approach based on extreme learning machine and fuzzy clustering." Ph.D. thesis, Université de Franche-Comté, 2014.

[24] Cheng, S., and Pecht, M., "A fusion prognostics method for remaining useful life prediction of electronic products," *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on*, IEEE, 2009, pp. 102–107.

[25] Karpatne, A., Watkins, W., Read, J., and Kumar, V., "Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling," *arXiv preprint arXiv:1710.11431*, 2017.

[26] Narendra, K. S., and Parthasarathy, K., "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, Vol. 1, No. 1, 1990, pp. 4–27.

[27] Young, C.-C., Liu, W.-C., and Wu, M.-C., "A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events," *Applied Soft Computing*, Vol. 53, 2017, pp. 205–216.

[28] Nourani, V., Alami, M. T., and Aminfar, M. H., "A combined neural-wavelet model for prediction of Ligvanchai watershed precipitation," *Engineering Applications of Artificial Intelligence*, Vol. 22, No. 3, 2009, pp. 466–472.

[29] McKay, M., Conover, W., and Beckman, R., "A comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[30] Friedman, J. H., Bentley, J. L., and Finkel, R. A., "An algorithm for finding best matches in logarithmic time," *ACM Trans. Math. Software*, Vol. 3, No. SLAC-PUB-1549-REV. 2, 1976, pp. 209–226.

[31] Chowdhury, S., Tong, W., Messac, A., and Zhang, J., "A mixed-discrete Particle Swarm Optimization algorithm with explicit diversity-preservation," *Structural and Multidisciplinary Optimization*, Vol. 47, No. 3, 2013, pp. 367–388.

[32] Kennedy, J., and Eberhart, R. C., "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, IEEE, Piscataway, NJ, USA, 1995, pp. 1942–1948.

[33] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., "Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions," *Renewable Energy*, Vol. 52, 2013, pp. 273–282.

[34] Chowdhury, S., Maldonado, V., Messac, A., and Tong, W., "A New Modular Product Platform Planning Approach to Design Macro-scale Reconfigurable Unmanned Aerial Vehicles (UAVs)," *AIAA Journal of Aircraft*, Vol. 53, No. 2, 2016, pp. 309–322.

[35] Lulekar, S., Ghassemi, P., and Chowdhury, S., "CFD-based Analysis and Surrogate-based Optimization of Bio-inspired Surface Riblets for Aerodynamic Efficiency," *2018 Multidisciplinary Analysis and Optimization Conference*, 2018, p. 3107.

[36] Ghassemi, P., Lulekar, S. S., and Chowdhury, S., “Adaptive Model Refinement with Batch Bayesian Sampling for Optimization of Bio-inspired Flow Tailoring,” *AIAA Aviation 2019 Forum*, 2019, p. 2983.

[37] Kennedy, J., Hsu, S., and Lin, J., “Turbulent flow past boundaries with small longitudinal fins,” *J. Hydraulic Division*, Vol. 99, No. 3, 1973.

[38] Boomsma, A., “Drag Reduction by Riblets & Sharkskin Denticles: A Numerical Study,” 2015.

[39] Bechert, D., Bruse, M., and Hage, W., “Experiments with three-dimensional riblets as an idealized model of shark skin,” *Experiments in fluids*, Vol. 28, No. 5, 2000, pp. 403–412.

[40] Choi, K.-S., “Turbulent drag-reduction mechanisms: strategies for turbulence management,” *Turbulence Structure and Modulation*, Springer, 2001, pp. 161–212.

[41] Walsh, M. J., “Riblets as a viscous drag reduction technique,” *AIAA journal*, Vol. 21, No. 4, 1983, pp. 485–486.

[42] Wilkinson, S., and Lazos, B., “Direct drag and hot-wire measurements on thin-element riblet arrays,” *Turbulence Management and Relaminarisation*, Springer, 1988, pp. 121–131.

[43] Lee, S.-J., and Lee, S.-H., “Flow field analysis of a turbulent boundary layer over a riblet surface,” *Experiments in fluids*, Vol. 30, No. 2, 2001, pp. 153–166.

[44] Giesing, J. P., “Nonlinear two-dimensional unsteady potential flow with lift.” *Journal of Aircraft*, Vol. 5, No. 2, 1968, pp. 135–143.

[45] Steinhoff, J., and Jameson, A., “Multiple solutions of the transonic potential flow equation,” *AIAA Journal*, Vol. 20, No. 11, 1982, pp. 1521–1525.

[46] Moran, J., *An introduction to theoretical and computational aerodynamics*, Courier Corporation, 2003.

[47] Leishman, G. J., *Principles of helicopter aerodynamics with CD extra*, Cambridge university press, 2006.

[48] Drela, M., “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,” *Low Reynolds Number Aerodynamics*, edited by T. J. Mueller, Springer Berlin Heidelberg, Berlin, Heidelberg, 1989, pp. 1–12.

[49] Zeng, C., Abnous, R., and Chowdhury, S., “Aerodynamic Modeling and Optimization of a Blended-Wing-Body Transitioning UAV,” *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017. https://doi.org/10.2514/6.2017-4000.

[50] Petelet, M., Iooss, B., Asserin, O., and Loredo, A., “Latin hypercube sampling with inequality constraints,” *AStA Advances in Statistical Analysis*, Vol. 94, No. 4, 2010, pp. 325–339.

[51] Menter, F. R., “Two-equation eddy-viscosity turbulence models for engineering applications,” *AIAA journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.

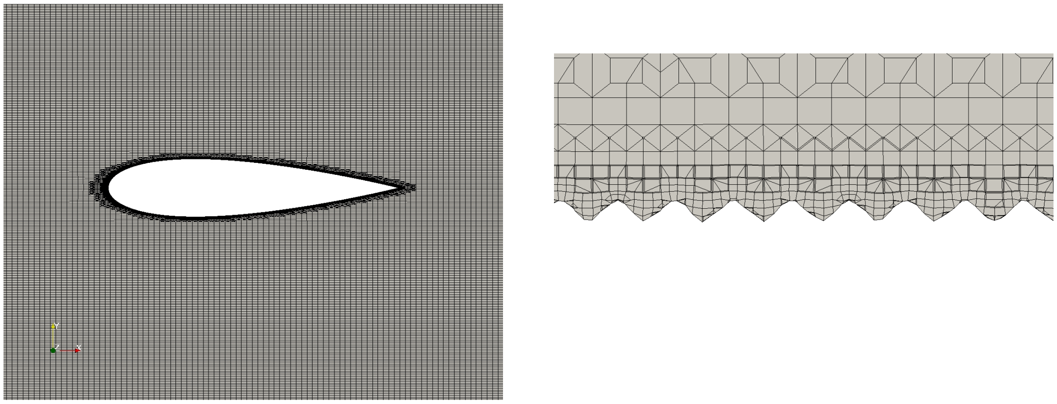
Appendix

Automated Riblet Design Framework: Full Phyiscs Model

Geometry and Meshing

A finite span of the NACA0012 airfoil is considered with riblets placed on the top and bottom surface of the airfoil. The airfoil contains a constant chord with an aspect ratio of 2. The chord length is 0.127m (5 inches) and is visualized in Fig. 5 along with the riblets. The riblets are a continuous extraction of a 2D Gaussian curve, where the riblet valleys and the peaks are aligned in the flow direction 5. Three-dimensional CAD model of symmetric NACA0012 airfoil is build using the `SALOME-8.3.0` and imported as an unstructured triangulated surface (.stl) with a precision of $O(10^{-2})$ mm. The surface mesh is imported for volumetric grid generation carried out using `SnappyHexMesh`. To promote a greater stability with reasonable mesh count, while capturing the boundary layer profile more accurately, hex cells are preferred. Ten inflation layers are used with a growth rate of 1.2 and the smallest cell height is determined from $y^+ \in [5, 10]$.

Figure 12 shows the bounding domain, which is defined sufficiently large to avoid choke flow, as well as to have minimal effects on the airfoil. To reduce the computational cost of the analysis, a symmetry plane is used and hence all the results are visualized on a half wing. The mesh generated in the domain approximately consists of 8 million cells. Fig. 11(a) shows the prism layer around the NACA0012 airfoil and a sliced section of the mesh accurately capturing the Gaussian riblet curve can be seen in Fig. 11(b).



(a) The prism Layers around the NACA0012 airfoil. (b) The mesh around the Gaussian riblets.

Fig. 11 The prism layers and mesh around the airfoil and the Gaussian riblets.

Boundary Conditions

The CFD simulations are carried out in a bounded box with a 2D illustration shown in Fig 12. The domain has a velocity inlet (`free-stream`) with a constant velocity input. At the outlet, `pressure outlet` for the pressure is provided and `zeroGradient` for the velocities. At the top, bottom and side wall we have `slip` condition for the velocities and `zeroGradient` for the pressure. A `symmetry plane` boundary condition is used to reduce the computational time, and `noSlip` boundary conditions is applied on the airfoil. The flow being transitional region, a 3.2% of turbulent intensity is given at the inlet. The turbulent kinetic energy (TKE) and the omega (ω) at the inlet are calculated using the following equations:

$$I = 0.16 N_{\mathrm{Re}}^{-1/8} \tag{6}$$

$$k = 1.5(UI)^2 \tag{7}$$

$$\omega = 0.9^{-1/4}\sqrt{k/l} \tag{8}$$

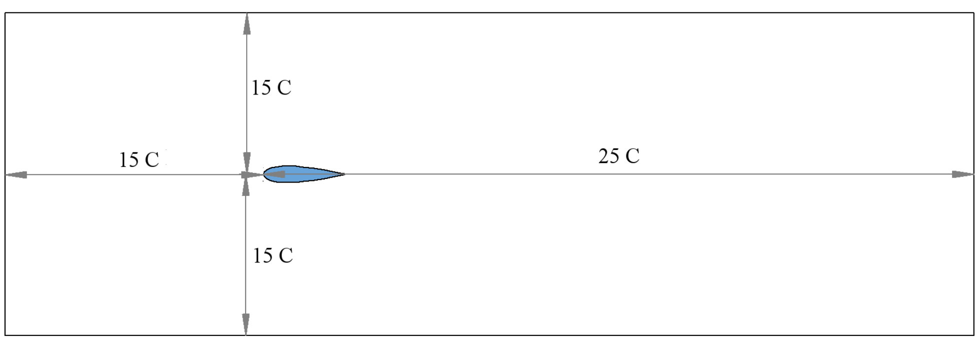


Fig. 12 The bounding domain.

Solver Settings

An open source tool, `OpenFOAM`, is used to perform the CFD simulations. A pressure-based solver is used with $k-\omega$ SST [51] turbulence models for wall bounded wall flow. The incoming $N_{\mathrm{Re}} = 3.38x10^5$, is calculated with a chord as the characteristic length, which is in the transitional ranger from laminar to turbulent. The fluid considered here is air, and treated as incompressible, as the Mach number is less than 0.4. The pressure and velocity are coupled using the SIMPLE scheme. For spatial discretization and to calculate gradient of velocity (∇U), linear Upwind scheme is used. As stated above, the whole process is automated to evaluate multiple designs, Simulations are carried out with a termination criteria of $O(10^{-6})$. To shorten the computation time, *MPI* library is used to parallelize the whole computation, and the simulations are performed using the distributed computing cluster (CCR) at the University of Buffalo. The primary quantities of interest obtained and derived from the CFD simulations are the shear stress data and co-efficient of drag.