Multihierarchy Gaussian Process Models for Probabilistic Aerodynamic Databases using Uncertain Nominal and Off-Nominal Configuration Data

James B. Scoggins*, T. J. Wignall †, Tenavi Nakamura-Zimmerer ‡, and Karen Bibb §
NASA Langley Research Center, Hampton, VA 23681

Probabilistic aerodynamic databases are a crucial component of the development lifecycle for aerospace vehicles. A key challenge when building aerodynamic databases is that most data used to construct them represent various simplifications of the real flight vehicle. For example, wind tunnel models often simplify the vehicle geometry and surface roughness characteristics, while CFD computations often make simplifications to the physics being modeled, such as fully laminar or turbulent calculations. Standard multifidelity data fusion models rely on a user being able to define a hierarchy of fidelity levels anchored to some "truth" data. This approach is unsatisfactory when no data can be considered to accurately reflect real flight conditions. In this work, we provide an alternative approach by presenting a consistent mathematical framework for building probabilistic aerodynamic databases in the form of a conditional probability distribution described by an ensemble of multifidelity Gaussian Processes. Instead of relying on a single hierarchy of data fidelity levels, the presented framework identifies a "nominal" configuration and potential corrections to the nominal which represent specific physical phenomena not represented in the nominal data. The nominal and correction functions themselves are constructed as multifidelity Gaussian Processes and linearly combined to form an ensemble model which fuses the uncertainties associated nominal and correction models. Results obtained using the proposed framework on a simplified Orion Crew Module wind tunnel dataset demonstrate the predictive capability of the multihierarchy framework. We further demonstrate the benefits of such a probabilistic aerodynamic database approach through function sampling and computing the conditional distributions of derived quantities, such as the trim angle of attack and aerodynamic coefficients at trim.

Nomenclature

AR1	=	First-order Autoregressive
ARD	=	Automatic Relevance Determining
CFD	=	Computational Fluid Dynamics
CM	=	Crew Module
DOF	=	Degrees of Freedom
EDL	=	Entry, Descent, and Landing
GP	=	Gaussian Process
GPR	=	Gaussian Process Regression
IDAT	=	Integrated Design Assessment Team
LAT	=	Launch Abort Tower
MPCV	=	Multi-Purpose Crew Vehicle
OML	=	Outer Mold Line
SE	=	Squared Exponential
SPD	=	Symmetric Positive Definite

*Research Aerospace Engineer, Aerothermodynamics Branch, Member AIAA
†Aerospace Engineer, Configuration Aerodynamics Branch
‡Research Aerospace Engineer, Flight Dynamics Branch
§Research Engineer, Aerothermodynamics Branch, Senior Member AIAA

TPS	=	Thermal Protection System
C_{A}	=	axial force coefficient
C_{D}	=	drag force coefficient
C_{L}	=	lift force coefficient
C_{N}	=	normal force coefficient
C_{Y}	=	side force coefficient
$C_{\mathrm{m,cg}}$	=	pitching moment coefficient
C_{n}	=	yawing moment coefficient
d	=	input dimension
D	=	vehicle diameter
Δz_{cg}	=	vertical offset of vehicle center of gravity from symmetry centerline
$\boldsymbol{I_n}$	=	$n \times n$ identity matrix
l_k	=	kernel length-scale for dimension k
M_∞	=	freestream Mach number
$p(a\|b)$	=	conditional probability of a given b
Re_D	=	Reynolds number based on vehicle diameter
w_i	=	correction weight associated with correction model i
$\boldsymbol{x}$	=	input vector of dimension d
$\boldsymbol{X}$	=	real matrix of n input vectors with size $n \times d$
$\boldsymbol{y}$	=	target (output) vector corresponding to n inputs
$\mathcal{GP}(\mu, k)$	=	Gaussian process with mean $\mu(\boldsymbol{x})$ and covariance kernel $k(\boldsymbol{x}, \boldsymbol{x}')$
$\mathcal{N}(m, K)$	=	Normal distribution with mean m and covariance matrix K
α	=	angle of attack, degrees
α_T	=	total angle of attack, degrees
β	=	sideslip angle, degrees
$\boldsymbol{\epsilon}$	=	random observation noise vector
ϕ	=	roll angle, degrees
ρ_i	=	correlation coefficient associated with fidelity levels i and $i-1$
σ^2	=	kernel variance
σ_n^2	=	observation noise variance
$\boldsymbol{\theta}$	=	kernel function hyperparameters
$\hat{f}$	=	denotes posterior distribution of f
$\otimes$	=	Kronecker product

I. Introduction

During the development lifecycle of an atmospheric entry vehicle, aerodynamic databases (aerodatabases) are required to provide aerodynamic data for 6-DOF trajectory simulations, stability analysis, and aerodynamic loading predictions. Trajectory simulations are crucial for developing guidance, navigation, and control systems for the vehicle, as well as to provide targeting and landing ellipse prediction during the vehicle's flight operations, while the stability and structural loading analyses are used to ensure a safe and reliable design of the vehicle. Typically, the aerodatabase is constructed as a tabulated set of static and dynamic aerodynamic coefficients from numerous computational and experimental sources, which aim to provide reliable data over the entire trajectory space of interest [1–3]. Uncertainty estimates for each aerodynamic coefficient in the database are generated by combining multiple sources of uncertainty in the data, including turbulence modeling, grid resolution, wind tunnel repeatability, roughness effects, and other physical phenomena.

This process is complicated by the fact that a typical vehicle operates under multiple configurations (and potentially in multiple atmospheres) during a single mission. For example, the Orion Multi-Purpose Crew Vehicle (MPCV), depicted in Fig. 1, consists of a Crew Module (CM), Service Module, Spacecraft Adapter, and Launch Abort Tower (LAT). During a normal Entry, Descent, and Landing (EDL) sequence, the CM is decelerated from hypersonic speeds before releasing its parachutes, while under a launch abort scenario, the CM and LAT are joined in the initial acceleration phase before the LAT is jettisoned, and the CM flies for a short period before releasing its parachutes. Each phase of each scenario requires a specific aerodatabase, comprised of static aerodynamic force and moment coefficients over the

trajectory space associated with each phase.

In addition to configuration changes during a vehicle's flight operations, numerous configuration changes may occur during the design cycle of the vehicle as the design analysis matures. This results in multiple computations or wind tunnel tests conducted with varying degrees of fidelity relative to the final vehicle geometry and hardware. For example, an initial design of a vehicle may call for an axisymmetric outer mold line (OML), which is later updated to be asymmetric due to various optimizations or constraints on the design. Wind tunnel data taken early in the design cycle will likely use the symmetric vehicle geometry. As the design matures, additional tests may be deemed necessary to account for the change in geometry. In a perfect world, all testing would be repeated with the vehicle's final designed geometry as closely reproduced as possible, however project budget and time constraints often restrict the scope of additional testing. In addition, a (perhaps unique) challenge for entry vehicles with ablative thermal protection systems (TPSs) is that their shapes and surface roughnesses actually change slightly during flight, due to uneven recession of the TPS surface. For these reasons, engineers often must rely on the data for slightly dissimilar geometries and make engineering assumptions based on limited updates to these data to design aerodatabases. There is therefore a strong need to develop methodologies to efficiently and robustly combine dissimilar datasets, including various sources of uncertainty and differing levels of fidelity with respect to their representation of actual flown hardware.

Multifidelity Gaussian Processes (GPs) are one class of methodologies which have been proposed to solve the data fusion problem. A GP is most easily described as a distribution over functions, where for any finite number of inputs, the function values at those inputs have a joint Gaussian distribution [4]. Multifidelity GPs are used to describe a joint Gaussian distribution over multiple predictors of a single function, each with varying degrees of fidelity. They were first introduced by Kennedy and O'Hagan [5] for modeling complex computer codes, given abundant fast approximations of the code output and sparse evaluations of the high-fidelity model. The main idea of Kennedy and O'Hagan is that each model can be ranked by fidelity level with respect to the true model with an autoregressive (AR1) dependency linking successive levels of fidelity. Le Gratiet and Garnier [6] later introduced a recursive AR1 model, which provides significant performance improvements when data from each fidelity level is taken from a subset of the preceding level's dataset. The AR1 method assumes a linear relationship between outputs of each fidelity level. More recently, models which generalize to nonlinear relationships have also been introduced such as the Nonlinear Autoregressive GP (NARGP) [7] and Multifidelity Deep Gaussian Processes (MF-DGPs) [8, 9]. For a review of multifidelity GP models, the reader is referred to the recent review of Brevault et al. [10]. Other surveys of multifidelity methods in general are provided in Refs. [11, 12]. Within the aerospace community, applications of multifidelity GPs are predominantly based on the linear coupled and recursive AR1 variants [13–19].

Note that all the above methods assume that there is a single hierarchy of fidelities with one truth dataset that is often sparsely populated, which must be approximated. In reality, no wind tunnel or CFD data will exactly reproduce combined flight conditions and vehicle geometry across the entire flight trajectory. In this work, we propose an alternate approach in which the aerodatabase is built from an ensemble of multifidelity GPs, each representing corrections to a nominal GP model. This allows engineers to make maximum use of the data available, while encoding expert judgment into the prior belief of the relationship between each source of data. The proposed methodology has several important benefits over conventional engineering approaches and other learning-based methods. First, uncertainties remain traceable to specific off-nominal phenomena, inline with the same spirit as current database construction. Second, the

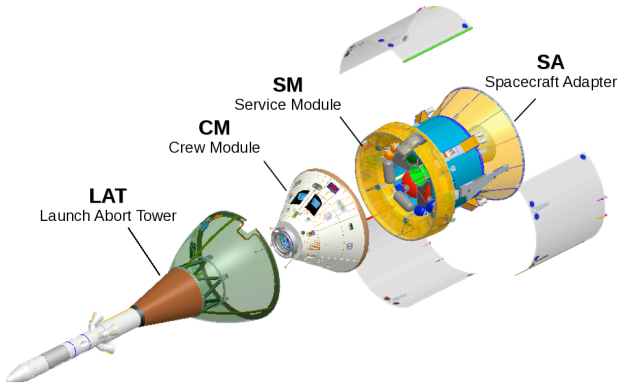


Fig. 1 Orion Multi-Purpose Crew Vehicle components [2].

framework treats observation noise and uncertainties associated with each correction function consistently and does not rely on pragmatic engineering judgments to layer multiple sources of uncertainty. In addition, the resulting probabilistic aerodatabase can be used during the design phase and updated based on flight data, as it becomes available.

The paper is organized as follows. In Sec. II, the multihierarchy GP regression model is introduced, beginning with a brief description of single and multi fidelity GP regression techniques. Next, Sec. III introduces the Orion spacecraft to motivate the proposed methodology with a concrete example. The datasets used to train the various GP models are detailed, based on recent wind tunnel testing on the Orion CM. In Sec. IV, we present results obtained with the proposed methodology, applied to the Orion dataset described in Sec. III, and demonstrate some benefits of using a probabilistic aerodatabase in this way. Finally, key conclusions are presented in Sec. V.

II. Methodology

In this section, we detail the multihierarchy Gaussian Process methodology developed in this work. We begin with a brief overview of Gaussian Processes (GPs) in general and the original multifidelity GP model proposed by Kennedy and O'Hagan [5]. This is not meant to be an exhaustive overview, rather to provide a simple introduction for readers without previous knowledge of GPs and to highlight the differences in our approach, compared to these classical approaches. For an in-depth introduction to GPs, the reader is referred to the excellent book by Rasmussen and Williams [4].

A. Gaussian Process Regression

Gaussian processes can be thought of as a generalization of the Gaussian distribution for random scalars to random functions. Specifically, they should be thought of as a prior over real-valued random functions of the form $f : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^d$ is the d-dimensional input space. For any finite set of inputs $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\} \subset \mathcal{X}$, the random function f is said to be a GP, parameterized by mean function $\mu : \mathcal{X} \mapsto \mathbb{R}$ and kernel function $k(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, if the random vector $\boldsymbol{f}(X) = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_m)]^T \in \mathbb{R}^m$ is a multivariate Gaussian, with mean $\boldsymbol{\mu}(X) = [\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_m)]^T \in \mathbb{R}^m$ and covariance matrix $\boldsymbol{K} \in \mathbb{R}^{m \times m}$, where the i, j^{th} entry of $\boldsymbol{K}$ is $\text{cov}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. We denote this as

$$f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')). \tag{1}$$

In order to generate valid covariance matrices, the kernel function $k(\cdot,\cdot)$ must satisfy the property that, for any finite set of inputs, the resulting covariance matrix $\boldsymbol{K}$ is symmetric positive definite (SPD). There are is a wide literature devoted to the creation and testing of different kernel functions (see, for example, Refs. [4**?**]). In this work, we make use of the Squared Exponential (SE) kernel function with Automatic Relevance Determining (ARD) weights [20],

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left[-\sum_{k=1}^{d} \frac{(x_k - x_k')^2}{2l_k^2} \right], \tag{2}$$

where σ^2 is the variance and l_k is a length-scale associated with dimension k. While simple in nature, the SE function is considered a standard first choice for most applications of GPs, because it ensures that the resulting functions are smooth (relative to the length-scales). Future work may explore other kernel functions. Finally, without loss of generality, we also assume that all GP priors in this work have a zero mean, $\mu(\boldsymbol{x}) = 0, \forall \boldsymbol{x} \in \mathcal{X}$. As we'll see, this does not mean the predictive distribution mean is zero.

In the absence of training data, the GP encodes our prior assumption on the regularity and variance of the function f, through the kernel function parameters. Inference on f is performed by conditioning the prior distribution of f on training data, where the resulting conditional distribution is called the GP posterior. Specifically, we define a set of m training points as $\boldsymbol{X} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_m^T]^T \in \mathbb{R}^{m \times d}$ and the corresponding noisy observations of the unknown function as

$$\boldsymbol{y} \sim f(\boldsymbol{X}) + \boldsymbol{\epsilon}, \tag{3}$$

where the function is evaluated row-wise on the input vector and the noise vector $\boldsymbol{\epsilon}$ is modeled as independent identically distributed (i.i.d.) Gaussian noise with mean zero and standard deviation σ_n, $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I}_m)$. The GP prior on f means that the observation vector is jointly normal with f evaluated at any test point $\boldsymbol{x}$, such that

$$\begin{bmatrix} \boldsymbol{y} \\ f(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma_n^2 I_m & K(X,\boldsymbol{x}) \\ K(\boldsymbol{x},X) & k(\boldsymbol{x},\boldsymbol{x}) \end{bmatrix} \right), \tag{4}$$

where we have used the notation $K(\boldsymbol{X}, \boldsymbol{X}')$ to represent the $m \times n$ covariance matrix associated with the input vectors $\boldsymbol{X} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{X}' \in \mathbb{R}^{n \times d}$, whose i, j^{th} component is given by $k(\boldsymbol{X}_i, \boldsymbol{X}'_j)$, where $\boldsymbol{X}_i$ is the i^{th} row of $\boldsymbol{X}$. Using standard results for multivariate Gaussian distributions, we can now condition the $f(\boldsymbol{x})$ distribution on the training data $\boldsymbol{X}$ and $\boldsymbol{y}$, to arrive at the predictive posterior distribution for f, which we will denote $\hat{f}(\boldsymbol{x})$ to distinguish it from the prior distribution. Specifically, we have

$$\hat{f}(\boldsymbol{x}) \sim p(f(\boldsymbol{x})|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}) = \mathcal{GP}(\hat{\mu}, \hat{k}), \tag{5}$$

where the mean and kernel functions for the posterior GP are

$$\hat{\mu}(\boldsymbol{x}) = K(\boldsymbol{x}, X)(K(\boldsymbol{X}, \boldsymbol{X}) - \sigma_n^2 I_m)^{-1}\boldsymbol{y}, \tag{6}$$

$$\hat{k}(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - K(\boldsymbol{x}, \boldsymbol{X})(K(\boldsymbol{X}, \boldsymbol{X}) - \sigma_n^2 I_m)^{-1}K(\boldsymbol{X}, \boldsymbol{x}'). \tag{7}$$

Note that while the prior mean was assumed to be zero, the posterior distribution is a linear combination of the observation values, where the weights are proportional to the proximity of the test point to the observation locations. The covariance is composed of two terms: the first is the prior covariance which is reduced by the second, strictly positive term, representing the information obtained from the observations. Note that the covariance is independent of the observation values and only depends on the location of the training points in the input space.

Apart from the kernel parameters (often called hyperparameters), the predictive distribution obtained from the Gaussian Process Regressor (GPR) is only dependent on the training data. Because of this, it is usually classified as a nonparametric supervised learning model. When the kernel hyperparameters are not known a priori, they are generally learned by maximizing the log of the marginal likelihood that the observations are produced by the GP prior, given as

$$\begin{aligned} \ln p(\boldsymbol{y}|\boldsymbol{X}, \sigma_n, \boldsymbol{\theta}) &= \ln \int p(\boldsymbol{y}|f(\boldsymbol{X}), \sigma_n)\ p(f(\boldsymbol{X})|\boldsymbol{X}, \boldsymbol{\theta})\ df \\ &= -\frac{1}{2}\boldsymbol{y}^T\left(K(\boldsymbol{X}, \boldsymbol{X}; \boldsymbol{\theta}) + \sigma_n^2 \boldsymbol{I}_m\right)^{-1}\boldsymbol{y} - \frac{1}{2}\ln\left|K(\boldsymbol{X}, \boldsymbol{X}; \boldsymbol{\theta}) + \sigma_n^2 \boldsymbol{I}_m\right| - \frac{m}{2}\ln 2\pi, \end{aligned} \tag{8}$$

where we have explicitly written the dependence on the kernel hyperparameters, denoted as the vector $\boldsymbol{\theta}$. Maximization of the marginal likelihood can be efficiently computed using a gradient descent algorithm [4].

B. Multifidelity Gaussian Processes Regression

The basic GPR presented in the previous subsection does not address the situation when we have multiple sources of data with varying degrees of fidelity. In particular, we are interested in situations were low fidelity data is abundant and relatively cheap to obtain while higher fidelity data is sparse and more costly. In the context of aerodynamic modeling, this stiutation can arise when we supplement data from wind tunnel tests or high-fidelity CFD simulations with cheap, lower-fidelity CFD simulations, or when a vehicle's design is modified and we use older data to inform predictions for the new configuration. Under these scenarios, we wish to use the dense, low fidelity data to learn general trends over the widest range of the input domain, while anchoring to the sparse high fidelity data, whenever possible. More specifically, we now consider the situation where we have s datasets $(\boldsymbol{X}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{X}_s, \boldsymbol{y}_s)$, which increase in fidelity from the lowest level labeled 1 up to the highest level s. Each dataset i is comprised of m_i observations, such that $\boldsymbol{X}_i \in \mathbb{R}^{m_i \times d}$ and $\boldsymbol{y}_i \in \mathbb{R}^{m_i}$. The first GPR model capable of dealing with this situation was the linear autoregressive (AR1) multifidelity GPR of Kennedy and O'Hagan [5], which can be written as follows

$$f_1(\boldsymbol{x}) = \delta_1(\boldsymbol{x}) \tag{9}$$

$$f_i(\boldsymbol{x}) = \rho_i f_{i-1}(\boldsymbol{x}) + \delta_i(\boldsymbol{x}), \quad \forall i \in \{2, \ldots, s\}, \tag{10}$$

$$\delta_i(\boldsymbol{x}) \sim \mathcal{GP}(0, k_i), \quad \forall i \in \{1, \ldots, s\}. \tag{11}$$

In this model, a GP prior with zero mean and kernel function k_1 is placed on the lowest level of fidelity. Each subsequent level i is assumed to be correlated with the level below it through a linear correlation coefficient ρ_i, plus a nonlinear correction function $\delta_i(\boldsymbol{x})$, which is also assumed to be a GP with zero mean and kernel function k_i. Each of the GPs are assumed to be independent of one another. Assuming, as we did for the single-fidelity GP, that the target vectors $\boldsymbol{y}_i$ are noisy observations of each f_i,

$$\boldsymbol{y}_i = f_i(\boldsymbol{X}_i) + \boldsymbol{\epsilon}_i, \tag{12}$$

where the noise vector $\boldsymbol{\epsilon}_i$ is composed of i.i.d. Gaussian noise variables with zero mean and standard deviation σ_{ni}, the multifidelity GP prior leads to a joint normal distribution for the combined vector of observations and test point at the

highest fidelity level $f_s(\boldsymbol{x})$. For example, with three levels of fidelity, we have

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ f_3(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_1^{11} & \rho_2 K_1^{12} & \rho_2\rho_3 K_1^{13} & \rho_2\rho_3 K_1^{1*} \\ \rho_2 K_1^{21} & \rho_2^2 K_1^{22} + K_2^{22} & \rho_2^2\rho_3 K_1^{23} + \rho_3 K_2^{23} & \rho_2^2\rho_3 K_1^{2*} + \rho_3 K_2^{2*} \\ \rho_2\rho_3 K_1^{31} & \rho_2^2\rho_3 K_1^{32} + \rho_3 K_2^{32} & \rho_2^2\rho_3^2 K_1^{33} + \rho_3^2 K_2^{33} + K_3^{33} & \rho_2^2\rho_3^2 K_1^{3*} + \rho_3^2 K_2^{3*} + K_3^{3*} \\ \rho_2\rho_3 K_1^{*1} & \rho_2^2\rho_3 K_1^{*2} + \rho_3 K_2^{*2} & \rho_2^2\rho_3^2 K_1^{*3} + \rho_3^2 K_2^{*3} + K_3^{*3} & \rho_2^2\rho_3^2 K_1^{**} + \rho_3^2 K_2^{**} + K_3^{**} \end{bmatrix} \right), \tag{13}$$

where we have used the short-hand notation $K_k^{ij} = K_k(\boldsymbol{X}_i, \boldsymbol{X}_j)$ and $*$ is used to denote the test point $\boldsymbol{x}$ in the place of $\boldsymbol{X}_i$ or $\boldsymbol{X}_j$. Additionally, in an effort to reduce the complexity of the covariance structure above, the observation variance $\sigma_{ni}^2 \boldsymbol{I}_{m_i}$ is implicitly included in each K_i^{ii} term. See Appendix V.A for the derivation of the covariance structure above. As with the single-fidelity GPR, the predictive posterior distribution on $f_s(\boldsymbol{x})$ ($s = 3$ in the above example) is obtained by conditioning the above multivariate normal distribution on the training data, which results in a posterior GP, similar to the one detailed in Eqs. (5 – 7).

We note that the mean and variance of Eqs. (6 and 7) both require the inverse of the data covariance matrix $K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n \boldsymbol{I}_m$. Since this matrix is SPD, the most efficient approach is to use the Cholesky factorization to perform the inverse, which, for the single-fidelity GPR with m data points, scales with $O(m^3)$. For the multifidelity approach, the data covariance matrix is the 3x3 block matrix in the upper left corner of the covariance matrix in Eq. (13). A Cholesky factorization of this matrix scales with $O([\sum_{i=1}^{s} m_i]^3)$. As the number of data points and fidelity levels increase, the Cholesky factorization will dominate the computational cost of the multifidelity GPR and can become prohibitive.

In an effort to reduce this computational cost, Le Gratiet [21] introduced a modification of the AR1 model, known as the recursive AR1 GPR framework, in which the linear correlation of Eq. (10) is replaced by a correlation on the posterior of the previous level,

$$f_i(\boldsymbol{x}) = \rho_i \hat{f}_i(\boldsymbol{x}) + \delta_i(\boldsymbol{x}). \tag{14}$$

This seemingly trivial modification results in a massive decrease in complexity for the AR1 scheme by allowing each level to be trained by itself, using a recursive formulation. In addition, when subsequent datasets are subsets of the previous dataset, the posterior distribution for each level is readily obtained as

$$\hat{f}_i(\boldsymbol{x}) \sim \mathcal{GP}(\hat{\mu}_i, \hat{k}_i), \tag{15}$$

$$\hat{\mu}_i(\boldsymbol{x}) = \rho_i \hat{\mu}_{i-1}(\boldsymbol{x}) + K_i(\boldsymbol{x}, \boldsymbol{X}_i)[K_i(\boldsymbol{X}_i, \boldsymbol{X}_i) + \sigma_{ni}^2 \boldsymbol{I}_{m_i}]^{-1}(\boldsymbol{y}_i - \rho_i \hat{\mu}_{i-1}(\boldsymbol{X}_i)), \tag{16}$$

$$\hat{k}_i(\boldsymbol{x}, \boldsymbol{x}') = \rho_i^2 \hat{k}_{i-1}(\boldsymbol{x}, \boldsymbol{x}') + k_i(\boldsymbol{x}, \boldsymbol{x}') - K_i(\boldsymbol{x}, \boldsymbol{X}_i)[K_i(\boldsymbol{X}, \boldsymbol{X}) - \sigma_{ni}^2 I_{m_i}]^{-1} K_i(\boldsymbol{X}_i, \boldsymbol{x}'), \tag{17}$$

where the correlation coefficients are obtained by integrating out the mean, leading to

$$\rho_i = \frac{\boldsymbol{y}_i^T \boldsymbol{y}_{i-1}(\boldsymbol{X}_i)}{\boldsymbol{y}_i^T \boldsymbol{y}_i}, \tag{18}$$

where $\boldsymbol{y}_{i-1}(\boldsymbol{X}_i)$ is used to denote the subset of observations at fidelity level $i-1$ corresponding to the data points for level i. Note that this approach is identical to training s separate GPs, one for each level of fidelity. The first level is trained directly on dataset 1, yielding the posterior GP for $\hat{f}_1$. Subsequent GPs are then trained on the difference between the observations at that level and the corresponding subset of observations from the previous level. The computational complexity of this approach is therefore $O(\sum_{i=1}^{s} m_i^3) \approx O(m_1^3)$, when the number of training observations decreases for each subsequent level. This approach will be used when appropriate in the multihierarchy framework detailed in the next section.

C. Multihierarchy Gaussian Processes

The multifidelity approach outlined in the previous subsection relies on expert judgment to rank different datasets in terms of their fidelity towards the true function of interest. In many situations this is both difficult and does not represent reality. In the case of the Orion vehicle, for example, wind tunnel testing was performed on different geometrical models with different roughness characteristics, though no model is likely a perfect representation of the real flight vehicle. Typically, aerodatabase designers rely instead on expert judgment about what data can be considered "nominal" and what data should be used to assign uncertainties off the nominal values. In this context, the word "nominal" is meant to describe a fictitious scenario in which the vehicle is in a "pristine" state. For example, a completely smooth and axisymmetric configuration. The term "off-nominal" represents differences in the real vehicle with respect to

this idealized configuration, such as including surface roughness and asymmetries in the geometry. This is not to be confused with the statistical meaning of nominal, which generally refers to the mean, or expected value of a distribution. The actual meaning of nominal and off-nominal for a given aerodatabase, will depend on the vehicle itself, and the engineers designing it. In this spirit, we propose a multihierarchy GP model, in which one dataset is taken to be the nominal condition, and other datasets are used to infer possible corrections to the nominal.

We will denote the nominal function of interest as $f_0(\boldsymbol{x})$ with corresponding data $(\boldsymbol{X}_0, \boldsymbol{y}_0)$. A zero-mean GP prior is then placed on the nominal function, such that

$$f_0(\boldsymbol{x}) = \delta_0(\boldsymbol{x}) \sim \mathcal{GP}(0, k_0), \tag{19}$$

where k_0 is the corresponding kernel function assigned to the nominal prediction. Next, we assume the unknown "true" function $f(\boldsymbol{x})$ is described by the nominal value plus a linear combination of c correction terms, which we denote $\Delta f_i(\boldsymbol{x})$, such that

$$f(\boldsymbol{x}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{c} w_i \Delta f_i(\boldsymbol{x}), \tag{20}$$

where $w_i \in \mathbb{R}$ are linear weights which control the amount of each correction that is applied to the nominal. Each correction function $\Delta f_i(\boldsymbol{x})$ is defined as the offset from the nominal necessary to recover the unknown function $f_i(\boldsymbol{x})$ underlying the corresponding dataset $(\boldsymbol{X}_i, \boldsymbol{y}_i)$, such that

$$\Delta f_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) - f_0(\boldsymbol{x}), \quad \forall i \in \{1, \ldots, c\}. \tag{21}$$

To close the system, we place a two-fidelity GP prior on each $f_i(\boldsymbol{x})$,

$$f_i(\boldsymbol{x}) = \rho_i f_0(\boldsymbol{x}) + \delta_i(\boldsymbol{x}), \tag{22}$$

$$\delta_i(\boldsymbol{x}) \sim \mathcal{GP}(0, k_i), \tag{23}$$

where the nominal function is taken as the low fidelity model.

This assignment of fidelities is not really crucial to the development of the model, but it does serve to highlight how this type of multihierarchy model can be formulated. In practice, it may make more sense to have some correction functions defined as a simple GP while others may be defined as multifidelity GPs with more than two levels. Assigning the nominal model as the lowest fidelity is also not necessary. Any set of hierarchies can be defined depending on what is appropriate given the available datasets. In this regard, the modeler has a significant advantage over using the classic AR1 scheme, in that they have more leeway to construct fidelity relationships which better approximate their intuition or reality, based on expert knowledge of the datasets at hand.

Once the fidelity relationships are determined, however, the model forms a GP prior on the unknown function $f(\boldsymbol{x})$, since it is a linear combination of GPs. This is easily seen by rewriting the above equations more compactly in a matrix vector form. For example, with two correction terms, the example fidelity structure above can be written as

$$\begin{bmatrix} f_0(\boldsymbol{x}) \\ f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \rho_1 & 1 & 0 \\ \rho_2 & 0 & 1 \\ w_0 & w_1 & w_2 \end{bmatrix} \begin{bmatrix} \delta_0(\boldsymbol{x}) \\ \delta_1(\boldsymbol{x}) \\ \delta_2(\boldsymbol{x}) \end{bmatrix}, \tag{24}$$

where we have defined $w_0 = 1 + \sum_{i=1}^{2} w_i(\rho_i - 1)$ for convenience. Written in this form, it is clear that the multihierarchy model is a special case of a general linear mixture of independent GPs. Following Appendix V.A, we once again find that the observation vectors and $f(\boldsymbol{x})$ are jointly normal, such that

$$\begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ f(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_0^{00} + \sigma_{n0}^2 \boldsymbol{I} & \rho_1 K_0^{01} & \rho_2 K_0^{02} & w_0 K_0^{0*} \\ \rho_1 K_0^{10} & \rho_1^2 K_0^{11} + K_1^{11} + \sigma_{n1}^2 \boldsymbol{I} & \rho_1 \rho_2 K_0^{12} & \rho_1 w_0 K_0^{1*} + w_1 K_1^{1*} \\ \rho_2 K_0^{20} & \rho_1 \rho_2 K_0^{21} & \rho_2^2 K_0^{22} + K_2^{22} + \sigma_{n2} \boldsymbol{I} & \rho_2 w_0 K_0^{2*} + w_2 K_2^{2*} \\ w_0 K_0^{*0} & \rho_1 w_0 K_0^{*1} + w_1 K_1^{*1} & \rho_2 w_0 K_0^{*2} + w_2 K_2^{*2} & w_0^2 K_0^{**} + w_1^2 K_1^{**} + w_2^2 K_2^{**} \end{bmatrix} \right), \tag{25}$$

where we have reused the short-hand notation introduced for the AR1 model. Like the AR1 model, we can now condition on the training data to derive the posterior distribution for $\hat{f}((x))$, however this approach suffers from the same

computational cost of the Cholesky factorization of the coupled covariance matrix. Instead, we follow the recursive approach of Le Gratiet and replace all the dependencies on the GP priors with their posteriors.

What remains is the determination of the weighting coefficients w_1 and w_2. In the case where we have no data that is believed to represent the true function of interest, these parameters should be treated as uncertain and given prior distributions based on the best available knowledge. When data becomes available (for example after a flight test), the weights can be updated to reflect the additional information using the Bayesian formalism. Alternatively, if some data $(\boldsymbol{X}, \boldsymbol{y})$ is present, but limited, another approach which could be used is to estimate the coefficients by integrating out the mean of the posterior on f, which for our example fidelity structure, yields the following linear system

$$[\mu_1(\boldsymbol{X}) - \mu_0(\boldsymbol{X}), \mu_2(\boldsymbol{X}) - \mu_0(\boldsymbol{X})] \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \boldsymbol{y} - \mu_0(\boldsymbol{X}), \tag{26}$$

which can be solved via a linear least squares regression.

III. Example Orion Dataset

A. Description of the Orion spacecraft and wind tunnel testing

The Orion vehicle is an Apollo-derived spacecraft currently being developed as NASA's flagship human-rated vehicle for deep-space missions to the Moon and Mars. In a series of increasingly complex missions, the NASA Artemis program will carry astronauts back to the Moon aboard Orion. The vehicle consists of four major subsystems, shown in Fig. 1. The Crew Module (CM) serves as both the habitable portion of the vehicle during space operations and the reentry vehicle used to return astronauts to Earth. During atmospheric entry, the CM will decelerate from hypersonic speeds to subsonic speeds before deploying parachutes to land the crew safely in the ocean. As the vehicle decelerates, the shock-heated atmospheric gases will heat the forebody of the vehicle, causing the forebody TPS to undergo a process known as ablation, which exchanges much of the heat of the impinging gas for thermal decomposition of the TPS material. This leads to a gradual shape change of the TPS itself over the reentry trajectory. The nominal Orion CM Outer Mold Line (OML) is an axisymmetric 30° sphere-cone geometry with nose and shoulder radii of 237.6 in and 9.9 in, respectively, a maximum diameter of 198 in, and height of 130 in. This geometry is referred to as the Integrated Design Assessment Team (IDAT) geometry for historical reasons (see e.g., Ref. [2]). Figure 2 provides a description of the Orion CM coordinate system and associated definitions of force and moment coefficients showing the IDAT geometry. Note that in this coordinate system, an angle of attack of 180° corresponds to a zero trim angle in forward flight. The actual geometry of the Orion CM has undergone numerous changes during its design cycle. The latest variant, denoted 700D, recently flew as the Artemis I vehicle. Details such as windows, hatches, aftbody tiles, protuberances, and block-Avcoat heathsield are included. The heatshield shape is asymmetric, with thicker TPS in regions of higher expected ablation. Both the IDAT and 700D (and earlier variants) have undergone extensive wind tunnel testing at NASA facilities during the design of the vehicle. In addition, thousands of Computational Fluid Dynamics (CFD) simulations have been performed to help validate the wind tunnel testing and extrapolate to flight conditions not easily produced in ground-based facilities. The current aerodatabase for the Orion CM provides tabulated nominal aerodynamic coefficients based on the IDAT configuration with uncertainties associated with various physical and computational phenomena, including turbulence and transition, CFD grid convergence, TPS shape change, and roughness effects. For an overview of the current Orion aerodatabase development methodologies, see Ref. [22].

In this work, we will limit the scope of the available data to a subset of a single test campaign, in order to focus on the proposed methodology and limit the sources of considered uncertainties. Specifically, we use data from the 133-CA Orion test campaign [23] performed in the National Transonic Facility (NTF). The NTF is a fan-driven, closed-circuit, continuous-flow, cryogenic, pressurized wind tunnel at the NASA Langley Research Center (LaRC) [24]. The tunnel has an 8.2x8.2x25 foot test section and is designed to obtain aerodynamic data on subscale vehicle models at subsonic and transonic speeds through the full-scale Reynolds number regimes for most flight vehicles [25]. The 133-CA test collected aerodynamic force and moment data for several model configurations of the Orion CM over angles of attack $158° \leq \alpha \leq 202°$, Mach numbers $0.2 \leq M_\infty \leq 0.9$, side-slip angles $|\beta| \leq 15°$, and Reynolds numbers $3.8 \times 10^6 \leq \mathrm{Re_D} \leq 28 \times 10^6$. These tests are an extension of an earlier [26] test campaign designated 89-CA with the primary difference being the types of heatshields tested. The model body, balance attachment, sting, etc. were all the same. The 89-CA test utilized 2 heatshields (607D smooth and 607D dimpled). 133-CA tested an additional 6 heatshields. In total, seven unique configurations were tested in order to understand the effects of vehicle asymmetry,

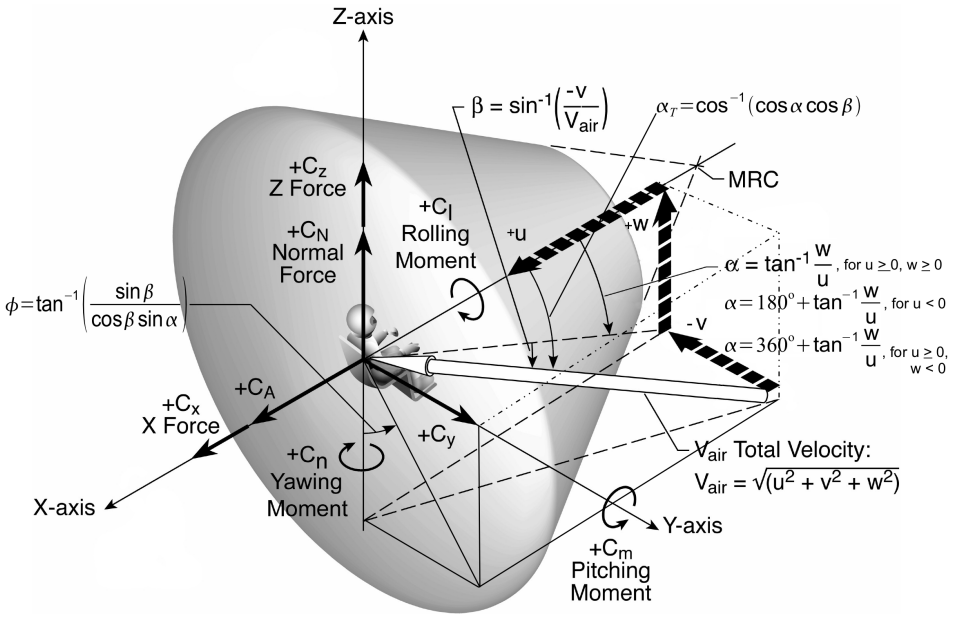


Fig. 2 Axis, force, and moment definitions for the Orion Crew Module [2].

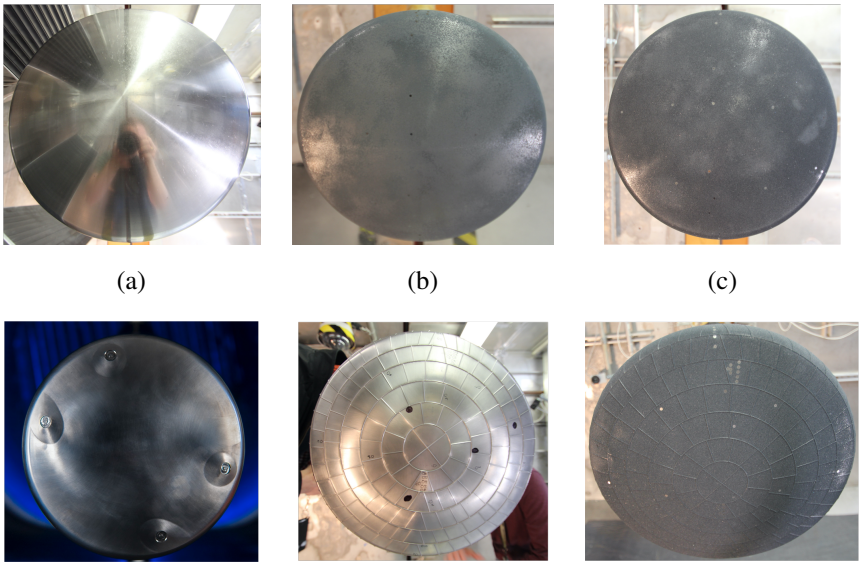
roughness, and the inclusion (or not) of the Forward Bay Cover (FBC) on the vehicle static aerodynamic coefficients. The configurations are summarized in the first four columns of Table 1. Data for both the IDAT and 700D geometries are available, with different heat shield simulators pictured in Fig. 3. Smooth heatshield data are provided for both geometries with roughness effects simulated using a fine and 240 grit random roughness for the IDAT and a "fencing" effect for the 700D geometry. Fencing occurs when the tiles of the heat shield recess due to ablation, leaving the gap filler between the tiles slightly raised above the recessed acreage surface. Additionally, the backshell details (windows, LAS attach wells, etc.) have been filled in to make the backshell geometry closer to axisymmetric.

B. Dataset curation from wind tunnel data

The datasets used for model training in this work are taken as a subset of the available 133-CA data as summarized in the fifth column of Table 1. The smooth IDAT data with and without the FBC filled is taken as the nominal configuration and labeled as dataset 1. To account for roughness effects, we use IDAT 240 grit data, labeled here as dataset 2. Uncertainties associated with differences between the nominal and true geometries are modeled as the difference between the smooth 700D data and the nominal configuration, which we label as dataset 3. Finally, a fourth dataset is extracted from the 700D fenced plus grit configuration to provide testing and comparison data for the trained model. In order to reduce the model complexity and to focus the effort of this work, only data at zero side-slip angles are

Table 1 133-CA test configurations and datasets used in this work.

Config.	Geometry	Heatshield	FBC	Dataset	Modifier	Raw Size	Binned Size
1	IDAT	smooth	not filled	1	symmetry	8476	1192
2	IDAT	smooth	filled	1	symmetry		
3	IDAT	600 grit	not filled	-			
4	IDAT	240 grit	filled	2	symmetry	1774	389
5	700D	smooth	not filled	3	$\beta = 0°$	1115	713
6	700D	fenced	not filled	-			
7	700D	fenced & 240 grit	not filled	4	$\beta = 0°$	284	204



(a) (b) (c)

(d) (e) (f)

Fig. 3 Heatshields tested during the 133-CA test campaign: (a) IDAT smooth, (b) IDAT 600 grit, (c) IDAT 240 grit, (d) 700D smooth, (e) 700D fenced, and (f) 700D fenced plus 240 grit.

considered, leaving the model inputs as $(\alpha, \mathrm{M}_\infty, \mathrm{Re}_D)$. We also only consider modeling the lift and drag coefficients, C_L and C_D, and the pitching moment coefficient at the center of gravity $C_{\mathrm{m,cg}}$ as outputs, though it is straightforward to extend the methodology to any output of interest.

In the case of the axisymmetric geometries (datasets 1 and 2), we make use of the model symmetry to extend the available data in two ways: first by transforming the data with non-zero side-slip angles to the equivalent angle of attack induced by algebraically rolling the model such that the equivalent side-slip angle is zero, and second, by duplicating data across the $\alpha = 180^\circ$ symmetry plane. The transformation to the equivalent zero side-slip data is as follows. First, for each data point, the angle of attack and side-slip angle are used to compute the total angle of attack α_T and roll angle ϕ, as shown in Fig. 2, where

$$\alpha_T = \arccos(\cos\alpha \cos\beta), \tag{27}$$

$$\phi = \arctan(\tan\beta \csc\alpha). \tag{28}$$

Next, the normal force and pitching moment at the axial center of gravity coefficients are rotated by the roll angle, such that

$$C_N = C_N \cos\phi - C_Y \sin\phi, \tag{29}$$

$$C_{m,\mathrm{cgx}} = C_{m,\mathrm{cgx}} \cos\phi - C_{n,\mathrm{cgx}} \sin\phi. \tag{30}$$

Finally, the angle of attack, side-slip, lift and drag forces, and pitching moment about the center of gravity are updated for the rotated system, yielding

$$\alpha = \alpha_T, \tag{31}$$

$$\beta = 0^\circ, \tag{32}$$

$$C_L = C_N \cos\alpha_T - C_A \sin\alpha_T, \tag{33}$$

$$C_D = C_N \sin\alpha_T + C_A \cos\alpha_T, \tag{34}$$

$$C_{m,\mathrm{cg}} = C_{m,\mathrm{cgx}} - C_A \Delta z_{\mathrm{cg}}/D, \tag{35}$$

where Δz_{cg} is the vertical offset of the center of gravity from the symmetric centerline. Once transformed, the updated data is then mirrored across the symmetry plane of $\alpha = 180^\circ$. For an axisymmetric geometry, the drag force is

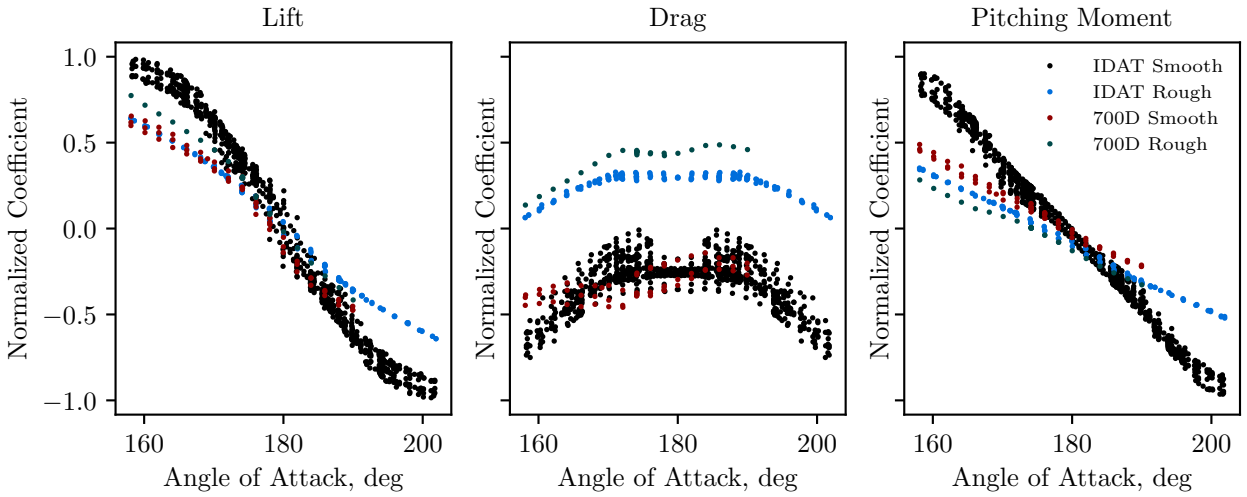


Fig. 4 Example slice of the raw dataset around $M_\infty = 0.3$ and $Re_D = 7.5 \times 10^6$.

an even function about $\alpha = 180°$, while the lift force and pitching moment about the axial center of gravity are odd. Thus, for each (α, M_∞, Re_D, C_L, C_D, $C_{m,cg}$) point in the transformed dataset, a duplicate point is added as ($360° - \alpha$, M_∞, Re_D, $-C_L$, C_D, $-C_{m,cgx} + C_A \Delta z_{cg}/D$).

For the asymmetric 700D datasets (3 and 4), no such transformations are possible, and the non-zero side-slip data is simply dropped from the dataset. Lastly, for all datasets, the output variables are normalized in the range of $[-1, 1]$ based on the minimum and maximum values across the entire set of data (datasets 1-4). This both allows for easy comparisons of uncertainties between different model outputs by placing those outputs on the same scale and alleviates the need for restrictions on the published data. Therefore, all data and figures presented here for C_L, C_D, and $C_{m,cg}$ are on this normalized scale. The final size of each dataset based on the described modifications are listed as the "raw" size in Table 1 and an example slice of the raw dataset is provided in Fig. 4.

C. Binning and observation noise estimation

Before model training, it is helpful to further reduce the dataset by binning the data around clusters in the input space. These clusters naturally arise as the wind tunnel data are typically taken at set Mach and Reynolds numbers and α-sweeps collect data at regular intervals. Binning the data is particularly useful when using the recursive AR1 model as it can ensure that higher fidelity dataset inputs belong to subsets of the lower fidelity data. Additionally, averaging the data on the cluster centroids allows for an estimation of the observation noise at each binned point by computing the variance of the test measurements for all data associated with the given bin. The bin centers for each input variable were determined independently using the k-means algorithm on all the values of the given input variable across all datasets in order to ensure that each dataset shared the same bin centers. The number of clusters k for each input variable was determined by progressively increasing k until the distance between the closest clusters fell below a given tolerance. This tolerance was taken to be 0.5°, 0.025, and 0.25×10^6 for α, M_∞, and Re_D, respectively. To ensure that at each iteration of k, the best possible set of clusters were used for this tolerance criteria, the k-means algorithm was restarted 100 times for each k with different random initial centroids. The resulting input bins and associated raw data points for each dataset are shown in Fig. 5. In addition, the binned data for the same slice in Fig. 4 is provided in Fig. 6.

The computed bin standard deviations in C_L, C_D, and $C_{m,cg}$versus the number of points in each bin are plotted in Fig. 7. Note that some bins have a single point, while others have more than 75. It is expected that as the number of points in a given bin grows, so does the accuracy at which the true variance in the data is approximated by the empirical variance plotted in Fig. 7. The convergence of the standard deviation across all datasets as the bin count increases indicates that there is a noise floor around a standard deviation of about 0.05, which is likely a fundamental limit of the wind tunnel itself. Note that some noise is expected due to the accuracy limits of the balance and repeatability of the tunnel. In addition, sting effects are completely ignored in this work, but should be treated carefully for a real aerodynamic database. Therefore, as a conservative estimate, the observation standard deviation σ_n is taken to be $\sigma_n = 0.1$ for all the results presented in this paper. For the normalized coefficients, this represents a standard deviation

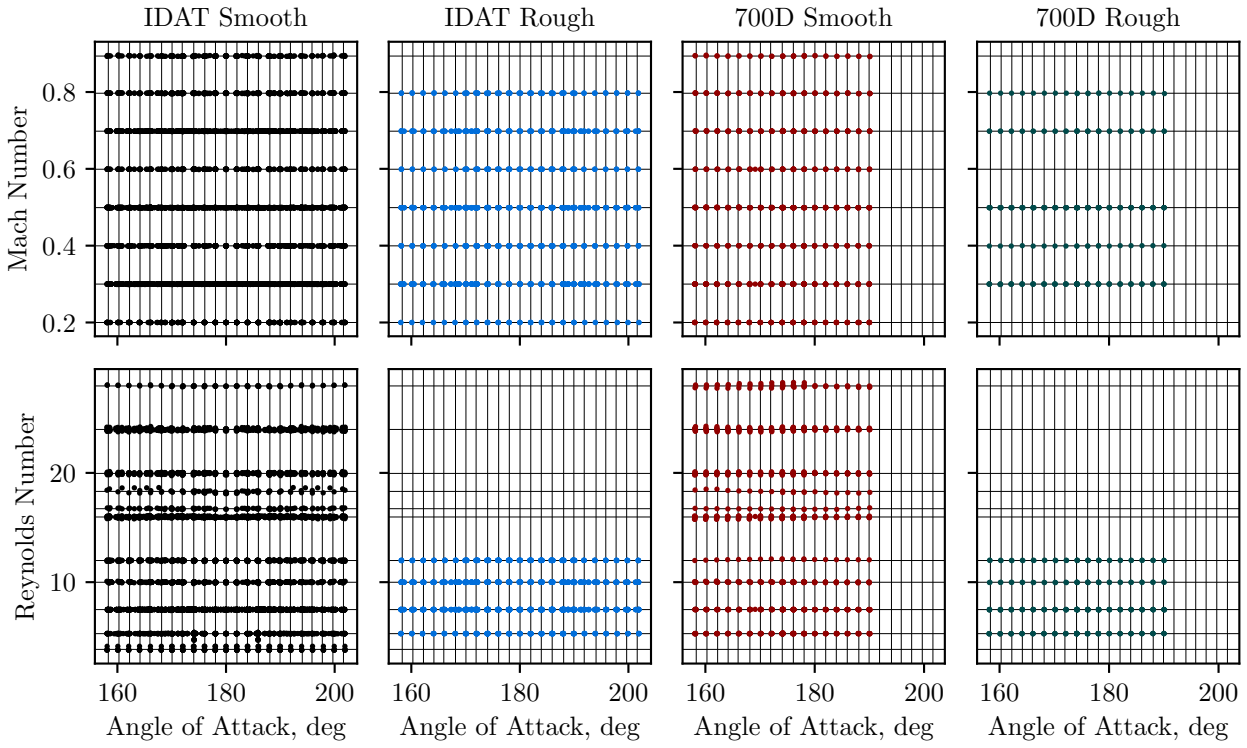


Fig. 5 Input bins (line intersections) with raw input data (points) for the four selected test model configurations.

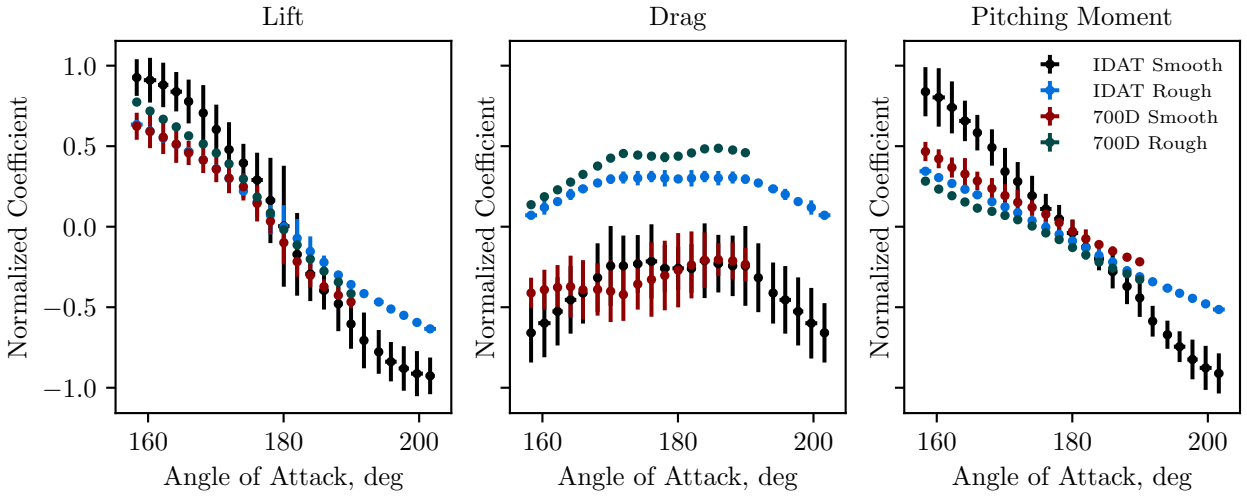


Fig. 6 Example slice of the binned dataset around $M_\infty = 0.3$ and $Re_D = 7.5 \times 10^6$. Error bars represent three standard deviations.

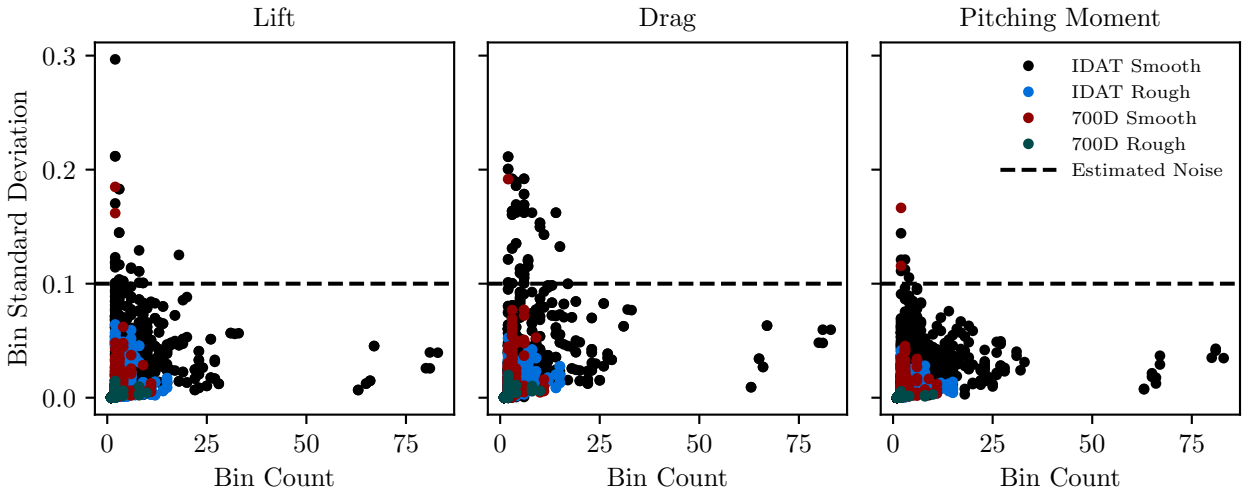


Fig. 7 Scatter plot of all bin standard deviations across each dataset.

of 5 % of the total range of the data between -1 and 1.

IV. Results and Discussion

In this section, we highlight some key results obtained with the multihierarchy GP model, applied to the Orion datasets described in the previous section. The lift, drag, and pitching moment coefficients are all treated as independent datasets and each coefficient is modeled as a separate multihierarchy GP, using the example fidelity structure described in Sec. II.C. Specifically, we treat the smooth, symmetric data (IDAT Smooth) as our nominal condition. Two corrections to the nominal aim to capture surface roughness effects (IDAT Rough) and asymmetries in the geometry (700D Smooth). For testing purposes, we use the 700D Rough dataset to simulate measured flight data from the real vehicle and our aim is to fit that data under the multihierarchy framework. All GPs are trained on the binned data with an assumed observation variance of $\sigma_{ni}^2 = \sigma_n^2 = 0.01$. Kernel hyperparameters are optimized for each independent GP by maximizing the log marginal likelihood on the data. The optimization is performed using the L-BFGS-B quasi-Newton algorithm with 20 random restarts to ensure global optima are obtained.

A. Predictive posterior distributions

We begin by looking at the predictive posterior distributions obtained, after training the multihierarchy GP models on the given wind tunnel data. One benefit of using GPs over other supervised regression techniques, like Neural Networks or Polynomial Regression, is that the trained models are interpretable and generally have very few parameters. For the multihierarchy model presented here, each aerodynamic coefficient is modeled using twelve model parameters, including two correction weights w_1 and w_2, two correlation weights ρ_1 and ρ_2, and four kernel parameters for each of the three independent GP models, which consist of the variance σ and separate length scales for each input dimension, l_α, l_{M_∞}, and l_{Re_D}. The learned values for each of these parameters are given in Table 2. These parameters can tell us a great deal about the underlying data. Looking at the correction weights in Table 2a, we see that the lift coefficient is fairly well reproduced by the nominal model alone, since the weights for both the roughness and asymmetry corrections are relatively small, at 0.48 and 0.16, respectively. Conversely, with roughness correction weights greater than one for the drag and pitching moment coefficients, we see that surface roughness plays an important role in these coefficients. The correlation parameters provide additional information. Specifically, since the asymmetry correlation parameters are all near one, the nominal and roughness data are highly correlated. This should allow us to more accurately extrapolate the model to regions of the input space where we have nominal data but limited or no asymmetry data. Conversely, the low correlation parameter for the drag coefficient roughness data shows that this coefficient may be more difficult to extrapolate to regions with less roughness data.

The kernel parameters provide useful information about the regularity of each model function (as well as the underlying data). The variance parameter provides an estimate of the uncertainty in the random model function

Table 2 Learned model parameters for the multihierarchy GP models trained on the datasets detailed in Sec. III.

(a) Correction and correlation weight coefficients.

Model	w_1	w_2	ρ_1	ρ_2
C_L	0.48	0.16	0.71	0.92
C_D	1.20	-0.32	0.39	0.90
$C_\mathrm{m,cg}$	1.48	-0.33	0.64	0.75

(b) Nominal GP kernel coefficients.

Model	σ	l_α, deg.	l_{M_∞}	$l_{\mathrm{Re}_D}/10^6$
C_L	0.45	23.18	0.40	11.74
C_D	0.45	29.69	0.17	1.69
$C_\mathrm{m,cg}$	0.44	23.22	0.48	7.54

(c) Roughness correction GP kernel coefficients.

Model	σ	l_α, deg.	l_{M_∞}	$l_{\mathrm{Re}_D}/10^6$
C_L	0.09	21.61	0.24	1000
C_D	0.23	33.35	0.24	8.02
$C_\mathrm{m,cg}$	0.13	20.52	0.34	1000

(d) Asymmetry correction GP kernel coefficients.

Model	σ	l_α, deg.	l_{M_∞}	$l_{\mathrm{Re}_D}/10^6$
C_L	0.10	9.88	0.20	7.22
C_D	0.10	12.07	0.07	2.21
$C_\mathrm{m,cg}$	0.06	16.06	0.30	7.13

predictions and is directly tied to the spread in the data. Note this is subtly different from the observation noise which indicates how accurate the measurements are, while the variance indicates how uncertain the underlying function is, given the noisy observations. The length scale parameters have units corresponding to the input dimension and indicate the distance in that dimension in which significant variations in the output variable occur. Larger length scales imply less variability in the output, while smaller length scales indicate the function varies more rapidly over that dimension. Comparing the length scales across outputs (C_L, C_D, $C_\mathrm{m,cg}$) gives an indication of which outputs are more irregular than others. Comparing across the datasets (nominal, roughness, asymmetry) sheds light on how each phenomenon affects the variability in the modeled function. For example, we can see that all the angle-of-attack length scales (across outputs and datasets) are roughly $O(10)$, ranging from 9.88° to 33.35°, and therefore we should not expect the functions to vary considerably over α ranges smaller than this. However, we see that the α scales for the asymmetry correction are roughly half that of the scales for the nominal model and roughness correction, which indicates that asymmetry introduces additional variability into the model output and implies that the valid extrapolation range for angles of attack are roughly halved when asymmetry effects are important.

Another interesting thing to notice in the kernel parameter data is the difference between the Reynolds number length scales for $C_\mathrm{L}$and $C_\mathrm{m,cg}$as compared to C_D, for the roughness correction kernel, presented in Table 2c. The length scales for $C_\mathrm{L}$and $C_\mathrm{m,cg}$are both exactly 1000, which was used as the upper bound on the Reynolds number scales during the optimization process. This indicates that there is essentially no additional variation in $C_\mathrm{L}$and $C_\mathrm{m,cg}$(apart from the nominal variation) due to roughness effects. The corresponding length scale for C_D, however, is found to be 8.02, implying roughness effects are more strongly dependent on Reynolds number for drag. This is inline with expectations, since drag will be more affected by the laminar-to-turbulent transition location and the size of the suction region on the windward heat shield shoulder [27], both of which are dependent on the Reynolds number. What is interesting about this data is that we have clear evidence that this is related specifically to roughness effects, since we do not see similar differences between the Reynolds number length scales for the nominal and asymmetry correction models (though drag is always shown to be more dependent on Reynolds number than lift or pitching moment).

Apart from model parameters, it is also instructive to plot the actual predictive distributions of the models. Since the input space is three-dimensional, we have chosen to plot the probability distributions for slices of data at fixed Mach numbers of 0.3, 0.5, and 0.8, and for two Reynolds numbers of 7.5×10^6 in Fig. 8 and 24.0×10^6 in Fig. 9. In both figures, the posterior distributions are plotted for a range of angles of attack for each aerodynamic coefficient and each model. For comparison, we also provide the corresponding raw (unbinned) wind tunnel data. The Mach numbers have been chosen to show the variation of each coefficient across the Mach space. The two Reynolds numbers were chosen because the low Re_D shows the behavior of the model where there is abundant data for all datasets and the high Re_D case demonstrates the model in a region where there is no roughness data (see Fig. 5).

In general, we observe excellent agreement with the noisy data in Fig. 8. Recall that each GP is trained on the binned datasets, and not the noisy datasets. Furthermore, there is reasonable agreement with the test dataset (700D Rough). It is important to note that the obtained model distributions for the 700D Rough data is based entirely off the two correction

factors for each coefficient, which were optimized in a linear least squares sense. This suggests that our prior assumption on the relationship between nominal coefficients and corrections for roughness and asymmetry are reasonable. Further improvements to the model could be made by making the correction weights be linear combinations of basis functions, dependent on the input features. Such an approach would likely improve the mean prediction accuracy at the expense of additional parameters which must be learned, though it would not change the general structure of the multihierarchy GP approach.

We can also note the visible increase in the model uncertainty at the boundaries of the angle-of-attack range in the figures where there is no training data. This is a classic feature of GP regression and allows so called out-of-distribution uncertainties to be estimated which essentially describe the uncertainty in the model when extrapolating outside the range of given data. The rate at which this uncertainty increases away from data is proportional to the inverse length scales for each kernel function. The out-of-distribution uncertainty is more visible in Fig. 9, which shows the model predictive distributions for a Reynolds number at which there is no roughness data available. As can be seen in the figure, the uncertainty in the drag coefficient is significantly increased at this Reynolds number which is in line with the observations previously discussed concerning the Reynolds number length scales for the roughness correction. Based on this observation, a possible suggestion for database designers could be to try and obtain more data on the IDAT Rough model at higher Reynolds numbers. It is worth noting that this is a somewhat contrived example, since a significant amount of data does exist from other test campaigns which could be factored into this approach. Rather, the observation serves to highlight how this approach can be used to inform future testing and reasoning about aerodatabase uncertainties.

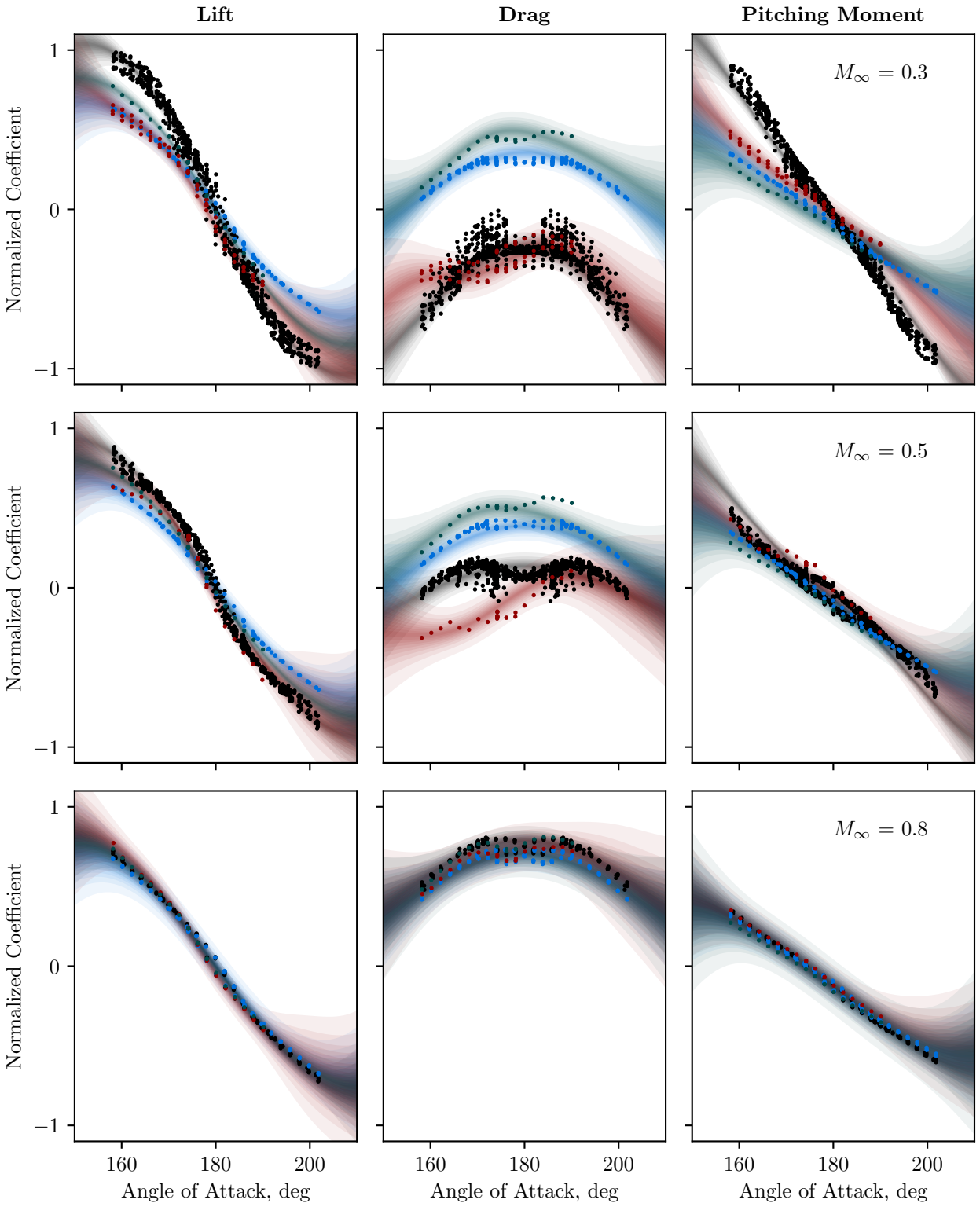


Fig. 8 Example model distributions for normalized aerodynamic coefficients over a range of angles of attack for three Mach numbers and fixed Reynolds number of 7.5×10^6. The shaded regions cover the 99 % confidence interval for each model and are progressively darker as the probability increases. For reference, the corresponding raw data associated with each condition is also plotted as small circles. The colors correspond to each dataset: black, IDAT Smooth; blue, IDAT Rough; red, 700D Smooth; green, 700D Rough.

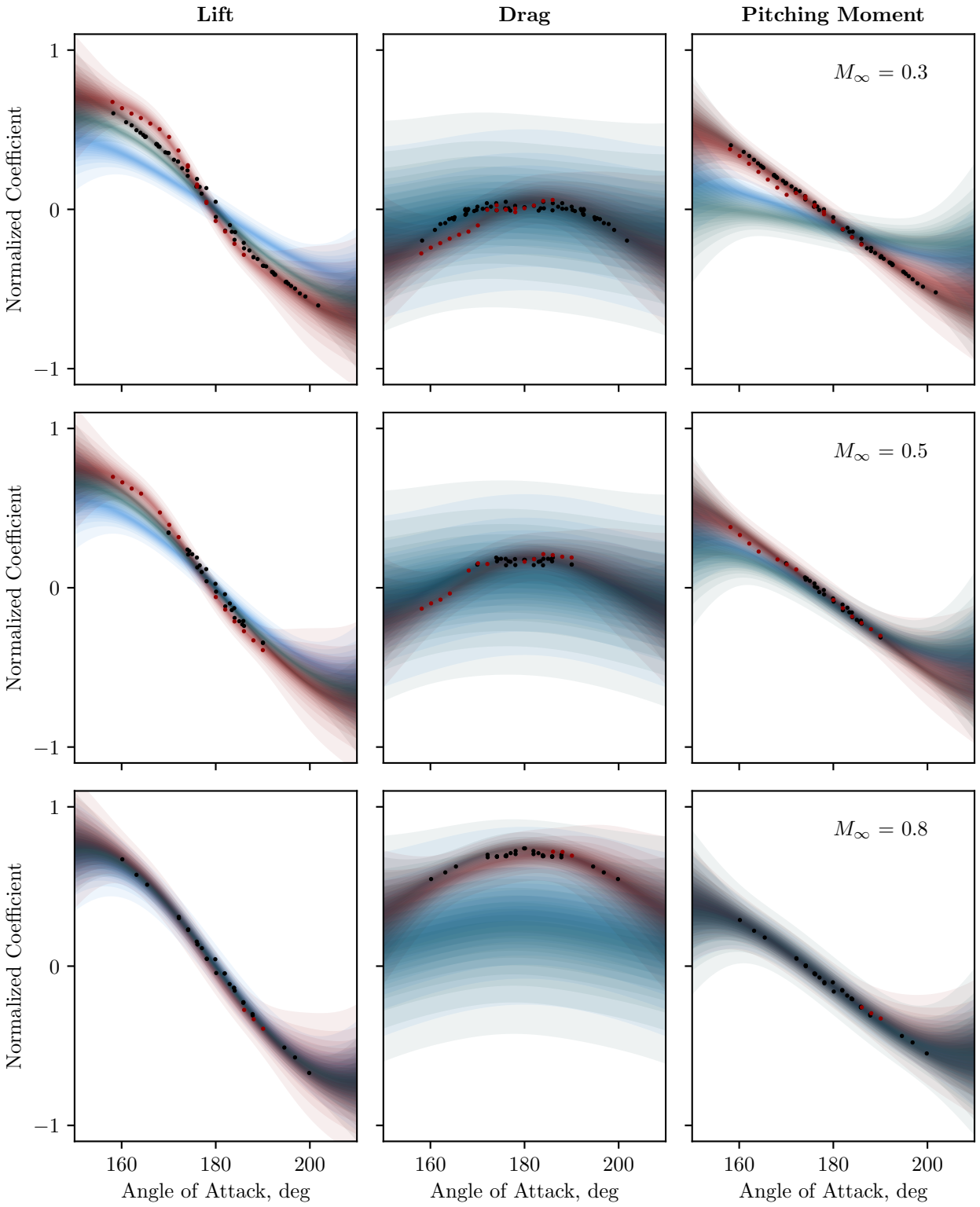


Fig. 9 Example model distributions for normalized aerodynamic coefficients over a range of angles of attack for three Mach numbers and fixed Reynolds number of 24×10^6. The shaded regions cover the 99 % confidence interval for each model and are progressively darker as the probability increases. For reference, the corresponding raw data associated with each condition is also plotted as small circles. The colors correspond to each dataset: black, IDAT Smooth; blue, IDAT Rough; red, 700D Smooth; green, 700D Rough.

Finally, as a last example of what information can be gained from the posterior distributions in the multihierachy framework, we consider the relative contribution of each source of uncertainty to the total uncertainty in the trained predictive model. We note that the standard deviation of the target function at any point in the input space is given as

$$\sigma(\boldsymbol{x}) = \sqrt{w_0^2\sigma_0^2(\boldsymbol{x}) + w_1^2\sigma_1^2(\boldsymbol{x}) + w_2^2\sigma_2^2(\boldsymbol{x})}, \tag{36}$$

where the variances for the nominal (σ_0^2) and roughness and asymmetry correction functions (σ_1^2 and σ_2^2) are obtained directly from the posterior kernel functions $\sigma_i^2(\boldsymbol{x}) = \hat{k}_i(\boldsymbol{x}, \boldsymbol{x})$. We can therefore assign a proportion of the total standard deviation to each component, denoted $\tilde{\sigma}_i(\boldsymbol{x})$, such that

$$\sigma(\boldsymbol{x}) = \tilde{\sigma}_0(\boldsymbol{x}) + \tilde{\sigma}_1(\boldsymbol{x}) + \tilde{\sigma}_2(\boldsymbol{x}), \tag{37}$$

$$\tilde{\sigma}_i(\boldsymbol{x}) = \frac{w_i^2\sigma_i^2(\boldsymbol{x})}{w_0^2\sigma_0^2(\boldsymbol{x}) + w_1^2\sigma_1^2(\boldsymbol{x}) + w_2^2\sigma_2^2(\boldsymbol{x})}\sqrt{w_0^2\sigma_0^2(\boldsymbol{x}) + w_1^2\sigma_1^2(\boldsymbol{x}) + w_2^2\sigma_2^2(\boldsymbol{x})}. \tag{38}$$

As an example, Fig. 10 shows the proportion of each contribution to the total standard deviation for a fixed Mach and Reynolds numbers. For this condition, we see that the uncertainty in the roughness correction tends to dominate the total uncertainty. In addition, the nominal function uncertainty only plays a role in the lift coefficient as it is significantly smaller than the correction uncertainties for the drag and pitching moment coefficients. Asymmetry factors contribute roughly the same uncertainty to all three coefficients at this condition. Note that these uncertainties are on the normalized aerodynamic coefficients, and should not be compared to other published values for the Orion aerodatabase uncertainties, which are generally expressed on the real scales of the coefficients. In addition, these uncertainties only express the uncertainties present in the data but do not incorporate other uncertainties associated with factors not considered here. Rather, this example is intended to highlight how the multihierarchy framework can be used to remove the human guesswork typically used to model such uncertainties.

B. Sampling continuous aerodynamic functions

As motivated in the introduction, aerodynamic databases are used for a number of applications during the design and testing phases of a vehicle's life cycle. Trajectory simulations are a crucial application, which require simulating the vehicle's flight dynamics. In order to achieve this, it is necessary to be able to sample consistent aerodynamic coefficient functions from the probabilistic model, in the context of Monte-Carlo trajectory simulations. Unique trajectory functions are used during each simulation. Historically, these functions have been generated for the Orion vehicle by combining the nominal aerodynamic coefficient values with a constant random dispersion applied to each Monte-Carlo run. While pragmatic in nature, this does not actually sample the true distribution of functions represented by the probabilistic model. In this sense, the GP framework presented here provides a better solution by allowing sampling of coefficient functions, which are equally likely from the GP model distribution but are not a constant offset from the GP mean. An example of some of these sample functions are provided in Fig. 11. As can be seen in the figure, each sampled function is a plausible explanation of the data. If enough functions are sampled, then the corresponding values of those functions for any given input point will correspond to a normal distribution, where the mean and variance are given by the GP model at that input point.

C. Uncertainty in trim angle of attack

Apart from providing samples of the static aerodynamic coefficients, the probabilistic model presented in this work can also be used to derive consistent distributions for other key quantities of interest. For example, the trim angle of attack is an import design parameter for steady flight and is associated with an equilibrium pitching moment about the center of mass of the vehicle, defined as the solution to

$$C_{\mathrm{m,cg}}(\alpha_{\mathrm{trim}}) = 0. \tag{39}$$

Using Bayes' rule, we can write the probability that a given α induces a zero pitching moment about the center of mass as

$$p(\alpha = \alpha_{\mathrm{trim}}) = p(\alpha | C_{\mathrm{m,cg}} = 0) \propto p(C_{\mathrm{m,cg}} = 0|\alpha)p(\alpha), \tag{40}$$

where $p(C_{\mathrm{m,cg}} = 0|\alpha)$ is the likelihood that $C_{\mathrm{m,cg}} = 0$ for a given α and $p(\alpha)$ represents our prior assumption on the possible values of the trim angle of attack. The likelihood function is a direct result of the GP model for $C_{\mathrm{m,cg}}$, where

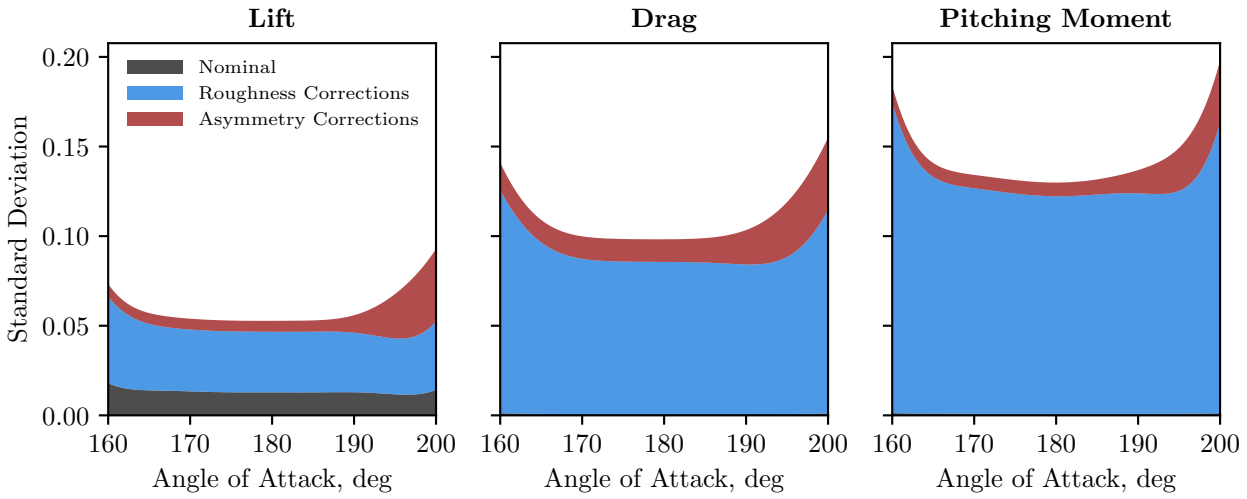


Fig. 10 Contributions to the standard deviation of the posterior on the combined nominal plus correction model for $M_\infty = 0.5$ and $Re_D = 7.5 \times 10^6$.

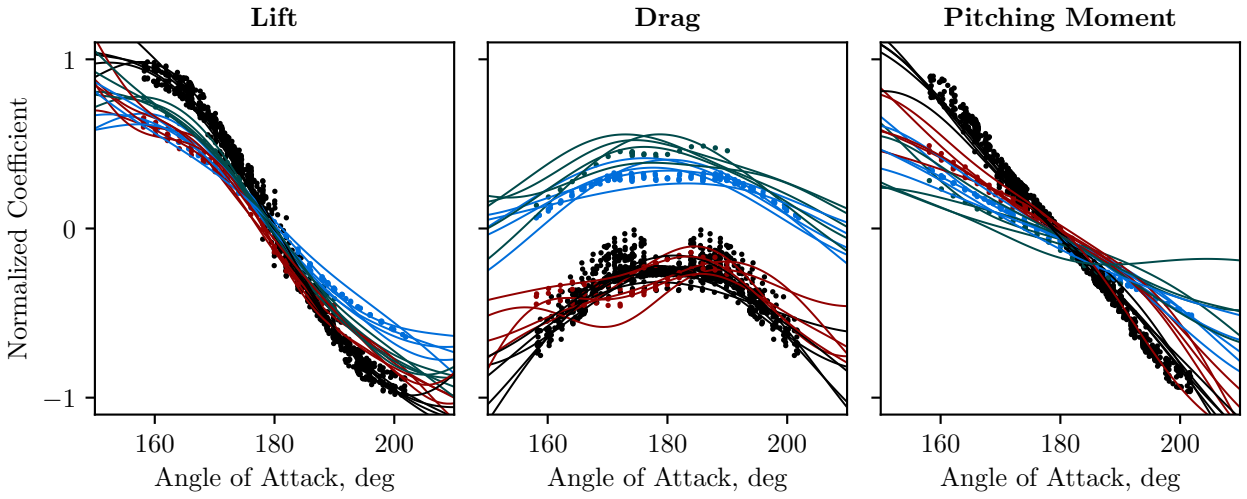


Fig. 11 Random sample C_D curves drawn from the model distributions at fixed $Re_D = 7.5 \times 10^6$ and $M_\infty = 0.3$.

for any input $(\alpha, \mathrm{M}_\infty, \mathrm{Re}_D)$, $C_{\mathrm{m,cg}}$ is normally distributed, with mean $\hat{\mu}(\alpha)$ and variance $\sigma^2(\alpha) = \hat{k}(\alpha, \alpha)$, such that

$$p(C_{\mathrm{m,cg}} = 0|\alpha) = \frac{1}{\sqrt{2\pi\sigma^2(\alpha)}} \exp\left(- \frac{\hat{\mu}(\alpha)^2}{2\sigma^2(\alpha)} \right), \tag{41}$$

where we have dropped the implicit dependence on Mach and Reynolds numbers, for convenience. For an uninformative prior on the angle of attack (e.g. a uniform distribution covering the whole α range), the resulting posterior distribution for the trim angle of attack is given by

$$p(\alpha = \alpha_{\mathrm{trim}}) = \frac{1}{Z} \frac{1}{\sqrt{2\pi\sigma^2(\alpha)}} \exp\left(- \frac{\hat{\mu}(\alpha)^2}{2\sigma^2(\alpha)} \right), \tag{42}$$

$$Z = \int \frac{1}{\sqrt{2\pi\sigma^2(\alpha)}} \exp\left(- \frac{\hat{\mu}(\alpha)^2}{2\sigma^2(\alpha)} \right) d\alpha. \tag{43}$$

The factor Z is a scaling factor to normalize the distribution which can be numerically integrated given the trained mean and variance from the GP posterior.

Example distributions for the trim angle of attack over a range of Mach numbers for fixed $\mathrm{Re}_D = 7.5 \times 10^6$ are shown in Fig. 12, where we have computed the distribution associated with each separate dataset. For reference, we have included the nominal trim angle of attack and 100 % confidence interval bounds for the v0.60 Orion aerodatabase [2]. This is not meant to be a one-to-one comparison, but rather to help provide context for the computed distributions. In fact, the v0.60 database does not include Reynolds numbers as an input dimension. Instead, the database is based on large Reynolds numbers at which the aerodynamic coefficients are assumed to independent of Reynolds number. The Reynolds number of 7.5×10^6 is significantly below this range. However, as we can see in **??**, the general behavior in the reference aerodatabase is reproduced by the multihierarchy GP model. In addition, we can see that a much tighter distribution is found for the nominal model than for the off-nominal cases. This is predominantly due to the fact that surface roughness and asymmetry effects tend to rotate the slope of the pitching moment curve such that the curve is more flat versus angle of attack (see, for example, Fig. 8). The result is that the intercept location for the pitching moment curve at a zero pitching moment has a wider uncertainty which is reflected in the distributions shown in Fig. 12.

Similar distributions can also be obtained for other derived quantities. For example, the drag coefficient at low Mach numbers tends to be a more important parameter for vehicle performance than L/D. During steady flight, angle of attack will remain near the trim value, and so we are interested in knowing the drag coefficient at trim. Under our probabilistic framework, this is easily obtained by marginalizing the drag distribution over the trim angle of attack,

$$p(C_{\mathrm{D}} = \mathrm{C}_{\mathrm{D,trim}}|\mathrm{M}_\infty, \mathrm{Re}_D) = \int p(C_{\mathrm{D}}|\alpha, \mathrm{M}_\infty, \mathrm{Re}_D) p(\alpha = \alpha_{\mathrm{trim}}|\mathrm{M}_\infty, \mathrm{Re}_D)\, d\alpha, \tag{44}$$

where the drag probability distribution is Gaussian with mean and variance given by the GP model and the trim angle-of-attack distribution is given in Eq. (42). The integral is easily computed through numerical integration. Example drag at trim distributions for the same conditions in Fig. 12 are plotted in Fig. 13, along with the reference nominal and confidence bounds from Ref. [2]. As with the trim angle-of-attack distributions, we see that the computed drag at trim distributions follow similar average trends as the reference. Interestingly, we also see a scalloping effect for the 700D Smooth distribution at low Mach numbers. This is another example of the out-of-distribution uncertainty discussed previously, because the distribution peaks occur at Mach numbers of 0.2, 0.3, and 0.4, which is where data is available. Looking at Table 2d, we see that the learned Mach length-scale for the asymmetry correction is 0.07, which is less than the distance between data in Mach space, and further explains the scalloping effect in the uncertainties for the drag coefficient at trim.

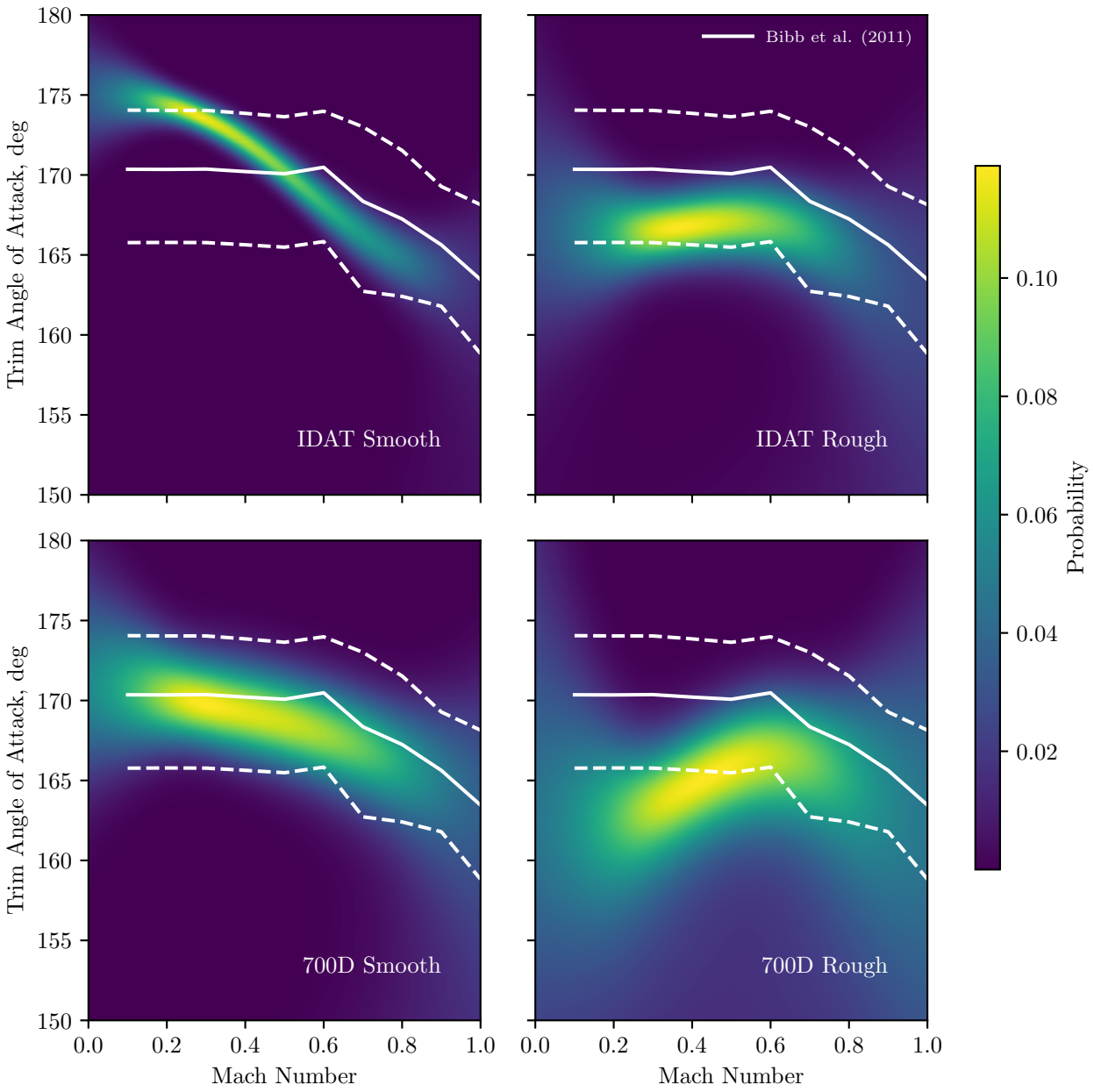


Fig. 12 Derived probability distributions for trim angle of attack over a range of Mach numbers for a fixed Reynolds number of 7.5×10^6. For reference, the nominal trim angle of attack and 100 % confidence interval bounds for the v0.60 Orion aerodatabase [2] are also included as white solid and dashed lines, respectively.

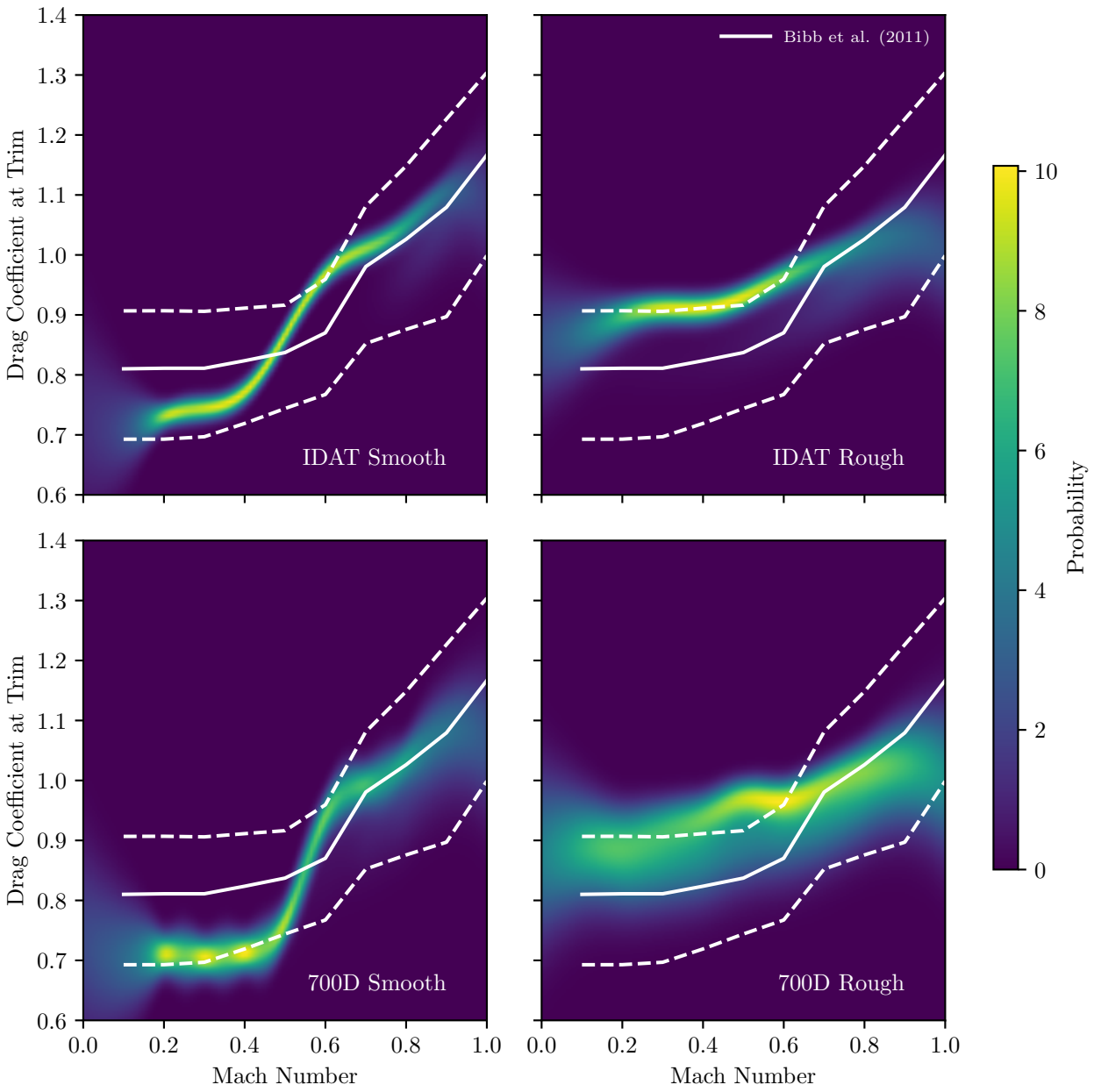


Fig. 13 Derived probability distributions for the drag coefficient at the trim angle of attack. For reference, the nominal C_D at trim and 100 % confidence interval bounds for the v0.60 Orion aerodatabase [2] are also included as white solid and dashed lines, respectively.

V. Concluding Remarks

In this paper, we have presented a consistent mathematical framework for building multihierarchy Gaussian Process models for probabilistic aerodynamic database construction. The model is a generalization of the original AR1 multifidelity framework and has been demonstrated on a dataset generated from recent wind tunnel testing of the Orion Command Module at subsonic speeds. The presented results demonstrate the implementation of the methodology and its possible use in real aerodatabase tasks such as function sampling for trajectory simulations and determining the probability of trim characteristics. We note that while we have presented a specific fidelity structure to create the example multihierarchy GP, the model framework is general enough to allow database developers to encode their own prior assumptions based on domain knowledge.

Additional improvements can also be made within this framework. For example, the correlation parameters and correction weights could be generalized to linear combinations of nonlinear basis functions, dependent on the input variables. This could provide some extra flexibility in fitting obtained flight data. This enhancement would not change the structure of the multihierarchy GP framework, but would simply add extra terms in the linear least squares problem, needed to estimate the weights. The added benefit this provides in terms of model accuracy should be weighed against the additional model complexity, based on the requirements of the user.

As presented, the multihierachy GP model provides a Gaussian distribution for the aerocoefficient of interest, which is not currently easy to incorporate into existing aerodatabases, since these generally consist of a nominal output and dispersion factors associated with various uncertainties. While the nominal (mean) output of the GP distribution can easily replace the nominal database, more work is required to incorporate the Gaussian distributions. As a half-step, dispersion coefficients could be computed as a multiple of the model standard deviation, such as the 95 % confidence interval. In the long run however, aerodatabase APIs should be updated to allow for arbitrary distributions, including computing expected values and sampling point-wise and random functions for use in Monte Carlo trajectory analyses.

Finally, determination of the correction weights is currently reliant on having some "truth" data. When no such data is available, it is less clear how these parameters should be determined. One possible approach is to build a model for previously flown vehicles that are expected to behave similarly to the one in question. For example, in the case of Orion, it is expected that the Apollo vehicle will have many similar flight characteristics. In this case, correction weights could be learned on Apollo data and then reused for Orion, perhaps with additional uncertainty included. Under a Bayesian framework, these estimates could represent the prior knowledge on the correction coefficients which could be updated as flight data becomes available. This remains out of the scope of this paper, but could be investigated as future work.

Appendix

A. Covariance structure for mixture of GPs

As we will show, the AR1 multifidelity GPR and the MHGPR models can be thought of as specific cases of a general linear mixture of Gaussian Processes which combines a set of independent GPs to form a new set of dependent GPs through a linear combination of the original set. Specifically, given n independent GPs, denoted $f_j(\boldsymbol{x}) \sim \mathcal{GP}(\mu_j(\boldsymbol{x}), k_j(\boldsymbol{x}, \boldsymbol{x}'))$, we can define m dependent stochastic processes, denoted $\hat{f}_i(\boldsymbol{x})$, as m linear combinations of these, such that $\hat{f}_i(\boldsymbol{x}) = \sum_j \phi_{ij} f_j(\boldsymbol{x})$, where $\phi_{ij} \in \mathbb{R}$ are real, linear coefficients. We now denote $\boldsymbol{f}(\boldsymbol{X}) \in \mathbb{R}^{NP} = [f_1(\boldsymbol{X})^T, \ldots, f_n(\boldsymbol{X})^T]^T$ as the random vector formed by stacking the result of each independent GP evaluated on an input matrix of p d-dimensional inputs vectors, $\boldsymbol{X} \in \mathbb{R}^{p \times d}$. Note, since each component of $\boldsymbol{f}(\boldsymbol{X})$ is distributed as a GP, $\boldsymbol{f}(\boldsymbol{X})$ is a multivariate Gaussian random vector,

$$\boldsymbol{f}(\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}), \boldsymbol{\Sigma}(\boldsymbol{X})) \tag{45}$$

where $\boldsymbol{\mu}(\boldsymbol{X}) = [\mu_1(\boldsymbol{X})^T, \ldots, \mu_n(\boldsymbol{X})^T]^T$ is the stacked vector of GP means, and $\boldsymbol{\Sigma}(\boldsymbol{X})$ is the covariance matrix. Due to the independence of each GP, the covariance matrix is block diagonal, where the j^{th} block is the covariance matrix of the j^{th} GP, $\boldsymbol{\Sigma}(\boldsymbol{X}) = \text{diag}(K_j(\boldsymbol{X}, \boldsymbol{X}))_j$, where $(K_j(\boldsymbol{X}, \boldsymbol{X}))_{lk} = k_j(\boldsymbol{X}_l, \boldsymbol{X}_k)$ and $\boldsymbol{X}_l$ is the l^{th} row of $\boldsymbol{X}$.

The corresponding mixture vector, $\hat{\boldsymbol{f}}(\boldsymbol{X}) \in \mathbb{R}^{NP} = [\hat{f}_1(\boldsymbol{X})^T, \ldots, \hat{f}_m(\boldsymbol{X})^T]^T$, is written compactly in matrix-vector form as

$$\hat{\boldsymbol{f}}(\boldsymbol{X}) = (\boldsymbol{\Phi} \otimes \boldsymbol{I}_p)\boldsymbol{f}(\boldsymbol{X}), \tag{46}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$ is the matrix of linear coefficients, where $(\boldsymbol{\Phi})_{ij} = \phi_{ij}$, and $\boldsymbol{I}_p$ denotes the $p \times p$ identity matrix. Since the mixture vector is a linear function of $\boldsymbol{f}(\boldsymbol{X})$, which is multivariate Gaussian, it is also a multivariate Gaussian vector,

such that

$$\hat{\boldsymbol{f}}(\boldsymbol{X}) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}(\boldsymbol{X}), \hat{\boldsymbol{\Sigma}}(\boldsymbol{X})), \tag{47}$$

where the mean and covariance are readily determined by their definitions. Namely,

$$\hat{\boldsymbol{\mu}}(\boldsymbol{x}) \equiv \mathbb{E}[\hat{\boldsymbol{f}}(\boldsymbol{x})] = (\boldsymbol{\Phi} \otimes \boldsymbol{I}_p)\mathbb{E}[\boldsymbol{f}(\boldsymbol{x})] = (\boldsymbol{\Phi} \otimes \boldsymbol{I}_p)\boldsymbol{\mu}(\boldsymbol{X}), \tag{48}$$

and

$$\begin{aligned} \hat{\boldsymbol{\Sigma}} = \operatorname{cov}(\hat{\boldsymbol{f}}(\boldsymbol{X}), \hat{\boldsymbol{f}}(\boldsymbol{X})) &\equiv \mathbb{E}[\hat{\boldsymbol{f}}(\boldsymbol{X})\hat{\boldsymbol{f}}(\boldsymbol{X})^T] - \mathbb{E}[\hat{\boldsymbol{f}}(\boldsymbol{X})]\mathbb{E}[\hat{\boldsymbol{f}}(\boldsymbol{X})] \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{I}_p)(\mathbb{E}[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})^T] - \mathbb{E}[\boldsymbol{f}(\boldsymbol{X})]\mathbb{E}[\boldsymbol{f}(\boldsymbol{X})^T])(\boldsymbol{\Phi}^T \otimes \boldsymbol{I}_p) \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{I}_p)\boldsymbol{\Sigma}(\boldsymbol{X})(\boldsymbol{\Phi}^T \otimes \boldsymbol{I}_p) \end{aligned} \tag{49}$$

From the above expression for the mixture vector covariance, it is trivial to show that the covariance of the two vectors formed by evaluating mixture models i and j on inputs $\boldsymbol{X}$ and $\boldsymbol{X}'$, respectively, is given as

$$\operatorname{cov}(\hat{f}_i(\boldsymbol{X}), \hat{f}_j(\boldsymbol{X}')) = \sum_k \phi_{ik}\phi_{jk}K_k(\boldsymbol{X}, \boldsymbol{X}'). \tag{50}$$

To arrive at this expression, recall that $\boldsymbol{\Sigma}(\boldsymbol{X})$ is block diagonal and note that here $\boldsymbol{X}$ can be thought of as the concatenation of the $\boldsymbol{X}$ and $\boldsymbol{X}'$ above.

Acknowledgments

The authors would like to thank the Space Technology Mission Directorate (STMD) Early Career Initiative (ECI) program for funding this work as well as the Orion Aerosciences Program for providing 133-CA wind tunnel data.

References

[1] Gnoffo, P. A., Braun, R. D., Weilmuenster, K. J., Mitcheltree, R. A., Engelund, W. C., and Powell, R. W., "Prediction and Validation of Mars Pathfinder Hypersonic Aerodynamic Data Base," *7th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, AIAA, 1998.

[2] Bibb, K., Brauckmann, G., Walker, E., and Robinson, P., "Development of the Orion Crew Module Static Aerodynamic Database, Part I: Hypersonic," *29th AIAA Applied Aerodynamics Conference*, AIAA, 2011. https://doi.org/10.2514/6.2011-3506.

[3] Bibb, K., Walker, E., Brauckmann, G., and Robinson, P., "Development of the Orion Crew Module Static Aerodynamic Database, Part II: Subsonic/Supersonic," *29th AIAA Applied Aerodynamics Conference*, AIAA, 2011. https://doi.org/10.2514/6.2011-3507.

[4] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[5] Kennedy, M. C., and O'Hagan, A., "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, Vol. 87, 2000, pp. 1–13.

[6] Le Gratiet, L., and Garnier, J., "Recursive co-kriging model for design of computer experiments with multiple levels of fidelity," *International Journal for Uncertainty Quantification*, Vol. 4, No. 5, 2014, pp. 365–386.

[7] Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N., and Karniadakis, G. E., "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 473, No. 2198, 2017, p. 20160751.

[8] Damianou, A., and Lawrence, N., "Deep Gaussian processes," *Artificial Intelligence and Statistics*, 2013, pp. 207–215.

[9] Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., and Gonzàlez, J., "Deep Gaussian processes for multi-fidelity modeling," , 2019.

[10] Brevault, L., Balesdent, M., and Hebbal, A., "Overview of Gaussian process based multi-fidelity techniques with variable relationship between fidelities," , 2020.

[11] Fernández-Godino, M. G., Park, C., Kim, N. H., and Haftka, R. T., "Review of multi-fidelity models," , 2017.

[12] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Survey of multifidelity methods in uncertainty propagation, inference, and optimizaiton," , 2018.

[13] Laurenceau, J., and Sagaut, P., "Building efficient response surfaces of aerodynamic functions with kriging and cokriging," *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 498–507.

[14] Kuya, Y., Takeda, K., Zhang, X., and Forrester, A. I. J., "Multifidelity surrogate modeling of experimental and computational aerodynamic data sets," *AIAA Journal*, Vol. 49, No. 2, 2011, p. 289–298.

[15] Toal, D. J. J., and Keane, A. J., "Efficient multipoint aerodynamic design optimization via cokriging," *Journal of Aircraft*, Vol. 48, No. 5, 2011, pp. 1685–1695.

[16] Keane, A. J., "Cokriging for Robust Design Optimization," *AIAA Journal*, Vol. 50, No. 11, 2012, p. 2351–2364.

[17] Toal, D. J. J., Keane, A. J., Benito, D., Dixon, J., Yang, J., Price, M., Robinson, T., Remouchamps, A., and Kill, N., "Multifidelity multidisciplinary whole-engine thermomechanical design optimization," *Journal of Propulsion and Power*, Vol. 30, No. 6, 2014, p. 1654–1666.

[18] Bailly, J., and Bailly, D., "Multifidelity aerodynamic optimization of a helicopter rotor blade," *AIAA Journal*, Vol. 57, No. 8, 2019, p. 3132–3144.

[19] Mukhopadhaya, J., Whitehead, B. T., Quindlen, J. F., Alonso, J. J., and Cary, A. W., "Multi-fidelity modeling of probabilistic aerodynamic databases for use in aerospace engineering," *International Journal for Uncertainty Quantification*, Vol. 10, No. 5, 2020, pp. 425–447.

[20] Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., and Zoubin, G., "Structure Discovery in Nonparametric Regression through Compositional Kernel Search," *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 28, edited by S. Dasgupta and D. McAllester, PMLR, Atlanta, Georgia, USA, 2013, pp. 1166–1174. URL https://proceedings.mlr.press/v28/duvenaud13.html.

[21] Le Gratiet, L., "Multi-fidelity Gaussian process regression for computer experiments," Ph.D. thesis, Université Paris-Diderot, 2013.

[22] CAP Aerodynamics Team, "CAP Orion MPCV Aerodynamic Database Development, Version 0.90," NASA EG-CAP-13-21, December 2017. In review.

[23] Brauckmann, G. J., Bibb, K., Ritter, J., and Moran, B., "Final Report for MPCV Test 89-CA: High-Reynolds Number Test of the Orion Crew Module in the NASA National Transonic Facility," Tech. Rep. NASA/TM-2023-xxxxxx, NASA Langley Research Center, 2013. Under preparation.

[24] Brauckmann, G., "Final Report for Test 89-CA: High-Reynolds Numbers Test of the Orion Crew Module in the NASA LaRC National Transonic Facility," CAP WTT Report EG-CAP-12-65, NASA Langley Research Center, September 2012.

[25] Fuller, D., and Gloss, B., "Guide for Users of the National Transonic Facility," TM 83124, NASA, July 1981.

[26] Braukmann, G. J., "Final Report for MPCV Test 133-CA: Orion Crew Module Heat Shields in the NASA National Transonic Facility," Tech. Rep. NASA/TM-2013-217895, NASA Langley Research Center, 2013.

[27] Murphy, K. J., Bibb, K. L., Brauckmann, G. J., Rhode, M. N., Owens, B., Chan, D. T., Walker, E. L., Bell, J. H., and Wilson, T. M., "Orion Crew Module Aerodynamic Testing," *29th AIAA Applied Aerodynamics Conference*, AIAA, 2011. https://doi.org/10.2514/6.2011-3502.