76th International Astronautical Congress (IAC 2025), Sydney, Australia, 29 Sep-3 Oct 2025.

IAC-25-D2.5.5

Uncertainty Estimation in the Aerodynamic Database of ReFEx using Bayesian Inference

Sven Krummen[a]*, **Roben Bhatti**[a], **Peter Rickmers**[a], **Viola Wartemann**[b], **Marius Franze**[b], **Dominik Neeb**[c], **Thomas Thiele**[c]

[a] *German Aerospace Center (DLR), Institute of Space Systems, Robert-Hooke-Straße 7, 28359 Bremen, Germany*

[b] *German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7, 38108 Braunschweig, Germany*

[c] *German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Linder Höhe, 51147 Köln, Germany*

* Corresponding author

Abstract

The Reusability Flight Experiment (ReFEx) is a vertical take-off horizontal landing demonstrator developed by the German Aerospace Center (DLR) to investigate technologies for aerodynamically controlled Reusable Launch Vehicles (RLVs). A key requirement for such missions is an Aerodynamic Database (AEDB) that reliably predicts aerodynamic forces and moments while quantifying the associated uncertainties, which are critical for mission and control design. Conventional AEDB generation from Computational Fluid Dynamics (CFD) and Wind Tunnel Test data relies heavily on heuristic interpolation and expert judgment, making uncertainty quantification subjective and difficult to reproduce. This study applies Bayesian inference, using Gaussian Process (GP) regression, to model aerodynamic coefficients of ReFEx and to provide principled uncertainty estimates. Both full GP models and sparse GP approximations were investigated on a representative AEDB subset covering critical phases of the re-entry trajectory. Results show that full GP models reproduce aerodynamic behavior with high accuracy and robust uncertainty bounds, while sparse GP models closely approximate their posteriors at significantly reduced computational cost. These findings demonstrate that Bayesian GP models enable objective, data-driven AEDB model generation, and extend previous applications on the vertical take-off vertical landing demonstrator CALLISTO to the vertical take-off horizontal landing configuration of ReFEx.

Keywords: ReFEx, Reusable Launch Vehicle, Aerodynamic Database, Bayesian Inference, Gaussian Process, Uncertainty Quantification

1. Introduction

The advent of Reusable Launch Vehicles (RLVs) has significantly reshaped space transportation, with both commercial and institutional actors investing in different reusability strategies. At the German Aerospace Center (DLR), two complementary flight demonstrators are currently under development to explore this technological spectrum. The *Cooperative Action Leading to Launcher Innovation for Stage Toss-back Operations (CALLISTO)* vehicle, a joint project with French National Centre for Space Studies (CNES) and Japan Aerospace Exploration Agency (JAXA), is a Vertical Take-off Vertical Landing (VTVL) demonstrator to be operated at the Guiana Space Centre (CSG) [1]. In parallel, the *Reusability Flight Experiment (ReFEx)* is being prepared as a Vertical Take-off Horizontal Landing (VTHL) re-entry experiment to be launched from the Koonibba Test Range (KTR) in Australia [2]. Together, these two demonstrators aim to provide complementary knowledge about different reusability strategies and to mature the key technologies required for future European RLV systems [3].

A crucial enabler for such vehicles is the availability of reliable Aerodynamic Databases (AEDBs), which provide the coefficients necessary to model aerodynamic forces and moments across the relevant flight envelope. For RLVs, constructing these databases is particularly challenging due to the wide range of flight conditions, the presence of control surfaces and other actuators, and the need to combine data from heterogeneous sources such as Computational Fluid Dynamics (CFD) and Wind Tunnel Test (WTT). Traditionally, the generation of AEDBs has relied heavily on manual inspection, expert judgment, and heuristic fitting of data, which made the process labor-intense and difficult to reproduce. Especially the quantification of uncertainties in both the input data and the resulting models constitutes a major challenge, despite its importance for robust system design, particularly for mission planning and the Guidance, Navigation & Control (GNC) subsystem. To ensure safe and reliable operations, engineering teams require not only accurate aerodynamic predictions but also an explicit quantification of the prediction uncertainty.

Recent studies have demonstrated that Bayesian inference provides a promising approach to tackle these challenges. In particular, Gaussian Process (GP) models have been successfully applied to estimate uncertainties in AEDBs for reusable demonstrators such as CALLISTO [4, 5]. Building on this foundation, the central research question addressed in this paper is: *Can the uncertainties in AEDBs of typical winged RLVs, such as ReFEx, be adequately modeled by (Sparse) Bayesian Gaussian Process models?*

The contribution of this study is twofold. First, we extend the Bayesian modeling approach to ReFEx, a VTHL vehicle, whereas previous studies were limited to the VTVL demonstrator CALLISTO. Second, we apply multi-input GP models to capture dependencies on several flight parameters, and systematically compare sparse GP approximations against their full GP counterparts in terms of predictive accuracy and similarity of posterior distributions. Furthermore, Section 4.7 relates these contributions to the broader literature on uncertainty quantification in aerodynamic modeling.

The remainder of this paper is structured as follows. Section 2 introduces the ReFEx system and its mission. Section 3 describes the aerodynamic dataset derived from CFD and WTT sources. Section 4 outlines the applied Bayesian modeling methodology, including GPs and their sparse approximations. The results of model fitting and evaluation are presented and discussed in Section 5. Finally, Section 6 concludes with a summary of key findings and an outlook on future work.

2. System and Mission Overview

ReFEx, currently under development at DLR and illustrated in Figure 1, pursues two main objectives. First, it will demonstrate a controlled, autonomous re-entry flight from hypersonic to subsonic speeds, thereby replicating essential phases of a winged RLV first-stage [2, 6]. Key aspects include validation of GNC strategies, aerodynamic design for stable flight across widely varying flow regimes, and the transition from exo-atmospheric to aerodynamic control. A defining maneuver during re-entry flight is a heading change exceeding 30°, which represents a critical capability for future RLV stages that must precisely return to designated landing zones [7, 8].

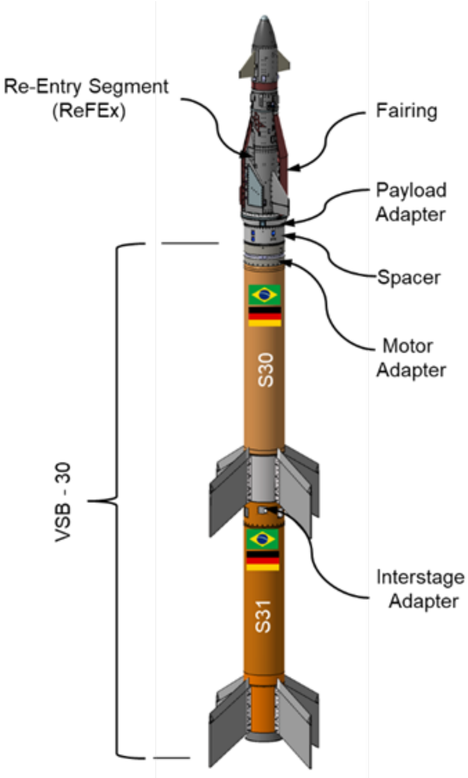

Fig. 1: ReFEx launch configuration on VSB-30.

The second objective of ReFEx is the maturation of enabling technologies for European reusable launch systems [6]. These include, but are not limited to, autonomous trajectory optimization, advanced flight instrumentation such as fiber-optic sensors and Flush Air-Data Sensing (FADS), and methods for generating high-fidelity aerodynamic data. The flight will deliver unique in-flight data to validate computational models, improve design methodologies, and reduce development risk for future RLVs. By relying exclusively on aerodynamic control surfaces for its return, ReFEx complements DLR's VTVL demonstrator CALLISTO, thus covering both major reusability concepts and building a broader European technology base [3].

As depicted in Figure 2, ReFEx will be launched atop a VSB-30 sounding rocket, reaching altitudes and velocities representative of a first-stage separation before entering its experimental phase [6, 9]. Following booster burnout, the vehicle is de-spun, the payload fairing is jettisoned, and ReFEx is released. After wing deployment and locking,

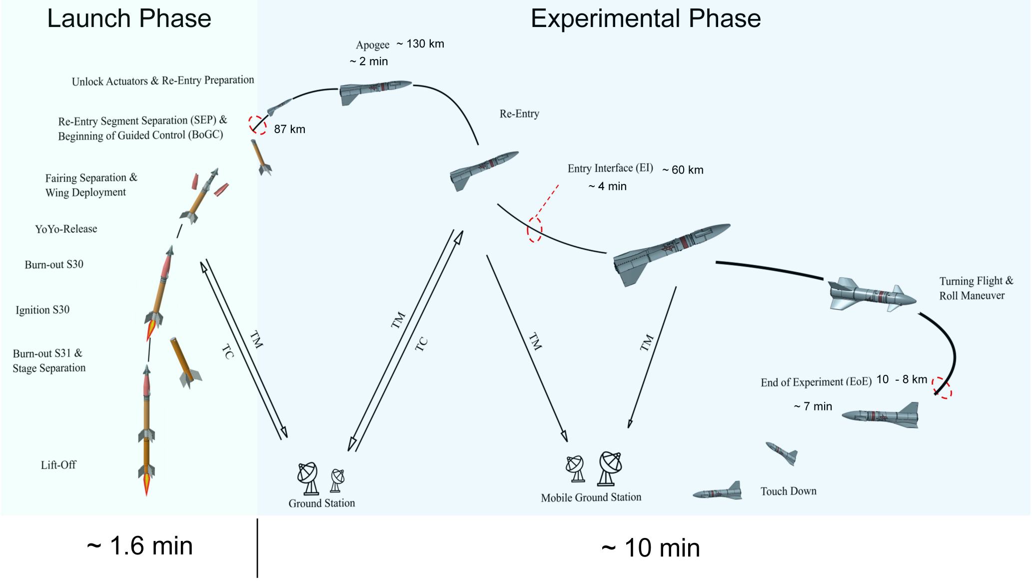

Fig. 2: ReFEx mission events and timeline.

control authority gradually transitions from the cold-gas reaction control system to the aerodynamic surfaces (canards and rudder) as dynamic pressure increases. The main mission challenge is to ensure stability and maneuverability during this transition, while covering the full flight envelope from hypersonic through subsonic regimes. Aerodynamic difficulties include control reversals at high angles of attack, ensuring trimmability across a wide Mach-altitude range, and executing the required heading change of more than 30° [7, 10]. Meeting these challenges demands a robust aerodynamic design, supported by a comprehensive AEDB, and an adaptive GNC system capable of maintaining precise trajectory control while handling extreme thermal and mechanical loads.

3. Aerodynamic Dataset

Accurate aerodynamic characterization is essential to define the flight envelope of a vehicle, to assess its controllability, and to support guidance and control design. For this purpose, an AEDB is commonly established which provide aerodynamic force and moment coefficients across all relevant flight conditions, covering combinations of velocity, altitude, control surface deflections, orientation and rotation rates. In the case of ReFEx, the AEDB was built from complementary CFD simulations and WTT campaigns conducted during vehicle development. The following subsections summarize these data sources, before introducing the representative subset of the AEDB that is used in this study as the basis for Bayesian uncertainty modeling.

3.1 Computational Fluid Dynamics

Several CFD simulation campaigns were carried out during the ReFEx development to both support the aerodynamic design and provide the aerodynamic coefficients required for flight dynamics and GNC analyses. The design process followed an iterative approach, where partial datasets of evolving vehicle geometries were computed until convergence toward the final configuration was achieved [11]. For this final geometry, a comprehensive dataset covering all relevant parameter variations was generated, enabling aerodynamic control design throughout the entire mission envelope.

For ReFEx, the list of parameters that describe the flight condition consists of Mach number, angle of attack, sideslip angle, altitude, canard deflections (left and right), rudder deflection, and rotation rates. The database spans the full flight regime, with $0.5 < Ma < 5.5$ and $-45^\circ < \alpha < 10^\circ$, supplemented by selected corner cases. In total, approximately 2,400 CFD data points were produced for the final AEDB [12].

All simulations were performed with the DLR TAU solver [13], a three-dimensional parallel hybrid multigrid code, validated for subsonic, transonic, and hypersonic flows [14–16]. It solves the Reynolds-Averaged Navier-Stokes (RANS) equations using a second-order finite-volume method optimized for High-Performance Computing (HPC) systems. For ReFEx, the one-equation Spalart-Allmaras turbulence model was used [17], whereas the influence of different turbulence model has also been investigated [18].

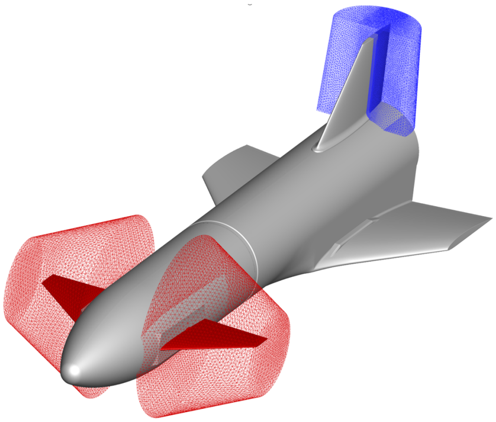

Fig. 3: Movable CFD mesh blocks for ReFEx' control surfaces, reprinted from [12].

To reduce the number of required meshes, a mesh-in-mesh technique was applied. This allows localized grids for movable surfaces, such as the rudder and canards as shown in Figure 3, to be translated or rotated relative to the main mesh s long as geometric and overlap requirements are fulfilled.

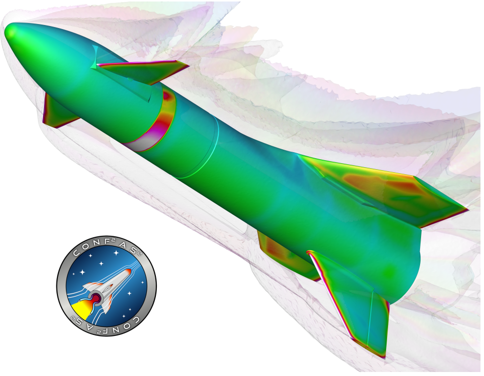

Fig. 4: Coupled Numerical Fluid-Flight Mechanics and Structure Simulation ($\mathrm{CoNF^2aS^2}$) Fluid-Structure Interaction (FSI) simulation of ReFEx' belly-up flight phase.

As an example of advanced CFD analysis conducted for ReFEx, Figure 4 shows a $\mathrm{CoNF^2aS^2}$ simulation of the belly-up flight phase [19]. In addition to predicting aerodynamic behavior, these simulations assessed thermal loading on sensitive components such as the Global Navigation Satellite System (GNSS) antenna wraped around the vehicle. The results confirmed that critical temperatures remain below material limits, with structural elements inside the vehicle acting as effective heat sinks.

3.2 *Wind Tunnel Tests*

In parallel to the CFD activities, several WTT campaigns were carried out in the Trisonic Wind Tunnel (TMK) at DLR Cologne to characterize the aerodynamic behavior of ReFEx and to provide experimental reference data. The objectives were to assess static and dynamic stability of the vehicle, and to calibrate the FADS system.

The TMK is a blow-down wind tunnel with a $0.6\,\mathrm{m} \times 0.6\,\mathrm{m}$ test section and covers Mach numbers from 0.5 to 5.7. It enables test durations of up to $60\,\mathrm{s}$, with rapid Mach number changes possible within supersonic runs. For the WTT campaigns, a $1:13$ scale ReFEx model was developed, which is shown in Figure 5. The modular design allowed variation of canard deflections ($0°$, $\pm5°$, $\pm10°$, $\pm15°$) and rudder deflections ($0°$, $5°$, $10°$) on this model.

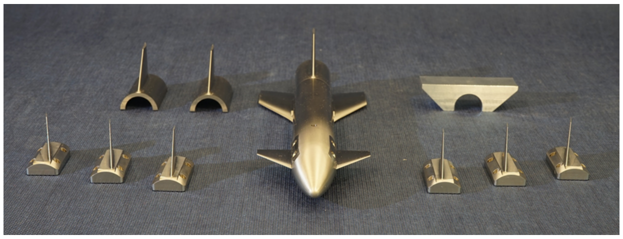

Fig. 5: ReFEx wind tunnel model for 6-DoF static and dynamic stability measurements.

Static stability tests were performed at Mach numbers $1.4 - 4.5$ and Reynolds numbers between $7 - 15$ million [20]. Angles of attack up to $40°$ and sideslip angles of $5°$ were tested in both belly-up and belly-down orientations. The model was instrumented with a six-component balance, and Schlieren imaging provided additional flow visualization as shown in Figure 6. A total of 84 runs were conducted to cover significant parts of ReFEx' flight envelope. Uncertainties of the aerodynamic conditions were estimated from proper calibrations of the balance and wind tunnel uncertainties. The retrieved data allowed the identification of static stability boundaries and regions of potential rudder reversal [18, 20]. The dataset also provided reference points to anchor CFD results and to investigate potential Reynolds and sting effects [18].

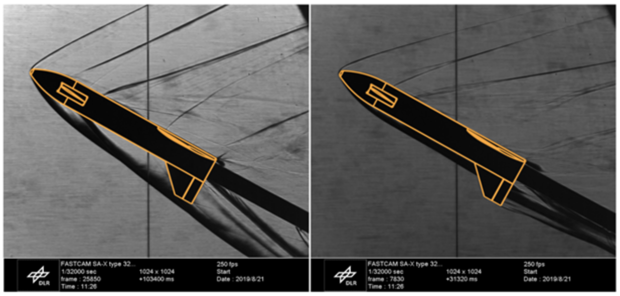

Fig. 6: Schlieren images of the ReFEx model during a Mach 2.5 (left) and Mach 4.5 (right) run; canard angle $-5°$, angle of attack $28°$.

Dynamic stability tests were performed with a modified model at Mach numbers $1.4-2.5$ and canard deflections of $0°$, $5°$, and $10°$. A total of 22 runs were completed with the free-oscillation measurement technique near the vehicle trim conditions. The aerodynamic derivatives were determined both from cross-spring measurements and from Schlieren imaging, showing excellent agreement. Uncertainties in the coefficients were estimated from proper calibrations and wind tunnel uncertainties, as well as from repeated model release during one run. The obtained results confirmed that ReFEx is longitudinally dynamically stable under the tested conditions, with damping increasing towards lower Mach numbers. The aerodynamic stiffness coefficients were consistent with those obtained in the static campaign, reinforcing the reliability of the measurements.

Finally, a dedicated FADS calibration campaign was performed using a $1:3$ scale model of ReFEx' nose section equipped with multiple pressure ports. Across 84 runs, angles of attack between $-45°$ and $20°$, sideslip angles up to $12°$, and Mach numbers from 0.4 to 4.5 were tested. This dataset enables in-flight reconstruction of angle of attack, sideslip angle, and dynamic and static pressure of the free stream based on FADS the measurements.

3.3 Data Subset Definition

The AEDB of ReFEx, generated through the CFD and WTT campaigns, contains more than 10,000 samples in total, covering a wide range of variables that characterize the vehicle's aerodynamic behavior across different flight conditions. The most relevant independent parameters include the Mach number (Ma), angle of attack (α), sideslip angle (β), rotational rates (p, q, r), canard deflections (η_r, η_l), and rudder deflection (ζ), as well as the aerodynamic force coefficients (c_x, c_y, c_z) and moment coefficients (c_{Mx}, c_{My}, c_{Mz}) as dependent parameters.

For this study, a targeted selection was made to reduce the complexity and computational requirements of the analysis, enabling a focused investigation of the most relevant aerodynamic effects within a reduced yet representative parameter space. The dedicated AEDB subset was defined by fixing the following conditions:

- Fixed Mach number: $Ma = 1.7$
- No vehicle rotation: $p = q = r = 0\,\mathrm{rad\,s^{-1}}$
- Symmetric canard deflections: $\eta_r = \eta_l =: \eta$
- No rudder deflection: $\zeta = 0°$

This subset includes 91 CFD samples and 670 WTT samples, resulting in 761 data points, with α & η remaining as independent parameters. Compared to CFD, the WTT samples are denser due to finer sampling in α and cover a wider range as shown in Figure 7. This subset was chosen because it corresponds to the roll maneuver phase of ReFEx' descent trajectory, a flight regime where aerodynamic control is especially critical. With the same rationale, the Normal Force Coefficient c_z was selected as the most relevant dependent variable, as it dominates lift and stability during the roll maneuver. Furthermore, this flight regime offers reliable data coverage from both WTT and CFD sources, ensuring robust model training and validation. Due to this subset selection, the aerodynamic modelling task can focus on the mapping $(\alpha, \eta) \mapsto c_z$.

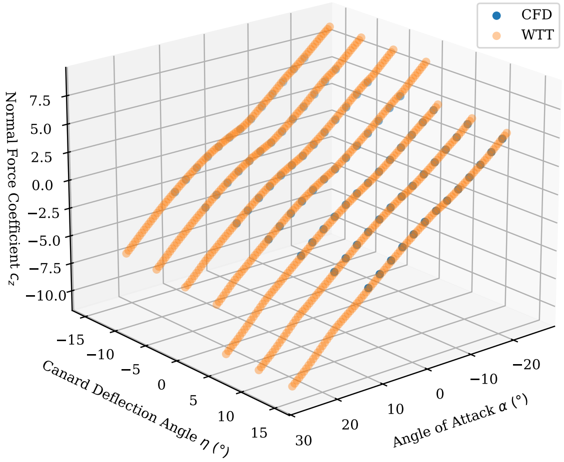

Fig. 7: Scatter plot of selected AEDB subset as used in this study. Left panel highlights slice of dataset for $\eta = 10°$.

4. Methods

4.1 Problem Setup & Notation

The creation of the aerodynamic model for ReFEx can be formalized as a regression task, aiming to learn a mapping function from the flight conditions to the aerodynamic coefficients of the vehicle. [4,5] As described in Section 3, the outputs of this function are the aerodynamic force and moment coefficients, which can be conveniently concatenated into a *coefficient vector* $\boldsymbol{c}$:

$$\boldsymbol{c} := \left(c_x, c_y, c_z, c_{Mx}, c_{My}, c_{Mz}\right) \tag{1}$$

These coefficients depend on the flow conditions, vehicle orientation and angular rates, control surface deflections, and the used data source. For simplicity, these dependencies can be aggregated to a *generalized input vector* $\boldsymbol{x}$. In line with the AEDB as described in Section 3, the following parametrization of inputs is adopted for ReFEx:

$$\boldsymbol{x} := (Ma, \alpha, \beta, p, q, r, \eta_r, \eta_l, \zeta, src) \tag{2}$$

Here, $src \in \{\mathrm{CFD}, \mathrm{WTT}, \mathrm{FLIGHT}\}$ represents a categorical indicator of the data source, which allows to consider source-specific uncertainties in the model. The FLIGHT is a specially unobserved category, which will be used for predictions as discussed in Section 4.5.

In a classical non-Bayesian setting, the goal of aerodynamic modelling would be to seek a deterministic mapping f from inputs $\boldsymbol{x}$ to output $\boldsymbol{c}$. Any regression parameters introduced by this model, collected in the *generalized parameter vector* $\boldsymbol{\theta}$, shall be chosen so that the observed AEDB data $(\tilde{\boldsymbol{c}}_i, \tilde{\boldsymbol{x}}_i) \in \mathcal{D}$ is well-approximated by this function:

$$\boldsymbol{c} = f(\boldsymbol{x}; \boldsymbol{\theta}) \tag{3}$$

Even though this classical approach is widely used in practice, often guided by expert heuristics, it does not provide an explicit quantification of uncertainties in the model and the observations. Consequently, a Bayesian alternative for AEDB modelling has been derived in previous work, which enables a principled representation and propagation of uncertainty. [4,5] In this formulation, the objective is to learn a conditional probability distribution over the aerodynamic coefficients $\boldsymbol{c}$, given the input vector $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$:

$$\boldsymbol{c} \sim p(\boldsymbol{c} \mid \boldsymbol{x}; \boldsymbol{\theta}) \tag{4}$$

In such a Bayesian setting, also the model parameters $\boldsymbol{\theta}$ are treated as random variables. Given the observed data $(\tilde{\boldsymbol{c}}_i, \tilde{\boldsymbol{x}}_i) \in \mathcal{D}$ and any fixed *hyperparameters* $\boldsymbol{\epsilon}$ (which encode modelling choices not updated during inference), the parameter distributions can be updated via application of Bayes' theorem. This update, commonly called *Bayesian Inference* or *Training*, yields the *posterior distribution* of the parameters $\boldsymbol{\theta}$:

$$\begin{aligned}
&\overbrace{p(\boldsymbol{\theta} \mid (\tilde{\boldsymbol{c}}_i, \tilde{\boldsymbol{x}}_i)_{i=1..|\mathcal{D}|}, \boldsymbol{\epsilon})}^{\text{posterior}} \\
&\quad = \frac{\overbrace{p(\tilde{\boldsymbol{c}}_{i=1..|\mathcal{D}|} \mid \tilde{\boldsymbol{x}}_{i=1..|\mathcal{D}|}; \boldsymbol{\theta}, \boldsymbol{\epsilon})}^{\text{likelihood}} \; \overbrace{p(\boldsymbol{\theta} \mid \boldsymbol{\epsilon})}^{\text{prior}}}{\int p(\tilde{\boldsymbol{c}}_{i=1..|\mathcal{D}|} \mid \tilde{\boldsymbol{x}}_{i=1..|\mathcal{D}|}; \boldsymbol{\theta}, \boldsymbol{\epsilon}) \; p(\boldsymbol{\theta} \mid \boldsymbol{\epsilon}) \, d\boldsymbol{\theta}}
\end{aligned} \tag{5}$$

Here, the *likelihood function* expresses how likely the data are for different parameter values, while the *prior distribution* captures the model knowledge or assumptions before any data is observed. In practice, weakly informative priors are often preferred. These priors avoid assigning probability mass to implausible regions of parameter space, while still maintaining enough flexibility to let the data drive the inference [21].

Once the posterior distribution has been inferred, it can be used to generate out-of-sample predictions for yet unobserved flight conditions $\hat{\boldsymbol{x}}$ via the *posterior predictive distribution*:

$$\begin{aligned}
&p(\hat{\boldsymbol{c}} \mid \hat{\boldsymbol{x}}; (\tilde{\boldsymbol{c}}_i, \tilde{\boldsymbol{x}}_i)_{i=1..|\mathcal{D}|}, \boldsymbol{\epsilon}) \\
&\quad = \int p(\hat{\boldsymbol{c}} \mid \hat{\boldsymbol{x}}; \boldsymbol{\theta}) \; p(\boldsymbol{\theta} \mid (\tilde{\boldsymbol{c}}_i, \tilde{\boldsymbol{x}}_i)_{i=1..|\mathcal{D}|}, \boldsymbol{\epsilon}) \, d\boldsymbol{\theta}
\end{aligned} \tag{6}$$

Similarly, the *prior predictive distribution* characterizes predictions before any data has been used for inference. This is commonly used to check the plausibility of model assumptions, which are encoded through the choice of likelihood function and prior distributions:

$$p(\hat{\boldsymbol{c}} \mid \hat{\boldsymbol{x}}; \boldsymbol{\epsilon}) = \int p(\hat{\boldsymbol{c}} \mid \hat{\boldsymbol{x}}; \boldsymbol{\theta}, \boldsymbol{\epsilon}) \; p(\boldsymbol{\theta} \mid \boldsymbol{\epsilon}) \, d\boldsymbol{\theta} \tag{7}$$

As discussed in [5], this Bayesian framework admits a variety of aerodynamic modelling choices. In this study, we focus on GP models only, which is one of the broadest and most versatile classes of Bayesian models for AEDB regression.

4.2 Gaussian Processes

GPs provide a flexible, non-parametric, and widely used class of probabilistic models [22]. Conceptually, they generalize the multivariate normal distribution to an infinite-dimensional setting, thereby defining a probability distribution over functions:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{8}$$

where $m(\mathbf{x})$ is the *mean function* and $k(\mathbf{x}, \mathbf{x}')$ the *covariance function*, also called *kernel*, of the GP.

For any finite collection of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ (e.g. training data), the corresponding function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$ follow, by definition, a multivariate normal distribution

$$\mathbf{f} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}) \tag{9}$$

with mean vector $\mathbf{m} = m(\mathbf{X})$ and covariance matrix $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$.

In most applications, the mean function typically plays a minor role and is therefore often set to zero. The kernel, however, encodes key characteristics of the represented functions, such as smoothness, stationarity, and other structural properties. Standard kernel families are well documented in the literature and can be combined through transformations (e.g. rescaling, convolution, warping) or compositions (sum, product) to tailor the GP to specific data characteristics [22].

In this study, we investigate several Bayesian GP models based on different covariance functions, as detailed in Section 4.6. The kernels used were selected and composed from the widely applied families summarized in Table 1. The main differences between these kernel families lie in their assumptions about the smoothness of the modeled function. The White Noise (WN) kernel represents uncorrelated, discontinuous functions. The Matérn family introduces controllable smoothness: for $\nu = 1/2, 3/2, 5/2$, the resulting functions are continuous and $(\lceil \nu \rceil - 1)$-times differentiable, with the limit $\nu \rightarrow \infty$ yielding the infinitely differentiable Squared-Exponential (SE) kernel. The Rational Quadratic (RQ) kernel can be interpreted as a scale mixture of SE kernels with varying length scales. Finally, the Intrinsic Coregionalization (IC) kernel encodes categorical data and is used here to represent the different error characteristics of the data sources in the AEDB.

For variables defined on periodic domains, such as angle of attack α or canard deflection angle η, a periodic input warping is applied to enforce cyclical continuity:

$$\begin{aligned} k^\circ(x, x') &= k\left(\mathbf{u}(x), \mathbf{u}(x')\right) \\ \mathbf{u}(x) &= (\sin x, \cos x)^\top \end{aligned} \tag{10}$$

This transformation converts any stationary kernel k into a *periodic kernel* k°, ensuring that the GP respects the cyclic nature of the underlying data.

4.3 Sparse Gaussian Process Approximation

A major limitation of GPs lies in their computational cost. Inference requires inverting the full $n \times n$ covariance matrix, which scales as $O(n^3)$ in time and $O(n^2)$ in memory. This quickly becomes infeasible as the dataset size n increases.

To mitigate this, sparse approximations introduce a smaller set of $m \ll n$ *inducing points*, which act as representative locations in the input space [23]. Conditioning on these inducing points yields a low-rank approximation of the covariance matrix, where $\mathbf{Z}$ denotes the vector of inducing inputs:

$$Q_{\mathbf{XX}} = K(\mathbf{X}, \mathbf{Z}) K(\mathbf{Z}, \mathbf{Z})^{-1} K(\mathbf{Z}, \mathbf{X}) \tag{11}$$

Since only the $m \times m$ covariance $K(\mathbf{Z}, \mathbf{Z})$ needs to be inverted, the overall computational cost is reduced to $O(nm^2 + m^3)$, which is more tractable when $m \ll n$.

Several sparse approximation schemes build on this construction by modifying how $Q_{\mathbf{XX}}$ is combined with the remaining variance terms:

Deterministic Training Conditional (DTC) replaces the full covariance $K_{\mathbf{XX}}$ with the low-rank approximation $Q_{\mathbf{XX}}$. While this greatly reduces computational cost, it ignores the residual variance and systematically underestimates predictive uncertainty [24]. For this reason, DTC is not further considered in this study.

Fully Independent Training Conditional (FITC) improves upon DTC by restoring the diagonal elements of the covariance:

$$K_{\text{FITC}} \approx Q_{\mathbf{XX}} + \operatorname{diag}[K_{\mathbf{XX}} - Q_{\mathbf{XX}}] \tag{12}$$

This preserves the marginal variances of the training data points while retaining a low-rank structure for off-diagonal terms.

Variational Free Energy (VFE) adopts a variational viewpoint, yielding an approximation of the form

$$K_{\text{VFE}} \approx Q_{\mathbf{XX}} + \sigma_n^2 I, \tag{13}$$

together with an additional trace correction for the residual $K_{\mathbf{XX}} - Q_{\mathbf{XX}}$. This ensures a principled approximation that maintains a valid lower bound to the full GP, while avoiding the overconfident behavior of DTC.

Besides the approximation scheme, the quality of sparse approximations strongly depends on the placement of the inducing points. In this study, three strategies are considered:

Table 1: List of common simple GP kernels used in this study.

Name	**Kernel** $k(x, x')$
White Noise (WN)	$k_{\mathrm{WN}}(x, x') = \delta_{x,x'}$
Matérn-1/2 ($\mathrm{M}_{1/2}$)	$k_{\mathrm{M1/2}}(x, x'; \ell) = \exp\left(-\frac{\lVert x-x' \rVert}{\ell}\right)$
Matérn-3/2 ($\mathrm{M}_{3/2}$)	$k_{\mathrm{M3/2}}(x, x'; \ell) = \left(1 + \frac{\sqrt{3}\lVert x-x' \rVert}{\ell}\right) \exp\left(-\frac{\sqrt{3}\lVert x-x' \rVert}{\ell}\right)$
Matérn-5/2 ($\mathrm{M}_{5/2}$)	$k_{\mathrm{M5/2}}(x, x'; \ell) = \left(1 + \frac{\sqrt{5}\lVert x-x' \rVert}{\ell} + \frac{5\lVert x-x' \rVert^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}\lVert x-x' \rVert}{\ell}\right)$
Squared-Exponential (SE)	$k_{\mathrm{SE}}(x, x'; \ell) = \exp\left(-\frac{\lVert x-x' \rVert^2}{2\ell^2}\right)$
Rational Quadratic (RQ)	$k_{\mathrm{RQ}}(x, x'; \ell, \beta) = \left(1 + \frac{\lVert x-x' \rVert^2}{2\beta\ell^2}\right)^{-\beta}$
Intrinsic Coregionalization (IC)	$k_{\mathrm{IC}}(x, x'; B) = B_{x,x'}$

K-means Clustering partitions the training data into m clusters using the K-means algorithm [25]. The cluster centroids are then used as inducing inputs. Since centroids are drawn toward regions of high density, this approach allocates more inducing points where the data is concentrated.

Farthest Point Sampling (FPS) is a greedy algorithm designed to maximize spatial coverage [26]. Starting from a randomly chosen point, subsequent inducing points are selected iteratively such that each new point is the farthest (in Euclidean distance) from all previously chosen ones.

Halton Sequence Sampling applies a low-discrepancy quasi-random sequence as inducing points. This Halton sequence is commonly used in numerical integration and space-filling designs [27]. Compared to purely random sampling, Halton sequences provide more uniform coverage of the data domain. In practice, a scrambled version is often used to avoid structural artifacts and improve uniformity.

These methods reflect different trade-offs between local density representation (K-means), space-filling coverage (FPS), and quasi-random uniformity (Halton), and are compared in this study on their impact on predictive accuracy and uncertainty quantification.

4.4 Model Training & Diagnostics

Bayesian inference typically requires approximating the posterior distribution defined in (5), which is in most cases intractable to compute analytically. The common approach for this task is Markov Chain Monte Carlo (MCMC) sampling, which provides asymptotically exact samples from the posterior and has become the de facto standard in many scientific applications. Alternative methods, such as Variational Inference (VI), offer potentially faster but approximate solutions by fitting a simpler distribution to the posterior. In this work, MCMC has been adopted in order to fully capture the complex posterior dependencies arising in aerodynamic regression models.

MCMC sampling approximates the posterior distribution by generating multiple chains of dependent samples. While conceptually straightforward, practical challenges arise when exploring regions of high curvature or strong correlations in the parameter space, where standard Metropolis-Hastings schemes may mix poorly. To address this, we employed the state-of-the-art Hamiltonian Monte Carlo (HMC) method with the No-U-Turn Sampler (NUTS) as implemented in the `PyMC` framework [28]. This algorithm efficiently explores the posterior by leveraging gradient information of the log-likelihood and automatically adapts path lengths to avoid redundant trajectories.

To ensure validity of the inferred posterior, multiple convergence diagnostics were monitored. First, the number of divergences during sampling as indicated by the NUTS algorithm was tracked and kept at zero for accepted models. Second, the Potential Scale Reduction Factor ($\hat{R}$) was required to remain below 1.01, indicating sufficient convergence across chains. Third, the Effective Sample Size (ESS) was evaluated both in the bulk and the tails of the distribution, requiring values to exceed 400 for stable estimation of posterior quantities. These thresholds are commonly used in practical applications [29]. In addition to these quantitative metrics, visual checks of prior predictive and posterior predictive distributions were performed to verify physical plausibility and adequate coverage of the data.

4.5 Model Evaluation

The primary criterion to assess the quality of an aerodynamic database model is its predictive performance. For the ReFEx dataset, no specific error metric tailored to capture development risks was prescribed, so the Root Mean Squared Error (RMSE) is considered an appropriate generalized metric. RMSE balances sensitivity to outliers with overall error magnitude, and provides a direct and interpretable measure of the average deviation between predicted and reference aerodynamic coefficients. Other error measures, such as the Maximum Residual Error or the Median Absolute Deviation, have also been analyzed in previous studies [5] and could be considered to complement RMSE, but are not focused here due to their limited additional value.

A well-known risk in model assessment is overfitting when predictive quality is judged solely on in-sample residuals. To mitigate this risk, performance in this study is also evaluated for out-of-sample predictions based on cross-validation. Two training setups are therefore defined: the *Nominal* training setup, where all available data are used to fit the model, and the *Stratified K-Fold Cross-Validation (K-Fold)* setup, where the dataset is partitioned into $k = 5$ folds that are stratified across data sources to ensure balanced coverage of aerodynamic data. The latter setup allows a reliable estimate of predictive generalization error via predicting the hold-out data set for each fold.

In this study, predictive quality is primarily evaluated on the FLIGHT data class. This choice is motivated by engineering relevance: while CFD and WTT data are crucial for model construction, they inherently include systematic source-dependent errors. The FLIGHT class represents the most realistic vehicle configuration and enables assessment of the model's generalization capability to yet unobserved conditions. For brevity in this study, the training and prediction setups are occasionally reported together, e.g. *Nominal(FLIGHT)*.

For sparse GP approximations, the evaluation requires a slightly different perspective. The central objective here is not only to achieve computational efficiency, but also to demonstrate that the sparse models remain faithful approximations of the underlying full GP model. Hence, prediction errors of sparse models should closely match those of the corresponding full GP models if sufficient similarity can be ensured.

To assess the fidelity of the sparse GP models, their posterior distributions have directly been compared to those of the corresponding full GP. Here, three complementary similarity metrics are applied: the *Wasserstein Distance* [30], the *Energy Distance* [31], and the *Kullback-Leibler (KL) Divergence* [32]. Together, these measures quantify how closely the sparse posterior matches the full posterior in terms of distributional shape, spread, and information content. This way, they provide robust means of assess which sparse approximations are suitable surrogates for the full GP model.

4.6 Study Setup

4.6.1 Full GP Models

This study investigates several Bayesian GP models with different covariance functions to evaluate their suitability for representing ReFEx' AEDB. The modelling approach builds on earlier work for CALLISTO [4, 5].

Adapting the general problem formulation from Section 4.1 to the AEDB subset defined in Section 3.3, the coefficient and input vector simplify to:

$$\boldsymbol{c} = (c_z) \tag{14}$$

$$\boldsymbol{x} = (\alpha, \eta, src) \tag{15}$$

All investigated models assume a GP prior with constant zero mean and a model-specific covariance function k:

$$\boldsymbol{c}(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}')) \tag{16}$$

In total, 16 full GP models were tested, as listed in Table 2. It should be noted that the models *GPModel1* to *GPModel4* depend only on α and were included primarily to verify reproducibility with earlier work [5]. As they cannot adequately represent the two-dimensional (α, η) dataset, they are not further considered in this paper.

The covariance functions of all models share a common structure, as can be seen for *GPModel9*:

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') = {} & \underbrace{\mu^2 k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha'; \ell_{f,\alpha}) k^{\circ}_{\mathrm{M5/2}}(\eta, \eta'; \ell_{f,\eta})}_{\text{nominal}} \\ & + \underbrace{k_{\mathrm{IC}}(src, src'; \boldsymbol{\sigma})^2 k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha'; \ell_{e,\alpha}) k^{\circ}_{\mathrm{M1/2}}(\eta, \eta', \ell_{e,\eta})}_{\text{error}} \\ & + \underbrace{\nu^2 k_{\mathrm{WN}}(\boldsymbol{x}, \boldsymbol{x}')}_{\text{noise}} \end{aligned} \tag{17}$$

Here, the *nominal kernel* captures the nominal dependency of the aerodynamic coefficients on the flight conditions, the *error kernel* accounts for source-dependent systematic errors in the training data, and the *noise kernel* models global measurement noise in the data while improving numerical stability of the inference algorithms.

Table 2: List of investigated GP models with different structures of covariance function.

Model	**Covariance Function** $k(\boldsymbol{x}, \boldsymbol{x}')$
GPModel1	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel2	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel3	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel4	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{WN}}(\alpha, \alpha') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel5	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{SE}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{SE}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel6	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{SE}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M1/2}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel7	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{SE}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M1/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k_{\mathrm{WN}} + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel8	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M5/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M5/2}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel9	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M5/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M1/2}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel10	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M5/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M3/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M3/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k_{\mathrm{WN}} + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel11	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{M3/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M3/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M1/2}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel12	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{M5/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M3/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M1/2}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel13	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M3/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M3/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M3/2}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k_{\mathrm{WN}} + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel14	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{RQ}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{RQ}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M1/2}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel15	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{SE}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{RQ}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{M1/2}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{M1/2}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$
GPModel16	$k(\boldsymbol{x}, \boldsymbol{x}') = \mu^2 \cdot k^{\circ}_{\mathrm{RQ}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{SE}}(\eta, \eta') + k_{\mathrm{IC}}(src, src')^2 \cdot k^{\circ}_{\mathrm{RQ}}(\alpha, \alpha') \cdot k^{\circ}_{\mathrm{RQ}}(\eta, \eta') + \nu^2 \cdot k_{\mathrm{WN}}$

For all kernel parameters, weakly-informative prior distributions were assigned to guide inference towards physically meaningful ranges while preserving model flexibility, as commonly recommended for Bayesian modelling [21]. Length-scales ℓ and amplitudes μ, σ follow inverse Gamma distributions, while the noise variance ν and the scale parameter β (for RQ kernels) follow Gamma distributions.

$$\begin{aligned} \ell &\sim \mathrm{InvGamma}(\mu_\ell, \sigma_\ell) \\ \mu &\sim \mathrm{InvGamma}(\mu_\mu, \sigma_\mu) \\ \sigma_{src} &\sim \mathrm{InvGamma}(\mu_{\sigma_{src}}, \sigma_{\sigma_{src}}) \\ \nu &\sim \mathrm{Gamma}(\mu_\nu, \sigma_\nu) \\ \beta &\sim \mathrm{Gamma}(\mu_\beta, \sigma_\beta) \end{aligned} \tag{18}$$

These priors were defined independently for each kernel and input dimension. Also, source-dependent errors were assumed to be uncorrelated between CFD and WTT data, so the IC kernel adopts a diagonal structure with $\boldsymbol{\sigma} = \mathrm{diag}(\sigma_{src})$.

All models were trained both in the *Nominal* and the *K-Fold* setups defined in Section 4.5, while predictions were made for the three data classes $src = \mathrm{CFD}, \mathrm{WTT}, \mathrm{FLIGHT}$.

Table 3: List of investigated sparse GP models with different approximation schemes, inducing point selection algorithms, and percentages of inducing point compared to full dataset.

Model	**Scheme**	**Algorithm**	**Coverage**
GPModelS9_0	VFE	FPS	20%
GPModelS9_1	FITC	FPS	20%
GPModelS9_2	VFE	K-Means	20%
GPModelS9_3	FITC	K-Means	20%
GPModelS9_4	VFE	FPS	30%
GPModelS9_5	FITC	FPS	30%
GPModelS9_6	VFE	K-Means	30%
GPModelS9_7	FITC	K-Means	30%
GPModelS9_8	VFE	Halton_v1	20%
GPModelS9_9	FITC	Halton_v1	20%
GPModelS9_10	VFE	Halton_v1	30%
GPModelS9_11	FITC	Halton_v1	30%
GPModelS9_12	VFE	Halton_v2	20%
GPModelS9_13	FITC	Halton_v2	20%
GPModelS9_14	VFE	Halton_v2	30%
GPModelS9_15	FITC	Halton_v2	30%

4.6.2 Sparse GP Models

To evaluate the effectiveness of sparse GP approximation, 16 sparse GP models were also investigated in this study, as listed in Table 3. All sparse models are derived from *GPModel9*, which was chosen as the benchmark due to its favorable balance between complexity, accuracy, and representativity (see Section 5.2).

Two sparse approximation schemes were applied, *VFE* and *FITC*, to assess their influence on the predictive accuracy and computational efficiency of the models.

Also, the placement of inducing points was varied using three selection algorithms: *K-means*, *FPS*, and *Halton* (as defined in Section 4.3). For the Halton sequence, two variants were tested: one spanning a fixed input domain $[-30°, 30°]^2$ (*Halton_v1*), and one restricted to the range of the training data (*Halton_v2*).

The number of inducing points was set to 20% and 30% coverage of the training dataset. Empirically, fewer than 20% often caused convergence issues, while exceeding 30% reduced the computational benefits of sparsification.

Except for the replacement of the full GP likelihood by a sparse approximation as defined in Table 3, all sparse GP models employed the same kernel structure, prior settings, and inference setups as their full GP counterpart.

4.7 Related Work

Early AEDBs were typically handcrafted by experts who fitted simple models to wind-tunnel data. For example, the Space Shuttle's aerodynamic data book [33] was based entirely on extensive wind-tunnel campaigns, where experts assigned a lower-bound *tolerance* (data scatter) and an upper-bound *variation* (systematic error) to the measured coefficients. Such classical aerodynamic models relied on low-degree interpolation of selected data points combined with expert judgment, which made objective and reproducible uncertainty quantification rather difficult. In contrast, modern AEDBs are typically assembled from multiple numerical and experimental data sources to better assess the influence of systematic errors. While systematic methods to quantify uncertainties in AEDBs have emerged in the recent years [4, 34, 35], they are not yet widely adopted by current vehicle development activities [36–38].

GP regression, also called *kriging*, has recently become a popular surrogate modeling tool in aerodynamics due to its ability to provide both accurate predictions and built-in uncertainty quantification. For example, GPs have been used to model the aerodynamics of missiles [39] and aircraft [40], and have been integrated into general aerodynamic modelling toolboxes [41, 42]. However, the applicability of GPs to large aerodynamic datasets is limited by their cubic computational cost. To address this, sparse GP approximations such as FITC and VFE have been successfully tested on wind-tunnel datasets [43]. However, these studies did not follow a full Bayesian approach, as the GP kernel parameters were fixed a-priori and have not been updated by Bayesian Inference.

In the Bayesian paradigm, GPs are one class among several probabilistic models. A recent study on the RLV demonstrator CALLISTO has compared multiple Bayesian AEDB models, particularly Fourier models, spline models and GP models [5]. While no clear evidence could be found to generally favor one Bayesian model class over the others, it could be shown that several Bayesian approaches yielded better accuracy in predicting uncertain aerodynamic coefficients than the classical expert-fitted model. Also for blunt-body entry vehicles, Bayesian inference has recently been applied within a nonlinear 6-DoF simulation framework, using a linearized aerodynamic model to estimate the stability derivatives of a capsule under uncertainty [35]. In a different field, deep Bayesian GPs were compared to both standard GPs and Bayesian Neural Networks for medical prediction tasks [44]. Their study showed that deep GP models capture prediction uncertainty more effectively than the investigated alternatives. Taken together, these results suggest that carefully chosen Bayesian models can significantly improve uncertainty estimates in complex systems, such as aerospace vehicles.

5. Results & Discussion

The models introduced in Section 4.6 were implemented with the probabilistic programming library `PyMC` 5.18 [28] and trained using the NUTS sampler from `NumPyro` [45, 46]. Inference on each model was conducted with four independent chains and 1,000 post-warmup draws per chain, while convergence diagnostics and posterior analyses were carried out with `ArviZ` [47]. All computations were performed on local workstations (8-16 logical cores, 32-128 GiB RAM) under Ubuntu 22.04 with Python 3.12, ensuring a consistent and reproducible environment.

The following sections present and discuss the results of this simulation campaign. We first analyze the fitting behavior of the full and sparse GP models, including convergence diagnostics and posterior characteristics (Section 5.1). We then evaluate their predictive performance across different training and validation setups (Section 5.2). Finally, we compare sparse approximations against their full GP counterpart to assess fidelity and computational efficiency (Section 5.3).

5.1 Training Convergence

Figure 8 illustrates the typical Bayesian inference workflow for *GPModel9*, by comparing its prior predictive and posterior predictive distributions. It can be observed that the prior is deliberately vague, reflecting only our weak assumptions made about data magnitude and smoothness. After training on the AEDB subset, the posterior adapts closely to the data: uncertainty levels shrink substantially, the 89% Highest Density Interval (HDI) covers all obser-

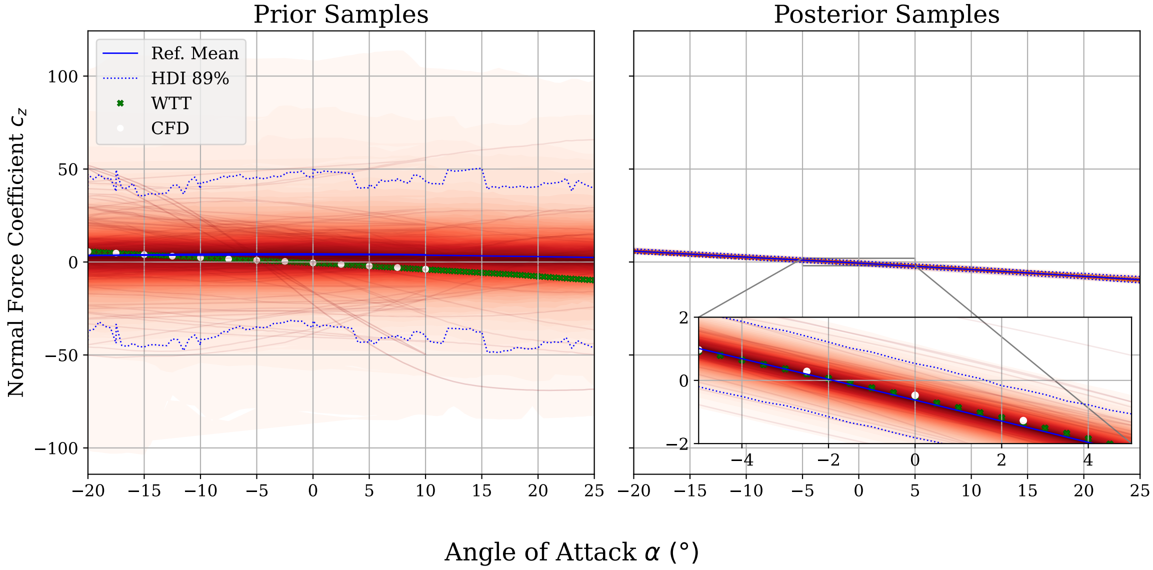

Fig. 8: Comparison of prior predictive distribution (left) and posterior predictive distribution (right) for *GPModel9*.

Table 4: Convergence diagnostics for the GP models listed in Table 2 in the nominal training setup.

Model	**Divergences**	**Max $\hat{R}$**	**Min ESS_{Bulk}**	**Min ESS_{Tail}**
GPModel5	0	1.00	2085.93	2236.30
GPModel6	0	1.00	1298.17	1152.37
GPModel7	0	1.00	3323.57	1742.15
GPModel8	0	1.00	2101.15	2321.38
GPModel9	0	1.00	1736.07	1301.80
GPModel10	0	1.00	2766.54	1866.47
GPModel11	0	1.00	1744.59	1704.82
GPModel12	0	1.00	1513.62	1057.12
GPModel13	0	1.00	2545.02	1664.01
GPModel14	0	1.00	1621.23	1460.59
GPModel15	0	1.00	1429.00	661.98
GPModel16	0	1.00	2156.35	2198.99

vations, and the underlying trend emerges directly from the dataset. This behavior exemplifies the intended advantage of Bayesian inference, to perform automated model fitting with principled uncertainty quantification, without manual parameter tuning by human experts.

Convergence of the MCMC algorithm was verified using the diagnostics introduced in Section 4.4. For the *nominal* training setup, all full GP models met the standard thresholds for $\hat{R}$ and ESS, as summarized in Table 4. Under the more demanding *K-Fold* setup, convergence issues were encountered for *GPModel6*, which showed $\hat{R} \approx 1.03$ and $ESS_{Bulk} \approx 200$. These indicators suggested poor chain mixing and insufficient exploration of the posterior by the NUTS sampler, leading to the exclusion of this model from further analyses as proper representation of the true posterior could not be ensured. Slight convergence issues were also noted for *GPModel5* and *GPModel10*, but these could be mitigated by adjusting the *target acceptance rate* of the NUTS sampler, allowing both models to be retained. The remaining GP models converged reliably also for the *K-Fold* setup as shown in Table A1.

For the sparse GP models, convergence diagnostics are reported in Table 5 for the *nominal* setup and Table A2 for the K-Fold setup. In both cases, most models satisfied the convergence criteria, with exception of *GPModelS9_7* (nominal) and *GPModelS9_5* (K-Fold). Attempts to improve their convergence were unsuccessful, and they were therefore also excluded from further evaluation.

Overall, the diagnostics confirm that the applied MCMC algorithm produced reliable posterior approximations for the majority of investigated models. This provides a robust basis for the subsequent analyses of model performance and predictive behavior.

5.2 Evaluation of Prediction

The predictive performance of the investigated Bayesian models was assessed using the RMSE as primary metric. RMSE provides a balanced measure of accuracy, accounting for both overall error magnitude and sensitivity to outliers, and is thus considered as suitable for the ReFEx dataset where no dedicated metric of the devel-

Table 5: Convergence diagnostics for the sparse GP models listed in Table 3 in the nominal training setup. Non-converged models are excluded from the table and from further analysis.

Model	**Divergences**	**Max** $\hat{R}$	**Min** ESS_{Bulk}	**Min** ESS_{Tail}
GPModelS9_0	0	1.00	2283.45	2209.22
GPModelS9_1	0	1.00	2284.90	2283.97
GPModelS9_2	0	1.00	2451.31	2389.57
GPModelS9_3	0	1.00	2195.29	1975.04
GPModelS9_4	0	1.00	2823.90	2646.76
GPModelS9_5	0	1.00	2258.58	2058.78
GPModelS9_6	0	1.00	2852.83	2470.23
GPModelS9_8	0	1.00	1295.03	1599.26
GPModelS9_9	0	1.00	1444.64	1535.89
GPModelS9_10	0	1.00	1375.51	1737.62
GPModelS9_11	0	1.00	1207.02	715.12
GPModelS9_12	0	1.00	1719.68	1838.20
GPModelS9_13	0	1.00	1478.46	1366.82
GPModelS9_14	0	1.00	1934.11	2158.33
GPModelS9_15	0	1.00	1446.09	1039.45

opment risk is prescribed. As outlined in Section 4.5, the *Nominal(FLIGHT)* setup is considered as most representative for evaluating the prediction quality of each model, provided that sufficient generalization capability of the model can be confirmed via the *KFold(FLIGHT)* setup. RMSE values for the CFD and WTT categories were also computed to check for inference inconsistencies within the training domains, but they are not further analyzed here.

Figure 9 summarizes the RMSE results for the GP models listed in Table 2. Models including a WN kernel (*GPModel7*, *GPModel10*, *GPModel13*) consistently show higher RMSE, indicating limited suitability of this kernel choice. In contrast, *GPModel5* yields the lowest RMSE in the FLIGHT category across both *nominal* and *K-Fold* setups, followed by *GPModel8*, *GPModel9*, and the RQ models (*GPModel14* to *GPModel16*). Despite their increased number of model parameters, the RQ models do not show clear benefit over the other models. However, it should be noted that the error bars of most models are overlapping, suggesting that this ranking should not be overinterpreted. Comparing the out-of-sample predictions for the *K-Fold* setup with the *nominal* setup, it can be observed that the RMSE consistently but not significantly increases for each model, indicating their sufficient generalization capabilities beyond their training data.

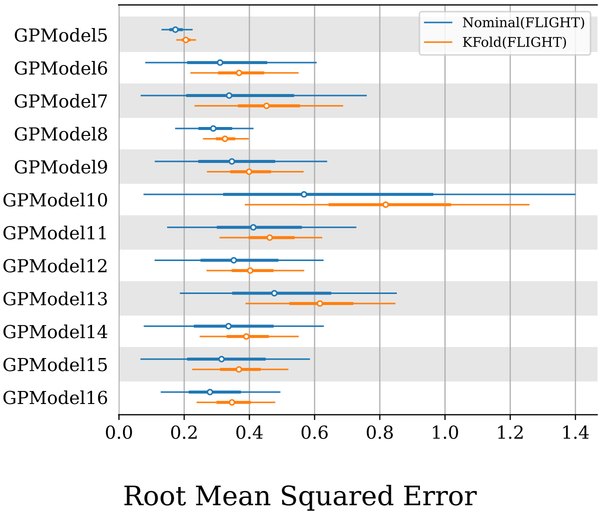

Fig. 9: RMSE results for the GP models listed in Table 2 across different training-prediction setups. For each model, the central marker indicates the median estimate, the thick bars show the interquartile range, and the thin lines represent the 89% HDI.

To illustrate the model behaviors, Figure 10 compares the prediction results of *GPModel5*, *GPModel8*, and *GPModel9*. While *GPModel5* achieves the lowest overall RMSE, its predictions exhibit a noticeable bias towards training points. This can be interpreted as underfitting of the data, leading to overly confident predictions that fail to capture the observed data trend. Also *GPModel8* shows a similar tendency, with the mean deviating from the training data despite an adequate data coverage by the HDI. In contrast, *GPModel9* aligns closely with the data, with both the mean and uncertainty bands capturing the observed trends more faithfully.

Overall, it can be concluded that *GPModel9* is one of the most representative models for the investigated AEDB dataset. It achieves competitive RMSE scores, while avoiding the underfitting behavior observed in *GPModel5* and *GPModel8*. For this reason, *GPModel9* was chosen as baseline for subsequent investigation of sparse GP approximation methods.

5.3 Similarity of Sparse Approximation

As discussed in Section 4.3, the computational complexity of full GP models makes sparse approximations an attractive alternative to reduce computational effort. This benefit, however, must be balanced carefully against the approximation errors they may introduce. To assess this trade-off, several sparse variants of *GPModel9*, as listed in Table 3, were evaluated against the full model.

During the inference runs in this study, the sparse approximations reduced the runtime from $60 - 150\,\mathrm{min}$ for the full GP model to about $20{-}40\,\mathrm{min}$, demonstrating clear computational savings in practical setups as well. Due to

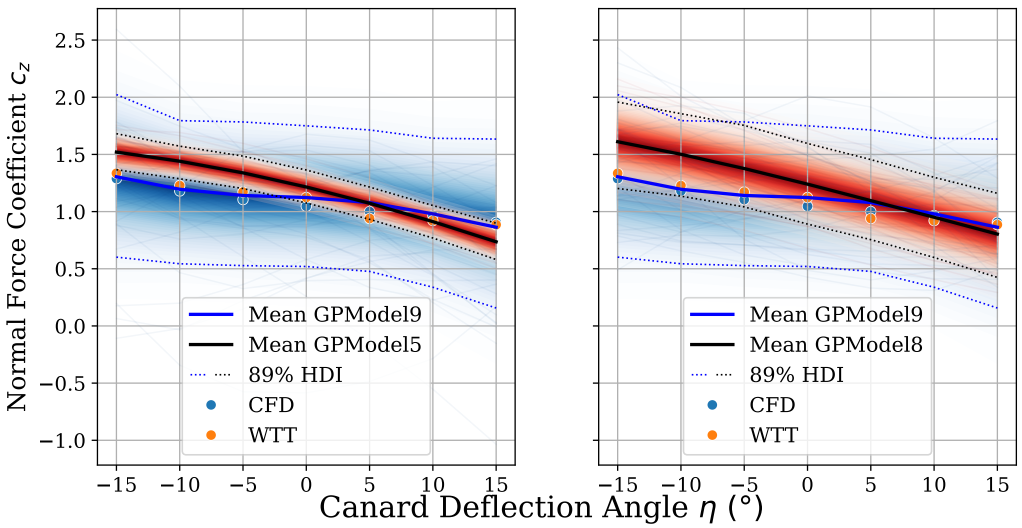

Fig. 10: Regression plots for *GPModel5* (left) and *GPModel8* (right) in comparison with *GPModel9*. Nominal training setup with predictions for src = FLIGHT and $\alpha = -5°$.

limited resources, a systematic benchmarking campaign could however not be conducted.

Figure 11 summarizes the RMSE results for sparse models under *nominal* and *K-Fold* training setups. Generally, the sparse models achieve similar RMSE levels compared to *GPModel9*, although their HDI intervals are systematically wider, indicating higher uncertainty in their predictions. Interestingly, in some cases the RMSE of the sparse models falls below that of the full model, particularly observable for Halton-sequence models (*GPModelS9_8* to *GPModelS9_15*). However, the largely overlapping error bars suggest these differences are indicative rather than statistically significant. Importantly, RMSE alone cannot capture the fidelity of the posterior distribution, which is the main criterion for evaluating the quality of the different sparse approximations.

To quantify the similarity more directly, posterior predictions of the sparse models were compared to *GPModel9* using Wasserstein and Energy distances, as shown in Figure 12. These metrics were computed over a structured grid in α and η, and evaluated for all 16 pairwise permutations of the MCMC chains. Among all candidates, *GPModelS9_2* achieved the lowest values across all similarity metrics and categories, marking it as the closest approximation of *GPModel9*.

A more detailed visual comparison is presented in Figure 13, showing regression plots for *GPModel9* (full model), *GPModelS9_4*, *GPModelS9_2*, and *GPModelS9_11* at selected values of α. The results show that *GPModelS9_2* and *GPModelS9_11* replicate the predictive behavior of the full model closely, while *GPModelS9_4* yields slightly deviating trends in data-sparse regions. Notably, *GPModelS9_2* and *GPModelS9_11* employ 20% and 30% of the training samples as inducing points, respectively, suggesting that accuracy depends more on the placement than the sheer number of inducing points.

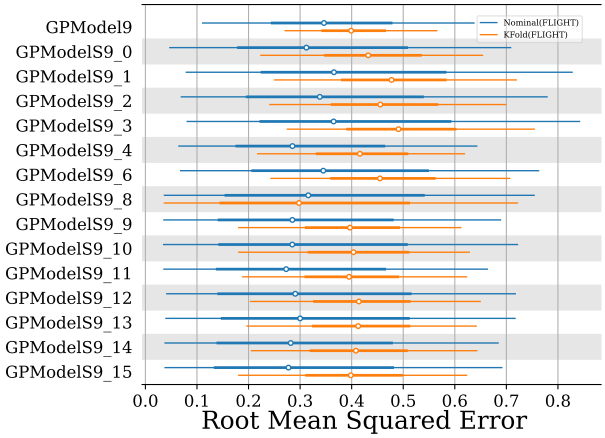

Fig. 11: RMSE results for the sparse GP models listed in Table 2 across different training-prediction setups. For each model, the central marker indicates the median estimate, the thick bars show the interquartile range, and the thin lines represent the 89% HDI.

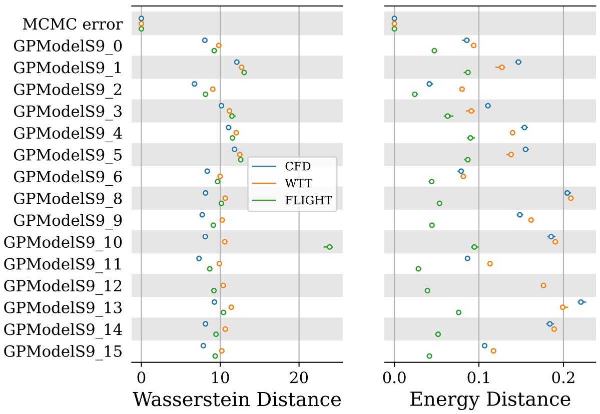

Fig. 12: Posterior predictive similarity metrics computed from all pairwise permutations of the four chains of *GPModel9* (full GP). For each sparse model, the central marker indicates the median estimate, the thick bars show the interquartile range, and the thin lines represent the 89% HDI.

Overall, these findings highlight the importance of inducing-point selection and approximation scheme. *GPModelS9_2*, using the VFE scheme with K-means clustering and 20% coverage of inducing points, emerged as the most reliable sparse model for the investigated setup. This aligns with the broader literature, where VFE is regarded as a state-of-the-art scheme for sparse GPs [48]. At the same time, no universally optimal selection strategy for inducing points can be deduced. For example, in VI-based inference, inducing points can be optimized jointly with kernel parameters [23]. In MCMC settings this is, however, impractical due to the greatly increased computational cost, which would jeopardize the benefits of sparse approximation. Therefore, fixed inducing points obtained via clustering methods or heuristic strategies remain subject for practical compromises.

In conclusion, while all investigated sparse GP models provided good agreement with the full GP model, *GPModelS9_2* stands out as the best-performing approximation for the ReFEx dataset. Nevertheless, optimal configurations may vary across datasets and applications, underscoring the need for case-specific evaluation.

6. Conclusions

The objective of this study was to evaluate whether Bayesian models can effectively capture and quantify aerodynamic uncertainties in the AEDB of ReFEx. Building on earlier work conducted for CALLISTO, a VTVL demonstrator, the present study extended the methodology to ReFEx, a VTHL vehicle, thereby testing the general applicability of Bayesian modeling across different RLV configurations.

The presented results confirm that Bayesian GP models can reliably reproduce the aerodynamic behavior of ReFEx within the investigated dataset and provide robust uncertainty quantification. Among the tested configurations, *GPModel5* (with SE kernels) achieved the lowest RMSE across both training and cross-validation predictions. However, *GPModel9* (with Matérn-5/2 ($M_{5/2}$) and Matérn-1/2 ($M_{1/2}$) kernels) was identified as the most representative model due to reduced bias with respect to the observed data, albeit at the cost of larger predictive variance. Sparse GP approximations proved capable of reproducing the posterior distribution of full GPs with high fidelity, while significantly reducing computational cost. RMSE analyses demonstrated that predictive accuracy remained close to that of the corresponding full GPs, and similarity metrics such as Wasserstein and Energy distances confirmed that the posterior distributions of sparse GPs closely matched their full GP counterpart. Overall, the Bayesian framework provided a systematic and reproducible treatment of aerodynamic uncertainties, offering clear advantages over classical expert-driven approaches.

The study is also subject to several limitations. The presented analysis was restricted to a reduced AEDB subset corresponding to the roll-maneuver regime of ReFEx and focused on the prediction of the normal force coefficient. Only a limited number of model configurations were investigated, and some had to be excluded due to convergence issues. Furthermore, out-of-sample evaluation was constrained by the available CFD and WTT coverage, meaning that extrapolation beyond the observed domain or to the flight configuration could not be validated (yet).

Despite these constraints, the findings demonstrate the feasibility of integrating Bayesian models into the AEDB generation workflow of RLVs. In particular, sparse GP enable scaling to larger datasets, which is essential for future operational databases. By reducing reliance on manual expert assessments, this approach offers a more traceable, reproducible, and automatable process for uncertainty quantification in aerodynamic modeling.

Future work will extend the models to cover the full AEDB of ReFEx, incorporating additional inputs such as Mach number, sideslip angle, and rudder deflections, as well as all aerodynamic force and moment coefficients, and their dynamic derivatives. A crucial next step will be the validation of the Bayesian AEDB against ReFEx flight data, which will provide direct insights into the rep-

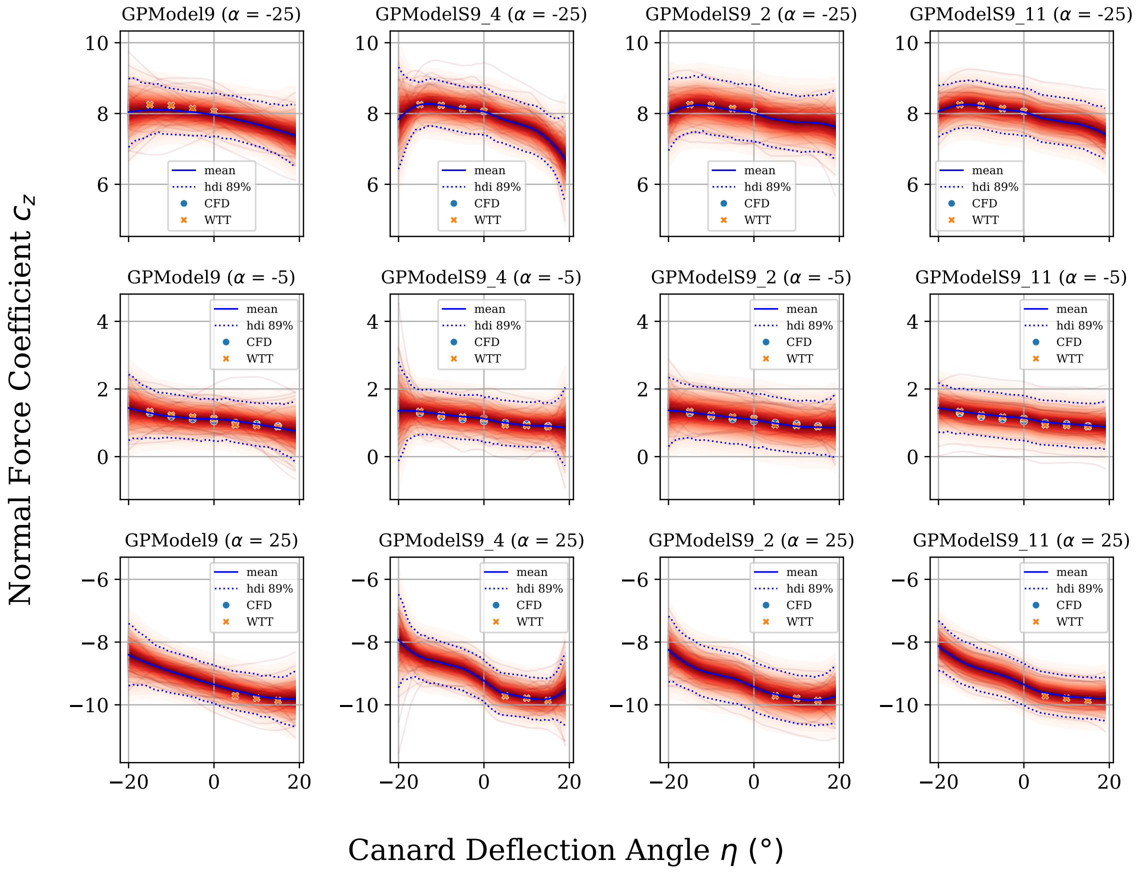

Fig. 13: Regression plots for *GPModel9*, *GPModelS9_4*, *GPModelS9_2*, and *GPModelS9_11* at selected values of α. Predictions are shown for c_z as a function of η.

resentativeness of the modeled uncertainties. Beyond GPs, also other Bayesian modeling approaches such as Bayesian Neural Networks (BNNs) or Bayesian Additive Regression Trees (BARTs) could be explored for comparison. More broadly, the successful application of Bayesian inference to aerodynamic modeling in ReFEx also underlines its potential for wider adoption across aerospace engineering and related domains.

Acknowledgements

The authors gratefully acknowledges the CFD development team at DLR, as well as the scientific support and HPC resources provided by DLR. The HPC system CARA is partially funded by the *Saxon State Ministry of Economic Affairs, Labour and Transport* and the *Federal Ministry of Economics and Climate Protection*. The HPC system CARO is partially funded by *Ministry of Science and Culture of Lower Saxony* and *Federal Ministry for Economic Affairs and Climate Action*. Furthermore, the authors would like to acknowledge the assistance of GPT-4o and GPT-5 in enhancing the grammar and wording used in this paper.

Appendix A. Convergence Diagnostics

Table A1: Convergence diagnostics for the GP models listed in Table 2 in the K-Fold training setup. Non-converged models are excluded from the table and from further analysis. Models in italic reached convergence thanks to a higher target acceptance rate of the NUTS sampler (95%).

Model	**Divergences**	**Max** $\hat{R}$	**Min** ESS_{Bulk}	**Min** ESS_{Tail}
GPModel5	0	1.00	1897.77	1618.76
GPModel7	0	1.00	2717.56	1491.86
GPModel8	0	1.00	1839.26	2004.22
GPModel9	0	1.00	1269.50	1125.24
GPModel10	0	1.00	2436.20	1497.17
GPModel11	0	1.00	1287.19	1003.34
GPModel12	0	1.00	1347.85	1364.35
GPModel13	0	1.00	2664.71	1470.15
GPModel14	0	1.01	1411.74	1327.59
GPModel15	0	1.00	1492.20	847.52
GPModel16	0	1.00	1763.10	2048.70

Table A2: Convergence diagnostics for the sparse GP models listed in Table 3 in the K-Fold training setup. Non-converged models are excluded from the table and from further analysis.

Model	**Divergences**	**Max** $\hat{R}$	**Min** ESS_{Bulk}	**Min** ESS_{Tail}
GPModelS9_0	0	1.00	2472.88	2330.53
GPModelS9_1	0	1.00	2274.21	2161.00
GPModelS9_2	0	1.00	2001.57	1964.38
GPModelS9_3	0	1.01	2290.44	1715.16
GPModelS9_4	0	1.00	2353.63	2312.25
GPModelS9_6	0	1.00	2555.32	1992.34
GPModelS9_8	0	1.00	1233.10	1418.26
GPModelS9_9	0	1.01	1283.95	999.52
GPModelS9_10	0	1.01	1297.61	1518.39
GPModelS9_11	0	1.00	1229.42	1230.32
GPModelS9_12	0	1.00	1603.51	1986.46
GPModelS9_13	0	1.00	1461.22	1286.90
GPModelS9_14	0	1.00	1846.12	1829.16
GPModelS9_15	0	1.00	1388.49	1379.02

References

[1] Sven Krummen, Lale Evrim Briese, Etienne Dumont, Tobias Ecker, Moritz Ertl, Sofia Giagkozoglou, Ansgar Heidecker, Josef Klevanski, Adrian Thomas Esteban Krieger, Felix Krziwanie, Hauke Martens, Lukas Opp, Margalida Puigserver Rosello, Jens Rathke, Jose Luis Redondo Gutierrez, Bodo Reimann, Anton Schneider, and Jens Windelberg. Callisto - on the design and development of a reusable first stage demonstrator. In *Deutscher Luft- und Raumfahrtkongress (DLRK)*, September 2024.

[2] Waldemar Bauer, Peter Rickmers, Alexander Kallenbach, Sven Stappert, Viola Wartemann, Clemens Hans-Joachim Merrem, René Schwarz, Marco Sagliano, Jan Thimo Grundmann, Andreas K. Flock, Thomas Thiele, Daniel Kiehn, Andreas Bierig, Jens Windelberg, Eugen Ksenik, Thorben E. Bruns, Tobias Ruhe, and Henning Elsaesser. Dlr reusability flight experiment refex. *Acta Astronautica*, 168:57–68, November 2019.

[3] Peter Rickmers, Etienne Dumont, Sven Krummen, Jose Luis Redondo Gutierrez, Leonid Bussler, Sebastian Kottmeier, Guido Wübbels, Hauke Martens, Svenja Woicke, Marco Sagliano, Janis Sebastian Häseker, Lars Witte, Martin Sippel, Waldemar Bauer, and Hendrik-Joachim Peetz. The callisto and refex flight experiments at dlr - challenges and opportunities of a wholistic approach. *Acta Astronautica*, 225:417–433, Dezember 2024.

[4] Sven Krummen, Pavan Tummala, Jascha Wilken, Etienne Dumont, Moritz Ertl, Tobias Ecker, Johannes Riehmer, and Josef Klevanski. Applying bayesian inference to estimate uncertainties in the aerodynamic database of callisto. In *2023 IEEE Aerospace Conference*, pages 1–18, 2023.

[5] Sven Krummen, Jan Michael Schraad, Tobias Ecker, Moritz Ertl, Bodo Reimann, Josef Klevanski, Johannes Riehmer, Silas Eichel, Marco Sagliano, Lale Evrim Briese, and Etienne Dumont. Bayesian models for uncertainty estimation in aerodynamic databases of reusable launch vehicles. In AIAA, editor, *AIAA SciTech 2024 Forum*, pages 1–28, Januar 2024.

[6] Peter Rickmers, Sebastian Kottmeier, Guido Wübbels, and Waldemar Bauer. Refex: Reusability flight experiment - a demonstration experiment for technologies for aerodynamically controlled rlv stages. In *Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS*, Juli 2023.

[7] Leonid Bussler, Jose Luis Redondo Gutierrez, Peter Rickmers, and Sven Stappert. Refex: Reusability flight experiment – trajectory design. In *Aerospace Europe Conference 2023 – 10th EUCASS – 9th CEAS*, Juli 2023.

[8] Waldemar Bauer, Eduard Schnoor, Andreas Rittweger, and Peter Rickmers. Numerical crash simulation of the reusability flight experiment refex. In *Pro-*

ceedings of the International Astronautical Congress, IAC, Oktober 2019.

[9] Peter Rickmers, Waldemar Bauer, Guido Wübbels, and Sebastian Kottmeier. Refex: Reusability flight experiment – a project overview. In *8TH European Conference for Aeronautics adn Space Science*, Juli 2019.

[10] Clemens Hans-Joachim Merrem, Daniel Kiehn, Viola Wartemann, and Thino Eggers. Aerodynamic design of a reusable booster stage flight experiment. In *EUCASS 2019*, Juli 2019.

[11] Merrem, Clemens and Wartemann, Viola and Eggers, Thino. Preliminary aerodynamic design of a reusable booster flight experiment. *CEAS Space Journal*, 12(3):429–439, April 2020.

[12] Merrem, Clemens and Wartemann, Viola and Eggers, Thino. *Aerodynamic Data Set Generation for the Experimental Vehicle ReFEx*. Springer International Publishing, 2021.

[13] Dieter Schwamborn, Thomas Gerhold, and Ralf Heinrich. The dlr tau-code: Recent applications in research and industry. In *ECCOMAS CFD 2006 CONFERENCE*, 2006. auf CD.

[14] Langer, Stefan and Schwöppe, Axel and Kroll, Norbert. The DLR Flow Solver TAU - Status and Recent Algorithmic Developments. In *52nd Aerospace Sciences Meeting*, National Harbor, Maryland, January 2014. American Institute of Aeronautics and Astronautics.

[15] Mack, Andreas and Hannemann, Volker. Validation of the Unstructured DLR-TAU-Code for Hypersonic Flows. In *32nd AIAA Fluid Dynamics Conference and Exhibit*. AIAA Paper 2002-3111, 2002.

[16] Schwamborn, Dieter and Gerhold, Thomas and Heinrich, Ralf. The DLR TAU-Code: Recent Applications in Research and Industry. European Conference on Computational Fluid Dynamics. In *ECCOMAS CFD*, 2006.

[17] Allmaras, S. and Johnson, F. and Spalart, P. Modifications and clarifications for theimplementation of the spalart-allmaras turbulence model. In *ICCFD7-1902. In: 7thInternational Conference on Computational Fluid Dynamics*, 2012.

[18] Wartemann, Viola and Konosidou, Ntarinai and Flock, Andreas and Merrem, Clemens. *Contribution of Numerical and Experimental Flow Simulations to the Aerodynamic Data Base of the DLR Reusable Flight Experiment ReFEx*, pages 141–150. Springer International Publishing, 2021.

[19] Marius Franze, Viola Wartemann, Clemens Merrem, Henning Elsäßer, Tobias Ruhe, Thino Eggers, and Hendrik Weihs. Aerothermal loads analysis of ReFEx by coupled fluid–structure simulations. *CEAS Space Journal*, 17:513–527, May 2025.

[20] Andreas K. Flock, Thomas Thiele, Dominik Neeb, and Ali Gülhan. Planned wind tunnel experiments at dlr köln for the reusability flight experiment (refex). In *International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions & Engineering*, 2019.

[21] Stan Development Team. Prior choice recommendations. https://github.com/stan-dev/stan/wiki/prior-choice-recommendations, 2024.

[22] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005.

[23] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[24] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations, 2017.

[25] Xin Jin and Jiawei Han. *K-Means Clustering*, pages 563–564. Springer US, Boston, MA, 2010.

[26] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.

[27] Art B. Owen. A randomized halton algorithm in r, 2017.

[28] Oriol Abril-Pla, Vincent Andreani, Colin Carroll, Lin Dong, Christopher J. Fonnesbeck, Mikhail Kochurov, Ravin Kumar, Junpeng Lao, Christian C. Luhmann, Osvaldo A. Martin, Max Osthege, Ricardo Vieira, Thomas Wiecki, and Radim Zinkov. PyMC: a modern, and comprehensive probabilistic programming framework in python. *PeerJ Computer Science*, 9:e1516, Sep 2023.

[29] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved r^ for assessing convergence of mcmc (with discussion). *Bayesian Analysis*, 16(2), June 2021.

[30] Gabriel Peyré and Marco Cuturi. Computational optimal transport, 2020.

[31] Gabor Szekely and Maria Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5, 11 2004.

[32] Qing Wang, Sanjeev R. Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009.

[33] Space Systems Group. Aerodynamic design data book orbiter vehicle sts-1. Technical report, Rockwell International, 1980.

[34] James Scoggins, Thomas Wignall, Tenavi Nakamura-Zimmerer, and Karen Bibb. Multihierarchy gaussian process models for probabilistic aerodynamic databases using uncertain nominal and off-nominal configuration data. 01 2023.

[35] Ashraf Kassem, Shafi Al Salman Romeo, Bipin Tiwari, Omer San, and Kursat Kara. Bayesian framework for estimating dynamic stability derivatives in 6-dof blunt-body entry vehicles. *Aerospace Science and Technology*, 167:110646, 2025.

[36] Josef Klevanski, Bodo Reimann, Sven Krummen, Moritz Ertl, Tobias Ecker, Johannes Riehmer, and Etienne Dumont. Progress in aerodynamic studies for callisto - reusable vtvl launcher first stage demonstrator. In *9th European Conference for Aeronautics and Space Sciences*, Lille, France, June 2022.

[37] Antonio Schettino, Giuseppe Pezzella, Marco Marini, Sara Di Benedetto, Victor F Villace, Johan Steelant, Rishabh Choudhury, Anatoly Gubanov, and Nina Voevodenko. Aerodynamic database of the hexafly-int hypersonic glider. *CEAS Space Journal*, 12(2):295–311, 2022.

[38] Pietro Roncioni, Pier Luigi Vitagliano, Fabrizio De Gregorio, Fabio Paglia, and Claudio Milana. Aerodatabase of vega-c launcher development and integration. In *8th European Conference for Aeronautics and Space Sciences*, Madrid, Spain, July 2019.

[39] Seungmin Lee, Shinseong Kang, and Kyunghoon Lee. Development of a gaussian process modeling application for the prediction of missile aerodynamic coefficients. *Journal of Mechanical Science and Technology*, 35(3):987–997, 2021.

[40] {Zachary A.} Miles, Sebastian Lopez, and {Michael P.} Kinzel. Gaussian process regression implemented as surrogate model to aid aerodynamic design process. In *AIAA Aviation Forum and ASCEND, 2024*, AIAA Aviation Forum and ASCEND, 2024. American Institute of Aeronautics and Astronautics Inc, AIAA, 2024. Publisher Copyright: © 2024, American Institute of Aeronautics and Astronautics Inc, AIAA. All rights reserved.; AIAA Aviation Forum and ASCEND, 2024 ; Conference date: 29-07-2024 Through 02-08-2024.

[41] Philipp Bekemeyer, Anna Bertram, Derrick Armando Hines Chaves, Mateus Dias Ribeiro, Andrea Garbo, Anna Kiener, Christian Sabater Campomanes, Mario Stradtner, Markus Widhalm, Stefan Görtz, Florian Jäckel, Robert Hoppe, and Nils Hoffmann. Data-driven aerodynamic modeling using the dlr smarty toolbox. In AIAA, editor, *AIAA Aviation 2022 Forum*, pages 1–19. American Institute of Aeronautics and Astronautics, Inc., Juni 2022. View Video Presentation: https://doi.org/10.2514/6.2022-3899.vid.

[42] Jigar Parekh and Philipp Bekemeyer. A surrogate-based approach for a comprehensive uq analysis in cfd. In *AIAA SciTech 2024 Forum*, Januar 2024.

[43] Hugo Valayer, Nathalie Bartoli, Mauricio Castaño-Aguirre, Rémi Lafage, Thierry Lefebvre, Andrés F. López-Lopera, and Sylvain Mouton. A python toolbox for data-driven aerodynamic modeling using sparse gaussian processes. *Aerospace*, 11(4), 2024.

[44] Yikuan Li, Shishir Rao, Abdelaali Hassaine, Rema Ramakrishnan, Yajie Zhu, Dexter Canoy, Gholamreza Salimi-Khorshidi, Thomas Lukasiewicz, and Kazem Rahimi. Deep bayesian gaussian processes for uncertainty estimation in electronic health records, 2020.

[45] Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

[46] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019.

[47] Ravin Kumar, Colin Carroll, Ari Hartikainen, and Osvaldo Martin. Arviz a unified library for exploratory

analysis of bayesian models in python. *Journal of Open Source Software*, 4(33):1143, 2019.

[48] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.