HiSST: 4th International Conference on High-Speed Vehicle Science & Technology
22–26 September 2025, Tours, France

Active learning-based approach for the trajectory simulation of an hypersonic vehicle involving high-fidelity models

Loïc Brevault[1], Mathieu Balesdent[2]

Abstract

Simulating the trajectory of hypersonic vehicles using high-fidelity aeropropulsive performance models is computationally intensive due to the need for thousands of evaluations during numerical ordinary differential equations integration. To address this, an active learning surrogate-based trajectory simulation strategy is proposed that enables accurate hypersonic vehicle trajectory computation while significantly reducing computational cost. The method builds Gaussian process surrogates using a tailored design of experiments and employs an adaptive enrichment process guided by trajectory simulation to selectively add informative samples. By leveraging prediction uncertainty from the surrogates, the approach identifies and evaluates new points with the high-fidelity aeropropulsive performance model only where needed, ensuring accurate trajectory integration with a reduced number of high-fidelity calls. The proposed methodology is demonstrated on a representative hypersonic vehicle ascent trajectory. Results show that the approach achieves trajectory accuracy comparable of the high-fidelity reference while reducing computational costs by more than an order of magnitude compared to direct integration and offline surrogate-based strategies. This work highlights the potential of inline, active learning surrogate modeling for enabling high-fidelity trajectory simulation of hypersonic vehicles in computationally constrained contexts.

Keywords: *Trajectory simulation, Active learning, Gaussian process, Hypersonic vehicle*

1. Introduction

The simulation of the trajectory of a hypersonic vehicle involves solving a system of ordinary differential equations (ODEs) representing the equations of motion. Estimating the evolution of the state variables (e.g., altitude, velocity, flight path angle) over time requires numerically computing an integral using an ODE integration algorithm (e.g., Runge-Kutta). However, a significant challenge arises when the system of differential equations involves a high-fidelity model, such as Nose-to-Tail (NtT) CFD RANS (Computational Fluid Dynamics, Reynolds-Averaged Navier-Stokes - CFD RANS) calculations, to estimate aero-propulsive forces and moments. In such cases, numerical integration may require thousands of sequential evaluations of the high-fidelity model, which is unaffordable in practice. To enable the trajectory simulation of hypersonic vehicles using high-fidelity models, a surrogate-based approach is proposed. This methodology relies on an active learning strategy guided by the trajectory simulation.

In the literature only few papers focused on surrogate-based trajectory simulation combined with active learning strategy. Most of the papers rely on Bayesian trajectory optimization and substitutes the objective and constraints function using a surrogate model. Needels *et al.* [8] proposed a multi-fidelity Gaussian process surrogate with a sensitivity-based strategy considering an hypersonic glider. The sensitivities of trajectory quantities of interest to aerodynamic parameters modeled by surrogates based on optimal control techniques is developed and relies on necessary adjoint equations for the dynamics of a hypersonic glide vehicle. Most of the existing approaches relies on and offline surrogate-based strategy in which the surrogate model is built before the final trajectory simulation of interest for instance using neural networks [2, 9], hypernetworks [10], or Gaussian process [6] and multi-fidelity surrogates [7]. In addition, such a surrogate-based approach has also been used for post-treatment and interpolation

[1] *DTIS, ONERA, Université Paris-Saclay, 91120 Palaiseau, France, loic.brevault@onera.fr*
[2] *DTIS, ONERA, Université Paris-Saclay, 91120 Palaiseau, France, mathieu.balesdent@onera.fr*

between integration points based on an exact high-fidelity trajectory dataset [5].

In this work, an active learning surrogate-based strategy dedicated to the trajectory simulation which involves a computationally intensive aeropropulsive balance model is proposed. This approach allows to build a dedicated design of experiments for surrogate modeling with an adapted selection of Design of Experiments (DoE) samples in adequacy with the trajectory simulation of interest. The proposed approach relies on the following steps. The first step involves conducting a DoE tailored to the context of trajectory simulation, allowing the evaluation of the high-fidelity model at a limited number of data points. The DoE is performed with respect to input variables (e.g., altitude, Mach number, angle of attack) to compute various quantities of interest—such as aerodynamic coefficients and thrust—after evaluating the high-fidelity model. Based on this limited dataset, Gaussian processes (GPs) are used to map the input variables to the output quantities of interest involved in the system of ODEs.

Using the GPs inside the right-hand side of the equations of motion instead of the high-fidelity simulator, it becomes possible to solve the ODEs and approximate the vehicle trajectory. However, due to the limited size dataset, the GPs only provide an approximation of the exact high-fidelity model, meaning that the initial trajectory is not fully representative of one obtained through direct high-fidelity simulation. In the second step, to improve the accuracy of the trajectory, an active learning strategy is developed to adaptively update the DoE with new data points. Leveraging the uncertainty model of the GPs, this strategy solves an optimization problem defined by an infill criterion to select new data points in specific regions of the input space where the prediction uncertainty is large on the current trajectory. As a result, the high-fidelity model is evaluated only at a limited number of points corresponding to regions of interest for the trajectory.

The proposed approach is applied to a representative test case of hypersonic vehicle trajectory simulation with a scramjet. A comparison is conducted between the proposed approach and a trajectory simulation using the high-fidelity model combined with an offline surrogate-based strategy, focusing on metrics such as accuracy of the trajectory and computational cost. The results demonstrate that the inline surrogate-based approach provides an accurate trajectory while significantly reducing the computational expense.

The rest of the paper is organized as follows. In Section 2, the general principle for trajectory simulation is presented with details on numerical integration for ODEs and the flight dynamics of interest for hypersonic vehicles. Then, in Section 3, the proposed approach is presented, with a brief introduction of Gaussian process and a detailed presentation of the different steps of the enrichment process. Then in Section 4, a representative test case of hypersonic vehicle trajectory simulation is carried out with an analysis of the performance of the proposed approach compared to alternative strategies.

2. Trajectory simulation

2.1. Flight dynamics

The estimation of hypersonic vehicles performance typically involves the simulation and the optimization of a trajectory. The trajectory simulation implies the integration of a system of Ordinary Differential Equations (ODE) according to the time. Simplified three-degrees-of-freedom equations of motion can be written as:

$$\begin{aligned}
\dot{r} &= v\ \sin(\gamma) \\
\dot{v} &= \frac{T\ \cos(\alpha) - D}{m} - g\ \sin(\gamma) \\
\dot{\gamma} &= \frac{L + T\ \sin(\alpha)}{mV} - \frac{g\ \cos(\gamma)}{v} + \frac{v\ \cos(\gamma)}{r} \\
\dot{m} &= -q
\end{aligned}$$

with r the radius, v the relative velocity, γ the flight path angle, T the thrust, D the drag, L the lift, α the angle of attack, m the mass and q the mass flow rate. For hypersonic air-breathing vehicles, thrust, drag and lift forces depend on the ambient atmospheric conditions. During the simulation of a trajectory, a control law which depends on time is considered fixed. This latter is optimized in the context of trajectory optimization in order to determine the optimal control law to obtain the optimal hypersonic vehicle performance. Classical control laws involve the definition of the temporal evolution of angles such as pitch, angle of attack, control surface deflection angle as well as the mixture ratio in order to reach the final trajectory target conditions in terms of state variables (altitude, velocity, *etc.*) and also to ensure the stability of the vehicle.

2.2. Numerical integration

It is possible to generalize the flight dynamics of the previous section under the form of a dynamical system governed by the following generic system of ODE:

$$\dot{\mathbf{x}}(t) = \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \tag{1}$$

$$\mathbf{x}(t_i) = \mathbf{x}_i \tag{2}$$

with $\mathbf{x}(\cdot)$ the vector of dimension d_x of the state variables (e.g., altitude, velocity, flight path angle) characterizing the evolution of the vehicle along the trajectory, $\mathbf{u}(\cdot)$ the vector of dimension d_u of the trajectory guidance (e.g., control law, mass flow rate), $\mathbf{f}(\cdot)$ the vectorial function computing the flight dynamics equations with an evaluation of the aeropropulsive performance and $\mathbf{x}_i$ the vector of the initial conditions of the state variables at the initial time t_i.

In a trajectory simulation process, considering an initial state $\mathbf{x}_i$ at t_i and a fixed guidance law $\mathbf{u}(t)$ (which depends of time), in order to get the state vector at the final time t_f of interest, it is necessary to determine :

$$\mathbf{x}(t_f) = \mathbf{x}_i + \int_{t_i}^{t_f} \dot{\mathbf{x}}(t)\mathrm{d}t = \int_{t_i}^{t_f} \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \mathrm{d}t \tag{3}$$

As $\mathbf{f}(\cdot)$ is not an analytic vector function, numerical techniques are required to compute this integral to get access to the state variable vector at the final time (and all the intermediate time of interest). Numerical integration consists in discretizing with respect to time the integral (using fixed or variable integration steps) to estimate the value of the state variable vector at intermediate step times $(t_{i+1}, t_{i+2}, \ldots, t_f)$. At each time step, depending on the numerical scheme used, one or several evaluations of the equations of the flight dynamics $\mathbf{f}(\cdot)$ are carried out requiring evaluations of the physical models for the aeropropulsive performance (Figure 1).

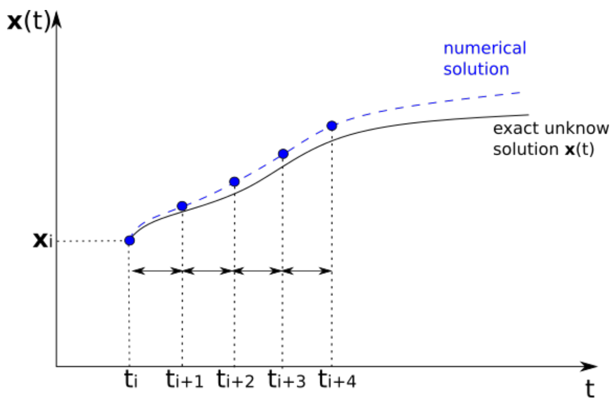

Fig 1. Temporal discretization of the trajectory

The objective of numerical integration for ODE is to obtain the desired accuracy with a computational budget (corresponding to the number of aeropropulsive model evaluations) as low as possible.

Different numerical integration techniques for ODE exist and may be organized according to different categories (Figure 2) [3].

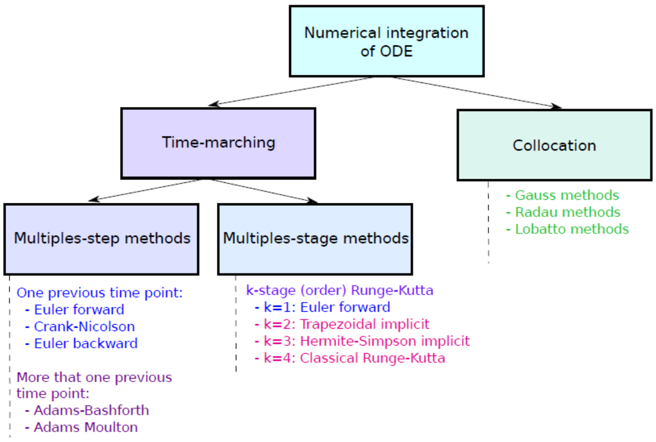

Fig 2. Classical numerical integration techniques for ODE

Among the time-marching approaches, the multi-step techniques consider one point (e.g., Euler forward, Euler backward) or several points (e.g., Adams-Bashforth, Adams-Moulton) per integration steps whereas the multi-stage techniques subdivide each interval of integration with several points of evaluation (e.g., Trapezoidal implicit, Hermite-Simpson implicit, Runge-Kutta). The collocation family of approaches are based on a reformulation of the integration problem by the introduction of an approximation of the state variable vector under the form of a polynomial where the polynomial coefficients are optimized in order to satisfied defection constraints at the collocation nodes.

Among the different integration techniques for ODE, Runge-Kutta approach [3] is classically used as it is simpler to implement compared to collocation methods and an adaptive time step can be defined based on an estimation of the error of integration at a reasonable computational cost. To control the integration error and adapt the step size, a Runge-Kutta 4-5 method (also known as the Runge-Kutta-Fehlberg method [4]) is often employed. This approach relies on an embedded Runge-Kutta scheme that computes two different approximations of the integral at each step: one using a fourth-order method and the other using a fifth-order method that includes all the nodes of the former. By comparing these two estimates, the integration error can be assessed with only one additional model evaluation. This error estimate is then used to adjust the step size dynamically, ensuring error control throughout the integration process.

Simulating the trajectory of a hypersonic vehicle using a classical integration scheme like Runge-Kutta 4-5 (which requires five model evaluations per time step) typically involves several thousand sequential evaluations of the aeropropulsive model. However, when the aeropropulsive model includes CFD RANS methods through NtT calculations, such a numerical integration approach becomes computationally impractical.

Therefore, in order to achieve simulation of the trajectory of a hypersonic vehicle based on computationally intensive aeropropulsive model, a surrogate-based strategy is proposed in the following.

3. Proposed approach: surrogate-based trajectory simulation

Due to computational time constraints, performing high-fidelity online aeropropulsive model evaluations during trajectory simulation is not possible (for example, in the context of Nose-to-Tail computations). One way to enable trajectory simulation at a manageable computational cost is to replace the CFD-based aeropropulsive model with a mathematical surrogate model (or metamodel), whose computational cost is negligible compared to the ”exact” model. A key challenge then lies in controlling the impact of introducing an approximation of the aeropropulsive model on the simulated trajectory. To address this, an adaptive metamodel construction strategy is implemented.

The idea is to carry out a DoE campaign to build an initial database from which a surrogate model is constructed. Various approaches for generating this database are discussed later. In this study, a simulation model that provides both the aeropropulsive performance and the center of gravity position is used. The inputs to the computational code are altitude, Mach number, angle of attack, control surface deflection angle, thrust, mass flow rate, and the position of the center of gravity. The outputs of the computational model include aerodynamic forces and moments referenced to the center of gravity.

Two main strategies can be identified for introducing surrogate models:

- Offline: the surrogate models are built prior to the analysis, in this case, before the trajectory simulation.
- Online: the surrogate models are updated (i.e., enriched) during the trajectory simulation itself.

A major challenge lies in estimating the level of confidence in the results, given the approximations introduced by using surrogate models. Online approaches (also called adaptive learning or goal oriented) allow the surrogate models to be refined in regions of interest relevant to the study, thereby improving both the predictive quality and the associated confidence level. These refinement strategies typically require surrogate models that provide an estimate of uncertainty along with each prediction (e.g., Gaussian Processes, as described later).

In the following, an online approach is presented within the context of the aeropropulsive performance model used for trajectory simulation.

More generally, the process of using a surrogate model in a study consists of the following steps (Figure 3):

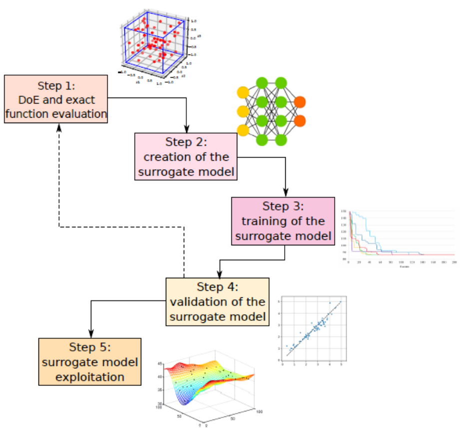

Fig 3. General process for surrogate model

1. Designing an initial DoE on which the computational code is evaluated,
2. Creating the surrogate model,
3. Training the surrogate model (optimizing the metamodel's hyperparameters to align its predictions with the DoE data),
4. Validating the metamodel,
5. Using the metamodel in the target application.

In the case of an online strategy, a loop is carried out between steps 4 and 1 until the desired accuracy for the trajectory is reached or the total available simulation budget is meet.

3.1. Active learning process

3.1.1. General process

To tailor the design of experiments (i.e., the database) to the specific needs of the trajectory simulation using a computationally expensive model, an adaptive enrichment strategy is implemented. This approach involves the following steps (Figure 4):

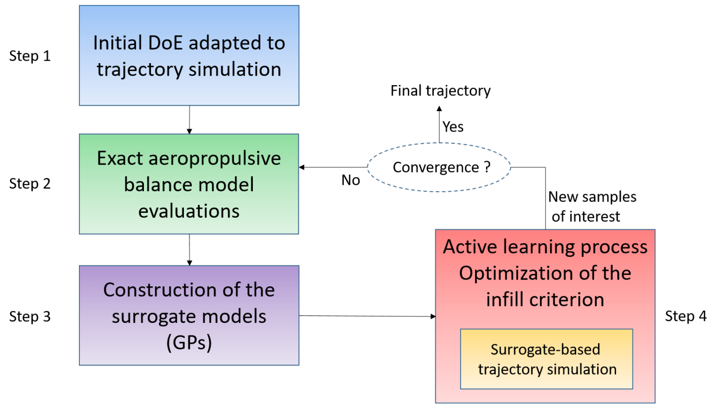

Fig 4. General active learning process for surrogate-based trajectory simulation

- Step 1: Generate a limited-size design of experiments using a tailored Latin Hypercube Sampling (LHS) strategy, designed to be relevant in the context of trajectory simulation;
- Step 2: Evaluate the high-fidelity aeropropulsive performance model to obtain an initial training dataset;
- Step 3: Construct a set of surrogate models in the form of Gaussian Processes, with one surrogate model for each output of interest from the high-fidelity model;
- Step 4: Apply an adaptive enrichment strategy to the database, adding new, relevant data points that improve the accuracy and confidence of trajectory predictions using the surrogate models.

The exact computationally intensive aeropropulsive performance model is noted $\mathbf{g} : \mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^6 \to \mathbf{y} = \mathbf{g}(\mathbf{z}) \in \mathbb{R}^4$. The input variable vector corresponds to the altitude, the Mach number, the angle of attack, the control surface deflection angle, the fuel-to-air ratio and the mass of propellant. The output variable vector corresponds to the drag and lift coefficients, the pitching moment coefficient and the air mass flow rate through the engine. The objective is to construct a surrogate model for $\mathbf{g}(\cdot)$ and to enrich it through active learning. Each step of the proposed process is detailed in the following sections.

3.1.2. Step 1: initial DoE tailored for trajectory simulation

In the first step, it is necessary to define an initial DoE. Several approaches can be considered: grid-based methods, Monte Carlo methods, and low-discrepancy methods (which aim to ensure a well-distributed sampling of points across the input domain), among others.

In the following the illustrations (Figure 5) corresponds to a cut of the overall input space of dimension 6 considering the two input variables “altitude” and “Mach number”. First, it is important to note that grid-based strategies (left of Figure 5) are often poorly suited for DoE construction. This is primarily because they result in significant information loss due to the uniform spacing (iso-distance) of sampling points along each dimension. Furthermore, the number of required points increases exponentially with the dimensionality of the input space, making this approach practically infeasible for dimensions greater than three.

Monte Carlo methods aim to randomly distribute points throughout the input space. These approaches are dimension-independent and ensure broad coverage of the space. However, they can result in

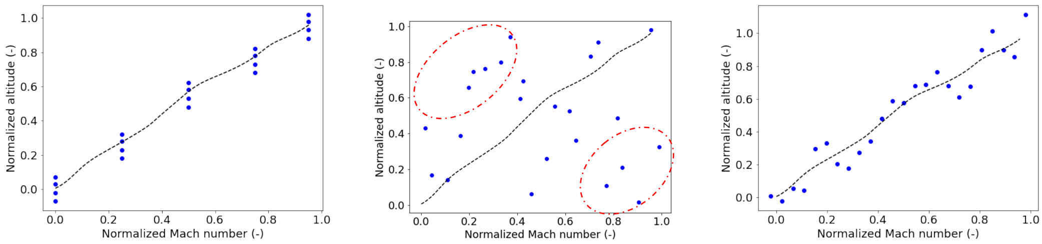

Fig 5. Different strategies for the initial DoE, left: grid-based approach, center: LHS over the entire space, right: dedicated LHS based on a prior flight envelop

clusters of nearby points, leading to redundant information and unnecessary computational cost. Low-discrepancy sampling methods (e.g., Sobol' sequences, Halton sequences), which typically incorporate a degree of randomness, are generally preferred for improving the distribution of training points across the surrogate model's input domain. One of the most widely used techniques is Latin Hypercube Sampling (LHS), which is the method employed in this study (middle of Figure 5). However, another important consideration when developing a surrogate model and its associated DoE is to tailor them to the specific objectives of the study. For instance, in trajectory simulation, analyzing only the two variables "altitude" and "Mach number" reveals that a full sampling across their entire definition domain would lead to a large number of unnecessary computations (middle of Figure 5, red areas), since the relevant trajectory is typically confined to a specific sub-region of the flight envelope.

Therefore, an adapted LHS sampling strategy to the context of hypersonic vehicle trajectory simulation is proposed in order to avoid the generation of points in regions of limited interest regarding the input space and therefore losses of computational resources.

To this end, a classical LHS is performed within the unit cube $[0., 1.]^6$, denoted $\mathbb{Z}_n^{\text{norm}}$, where n is the number of points. Then, for each variable of the vector $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^6$, bounds of variation (representing a potential flight corridor) are defined to scatter the normalized data around a "prior flight envelop, yielding the appropriate design of experiments $\mathbb{Z}_n = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$. This reference trajectory may come from a previous study, be obtained using an alternative guidance method, or be defined through a broad potential flight corridor. It is not necessary for this flight corridor (and the bounds of variation) to encompass the (unknown) trajectory intended for simulation. This "adapted LHS" approach makes it possible to rationalize the positioning of points within the definition domain $\mathcal{Z}$ relative to a hypersonic vehicle trajectory, thereby avoiding the generation of points in regions of $\mathcal{Z}$ outside the reachable trajectory envelop (e.g., very high altitude and very low Mach number simultaneously).

For the variables related to the angle of attack and control surface deflection angle, since these values are often defined by a trim process at each point along the trajectory (numerically solving for a zero), the entire definition domain is covered by the initial DoE.

This initial DoE is intended solely to distribute points within regions of interest relative to a hypersonic vehicle trajectory and to avoid wasting computational time. The adaptive enrichment strategy will subsequently allow this initial DoE to be refined through a goal-oriented approach.

3.2. Step 2: evaluation of the exact computationally intensive aeropropulsive performance model

This step aims to obtain the outputs corresponding to the design of experiments $\mathbb{Z}_n = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ (noted $\mathbf{Z}_n = [\mathbf{z}_1, \ldots, \mathbf{z}_n]^T$ under the matrix form) by evaluating the exact aeropropulsive performance model, yielding $\mathbb{Y}_n = \{\mathbf{y}_1 = \mathbf{g}(\mathbf{z}_1), \ldots, \mathbf{y}_n = \mathbf{g}(\mathbf{z}_n)\}$ (similarly $\mathbf{Y}_n = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^T$). This step is computationally expensive since the high-fidelity model is evaluated n times. However, these evaluations can be performed in parallel. Additionally, the size of the initial design of experiments is limited.

3.3. Step 3: surrogate models creation based on Gaussian process

In this step, based on the design of experiments $\{\mathbb{Z}_n, \mathbb{Y}_n\}$, a surrogate model is constructed to approximate the exact function $\mathbf{g} : \mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^6 \to \mathbf{y} = \mathbf{g}(\mathbf{z}) \in \mathbb{R}^4$. The surrogate model, denoted $G(\cdot)$, is based on Gaussian processes (GP) [11]. A Gaussian process $G(\cdot) \sim \mathcal{GP}(\mu(\cdot), \text{cov}(\cdot,\cdot))$ is fully defined by a prior through its mean function $\mu(\cdot)$ and its covariance function $\text{cov}(\cdot,\cdot)$.

In the context of this study, one particular aspect of the function of interest $\mathbf{g}(\cdot)$ is that it is a vector-valued output function. Several strategies exist to approximate this type of function depending on its intended use:

- constructing independent Gaussian processes, one for each component of the output vector $\mathbf{y}$,
- constructing a single Gaussian process that predicts all components of the output vector simultaneously.

The enrichment strategy presented later does not require sampling realizations from the Gaussian process, therefore there is no risk of consistency loss between the coordinate of the output vector if independent GPs are built. For simplicity reasons and without loss of generality, independent Gaussian processes for each component of the output vector are constructed : $G_{j=1,\dots,4}(\cdot) \sim \mathcal{GP}(\mu_j(\cdot), \text{cov}_j(\cdot,\cdot))$.

In practice, the covariance model is represented using a parametric multivariate kernel $k_{\boldsymbol{\theta}}(\cdot,\cdot)$ with parameter vector $\boldsymbol{\theta}$. The kernel is a symmetric positive semi-definite function. The most known kernels [11] are the Squared Exponential kernel (also known as Radial Basis Function), the Rational Quadratic kernel, the Matérn kernel, *etc.* The covariance function is a key element in GP. The covariance function allows to encode some assumptions on the behavior of the exact function (*e.g.*, smoothness, periodicity, stationarity, separability). Multidimensional kernels may be obtained by combining single dimensional kernels through for instance a product operator following the formalism of Reproducing - Kernel - Hilbert - Space (RKHS) [1]. In addition, without any prior knowledge, a constant mean function μ is assumed as a prior.

The training phase of the Gaussian process aims to determine the values of the parameters associated with the mean function (generally assumed to be constant) and the parameter vector $\boldsymbol{\theta}$ of the covariance model. For that matter, from these input and output training sets and the prior on the GP, it is possible to train the surrogate model using the marginal likelihood. It is obtained by integrating out the latent function giving $p\left(\mathbf{Y}_n|\mathbf{Z}_n, \boldsymbol{\theta}, \mu, \sigma\right) = \mathcal{N}\left(\mathbf{Y}_n|\mu, \mathbf{K}_{nn} + \ \sigma^2\mathbf{I}_{nn}\right)$ with $\mathbf{I}_{nn}$ the identity matrix of size n, $\mathbf{K}_{nn}$ a covariance matrix built from $k_{\boldsymbol{\theta}}(\cdot,\cdot)$ evaluated on the input DoE $\mathbf{Z}_n$ and σ^2 a Gaussian homoscedastic noise variance in the case of noisy numerical DoE (in can be adapted in case of heteroscedastic noise). For instance, this numerical noise may be used to model the numerical convergence error of NtT calculations. To simplify the notations $\hat{\mathbf{K}}_{nn} = \mathbf{K}_{nn} + \sigma^2\mathbf{I}_{nn}$ is introduced. In practice, the GP training requires to minimize the negative Log-Marginal Likelihood (LML) with respect to the hyperparameters $\boldsymbol{\theta}$, μ and σ. The LML $L(\cdot)$ is given by:

$$
\begin{aligned}
L\left(\boldsymbol{\theta}, \mu, \sigma|\mathbf{Z}_n, \mathbf{Y}_n\right) &= \log\left(p\left(\mathbf{Y}_n|\mathbf{Z}_n, \boldsymbol{\theta}, \mu, \sigma\right)\right) && (4)\\
&\propto \log\left(|\hat{\mathbf{K}}_{nn}|\right) - \mathbf{Y}_n^T\hat{\mathbf{K}}_{nn}^{-1}\mathbf{Y}_n && (5)
\end{aligned}
$$

where all the kernel matrices implicitly depend on the hyperparameters $\boldsymbol{\theta}$ (similarly for below). To solve the optimization problem, any optimizer may be used (*e.g.*, gradient-based, population-based algorithms). Moreover, a closed form of the constant mean function may be sometimes determined [11].

Once the training is complete and the optimal parameters have been identified, the Gaussian process is conditioned on the design of experiments $\{\mathbb{Z}_n, \mathbb{Y}_n\}$ to determine the Gaussian process posterior distribution used for prediction. The prediction at a new unknown location $\mathbf{z} \in \mathcal{Z}$ is done by using the conditional properties of a multivariate normal distribution:

$$
p\left(y|\mathbf{z}, \mathbf{Z}_n, \mathbf{Y}_n, \hat{\boldsymbol{\theta}}, \hat{\mu}, \hat{\sigma}\right) = \mathcal{N}\left(y|\hat{g}(\mathbf{z}), \hat{s}^2(\mathbf{z})\right) \tag{6}
$$

where $\hat{g}(\mathbf{z}), \hat{s}^2(\mathbf{z})$ are the mean prediction and the associated variance. These terms are defined by:

$$\hat{g}(\mathbf{z}) = \hat{\mu} + \mathbf{k}_{\mathbf{z}}^T \left(\mathbf{K}_{nn} + \hat{\sigma}^2 \mathbf{I}_{nn}\right)^{-1} (\mathbf{y}_n - \mathbf{1}\hat{\mu}) \tag{7}$$

$$\hat{s}^2(\mathbf{z}) = k_{\mathbf{z},\mathbf{z}} - \mathbf{k}_{\mathbf{z}}^T \left(\mathbf{K}_{nn} + \hat{\sigma}^2 \mathbf{I}_{nn}\right)^{-1} \mathbf{k}_{\mathbf{z}} \tag{8}$$

where $k_{\mathbf{z},\mathbf{z}} = k_{\hat{\boldsymbol{\theta}}}(\mathbf{z}, \mathbf{z})$ and $\mathbf{k}_{\mathbf{z}} = \left[k_{\hat{\boldsymbol{\theta}}}\left(\mathbf{z}_i, \mathbf{z}\right)\right]_{i=1,\dots,n}$

3.4. Step 4: active learning enrichment strategy

Once the Gaussian processes $G_{j=\{1,\dots,4\}}(\cdot)$ have been trained and conditioned to determine the posterior predictive models, it becomes possible to replace the exact model with the obtained Gaussian processes within the flight dynamics equations:

$$\dot{\mathbf{x}}(t) = \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] \quad \Rightarrow \quad \dot{\hat{\mathbf{x}}}(t) = \frac{\mathrm{d}\hat{\mathbf{x}}(t)}{\mathrm{d}t} = \hat{\mathbf{f}}_{\mathbf{Z}_n,\mathbf{Y}_n}[\hat{\mathbf{x}}(t), \mathbf{u}(t), t],$$

where $\hat{\mathbf{f}}_{\mathbf{Z}_n,\mathbf{Y}_n}[\cdot]$ denotes the system of differential equations in which the exact model has been replaced by the Gaussian processes $G_{j=\{1,\dots,4\}}$ constructed from the design of experiments $\{\mathbf{Z}_n, \mathbf{Y}_n\}$.

Trajectory simulation then allows the computation of the final state $\hat{\mathbf{x}}(t_f)$ at time t_f as:

$$\hat{\mathbf{x}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}(t_f) = \mathbf{x}_i + \int_{t_i}^{t_f} \dot{\hat{\mathbf{x}}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}(t)\, \mathrm{d}t = \mathbf{x}_i + \int_{t_i}^{t_f} \hat{\mathbf{f}}_{\mathbf{Z}_n,\mathbf{Y}_n}[\hat{\mathbf{x}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}(t), \mathbf{u}(t), t]\, \mathrm{d}t,$$

where $\hat{\mathbf{x}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}(\cdot)$ denotes the approximation of the state variable vector when using the surrogate models. The numerical integration of the trajectory is computationally inexpensive since the exact model is not used, but rather the surrogate models.

The objective of the adaptive enrichment strategy is to add new points to the database such that, in the input space regions of interest:

$$G_{j=\{1,\dots,4\}\{\mathbf{Z}_n,\mathbf{Y}_n\}}(\cdot) \to \mathbf{g}(\cdot),$$

meaning that the Gaussian processes converge toward the exact model, ensuring:

$$\hat{\mathbf{f}}_{\mathbf{Z}_n,\mathbf{Y}_n}(\cdot) \to \mathbf{f}(\cdot),$$

(convergence of the differential equations), and consequently:

$$\hat{\mathbf{x}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}(\cdot) \to \mathbf{x}(\cdot),$$

enabling the trajectory obtained using the surrogate models to converge toward the trajectory obtained using the exact aeropropulsive performance model.

To select the next points to add to the database, the following optimization problems are proposed:

For each output of the exact model $j = \{1, \dots, 4\}$, solve:

$$t_{\text{opt}}(j) = \underset{t_{k=\{1,\dots,N\}}}{\arg\max} \;\; \hat{s}^2 \left[G_j\left(\hat{\mathbf{x}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}(t_k), \mathbf{u}(t_k), t_k\right)\right]$$

subject to:

$$\dot{\hat{\mathbf{x}}}(t_k) = \hat{\mathbf{f}}_{\mathbf{Z}_n,\mathbf{Y}_n}[\hat{\mathbf{x}}(t_k), \mathbf{u}(t_k), t_k],$$

where $t_{k=\{1,\dots,N\}}$ are the N integration time steps used for numerically solving the system of differential equations with the surrogate models.

This optimization problem identifies, for each output component of the exact model, the time-point along the integrated trajectory where the Gaussian process predictive variance is maximal. This allows identifying where the surrogate models are the least "confident" along the trajectory, and these points are added to the database. This optimization step does not add significant computational cost since it only involves a maximum search over a finite list. From the optimal point $\hat{\mathbf{x}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}(t_{\text{opt}}(j))$ for $j = \{1,\ldots,4\}$, it is possible to determine the corresponding input parameter vector $\mathbf{z}_{\text{opt}} \in \mathcal{Z}$ for the exact model.

The points added to the database are thus relevant for trajectory simulation since they lie along the current integrated trajectory (i.e., within the region of interest of the definition domain $\mathcal{Z}$) and correspond to regions of low confidence in the surrogate model predictions.

At each enrichment iteration, it is possible to add between 1 and 4 points to the database (depending on whether the optima for the four output components correspond to the same time $t_{\text{opt}}(j)$ for $j = \{1,\ldots,4\}$ or to different times). Furthermore, if multiple enrichment points are identified, the exact model can be evaluated at these points in parallel. In the following, without loss of generality, the notation considers the addition of a single point to the database. Once the enrichment point $\mathbf{z}_{\text{opt}}$ is defined, it is added to the database:

$$\mathbb{Z}_{n+1} = \mathbb{Z}_n \cup \{\mathbf{z}_{\text{opt}}\},$$

and the exact model is evaluated to enrich the output database:

$$\mathbb{Y}_{n+1} = \mathbb{Y}_n \cup \{\mathbf{g}(\mathbf{z}_{\text{opt}})\}.$$

Based on the new DoE $\{\mathbb{Z}_{n+1}, \mathbb{Y}_{n+1}\}$, the Gaussian processes are retrained, and a new enrichment iteration is performed. Since the "exact" trajectory is not known, the stopping criterion for enrichment is based on the stagnation of the convergence of the trajectory between successive enrichment iterations (e.g., stagnation of the difference between the state variables $\hat{\mathbf{x}}_{\{\mathbf{Z}_n,\mathbf{Y}_n\}}\left(t_{k=\{1,\ldots,N\}}\right)$ and $\hat{\mathbf{x}}_{\{\mathbf{Z}_{n+q},\mathbf{Y}_{n+q}\}}\left(t_{k=\{1,\ldots,N\}}\right)$ with $q \in \{1,\ldots,5\}$). Here, the number of iterations with stagnation used to define convergence is set to 5.

4. Illustrative test case of hypersonic vehicle trajectory simulation

In order to illustrate the proposed approach, a representative test case is presented. It corresponds to the flight of an hypersonic vehicle for an ascent mission.

The aeropropulsive performance model aims to compute trimmed flight conditions for an hypersonic vehicle by ensuring: the longitudinal force balance between thrust and drag, the vertical force balance between lift and weight component, the moment balance (pitching moment equilibrium) and consistent engine operating point (matching mass flow rates, pressures, and combustion conditions). This model is required for trajectory simulation.

The aeropropulsive performance model is provided by a representative model with a computational cost lower than classical CFD RANS NtT calculations. However, the proposed method is dedicated to computationally intensive model.

The reference unknown trajectory simulated with the exact aeropropulsive performance model is represented in Figure 6.

This exact unknown trajectory is obtained using a classical Runge-Kutta 45 (RK45) numerical ODE integrator, requiring 1982 evaluations of the exact aeropropulsive performance model. It is important to note that the settings of the numerical ODE integrator (absolute and relative tolerances, integration order, etc.), or even the choice of the numerical ODE integrator itself (Runge-Kutta, Adams/BDF method, collocation with Radau technique, etc.), have a significant influence on both the number of exact model evaluations and the accuracy of the resulting trajectory. Therefore, a trade-off must be made between integration accuracy, computational cost, and the intrinsic uncertainty of the "exact" aeropropulsive model. Indeed, there is no benefit in performing a highly accurate numerical integration of a trajectory when the underlying aeropropulsive model exhibits a high level of predictive uncertainty. In the present

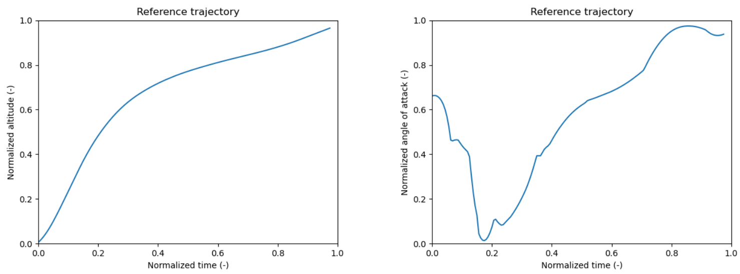

Fig 6. Reference unknown trajectory simulated with the exact aeropropulsive performance model

study, taking all these considerations into account, an appropriate trade-off has been established when configuring the ODE integrator.

4.1. Initial DoE

To build the initial DoE, $n = 20$ samples are generated using the proposed adapted LHS strategy by sampling within the hypercube $[0,1]^6$. Then, for each variable of the vector $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^6$, bounds of variation (representing a potential flight corridor) are defined to scatter the normalized data around a "prior knowledge trajectory," yielding the appropriate DoE $\mathbb{Z}_n = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$. Selected projections of the initial DoE into subspaces are shown in Figure 7.

Certain subspaces are sampled within specific regions (for instance, the altitude–Mach number subspace), taking into account the global regions of interest for this type of vehicle and ascent trajectory. In other subspaces of the input space (for example, angle of attack and Mach number), the entire domain of definition is covered since a trim process (numerically solving for a zero) is performed at each point along the trajectory.

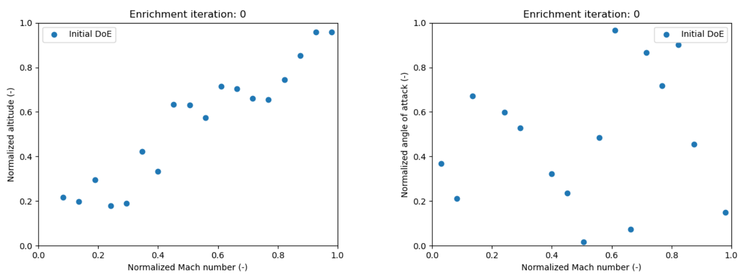

Fig 7. Initial DoE represented through cuts of the input space of dimension 6.

4.2. First iteration of the enrichment process

From the initial DoE, four Gaussian processes are constructed, and a first trajectory simulation is performed using these surrogate models. As illustrated in Figure 8, the resulting surrogate-based trajectory is far from the exact unknown trajectory obtained with the full aeropropulsive model. This discrepancy arises from the limited size of the initial DoE ($n = 20$) and its lack of adaptation to the simulation of the unknown trajectory.

To improve the accuracy of the surrogate-based trajectory, an enrichment strategy is implemented to identify new simulations of the exact aeropropulsive performance model that should be performed to refine the initial DoE. By solving the infill problem, new candidate points are determined. Figure 9 shows the initial DoE in the input space, the surrogate-based trajectory using this initial DoE in the altitude–Mach number and angle-of-attack–Mach number subspaces, and the two infill candidate points (green dots) selected for the next iteration.

By evaluating the exact aeropropulsive performance model at these two infill candidate points, the DoE

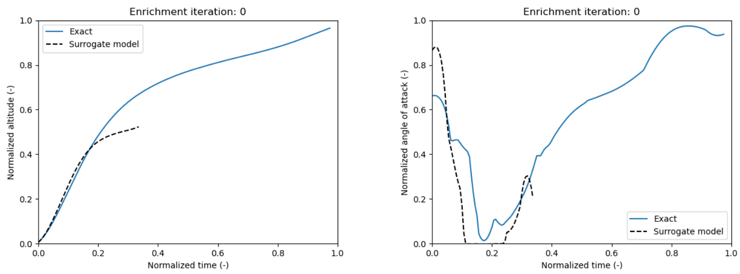

Fig 8. Reference unknown trajectory simulated with the exact aeropropulsive performance model and the trajectory simulated with Gaussian process constructed from the initial DoE. Evolution as a function of time of the normalized altitude (left), and normalized angle of attack (right).

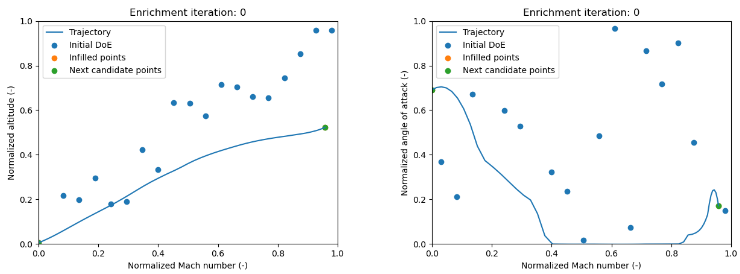

Fig 9. Initial DoE represented in subspaces altitude-Mach number and angle of attack-Mach number of the 6 dimension input space and corresponding surrogate-based trajectory.

can be enriched, allowing the Gaussian processes to be rebuilt. The updated DoE at iteration 1 (in the altitude–Mach number subspace) and the resulting trajectory (altitude as functions of time) are presented in Figure 10. This enrichment of the DoE allows the surrogate-based trajectory to approach the exact unknown trajectory more closely.

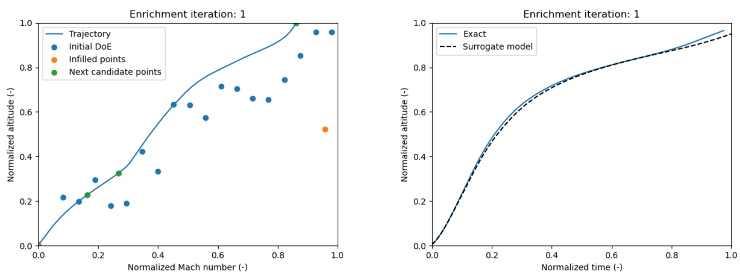

Fig 10. DoE at iteration 1 in subspace altitude-Mach number (left), surrogate-based trajectory with the altitude as a function of time (right).

4.3. Full enrichment process

The enrichment process is repeated over several iterations to rationally select new infill candidate points on which the exact aeropropulsive performance model is evaluated. These evaluations enrich the DoE and update the surrogate-based trajectory. The resulting surrogate-based trajectories at iterations 10, 30, and 79 (final iteration) are shown in Figure 11, illustrating the evolution of normalized altitude and angle of attack as functions of time. A significant improvement in the accuracy of the surrogate-based trajectory is observed, with the trajectory converging toward the exact unknown trajectory. This demonstrates the value of an in-line approach for surrogate-based simulation of the trajectory of a hypersonic vehicle when using a computationally intensive aeropropulsive performance model.

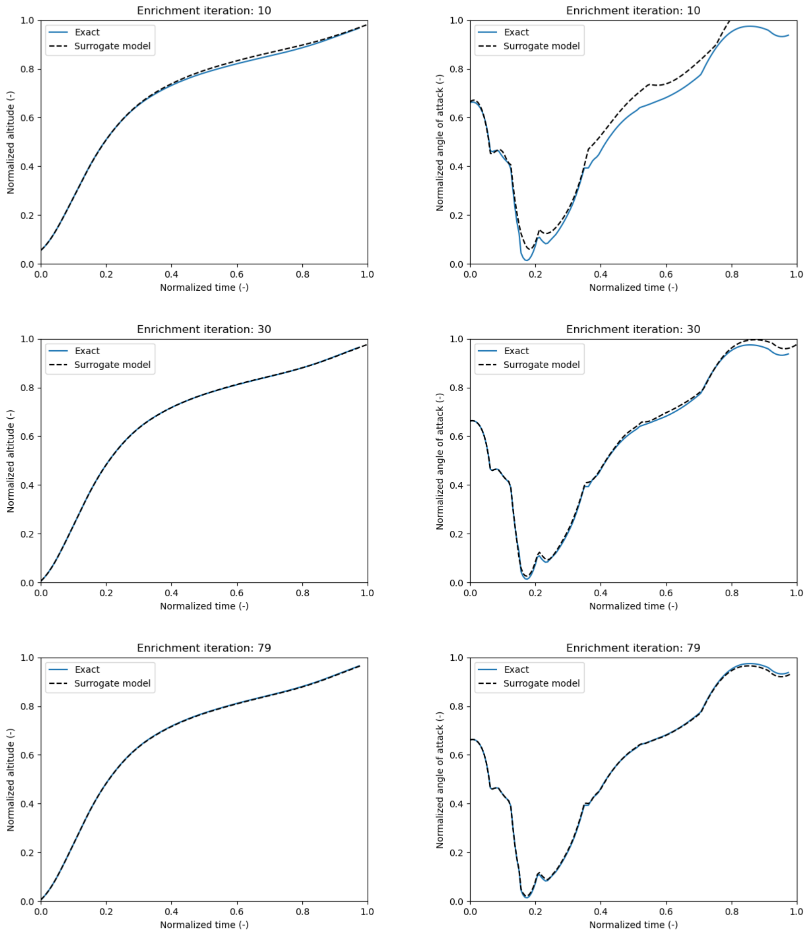

Fig 11. Reference unknown trajectory simulated with the exact aeropropulsive performance model and the surrogate-based trajectory simulated at iteration 10 (first row), 30 (middle row) and 79 (last row). Evolution as a function of time of the normalized altitude (first column) and normalized angle of attack (second column).

Additionally, Figure 12 displays the evolution of the DoE across the enrichment process for iterations 10 (first row), 30 (middle row), and 79 (last row). The samples added to the DoE correspond to specific regions of the input space, primarily distributed along the unknown final trajectory, which is challenging to target using an offline approach since the evolution of the trajectory within the six-dimensional input space is unknown in practice. The enrichment process thus enables the selection of informative samples on which the computationally intensive aeropropulsive performance model is evaluated.

In comparison, the adaptive enrichment strategy makes it possible to obtain a trajectory similar to the reference one using only 232 calls to the exact aeropropulsive performance model (including the 20 calls from the initial design of experiments), corresponding to an 8.5-fold reduction in the number of calls to the exact model.

Moreover, these 232 calls correspond to 80 iterations of the enrichment process (one initial plus 79 enrichment iterations), which amounts to an average of 2.9 calls to the exact aeroropulsive performance model per iteration. These 2.9 calls per iteration can be performed in parallel, and if a sufficiently large computing cluster is available, they can be considered equivalent to a single call in terms of computation time. Therefore, the 1982 sequential calls required by the “exact” approach should be compared with the 20 + 80 = 100 calls (with parallelization) in the refinement strategy, representing approximately a 20-fold reduction in computation time.

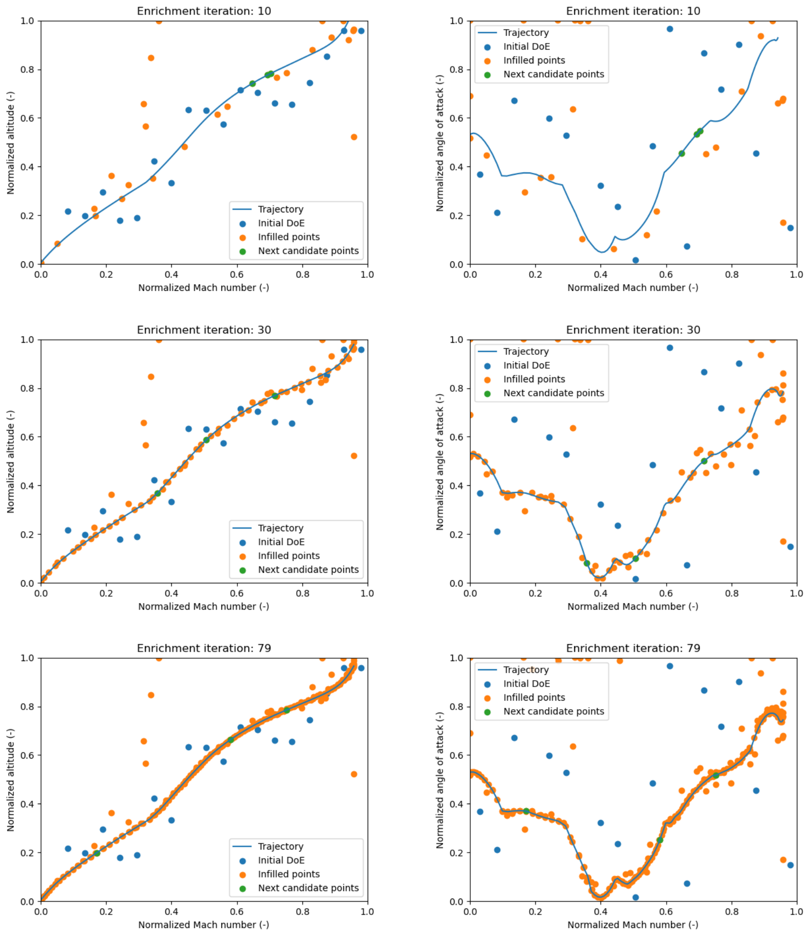

Fig 12. DoE represented in subspaces altitude-Mach number (left) and angle of attack-Mach number (right) at iteration 10 (first row), 30 (middle row) and 79 (last row)

4.4. Robustness to initial DoE

To evaluate the robustness of the proposed surrogate-based trajectory simulation process with respect to the initial DoE, 20 repetitions of the approach are performed using different initial DoEs. Furthermore, to illustrate the benefits of the proposed enrichment strategy, it is compared with two alternative approaches:

- an offline approach (referred to as "No infill"), in which the surrogate-based trajectory relies solely on an initial DoE (generated using the proposed adapted LHS strategy) with the same number of samples as the current iteration of the enrichment process, allowing for comparison at equal computational cost,
- an online approach (referred to as "Random infill") that uses the proposed infill criterion but, instead of selecting the samples that maximize the infill criterion, randomly selects samples among those satisfying the infill constraints.

Figure 13 shows the convergence of the mean error (in percent) of the normalized altitude and normalized Mach number as a function of the number of evaluations of the exact aeropropulsive performance model, compared to the exact unknown trajectory. This error corresponds to the mean error computed over the entire trajectory (with respect to time). The shaded areas represent the 25% and 75% interquartile ranges, while the line indicates the median across the repetitions.

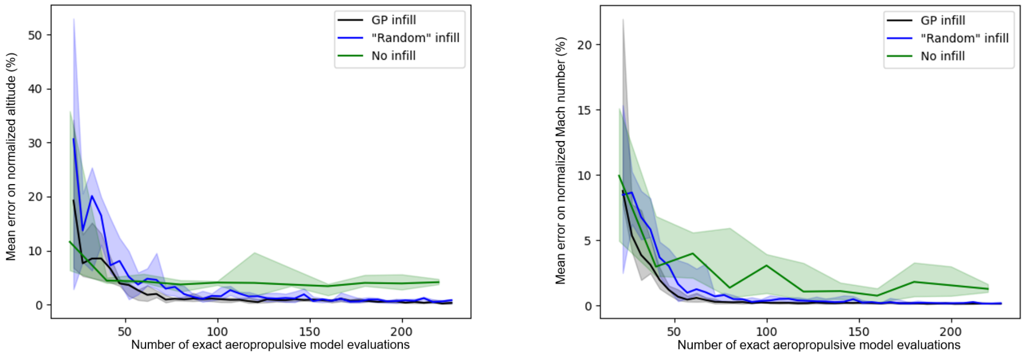

Fig 13. Mean error in percent on normalized altitude (left) and normalized Mach number (right) compared to the exact unknown exact trajectory for three surrogate-based trajectory strategies : an offline approach ("No infill"), a random online infill ("Random infill") and the proposed approach ("GP infill"). The shaded areas correspond to the 25% and 75% interquartiles. The line corresponds to the median of the repetitions.

First, it is interesting to note the benefit of adaptive enrichment strategies for the convergence of the surrogate-based trajectory toward the exact unknown trajectory. Indeed, the offline strategy does not converge toward the exact trajectory despite more than 200 calls to the exact aeropropulsive performance model. Although a decrease in the mean error is observed as the size of the DoE increases, the added points are not necessarily relevant for this specific trajectory. Furthermore, there is a large dispersion between repetitions.

In contrast, enrichment strategies allow convergence toward the exact trajectory and are robust to initialization (low dispersion around the median). It is observed that the adaptive enrichment strategy with point selection based on the lowest prediction confidence enables faster convergence toward the exact unknown trajectory. For example, after 75 calls to the exact aeropropulsive performance model, the median error with the "random infill" strategy is 5 times higher to the median error with the proposed approach. Indeed, the "random infill" strategy may potentially add points to the database in regions where there is already a high confidence in the metamodel predictions, thereby "wasting" computational cost by evaluating the exact performance model in areas of the definition domain $\mathcal{Z}$ where prediction confidence is high.

Moreover, with the proposed enrichment strategy, it is noted that with a median of only 75 calls (20 initial points + 55 enrichment points), representing a 26.4-fold reduction in the number of calls, the obtained accuracy is in the same order of magnitude of the trajectory simulation with the exact aeropropulsive performance model considering both RK45 settings and uncertainty modeling associated to the aeropropulsive model.

5. Conclusion

In this paper, a surrogate-based trajectory simulation strategy has been proposed for hypersonic vehicle applications, where the aeropropulsive performance model relies on computationally intensive calculations such as CFD RANS NtT evaluations. The approach leverages an active learning strategy guided by the uncertainty model of Gaussian Processes to substitute the exact aeropropulsive performance model during trajectory simulation. By iteratively identifying and selecting new data points in specific regions of the input space where the prediction uncertainty is high along the current trajectory, the high-fidelity model is evaluated only at a limited number of informative points relevant to the trajectory of interest.

The proposed methodology has been applied to a representative test case of hypersonic vehicle trajectory simulation. The results demonstrate that the surrogate-based approach can accurately reproduce the reference trajectory while significantly reducing the computational cost compared to direct high-fidelity simulations within the ODE integration process.

Future work will focus on incorporating engine operability constraints into the active learning process. Additionally, the extension of the proposed approach to optimal control problems will be investigated to enable the determination of guidance laws for hypersonic vehicles while maintaining control over computational costs.

6. Acknowledgments

The authors want to thanks Glen Sire for its help in the implementation of the trajectory simulation.

References

[1] Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

[2] Shanti Bhushan, Greg W Burgreen, Anup Zope, Parker Bailey, and Ian D Dettwiller. Hypersonic trajectory optimization for minimum heating using machine learned physics surrogates. In *AIAA AVIATION FORUM AND ASCEND 2024*, page 4236, 2024.

[3] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

[4] Erwin Fehlberg. New high-order runge-kutta formulas with step size control for systems of first- and second-order differential equations. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 44(S1 S1):T17–T29, 1964.

[5] Marius Franze and Fynn Barz. Comparison of models for aerothermal load prediction using coupled trajectory simulations of a high lift reentry vehicle. *CEAS Space Journal*, pages 1–17, 2025.

[6] Jacob R Johanik. Surrogate modeling of computational aerodynamic responses for a generic hypersonic vehicle. Master's thesis, Air Force Institute of Technology, 2022.

[7] Tyler E Korenyi-Both, Nathan J Falkiewicz, and Matthew C Jones. Open-source high-speed flight surrogate modeling framework. *arXiv preprint arXiv:2411.03598*, 2024.

[8] Jacob T Needels and Juan J Alonso. Trajectory-informed sampling for efficient construction of multi-fidelity surrogate models for hypersonic vehicles. In *AIAA SciTech 2024 Forum*, page 1013, 2024.

[9] Yang Shi and Zhenbo Wang. A deep learning-based approach to real-time trajectory optimization for hypersonic vehicles. In *AIAA SciTech 2020 forum*, page 0023, 2020.

[10] Alexander S Way, Adrian Sescu, Edward Luke, and Ian D Dettwiller. Hypernetwork based surrogate modeling of hypersonic glide vehicle aerothermodynamics. In *AIAA AVIATION FORUM AND ASCEND 2024*, page 4107, 2024.

[11] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.