Journal of Machine Learning Research 27 (2026) 1-63 Submitted 9/25; Revised 6/26; Published 6/26

Prob-GPararealal: A Probabilistic Numerical Parallel-in-Time Solver for Differential Equations

Guglielmo Gattiglio G.GATTIGLIO@GMAIL.COM
Department of Statistics
University of Warwick
Coventry, CV4 7AL, UK

Lyudmila Grigoryeva LYUDMILA.GRIGORYEVA@UNISG.CH
Division of Mathematics and Statistics
University of St. Gallen
St. Gallen, CH-9000, Switzerland

Massimiliano Tamborrino MASSIMILIANO.TAMBORRINO@WARWICK.AC.UK
Department of Statistics
University of Warwick
Coventry, CV4 7AL, UK

Editor: Samuel Kaski

Abstract

We introduce Prob-GPararealal, a probabilistic extension of the GPararealal algorithm designed to provide uncertainty quantification for the Parallel-in-Time (PinT) solution of (ordinary and partial) differential equations (ODEs, PDEs). The method employs Gaussian processes (GPs) to model the Parareal correction function, in line with GPararealal, further enabling the propagation of numerical uncertainty across time and yielding probabilistic forecasts of the system's evolution. Furthermore, Prob-GPararealal accommodates probabilistic initial conditions and maintains compatibility with classical numerical solvers, ensuring its straightforward integration into existing Parareal frameworks. Here, we first conduct a theoretical analysis of the computational complexity and derive error bounds of Prob-GPararealal. Then, we numerically demonstrate the accuracy and robustness of the proposed algorithm on five benchmark ODE systems, including chaotic, stiff, and bifurcation problems. To showcase the flexibility and potential scalability of the proposed algorithm, we also consider Prob-nnGPararealal, a variant obtained by replacing the GPs in Parareal with the nearest-neighbors GPs, illustrating its improved computational performance on an additional PDE example. This work bridges a critical gap in the development of probabilistic counterparts to established PinT methods.

Keywords: Uncertainty Quantification; Gaussian Processes; Probabilistic Solver; parallel-in-time methods; nearest-neighbors Gaussian processes

1. Introduction

Efficient numerical methods for solving differential equations (DEs) are a cornerstone of modern simulation and modeling techniques, with applications ranging from climate, medical, and financial modeling to aerospace engineering and many other fields where understanding dynamic processes is crucial. Among many advances in this field, Parallel-in-Time (PinT)

methods have emerged as a powerful approach to accelerate the solution of large-scale, high-dimensional DEs where traditional parallelization strategies, such as spatial decomposition, reach saturation, preventing full use of available computational resources (Samaddar et al., 2019). PinT techniques address the limitations of conventional sequential solvers by enabling concurrent computations over the time domain, which is especially useful when applied to problems with a long simulation horizon, such as, for example, in the case of molecular dynamics simulations (Gorynina et al., 2023).

Over the past two decades, the PinT field has grown substantially, with several algorithmic classes introduced. Following Gander (2015), these can be categorized according to how they discretize the space-time domain to implement parallelization: multiple shooting, space-time domain decomposition, multigrid methods, and direct solvers. Notable examples include Parareal (Lions et al., 2001), the Parallel Full Approximation Scheme in Space and Time (Emmett and Minion, 2012; Minion, 2011), and Multigrid Reduction in Time (Falgout et al., 2014; Friedhoff et al., 2012), see Gander et al. (2023) for an overview. Among these approaches, Parareal has received the most attention due to its simplicity, flexibility, and demonstrated success in a wide range of applications (Reynolds-Barredo et al., 2012; Samaddar et al., 2010, 2019; Bal and Maday, 2002; Pages et al., 2018; Philippi and Slawig, 2022, 2023). Extensive work on theoretical analysis (Bal, 2005; Gander and Vandewalle, 2007; Gander and Hairer, 2008; Ruprecht, 2018; Staff and Rønquist, 2005; Pentland et al., 2023a), and algorithmic extensions (Haut and Wingate, 2014; Peddle et al., 2019; Legoll et al., 2013; Pentland et al., 2023; Pentland et al., 2023b; Gattiglio et al., 2025, 2024) have further established Parareal as a foundational method in PinT research. This algorithm uses a computationally cheap coarse numerical solver to obtain an approximate sequential solution of the DE system. The intermediate solution is iteratively refined by executing a precise, although expensive, numerical solver in parallel, and correcting the coarse evaluation by accounting for the estimated error between these two solvers.

One of the recent PinT advances, the so-called GParareal, introduced in Pentland et al. (2023b), is especially relevant for this work. By using Gaussian processes (GPs) to learn the solver's discrepancy from Parareal's past iteration data, GParareal approximates the Parareal correction function using the GP posterior mean, yielding substantial speed-ups over the canonical Parareal algorithm. This approach was later extended to utilize nearest-neighbor GPs (nnGPs) in Gattiglio et al. (2025), resulting in the nnGParareal algorithm, offering improved scalability for higher-dimensional systems and an increased number of discretization points in the time domain. High-quality and numerically efficient approximation of the Parareal correction function has been further pursued using other families of models with universal approximation properties. For example, RandNet-Parareal, proposed in Gattiglio et al. (2024), uses shallow random weights neural networks to learn the solvers' discrepancy, allowing its application to partial differential equations (PDEs), accommodating up to 10^5 spatial discretization points. It is worth noting that these algorithms yield deterministic solvers. However, while the RandNet-Parareal method is deterministic by construction, GParareal and its extension have probabilistic foundations. Indeed, although GParareal (Pentland et al., 2023b) relies solely on the prediction with the posterior mean of the GP, its posterior covariance structure contains important uncertainty information that the resulting solver could potentially leverage.

During the last two decades, the field of probabilistic numerics has also experienced rapid development. Although its origins can be traced back to the pioneering works of Suldin (1959) and Larkin (1972) in the second half of the 20th century (see Oates and Sullivan 2019 for a review), it is in recent years that it has experienced a surge of interest, advancing on a broad front: quadrature methods (Minka, 2000; Särkkä et al., 2014; Xi et al., 2018), linear algebra (Selig et al., 2012; Fitzsimons et al., 2017), global (Hennig and Schuler, 2012) and local (Mahsereci and Hennig, 2017) optimization, and DEs (Kersting et al., 2020; Krämer and Hennig, 2024; Tronarp et al., 2019; Teymur et al., 2018; Abdulle and Garegnani, 2020). Probabilistic numerics seeks to quantify epistemic uncertainty arising from intractable or incomplete numerical computation (Hennig et al., 2022). For instance, quadrature methods and DE solvers rely on finite evaluations of the integrand and vector field, respectively, while the exact solution would theoretically require infinitely many of those. By framing numerical problems as inference tasks, probabilistic numerics provides a principled approach to uncertainty quantification (UQ), yielding uncertainty-aware computation tools.

In the specific context of ordinary differential equations (ODEs), probabilistic numerical methods can be classified into *ODE filters and smoothers*, based on GP regression, and *perturbative solvers*, which characterize uncertainty through perturbation of classic numerical methods (Hennig et al., 2022). The cost of the former is cubic in the number of time steps, and, although an approximate computation of GP regression using Bayesian filters can be achieved in linear time (Kersting and Hennig, 2016; Kersting et al., 2020; Schober et al., 2019; Krämer and Hennig, 2024), these methods underperform in representing more complex dynamics (for example, for chaotic systems, see Hennig et al. 2022; Tronarp et al. 2019). At the same time, perturbative solvers are more expressive, though more numerically expensive, requiring multiple simulations of the ODE.

Contribution. In this work, we bridge the fields of PinT computation and UQ by introducing Prob-GParareal, which, to the best of our knowledge, is the first probabilistic extension of the (G)Parareal algorithm. The main idea consists in modeling the uncertainty in the Parareal update rule using GPs, and propagating it nonlinearly through time via sampling. Unlike GParareal (Pentland et al., 2023b), which exclusively exploits the posterior mean of GPs, Prob-GParareal also effectively uses additional probabilistic information provided by their posterior covariance. Our proposed probabilistic solver, Prob-GParareal, offers several key advantages over its deterministic counterparts:

- *Uncertainty quantification*. Prob-GParareal produces a probabilistic forecast of the system's evolution, explicitly quantifying the dynamics of the error across both the temporal domain and the algorithm's iterations. We demonstrate the accuracy of our method for both non-chaotic and chaotic systems, addressing the well-documented challenges of ODE filters in accurately representing chaotic behavior (Hennig et al., 2022; Tronarp et al., 2019).

- *Support for random initial conditions*. Prob-GParareal generalizes deterministic initial value problems (IVPs) to random initial conditions under non-restrictive assumptions. This extension enables its use for systems where precise information about the initial condition is not available.

- *Solver compatibility*. Our method is agnostic to the choice of the numerical solver used, and it can be integrated with any existing (deterministic) Parareal implementation without major modifications to the simulation specifics. This flexibility is an important feature, as stability guarantees in Parareal applications often require problem-specific solvers (Krzysik et al., 2025; Ruprecht, 2018). Since these can be directly embedded in our approach, Prob-GPareareal naturally inherits enhanced stability properties.

- *Flexible and controlled resource allocation*. Prob-GPareareal enables flexible control over computational resources by supporting early termination, either based on predefined solution variance thresholds or computational budget constraints, without requiring full convergence of the algorithm. Empirical results suggest that probabilistic forecasts remain well-calibrated under early termination, with the predictive variance accurately reflecting the uncertainty induced by incomplete convergence.

- *Scalability*. To achieve scalable performance under an increasing number of processors and DE dimensions, we build upon Gattiglio et al. (2025) and extend our Prob-GPareareal framework to Prob-nnGPareareal using nnGPs, improving the computational efficiency of our probabilistic solver.

While probabilistic numerical methods have seen extensive development in other domains, their application to PinT methods remains unexplored, with the notable exceptions of Bosch et al. (2024); Iqbal et al. (2024). The authors leverage the associativity of the Bayesian smoothing operator (Särkkä and García-Fernández, 2020) to achieve temporal parallelization. Their method differs from Prob-GPareareal in several key aspects. First, their algorithm relies on extended Kalman filtering and smoothing. This technique provides exact solutions for affine ODEs, but requires first-order Taylor approximations of nonlinear vector fields. Second, their approach to UQ is intrinsically tied to the Bayesian smoothing procedure. Our proposed Prob-GPareareal does not have these constraints, and can be applied to nonlinear and/or chaotic ODEs/PDEs in combination with any classical solver.

The paper is organized as follows. In Section 2, we review the Parareal and (nn)GPareareal algorithms. In Section 3, we introduce our novel method, Prob-GPareareal, present its probabilistic formulation, algorithmic structure, and computational complexity. In Section 4, we provide its theoretical and error bound properties. Section 5 validates our novel framework through numerical experiments on five benchmark ODE systems that exhibit stiff behavior, bifurcations, and chaotic dynamics. The extension of the Prob-GPareareal algorithm to systems with probabilistic initial conditions is contained in Section 5.4. In Section 6, we demonstrate the scalability improvements and the numerical efficiency increase achieved with Prob-nnGPareareal. Conclusions and future directions are presented in Section 7.

2. Background: Parareal and (nn)GPareareal

Without loss of generality, to simplify the notation, we focus on autonomous IVPs (a non-autonomous PDE is considered in Section 6), described by a system of d ODEs, $d \in \mathbb{N}$,

$$\frac{d\boldsymbol{u}}{dt} = h(\boldsymbol{u}(t)), \quad t \in [t_0, t_N], \quad \boldsymbol{u}(t_0) = \boldsymbol{u}_{(0)}, \tag{1}$$

where $\boldsymbol{u} : [t_0, t_N] \to \mathbb{R}^d$, $N \in \mathbb{N}$, is the time-dependent vector solution, $h : \mathbb{R}^d \to \mathbb{R}^d$ is a smooth multivariate function, and $\boldsymbol{u}_{(0)} \in \mathbb{R}^d$ is the known initial value at time t_0, which we assume deterministic, unless otherwise stated. As an exact solution to (1) is generally not available, one typically relies on a numerical solver $\mathscr{F}$ to obtain a high-accuracy numerical solution. Depending on the system (1) and the length of the interval over which it is integrated, the sequential application of $\mathscr{F}$ may be computationally infeasible. The Parareal algorithm offers a remedy to this by partitioning the time domain into N sub-intervals (usually of equal length), so that the problem can be split into N IVPs given by:

$$\frac{d\boldsymbol{u}_i}{dt} = h\left(\boldsymbol{u}_i(t)\right), \quad t \in [t_i, t_{i+1}], \quad \boldsymbol{u}_i(t_i) = \boldsymbol{u}_{(i)}, \quad i = 0, \dots, N-1, \tag{2}$$

with $\boldsymbol{u}_{(i)} = \varphi_{\Delta t_i}(\boldsymbol{u}_{(i-1)})$, $i = 1, \dots, N-1$, where $\Delta t_i = t_i - t_{i-1}$ is the ith time step and $\varphi_{\Delta t_i} : \mathbb{R}^d \to \mathbb{R}^d$ denotes the Δt_i-time flow map (i.e. the solution) of the ith IVP with initial condition $\boldsymbol{u}_{(i-1)}$ after time Δt_i. Since only $\boldsymbol{u}_{(0)}$ is known, the other initial conditions $\boldsymbol{u}_{(i)}$, $i = 1, \dots, N-1$, would need to be estimated. To do this, Parareal relies on an iterative scheme using a faster but less accurate coarse solver $\mathscr{G}$. At iteration 0, the initial conditions $\boldsymbol{u}_{(i)}$ are approximated as $\boldsymbol{u}_{i,0} = \mathscr{G}(\boldsymbol{u}_{i-1,0})$, $i = 1, \dots, N-1$, where $\boldsymbol{u}_{i,k}$ denotes the Parareal solution at time t_i and iteration k. These approximations at iteration $k = 0$ are then updated *sequentially*, for iteration $k \geq 1$, using the Parareal predictor-corrector rule

$$\boldsymbol{u}_{i,k} = \mathscr{G}(\boldsymbol{u}_{i-1,k}) + (\mathscr{F} - \mathscr{G})(\boldsymbol{u}_{i-1,k-1}), \quad i = 1, \dots, N, \tag{3}$$

where $\mathscr{F}(\boldsymbol{u}_{i-1,k-1})$ is computed *in parallel* over N processors. Parareal is said to ϵ-converge at iteration k for some chosen accuracy level $\epsilon > 0$ whenever

$$\max_{1 \leq i \leq N-1} \|\boldsymbol{u}_{i,k} - \boldsymbol{u}_{i,k-1}\|_\infty < \epsilon. \tag{4}$$

Recent contributions modify (3) by using the current kth iteration data $\boldsymbol{u}_{i-1,k}$, instead of $\boldsymbol{u}_{i-1,k-1}$ at iteration $k-1$, and by modeling the correction (discrepancy) function

$$f_c := (\mathscr{F} - \mathscr{G}) : \mathbb{R}^d \to \mathbb{R}^d, \tag{5}$$

using alternative techniques. In particular, GParareal (Pentland et al., 2023b) and nnG-Parareal (Gattiglio et al., 2025) approximate this function using d independent scalar GPs and nnGPs, respectively. More precisely, in GParareal, each sth coordinate of the correction function f_c, is modeled as

$$f_c^{(s)} = (\mathscr{F} - \mathscr{G})^{(s)} \sim GP(0, K_{\text{GP}}), \;\; s = 1, \dots, d, \tag{6}$$

that is, a one-dimensional GP with zero mean and variance kernel function $K_{\text{GP}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. The GPs are then trained on the accumulated dataset $\mathcal{D}_k$ (or a subset for nnGParareal) defined as

$$\mathcal{D}_k = \{(\boldsymbol{u}_{i-1,j}, f_c(\boldsymbol{u}_{i-1,j})) \,|\, i = 1, \dots, N, \; j = 0, \dots, k-1\}, \quad k \in \mathbb{N},$$

leading to the following posterior distribution for the sth coordinate of a point $\boldsymbol{u}' \in \mathbb{R}^d$,

$$f_c^{(s)}(\boldsymbol{u}') | \mathcal{D}_k \sim \; \mathcal{N}(\boldsymbol{\mu}_{\mathcal{D}_k}^{(s)}(\boldsymbol{u}'), \sigma_{\mathcal{D}_k}^{(s)}(\boldsymbol{u}')^2), \tag{7}$$

with posterior mean $\boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}') \in \mathbb{R}$ and posterior variance $\sigma^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}')^2 \in \mathbb{R}^+$, whose expressions are given in Appendix A, when K_{GP} is a Gaussian kernel, also known as the radial basis function or square exponential kernel (an alternative kernel, the Matérn kernel, popular in spatial statistics analysis (Matérn, 1986), is also presented there). The Parareal update rule (3) for GPararеal is then constructed by taking the posterior means of the GPs as predictions of the discrepancy as follows:

$$\boldsymbol{u}_{i,k} = \mathscr{G}(\boldsymbol{u}_{i-1,k}) + \widehat{f}_{\text{GPara}}(\boldsymbol{u}_{i-1,k}), \tag{8}$$

where $\widehat{f}_{\text{GPara}}(\boldsymbol{u}_{i-1,k}) = \left(\widehat{f}^{(1)}_{\text{GPara}}(\boldsymbol{u}_{i-1,k}), \dots, \widehat{f}^{(d)}_{\text{GPara}}(\boldsymbol{u}_{i-1,k})\right)^\top \in \mathbb{R}^d$ is the vector of posterior means, with $\widehat{f}^{(s)}_{\text{GPara}}(\boldsymbol{u}_{i-1,k}) := \boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}_{i-1,k})$, $s = 1, \dots, d$.

The framework is unchanged for nnGPararеal, except that the nearest neighbors are recomputed for each test point $\boldsymbol{u}' \in \mathbb{R}^d$, and nnGPs are trained on these neighbors instead of the entire $\mathcal{D}_k$ before making a prediction $\widehat{f}_{\text{nnGPara}}$. Although GPararеal (and similarly nnGPararеal) uses only the posterior mean in (8), the GP framework naturally provides uncertainty estimates through its posterior variance in (7). This feature motivates our proposed probabilistic extension described in Section 3.

3. Prob-GPararеal: a probabilistic Parareal framework

The primary source of error in the Parareal algorithm is the coarse solver $\mathscr{G}$, which is inaccurate by design. In the context of sequential solvers of deterministic DEs, the true solution is recovered in the limit as the time step $\Delta t \to 0$ for convergent numerical schemes. However, infinitesimal time steps are not computable, and, in practice, finite Δt leads to arbitrarily small errors consistent with the scheme's order of accuracy. Given an autonomous ODE with a unique solution and equidistant time steps $\Delta t = t_i - t_{i-1}, i = 1, \dots, N$, the coarse solver solution to the IVP starting in $\boldsymbol{u}_{i-1,k}$ can be written as

$$\mathscr{G}(\boldsymbol{u}_{i-1,k}) = \varphi_{\Delta t}(\boldsymbol{u}_{i-1,k}) + \epsilon^{ep}_{\mathscr{G}}(\boldsymbol{u}_{i-1,k}), \quad i = 1, \dots, N,$$

where $\epsilon^{ep}_{\mathscr{G}}(\boldsymbol{u}_{i-1,k})$ is the numerical error associated with $\boldsymbol{u}_{i-1,k}$ and φ is the Δt-flow map defined as in (2). Taking $\mathscr{F}$ as φ is equivalent to assuming that $\mathscr{F}$ is sufficiently accurate to represent the true solution to (1), a common assumption in the (theoretical and applied) Parareal literature (Gander and Hairer, 2008; Pentland et al., 2023b). Under this assumption, the correction function f_c in (5) accounts for all sources of uncertainty. While a direct application of f_c is unfeasible, as it would require *sequential* runs of $\mathscr{F}$, GPs could be used instead, as described in Section 2. In particular, the GPararеal predictor-corrector rule (8) can be readily extended by including the posterior distribution of the GP, to model the error incurred by approximating f_c. This leads to random solutions $\{\boldsymbol{U}_{i,k}\}_{1 \le i \le N, k \in \mathbb{N}}$ (with $\boldsymbol{u}_{i,k}$ representing one of their possible outcomes/draws/realizations) that define a process, Markov conditionally on the training data $\mathcal{D}_k$, with the law of $\boldsymbol{U}_{i,k}$ conditionally independent of $\boldsymbol{U}_{j,\ell}$, given $\boldsymbol{U}_{i-1,k}$, for $j \le i-2$, $\ell \neq k$. In particular, the probabilistic update rule for $\boldsymbol{U}_{i,k}$ is then given by

$$\boldsymbol{U}_{i,k} = \mathscr{G}\left(\boldsymbol{U}_{i-1,k}\right) + \boldsymbol{Z}_{i,k}, \quad i = 1, \dots, N, \quad k \ge 1, \tag{9}$$

with

$$\boldsymbol{Z}_{i,k} | (\boldsymbol{U}_{i-1,k} = \boldsymbol{u}_{i-1,k}, \mathcal{D}_k) \ \sim \mathcal{N}_d(\boldsymbol{\mu}_{\mathcal{D}_k}\left(\boldsymbol{u}_{i-1,k}\right), \Sigma_{\mathcal{D}_k}(\boldsymbol{u}_{i-1,k})), \tag{10}$$

where $\mathcal{N}_d(\boldsymbol{a}, B)$ denotes a d-dimensional normal distribution with mean vector $\boldsymbol{a} \in \mathbb{R}^d$ and $d \times d$-dimensional covariance matrix B, and $\boldsymbol{Z}_{i,k}$ is conditionally independent of $\boldsymbol{U}_{j,\ell}$ given $\boldsymbol{U}_{i-1,k}$ for all $j \leq i-2$ and $\ell \neq k$. Once a realization $\boldsymbol{u}_{i-1,k}$ of $\boldsymbol{U}_{i-1,k}$ is given, $\boldsymbol{Z}_{i,k}$ represents the only source of uncertainty in (9).

From the update rule (9), it is immediate to see that the conditional distribution of $\boldsymbol{U}_{i,k}$, given $\boldsymbol{U}_{i-1,k} = \boldsymbol{u}_{i-1,k}$, follows a d-dimensional normal distribution

$$\boldsymbol{U}_{i,k} | \boldsymbol{U}_{i-1,k} = \boldsymbol{u}_{i-1,k} \sim \mathcal{N}_d \left(\mathscr{G}(\boldsymbol{u}_{i-1,k}) + \boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{u}_{i-1,k}), \Sigma_{\mathcal{D}_k}(\boldsymbol{u}_{i-1,k}) \right). \tag{11}$$

Instead, the unconditional distribution of $\boldsymbol{U}_{i,k}$ is unknown. Even if it were known and Gaussian, determining the distribution of $\boldsymbol{U}_{i+1,k}$ via (9) using $\mathscr{G}$ would be neither theoretically possible nor computationally feasible for nonlinear IVPs (Pentland et al., 2023b). This is why Bayesian/probabilistic numerics ODE filters and smoothers linearize the vector field of nonlinear ODEs, resulting in approximate solutions (see, e.g. Hennig et al. 2022). In the following subsections, we propose a sampling scheme overcoming these limitations, circumventing the need for linearization.

3.1 Derivation of Prob-GParareal

The update rule (11) allows modeling the correlation between the components of $\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}$ with multi-output GPs. However, the computational costs associated with such a flexible approach may be significantly high, with marginal, if any, improvements in terms of accuracy and the number of iterations until the algorithm converges, as discussed in Pentland et al. (2023b) for GParareal. Moreover, for systems of DEs, one of the simplest multi-output GP extensions, the intrinsic coregionalization model (Goovaerts, 1997), has been shown to be equivalent to independent GPs' predictions over each coordinate, with each GP trained on the same dataset (Alvarez et al., 2012; Wackernagel, 2003). Hence, we adopt the same strategy here for Prob-GParareal and, similarly to GParareal, assume that the entries of $\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}$ are uncorrelated and, thus, independent, given the underlying multivariate Gaussian distribution. More precisely, the covariance matrix $\Sigma_{\mathcal{D}_k}$ in (11) is assumed to be diagonal with diagonal components $\sigma^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}_{i-1,k})^2$, $s = 1, \ldots, d$ as in (7).

We now formalize the Prob-GParareal sampling procedure to obtain realizations of $\boldsymbol{U}_{i,k}$. Let $P_{\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}}$, $i = 1, \ldots, N$, $k \in \mathbb{N}$, denote the conditional distribution of $\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}$ given in (11) (in some cases, we explicitly write $P_{\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}=\boldsymbol{u}_{i-1,k}}$ to specify a particular realization $\boldsymbol{u}_{i-1,k}$), and let $P_{\boldsymbol{U}_{i,k}}$ be the unconditional marginal distribution of $\boldsymbol{U}_{i,k}$, given by

$$P_{\boldsymbol{U}_{i,k}}(\boldsymbol{u}_{i,k}) = \int_{\mathbb{R}^d} P_{\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}}(\boldsymbol{u}_{i,k} \mid \boldsymbol{u}_{i-1,k}) \, P_{\boldsymbol{U}_{i-1,k}}(\boldsymbol{u}_{i-1,k}) d\boldsymbol{u}_{i-1,k}, \tag{12}$$

which, in general, cannot be solved analytically. However, it can be interpreted as a continuous Gaussian mixture, for which sampling is feasible via many available procedures. Here, we employ a two-step procedure known as ancestral sampling (Deisenroth et al., 2020, Chapter 11), designed to draw $n \in \mathbb{N}$ *observed* samples, or samples of *observations/realizations*, $\mathcal{U}_{i,k} = \{\boldsymbol{u}^{(j)}_{i,k}\}_{j=1}^n$, from $P_{\boldsymbol{U}_{i,k}}$, $i = 1, \ldots, N$, $k \in \mathbb{N}$, in (12) using (11), as follows.
For $i = 1, \ldots, N$, $k \in \mathbb{N}$:

1. Sample independently n observations $\boldsymbol{u}^{(1)}_{i-1,k}, \ldots, \boldsymbol{u}^{(n)}_{i-1,k}$ from the marginal distribution $P_{\boldsymbol{U}_{i-1,k}}$.

2. Given $\boldsymbol{U}_{i-1,k} = \boldsymbol{u}_{i-1,k}^{(j)}$, sample $\boldsymbol{u}_{i,k}^{(j)}$ from $P_{\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}=\boldsymbol{u}_{i-1,k}^{(j)}}$ in (11), $j = 1, \dots, n$.

This sampling procedure is then embedded within the GParareal framework, as described in the following Subsection.

3.2 Algorithm

The schematic description of Prob-GParareal is provided in Algorithm 1. The algorithm is initialized at iteration $k = 0$ by drawing n initial values from the initial distribution $P_{\boldsymbol{U}_{0,0}}$ (Line 2). When the initial condition $\boldsymbol{u}_{(0)}$ is deterministic, as considered in the experiment results in Section 5 (other than Section 5.4) and Section 6, we set $P_{\boldsymbol{U}_{0,0}} = \delta_{\boldsymbol{u}_{0,0}}$, a Dirac measure centered at $\boldsymbol{u}_{0,0} = \boldsymbol{u}_{(0)}$, leading to $\boldsymbol{u}_{0,0}^{(j)} = \boldsymbol{u}_{(0)}$ and $\boldsymbol{u}_{i,0}^{(j)} = \mathscr{G}(\boldsymbol{u}_{i-1,0}^{(j)}), i = 1, \dots, N$, $j = 1, \dots, n$, otherwise the n sampled values are propagated sequentially via $\mathscr{G}$ (Line 5), leading to $\mathcal{U}_{i,0} = \{\boldsymbol{u}_{i,0}^{(j)}\}_{j=1}^{n}, i = 0, \dots, N$, collection of observed samples at iteration $k = 0$.

After the initialization phase, a recursive execution consisting of a five-step procedure is launched, running until either the algorithm converges or an early stopping criterion is met, as described below.

Let L be the number of converged intervals, which is initially set to 0. At iteration k and interval $i = L, \dots, N$, the sample means $\overline{\boldsymbol{u}}_{i,k-1}$ of the observation samples $\mathcal{U}_{i,k-1}$ at iteration $k-1$ are first computed and then propagated through $\mathscr{G}$ and $\mathscr{F}$ *in parallel* (Lines 12-13, **Step 1**). The pairs $(\overline{\boldsymbol{u}}_{i,k-1}, f_c(\overline{\boldsymbol{u}}_{i,k-1}))$, $i = L, \dots, N$, are then added to the dataset $\mathcal{D}_{k-1}$, yielding the updated dataset $\mathcal{D}_k$ (Line 15, **Step 2**), which is then used to train d scalar GPs to obtain the estimates of the posterior mean $\boldsymbol{\mu}_{\mathcal{D}_k}(\cdot)$ and posterior covariance $\Sigma_{\mathcal{D}_k}(\cdot)$ functions (Line 16, **Step 3**). After that, the Prob-GParareal predictor-corrector rule $\boldsymbol{U}_{i,k} = \mathscr{G}(\boldsymbol{U}_{i-1,k}) + \boldsymbol{Z}_{i,k}$ in (9) is applied by drawing, *in parallel* over $j = 1, \dots, n$, a realization $\boldsymbol{z}_{i,k}^{(j)}$ from the multivariate conditional Gaussian distribution $\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k} = \boldsymbol{u}_{i-1,k}^{(j)}$[1], and then computing, sequentially over $i = L+1, \dots, N$, the resulting sampled predictor-corrector values $\boldsymbol{u}_{i,k}^{(j)} = \mathscr{G}(\boldsymbol{u}_{i-1,k}^{(j)}) + \boldsymbol{z}_{i,k}^{(j)}$ (Lines 17-23, **Step 4**). This defines the resulting sampled values $\mathcal{U}_{i,k} = \{\boldsymbol{u}_{i,k}^{(j)}\}_{j=1}^{n}$ at iteration k (Line 22). At this point, (**Step 5**):

- We check whether the Prob-GParareal solutions have converged up to some interval l, $L+1 \leq l \leq N$. Intuitively, the Prob-GParareal solutions $\mathcal{U}_{i,k}$, $i = L+1, \dots, N$ have converged up to time $t_l \leq t_N$, with $i \leq l \leq N$, if the sampled points in $\mathcal{U}_{i,k-1}$ and $\mathcal{U}_{i,k}$ at interval $i = L+1, \dots, l$ and iteration $k-1$ and k, respectively, are "close enough". To formalize this, we choose an accuracy threshold $\epsilon > 0$, a statistic $g : (\mathbb{R}^d)^n \to \mathcal{X}$, for some metric space $\mathcal{X}$, applied to both $\mathcal{U}_{i,k-1}$, and $\mathcal{U}_{i,k}$, and a distance metric $d_{\mathcal{U}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, which measures their similarity. In our implementation and theoretical analysis, g is defined as the empirical measure with $\widehat{P}_{\mathcal{U}_{i,k}} := g(\mathcal{U}_{i,k}) = \frac{1}{n}\sum_{j=1}^{n} \delta_{\boldsymbol{u}_{i,k}^{(j)}}$. Additionally, $d_{\mathcal{U}}$ is chosen as the p-power Wasserstein-p ($p \in \mathbb{N}$) distance (Villani, 2008) between these empirical measures. We say that the

1. Note that, given the chosen GP implementation consisting of d independent scalar GPs, see (5)-(7), we sample the sth coordinate of $\boldsymbol{z}_{i,k}^{(j)}$ from $\mathcal{N}\left(\boldsymbol{\mu}_{\mathcal{D}_k}^{(s)}(\boldsymbol{u}_{i-1,k}^{(j)}), \sigma_{\mathcal{D}_k}^{(s)}(\boldsymbol{u}_{i-1,k}^{(j)})^2\right)$.

Prob-GPararealal solution has ϵ-converged up to iteration l when

$$W_p(\widehat{P}_{\mathcal{U}_{i,k}}, \widehat{P}_{\mathcal{U}_{i,k-1}})^p = \min_{\pi \in \Pi} \left(\frac{1}{n} \sum_{j=1}^{n} \left\| \boldsymbol{u}_{i,k}^{(j)} - \boldsymbol{u}_{i,k-1}^{(\pi(j))} \right\|^p \right) < \epsilon, \quad 0 \leq L < i \leq l \leq N, \tag{13}$$

where Π is the set of all permutations of $\{1, 2, \ldots, n\}$. If this happens, we set $L = l$ and, if $L < N$, continue the execution run. Here, the tolerance ϵ is expressed on the pth-power scale.

- We verify whether an early termination rule, based on practical constraints, such as maximum solution variance or computational budget limits, is met. If so, the algorithm terminates and the solutions $\mathcal{U}_{i,k}$, $i = 0, \ldots, N$, are returned.

Remark 1 *The Prob-GPararealal algorithm can be easily modified to evaluate $\mathscr{F}$ and $\mathscr{G}$ in parallel on other summaries (e.g., median or a certain quantile of $\mathcal{U}_{i,k-1}$) than the sample mean $\overline{\boldsymbol{u}}_{i,k-1}$ (Lines 12-13 of Algorithm 1). One could also potentially propagate all n samples through $\mathscr{F}$ for each ith time interval (as done in SPararealal by Pentland et al. 2023), but this would require nN processors, which is a prohibitively expensive requirement in practice. Despite the limitation of considering only a summary of the sample, we derive theoretical error bounds in Section 4, and numerically illustrate the algorithm's convergence on several models in Section 5.*

Remark 2 *In* (13)*, we used empirical measures and the Wasserstein-p distance between them. This distance, computed with the Hungarian (Kuhn-Munkres) algorithm, typically has an $O(n^3)$ complexity, although certain geometric assumptions or approximation techniques can mitigate this cost (see, e.g., Bernton et al. 2019 and Cuturi 2013). In our implementation, rather than computing the exact Wasserstein distance between the empirical measures in* (13)*, we approximate each empirical distribution by a multivariate Gaussian with its sample mean and covariance matrix and compute the squared Wasserstein-2 distance between these Gaussian approximations (cost of $O(nd)$ is achieved under diagonal covariance structure). Other statistics and distance metrics could be considered. For instance, multivariate statistics $g : (\mathbb{R}^d)^n \to \mathbb{R}^q$, $q \in \mathbb{N}$, paired with an appropriate metric $d_{\mathcal{U}} : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}^+$ represent a viable alternative. Among other probability metrics for empirical measures (see Sriperumbudur et al. 2009 for an overview), the Maximum Mean Discrepancy (MMD) (Gretton et al., 2007) stands out due to its computational efficiency, with an $O(n^2)$ complexity (Sriperumbudur et al., 2010), which can be further reduced via approximation methods. Despite the computational advantages and the availability of simple upper and lower bounds linking MMD to the Wasserstein-p distance (Sriperumbudur et al., 2010), its performance is more sensitive to ad-hoc tuning of the kernel bandwidth, which is why it is not considered here.*

Remark 3 *The Prob-GPararealal algorithm introduces two complementary stopping mechanisms in Step 5; the ϵ-convergence, controlling the accuracy of the desired solution (Line 25 in Algorithm 1); an independent termination rule (that we call the exit condition, see Line 28 in Algorithm 1). By explicitly accounting for the uncertainty of the solution throughout its execution, Prob-GPararealal allows these two stopping criteria to operate simultaneously. We explore the impact of early termination of the algorithm in Section 5.3.*

3.3 Computational complexity

The computational cost of Prob-GParareal, denoted as $T_{\text{Prob-GPara}}$, can be divided into that of running $\mathscr{F}$ and $\mathscr{G}$ over one interval $[t_i, t_{i+1}]$, $i = 0, \dots, N-1$, denoted by $T_{\mathscr{F}}$ and $T_{\mathscr{G}}$, respectively, and that of computing the correction function f_c in (5) during iteration $k \in \mathbb{N}$, denoted as $T_f^{\text{Prob-GPara}}(k)$. At iteration k, the d GPs are trained in parallel on the dataset $\mathcal{D}_k$, containing up to Nk observations, and are used to sample $\boldsymbol{z}_{i,k}^{(j)}$, $j = 1, \dots, n$, for up to N intervals, which yields the following complexity:

$$T_f^{\text{Prob-GPara}}(k) = \underbrace{(\tfrac{d}{N} \vee 1)\, O\left(d(Nk)^2 + (Nk)^3\right)}_{\text{Training}} + \underbrace{N\, (\tfrac{dn}{N} \vee 1)\, O\left(dNk + (Nk)^2\right)}_{\text{Sampling of } \{\boldsymbol{z}_{i,k}^{(j)}\}_{j=1}^n},$$

where $\vee$ denotes the maximum operator. The factors $(\frac{d}{N} \vee 1)$ and $(\frac{dn}{N} \vee 1)$ result from parallelizing computations across the d GPs. The training term complexity $O(d(Nk)^2)$ corresponds to constructing the Nk-dimensional symmetric kernel matrix, while the $O((Nk)^3)$ term accounts for its inversion, with both operations carried out once per GP. In the sampling term, the complexities $O(dNk)$ and $O((Nk)^2)$ represent the cost of evaluating the posterior mean and variance, respectively, repeated across each GP and each sample $j = 1, \dots, n$ *in parallel*. Moreover, the N factor in the sampling term accounts for *sequential* repetition of sampling for up to N intervals.

At iteration k, the training cost of Prob-GParareal is the same as that of GParareal. Instead, the sampling cost of GParareal is lower, being $(N-k)$ $(\frac{d}{N} \vee 1)$ $O(dNk)$ (Pentland et al., 2023b) as it does not need to evaluate the posterior variance. This is due to the fact that GParareal applies the discrepancy function only once per interval, with no need to estimate the variance. The factor $(N-k)$ follows from the application of $\mathscr{F}$ directly on the solution $\boldsymbol{u}_{i-1,k}$, rather than on the mean $\bar{\boldsymbol{u}}_{i-1,k}$ as in Prob-GParareal. For moderate values of d and when $\frac{n}{N} \in O(k)$, the training and sampling steps in Prob-GParareal share the same cubic complexity, indicating no additional overhead for uncertainty estimation compared to the deterministic GParareal.

To mitigate the cubic inversion cost of GPs, we follow the approach proposed by Gattiglio et al. (2025) in nnGParareal, using nnGPs trained on a reduced dataset consisting of the m observations (typically $m \approx 15 \ll Nk$) that are nearest neighbors in terms of Euclidean distance to the prediction point $\boldsymbol{u}_{i-1,k}^{(j)}, j = 1, \dots, n$. Although this method would require re-training the nnGPs for each of the n new prediction points, as the data subset may change, we empirically observe that most of the n observations belonging to the sample $\mathcal{U}_{i,k}$ at interval i share the same nearest neighbors. For this reason, we train the nnGP once per interval i, obtaining the following worst-case cost per iteration for the Prob-nnGParareal algorithm:

$$T_f^{\text{Prob-nnGPara}}(k) = \underbrace{N}_{\text{For each } i} \underbrace{(\tfrac{d}{N} \vee 1)\, O\left(dm^2 + m^3\right)}_{\text{Training once}} + \underbrace{N\, (\tfrac{dn}{N} \vee 1)\, O\left(dm + m^2 + \log(Nk)\right)}_{\text{Sampling of } \{\boldsymbol{z}_{i,k}^{(j)}\}_{j=1}^n},$$

where $\log(Nk)$ is the cost associated with maintaining a kd-tree structure (Bentley, 1975) for fast nearest neighbor computation (Gattiglio et al., 2025). Similar considerations of the training costs of Prob-GParareal and GParareal hold when comparing Prob-nnGParareal and nnGParareal.

Specifically, the cost of Prob-nnGParareal includes the additional term m^2 to compute the covariance matrix, and the parallelizable cost of making n posterior evaluations per interval i. Overall, Prob-nnGParareal is considerably faster than Prob-GParareal, leading to similar results, albeit with slightly higher uncertainty in the algorithm solution due to the additional approximation, as shown in Section 6.

As mentioned above, the total computational cost $T_{\text{Prob-GPara}}$ of Prob-GParareal combines the cost of running $\mathscr{F}$ and $\mathscr{G}$, with the cost of computing the correction function. A validation of the cost model against observed wall-clock time, including scaling and speedup results with respect to the number of time intervals N, is reported in Appendix I. Let K_{conv} be the number of iterations required for the algorithm to converge. Prob-GParareal performs up to N parallel evaluations of $\mathscr{F}$ per iteration $k = 1, \ldots, K_{\text{conv}}$, and n parallel evaluations of $\mathscr{G}$ per interval i, yielding an approximate worst-case total computational cost, ignoring serial overheads, of

$$T_{\text{Prob-GPara}} \approx K_{\text{conv}} T_{\mathscr{F}} + (\tfrac{n}{N} \vee 1)(K_{\text{conv}} + 1) N T_{\mathscr{G}} + \sum_{k=1}^{K_{\text{conv}}} T_f^{\text{Prob-GPara}}(k) + N K_{\text{conv}} C_{\text{dist}},$$

where C_{dist} is the general cost of computing the statistics in (13), which is $O(n^3)$ for the Wasserstein-p case, and $O(nd)$ for the special case $p = 2$ under Gaussian assumption, as discussed in Remark 2.

The overall Prob-GParareal cost is slightly higher than that of GParareal (i.e., $T_{\text{GPara}} = K_{\text{conv}} T_{\mathscr{F}} + (K_{\text{conv}} + 1)(N - K_{\text{conv}}/2) T_{\mathscr{G}} + T_f$, Pentland et al. 2023b), due to the execution of n applications of $\mathscr{G}$ per interval during the sequential update procedure. With sufficient computational resources, the n runs of $\mathscr{G}$ can be parallelized, reducing the total cost of Prob-GParareal to that of GParareal.

4. Theoretical analysis

In this section, we derive two main theoretical results for Prob-GParareal: an error bound of its solution to the true one in terms of W_2^2 (Theorem 13), and an error bound between the Prob-GParareal mean behavior and the GParareal solution (Theorem 21), which we use to provide a theoretical comparison between the two algorithms.

4.1 Notation

Here, we adopt the notation introduced, for example, in Zhou (2008). Given $p \in \mathbb{N}, \alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ and $|\alpha| = \sum_{s=1}^d \alpha_s$, we define the index set $I_p := \left\{\alpha \in \mathbb{N}^d : |\alpha| \leq p\right\}$. For a function $f : \mathbb{R}^d \to \mathbb{R}^d$, we denote the αth partial derivative of its sth coordinate, if it exists, as

$$D^\alpha f^{(s)}(\boldsymbol{u}) = \frac{\partial^{|\alpha|}}{\partial u_1^{\alpha_1} \partial u_2^{\alpha_2} \cdots \partial u_d^{\alpha_d}} f^{(s)}(\boldsymbol{u}), \quad \text{for all } \boldsymbol{u} \in \mathbb{R}^d.$$

Given the set $C^p\left(\mathbb{R}^d, \mathbb{R}^d\right)$ of p-continuously differentiable functions from $\mathbb{R}^d$ to $\mathbb{R}^d$, let $C_b^p\left(\mathbb{R}^d, \mathbb{R}^d\right)$ be the set of the corresponding bounded p-continuously differentiable functions

with bounded derivatives given by

$$C_b^p(\mathbb{R}^d,\mathbb{R}^d)=\left\{f\in C^p(\mathbb{R}^d,\mathbb{R}^d):\|f\|_{C_b^p}:=\max_{1\le s\le d}\|f^{(s)}\|_{C_b^p}=\max_{1\le s\le d}\left\{\sup_{\alpha\in I_p}\left\|D^\alpha f^{(s)}\right\|_\infty\right\}<\infty\right\},$$

where $\|g\|_\infty=\sup_{\boldsymbol{u}\in\mathbb{R}^d}|g(\boldsymbol{u})|$ for any $g:\mathbb{R}^d\to\mathbb{R}$. Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm for vectors, and its induced matrix (spectral) norm if applied to matrices.

4.2 Kernels and induced RKHS

For our Prob-GPararеal error bound analysis, the reproducing kernel Hilbert space (RKHS) structure is especially advantageous. We start by recalling standard facts and some recent results in the literature, which are convenient for our derivations.

Definition 4 (RKHS, Christmann and Steinwart (2008)) *Let $K:\mathcal{U}\times\mathcal{U}\to\mathbb{R}$ be a Mercer (symmetric positive semi-definite) kernel on a nonempty set $\mathcal{U}\subset\mathbb{R}^d$. A Hilbert space $\mathcal{H}_K$ of real-valued functions on $\mathcal{U}$ endowed with the pointwise sum and pointwise scalar multiplication, and with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_K}$ is called a RKHS associated to K if the following properties hold:*

(i) *For all $\boldsymbol{u}\in\mathcal{U}$, the function $K(\boldsymbol{u},\cdot)\in\mathcal{H}_K$.*

(ii) *For all $\boldsymbol{u}\in\mathcal{U}$ and for all $f\in\mathcal{H}_K$, the reproducing property $f(\boldsymbol{u})=\langle f,K(\boldsymbol{u},\cdot)\rangle_{\mathcal{H}_K}$ holds.*

Note that two commonly used Mercer kernels are the Gaussian and the Matérn kernels defined in Appendix A. Given a Mercer kernel K on a set $\mathcal{U}$, by Moore-Aronszajn Theorem (Aronszajn, 1950), there exists a unique Hilbert space of real-valued functions for which K is a reproducing kernel. The RKHS $\mathcal{H}_K$ induced by K is given by

$$\mathcal{H}_K=\left\{f=\sum_{i=1}^{\infty}c_iK\left(\boldsymbol{u}_i,\cdot\right)\mid c_i\in\mathbb{R},\boldsymbol{u}_i\in\mathcal{U},\|f\|_{\mathcal{H}_K}^2=\sum_{i,j=1}^{\infty}c_iK\left(\boldsymbol{u}_i,\boldsymbol{u}_j\right)c_j<\infty\right\},\quad(14)$$

see Christmann and Steinwart (2008) for more details.

4.3 Preparatory assumptions

Here, we introduce the assumptions on the numerical coarse solvers and on the properties of the GP posterior variance, which will be needed to prove the theoretical results of Section 4.4. Following Gander and Hairer (2008), we assume that $h:\mathbb{R}^d\to\mathbb{R}^d$ in (1) is of appropriate regularity, the time steps are uniform with $\Delta t_i=\Delta t:=(t_N-t_0)/N$, and that $\mathscr{F}$ yields an exact solution, that is $\boldsymbol{u}(t_i)=\mathscr{F}(\boldsymbol{u}(t_{i-1}))=\varphi_{\Delta t}(\boldsymbol{u}(t_{i-1}))$, for all $i=1,\dots,N$.

Assumption 5 (Order of the one-step coarse solver $\mathscr{G}$) *$\mathscr{G}$ is a one-step numerical solver with uniform local truncation error $O(\Delta t^{p+1})$ for $p\ge1$. More precisely, for all $\boldsymbol{u}\in\mathbb{R}^d$, it holds that*

$$\mathscr{F}(\boldsymbol{u})-\mathscr{G}(\boldsymbol{u})=c^{(p+1)}(\boldsymbol{u})\Delta t^{p+1}+R_{p+2}(\boldsymbol{u},\Delta t),\quad(15)$$

where $c^{(p+1)}\in C_b^1(\mathbb{R}^d,\mathbb{R}^d)$, and there exist constants $C_R>0$ and $\Delta t_0>0$ such that, for all $0<\Delta t\le\Delta t_0$, $\|R_{p+2}(\cdot,\Delta t)\|_{C_b^1}\le C_R\Delta t^{p+2}$.

Note that, the one-step truncation error in (15) corresponds by definition (5) to the correction function f_c, so Assumption 5 can also be seen as an assumption on f_c rather than on $\mathscr{G}$.

Assumption 6 (The correction function is Lipschitz) *The correction function f_c in (5) is Lipschitz continuous, that is, for all $\boldsymbol{u}, \boldsymbol{u}' \in \mathbb{R}^d$, there exists some constant $L_c > 0$, such that*

$$\|f_c(\boldsymbol{u}) - f_c(\boldsymbol{u}')\| \leq L_c \|\boldsymbol{u} - \boldsymbol{u}'\|. \tag{16}$$

Assumption 7 ($\mathscr{G}$ is Lipschitz) *$\mathscr{G}$ is Lipschitz continuous, that is, for all $\boldsymbol{u}, \boldsymbol{u}' \in \mathbb{R}^d$, there exists some constant $L_{\mathscr{G}} > 0$, such that*

$$\|\mathscr{G}(\boldsymbol{u}) - \mathscr{G}(\boldsymbol{u}')\| \leq L_{\mathscr{G}} \|\boldsymbol{u} - \boldsymbol{u}'\|. \tag{17}$$

Remark 8 *The vector field $h : \mathbb{R}^d \to \mathbb{R}^d$ in (1) must be at least $C_b^{p+2}(\mathbb{R}^d, \mathbb{R}^d)$ for Assumption 5 to hold. If $h \in C_b^{p+2}(\mathbb{R}^d, \mathbb{R}^d)$, the flow is Lipschitz by Grönwall's lemma and so is the exact solver $\mathscr{F}$, meaning that Assumptions 5-7 immediately hold.*

Remark 9 *Under Assumption 5, let $c_1 := \max_{1 \leq s \leq d} \|D(c^{(p+1)})^{(s)}\|_\infty$. Then the correction function f_c is Lipschitz continuous (Assumption 6) with Lipschitz constant satisfying $L_c \leq \sqrt{d}\,(c_1 + C_R\,\Delta t)\,\Delta t^{p+1}$, where C_R is as in Assumption 5. In particular, for sufficiently small Δt, L_c is of order Δt^{p+1}.*

Our final assumption is stated in terms of the fill distance, which we now define.

Definition 10 (Fill distance or covering radius) *Let $\mathcal{D} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\} \subset \mathbb{R}^d$. The global fill distance for $\mathcal{D} \subset \mathcal{U} \subset \mathbb{R}^d$, that is, the largest distance from any point in $\mathcal{U}$ to its nearest point in $\mathcal{D}$, is defined as*

$$h_{\mathcal{U},\mathcal{D}} := \sup_{\boldsymbol{u} \in \mathcal{U}} \min_{\boldsymbol{u}_i \in \mathcal{D}} \|\boldsymbol{u} - \boldsymbol{u}_i\|. \tag{18}$$

For a constant $\rho > 0$, the local fill distance at $\boldsymbol{u}' \in \mathcal{U}$ is defined as

$$h_{\rho,\mathcal{D}}(\boldsymbol{u}') := \sup_{\boldsymbol{u} \in B_\rho(\boldsymbol{u}') \cap \mathcal{U}} \min_{\boldsymbol{u}_i \in \mathcal{D}} \|\boldsymbol{u} - \boldsymbol{u}_i\|, \tag{19}$$

where $B_\rho(\boldsymbol{u}') \subset \mathbb{R}^d$ denotes the ball of radius $\rho > 0$ centered at $\boldsymbol{u}'$.

Assumption 11 (Posterior variance decay) *Let K be a kernel on $\mathbb{R}^d$ and let $\mathcal{H}_K$ be the RKHS induced by it. For $s = 1, \ldots, d$, let the sth coordinate of the correction function $f_c^{(s)} = (\mathscr{F} - \mathscr{G})^{(s)} \in \mathcal{H}_K$, and let $\sigma_{\mathcal{D}}^{(s)}(\boldsymbol{u}')^2 \in \mathbb{R}$ be the posterior variance of a scalar-output GP built on a dataset $\mathcal{D}$ to approximate $f_c^{(s)} \in \mathcal{H}_K$. We assume that there exist some constants $\beta > 0$, $h_0 > 0$, and $\{C_{\beta,s}\}_{s=1}^d$, $C_{\beta,s} > 0$ such that for any $\boldsymbol{u}' \in \mathbb{R}^d$ and any set of points $\mathcal{D} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\} \subset \mathbb{R}^d$ satisfying $h_{\rho,\mathcal{D}}(\boldsymbol{u}') \leq h_0$, the GP regression posterior variance satisfies $\sigma_{\mathcal{D}}^{(s)}(\boldsymbol{u}')^2 \leq C_{\beta,s} h_{\rho,\mathcal{D}}(\boldsymbol{u}')^\beta$.*

Additionally, it is convenient to introduce the following definition.

Definition 12 (Maximum norm for vector-valued functions) *Let $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ be a normed vector space of scalar-valued functions. For a vector-valued function f with the sth coordinate $f^{(s)} \in \mathcal{H}$ for all $s = 1, \dots, d$, define the maximum norm of f induced by $\|\cdot\|_{\mathcal{H}}$ as $\|f\|_{\infty,\mathcal{H}} := \max_{1\le s\le d} \|f^{(s)}\|_{\mathcal{H}}$.*

All assumptions are stated globally on $\mathbb{R}^d$ for notational simplicity, but may need to be imposed, instead, locally on a compact convex set $\Omega \subset \mathbb{R}^d$ with nonempty interior chosen large enough to contain the true trajectory $\{\boldsymbol{u}(t_i)\}$, the GPararead trajectory $\{\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\}$, and the Prob-GPararead means $\{\boldsymbol{\mu}_{i,k} = \mathbb{E}[\boldsymbol{U}_{i,k}]\}$. The constants appearing in the bounds are then interpreted as local constants on Ω, with the posterior variance and fill-distance assumptions imposed on a fixed neighborhood $\Omega_\rho := \{\boldsymbol{v} \in \mathbb{R}^d \mid \exists \boldsymbol{u} \in \Omega \text{ such that } \|\boldsymbol{v} - \boldsymbol{u}\| \le \rho\}$ of Ω, where $\rho > 0$ and one takes $\mathcal{U} = \Omega_\rho$ in Definition 10. Note that for every evaluation point $\boldsymbol{u}' \in \Omega$ appearing in the bounds, the ball $B_\rho(\boldsymbol{u}')$ entering the local fill distance is contained in Ω_ρ. The condition $f_c^{(s)} \in \mathcal{H}_K$ is then understood locally, by requiring the restrictions $f_c^{(s)}|_{\Omega_\rho}$ to admit extensions belonging to the corresponding RKHS on $\mathbb{R}^d$.

4.4 Prob-GPararead error bounds

We are now ready to study the behavior of the probabilistic Prob-GPararead solution. Consider $W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2$, the squared Wasserstein-2 distance between $\boldsymbol{u}(t_i)$, the true solution of (1) at time t_i, and $P_{\boldsymbol{U}_{i,k}}$, the distribution of the Prob-GPararead solution at interval i and iteration k. This quantity can be bounded as a function of the expected local fill distance, as stated below.

Theorem 13 (Prob-GPararead error bound) *Let $P_{\boldsymbol{U}_{0,k}}$ be an initial distribution satisfying $\mathbb{E}[\boldsymbol{U}_{0,k}] = \boldsymbol{u}_{0,0} = \boldsymbol{u}_{(0)}$ and $\mathrm{Var}(\boldsymbol{U}_{0,k}) = \Sigma_{0,k}$, $P_{\boldsymbol{U}_{i,k}}$ be the distribution of the Prob-GPararead solution at interval $i \in \{1, \dots, N\}$ and iteration $k \in \mathbb{N}$, and $\delta_{\boldsymbol{u}(t_i)}$ be a Dirac measure centered at $\boldsymbol{u}(t_i)$, the true solution of the system (1) at time t_i. Let K be a kernel on $\mathbb{R}^d$ and let $\mathcal{H}_K$ be the RKHS induced by this kernel. For $s = 1, \dots, d$, let the sth coordinate of the correction function $f_c^{(s)} = (\mathscr{F} - \mathscr{G})^{(s)} \in \mathcal{H}_K$, and let ${\sigma_{\mathcal{D}_k}^{(s)}}^2$, be the posterior variance function of a scalar-output GP built on a dataset $\mathcal{D}_k$ to approximate $f_c^{(s)} \in \mathcal{H}_K$. Further, assume that one of the following holds:*

- **(i) Differentiability:** *$K \in C_b^{2p+1}(\mathbb{R}^d \times \mathbb{R}^d)$, $p \ge 1$, and Assumptions 5, 7, and 11 (with some $\beta > 0$) hold.*
- **(ii) Sobolev norm-equivalence:** *K is such that its induced RKHS $\mathcal{H}_K$ is norm-equivalent[2] to $(W_2^q(\mathbb{R}^d), \|\cdot\|_{W_2^q})$, the Sobolev space of order $q > d/2$, and Assumptions 5 and 7 hold.*
- **(iii) Smoothness:** *K is infinitely smooth and Assumptions 5 and 7 hold.*

Additionally, define

$$a := 4\left(L_{\mathscr{G}}^2 + L_c^2\right). \tag{20}$$

2. Vector spaces $(\mathcal{H}_1, \|\cdot\|_{\mathcal{H}_1})$ and $(\mathcal{H}_2, \|\cdot\|_{\mathcal{H}_2})$ are called norm-equivalent, if $\mathcal{H}_1 = \mathcal{H}_2$ as a set, and if there are constants $c_1, c_2 > 0$ such that $c_1\|f\|_{\mathcal{H}_2} \le \|f\|_{\mathcal{H}_1} \le c_2\|f\|_{\mathcal{H}_2}$ holds for all $f \in \mathcal{H}_1 = \mathcal{H}_2$.

Then, for all $i = 1, \dots, N$, and $k \in \mathbb{N}$, it holds that

$$W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2 = \mathbb{E}\left[\|\boldsymbol{u}(t_i) - \boldsymbol{U}_{i,k}\|^2\right], \tag{21}$$

i.e., the W_2^2 distance equals the mean squared error (MSE) of $\boldsymbol{U}_{i,k}$, and

$$W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2 \leq a^i \operatorname{tr}(\Sigma_{0,k}) + \sum_{j=1}^{i} a^{i-j} b_{j-1,k}, \tag{22}$$

where $b_{l,k}$, $l = 0, \dots, N-1$, is defined separately for each of the cases **(i)**-**(iii)** *as follows:*

(i) Differentiability:

$$b_{l,k} = b_{l,k}(\beta) := 4d\, C_\beta (1 + \|f_c\|^2_{\infty,\mathcal{H}_K}) \mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^\beta\right],$$

with $C_\beta = \max_{1\leq s\leq d} C_{\beta,s}$ and $\{C_{\beta,s}\}_{s=1}^d$, $C_{\beta,s} > 0$, as in Assumption 11.

(ii) Sobolev norm-equivalence:

$$b_{l,k} = 4d\, C (1 + \|f_c\|^2_{\infty,W_2^q}) \mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^{2q-d}\right],$$

with $C = \max_{1\leq s\leq d} C_s$ and $\{C_s\}_{s=1}^d$, $C_s > 0$, defined in Theorem 5.4 in Kanagawa et al. (2018) (reported in part **(i)** *in Theorem 25 in Appendix B.1 for convenience).*

(iii) Smoothness:

$$b_{l,k} = b_{l,k}(\alpha) := 4d\, C_\alpha (1 + \|f_c\|^2_{\infty,\mathcal{H}_K}) \mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^\alpha\right],$$

with $C_\alpha = \max_{1\leq s\leq d} C_{\alpha,s}$ and $\{C_{\alpha,s}\}_{s=1}^d$, $C_{\alpha,s} > 0$, defined in Theorem 5.14 in Wu and Schaback (1993) (reported in part **(ii)** *in Theorem 25 in Appendix B.1 for convenience).*

In all cases, $h_{\rho,\mathcal{D}_k}$ is the local fill distance defined in (19).

Remark 14 *In* **(ii)**, *whenever $q > d/2 + 1$, the Sobolev embedding $W_2^q(\mathbb{R}^d) \hookrightarrow C_b^1(\mathbb{R}^d)$ holds. Since $f_c^{(s)} \in \mathcal{H}_K \simeq W_2^q(\mathbb{R}^d)$ for each $s = 1, \dots, d$, it follows that $f_c^{(s)} \in C_b^1(\mathbb{R}^d, \mathbb{R})$ for each coordinate. Hence f_c is globally Lipschitz, Assumption 6 is automatically satisfied with $L_c \leq \sqrt{d} \max_{1\leq s\leq d} \|Df_c^{(s)}\|_\infty$, and Assumption 5 is not needed to obtain Lipschitz continuity of f_c.*

Remark 15 *Assumption 5 is listed in cases* **(i)**-**(iii)** *to obtain Lipschitz continuity of f_c via Remark 9 and could be replaced by Assumption 6. However, Assumption 5 additionally yields the explicit rate $L_c \leq \sqrt{d}\,(c_1 + C_R\,\Delta t)\,\Delta t^{p+1}$, which connects the bound via* (20) *to the order p of the coarse solver $\mathscr{G}$.*

Remark 16 (RKHS membership and kernel choice) *The condition $f_c^{(s)} \in \mathcal{H}_K$ is used to construct the error bounds and is not required to run Prob-GParareal. Assumption 5 implies global Lipschitz continuity of f_c (see Remark 9), but not RKHS membership, which requires additional regularity of h and of the one-step solvers. For Matérn-ν kernels, membership*

reduces to $f_c^{(s)} \in W_2^{\nu+d/2}(\Omega)$ *on the relevant compact region* Ω *of the state space, and can be verified through standard regularity assumptions on* h *and on the one-step solvers. For Gaussian kernels, the condition is strictly stronger than analyticity, requiring Gaussian-weighted Fourier integrability of* $f_c^{(s)}$*. Hence, the Gaussian kernel should be understood as a practical high-regularity modeling choice. In either case, kernel hyperparameters can be selected by marginal likelihood optimization and assessed empirically through the observed decay of the posterior variance.*

The proof of Theorem 13 is provided in Appendix B.2.

Corollary 17 (Uniform Prob-GPararеal error bound) *Under the same Assumptions of Theorem 13, fix an iteration* $k \geq 1$ *and suppose there exists a constant* $B_k \geq 0$ *such that* $b_{l,k} \leq B_k$*, for all* $l = 0, \dots, N-1$*. Then, it holds that*

$$W_2\big(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}}\big)^2 \;\leq\; a^i \operatorname{tr}(\Sigma_{0,k}) + B_k \,\frac{1-a^i}{1-a}, \tag{23}$$

where a *is defined in* (20) *and satisfies* $a \neq 1$*.*

Remark 18 *The bound* (22) *depends on the initial condition variance,* $\Sigma_{0,k}$*, with* $\Sigma_{0,k}$ *equal to the zero matrix when the initial condition* $\boldsymbol{u}_{(0)}$ *is deterministic (*$P_{\boldsymbol{U}_{0,k}} = \delta_{\boldsymbol{u}_{0,k}}$*), and on the expected local fill distance* $h_{\rho,\mathcal{D}_k}$ *through the coefficients* $b_{l,k}$*. Similar fill-distance bounds appear in the deterministic (nn)GPararеal literature (Pentland et al., 2023b; Gattiglio et al., 2025). In the contractive regime, i.e.* $a < 1$*, the first term decays exponentially in* i *and* $W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2$ *is controlled by a geometric sum. When instead* $a > 1$*, the worst case bound would grow exponentially in* i*. However, empirical evidence shows that the coefficients* $b_{l,k}$ *decay exponentially with the iteration number* k*. This behavior is driven by the progressive refinement of the Prob-GPararеal solution, which increases the informativeness of the training dataset* $\mathcal{D}_k$ *and induces a corresponding decay in the expected local fill distance* $h_{\rho,\mathcal{D}_k}$*. Although the theoretical analysis of this decay as a function of* k *is challenging, Appendix D presents empirical evidence of its exponential decay, mitigating the exponential growth term. An empirical assessment of the sharpness of the bounds in Theorem 13 and Corollary 17 is provided in Appendix H for the deterministic initial condition case, where we compare the Wasserstein error with the corresponding fill-distance terms in the coefficients* $b_{i,k}$*.*

After having bounded the Prob-GPararеal error with respect to the true solution, we now draw a connection between the Prob-GPararеal solution (in particular, its mean) and its deterministic counterpart, the GPararеal solution $\boldsymbol{u}_{i,k}^{\mathrm{GPara}}$ obtained by sequential applications of $\mathscr{G}$ corrected by the GP posterior mean $\widehat{f}_{\mathrm{GPara}}(\cdot) = \boldsymbol{\mu}_{\mathcal{D}_k}(\cdot)$ as in (8), namely

$$\boldsymbol{u}_{i,k}^{\mathrm{GPara}} := \underbrace{\big((\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k}) \circ (\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k}) \circ \cdots \circ (\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})\big)}_{i \text{ times}} (\boldsymbol{u}_{0,k}^{\mathrm{GPara}}), \quad i = 1, \dots, N, \;\; k \in \mathbb{N},$$

with $\boldsymbol{u}_{0,k}^{\mathrm{GPara}} = \boldsymbol{u}_{(0)}$. In particular, in Theorem 21, we derive an error bound between the Prob-GPararеal mean and the GPararеal solution as a function of the maximum variance of $\boldsymbol{U}_{i,k}$ over the d dimensions. While such variance

is generally unknown, we provide an explicit bound in the following Theorem 19.

Theorem 19 (Variance bound for $\boldsymbol{U}_{i,k}$) *Let the conditions of Theorem 13 hold. Let $\boldsymbol{U}_{i,k}$ be the Prob-GParareal solution at the ith interval, $i \in \{1, \dots, N\}$, and kth iteration, $k \in \mathbb{N}$, with mean $\boldsymbol{\mu}_{i,k}^{(s)} = \mathbb{E}[\boldsymbol{U}_{i,k}^{(s)}]$ and variance $\sigma_{i,k}^{(s),2} = \mathrm{Var}(\boldsymbol{U}_{i,k}^{(s)})$ of its sth coordinate, $s = 1, \dots, d$. Denote the maximum variance of $\boldsymbol{U}_{i,k}$ as*

$$\sigma_{i,k}^{\max,2} := \max_{1 \le s \le d} \sigma_{i,k}^{(s),2}, \quad i = 1, \dots, N, \;\; k \in \mathbb{N}. \tag{24}$$

Assume that one of the **(i)**-**(iii)** *cases of Theorem 13 holds, and define*

$$a := 2\, d(L_{\mathscr{G}}^2 + 3L_c^2). \tag{25}$$

Then, for all $i = 1, \dots, N$, and $k \in \mathbb{N}$, it holds that

$$\sigma_{i,k}^{\max,2} \le a^i\, \sigma_{0,k}^{\max,2} + \sum_{j=1}^{i} a^{\,i-j}\, b_{j-1,k}, \tag{26}$$

where $b_{l,k}$, $l = 0, \dots, N-1$, is defined separately for each of the cases **(i)**-**(iii)** *in Theorem 13 as follows:*

(i) Differentiability:

$$b_{l,k} = C_\beta \mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^\beta\right] + 6C_\beta \|f_c\|_{\infty,\mathcal{H}_K}^2 \left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^\beta\right] + h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^\beta\right\},$$

with C_β defined as in part **(i)** *of Theorem 13.*

(ii) Sobolev norm-equivalence:

$$b_{l,k} = C\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^{2q-d}\right] + 6C\|f_c\|_{\infty,W_2^q}^2 \left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^{2q-d}\right] + h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^{2q-d}\right\}$$

with C defined as in part **(ii)** *of Theorem 13.*

(iii) Smoothness:

$$b_{l,k} = C_\alpha \mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^\alpha\right] + 6C_\alpha \|f_c\|_{\infty,\mathcal{H}_K}^2 \left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{l,k})^\alpha\right] + h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^\alpha\right\},$$

with C_α defined as in part **(iii)** *of Theorem 13.*

In addition, Remark 9 holds under the assumptions of this theorem.

The proof of this theorem is provided in Appendix B.3.

Corollary 20 (Uniform variance bound for $\boldsymbol{U}_{i,k}$) *Under the same assumptions as in Theorem 19, fix an iteration $k \in \mathbb{N}$, and suppose there exists a constant $B_k \ge 0$ such that $b_{l,k} \le B_k$ for all $l = 0, \dots, N-1$. Then, it holds that*

$$\sigma_{i,k}^{\max,2} \le B_k \frac{1 - a^i}{1 - a} + a^i \sigma_{0,k}^{\max,2}, \quad i = 1, \dots, N, \;\; k \in \mathbb{N}, \tag{27}$$

where a is defined in (25) *and satisfies $a \neq 1$.*

After having bounded $\sigma_{i,k}^{\max,2}$, we now present a bound on the difference between the mean of the Prob-GPareareal solution and its deterministic counterpart, the GPareareal solution, as a function of such quantity. An error bound between the mean of Prob-GPareareal and the true solution will then follow.

Theorem 21 (Mean error bound with respect to GPareareal) *Let $\sigma_{i,k}^{\max,2}$ be the maximum coordinate-wise variance of the Prob-GPareareal solution $\boldsymbol{U}_{i,k}$ defined as in (24) and let $\boldsymbol{\mu}_{\mathcal{D}_k}$ be the GP posterior mean computed on the dataset $\mathcal{D}_k$ at interval $i \in \{1,\dots,N\}$ and iteration $k \in \mathbb{N}$. Let $(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k}) \in C^2(\mathbb{R}^d, \mathbb{R}^d)$ and $H_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}}(\boldsymbol{U})$ be the Hessian matrix of $(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}$, $s = 1,\dots,d$, evaluated at $\boldsymbol{U} \in \mathbb{R}^d$. Assume there exists a constant $M_s > 0$ such that $\|H_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}}(\boldsymbol{U})\| \le M_s$ for all $\boldsymbol{U} \in \mathbb{R}^d$. Define $M := \max_{1\le s\le d} M_s$. Let either of the cases* **(i)**, **(ii)**, *or* **(iii)** *of Theorem 13 holds, and define*

$$a := L_{\mathscr{G}} + L_c. \tag{28}$$

Then, for all $i = 1,\dots,N$, and $k \in \mathbb{N}$ it holds that

$$\|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| \le \sum_{j=1}^{i} a^{i-j} b_{j-1,k}, \tag{29}$$

where $b_{l,k}$, $l = 0,\dots,N-1$, is defined separately for each of the cases **(i)**-**(iii)** *in Theorem 13 as follows:*

(i) Differentiability:

$$b_{l,k} = \frac{M\, d^{3/2}}{2}\, \sigma_{l,k}^{\max,2} + \sqrt{C_\beta d}\|f_c\|_{\infty,\mathcal{H}_K} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^{\beta/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{l,k}^{\mathrm{GPara}})^{\beta/2} \right)$$

with C_β defined as in part **(i)** *of Theorem 13 for $\beta > 0$.*

(ii) Sobolev norm-equivalence:

$$b_{l,k} = \frac{M\, d^{3/2}}{2}\, \sigma_{l,k}^{\max,2} + \sqrt{Cd}\|f_c\|_{\infty,W_2^q} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^{q-d/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{l,k}^{\mathrm{GPara}})^{q-d/2} \right).$$

with C defined as in part **(ii)** *of Theorem 13.*

(iii) Smoothness:

$$b_{l,k} = \frac{M\, d^{3/2}}{2}\, \sigma_{l,k}^{\max,2} + \sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{l,k}^{\mathrm{GPara}})^{\alpha/2} \right),$$

with C_α defined as in part **(iii)** *of Theorem 13 for $\alpha > 0$.*

In addition, Remark 9 holds under the assumptions of this theorem.

The proof of this theorem is provided in Appendix B.4. Following the same approach as in Corollary 17, we simplify (29) by assuming the existence of a constant $B_k \ge 0$ such that $b_{l,k} \le B_k$; this is presented in Corollary 22 below. Alternatively, $\sigma_{l,k}^{\max,2}$ in $b_{l,k}$ can be replaced by the variance bound established in Theorem 19, allowing (29) to be expressed in terms of the variance of the initial condition, $\sigma_{0,k}^{\max,2}$, which is typically known.

Corollary 22 (Uniform mean error bound) *Under the same assumptions as in Theorem 21, fix an iteration $k \geq 1$ and suppose there exists a constant $B_k \geq 0$ such that $b_{l,k} \leq B_k$, for all $l = 0, \ldots, N-1$. Then, it holds that*

$$\left\|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\right\| \leq B_k \frac{1-a^i}{1-a}, \quad i = 1, \ldots, N, \;\; k \in \mathbb{N}, \tag{30}$$

where a is defined in (28) *and satisfies $a \neq 1$.*

Corollary 23 (Explicit mean-error bound) *Under the assumptions of Theorem 19 and Theorem 21, let*

$$a \;=\; 2\,d\,(L_{\mathscr{G}}^2 + 3L_c^2), \quad \tilde{a} \;=\; L_{\mathscr{G}} + L_c,$$

and for each of the cases **(i)**-**(iii)**, *let $b_{l,k}$, $l = 0, \ldots, N-1$, be as in Theorem 19. If $a \neq \tilde{a}$, then for every $i = 1, \ldots, N$ and $k \in \mathbb{N}$, the result of Theorem 21 simplifies to*

$$\|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| \leq \frac{Md^{3/2}}{2(\tilde{a}-a)} \left(\sigma_{0,k}^{\max,2} \;\; (\tilde{a}^i - a^i) + \sum_{j=1}^{i-1} b_{j-1,k} (\tilde{a}^{\,i-j} - a^{\,i-j}) \right) + r_{i,k}.$$

If $a = \tilde{a}$, the same bound holds with the first term replaced by its limiting value, namely

$$\|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| \leq \frac{Md^{3/2}}{2} \left(\sigma_{0,k}^{\max,2}\, i a^{i-1} + \sum_{j=1}^{i-1} b_{j-1,k} (i-j) a^{i-j-1} \right) + r_{i,k}.$$

In these bounds $r_{i,k}$ is defined separately for each of the cases **(i)**-**(iii)** *in Theorem 13 as follows:*

(i) Differentiability:

$$r_{i,k} = \sqrt{C_\beta d}\, \|f_c\|_{\infty,\mathcal{H}_K} \sum_{j=1}^{i} \tilde{a}^{\,i-j} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{j-1,k})^{\beta/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{j-1,k}^{\mathrm{GPara}})^{\beta/2} \right),$$

with C_β defined as in part **(i)** *of Theorem 13 for $\beta > 0$.*

(ii) Sobolev norm-equivalence:

$$r_{i,k} = \sqrt{Cd}\, \|f_c\|_{\infty,W_2^q} \sum_{j=1}^{i} \tilde{a}^{\,i-j} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{j-1,k})^{q-d/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{j-1,k}^{\mathrm{GPara}})^{q-d/2} \right),$$

with C defined as in part **(ii)** *of Theorem 13.*

(iii) Smoothness:

$$r_{i,k} = \sqrt{C_\alpha d}\, \|f_c\|_{\infty,\mathcal{H}_K} \sum_{j=1}^{i} \tilde{a}^{\,i-j} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{j-1,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{j-1,k}^{\mathrm{GPara}})^{\alpha/2} \right),$$

with C_α defined as in part **(iii)** *of Theorem 13 for $\alpha > 0$.*

Proof is provided in Appendix B.4.

Remark 24 *We can bound the error between the mean $\boldsymbol{\mu}_{i,k}$ of the Prob-GPareareal solution $\boldsymbol{U}_{i,k}$ and the true solution $\boldsymbol{u}(t_i)$, that is the bias of $\boldsymbol{U}_{i,k}$, using the triangular inequality*

$$\|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}(t_i)\| \leq \|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\text{GPara}}\| + \|\boldsymbol{u}_{i,k}^{\text{GPara}} - \boldsymbol{u}(t_i)\|, \tag{31}$$

where the first term is bounded by Theorem 21 or Corollaries 22 and 23, and the second term by Theorem 3.4 in Pentland et al. (2023b). Combining (31) *(squared) with the variance bound in Theorem 19 gives an upper bound of the MSE of $\boldsymbol{U}_{i,k}$, or, equivalently, by* (21) *the upper bound of distance W_2^2.*

5. Empirical results for Prob-GPareareal

In this section, we illustrate Prob-GPareareal and its performance on five different ODE systems with deterministic initial conditions (except in Section 5.4, with random initial conditions), four of which were considered for GPareareal (Pentland et al., 2023b), allowing for a direct comparison. These include the FitzHugh-Nagumo (FHN) model, a two-dimensional representation of an animal nerve axon (Nagumo et al., 1962); the Rössler system, a three-dimensional chaotic model for turbulence (Rössler, 1976); the nonlinear Hopf bifurcation model, a two-dimensional non-autonomous ODE (Seydel, 2009); and the double pendulum, a four-dimensional system describing the dynamics of two connected pendula (Pettet, 1999). The fifth added model is the Lorenz system, a commonly studied chaotic system used in simplified weather modeling (Lorenz, 1963). Overall, these systems offer a wide range of complexities, from stiff to non-autonomous and chaotic dynamics, to assess the performance of Prob-GPareareal. For a detailed description of the models, we refer to Pentland et al. (2023b), Gattiglio et al. (2025), and the original references. A summary of the simulation setup, including the coarse and fine solvers, initial conditions, evolution timespans, and other relevant parameters required to replicate the results is reported in Table 3 in Appendix G. Finally, a comparison with the parallel-in-time probabilistic ODE solver of Bosch et al. (2024) is reported in Appendix J. As previously done in Gattiglio et al. (2025, 2024), we rescale each ODE by a change of variables such that each coordinate lies within $[-1, 1]$. This normalization step does not alter the system's properties, helps stabilize the GP learning, and facilitates comparisons across systems with different scales.

5.1 Prob-GPareareal accuracy and probabilistic forecast

Prob-GPareareal produces a *probabilistic* forecast for the evolution of a DE system - a sequence of predicted distributions $P_{\boldsymbol{U}_{i,K_{\text{end}}}}$ for the target $\boldsymbol{u}(t_i)$, $i = 1, \dots, N$, - as opposed to the single-point forecast of classical deterministic DE solvers. These distributions are then approximated using their empirical counterparts $\widehat{P}_{\boldsymbol{U}_{i,K_{\text{end}}}}$, based on the n drawn samples in $\mathcal{U}_{i,K_{\text{end}}}$, where, K_{end} denotes the final Prob-GPareareal iteration, when the algorithm either converges (so $K_{\text{end}} = K_{\text{conv}}$) or is stopped due to early termination (so $K_{\text{end}} = K_{\text{stop}}$).

Evaluating probabilistic forecasts requires metrics that capture the similarity of the predicted distribution with empirical data in terms of both *calibration* and *sharpness* (Gneiting and Raftery, 2007). Calibration refers to the statistical consistency between predicted probabilities and observed event frequencies, indicating the reliability of a forecast. Sharpness

measures the concentration of the predictive distribution, with sharper (more concentrated) distributions being preferable when well-calibrated. Proper scoring rules, extensively studied in the literature (Gneiting and Raftery, 2007; Bröcker and Smith, 2007; Gneiting and Katzfuss, 2014) and widely adopted across disciplines (Galbraith and van Norden, 2012; Dumas et al., 2022; Lerch et al., 2017; Bogner et al., 2016), are a natural choice for evaluating probabilistic forecasts, as they balance sharpness and calibration. Comparative studies (Pic et al., 2025; Alexander et al., 2024; Ziel and Berk, 2019) emphasize the importance of employing multiple scoring rules, as each metric highlights distinct distributional characteristics. For instance, for $i = 1, \dots, N$, $k \geq 1$, and $s = 1, \dots, d$, the energy score (ES) (Gneiting et al., 2007), defined as

$$\mathrm{ES}(\mathcal{U}_{i,k}, \boldsymbol{u}(t_i)) = \frac{1}{n}\sum_{j=1}^{n} \|\boldsymbol{u}_{i,k}^{(j)} - \boldsymbol{u}(t_i)\| - \frac{1}{2n^2}\sum_{j=1}^{n}\sum_{l=1}^{n} \|\boldsymbol{u}_{i,k}^{(j)} - \boldsymbol{u}_{i,k}^{(l)}\|,$$

prioritizes mean accuracy over capturing covariance structure, while the variogram score (VS) (Scheuerer and Hamill, 2015)

$$\mathrm{VS}(\mathcal{U}_{i,k}, \boldsymbol{u}(t_i)) = \sum_{s_1,s_2=1}^{d} w_{s_1,s_2} \left(\frac{1}{n}\sum_{j=1}^{n} \left| \boldsymbol{u}_{i,k}^{(j)(s_1)} - \boldsymbol{u}_{i,k}^{(j)(s_2)} \right|^p - \left| \boldsymbol{u}(t_i)^{(s_1)} - \boldsymbol{u}(t_i)^{(s_2)} \right|^p \right)^2$$

addresses dependence structures more effectively. Here, $\boldsymbol{u}_{i,k}^{(j)(s)}$ denotes the sth coordinate of the jth forecast sample, $\boldsymbol{u}(t_i)^{(s)}$ the sth coordinate of the observed solution at time t_i, and $w_{s_1,s_2} \geq 0$ are user-specified weights that determine the contribution of each coordinate pair. In our setting, we take $w_{s_1,s_2} = 1$ for all pairs. Following common practice (Alexander et al., 2024), we set $p = 0.5$, which has been shown to discriminate between forecasts of different dependence structures effectively. Henceforth, we omit the subscript p for simplicity. To capture the temporal performance of probabilistic forecasts, both ES and VS are averaged over time intervals as

$$\mathrm{ES}_k = \frac{1}{N}\sum_{i=1}^{N} \mathrm{ES}(\mathcal{U}_{i,k}, \boldsymbol{u}(t_i)) \qquad \text{and} \qquad \mathrm{VS}_k = \frac{1}{N}\sum_{i=1}^{N} \mathrm{VS}(\mathcal{U}_{i,k}, \boldsymbol{u}(t_i)).$$

Although ES and VS quantify forecast quality, their interpretation can be challenging. Therefore, we additionally employ the mean absolute deviation (MAD) of the observed solution from forecasted confidence intervals (CIs) (Winkler, 1972), defined as:

$$\mathrm{MAD}_k = \frac{1}{N}\sum_{i=1}^{N} \mathrm{MAD}(\mathcal{U}_{i,k}, \boldsymbol{u}(t_i))$$

with

$$\mathrm{MAD}(\mathcal{U}_{i,k}, \boldsymbol{u}(t_i)) = \frac{1}{d}\sum_{s=1}^{d} \left(\max\left(\boldsymbol{u}(t_i)^{(s)} - \boldsymbol{B}_{i,k}^{(s)}, 0\right) + \max\left(\boldsymbol{A}_{i,k}^{(s)} - \boldsymbol{u}(t_i)^{(s)}, 0\right) \right),$$

where $\boldsymbol{A}_{i,k}^{(s)}$ and $\boldsymbol{B}_{i,k}^{(s)}$ represent the lower and upper bounds, respectively, of a one-dimensional two-sided symmetric 95% CI of the sth coordinate, thus neglecting the dependence structure.

System	$\text{MAD}_{K_{\text{conv}}}$	$\text{VS}_{K_{\text{conv}}}$	$\text{ES}_{K_{\text{conv}}}$	$\text{MSE}_{K_{\text{conv}}}$
FHN	0 (0)	$2.9e^{-13}$ ($1.5e^{-13}$)	$7.9e^{-8}$ ($6.6e^{-9}$)	$2.1e^{-13}$ ($4.0e^{-14}$)
Hopf	$2.8e^{-7}$ ($3.9e^{-8}$)	$6.9e^{-9}$ ($8.5e^{-9}$)	$2.3e^{-5}$ ($9.2e^{-6}$)	$9.1e^{-11}$ ($4.8e^{-11}$)
Dbl Pend.	$3.9e^{-7}$ ($4.6e^{-7}$)	$2.3e^{-8}$ ($2.2e^{-8}$)	$1.2e^{-5}$ ($7.3e^{-6}$)	$1.3e^{-11}$ ($5.0e^{-12}$)
Lorenz	0 (0)	$2.9e^{-6}$ ($4.6e^{-6}$)	$2.1e^{-4}$ ($4.6e^{-5}$)	$3.1e^{-11}$ ($1.5e^{-12}$)
Rössler	0 (0)	$7.0e^{-8}$ ($2.6e^{-8}$)	$2.7e^{-5}$ ($2.7e^{-6}$)	$3.7e^{-11}$ ($1.4e^{-12}$)

System	$T_{\text{Prob-GPara}}$	T_{GPara}	$K_{\text{Prob-GPara}}$	K_{GPara}
FHN	6s (0.3s)	5s	5 (0)	5
Hopf	21s (1.0s)	22s	9 (0)	10
Dbl Pend.	30s (1.4s)	29s	7.5 (0.5)	10
Lorenz	80s (4.1s)	46s	13.6 (0.67)	12
Rössler	41s (0.8s)	37s	6 (0)	7

Table 1: Evaluation of the Prob-GParareal probabilistic forecast across various DEs. The reported metrics are averaged over ten independent runs of the simulations (launched with a different random seed to account for algorithmic randomness), with standard deviation in parentheses. $T_\cdot$ and $K_\cdot$ denote the runtime and iterations to converge, respectively, for Prob-GParareal (Prob-GPara) and GParareal (GPara). For Prob-GParareal (GParareal) an accuracy threshold of $\epsilon_{\text{Prob-GPara}} = 1e^{-7}$ ($\epsilon_{\text{GPara}} = 5e^{-6}$) is used for all systems except Lorenz, which required a higher precision, so $\epsilon_{\text{Prob-GPara}} = 1e^{-9}$ ($\epsilon_{\text{GPara}} = 5e^{-9}$) was taken. The number of samples is $n = 5000$ for all systems.

Although MAD captures forecast calibration in a straightforward manner, it does not explicitly consider sharpness, and loses distributional information by summarizing forecasts with CIs. Nonetheless, it serves as a baseline measure complementary to ES and VS, offering more intuitive interpretability.

Finally, given the equality between $W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2$ and the MSE of $\boldsymbol{U}_{i,k}$ shown in Theorem 13, we include the MSE as evaluation metric alongside those presented above. We use a similar notation where $\text{MSE}_{i,k}$ denotes the MSE of $\boldsymbol{U}_{i,k}$ computed using the intermediate solution at iteration k (e.g. $\mathcal{U}_{i,k}$ for Prob-GParareal), and MSE_k is its average over the intervals. Moreover, in Figure 1 and in Table 2, we additionally show the bias, which we compute as $\text{Bias}_{i,k} = \|\bar{\boldsymbol{u}}_{i,k} - \boldsymbol{u}(t_i)\|$ where $\bar{\boldsymbol{u}}_{i,k}$ is the sample mean of $\mathcal{U}_{i,k}$ (i.e. the sample estimate of $\mathbb{E}[\boldsymbol{U}_{i,k}]$). Note that the MSE and the bias squared coincide for all algorithms but Prob-(nn)GParareal, as they yield a deterministic solution.

5.1.1 Performance of Prob-GParareal

Table 1 summarizes the performance of Prob-GParareal across various ODE systems evaluated using the metrics described in Section 5.1, and reports the runtimes T and the number of iterations to converge K_{conv} of Prob-GParareal and GParareal for comparison. As reported in the acknowledgments, all simulations were executed on Warwick University High Performance Computing (HPC) facilities.

The results demonstrate a good calibration and sharpness for all systems, with some performance reduction observed for Lorenz due to the chaotic nature of the system, as further discussed in Section 5.2. Since the metrics in Table 1 are averaged over the entire time

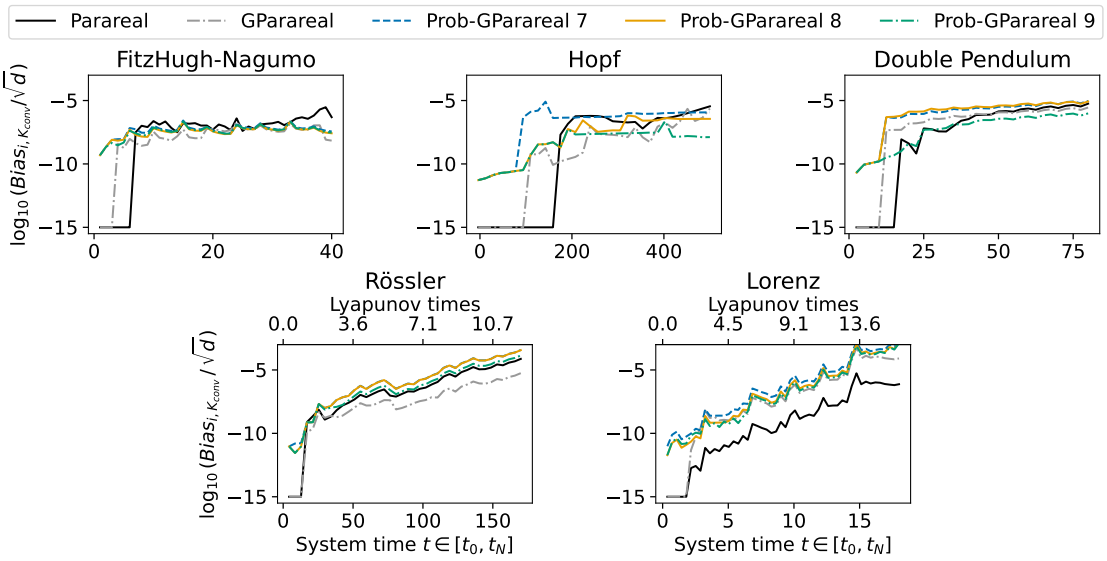

Figure 1: Performance comparison, evaluated using the normalised bias $\text{Bias}_{i,K_{\text{conv}}}/\sqrt{d}$ (in $\log_{10}$), of the Parareal, GParareal and Prob-GParareal solutions with respect to the solution of the fine solver $\mathscr{F}$. Prob-GParareal is run using different values of ϵ, with "Prob-GParareal l" in the legend referring to $\epsilon = 1e^{-l}$, $l = 7, 8, 9$. The Lyapunov times (for Rössler and Lorenz) are computed as the ratio between the system time and the Lyapunov time, as described in Section 5.2.

interval, they do not reveal the properties of the Prob-GParareal solution at individual time t_i. To this end, in Figure 1, we show the evolution of $\text{Bias}_{i,k}$ normalized by $\sqrt{d}$, where d is the dimension of the ODE, between the point solution provided by the fine solver $\mathscr{F}$, and that of Parareal, GParareal and Prob-GParareal. The normalization of the biases allows for a direct comparison of these metrics across models. Since finding the value of the Prob-GParareal threshold ϵ that matches those of Parareal and GParareal is analytically challenging and system-dependent, particularly due to the different interpretations of the stopping thresholds ϵ in (4) and (13), we report in Figure 1 the normalized bias for a range of values of the Prob-GParareal ϵ values. This shows that improved accuracy may be obtained by lowering such threshold. Further results on the impact of the threshold ϵ on the solution are given in Appendix C. When looking at the normalized bias, we see that the mean accuracy of Prob-GParareal is comparable to that of the other deterministic algorithms, with larger values for Lorenz and Rössler, the two chaotic systems.

To better understand the different behavior across systems and characterize the Prob-GParareal solutions, in Figure 2, we display the evolution of $\sigma_{i,k}^{(s)}$, the empirical coordinate-wise standard deviation of $\mathcal{U}_{i,K_{\text{conv}}}$, across system time t upon convergence. Note that Prob-GParareal shows steadily increasing uncertainty over time for the two chaotic systems, Lorenz

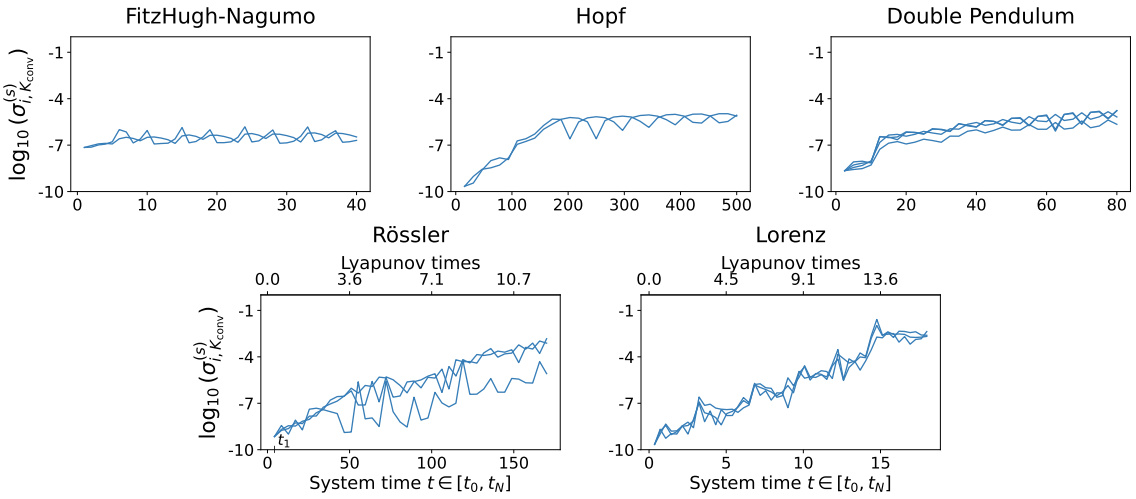

Figure 2: Evolution of the coordinate-wise standard deviation $\sigma^{(s)}_{i,K_{\text{conv}}}$ (in $\log_{10}$), $s = 1, \dots, d$, of the converged Prob-GPararealsolution $\mathcal{U}_{i,K_{\text{conv}}}$ for $i \in 0, \dots, N$, $t \in [t_0, t_N]$, with N and t_N which are system-specific, see Table 3 in Appendix G. The Lyapunov times (for Rössler and Lorenz) are computed as described in Section 5.2. The simulation setup is as in Table 1.

and Rössler, which we explore further in Section 5.2. A less pronounced effect is observed for the double pendulum, while the uncertainty stabilizes for FHN and Hopf.

5.2 Focus: Chaotic systems

As seen in Table 1, the performance for the Lorenz system is slightly worse than for the other systems in terms of scoring rules. Additionally, in Figure 1 and Figure 2, we noticed a steady increase in the normalized bias and in the coordinate-wise standard deviation of the solution for both Lorenz and Rössler shown. This is an unavoidable consequence of the chaotic nature of the systems, specifically their sensitivity to initial conditions: two arbitrarily close trajectories diverge exponentially fast over time, with the divergence rate governed by the largest Lyapunov exponent (Alligood et al., 1996). This increasing uncertainty limits the predictive capability of data-driven models, with reasonably accurate prediction horizons characterized in terms of the Lyapunov time, defined as the inverse of the largest Lyapunov exponent. These challenges are well known in the literature (Pathak and Ott, 2021; Pathak et al., 2017; Lu et al., 2018; Griffith et al., 2019; Vlachas et al., 2020). In the figures, we show, when relevant, the Lyapunov times, defined as the ratio between the system time and the Lyapunov time, to contextualize the prediction horizons. Using the estimates for the Lyapunov exponents in Sprott (2003), we obtain a Lyapunov time equal to 14 for Rössler, and 1.1 for Lorenz.

In Figure 3, top panel, we illustrate the impact of the initial condition on the evolution of the fine solver $\mathscr{F}$ for the 2nd coordinate of the Rössler system over $t \in [0, 340]$, twice as much

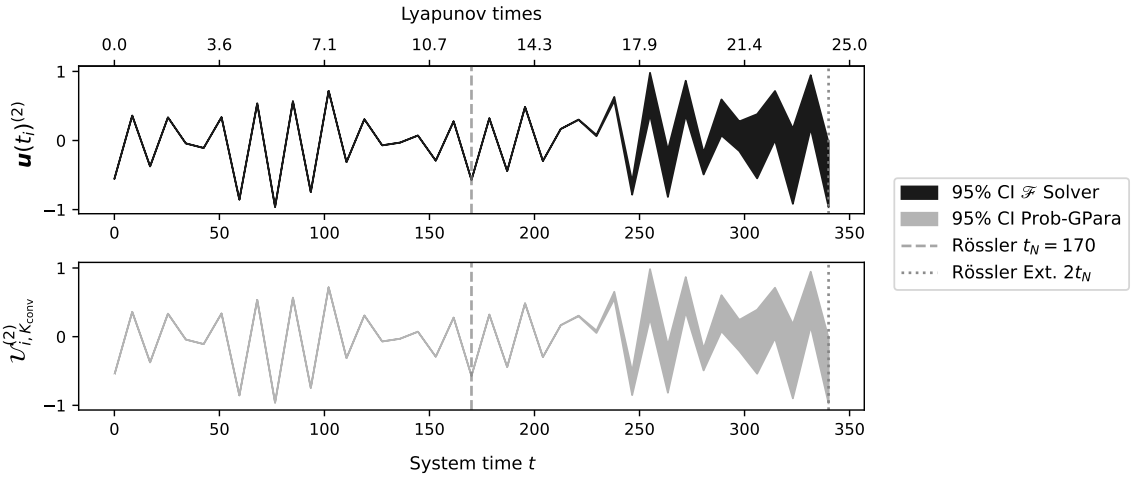

Figure 3: Impact of the initial condition on the solution of the chaotic Rössler system evolved over a timespan of $t \in [0, 340]$ (Rössler Ext. in the legend), twice longer than the timespan previously considered with $t_N = 170$ in Figure 1 and Figure 2. Only the second coordinate (the y-coordinate) is shown. Top: empirical 95% CI for the solution obtained via the fine solver $\mathscr{F}$, estimated using 1000 initial conditions sampled from a multivariate Gaussian distribution centered at $\boldsymbol{u}_{(0)}$ with diagonal covariance matrix and identical standard deviation of $5e^{-10}$. Bottom: Prob-GParareal 95% CI obtained with $\epsilon = 1e^{-9}$, and initial condition $\boldsymbol{u}_{0,0} = \boldsymbol{u}_{(0)}$. The Lyapunov times are computed as described in Section 5.2.

as the previously considered timespan. We do this by sampling 1000 initial conditions from a multivariate Gaussian distribution centered at $\boldsymbol{u}_{(0)}$ with diagonal covariance matrix and identical standard deviation of $5e^{-10}$, comparable to that observed at interval $i = 1$ (time t_1), and solve each IVP independently with $\mathscr{F}$. We adopt the same $\boldsymbol{u}_{(0)} = (0, -6.78, 0.02)$ as in previous works (Pentland et al., 2023b; Gattiglio et al., 2025). Note that the chosen $\boldsymbol{u}_{(0)}$ lies outside the Rössler attractor, and off-attractor transients may diverge differently before converging toward the invariant set, leading to more challenging numerical behavior. The exponential divergence after time 220 is noticeable, with an empirical 95% CI for the y-coordinate of Rössler covering half the sample space by the end of the simulation. In Figure 3, bottom, we present the corresponding 95% CI generated by Prob-GParareal, which closely matches that on top. Thus, contrary to what Table 1 and Figures 1 and 2 alone might suggest, Prob-GParareal accounts for uncertainty equally well in both chaotic and non-chaotic systems.

5.3 Impact of early algorithm termination

As seen in Table 1, Prob-GParareal (and GParareal) converges with relatively low runtimes. However, if these simulations were to become more computationally intensive, the runtime

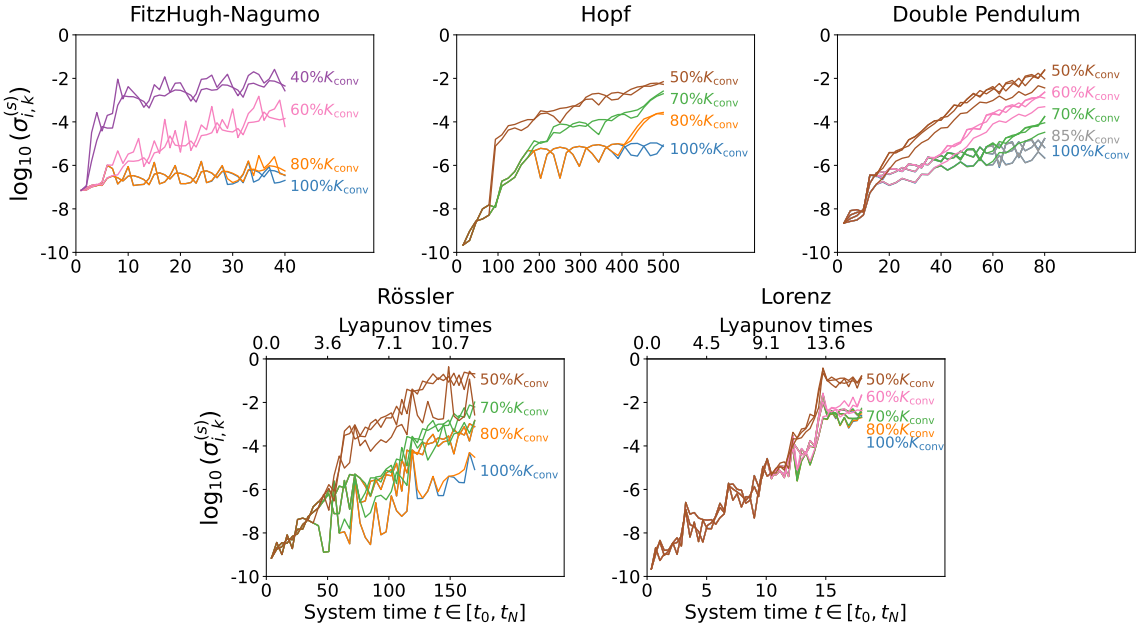

Figure 4: Evolution of the coordinate-wise standard deviation $\sigma_{i,k}^{(s)}$ (in $\log_{10}$), $s = 1, \dots, d$, of the Prob-GParareal solution $\mathcal{U}_{i,k}$ across system time t for different phases of the algorithm, when $k = l\%K_{\text{conv}}$ iterations to convergence are completed, with K_{conv} denoting the number of iterations to converge. The Lyapunov times are computed as described in Section 5.2. The simulation setup is the same as that used in Table 1.

would increase significantly, without a reliable way to estimate completion time in advance. For instance, this would be the case when solving PDEs with thousands of spatial discretization points (Gattiglio et al., 2024). In the worst-case scenario, Parareal-like algorithms ensure convergence within a time comparable to the sequential fine solver execution, which is often impractical. If simulations are stopped prior to convergence, say at time t_i, there are no guarantees or information on the error of unconverged intervals $t_{i+1}, \dots, t_N$. In contrast, a probabilistic numerical solver enables UQ at *any* point during execution, providing insight both at the conclusion of the algorithm and while running it. This is what we observe in Figure 4, where we report the evolution of the coordinate-wise standard deviation of the solution $\mathcal{U}_{i,k}$ across system time, at different stages of the algorithm, with $K_{\text{end}} = l\%K_{\text{conv}}$, $l \in [40, 100]$ iterations to convergence completed when the algorithm is terminated, potentially prior to convergence at iteration K_{conv} (generally unknown before execution), i.e., $K_{\text{end}} = K_{\text{stop}} < K_{\text{conv}}$ (early termination). As more Prob-GParareal iterations are carried out, and the algorithm approaches convergence, the uncertainty in the solution decreases.

The information carried by the UQ embedded in Prob-GParareal may also be used to set stopping criteria (e.g., based on the Wasserstein distance or the variance of the solution) and/or termination rules (e.g., defined in terms of the maximum allowed runtime). The

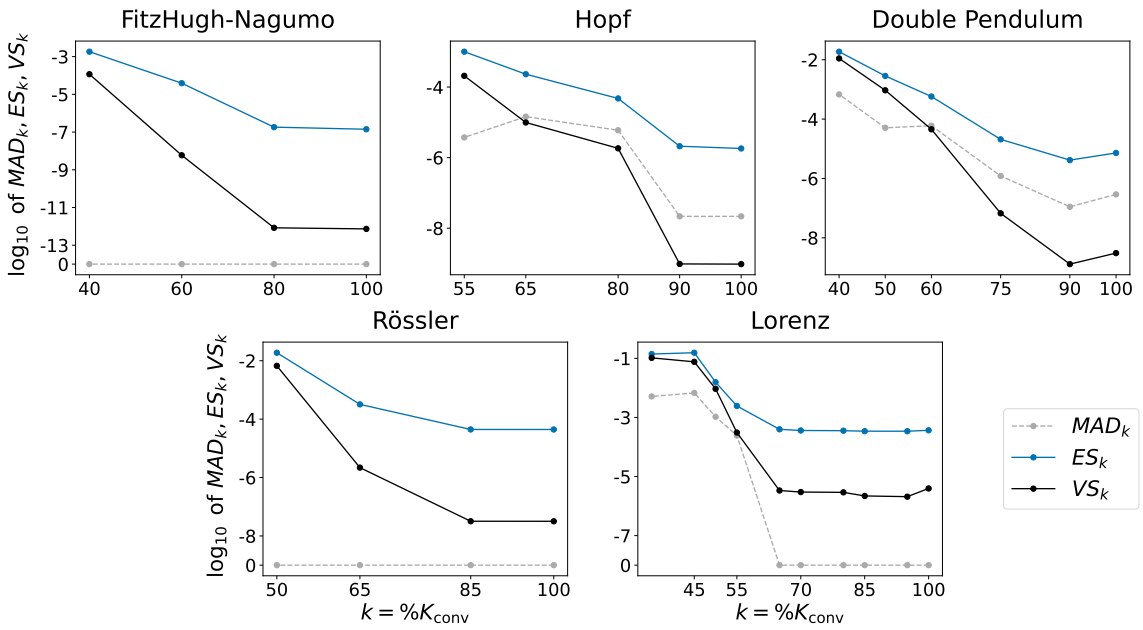

Figure 5: Evolution of MAD_k, ES_k, and VS_k (all in $\log_{10}$) of the Prob-GParareal solution $\mathcal{U}_{i,k}$ for different phases of the algorithm, i.e., when $k = l\%K_{\text{conv}}$ iterations to convergence are completed, with K_{conv} denoting the number of iterations to converge. The simulation setup is the same as that used in Table 1.

advantages of early termination compared with traditional convergence are two-fold. From an algorithm design perspective, choosing a priori an optimal/suitable threshold ϵ for the Wasserstein distance can be challenging. However, by taking UQ into account, we may choose it based on the variance of the Prob-GParareal solution, or use such variance to construct early stopping criteria. For example, we may terminate the algorithm when the variance of the solution stops decreasing in consecutive iterations, as a sign that there is no further improvement in the learning of Prob-GParareal. For instance, by looking at Figure 4, we may stop Prob-GParareal for the FHN, Lorenz and double pendulum systems at $K_{\text{stop}} = 80\%K_{\text{conv}}$, with similar results in terms of performance to the converged Prob-GParareal solution, as seen when looking at the score metrics (MAD_k, ES_k, and VS_k, in $\log_{10}$) in Figure 5. Interestingly, similar results are observed for different values of ϵ and n, as shown in Figure 8 in Appendix C. This early termination of the algorithm results in a lower runtime, with a gain that depends on the model, as shown in Figure 12 in Appendix E. See also Figure 7 and Section 5.4 for a further discussion on the impact of the variance of the solution on the convergence of the algorithm.

Alternatively, we may stop the algorithm based on the largest variance allowed over the solution timespan specified in the considered applications. For example, suppose the user can tolerate a maximum coordinate-wise standard deviation of $\sigma_{i,k}^{(s)} = 5e^{-4}$ in the probabilistic forecast. As shown in Figure 4, FHN, Hopf, and Rössler would be terminated

before convergence, at $K_{\text{stop}} = 80\%K_{\text{conv}}$, the double pendulum at $K_{\text{stop}} = 70\%K_{\text{conv}}$, while Lorenz would continue until convergence, as its maximum (converged) $\sigma_{i,k}^{(s)}$ exceeds $5e^{-4}$. The performance of the early terminated solution in terms of score metrics is similar to the converged one, see Figure 5, with advantages in terms of reduced runtime illustrated in Figure 12 in Appendix E.

The second advantage of early termination arises when the runtime of the algorithm exceeds the available computational budget. We may think, for example, of processes running on cloud infrastructure that are typically priced by usage. In such cases, forced termination ensures that the computations remain within budget constraints. The resulting forecast variance then reflects the uncertainty due to incomplete computation, which tends to be higher for intervals i farther away from the initial condition, as shown in Figure 4.

5.4 Impact of random initial conditions on Prob-GParareal

We now investigate the impact of random initial conditions on the forecast and convergence of Prob-GParareal. Let $\boldsymbol{U}_{0,0}$ be multivariate Gaussian distributed, with mean $\boldsymbol{u}_0$, and diagonal covariance matrix with identical diagonal entries $\sigma_{\text{init}}^2 \in \mathbb{R}^+$, which we now assume strictly positive after having focused on $\sigma_{\text{init}} = 0$ (i.e. deterministic starting conditions) before. We explore values for the initial standard deviation σ_{init} ranging from $1e^{-6}$ to $1e^{-2}$. Larger values, such as $\sigma_{\text{init}} = 5e^{-2}$, lead to high uncertainty in the solution of the considered models, and are therefore excluded. Although these values may appear small , it is important to recall that the data have been normalized to lie within $[-1, 1]^d$. Thus, on this scale, such standard deviations are not negligible.

The impact of random initial conditions on the uncertainty of the Prob-GParareal solution is illustrated in Figure 6. The standard deviation of the final solution at $i = 0$, that is, $\sigma_{0,K_{\text{end}}}^{(s)}$, is always higher than that in the deterministic case, reported in Figure 2, and matches σ_{init}, which sets a lower bound on the solution standard deviation across all considered systems. Moreover, the larger is σ_{init}, the larger is the solution standard deviation, resulting in stratified graphs. This behavior appears in both non-chaotic and chaotic systems: the former exhibit a relatively flat standard deviation trend over time, while the latter (here Rössler and Lorenz, and to a minor extent, the double pendulum) show the expected exponential growth in uncertainty (note the logarithmic scale on the y-axis).

As expected, high standard deviations in the Prob-GParareal solution (e.g., $\sigma_{i,k}^{(s)} \in [0.05, 1]$, depending on the system) are associated with slow or nonconvergent behavior of the algorithm. This can be seen in Figure 7 for the FHN and Lorenz systems, where the left panels show the percentage of converged intervals across iterations, and the right panels the standard deviation $\sigma_{L+1,k}^{(s)}$ of the intermediate solution at iteration k, for the first unconverged interval $L+1$ (recall that L denotes the number of converged intervals, see Section 3.2). We choose to track the standard deviation for the first $L+1$ interval as it tends to be the smallest across all the unconverged intervals $i = L+1, \dots, N$. We note that when the coordinate-wise standard deviation $\sigma_{L+1,k}^{(s)}$ reaches values around $5e^{-2}$ (from $k = 1$ for FHN when $\sigma_{\text{init}} = 1e^{-2}$, at iteration $k = 10$ for Lorenz over most values of σ_{init}; right panels), the algorithm shows slow or nonconvergent behavior. This is reflected by the evolution of the percentage of converged intervals in the left panels, which for FHN increases to 100% slowly, for $\sigma_{\text{init}} = 1e^{-2}$, while for Lorenz flattens around $k = 10$ for most values of σ_{init}. To save computation, we

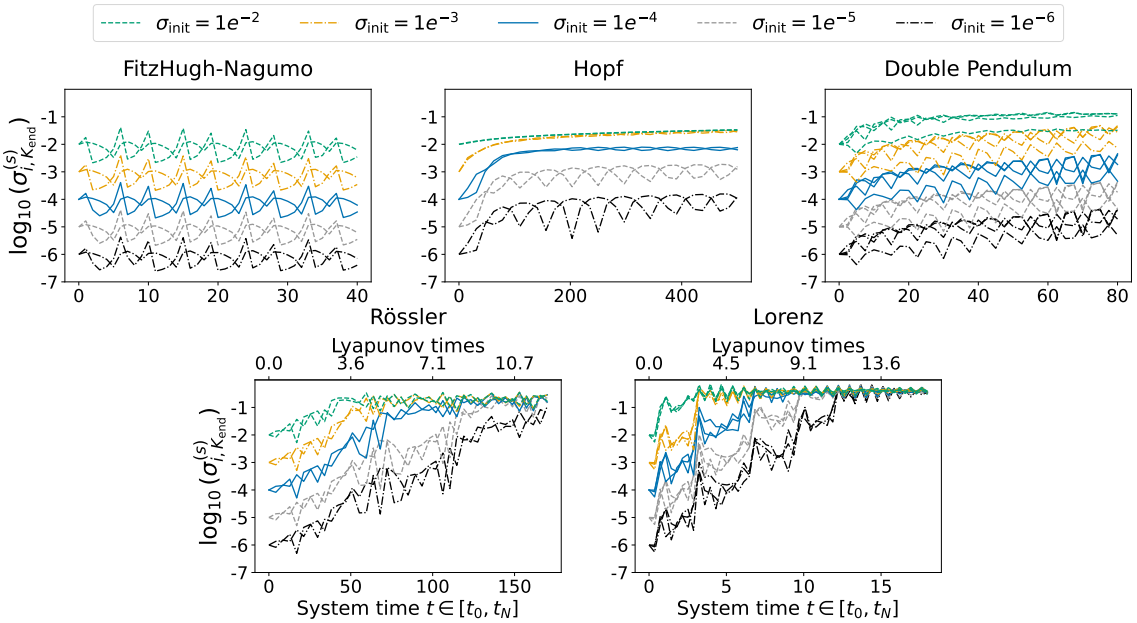

Figure 6: Impact of random initial conditions $\boldsymbol{U}_{0,0}$ on the coordinate-wise standard deviation $\sigma^{(s)}_{i,K_{\text{end}}}, s = 1, \ldots, d$ (in $\log_{10}$) of the converged Prob-GPArareal solution $\boldsymbol{U}_{i,K_{\text{end}}}$ for different systems. Here, $\boldsymbol{U}_{0,0}$ is a d-dimensional Gaussian distribution centered at $\boldsymbol{u}_{(0)}$ with diagonal covariance matrix with diagonal entries equal to σ^2_{init}, where $\sigma_{\text{init}} = 1e^{-l}$, $l = 2, 3, 4, 5, 6$. Simulations displaying nonconvergent behavior were stopped before convergence, at $K_{\text{end}} = K_{\text{stop}}$, with K_{stop} given in Table 3 in Appendix G. The Lyapunov times are computed as described in Section 5.2.

suggest stopping the algorithm once this behavior is detected , leading to an additional early termination criterion, similar to the first proposed in Section 5.3. Overall, we recommend choosing a starting condition $\boldsymbol{U}_{0,0}$ with $\sigma_{\text{init}} < \epsilon$ and $\sigma_{\text{init}} \ll \epsilon$ for chaotic systems.

6. Empirical results for Prob-nnGPArareal

In Section 3.3, we showed that replacing the full GP with an nnGP (as proposed in nnGPArareal Gattiglio et al. 2025) yields substantial computational savings in the proposed probabilistic algorithm, Prob-nnGPArareal. Importantly, using the nearest neighbors approximation has a limited impact, if any, on the predictive accuracy and uncertainty estimation of the algorithm, as shown in Figure 13 in Appendix F, where we replicate the analysis carried out in Figure 6, Section 5.4, replacing the GP with an nnGP. While in some cases the use of nnGPs results in a slight increase in uncertainty over time compared to Prob-GPArareal, the overall performance of the algorithm remains largely unaffected.

The reduced computational cost of Prob-nnGPArareal (combined with an accuracy comparable to that of Prob-GPArareal) makes it appropriate for solving PDEs. Here, we

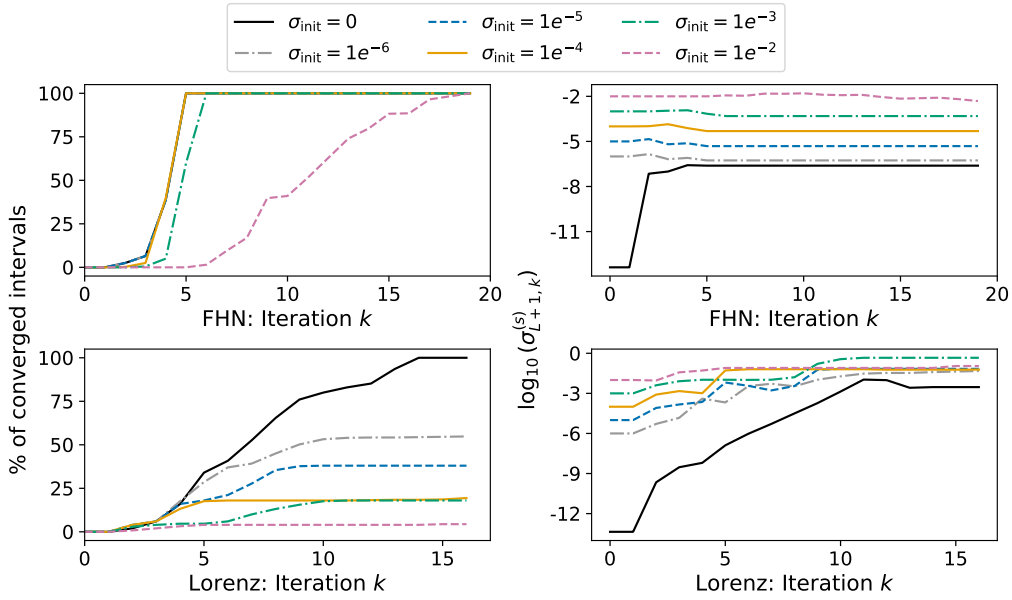

Figure 7: Impact of the initial standard deviation σ_{init} of $\boldsymbol{U}_{0,0}$ on the Prob-GParareal convergence (left panels) and standard deviation (right panels) for the FHN (top panels) and Lorenz (bottom panels) systems. Left panels: percentage of converged intervals per iteration k for different σ_{init} values. Right panels: standard deviation of $\mathcal{U}_{L+1,k}$, $\sigma^{(s)}_{L+1,k}$ (in $\log_{10}$), where $L+1$ is the first unconverged interval. High variance at $L+1$ is associated with slow convergence.

consider the Viscous Burgers equation (Schmitt et al., 2018), a nonlinear one-dimensional PDE defined as

$$u_t = \nu u_{xx} - u u_x, \quad (x,t) \in [-L, L] \times [t_0, t_N], \tag{32}$$

with initial condition $u(x, t_0) = u_{(0)}(x)$, $x \in [-L, L], L > 0$, and periodic boundary conditions $u(-L, t) = u(L, t)$, $u_x(-L, t) = u_x(L, t)$, $t \in [t_0, t_N]$. We use the same setting and parameter values as in nnGParareal (Gattiglio et al., 2025) for a direct comparison. Specifically, we take $L = 1$, the diffusion coefficient $\nu = 0.01$, and discretize the spatial domain using finite difference (Fornberg, 1988) and equally spaced points $x_{l+1} = x_l + \Delta x$, with $\Delta x = 2L/d$ and $l = 0, \dots, d$, thus reformulating the PDE as a d-dimensional ODE system. We choose $N = d = 128$, $u_{(0)}(x) = 0.5(\cos(\frac{9}{2}\pi x) + 1)$, $t_0 = 0$, and $t_N = 5$. Moreover, we took $\mathscr{G}$ and $\mathscr{F}$ to be Runge-Kutta of order 1 and 8, respectively, with a number of integration steps for numerical solvers over $[t_0, t_N]$ of 512 for $\mathscr{G}$ and $5.12e^6$ for $\mathscr{F}$.

Note that in the present experiment both the fine and coarse propagators are applied to the same spatial discretization. More generally, however, the framework can accommodate different spatial discretizations for $\mathscr{F}$ and $\mathscr{G}$ by introducing suitable restriction and interpolation/prolongation operators between the corresponding spatial grids, as commonly done in Parareal methods with spatial coarsening and related multigrid-in-time methods (Angel et al., 2021; Howse et al., 2019). This would not fundamentally alter the structure of the

Algorithm	$\text{Bias}_{K_{\text{conv}}}$	$\text{MSE}_{K_{\text{conv}}}$	$\text{MAD}_{K_{\text{conv}}}$	$\text{VS}_{K_{\text{conv}}}$	$\text{ES}_{K_{\text{conv}}}$
Fine solver $\mathscr{F}$	–	–	–	–	–
Parareal	$1.73e^{-07}$	$7.12e^{-14}$	–	–	–
GParareal	$1.03e^{-07}$	$3.21e^{-14}$	–	–	–
nnGParareal	$3.47e^{-07}$	$1.33e^{-13}$	–	–	–
Prob-nnGPara	$3.23e^{-07}$	$1.32e^{-12}$	$3.53e^{-13}$	$5.16e^{-10}$	$3.65e^{-07}$

Algorithm	K_{conv}	$T_{\mathscr{G}}$	$T_{\mathscr{F}}$	T_{model}	T_{alg}
Fine solver $\mathscr{F}$	–	–	–	–	13h 5m
Parareal	10	0s	6m	0s	1h 4m
GParareal	6	0s	6m	1h 2m	1h 39m
nnGParareal	9	0s	6m	7m	1h 3m
Prob-nnGPara	5	3s	6m	3m	33m

Table 2: Performance of Parareal, GParareal, nnGParareal, and Prob-nnGParareal on Viscous Burgers' PDE (32), solved as a d-dimensional ODE, with $N = d = 128$. K_{conv} is the number of iterations to convergence, $T_{\mathscr{G}}$ and $T_{\mathscr{F}}$ are the runtimes of $\mathscr{G}$ and $\mathscr{F}$ per iteration, respectively, while T_{model}, and T_{alg} are the total cost of evaluating the correction function f_c in (5), and the total runtime of the algorithm, respectively.

algorithm, although the transfer operators would introduce additional approximation errors that should be accounted for in a refined analysis.

In Table 2, we compare the performance of Parareal, GParareal, nnGParareal, and Prob-nnGParareal, in terms of the number of iterations to converge K_{conv}, computational cost of running $\mathscr{G}$, $\mathscr{F}$ and each algorithm, and the training cost T_{model}, together with an evaluation of each algorithm's performance with respect to the solution provided by the fine solver $\mathscr{F}$. Specifically, we compute the bias and MSE for all the algorithms, and we additionally use the scoring rules for the probabilistic Prob-nnGParareal solution. On the one hand, the runtime of Prob-nnGParareal is lower than its deterministic counterpart, nnGParareal, thanks to a lower K_{conv} to converge (with similar runtimes/costs per iteration). On the other hand, the solution accuracy, given in terms of both Bias_k and MSE_k, is comparable. However, such comparisons should be made with caution, as Prob-nnGParareal returns samples from a distribution rather than a single value, and, most importantly, the convergence criteria are inherently different, as the threshold ϵ controls the Wasserstein distance between probability distributions in Prob-(nn)GParareal, and the L_∞ distance between vectors in the other algorithms. Thus, the two ϵ are not immediately comparable, even if a lower ϵ leads to higher accuracy, as shown in Figure 8 in Appendix C for Prob-GParareal.

7. Discussion

In this paper, we have introduced Prob-(nn)GParareal, a probabilistic extension of the (nn)GParareal algorithm(s) capable of capturing and propagating the underlying numerical uncertainty. This novel contribution addresses a general gap in probabilistic approaches in the PinT literature. Unlike the work of Bosch et al. (2024), our approach does not rely on approximations for nonlinear ODE systems, and can be directly applied to any underlying numerical solver, facilitating its adoption in existing Parareal applications (e.g., for specific parabolic, hyperbolic or chaotic problems). UQ is achieved by modeling the correction function with (nn)GPs, and propagating the resulting posterior distribution through the

(generally nonlinear) coarse solver $\mathscr{G}$ via Monte Carlo sampling. The additional complexity introduced by Prob-(nn)GParareal is fully parallelizable, reducing the total cost to that of the corresponding deterministic version. Prob-GParareal also offers flexible resource management via early termination rules, supports probabilistic initial conditions, and shows good scalability in the number of processors by implementing recent advances based on nnGPs (Gattiglio et al., 2025), allowing the use of Prob-nnGParareal for PDE solutions.

Despite its several advantages, Prob-GParareal's ability to properly account for the numerical uncertainty depends on the assumptions that: 1) the fine solver $\mathscr{F}$ yields the true solution (which is common within the PinT literature); 2) the variance across the coordinates of the system can be modeled using an intrinsic coregionalization model (Goovaerts, 1997), as discussed in Section 3.1. While these assumptions enable computational feasibility without requiring a linear vector field for the DE system, they may not always hold, leading to a bias in the mean and variance of the solution, respectively. The first assumption may be relaxed by either training local GPs to learn it, or using probabilistic solvers of it, allowing, in both cases, UQ for $\mathscr{F}$. The GP covariance assumption can be relaxed by selecting more flexible GP models. In the most general multi-output case, this would require forming a covariance matrix of size $(Dd) \times (Dd)$, where D is the size of the dataset and d is the number of coordinates to be jointly modeled, with exact GP inference scaling as $\mathcal{O}(D^3 d^3)$. This may be feasible for nnGPs used on low-dimensional ODEs. Scaling to higher-dimensional ODEs and PDEs may require replacing the (nn)GPs with more scalable alternatives, as done for RandNet-Parareal in Gattiglio et al. (2024) for a deterministic algorithm. We defer the exploration of these problems to future research.

Acknowledgments

GG is funded by the Warwick Centre for Doctoral Training in Mathematics and Statistics. MT and LG acknowledge the financial support of the United Kingdom Engineering and Physical Sciences Research Council (EPSRC) grant EP/X020207/1. We are grateful to the Scientific Computing Research Technology Platform (SCRTP) at Warwick for the provision of computational resources. All experiments were performed using the Warwick University HPC facilities on Dell PowerEdge C6420 compute nodes each with 2 x Intel Xeon Platinum 8268 (Cascade Lake) 2.9 GHz 24-core processors, with 48 cores per node and 192 GB DDR4-2933 RAM per node. Python code accompanying this manuscript is available at https://github.com/Parallel-in-Time-Differential-Equations/ProbParareal.

Appendix A. Additional details on Gaussian process formulation

In this section, we provide the expressions for the GP posterior mean $\boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}') \in \mathbb{R}$ and posterior variance $\sigma^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}')^2 \in \mathbb{R}^+$ introduced in Section 2. These are given by

$$\mu^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}') = K(X, \boldsymbol{u}')^\top \left(K(X,X) + \sigma^2_{\text{reg}} \mathbb{I}_{Nk}\right)^{-1} Y_{(\cdot,s)}, \tag{33}$$

$$\sigma^{(s)}_{\mathcal{D}_k}(\boldsymbol{u}')^2 = K_{\text{GP}}(\boldsymbol{u}', \boldsymbol{u}') - K(X, \boldsymbol{u}')^\top \left(K(X,X) + \sigma^2_{\text{reg}} \mathbb{I}_{Nk}\right)^{-1} K(X, \boldsymbol{u}'), \tag{34}$$

where $X \in \mathbb{R}^{Nk\times d}$ denotes the matrix of inputs $\boldsymbol{u}_{i-1,j}$ and $Y \in \mathbb{R}^{Nk\times d}$ the matrix of outputs $f_c(\boldsymbol{u}_{i-1,j})$, taken from the dataset $\mathcal{D}_k$, with both matrices arranged by rows. The symbol $\mathbb{I}_{Nk}$ denotes the identity matrix of size Nk, while the vector $K(X,\boldsymbol{u}') \in \mathbb{R}^{Nk}$ contains the covariances between every input in X and the test point $\boldsymbol{u}'$, defined as

$$(K(X,\boldsymbol{u}'))_r = K_{\mathrm{GP}}({X_{(r,\cdot)}}^\top, \boldsymbol{u}'), \quad r = 1,\dots,Nk,$$

where ${X_{(r,\cdot)}}^\top$ denotes the transpose of the rth row of X. Similarly, the covariance matrix $K(X,X) \in \mathbb{R}^{Nk\times Nk}$ is given by

$$(K(X,X))_{r,q} = K_{\mathrm{GP}}({X_{(r,\cdot)}}^\top, {X_{(q,\cdot)}}^\top), \quad r,q = 1,\dots,Nk.$$

Here, as K_{GP} kernel, we use the Gaussian kernel K_{G}, also known as radial basis function or square-exponential kernel, defined as

$$K_{\mathrm{G}}(\boldsymbol{u},\boldsymbol{u}') = \sigma_{\mathrm{o}}^2 \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{u}'\|^2}{\sigma_{\mathrm{i}}^2}\right), \quad \boldsymbol{u},\boldsymbol{u}' \in \mathbb{R}^d, \tag{35}$$

where σ_{o}^2 and σ_{i}^2 represent the output and input length scales, respectively, and $\|\cdot\|$ denotes the Euclidean norm[3]. The term σ_{reg}^2 in (33) is a regularization term, often referred to as jitter or nugget, which is used to improve the condition number of the covariance matrix during inversion. Note that the posterior variance in (34) depends on the coordinate s, even though there is no explicit reference to s in the expression. This is because the regularization term σ_{reg}^2 and the input and output length scales (σ_{i}^2 and σ_{o}^2) form the hyperparameters of the sth GP, $\boldsymbol{\theta}^{(s)} := (\sigma_{\mathrm{i}}^{(s)\,2}, \sigma_{\mathrm{o}}^{(s)\,2}, \sigma_{\mathrm{reg}}^{(s)\,2})$, where we add the superscript s for clarity. Hence, $\boldsymbol{\theta}^{(s)}$ must be tuned independently for each of the d coordinates. This is typically achieved through the numerical maximization of the marginal log-likelihood. Define $K_\theta^{(s)} := K^{(s)}(X,X) + \sigma_{\mathrm{reg}}^{(s)2}\mathbb{I}_{|\mathcal{D}_k|}$ and consider

$$\log p(Y_{(\cdot,s)}|X,\boldsymbol{\theta}^{(s)}) \propto -{Y_{(\cdot,s)}}^\top {K_\theta^{(s)}}^{-1} Y_{(\cdot,s)} - \log\det(K_\theta^{(s)}),$$

where $K^{(s)}(X,X)$, and hence $K_\theta^{(s)}$, depends on $\boldsymbol{\theta}^{(s)}$ through the kernel function K_{GP}. For additional details on this process and the role of the regularization parameter σ_{reg}^2, refer to Gattiglio et al. (2025). Once the optimal hyperparameters are identified, the GPs are trained and can be used to make predictions.

Appendix B. Proof of theoretical results

In this section, we provide the proofs for the theorems and corollary presented in the manuscript. Before doing that, we recall some prior results needed for the proofs.

3. Other choices of kernels are also widely used, e.g. the Matérn kernel in spatial statistics analysis (Matérn, 1986). For constants $\nu > 0$ (smoothness parameter), $\sigma_{\mathrm{i}}^2 > 0$ (input length scale), $\sigma_{\mathrm{o}}^2 > 0$ (output length scale), the Matérn kernel $K_{\nu,\sigma_{\mathrm{i}},\sigma_{\mathrm{o}}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is defined as

$$K_{\nu,\sigma_{\mathrm{i}},\sigma_{\mathrm{o}}}(\boldsymbol{u},\boldsymbol{u}') = \sigma_{\mathrm{o}}^2 \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\,\|\boldsymbol{u}-\boldsymbol{u}'\|}{\sigma_{\mathrm{i}}}\right)^\nu k_\nu \left(\frac{\sqrt{2\nu}\,\|\boldsymbol{u}-\boldsymbol{u}'\|}{\sigma_{\mathrm{i}}}\right), \quad \boldsymbol{u},\boldsymbol{u}' \in \mathbb{R}^d,$$

where $\Gamma(\cdot)$ is the gamma function, and k_ν is the modified Bessel function of the second kind of order ν.

B.1 Prior results from the literature

We recall two results establishing the convergence of scalar-output GP posterior mean prediction (Theorem 27 from Wendland 2004) and the rate of decay of the prediction variance (Wu and Schaback 1993; Kanagawa et al. 2018), which we state here for the scalar-output GP used to model the sth coordinate of the correction function $f_c^{(s)} = (\mathscr{F} - \mathscr{G})^{(s)}$. Starting from the variance decay, we report a two-part theorem, using the formulation of Theorem 5.4 in Kanagawa et al. (2018) in part **(i)**, and Theorem 5.14 in Wu and Schaback (1993) in part **(ii)**.

Theorem 25 (Kanagawa et al. 2018; Wu and Schaback 1993) *Let K be a kernel on $\mathbb{R}^d$ and let $\mathcal{H}_K$ be the RKHS induced by it. Let $f_c^{(s)} = (\mathscr{F} - \mathscr{G})^{(s)} \in \mathcal{H}_K$, $s = 1, \dots, d$, and let $\sigma_{\mathcal{D}}^{(s)}(\boldsymbol{u}')^2 \in \mathbb{R}$, $s = 1, \dots, d$, be the posterior variance of a scalar-output GP built on a dataset $\mathcal{D}$ to approximate $f_c^{(s)} \in \mathcal{H}_K$, $s = 1, \dots, d$. Then:*

(i) *If K is such that its associated RKHS is norm-equivalent to $W_2^q(\mathbb{R}^d)$, the Sobolev space of order q, then for any $\rho > 0$, there exist constants $h_0 > 0$ and $\{C_s\}_{s=1}^d$, $C_s > 0$, such that for any $\boldsymbol{u}' \in \mathbb{R}^d$ and any set of points $\mathcal{D} = \{\boldsymbol{u}_1, \dots, \boldsymbol{u}_n\} \subset \mathbb{R}^d$ satisfying $h_{\rho,\mathcal{D}}(\boldsymbol{u}') \leq h_0$, it holds that $\sigma_{\mathcal{D}}^{(s)}(\boldsymbol{u}')^2 \leq C_s h_{\rho,\mathcal{D}}(\boldsymbol{u}')^{2q-d}$, for $s = 1, \dots, d$.*

(ii) *If K is infinitely smooth, then for any $\rho > 0$ and $\alpha > 0$, there exist constants $h_\alpha > 0$ and $\{C_{\alpha,s}\}_{s=1}^d$, $C_{\alpha,s} > 0$, all depending on α, such that for any $\boldsymbol{u}' \in \mathbb{R}^d$ and any set of points $\mathcal{D} = \{\boldsymbol{u}_1, \dots, \boldsymbol{u}_n\} \subset \mathbb{R}^d$ satisfying $h_{\rho,\mathcal{D}}(\boldsymbol{u}') \leq h_\alpha$, it holds that $\sigma_{\mathcal{D}}^{(s)}(\boldsymbol{u}')^2 \leq C_{\alpha,s} h_{\rho,\mathcal{D}}(\boldsymbol{u}')^\alpha$, for $s = 1, \dots, d$.*

Remark 26 *The Gaussian kernel in Appendix A is infinitely smooth. By Corollary 10.48 in Wendland (2004), the Matérn kernel with smoothness parameter $\nu > 0$ on $\mathbb{R}^d$ presented in Appendix A induces an RKHS $\mathcal{H}_K$ that is norm-equivalent to the Sobolev space $W_2^q(\mathbb{R}^d)$ with $q = \nu + d/2$. Note that functions in $\mathcal{H}_K$ are weak differentiable up to order q and differentiable in the classical sense only up to ν (see Remark 2.11 in Kanagawa et al. 2018).*

Theorem 27 (Wendland (2004), Theorem 11.4) *In the conditions of Theorem 25, the error between the correction function $f_c^{(s)} \in \mathcal{H}_K$ and the scalar-valued GP posterior mean $\mu_{\mathcal{D}}^{(s)}$, $s = 1, \dots, d$, estimated on $\mathcal{D} \subset \mathbb{R}^d$, satisfies*

$$|f_c^{(s)}(\boldsymbol{u}') - \mu_{\mathcal{D}}^{(s)}(\boldsymbol{u}')| \leq \sigma_{\mathcal{D}}^{(s)}(\boldsymbol{u}') \, \|f_c^{(s)}\|_{\mathcal{H}_K}, \quad \text{for any } \boldsymbol{u}' \in \mathbb{R}^d.$$

We are now ready to prove Theorem 13 in Appendix B.2, Theorem 19 in Appendix B.3, Theorem 21 and Corollary 23 in Appendix B.4.

B.2 Proof of Theorem 13

In this section, we prove Theorem 13 for the infinite smoothness case, case **(iii)**. Cases **(i)** and **(ii)** require minor modifications, which are detailed at the end of the Section.

Case (iii) Smoothness. Consider the squared Wasserstein-2 distance between the true solution $\boldsymbol{u}(t_i)$ of (1) at time t_i and the distribution $P_{\boldsymbol{U}_{i,k}}$ of the Prob-GParareal solution at interval $i \in \{1, \dots, N\}$ and iteration $k \in \mathbb{N}$ given by

$$e_{i,k} := W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2 = \inf_{\gamma \in \Gamma(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\boldsymbol{u}(t_i) - \boldsymbol{u}_{i,k}\|^2 d\gamma(\boldsymbol{u}(t_i), \boldsymbol{u}_{i,k}),$$

where $\Gamma(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})$ is the set of all couplings (joint distributions) with marginals $\delta_{\boldsymbol{u}(t_i)}$ and $P_{\boldsymbol{U}_{i,k}}$. Since $\delta_{\boldsymbol{u}(t_i)}$ is a Dirac measure, the optimal coupling is unique and is given by the product measure (see Villani 2008) $\gamma = \delta_{\boldsymbol{u}(t_i)} \otimes P_{\boldsymbol{U}_{i,k}}$. Thus,

$$\begin{aligned} e_{i,k} &= \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\boldsymbol{u}(t_i) - \boldsymbol{u}_{i,k}\|^2 d(\delta_{\boldsymbol{u}(t_i)} \otimes P_{\boldsymbol{U}_{i,k}})(\boldsymbol{u}(t_i), \boldsymbol{u}_{i,k}) \\ &= \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\boldsymbol{u}(t_i) - \boldsymbol{u}_{i,k}\|^2 \, d\delta_{\boldsymbol{u}(t_i)}\left(\boldsymbol{u}(t_i)\right) \, dP_{\boldsymbol{U}_{i,k}}(\boldsymbol{u}_{i,k}) \\ &= \int_{\mathbb{R}^d} \|\boldsymbol{u}(t_i) - \boldsymbol{u}_{i,k}\|^2 \, dP_{\boldsymbol{U}_{i,k}}(\boldsymbol{u}_{i,k}) = \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\|\boldsymbol{u}(t_i) - \boldsymbol{U}_{i,k}\|^2\right], \end{aligned} \tag{36}$$

i.e., the W_2^2 distance equals the MSE of $\boldsymbol{u}(t_i)$, proving (21). We then seek to build and solve a recurrence relation for $e_{i,k}$. We first use the Prob-GParareal update rule (9) and the assumption that the fine solver $\mathscr{F}$ is exact, and write

$$\boldsymbol{u}(t_i) - \boldsymbol{U}_{i,k} = \mathscr{F}(\boldsymbol{u}(t_{i-1})) - \mathscr{G}(\boldsymbol{U}_{i-1,k}) - \boldsymbol{Z}_{i,k}, \quad i = 1, \dots, N, \quad k \geq 1, \tag{37}$$

where the conditional distribution of $\boldsymbol{Z}_{i,k}$ given $\boldsymbol{U}_{i-1,k}$ is $\boldsymbol{Z}_{i,k}|\boldsymbol{U}_{i-1,k} \sim \mathcal{N}_d(\boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{U}_{i-1,k}), \Sigma_{\mathcal{D}_k}(\boldsymbol{U}_{i-1,k}))$, see (10), where $\boldsymbol{\mu}_{\mathcal{D}_k}$ and $\Sigma_{\mathcal{D}_k}$ are deterministic functions obtained by training a GP on $\mathcal{D}_k$, and for convenience, we write $\boldsymbol{Z}_{i,k}$ as

$$\boldsymbol{Z}_{i,k} = \boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{U}_{i-1,k}) + \boldsymbol{\xi}_{i,k}, \quad \text{with} \quad \boldsymbol{\xi}_{i,k}|\boldsymbol{U}_{i-1,k} \sim \mathcal{N}_d\left(\boldsymbol{0}, \Sigma_{\mathcal{D}_k}(\boldsymbol{U}_{i-1,k})\right).$$

To derive a recurrence for errors $e_{i,k}$, $i = 1, \dots, N$, $k \in \mathbb{N}$, we add and subtract $\mathscr{F}(\boldsymbol{U}_{i-1,k})$, $\mathscr{G}(\boldsymbol{u}(t_{i-1}))$, and $\mathscr{G}(\boldsymbol{U}_{i-1,k})$ on the right-hand side of (37), and making use of the definition of f_c, we obtain the following decomposition:

$$\begin{aligned} \boldsymbol{u}(t_i) - \boldsymbol{U}_{i,k} = &\underbrace{\left(f_c(\boldsymbol{U}_{i-1,k}) - \boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{U}_{i-1,k})\right)}_{Q} + \underbrace{(-\boldsymbol{\xi}_{i,k})}_{W} + \underbrace{\left(\mathscr{G}(\boldsymbol{u}(t_{i-1})) - \mathscr{G}(\boldsymbol{U}_{i-1,k})\right)}_{R} \\ &+ \underbrace{\left(f_c(\boldsymbol{u}(t_{i-1})) - f_c(\boldsymbol{U}_{i-1,k})\right)}_{S}, \end{aligned} \tag{38}$$

for all $i = 1, \dots, N$ and $k \in \mathbb{N}$. By taking norm squared and applying the expectation operator on both sides of (38), we obtain the following bound for the error expression (36):

$$\begin{aligned} e_{i,k} &= \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[\mathbb{E}_{\boldsymbol{U}_{i,k}|\boldsymbol{U}_{i-1,k}}\left[\|\boldsymbol{u}(t_i) - \boldsymbol{U}_{i,k}\|^2\right]\right] \\ &\leq 4\left(\mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[\|Q\|^2\right] + \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[\|W\|^2\right] + \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[\|R\|^2\right] + \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[\|S\|^2\right]\right). \end{aligned}$$

In this derivation, the first equality follows by the law of total expectation with the conditional expectation taken with respect to the conditional law in (11), while the inequality holds due to

$$\left\| \sum_{j=1}^{m} \boldsymbol{v}_j \right\|^2 \leq m \sum_{j=1}^{m} \|\boldsymbol{v}_j\|^2 , \tag{39}$$

for any $\boldsymbol{v}_j$, $j = 1, \ldots, m$, using the fact that the terms in (38) are measurable with respect to the conditional expectation. We continue by analyzing the terms on the right-hand side of this expression one by one. For the R term, by Assumption 7, we write

$$\mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\|R\|^2 \right] \leq L_{\mathscr{G}}^2 \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\|\boldsymbol{u}(t_{i-1}) - \boldsymbol{U}_{i-1,k}\|^2 \right] = L_{\mathscr{G}}^2 \, e_{i-1,k}.$$

Next, for the S term, note that Assumption 6 holds by Remark 9 under Assumption 5, and hence

$$\mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\|S\|^2 \right] \leq L_c^2 \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\|\boldsymbol{u}(t_{i-1}) - \boldsymbol{U}_{i-1,k}\|^2 \right] = L_c^2 \, e_{i-1,k}.$$

For the Q term, we proceed as follows:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\|Q\|^2 \right] &= \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\sum_{s=1}^{d} \left| f_c^{(s)}(\boldsymbol{U}_{i-1,k}) - \boldsymbol{\mu}_{\mathcal{D}_k}^{(s)}(\boldsymbol{U}_{i-1,k}) \right|^2 \right] \\
&\leq \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\sum_{s=1}^{d} \sigma_{\mathcal{D}_k}^{(s)} \left(\boldsymbol{U}_{i-1,k} \right)^2 \, \left\| f_c^{(s)} \right\|_{\mathcal{H}_K}^2 \right] \\
&\leq \max_{1 \leq s \leq d} \left\| f_c^{(s)} \right\|_{\mathcal{H}_K}^2 \sum_{s=1}^{d} \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\sigma_{\mathcal{D}_k}^{(s)} (\boldsymbol{U}_{i-1,k})^2 \right] \\
&= \|f_c\|_{\infty,\mathcal{H}_K}^2 \sum_{s=1}^{d} \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\sigma_{\mathcal{D}_k}^{(s)} (\boldsymbol{U}_{i-1,k})^2 \right],
\end{aligned}$$

where we used Theorem 27 in the first inequality and applied Definition 12 of the maximum norm in the last equality. Lastly, we apply Theorem 25 part **(ii)** to bound posterior variances ${\sigma_{\mathcal{D}_k}^{(s)}}^2$, $s = 1, \ldots, d$, and obtain

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\|Q\|^2 \right] &\leq \|f_c\|_{\infty,\mathcal{H}_K}^2 \sum_{s=1}^{d} C_{s,\alpha} \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[h_{\rho,\mathcal{D}_k} (\boldsymbol{U}_{i-1,k})^{\alpha} \right] \\
&\leq C_\alpha \, d \, \|f_c\|_{\infty,\mathcal{H}_K}^2 \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[h_{\rho,\mathcal{D}_k} (\boldsymbol{U}_{i-1,k})^{\alpha} \right],
\end{aligned}$$

with $C_\alpha = \max_{1 \leq s \leq d} C_{s,\alpha}$.

Finally, for the W term, we use Theorem 25 part **(ii)** and write:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\|W\|^2 \right] &= \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\mathrm{Tr} \left(\Sigma_{\mathcal{D}_k} (\boldsymbol{U}_{i-1,k}) \right) \right] = \sum_{s=1}^{d} \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[\sigma_{\mathcal{D}_k}^{(s)} (\boldsymbol{U}_{i-1,k})^2 \right] \\
&\leq \sum_{s=1}^{d} C_{s,\alpha} \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[h_{\rho,\mathcal{D}_k} (\boldsymbol{U}_{i-1,k})^{\alpha} \right] \leq d \, \max_{1 \leq s \leq d} C_{s,\alpha} \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[h_{\rho,\mathcal{D}_k} (\boldsymbol{U}_{i-1,k})^{\alpha} \right] \\
&\leq d \, C_\alpha \mathbb{E}_{\boldsymbol{U}_{i-1,k}} \left[h_{\rho,\mathcal{D}_k} (\boldsymbol{U}_{i-1,k})^{\alpha} \right].
\end{aligned}$$

Overall, we have the following bound for $e_{i,k}$

$$\begin{aligned} e_{i,k} \leq & 4L_{\mathscr{G}}^2\, e_{i-1,k} + 4L_c^2\, e_{i-1,k} + 4d\, C_\alpha \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i-1,k})^\alpha\right] \\ & + 4d\, C_\alpha \|f_c\|_{\infty,\mathcal{H}_K}^2 \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i-1,k})^\alpha\right], \quad i = 1,\dots,N, \;\; k \in \mathbb{N}. \end{aligned}$$

This upper bound can be written as the following recurrence

$$e_{i,k} \leq a\, e_{i-1,k} + b_{i-1,k}, \quad i = 1,\dots,N, \;\; k \in \mathbb{N},$$

with $a = 4\left(L_{\mathscr{G}}^2 + L_c^2\right)$ and

$$b_{i-1,k} = 4d\, C_\alpha (1 + \|f_c\|_{\infty,\mathcal{H}_K}^2) \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i-1,k})^\alpha\right]. \tag{40}$$

Unrolling the recurrence, we obtain

$$e_{i,k} = W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2 \leq \; a^i\, e_{0,k} \; + \; \sum_{j=1}^{i} a^{\,i-j}\, b_{j-1,k} \quad \text{for all } \; 1 \leq i \leq N, \;\; k \in \mathbb{N}.$$

By (36),

$$e_{0,k} = \mathbb{E}_{\boldsymbol{U}_{0,k}}\left[\|\boldsymbol{u}(t_0) - \boldsymbol{U}_{0,k}\|^2\right],$$

and since $\mathbb{E}[\boldsymbol{U}_{0,k}] = \boldsymbol{u}_{(0)} = \boldsymbol{u}(t_0)$ and $\mathrm{Var}(\boldsymbol{U}_{0,k}) = \Sigma_{0,k}$ by assumption,

$$e_{0,k} = \mathbb{E}_{\boldsymbol{U}_{0,k}}\left[\|\boldsymbol{U}_{0,k} - \mathbb{E}[\boldsymbol{U}_{0,k}]\|^2\right] = \mathrm{Tr}(\mathrm{Var}(\boldsymbol{U}_{0,k})) = \mathrm{tr}(\Sigma_{0,k}).$$

For the remaining cases, the structure of the proof is unchanged, with minor modifications, which we briefly point out below.

Case (i) Differentiability. The differentiability case differs from the infinite smoothness one in that Theorem 25 does not apply, and is replaced by Assumption 11. In fact, under the conditions of Assumption 11, the W and Q terms are bounded using

$$\sigma_{\mathcal{D}_k}^{(s)}(\boldsymbol{u}')^2 \leq C_{\beta,s} h_{\rho,\mathcal{D}_k}(\boldsymbol{u}')^\beta, \quad \text{for } \; s = 1,\dots,d.$$

This leads to a similar expression for $b_{i-1,k}$ in (40), differing in the constant and the exponents of the fill distance:

$$b_{i-1,k} = 4d\, C_\beta \left(1 + \|f_c\|_{\infty,\mathcal{H}_K}^2\right) \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i-1,k})^\beta\right],$$

with $C_\beta = \max_{1\leq s\leq d} C_{\beta,s}$ with $\{C_{\beta,s}\}_{s=1}^d$, $C_{\beta,s} > 0$, as in Assumption 11.

Case (ii) Sobolev norm-equivalence. The only change required is the application of part **(i)** of Theorem 25, instead of part **(ii)** as for the infinite smoothness case. This leads to a different constant in the $b_{i-1,k}$ expression, and different exponent of the fill distance, namely

$$b_{i-1,k} = 4d\, C \left(1 + \|f_c\|_{\infty,W_2^q}^2\right) \mathbb{E}_{\boldsymbol{U}_{i-1,k}}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i-1,k})^{2q-d}\right],$$

where $C = \max_{1\leq s\leq d} C_s$ with $\{C_s\}_{s=1}^d$, $C_s > 0$, defined in part **(i)** of Theorem 25. ■

B.3 Proof of Theorem 19

Similarly to the proof of Theorem 13 in Appendix B.2, here we provide the proof of Theorem 19 for the infinite smoothness case, part **(iii)**. Parts **(i)** and **(ii)** require minor modifications, which are detailed at the end of the section.

Case (iii) Smoothness. Our goal is to upper bound the coordinate-wise variances of the Prob-GPararеal solution $\boldsymbol{U}_{i+1,k}$ for all $i = 1, \dots, N$ and $k \in \mathbb{N}$, i.e., the diagonal elements of the covariance matrix $\mathrm{Var}(\boldsymbol{U}_{\mathrm{i+1,k}})$, which we denote $\sigma^{(s),2}_{i+1,k} := (\mathrm{Var}(\boldsymbol{U}_{i+1,k}))^{(s)}, s = 1, \dots, d$, where for a matrix S of dimension d, we write $S^{(s)}$ to denote the sth entry of its main diagonal. Then, we have

$$\begin{aligned}\sigma^{(s),2}_{i+1,k} &= \left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\mathrm{Var}_{\boldsymbol{U}_{i+1,k}|\boldsymbol{U}_{i,k}}\left(\boldsymbol{U}_{i+1,k}\right)\right]\right)^{(s)} + \left(\mathrm{Var}_{\boldsymbol{U}_{i,k}}\left(\mathbb{E}_{\boldsymbol{U}_{i+1,k}|\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i+1,k}\right]\right)\right)^{(s)}, \\ &= \left(\mathbb{E}_{\boldsymbol{U}_{i,k}}[\Sigma_{\mathcal{D}_k}(\boldsymbol{U}_{i,k})]\right)^{(s)} + \left(\mathrm{Var}_{\boldsymbol{U}_{i,k}}\left((\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})(\boldsymbol{U}_{i,k})\right)\right)^{(s)}, \end{aligned} \tag{41}$$

where we used the law of total covariance in the first equality, and the Prob-GPararеal update rule (9) given by $\boldsymbol{U}_{i+1,k} = \mathscr{G}(\boldsymbol{U}_{i,k}) + \boldsymbol{Z}_{i+1,k},\ i = 1, \dots, N,\ k \in \mathbb{N}$, with $\boldsymbol{Z}_{i+1,k}|\boldsymbol{U}_{i,k} \sim \mathcal{N}_d\left(\boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{U}_{i,k}), \Sigma_{\mathcal{D}_k}(\boldsymbol{U}_{i,k})\right)$, see (10) in the second equality.

By applying part **(ii)** in Theorem 25 on the first term on (41), we obtain the following bound for $s = 1, \dots, d$:

$$\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}[\Sigma_{\mathcal{D}_k}(\boldsymbol{U}_{i,k})]\right)^{(s)} = \sigma^{(s)}_{\mathcal{D}_k}(\boldsymbol{U}_{i,k})^2 \leq C_{\alpha,s}\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^{\alpha}\right]. \tag{42}$$

Denoting $g := \mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k}$, using the definition of variance, the second term on (41) becomes

$$\begin{aligned}(\mathrm{Var}_{\boldsymbol{U}_{i,k}}\left(g(\boldsymbol{U}_{i,k})\right))^{(s)} &= \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}(\boldsymbol{U}_{i,k}) - \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[g^{(s)}(\boldsymbol{U}_{i,k})\right]\right)^2\right] \\ &= \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}(\boldsymbol{U}_{i,k}) \mp g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right) - \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[g^{(s)}(\boldsymbol{U}_{i,k})\right]\right)^2\right] \\ &= \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}(\boldsymbol{U}_{i,k}) - g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right)\right)^2\right] \\ &\quad + \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right) - \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[g^{(s)}(\boldsymbol{U}_{i,k})\right]\right)^2\right] \\ &\quad + 2\,\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}(\boldsymbol{U}_{i,k}) - g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right)\right)\left(g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right) - \mathbb{E}\left[g^{(s)}(\boldsymbol{U}_{i,k})\right]\right)\right] \\ &= \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}(\boldsymbol{U}_{i,k}) - g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right)\right)^2\right] \\ &\quad - \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right) - \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[g^{(s)}(\boldsymbol{U}_{i,k})\right]\right)^2\right] \\ &\leq \mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\left(g^{(s)}(\boldsymbol{U}_{i,k}) - g^{(s)}\left(\mathbb{E}_{\boldsymbol{U}_{i,k}}\left[\boldsymbol{U}_{i,k}\right]\right)\right)^2\right].\end{aligned}$$

In this derivation, we added and subtracted an intermediate term (second equality), calculated the square (third equality), and used the fact that the second multiplier in the cross-term

is deterministic (fourth equality). Next, by replacing g with its original definition on the right-hand side, and using inequality (39), we obtain

$$(\mathrm{Var}\left(g(\boldsymbol{U}_{i,k})\right))^{(s)} \leq 2\underbrace{\mathbb{E}\left[\left(\mathscr{G}^{(s)}(\boldsymbol{U}_{i,k}) - \mathscr{G}^{(s)}(\mathbb{E}[\boldsymbol{U}_{i,k}])\right)^2\right]}_{\text{Term } T_1} + 2\underbrace{\mathbb{E}\left[\left(\boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\boldsymbol{U}_{i,k}) - \boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\mathbb{E}[\boldsymbol{U}_{i,k}])\right)^2\right]}_{\text{Term } T_2}.$$

Assumption 7 implies

$$T_1 \leq L^2_{\mathscr{G}}\, \mathbb{E}\left[\|\boldsymbol{U}_{i,k} - \mathbb{E}[\boldsymbol{U}_{i,k}]\|^2\right] = L^2_{\mathscr{G}}\, \mathrm{Tr}(\mathrm{Var}(\boldsymbol{U}_{i,k})) = L^2_{\mathscr{G}} \sum_{s=1}^{d} \sigma^{(s),2}_{i,k}. \tag{43}$$

For the T_2 term, we add and subtract intermediate terms $f_c^{(s)}(\boldsymbol{U}_{i,k})$ and $f_c^{(s)}(\mathbb{E}[\boldsymbol{U}_{i,k}])$, and obtain

$$\begin{aligned} T_2 &= \mathbb{E}\Big[\Big(\boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\boldsymbol{U}_{i,k}) \pm f_c^{(s)}(\boldsymbol{U}_{i,k}) \mp f_c^{(s)}(\mathbb{E}[\boldsymbol{U}_{i,k}]) - \boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\mathbb{E}[\boldsymbol{U}_{i,k}])\Big)^2\Big] \\ &\leq 3\mathbb{E}\left[\left(\boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\boldsymbol{U}_{i,k}) - f_c^{(s)}(\boldsymbol{U}_{i,k})\right)^2\right] + 3\left(f_c^{(s)}(\mathbb{E}[\boldsymbol{U}_{i,k}]) - \boldsymbol{\mu}^{(s)}_{\mathcal{D}_k}(\mathbb{E}[\boldsymbol{U}_{i,k}])\right)^2 \\ &\quad + 3\mathbb{E}\left[\left(f_c^{(s)}(\boldsymbol{U}_{i,k}) - f_c^{(s)}(\mathbb{E}[\boldsymbol{U}_{i,k}])\right)^2\right], \end{aligned}$$

where the last inequality follows again from (39). Using Theorem 27 for the first and the second summand, and the fact that Assumption 6 holds by Remark 9 under Assumption 5 for the third one, we get

$$\begin{aligned} T_2 \leq& 3\mathbb{E}\left[\sigma^{(s)}_{\mathcal{D}_k}(\boldsymbol{U}_{i,k})^2\right] \|f_c^{(s)}\|^2_{\mathcal{H}_K} + 3\sigma^{(s)}_{\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^2 \|f_c^{(s)}\|^2_{\mathcal{H}_K} + 3L_c^2\mathbb{E}\left[\|\boldsymbol{U}_{i,k} - \mathbb{E}[\boldsymbol{U}_{i,k}]\|^2\right] \\ \leq& 3C_{\alpha,s}\|f_c^{(s)}\|^2_{\mathcal{H}_K}\left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\alpha\right] + h_{\rho,\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^\alpha\right\} + 3L_c^2 \sum_{s=1}^{d} \sigma^{(s),2}_{i,k}, \end{aligned}$$

where in the last inequality we used again part **(ii)** in Theorem 25 and analogous derivation to the one in (43).

Finally, using (42) for the first summand on the right-hand side of (41), and terms T_1 and T_2 for the second summand, we obtain for (41)

$$\begin{aligned} \sigma^{(s),2}_{i+1,k} \leq& C_{\alpha,s}\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\alpha\right] + 2L^2_{\mathscr{G}} \sum_{s=1}^{d} \sigma^{(s),2}_{i,k} \\ &+ 6C_{\alpha,s}\|f_c^{(s)}\|^2_{\mathcal{H}_K}\left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\alpha\right] + h_{\rho,\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^\alpha\right\} + 6L_c^2 \sum_{s=1}^{d} \sigma^{(s),2}_{i,k}. \end{aligned}$$

Maximizing the variance across all components on both sides of the inequality, we obtain a recursion for $\sigma^{\max,2}_{i+1,k}$, defined in (24), as:

$$\sigma^{\max,2}_{i+1,k} \leq a\, \sigma^{\max,2}_{i,k} + b_{i,k}, \quad i = 0, \ldots, N-1, \;\; k \in \mathbb{N}, \tag{44}$$

where $a = 2\,d(L_{\mathscr{G}}^2 + 3L_c^2)$ and

$$b_{i,k} = C_\alpha \mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\alpha\right] + 6C_\alpha \|f_c\|_{\infty,\mathcal{H}_K}^2 \left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\alpha\right] + h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{i,k})^\alpha\right\},$$

with $C_\alpha = \max_{1\le s\le d} C_{\alpha,s}$. Unrolling the recurrence and re-indexing the sum, we obtain

$$\sigma_{i,k}^{\max,2} \le a^i\, \sigma_{0,k}^{\max,2} + \sum_{j=1}^{i} a^{\,i-j}\, b_{j-1,k}, \quad i = 1,\dots,N,\ \ k \in \mathbb{N},$$

as required.

We now cover the remaining cases.

Case (i) Differentiability. As seen in Appendix B.2, the changes from the infinite smoothness case involve updating constant and exponents of the fill distance whenever either Theorem 25 was applied (note that Theorem 27 depends on Theorem 25), which is now replaced by Assumption 11, or by Assumption 6, which holds automatically with $L_c \le \sqrt{d}\,(c_1 + C_R\,\Delta t)\,\Delta t^{p+1}$ as in Remark 9. In particular, using Assumption 11 (instead of Theorem 25 in (42)), we obtain

$$(\mathbb{E}[\Sigma_{\mathcal{D}_k}(\boldsymbol{U}_{i,k})])^{(s)} = \sigma_{\mathcal{D}_k}^{(s)}(\boldsymbol{U}_{i,k})^2 \le C_{\beta,s}\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\beta\right]. \tag{45}$$

The other change is in the derivation of the T_2 term, which relied on both Assumptions 6 and 11. Similarly to what shown in Appendix B.2, only the constant and the exponents change, yielding

$$T_2 \le 3C_{\beta,s}\|f_c^{(s)}\|_{\mathcal{H}_K}^2 \left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\beta\right] + h_{\rho,\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^\beta\right\} + 3L_c^2 \sum_{s=1}^{d} \sigma_{i,k}^{(s),2}.$$

Overall, we obtain that there exists constant $C_\beta > 0$ such that

$$b_{i,k} = C_\beta \mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\beta\right] + 6C_\beta \|f_c\|_{\infty,\mathcal{H}_K}^2 \left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^\beta\right] + h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{i,k})^\beta\right\},$$

where $C_\beta = \max_{1\le s\le d} C_{\beta,s}$ with $\{C_{\beta,s}\}_{s=1}^d$, $C_{\beta,s} > 0$, as in Assumption 11.

Case (ii) Sobolev norm-equivalence. The only change required here is the application of part **(i)** of Theorem 25 instead of part **(ii)** as for the infinite smoothness case. This leads to a different constant in the $b_{i,k}$ expression, and different exponent of the fill distance

$$b_{i,k} = C\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^{2q-d}\right] + 6C\|f_c\|_{\infty,W_2^q}^2 \left\{\mathbb{E}\left[h_{\rho,\mathcal{D}_k}(\boldsymbol{U}_{i,k})^{2q-d}\right] + h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{i,k})^{2q-d}\right\},$$

where $C = \max_{1\le s\le d} C_s$ with $\{C_s\}_{s=1}^d$, $C_s > 0$, defined in part **(i)** of Theorem 25. ■

B.4 Proof of Theorem 21

We provide the proof of Theorem 21, part **(iii)**. Parts **(i)** and **(ii)** follow the same strategy as detailed in Appendix B.3, and are therefore omitted. Explicit values for the coefficients a and $b_{i,k}$ for these cases are given in Theorem 21. The proof for Corollary 23 is provided at the end of the section.

Case (iii) Smoothness. Although the following derivations are demonstrated for some fixed interval i and iteration k, they hold true for any $i \in \{1, \dots, N\}$ and $k \in \mathbb{N}$. We start by recalling that, according to the introduced notation,

$$\left\|\boldsymbol{\mu}_{i+1,k} - \boldsymbol{u}_{i+1,k}^{\mathrm{GPara}}\right\| = \left\|\mathbb{E}\left[\boldsymbol{U}_{i+1,k}\right] - \left(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k}\right)\left(\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\right)\right\|. \tag{46}$$

Using the law of total expectation and the Prob-GParareal update rule (9)-(10), we obtain

$$\mathbb{E}[\boldsymbol{U}_{i+1,k}] = \mathbb{E}_{\boldsymbol{U}_{i,k}}[\mathbb{E}_{\boldsymbol{U}_{i+1,k}|\boldsymbol{U}_{i,k}}[\boldsymbol{U}_{i+1,k}]] = \mathbb{E}[(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})(\boldsymbol{U}_{i,k})],$$

By the hypotheses of the theorem, $(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k}) \in C^2$, so we expand it around $\mathbb{E}\left[\boldsymbol{U}_{i,k}\right] \in \mathbb{R}^d$ using the second-order Taylor expansion as follows:

$$(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})(\boldsymbol{U}_{i,k}) = \left(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k}\right)\left(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right) + J_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})}\left(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right)\left(\boldsymbol{U}_{i,k} - \mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right) + \boldsymbol{R}_{i,k}. \tag{47}$$

In this expansion, $J_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})}$ is the Jacobian matrix of $(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})$ and $\boldsymbol{R}_{i,k} \in \mathbb{R}^d$ is defined as

$$\boldsymbol{R}_{i,k}^{(s)} = \frac{1}{2}\left(\boldsymbol{U}_{i,k} - \mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right)^\top H_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}}(\boldsymbol{\zeta}_{i,k})\left(\boldsymbol{U}_{i,k} - \mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right), \quad s = 1, \dots d,$$

where $H_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}}$ denotes the Hessian matrix of $(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}$ evaluated at $\boldsymbol{\zeta}_{i,k} \in \mathbb{R}^d$ on the line segment between $\boldsymbol{U}_{i,k}$ and $\mathbb{E}\left[\boldsymbol{U}_{i,k}\right]$.

By taking expectation with respect to $\boldsymbol{U}_{i,k}$ on both sides of (47), the term involving the Jacobian vanishes and one writes

$$\mathbb{E}\left[(\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})(\boldsymbol{U}_{i,k})\right] = (\mathscr{G} + \boldsymbol{\mu}_{\mathcal{D}_k})(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right]) + \mathbb{E}\left[\boldsymbol{R}_{i,k}\right]. \tag{48}$$

We now further analyze the remainder $\boldsymbol{R}_{i,k}$:

$$\begin{aligned}
\left|\mathbb{E}\left[\boldsymbol{R}_{i,k}^{(s)}\right]\right| &\leq \mathbb{E}\left[\left|\boldsymbol{R}_{i,k}^{(s)}\right|\right] = \frac{1}{2}\mathbb{E}\left[\left|\left(\boldsymbol{U}_{i,k} - \mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right)^\top H_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}}(\boldsymbol{\zeta}_{i,k})\left(\boldsymbol{U}_{i,k} - \mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right)\right|\right] \\
&\leq \frac{1}{2}\mathbb{E}\left[\left\|H_{(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})^{(s)}}(\boldsymbol{\zeta}_{i,k})\right\|\left\|\boldsymbol{U}_{i,k} - \mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right\|^2\right] \leq \frac{M_s}{2}\,\mathbb{E}\left[\left\|\boldsymbol{U}_{i,k} - \mathbb{E}\left[\boldsymbol{U}_{i,k}\right]\right\|^2\right] \\
&= \frac{M_s}{2}\,\mathrm{Tr}\left(\mathrm{Var}\left(\boldsymbol{U}_{i,k}\right)\right) \leq \frac{M_s\, d}{2}\,\sigma_{i,k}^{\max,2},
\end{aligned} \tag{49}$$

for all $s = 1, \dots, d$, where we used that $|\boldsymbol{x}^\top A \boldsymbol{x}| \leq \|A\|\,\|\boldsymbol{x}\|^2$ for any $\boldsymbol{x} \in \mathbb{R}^d$ and symmetric matrix A (second inequality), the assumption on the boundedness of the spectral norm of the Hessian (third inequality), and the definition of variance and (24) (last inequality).

We now obtain

$$\begin{aligned}
\left\|\mathbb{E}\left[\boldsymbol{R}_{i,k}\right]\right\| &\leq \sqrt{d}\left\|\mathbb{E}\left[\boldsymbol{R}_{i,k}\right]\right\|_\infty \leq \sqrt{d}\max_{1\leq s\leq d}\left|\mathbb{E}\left[\boldsymbol{R}_{i,k}^{(s)}\right]\right| \\
&\leq \sqrt{d}\max_{1\leq s\leq d}\frac{M_s d}{2}\sigma_{i,k}^{\max,2} = \frac{M d^{3/2}}{2}\sigma_{i,k}^{\max,2},
\end{aligned} \tag{50}$$

where we use that $M = \max_{1\leq s\leq d} M_s$.

Using (48) and (50) in (46), we obtain

$$\begin{aligned}\left\|\boldsymbol{\mu}_{i+1,k}-\boldsymbol{u}_{i+1,k}^{\mathrm{GPara}}\right\| &\leq \left\|(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-\left(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k}\right)\left(\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\right)\right\|+\|\mathbb{E}[\boldsymbol{R}_{i,k}]\| \\ &\leq \left\|(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k})(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-\left(\mathscr{G}+\boldsymbol{\mu}_{\mathcal{D}_k}\right)\left(\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\right)\right\|+\frac{M\,d^{3/2}}{2}\,\sigma_{i,k}^{\max,2} \\ &\leq \left\|\mathscr{G}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-\mathscr{G}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})\right\|+\left\|\boldsymbol{\mu}_{\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-\boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})\right\| \\ &\quad+\frac{M\,d^{3/2}}{2}\sigma_{i,k}^{\max,2}, \end{aligned} \tag{51}$$

where we used the triangle inequality in the second and third inequalities. For the second summand on the right-hand side of (51) we can use the same approach as for term T_2 in the proof of Theorem 19 (see Appendix B.3), that is

$$\begin{aligned}&\|\boldsymbol{\mu}_{\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-\boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})\| \\ &\leq \|\boldsymbol{\mu}_{\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-f_c(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])\|+\|f_c(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})-\boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})\|+\|f_c(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-f_c(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})\|.\end{aligned}$$

We now notice that by Theorem 27 one can bound the first summand on the right-hand side of this inequality. More precisely,

$$\begin{aligned}\|\boldsymbol{\mu}_{\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-f_c(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])\|^2 &\leq \sum_{s=1}^{d}|\boldsymbol{\mu}_{\mathcal{D}_k}^{(s)}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-f_c^{(s)}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])|^2 \\ &\leq \|f_c\|_{\infty,\mathcal{H}_K}^2\sum_{s=1}^{d}\sigma_{\mathcal{D}_k}^{(s)}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^2 \leq \|f_c\|_{\infty,\mathcal{H}_K}^2\sum_{s=1}^{d}C_{\alpha,s}h_{\rho,\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^{\alpha} \\ &\leq \|f_c\|_{\infty,\mathcal{H}_K}^2 d C_\alpha h_{\rho,\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^{\alpha},\end{aligned}$$

with $C_\alpha=\max_{1\leq s\leq d}C_{\alpha,s}$, and the second summand can be bounded analogously. For the third one, we use (16). We therefore write that

$$\begin{aligned}\|\boldsymbol{\mu}_{\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])-\boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})\| \leq& \sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K}h_{\rho,\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^{\alpha/2} \\ &+\sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K}h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})^{\alpha/2}+L_c\|\mathbb{E}\left[\boldsymbol{U}_{i,k}\right]-\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\|.\end{aligned}$$

Using this bound in (51), and noticing that the second summand on its right-hand side can be bounded using Assumption 7, we get

$$\begin{aligned}\left\|\boldsymbol{\mu}_{i+1,k}-\boldsymbol{u}_{i+1,k}^{\mathrm{GPara}}\right\| \leq& \frac{M\,d^{3/2}}{2}\,\sigma_{i,k}^{\max,2}+L_{\mathscr{G}}\|\mathbb{E}\left[\boldsymbol{U}_{i,k}\right]-\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\|+L_c\|\mathbb{E}\left[\boldsymbol{U}_{i,k}\right]-\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| \\ &+\sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K}\left(h_{\rho,\mathcal{D}_k}(\mathbb{E}\left[\boldsymbol{U}_{i,k}\right])^{\alpha/2}+h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})^{\alpha/2}\right) \\ \leq& \frac{M\,d^{3/2}}{2}\sigma_{i,k}^{\max,2}+(L_{\mathscr{G}}+L_c)\left\|\boldsymbol{\mu}_{i,k}-\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\right\| \\ &+\sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K}\left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{i,k})^{\alpha/2}+h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})^{\alpha/2}\right).\end{aligned}$$

Notice that this expression is a recursive relation which could be written more compactly as:

$$\|\boldsymbol{\mu}_{i+1,k}-\boldsymbol{u}_{i+1,k}^{\mathrm{GPara}}\|\leq a\|\boldsymbol{\mu}_{i,k}-\boldsymbol{u}_{i,k}^{\mathrm{GPara}}\|+b_{i,k},\quad i=1,\dots,N,\ \ k\in\mathbb{N},$$

where $a = L_{\mathscr{G}} + L_c$ and

$$b_{i,k} = \frac{M\, d^{3/2}}{2}\, \sigma_{i,k}^{\max,2} + \sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{i,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{i,k}^{\mathrm{GPara}})^{\alpha/2} \right).$$

Unrolling the recurrence from $\|\boldsymbol{\mu}_{0,k} - \boldsymbol{u}_{0,k}^{\mathrm{GPara}}\| = 0$ and re-indexing, we obtain

$$\|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| \leq \sum_{j=1}^{i} a^{i-j} b_{j-1,k} \quad \text{for all } 1 \leq i \leq N,\ \ k \in \mathbb{N},$$

which proves the claim. ■

Proof of Corollary 23. The proof is given for case **(iii)**. Cases **(i)** and **(ii)** can be proven analogously. Let a and $b_{i,k}$ be defined as in (25) and in part **(iii)** in Theorem 19, respectively. We use $\tilde{a}$ to denote a defined in (28) and $\tilde{b}_{i,k}$ to denote $b_{i,k}$ in part **(iii)** of Theorem 21, respectively. For convenience, we recall the latter below

$$\tilde{b}_{l,k} = \frac{M\, d^{3/2}}{2}\, \sigma_{l,k}^{\max,2} + \sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{l,k}^{\mathrm{GPara}})^{\alpha/2} \right),$$

with $l = 0, \ldots, N-1$, $k \in \mathbb{N}$.

Next, we substitute into this expression the variance bound (26) in Theorem 19, namely,

$$\sigma_{l,k}^{\max,2} \leq a^l\, \sigma_{0,k}^{\max,2} + \sum_{j=1}^{l} a^{\,l-j}\, b_{j-1,k},$$

and obtain the following inequality:

$$\begin{aligned}\tilde{b}_{l,k} \leq & \frac{M\, d^{3/2}}{2} \Big(a^l\, \sigma_{0,k}^{\max,2} \ + \ \sum_{j=1}^{l} a^{\,l-j}\, b_{j-1,k} \Big) \\ & + \sqrt{C_\alpha d}\|f_c\|_{\infty,\mathcal{H}_K} \left(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{l,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{l,k}^{\mathrm{GPara}})^{\alpha/2} \right).\end{aligned}$$

Using this expression, we now rewrite the mean error bound (29) in Theorem 21 as follows:

$$\begin{aligned}\|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| &\leq \sum_{j=1}^{i} \tilde{a}^{\,i-j}\, \tilde{b}_{j-1,k} \\ &\leq \sum_{j=1}^{i} \tilde{a}^{\,i-j} \Big[\frac{M\, d^{3/2}}{2} \Big(a^{j-1}\, \sigma_{0,k}^{\max,2} + \sum_{m=1}^{j-1} a^{\,j-1-m}\, b_{m-1,k} \Big) \\ &\quad + \sqrt{C_\alpha d}\, \|f_c\|_{\infty,\mathcal{H}_K} \big(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{j-1,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{j-1,k}^{\mathrm{GPara}})^{\alpha/2} \big) \Big] \\ &= \frac{M\, d^{3/2}}{2} \left(\underbrace{\sigma_{0,k}^{\max,2} \sum_{j=1}^{i} \tilde{a}^{\,i-j} a^{j-1}}_{\text{Term } T_1} + \underbrace{\sum_{j=1}^{i} \sum_{m=1}^{j-1} \tilde{a}^{\,i-j} a^{\,j-1-m}\, b_{m-1,k}}_{\text{Term } T_2} \right) \\ &\quad + \sqrt{C_\alpha d}\, \|f_c\|_{\infty,\mathcal{H}_K} \sum_{j=1}^{i} \tilde{a}^{\,i-j} \big(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{j-1,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{j-1,k}^{\mathrm{GPara}})^{\alpha/2} \big). \quad (52)\end{aligned}$$

Whenever $a \neq \tilde{a}$, we can further develop the terms T_1 and T_2 as follows. Term T_1 can be written as

$$T_1 = \sigma_{0,k}^{\max,2} \sum_{j=1}^{i} \tilde{a}^{\,i-j} a^{j-1} = \sigma_{0,k}^{\max,2}\, \tilde{a}^{i-1} \sum_{j=1}^{i} \Big(\frac{a}{\tilde{a}}\Big)^{j-1} = \sigma_{0,k}^{\max,2}\, \frac{\tilde{a}^i - a^i}{\tilde{a} - a}.$$

while for T_2 we get

$$T_2 = \sum_{m=1}^{i-1} b_{m-1,k} \sum_{j=m+1}^{i} \tilde{a}^{\,i-j} a^{\,j-1-m} = \frac{1}{\tilde{a} - a} \sum_{m=1}^{i-1} b_{m-1,k} \big(\tilde{a}^{\,i-m} - a^{\,i-m}\big).$$

Substituting T_1 and T_2 into (52) yields, for all $i = 1, \ldots, N$ and $k \in \mathbb{N}$,

$$\begin{aligned} \|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| \leq \frac{M\, d^{3/2}}{2(\tilde{a} - a)} \left(\sigma_{0,k}^{\max,2} \big(\tilde{a}^i - a^i\big) + \sum_{j=1}^{i-1} b_{j-1,k} \big(\tilde{a}^{\,i-j} - a^{\,i-j}\big) \right) \\ + \sqrt{C_\alpha d}\, \|f_c\|_{\infty,\mathcal{H}_K} \sum_{j=1}^{i} \tilde{a}^{\,i-j} \big(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{j-1,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{j-1,k}^{\mathrm{GPara}})^{\alpha/2} \big), \end{aligned}$$

as required.

It remains to consider the case $a = \tilde{a}$. Starting again from (52), the terms T_1 and T_2 can then be evaluated directly as

$$T_1 = \sigma_{0,k}^{\max,2} \sum_{j=1}^{i} a^{\,i-j} a^{j-1} = \sigma_{0,k}^{\max,2}\, i a^{i-1},$$

and

$$T_2 = \sum_{m=1}^{i-1} b_{m-1,k} \sum_{j=m+1}^{i} a^{\,i-j} a^{\,j-1-m} = \sum_{m=1}^{i-1} b_{m-1,k} (i-m) a^{\,i-m-1}.$$

Substituting these expressions into (52) gives

$$\begin{aligned} \|\boldsymbol{\mu}_{i,k} - \boldsymbol{u}_{i,k}^{\mathrm{GPara}}\| \leq \frac{M\, d^{3/2}}{2} \left(\sigma_{0,k}^{\max,2}\, i a^{i-1} + \sum_{j=1}^{i-1} b_{j-1,k} (i-j) a^{\,i-j-1} \right) \\ + \sqrt{C_\alpha d}\, \|f_c\|_{\infty,\mathcal{H}_K} \sum_{j=1}^{i} a^{\,i-j} \big(h_{\rho,\mathcal{D}_k}(\boldsymbol{\mu}_{j-1,k})^{\alpha/2} + h_{\rho,\mathcal{D}_k}(\boldsymbol{u}_{j-1,k}^{\mathrm{GPara}})^{\alpha/2} \big). \end{aligned}$$

Equivalently, the equality case is obtained from the expression for $a \neq \tilde{a}$ by replacing each factor

$$\frac{\tilde{a}^q - a^q}{\tilde{a} - a}$$

with its limiting value $q a^{q-1}$, for $q \geq 1$. ■

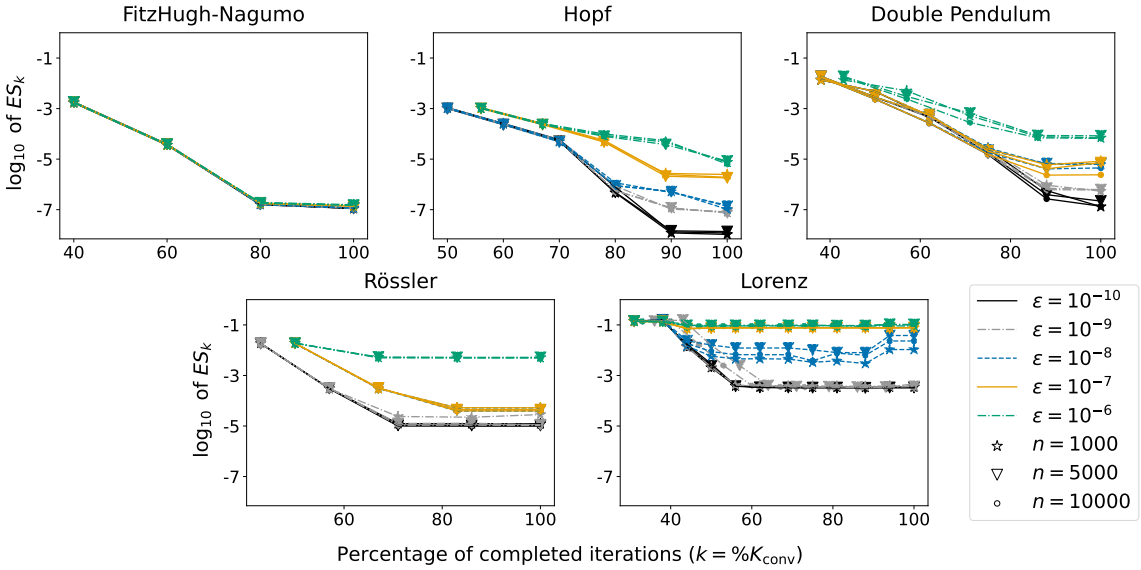

Figure 8: Impact of the tolerance level ϵ and the number of samples n on the Prob-GParareal performance (ES_k, in $\log_{10}$) across systems. The x-axes show the percentage of completed iterations relative to K_{conv}, the number of iterations to converge. All results are averaged over ten independent Prob-GParareal runs.

Appendix C. Robustness to n and effect of ϵ on the Prob-GParareal performance

In this section, we examine the impact of n (the number of draws used to represent the probabilistic solution at each interval i) and ϵ (the tolerance threshold used to establish convergence in (13)) on the accuracy (Figure 8), convergence (Figure 9) and runtime (Figure 10) of the Prob-GParareal algorithm. All results are averaged over ten independent runs using the same setup as in Section 5.3, with $\sigma_{\text{init}} = 0$. Smaller thresholds generally lead to more accurate results but slightly increased runtimes due to additional iterations required for convergence (results not shown), with an expected more notable impact for more expensive evaluations of $\mathscr{F}$.

Figure 8 highlights the impact of n and ϵ on the quality of the forecast. For clarity of presentation, we only report the energy score (ES), although similar patterns are observed for the other metrics. In this figure, the gray colors indicate different values of ϵ, while the marker symbols represent n. The most notable difference occurs in ϵ, with lower threshold values yielding better performance. The effect is system dependent, though: while FHN is unaffected by the choice of ϵ, Lorenz, being chaotic, demonstrates poor performance with $\epsilon = 10^{-7}$, as it fails to capture the distribution's characteristics adequately. Generally, the ϵ values around 10^{-7} or 10^{-8} provide sufficient performance for most cases. However, the lower is ϵ, the longer the algorithm may take to converge, as shown in Figure 9, with an impact on the runtime as well, see Figure 10. Although this effect is not particularly pronounced

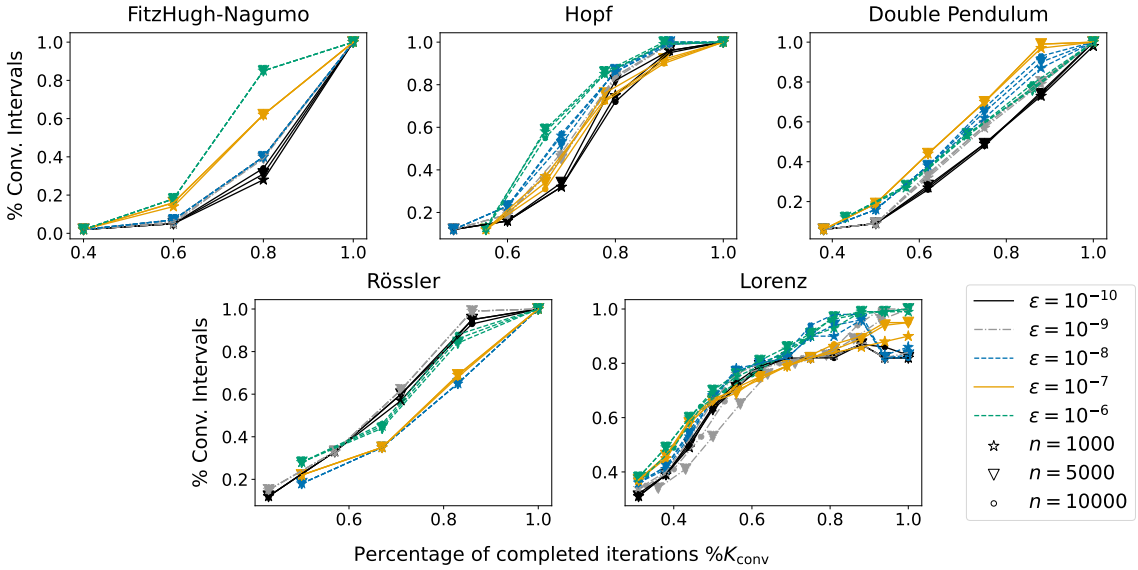

Figure 9: Impact of the tolerance level ϵ and the number of observations n on the Prob-GParareal convergence. The x-axis shows the percentage of completed iterations relative to K_{conv}, the iterations required to converge (generally unknown). The y-axis displays the percentage of converged intervals at iteration $k = \%K_{\text{conv}}$. All results are averaged over ten independent Prob-GParareal runs.

here, it may become significant in real-world applications with costly fine solvers, where each additional iteration is expensive. The impact of n on convergence is marginal, while it affects the algorithm runtime, as shown in Figure 10.

Appendix D. Fill distance analysis

In this section, we present empirical evidence on the evolution of the local fill distance, which was introduced in Section 4 to quantify dataset quality and establish bounds on the Wasserstein distance and on the variance of the solution.

To compute the fill distance at iteration k, we evaluate $h_{\rho,\mathcal{D}_k}(\boldsymbol{u}')$ at every point $\boldsymbol{u}' = \boldsymbol{u}_{i,k}^{(j)}$ for $i = 1, \dots, N$ and $j = 1, \dots, n$. Recall that for a constant $\rho > 0$, the local fill distance at $\boldsymbol{u}' \in \mathcal{U}$ is defined by

$$h_{\rho,\mathcal{D}}(\boldsymbol{u}') := \sup_{\boldsymbol{u} \in B_\rho(\boldsymbol{u}')} \min_{\boldsymbol{u}_i \in \mathcal{D}} \|\boldsymbol{u} - \boldsymbol{u}_i\|,$$

where $B_\rho(\boldsymbol{u}') \in \mathbb{R}^d$ denotes a ball of radius $\rho > 0$ around $\boldsymbol{u}'$, with ρ chosen as the smallest radius which ensures that the ball contains at least one observation. Rather than evaluating $h_{\rho,\mathcal{D}_k}$ at every $\boldsymbol{u}' \in \mathcal{U}$, we select representative points of the sample $\mathcal{U}_{i,k}$. In one-dimensional settings, quantiles provide a natural choice. For multivariate data, various generalizations of quantiles exist (see Cai (2010) and references therein). Here, we use the highest density regions (HDR, Hyndman 1996), defined as the smallest regions that contain $\alpha\%$ of the probability

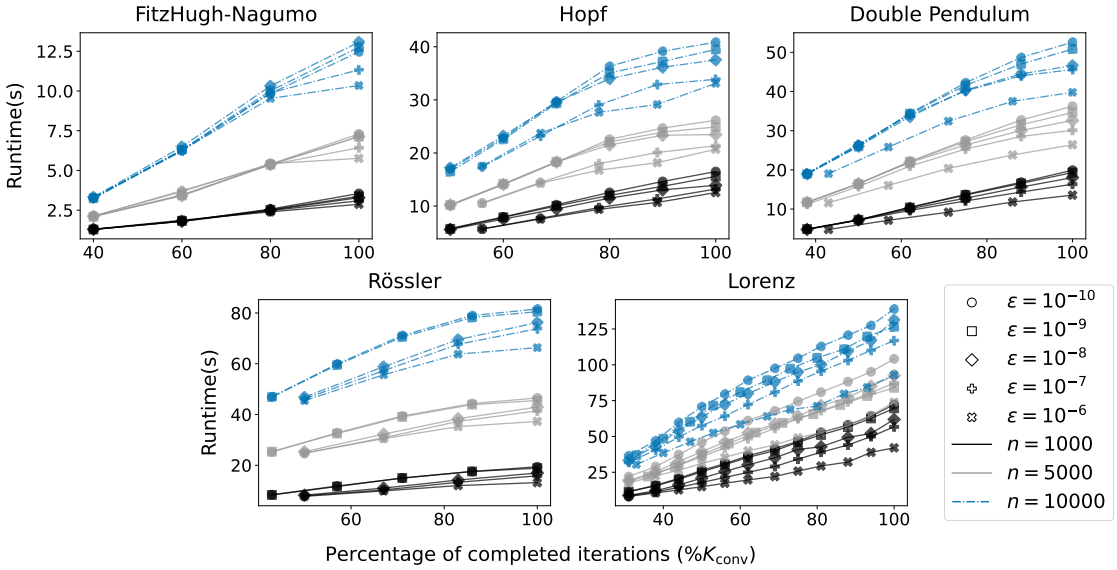

Figure 10: Impact of the tolerance level ϵ and the number of samples n on the empirical Prob-GParareal runtime. The x-axes show the percentage of completed iterations relative to K_{conv}, the number of iterations to converge. All results are averaged over ten independent Prob-GParareal runs.

mass. For example, for multivariate Gaussians, HDRs correspond to hyperellipsoids. For each interval i and α-HDR, we take two observations from $\mathcal{U}_{i,k}$ that lie on the HDR boundary or are closest to it in the Euclidean sense, evaluate the fill distance at these points, and then average them across intervals i. By repeating this process across iterations, we observe how the fill distance changes for points located near or far from the bulk of the data. Recall that the dataset at iteration k is updated with the observations $(\overline{\boldsymbol{u}}_{i,k-1}, f_c(\overline{\boldsymbol{u}}_{i,k-1}))$, $i = 1, \ldots, N$, so we expect the points closer to the mean $\overline{\boldsymbol{u}}_{i,k-1}$ to be better represented, with lower fill distances. This aligns with the empirical results shown in Figure 11, where we observe an exponentially fast decrease in the local fill distance over the iterations.

Appendix E. Impact of early termination on the algorithm runtime

In this section, we investigate the computational cost savings obtained by stopping Prob-GParareal prior to convergence. In Figure 12, we report the algorithm runtime (in seconds) as a function of the percentage of completed iterations to convergence ($\%K_{\text{conv}}$). Reducing the Prob-GParareal execution by even one iteration leads to one less parallel application of the fine solver $\mathscr{F}$, and fewer (nn)GP operations. Hence, the impact is system dependent, with more expensive $\mathscr{F}$ having the largest effect.

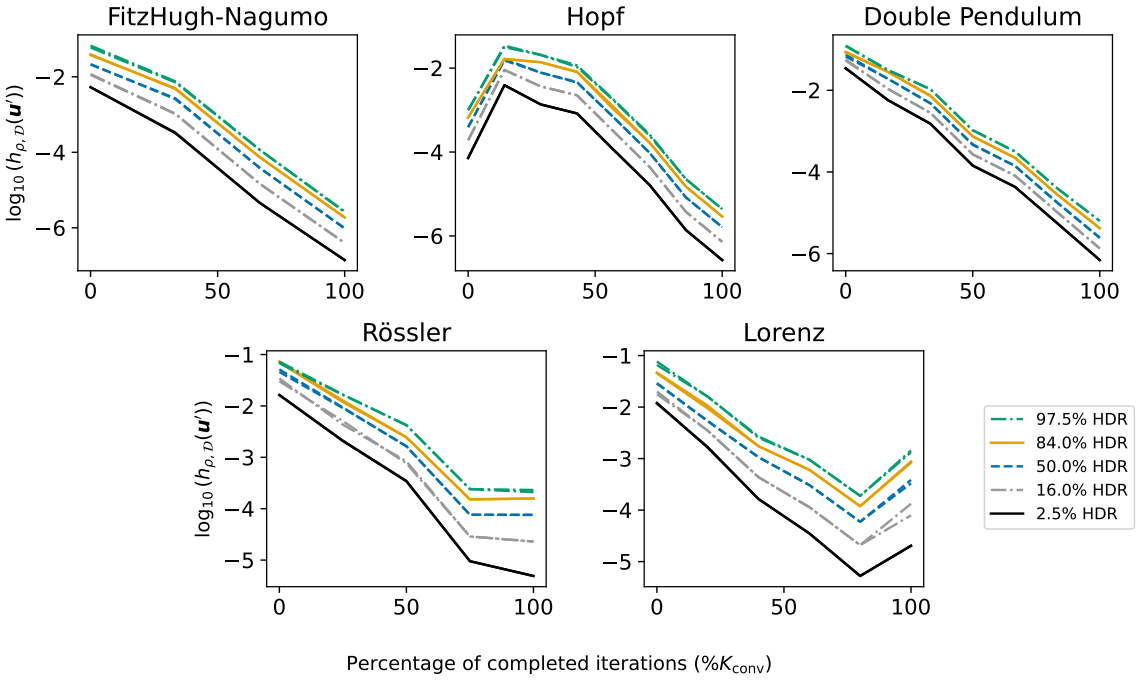

Figure 11: Evolution of the fill distance evaluated at representative observations placed on the boundary of the α-HDR. Fill distance values are averaged over intervals i and plotted as a function of the percentage of completed iterations for $\sigma_{\text{init}} = 0$.

Appendix F. Additional results on Prob-nnGPareareal

In Figure 13, we report the impact of random initial conditions $\boldsymbol{U}_{0,0,}$ on the coordinate-wise standard deviation of the converged Prob-nnGPareareal solution $\mathcal{U}_{i,K_{\text{conv}}}$ across intervals i for different systems. We sample the initial state as $\boldsymbol{U}_{0,0} \sim \mathcal{N}(\boldsymbol{u}_{(0)}, \sigma^2_{\text{init}}\mathbb{I}_d)$, where $\sigma_{\text{init}} \in \{0, e^{-2}, e^{-3}, e^{-4}, e^{-5}, e^{-6}\}$, is the standard deviation of all coordinates. This setup mimics the experiments in Figure 6, Section 5.4 for Prob-GPareareal, and is used here to assess the impact of switching from GPs to nnGPs. While in some cases the use of nnGP results in a slight increase in uncertainty over time compared to Prob-GPareareal, the overall algorithm performance remains largely unaffected.

Appendix G. Empirical results setup

In Table 3, we summarize the experimental setup used to obtain the results reported in the main text. The system definitions and parameters follow Gattiglio et al. (2025), while all implementation details, including solver configurations and evaluation scripts, are available in the accompanying code repository. Unless otherwise stated, Prob-GPareareal is run with a fixed number of samples $n = 5000$, and convergence is determined using the stopping criterion

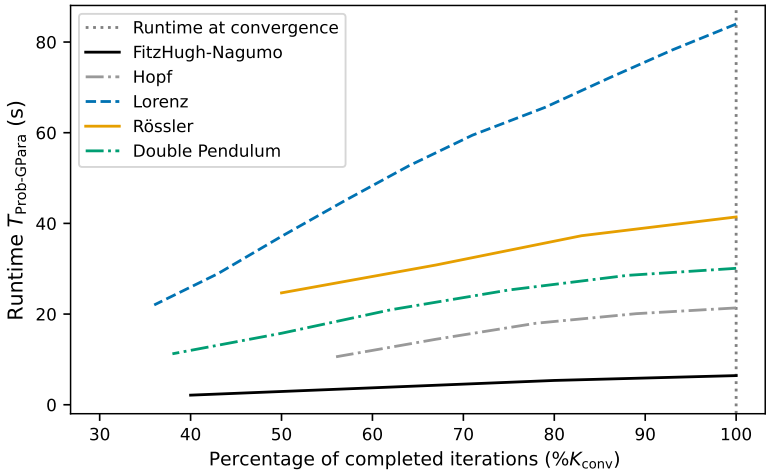

Figure 12: Effect of early termination of Prob-GParareal on its runtime, in seconds. All results are averaged over ten independent runs of the Prob-GParareal algorithm.

in (13) with tolerance $\epsilon = 10^{-8}$. All reported results are averaged over ten independent runs with different random seeds.

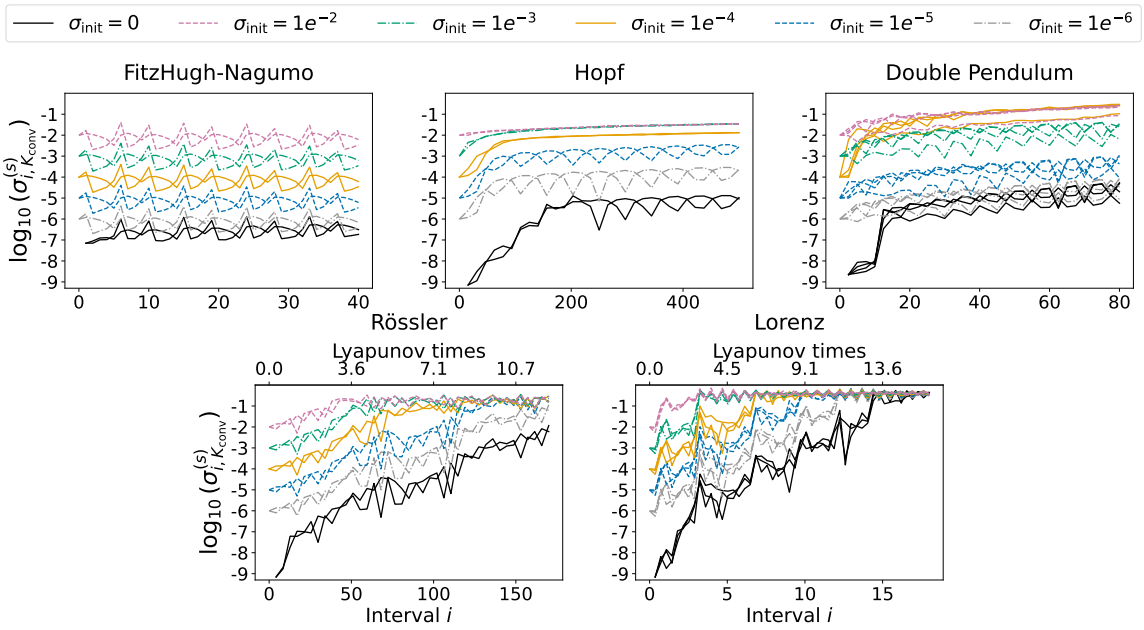

Figure 13: Impact of random initial conditions $\boldsymbol{U}_{0,0}$ on the coordinate-wise standard deviation of the converged Prob-nnGPareareal solution $\mathcal{U}_{i,K_{\text{conv}}}$ (with $m = 15$ nearest neighbors) across intervals i for different systems, using the same settings as Figure 6, to favor a comparison with the Prob-GPareareal results.

Algorithm 1 Prob-GPareareal Algorithm

Input: initial distribution $P_{\boldsymbol{U}_{0,0}}$, number of samples $n \in \mathbb{N}$, coarse solver $\mathscr{G}$, fine solver $\mathscr{F}$, tolerance $\epsilon > 0$, exit condition, statistic $g : (\mathbb{R}^d)^n \to \mathcal{X}$, distance metric $d_{\mathcal{U}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$.

Initialization of the Algorithm

Initialize the dataset $\mathcal{D}_0 = \emptyset$, the number of converged intervals $L = 0$, the iteration $k = 0$ and $K_{\text{conv}} = K_{\text{stop}} = 0$.
Sample n observations $\boldsymbol{u}_{0,0}^{(j)}$, $j = 1, \dots, n$ from $P_{\boldsymbol{U}_{0,0}}$, set $\mathcal{U}_{0,0} = \{\boldsymbol{u}_{0,0}^{(j)}\}_{j=1}^n$.
for $i = 1, \dots, N$ **do**
  for $j = 1, \dots, n$ **do** *(in parallel)*
    Compute $\boldsymbol{u}_{i,0}^{(j)} = \mathscr{G}(\boldsymbol{u}_{i-1,0}^{(j)})$.
  end for
  Set $\mathcal{U}_{i,0} = \{\boldsymbol{u}_{i,0}^{(j)}\}_{j=1}^n$.
end for

Execution of the recursive algorithm

while $L < N$ **and** exit condition not met **do**
  Set $k = k + 1$
  Set $\mathcal{U}_{i,k} = \mathcal{U}_{i,k-1}$ for $i = 0, \dots, L$.
  for $i = L, \dots, N-1$ **do** *(in parallel)*
    Compute the sample means $\overline{\boldsymbol{u}}_{i,k-1} = \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{u}_{i,k-1}^{(j)}$ for observation samples $\mathcal{U}_{i,k-1}$.
    Evaluate $\mathscr{F}(\overline{\boldsymbol{u}}_{i,k-1})$ and $\mathscr{G}(\overline{\boldsymbol{u}}_{i,k-1})$.
  end for
  Update the dataset $\mathcal{D}_k = \mathcal{D}_{k-1} \cup \{(\overline{\boldsymbol{u}}_{i,k-1}, f_c(\overline{\boldsymbol{u}}_{i,k-1}))\}_{i=L}^{N-1}$, with f_c defined in (5).
  Train the d scalar GPs on $\mathcal{D}_k$ to compute the posterior mean $\boldsymbol{\mu}_{\mathcal{D}_k}(\cdot)$ and covariance $\Sigma_{\mathcal{D}_k}(\cdot)$.
  for $i = L+1, \dots, N$ **do**
    for $j = 1, \dots, n$ **do** *(in parallel)*
      Draw $\boldsymbol{z}_{i,k}^{(j)}$ from $\mathcal{N}_d\left(\boldsymbol{\mu}_{\mathcal{D}_k}(\boldsymbol{u}_{i-1,k}^{(j)}), \Sigma_{\mathcal{D}_k}(\boldsymbol{u}_{i-1,k}^{(j)})\right)$, i.e. (10) conditioned on $\boldsymbol{U}_{i-1,k} = \boldsymbol{u}_{i-1,k}^{(j)}$.
      Compute $\boldsymbol{u}_{i,k}^{(j)} = \mathscr{G}(\boldsymbol{u}_{i-1,k}^{(j)}) + \boldsymbol{z}_{i,k}^{(j)}$ by the Prob-GPareareal predictor-corrector rule (9).
    end for
    Set $\mathcal{U}_{i,k} = \{\boldsymbol{u}_{i,k}^{(j)}\}_{j=1}^n$.
  end for
  for $i = L+1, \dots, N$ **do**
    if $d_{\mathcal{U}}(g(\mathcal{U}_{i,k}), g(\mathcal{U}_{i,k-1})) < \epsilon$ **then**
      Set $L = i$.
    else
      Break
    end if
  end for
end while
if $L = N$ **then**
  Set $K_{\text{end}} = K_{\text{conv}} = k$.
else
  Set $K_{\text{end}} = K_{\text{stop}} = k$.
end if

Output: Set of solution samples $\{\mathcal{U}_{i,K_{\text{end}}}\}_{i=0}^N$, $K_{\text{end}}, K_{\text{conv}}$, and K_{stop}.

System	d	System Parameters	$[t_0, t_N]$	N	K_{stop}	$\boldsymbol{u}_{(0)}$
FHN	2	$a = b = 0.2,\ c = 3$	$[0, 40]$	40	9	$(-1, 1)$
Rössler	3	$a = b = 0.2,\ c = 5.7$	$[0, 170]$	40	14	$(0, -6.78, 0.02)$
Double Pend.	4	None	$[0, 80]$	32	12	$(-0.5, 0, 0, 0)$
Hopf	2	None	$[-20, 500]$	32	12	$(0.1, 0.1)$
Lorenz	3	$(\gamma_1, \gamma_2, \gamma_3) = (10, 28, 8/3)$	$[0, 18]$	50	16	$(-15, -15, 20)$
Rössler Ext.	3	$a = b = 0.2,\ c = 5.7$	$[0, 340]$	40	14	$(0, -6.78, 0.02)$

System	$\mathscr{G}$	$\mathscr{F}$	$\mathscr{G}$ steps	$\mathscr{F}$ steps
FHN	RK2	RK4	160	$1.6e^5$
Rössler	RK1	RK4	$9e^4$	$4.5e^7$
Double Pend.	RK1	RK8	3104	$2.17e^5$
Hopf	RK1	RK8	2048	$1.7e^5$
Lorenz	RK4	RK4	$3e^2$	$2.25e^4$
Rössler Ext.	RK1	RK4	$9e^4$	$4.5e^7$

Table 3: Description of the simulation setup used to obtain the experimental results. The system equations and corresponding parameters are provided in Gattiglio et al. (2025). Here, d represents the system dimension, $[t_0, t_N]$ the evolution timespan $t \in [t_0, t_N]$, N the number of intervals, K_{stop} the maximum number of iterations before the execution is stopped, $\boldsymbol{u}_{(0)}$ the deterministic initial condition, and $\mathscr{F}$ and $\mathscr{G}$ are the fine and coarse solvers, respectively, with 'RKp' indicating a Runge-Kutta method of order p. Finally, '$\mathscr{F}$ steps' refers to the number of integration steps performed by the fine solver over $[t_0, t_N]$, where a higher count implies greater accuracy. The same applies for '$\mathscr{G}$ steps'. Rössler Ext. refers to the Rössler system over an extended solution timespan, namely $t \in [0, 340]$, twice as much as the previously considered value of $t_N = 170$. Additional implementation details, including the stopping criterion, number of samples, and random seeds, are provided in the accompanying code repository.

Appendix H. Empirical assessment of bound sharpness

In this section, we provide an empirical assessment of the bounds in Theorem 13 by comparing the Wasserstein error $W_2\big(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}}\big)^2$ with the local fill distance $h_{\rho,\mathcal{D}_k}$ appearing in the coefficients $b_{l,k}$. We consider the Double Pendulum system and evaluate both quantities across Parareal iterations. We focus on the Wasserstein error at the final time t_N, namely $W_2\big(\delta_{\boldsymbol{u}(t_N)}, P_{\boldsymbol{U}_{N,k}}\big)^2$. As shown in Figure 14 (left), this quantity decays rapidly with the iteration number k and closely tracks the maximum error across time intervals i. The reference solution $\boldsymbol{u}(t_i)$ is obtained by propagating the initial condition $\boldsymbol{u}_{(0)}$ with the fine solver $\mathcal{F}$. For each iteration k, the Prob-GParareal solution is represented by a particle approximation $P_{\boldsymbol{U}_{i,k}}$ with $n = 10^4$ samples. The quantity

$$W_2\big(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}}\big)^2 = \mathbb{E}\big[\|\boldsymbol{u}(t_i) - \boldsymbol{U}_{i,k}\|^2\big]$$

is estimated via Monte Carlo using the available particles. The local fill distance $h_{\rho,\mathcal{D}_k}$ is evaluated as described in Appendix D, averaging over quantiles. To directly assess the dependence on $h_{\rho,\mathcal{D}_k}$ predicted by the bound, Figure 14 (right) reports the empirical error $W_2\big(\delta_{\boldsymbol{u}(t_N)}, P_{\boldsymbol{U}_{N,k}}\big)^2$ as a function of the fill distance on a log-log scale. A power-law relationship of the form

$$W_2\big(\delta_{\boldsymbol{u}(t_N)}, P_{\boldsymbol{U}_{N,k}}\big)^2 \;\approx\; C h_{\rho,\mathcal{D}_k}^{\eta},$$

with an estimated exponent $\eta \approx 2$ is observed. Since we employ a Gaussian kernel, the relevant theoretical regime is case (iii) of Theorem 13. In this case, the bound is consistent with arbitrarily algebraic $h_{\rho,\mathcal{D}_k}^{\alpha}$ for $\alpha > 0$ and does not prescribe a specific exponent α. The observed value $\eta \approx 2$ should therefore be interpreted as an effective algebraic rate over the finite range of fill distances explored in the experiment, rather than as a sharp theoretical exponent. As expected for worst-case estimates, the theoretical constants are conservative and the bound is not intended to be tight in magnitude. Nevertheless, the experiment supports the interpretation that the convergence of Prob-GParareal is governed by the decay of the local fill distance.

Appendix I. Empirical validation of the Prob-GParareal computational complexity

In this section, we provide an empirical validation of the Prob-(nn)GParareal computational complexity derived in Section 3.3. To do so, we extend the runtime comparison in Figure 10 by adding, for each system, the runtime predicted by the cost model for a representative configuration with $\epsilon = 10^{-8}$ and $n = 5000$. The theoretical curves are obtained by evaluating the expression in Section 3.3 at the corresponding values of N, n, and K_{conv}, up to proportionality constants estimated from the data.

Figure 15 compares the resulting theoretical runtime with the empirical wall-clock time averaged over ten independent runs. Overall, the model captures the observed trends across systems and reproduces the increase in computational cost as the number of completed iterations grows. The small irregularities visible in some theoretical curves reflect the discrete structure of the cost model, in particular its dependence on the number of converged intervals and hence on the number of coarse solver evaluations, together with the nonlinear

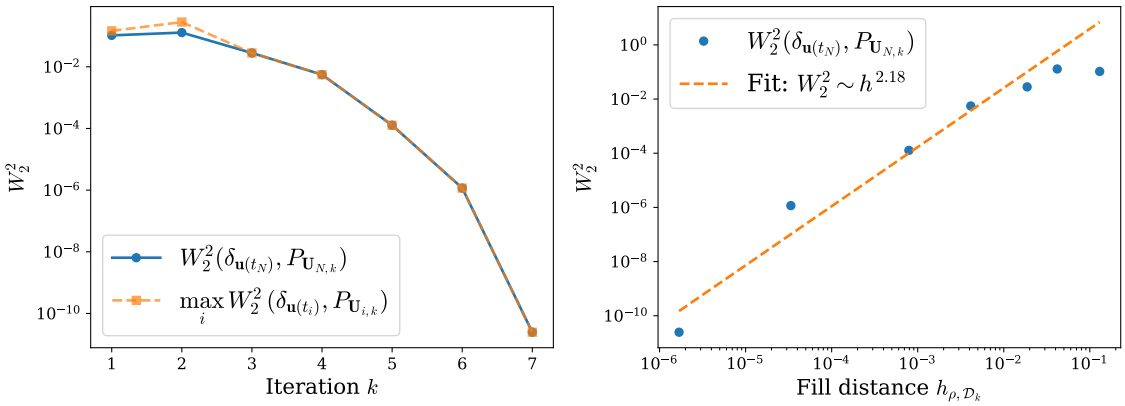

Figure 14: Left: decay of the empirical Wasserstein error $W_2(\delta_{\boldsymbol{u}(t_N)}, P_{\boldsymbol{U}_{N,k}})^2$ as a function of the iteration number k, together with the maximum error over time $\max_i W_2(\delta_{\boldsymbol{u}(t_i)}, P_{\boldsymbol{U}_{i,k}})^2$. Right: empirical Wasserstein error at final time versus the local fill distance $h_{\rho,\mathcal{D}_k}$ on a log-log scale. Each point corresponds to one iteration. A power-law relation is observed with slope approximately equal to 2, indicating that $W_2^2 \sim h_{\rho,\mathcal{D}_k}^{\eta}$ with $\eta \approx 2$ over the range of iterations considered.

contribution of the model-cost term. While the agreement is not exact, which is expected since the analysis neglects serial overheads, the comparison provides empirical support for the derived theoretical cost model as a predictor of practical runtime behavior.

We further validate the cost model by examining the dependence of the runtime on the number of time intervals N. Figure 16, Panel A, shows the empirical wall-clock runtime for both Prob-GPArareal and Prob-nnGPArareal as N increases, together with the scaling predicted by the cost model. All experiments are conducted on a fixed problem setup (same ODE, solver configuration, tolerance ϵ, and number of samples n), varying only the number of time intervals N; runtimes are measured as total wall-clock time. As before, the theoretical curves are obtained by evaluating the dominant terms in Section 3.3, up to proportionality constants estimated from the data. For Prob-GPArareal, the model predicts a superlinear growth driven by the cubic dependence of the GP training cost on the dataset size, which is reflected in the empirical scaling. In contrast, Prob-nnGPArareal exhibits a significantly milder dependence on N, consistent with the reduced complexity induced by the nearest-neighbour approximation. Additionally, Figure 16, Panel B, reports the corresponding speedup as a function of N. Prob-nnGPArareal closely follows the ideal scaling N/K_{conv}, maintaining a substantial speedup as N increases. In contrast, Prob-GPArareal exhibits a progressive loss of efficiency for large N, as the cubic GP training cost eventually dominates the runtime. Overall, these results confirm that the cost model captures both the scaling of the runtime and the resulting parallel efficiency, and highlight the importance of the nnGP approximation for achieving scalable performance as the number of time intervals increases.

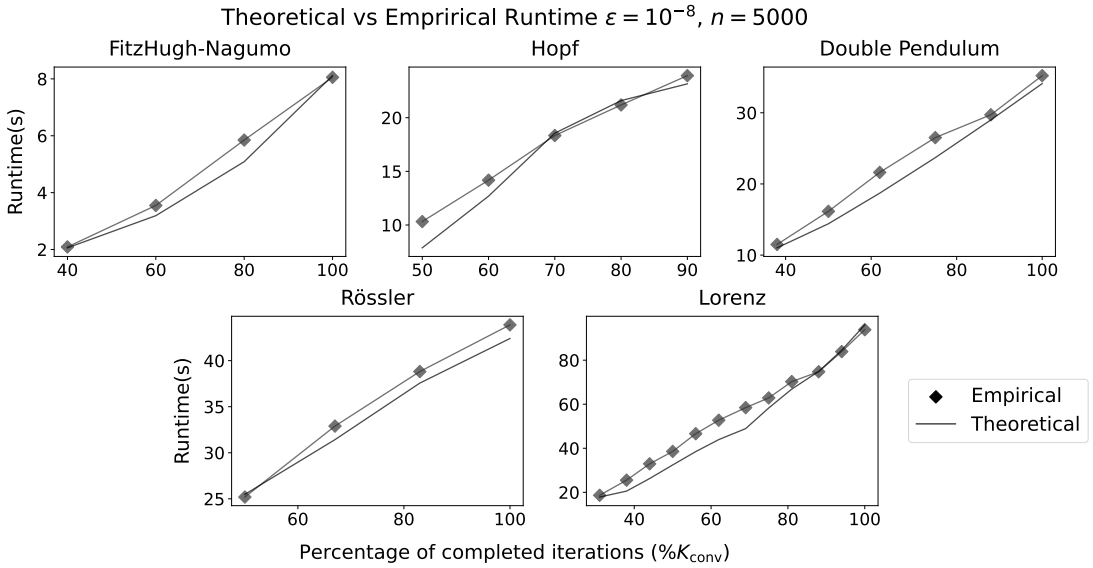

Figure 15: Comparison between empirical wall-clock time and runtime predicted by the cost model for Prob-GParareal across systems. The x-axis shows the percentage of completed iterations relative to K_{conv}. Empirical runtimes are averaged over ten independent runs, while the theoretical costs are obtained by evaluating the cost model from Section 3.3 at the corresponding values of N, n, and K_{conv}.

Appendix J. Comparison with Bosch et al. (2024) parallel-in-time probabilistic ODE solver

In this section, we provide a comparison with the parallel-in-time probabilistic ODE solver of Bosch et al. (2024). Their method formulates probabilistic ODE solving as a time-parallel iterated extended Kalman smoother (IEKS), combining a global first-order linearization of the nonlinear observation model with a time-parallel Kalman filtering and smoothing step. For affine problems, the resulting Gaussian state estimation problem can be solved exactly, while for nonlinear vector fields the method relies on the IEKS approximation.

We consider the FHN system and compare Prob-GParareal with the solver of Bosch et al. (2024) in two representative settings. In the first, the baseline is run on a time grid matching the number of time intervals used by Prob-GParareal, i.e., $N = 32$. In the second, it is run on a substantially denser time grid (5000 points), chosen so that the resulting posterior standard deviations of the two algorithms are of comparable magnitude in the displayed solution. For each method, shown in Figure 17, we report the posterior mean, the associated uncertainty bands, and the observed runtime. The matching-N configuration (Panel B) is approximately three times faster than Prob-GParareal (Panel A), but yields substantially larger posterior uncertainty, with average marginal standard deviations several orders of magnitude higher. In contrast, when the baseline is run on a denser grid (Panel C), the resulting uncertainty becomes comparable to that of Prob-GParareal, at the cost of an increase in runtime by a

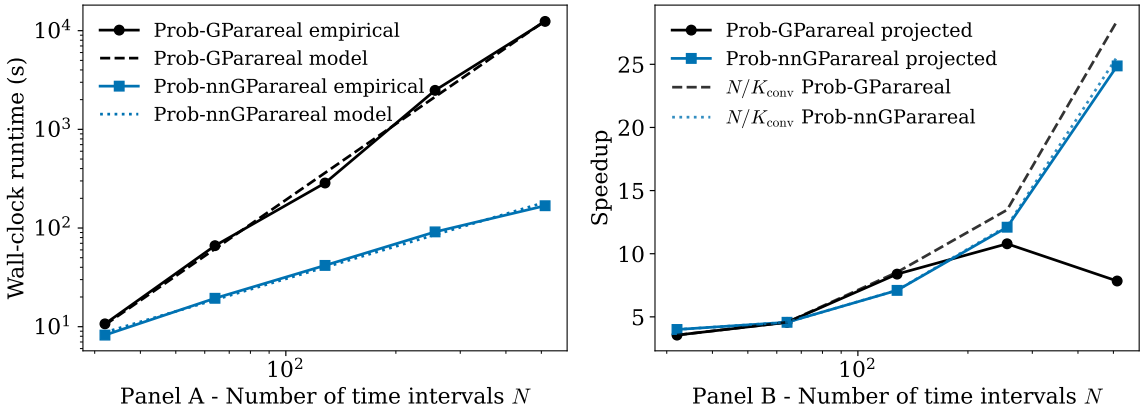

Figure 16: Empirical validation of the Prob-(nn)GPararеal cost model with respect to the number of time intervals N. Panel A: empirical wall-clock runtime (solid lines with markers) together with the scaling predicted by the cost model from Section 3.3 for Prob-GPararеal (dashed lines) and Prob-nnGPararеal (dotted lines). Panel B: corresponding speedup as a function of N. The nnGP variant closely follows the ideal scaling N/K_{conv}, while the full GP variant exhibits reduced efficiency for large N due to the higher cost of the correction step.

factor of approximately 1.5. In terms of solution accuracy, both Prob-GPararеal (Panel A) and the denser-grid baseline (Panel C) achieve similar performances, while the matching-N configuration (Panel B) exhibits a larger error.

Some cautionary words. This comparison should be interpreted as contextual rather than definitive. The two methods are based on different modeling assumptions: Bosch et al. (2024) place a prior directly on the continuous-time trajectory and perform inference through IEKS, whereas Prob-GPararеal models the Parareal correction and propagates uncertainty through the coarse solver by sampling. Accordingly, the aim here is not to establish superiority of one method over the other, but to provide a baseline for context. As is well known in the PinT literature, comparisons across methods with different underlying principles are inherently challenging and therefore uncommon (Gattiglio et al., 2024). Nonetheless, presenting both approaches on the same problem offers useful insight into the practical trade-offs in runtime, solution accuracy, and uncertainty quantification.

References

A. Abdulle and G. Garegnani. Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. *Statistics and Computing*, 30(4):907–932, 2020.

C. Alexander, M. Coulon, Y. Han, and X. Meng. Evaluating the discrimination ability of proper multi-variate scoring rules. *Annals of Operations Research*, 334(1):857–883, 2024.

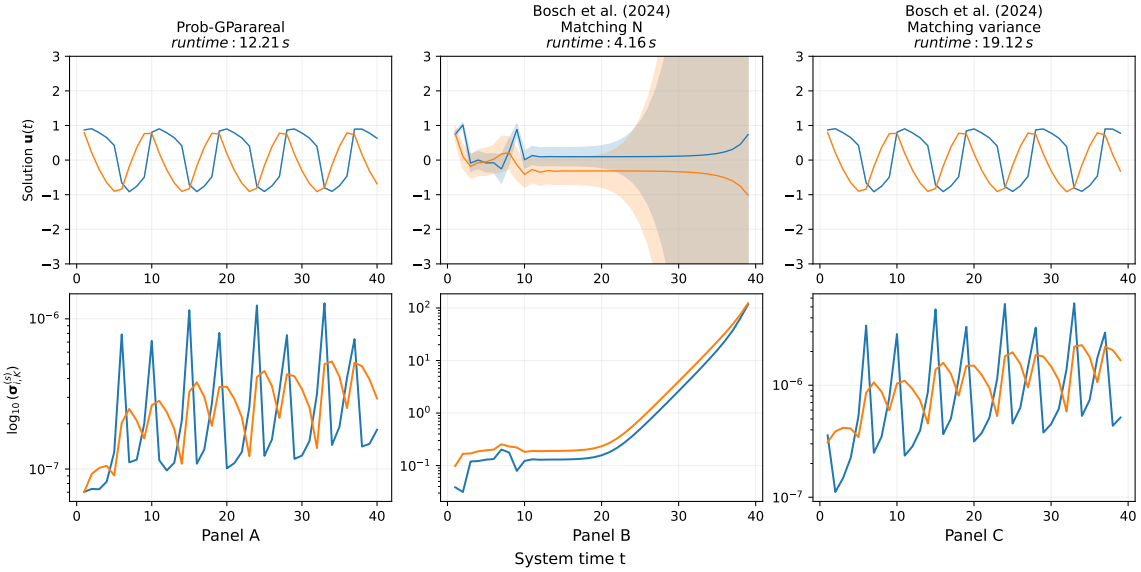

Figure 17: Comparison between Prob-GParareal and the parallel-in-time probabilistic ODE solver of Bosch et al. (2024) for the FitzHugh-Nagumo system. Columns (A), (B), and (C) correspond respectively to Prob-GParareal, the baseline run on a grid with matching number of time intervals, and the baseline run on a denser grid (5000 points). The first row shows the posterior mean (together with ± 2 standard deviation bands for Bosch et al. (2024)) for each coordinate of the system ($d = 2$), while the second row reports the marginal posterior standard deviations. The runtime of each method is shown in the column title.

K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Textbooks in Mathematical Sciences. Springer, New York, 1996. doi: 10.1007/b97589.

M. A. Alvarez, L. Rosasco, N. D. Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

J. Angel, S. Götschel, and D. Ruprecht. Impact of spatial coarsening on Parareal convergence for the linear advection equation. *arXiv preprint arXiv:2111.10228*, 2021.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

G. Bal. On the convergence and the stability of the Parareal algorithm to solve partial differential equations. In *Domain Decomposition Methods in Science and Engineering*, pages 425–432. Springer, 2005.

G. Bal and Y. Maday. A "Parareal" time discretization for non-linear PDE's with application to the pricing of an American put. In *Recent Developments in Domain Decomposition*

Methods, volume 23 of *Lecture Notes in Computational Science and Engineering*, pages 189–202. Springer, 2002.

J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

E. Bernton, P. E. Jacob, M. Gerber, and C. P. Robert. Approximate Bayesian computation with the Wasserstein distance. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(2):235–269, 2019.

K. Bogner, K. Liechti, and M. Zappa. Post-processing of stream flows in Switzerland with an emphasis on low flows and floods. *Water*, 8(4):115, 2016.

N. Bosch, A. Corenflos, F. Yaghoobi, F. Tronarp, P. Hennig, and S. Särkkä. Parallel-in-time probabilistic numerical ODE solvers. *Journal of Machine Learning Research*, 25(206):1–27, 2024.

J. Bröcker and L. A. Smith. Scoring probabilistic forecasts: The importance of being proper. *Weather and Forecasting*, 22(2):382–388, 2007.

Y. Cai. Multivariate quantile function models. *Statistica Sinica*, pages 481–496, 2010.

A. Christmann and I. Steinwart. *Support Vector Machines*. Springer, 2008.

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

M. P. Deisenroth, A. A. Faisal, and C. S. Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

J. Dumas, A. Wehenkel, D. Lanaspeze, B. Cornélusse, and A. Sutera. A deep generative model for probabilistic energy forecasting in power systems: normalizing flows. *Applied Energy*, 305:117871, 2022.

M. Emmett and M. Minion. Toward an efficient parallel in time method for partial differential equations. *Communications in Applied Mathematics and Computational Science*, 7(1): 105–132, 2012.

R. Falgout, S. Friedhoff, T. Kolev, S. MacLachlan, and J. Schroder. Parallel time integration with multigrid. *SIAM Journal on Scientific Computing*, 36(6):C635–C661, 2014.

J. Fitzsimons, K. Cutajar, M. Osborne, S. Roberts, and M. Filippone. Bayesian inference of log determinants. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, Sydney, Australia, 2017.

B. Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706, 1988.

S. Friedhoff, R. Falgout, T. Kolev, S. MacLachlan, and J. Schroder. A multigrid-in-time algorithm for solving evolution equations in parallel. In *Sixteenth Copper Mountain Conference on Multigrid Methods*, 2012.

J. W. Galbraith and S. van Norden. Assessing gross domestic product and inflation probability forecasts derived from Bank of England fan charts. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 175(3):713–727, 2012.

M. J. Gander. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition Methods: MuS-TDD, Heidelberg, May 6-8, 2013*, pages 69–113. Springer, 2015.

M. J. Gander and E. Hairer. Nonlinear convergence analysis for the Parareal algorithm. In *Domain Decomposition Methods in Science and Engineering XVII*, pages 45–56. Springer, 2008.

M. J. Gander and S. Vandewalle. Analysis of the Parareal time-parallel time-integration method. *SIAM Journal on Scientific Computing*, 29(2):556–578, 2007.

M. J. Gander, T. Lunet, D. Ruprecht, and R. Speck. A unified analysis framework for iterative parallel-in-time algorithms. *SIAM Journal on Scientific Computing*, 45(5):A2275–A2303, 2023.

G. Gattiglio, L. Grigoryeva, and M. Tamborrino. Randnet-Parareal: a time-parallel PDE solver using Random Neural Networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:94993–95025, 2024.

G. Gattiglio, L. Grigoryeva, and M. Tamborrino. Nearest neighbors GParareal: Improving scalability of Gaussian processes for parallel-in-time solvers. *SIAM Journal on Scientific Computing*, 47(6):B1400–B1423, 2025.

T. Gneiting and M. Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

T. Gneiting, F. Balabdaoui, and A. E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69 (2):243–268, 2007.

P. Goovaerts. *Geostatistics for Natural Resources Evaluation*, volume 483. Oxford University Press, 1997.

O. Gorynina, F. Legoll, T. Lelièvre, and D. Perez. Combining machine-learned and empirical force fields with the Parareal algorithm: application to the diffusion of atomistic defects. *Comptes Rendus. Mécanique*, 351(S1):479–503, 2023.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, 2007.

A. Griffith, A. Pomerance, and D. J. Gauthier. Forecasting chaotic systems with very low connectivity reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):123108, 2019. doi: 10.1063/1.5120710.

T. Haut and B. Wingate. An asymptotic parallel-in-time method for highly oscillatory PDEs. *SIAM Journal on Scientific Computing*, 36(2):A693–A713, 2014.

P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

P. Hennig, M. A. Osborne, and H. P. Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.

A. J. M. Howse, H. De Sterck, R. D. Falgout, S. P. MacLachlan, and J. B. Schroder. Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations. *SIAM Journal on Scientific Computing*, 41(1):A538–A565, 2019. doi: 10.1137/17M1144982.

R. J. Hyndman. Computing and graphing highest density regions. *The American Statistician*, 50(2):120–126, 1996.

S. Iqbal, H. Abdulsamad, T. Cator, U. Braga-Neto, and S. Särkkä. Parallel-in-time probabilistic solutions for time-dependent nonlinear partial differential equations. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2024.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

H. Kersting and P. Hennig. Active uncertainty calibration in Bayesian ODE solvers. *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence (UAI 2016)*, pages 309–318, 2016.

H. Kersting, T. J. Sullivan, and P. Hennig. Convergence rates of Gaussian ODE filters. *Statistics and Computing*, 30(6):1791–1816, 2020.

N. Krämer and P. Hennig. Stable implementation of probabilistic ODE solvers. *Journal of Machine Learning Research*, 25(111):1–29, 2024.

O. A. Krzysik, H. De Sterck, R. D. Falgout, and J. B. Schroder. Parallel-in-time solution of scalar nonlinear conservation laws. *SIAM Journal on Scientific Computing*, 47(6): A3134–A3160, 2025. doi: 10.1137/24M1630268.

F. M. Larkin. Gaussian measure in Hilbert space and applications in numerical analysis. *Rocky Mountain J. Math.*, 2(3):379–421, 1972.

F. Legoll, T. Lelievre, and G. Samaey. A micro-macro Parareal algorithm: application to singularly perturbed ordinary differential equations. *SIAM Journal on Scientific Computing*, 35(4):A1951–A1986, 2013.

S. Lerch, T. L. Thorarinsdottir, F. Ravazzolo, and T. Gneiting. Forecaster's dilemma: extreme events and forecast evaluation. *Statistical Science*, pages 106–127, 2017.

J.-L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps pararéel. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 332(7):661–668, 2001.

E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20: 130–141, 1963.

Z. Lu, B. R. Hunt, and E. Ott. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6), 2018.

M. Mahsereci and P. Hennig. Probabilistic line searches for stochastic optimization. *Journal of Machine Learning Research*, 18(119):1–59, 2017.

B. Matérn. *Spatial Variation*. Springer New York, 2 edition, 1986.

M. Minion. A hybrid Parareal spectral deferred corrections method. *Communications in Applied Mathematics and Computational Science*, 5(2):265–301, 2011.

T. P. Minka. Deriving quadrature rules from Gaussian processes. Technical report, Technical report, Statistics Department, Carnegie Mellon University, 2000.

J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

C. J. Oates and T. J. Sullivan. A modern retrospective on probabilistic numerics. *Statistics and Computing*, 29(6):1335–1351, 2019.

G. Pages, O. Pironneau, and G. Sall. The Parareal algorithm for American options. *SIAM Journal on Financial Mathematics*, 9(3):966–993, 2018.

J. Pathak and E. Ott. Reservoir computing for forecasting large spatiotemporal dynamical systems. In *Reservoir Computing*, pages 117–138. Springer, 2021.

J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(12):121102, 2017.

A. G. Peddle, T. Haut, and B. Wingate. Parareal convergence for oscillatory pdes with finite time-scale separation. *SIAM Journal on Scientific Computing*, 41(6):A3476–A3497, 2019.

K. Pentland, M. Tamborrino, and T. J. Sullivan. Error bound analysis of the stochastic Parareal algorithm. *SIAM Journal on Scientific Computing*, 45(5):A2657–A2678, 2023a.

K. Pentland, M. Tamborrino, T. J. Sullivan, J. Buchanan, and L. C. Appel. GParareal: a time-parallel ODE solver using Gaussian process emulation. *Statistics and Computing*, 33 (1):23, 2023b.

K. Pentland, M. Tamborrino, D. Samaddar, and L. C. Appel. Stochastic Parareal: an application of probabilistic methods to time-parallelization. *SIAM Journal on Scientific Computing*, 45:S82–S102, 2023.

G. Pettet. Computer modeling: From sports to spaceflight... from order to chaos. *Australian Science Teachers Journal*, 45(2):69, 1999.

B. Philippi and T. Slawig. The Parareal algorithm applied to the FESOM2 ocean circulation model. *arXiv preprint arXiv:2208.07598*, 2022.

B. Philippi and T. Slawig. A micro-macro Parareal implementation for the ocean-circulation model FESOM 2. *arXiv preprint arXiv:2306.17269*, 2023.

R. Pic, C. Dombry, P. Naveau, and M. Taillardat. Proper scoring rules for multivariate probabilistic forecasts based on aggregation and transformation. *Advances in Statistical Climatology, Meteorology and Oceanography*, 11(1):23–58, 2025.

J. M. Reynolds-Barredo, D. E. Newman, R. Sánchez, D. Samaddar, L. A. Berry, and W. R. Elwasif. Mechanisms for the convergence of time-parallelized, Parareal turbulent plasma simulations. *Journal of Computational Physics*, 231(23):7851–7867, 2012.

O. E. Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976.

D. Ruprecht. Wave propagation characteristics of Parareal. *Computing and Visualization in Science*, 19(1):1–17, 2018.

D. Samaddar, D. E. Newman, and R. Sánchez. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the Parareal algorithm. *Journal of Computational Physics*, 229(18):6558–6573, 2010.

D. Samaddar, D. P. Coster, X. Bonnin, L. A. Berry, W. R. Elwasif, and D. B. Batchelor. Application of the Parareal algorithm to simulations of ELMs in ITER plasma. *Computer Physics Communications*, 235:246–257, 2019.

S. Särkkä and Á. F. García-Fernández. Temporal parallelization of Bayesian smoothers. *IEEE Transactions on Automatic Control*, 66(1):299–306, 2020.

S. Särkkä, J. Hartikainen, L. Svensson, and F. Sandblom. Gaussian process quadratures in nonlinear sigma-point filtering and smoothing. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2014.

M. Scheuerer and T. M. Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

A. Schmitt, M. Schreiber, P. Peixoto, and M. Schäfer. A numerical study of a semi-Lagrangian Parareal method applied to the viscous Burgers equation. *Computing and Visualization in Science*, 19(1):45–57, 2018.

M. Schober, S. Särkkä, and P. Hennig. A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 29(1):99–122, 2019.

M. Selig, N. Oppermann, and T. A. Enßlin. Improving stochastic estimates with inference methods: Calculating matrix diagonals. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 85(2):021134, 2012.

R. Seydel. *Practical Bifurcation and Stability Analysis*, volume 5. Springer Science & Business Media, 2009.

J. C. Sprott. *Chaos and Time-Series Analysis*. Oxford university press, 2003.

B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet. On integral probability metrics, ϕ-divergences and binary classification. *arXiv preprint arXiv: 0901.2698*, 2009.

B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

G. A. Staff and E. M. Rønquist. Stability of the Parareal algorithm. In *Domain Decomposition Methods in Science and Engineering*, pages 449–456. Springer, 2005.

A. Suldin. Wiener measure and its applications to approximation methods. *Izv. Vyssh. Uchebn. Zaved. Matematika*, 6(13):145–158, 1959.

O. Teymur, H. C. Lie, T. Sullivan, and B. Calderhead. Implicit probabilistic integrators for ODEs. *Advances in Neural Information Processing Systems*, 31, 2018.

F. Tronarp, H. Kersting, S. Särkkä, and P. Hennig. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. *Statistics and Computing*, 29:1297–1315, 2019.

C. Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2008.

P.-R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

H. Wackernagel. *Multivariate Geostatistics: an Introduction with Applications*. Springer Science & Business Media, 2003.

H. Wendland. *Scattered Data Approximation*, volume 17. Cambridge university press, 2004.

R. L. Winkler. A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association*, 67(337):187–191, 1972.

Z.-M. Wu and R. Schaback. Local error estimates for radial basis function interpolation of scattered data. *IMA journal of Numerical Analysis*, 13(1):13–27, 1993.

X. Xi, F.-X. Briol, and M. Girolami. Bayesian quadrature for multiple related integrals. In *International Conference on Machine Learning*, pages 5373–5382. PMLR, 2018.

D.-X. Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, 220(1-2):456–463, 2008.

F. Ziel and K. Berk. Multivariate forecasting evaluation: On sensitive and strictly proper scoring rules. *arXiv preprint arXiv:1910.07325*, 2019.