PFEM-GP-dPHS : a finite element framework for combining Gaussian processes and infinite-dimensional port-Hamiltonian systems

Florian Courteville FBCO@KTH.SE
KTH royal institute of technology, Stockholm, Sweden
Institut Laue-Langevin, 38000, Grenoble, France

Iain Henderson IAIN.HENDERSON@ISAE-SUPAERO.FR
Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, 31000, Toulouse, France

Denis Matignon DENIS.MATIGNON@ISAE-SUPAERO.FR
Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, 31000, Toulouse, France

Sylvain Dubreuil SYLVAIN.DUBREUIL@ONERA.FR
DTIS, ONERA, Université de Toulouse, 31000, Toulouse, France
Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, 31000, Toulouse, France

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

In order to learn distributed port-Hamiltonian systems (dPHS) using Gaussian processes (GPs), the partitioned finite element method (PFEM) is combined with the Gp-dPHS method. By following a late lumping approach, the discretization of the functional hyperparameters of the GP prior over the Hamiltonian functional is chosen independently from the discretization of the dPHS, thus reducing the numerical complexity of our method. We next model the mean of the GP prior of the Hamiltonian as a quadratic form, enabling the GP kernel to focus on the nonlinear part of a given dPHS. We illustrate our method on a nonlinear one dimensional wave equation with unknown physical parameters (tension and linear mass).

Code : https://github.com/olimarlouis/PFEM-GP-dpHs.git

Keywords: Physics-informed Machine Learning, Distributed Parameter Port-Hamiltonian System, Partitioned Finite element Method, System Identification

1. Introduction and contribution

The modelling and identification of dynamical systems is a crucial task in a broad range of domains such as physics, engineering and applied mathematics. The mathematical framework of Port-Hamiltonian Systems (PHS) is especially well suited to describe interacting dynamical systems. Gaussian processes are functional Bayesian models that can be used for both model approximation and identification while also providing uncertainty quantification (Rasmussen and Williams, 2006). In this context, T. Beckers and co-authors introduced Gaussian process port-Hamiltonian systems (GP-PHS) for ordinary differential equations (ODEs) (Beckers et al., 2022). This method couples Gaussian processes (GPs) with the PHS formalism to obtain physics-based GP priors, so that the result obtained from the data-driven modelling respects the PHS structure of the unknown system, which has been proved useful in (Negareh et al., 2025). Attempts to extend this approach to partial differential equations (PDEs), known as distributed port-Hamiltonian system (dPHS) have been proposed in (Tan et al., 2024; Li et al., 2025; Tan et al., 2025), leading to the Gaussian process distributed port-Hamiltonian system (GP-dPHS) method. These two contributions apply the GP-PHS

approach to the discretised system obtained using an ad hoc (finite difference) discretization method leading to an approximate finite-dimensional system (an ODE system).

In the present article, we make use of PFEM (Cardoso-Ribeiro et al., 2021), a finite element method designed to respect the structure of a given dPHS at the discrete level. However, applying the GP-PHS method directly to the resulting ODE system would result in a prohibitively large number of hyperparameters. To solve this issue, we adapt the physics-based priors from the GP-dPHS method within the discretization method, PFEM, to obtain a prior whose hyperparameters are defined continuously, and can then be discretized with their own adapted method.

2. Background

We first introduce GPs and PHS, and then present the GP-dPHS method.

2.1. Scalar, vector and function-valued Gaussian processes

In this section we introduce scalar, vector and function-valued Gaussian processes indexed by a Hilbert space, and the properties that we will use. Let $\mathcal{X}$ be a Hilbert space, and $\mathcal{Y}$ be either $\mathbb{R}^n, n \geq 1$ or a Hilbert space. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $f = (f(\boldsymbol{\alpha}))_{\boldsymbol{\alpha} \in \mathcal{X}}$ a family of $\mathcal{Y}$-valued random variables, $f(\boldsymbol{\alpha}) : \Omega \to \mathcal{Y}$. Below, f will be a Gaussian process; its distribution will be determined by its mean function $m : \mathcal{X} \to \mathcal{Y},\ \alpha \mapsto \mathbb{E}(f(\boldsymbol{\alpha}))$ and covariance function (or kernel) k, per detailed below in the cases where $\mathcal{Y} = \mathbb{R},\ \mathcal{Y} = \mathbb{R}^n$ and $\mathcal{Y}$ is a Hilbert space. We summarize this as $f \sim \mathcal{GP}(m, k)$.

Scalar-valued GP (Rasmussen and Williams, 2006) If $\mathcal{Y} = \mathbb{R}$ then $f \sim \mathcal{GP}(m, k)$ is a scalar-valued GP, meaning that for all $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N \in \mathcal{X}$, $(f(\boldsymbol{\alpha}_1), \ldots, f(\boldsymbol{\alpha}_N))$ follows a multivariate Gaussian distribution. The kernel k is given by $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, (\boldsymbol{\alpha}, \boldsymbol{\alpha}') \mapsto \mathbb{C}\mathrm{ov}(f(\boldsymbol{\alpha}), f(\boldsymbol{\alpha}'))$. For k to be the covariance function of a GP, it is necessary and sufficient that k be symmetric and positive semi-definite (p.s.d.), i.e.

$$\forall N \in \mathbb{N}, \quad \forall \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N \in \mathcal{X}, \quad \forall\, c_1, \ldots, c_N \in \mathbb{R}, \quad \sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j) \geq 0. \tag{1}$$

GPs can be conditioned on measurement data. For this, consider outputs y_i at location $\boldsymbol{\alpha}_i$, affected by Gaussian noise $\eta_i \sim \mathcal{N}(0, \sigma^2)$, with (n_i) i.i.d. : $y_i = f(\boldsymbol{\alpha}_i) + \eta_i$. This yields a training set $\mathcal{D} = \{X, Y\}$ of input/output data, $X = [\boldsymbol{\alpha}_1, \ldots \boldsymbol{\alpha}_N]$ and $Y = [y_1, \ldots y_N]$. Then $f \sim \mathcal{GP}(m, k)$ can be conditioned on $\mathcal{D}$, yielding a *posterior* GP with explicit mean and covariance functions $m_{\mathcal{D}}(\alpha) = \mathbb{E}(f(\boldsymbol{\alpha})|\mathcal{D})$ and $k_{\mathcal{D}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \mathbb{C}\mathrm{ov}(f(\boldsymbol{\alpha}), f(\boldsymbol{\alpha}')|\mathcal{D})$ (see (Roustant, 2024), Chapter 4 for further details).

Vector-valued GP (Alvarez et al., 2012) If $\mathcal{Y} = \mathbb{R}^n$ then $f \sim \mathcal{GP}(m, k)$ is a vector-valued GP with *vector-valued* mean function $m : \mathcal{X} \to \mathbb{R}^n$ and symmetric *matrix-valued* kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{n \times n}$; the entries of $k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ describe the covariance between the coordinates of $f(\boldsymbol{\alpha})$ and $f(\boldsymbol{\alpha}')$, through $k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')_{ij} = \mathbb{C}\mathrm{ov}(f(\boldsymbol{\alpha})_i, f(\boldsymbol{\alpha}')_j)$. The p.s.d. property of k reads as

$$\forall N \in \mathbb{N}, \quad \forall \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N \in \mathcal{X}, \quad \forall\, c_1, \ldots, c_N \in \mathbb{R}^n, \quad \sum_{i=1}^{N} \sum_{j=1}^{N} c_i^\top k(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j) c_j \geq 0. \tag{2}$$

Function-valued GP (Batlle et al., 2024, Appendix A) Finally, if $\mathcal{Y}$ is a Hilbert space then $f \sim \mathcal{GP}(m, k)$ is a $\mathcal{Y}$-valued GP with function mean function $m : \mathcal{X} \to \mathcal{Y}$ and symmetric kernel $k : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$. The p.s.d property of k reads as

$$\forall N \in \mathbb{N}, \quad \forall \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N \in \mathcal{X}, \quad \forall\, c_1, \dots, c_N \in \mathcal{Y}, \qquad \sum_{i=1}^{N} \sum_{j=1}^{N} \langle c_i, k(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j) c_j \rangle_{\mathcal{Y}} \geq 0. \tag{3}$$

2.2. Distributed Port-Hamiltonian Systems (dPHS)

Adding input/output ports to a Hamiltonian system leads to a port-Hamiltonian system. The port describes the exchange of energy between the energy storage components (spring, kinetic energy...), the dissipative components (friction, resistor...) and the environment. If the energy variables $\boldsymbol{\alpha} \in \mathcal{X}$ are distributed within a continuous space domain $\mathcal{O}$, so that $\boldsymbol{\alpha} : x \in \mathcal{O} \mapsto \boldsymbol{\alpha}(x)$ are function of space, then we talk about distributed port-Hamiltonian system (van der Schaft and Maschke, 2002). To write down the dPHS equation of a system, we first need to define a Hamiltonian functional $\mathcal{H}$ describing the total energy of the system, with

$$\mathcal{H}(\boldsymbol{\alpha}) = \int_{\mathcal{O}} H(\boldsymbol{x}, \boldsymbol{\alpha}) d\boldsymbol{x} \tag{4}$$

where $H : \mathcal{O} \times \mathcal{X} \to \mathbb{R}$ is the energy density. Then, using the variational derivative of $\mathcal{H}$, we define the co-energy variables $\boldsymbol{e} := \delta_{\boldsymbol{\alpha}} \mathcal{H}$, see (van der Schaft and Maschke, 2002). We call constitutive relation the relation between $\boldsymbol{e}$ and $\boldsymbol{\alpha}$. Finally, the dPHS form is obtained by rewriting the PDE of the system using the energy and co-energy variable such that

$$\partial_t \boldsymbol{\alpha} := \dot{\boldsymbol{\alpha}} = (\mathcal{J} - \mathcal{R}) \boldsymbol{e}(\boldsymbol{\alpha}), \qquad \frac{d}{dt} \mathcal{H}(\boldsymbol{\alpha}) = - \int_{\mathcal{O}} \boldsymbol{e}(\boldsymbol{\alpha}) . \mathcal{R} \boldsymbol{e}(\boldsymbol{\alpha}) + \langle u_\partial, y_\partial \rangle_{\partial \mathcal{O}}. \tag{5}$$

Above $\mathcal{J}$ is a constant skew-symmetric differential operator which describes the exchange of energy with the storage components and $\mathcal{R}$ is a constant nonnegative symmetric operator which describes the exchange with the dissipative components. Collocated input-output (u_∂, y_∂) behaviour describing the interaction with the environment are defined by the boundary conditions of the problem (such as Dirichlet control, Neumann control, Mixed control...), see equation (5). In fact, boundary conditions are required to describe a dPHS even if not explicitly appearing in equation (5). An intrinsic difficulty associated with dPHS is their high dimensionality when discretized, which requires special care even in one-dimensional dPHS such as the one illustrated in Section 5.

2.3. GP-dPHS

We can now introduce GP-dPHS. Consider a dPHS whose Hamiltonian functional $\mathcal{H}$ is partially or totally unknown. We want to learn an approximation of the Hamiltonian $\widehat{\mathcal{H}}$. For this, we follow the GP-pHs method (Beckers et al., 2022) by setting a function-indexed GP prior $\widehat{\mathcal{H}}$ on the true Hamiltonian functional $\mathcal{H}$, yielding $\widehat{\mathcal{H}} \sim \mathcal{GP}(m(\boldsymbol{\alpha}), k(\boldsymbol{\alpha}, \boldsymbol{\alpha}'))$, where $m : \mathcal{X} \to \mathbb{R}$ and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are prescribed mean and kernel functions. Let's also suppose that operators $\mathcal{J}$ and $\mathcal{R}$ are also unknown. We replace them by parametrized operators $\widehat{\mathcal{J}}$ formally skew-symmetric and $\widehat{\mathcal{R}}$ formally symmetric. We note $\widehat{\mathcal{J}_R} := \widehat{\mathcal{J}} - \widehat{\mathcal{R}}$. The resulting approximate dPHS is

$$\dot{\boldsymbol{\alpha}} = \widehat{\mathcal{J}_R}\, \delta_{\boldsymbol{\alpha}} \widehat{\mathcal{H}}(\boldsymbol{\alpha}) \quad + \quad \text{boundary conditions.} \tag{6}$$

We denote $f_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha}) := \widehat{\mathcal{J}_R}\,\delta_{\boldsymbol{\alpha}}\widehat{\mathcal{H}}(\boldsymbol{\alpha})$, the right-hand side of (6) viewed as a function of $\boldsymbol{\alpha}$. From the stability of GPs via the linear map $\widehat{\mathcal{H}} \mapsto \widehat{\mathcal{J}_R}\,\delta_{\boldsymbol{\alpha}}\widehat{\mathcal{H}}$, $f_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha})$ is an $\mathcal{X}$-valued GP (Section 2.1) with

$$f_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha}) \sim \mathcal{GP}(\widehat{\mathcal{J}_R}\,\delta_{\boldsymbol{\alpha}} m(\boldsymbol{\alpha}), \widehat{\mathcal{J}_R}\,(\delta_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} k(\boldsymbol{\alpha},\boldsymbol{\alpha}'))\,\widehat{\mathcal{J}_R^*}).$$

Then, given $m_{\dot{\boldsymbol{\alpha}}}$, the posterior expectation of the GP $f_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha})$ conditioned on measurement data, we can solve the differential equation (coupled with boundary conditions) $\dot{\boldsymbol{\alpha}}(t) = m_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha}(t))$ with $\boldsymbol{\alpha}(0) = \boldsymbol{\alpha}_0 \in \mathcal{X}$. Alternatively we can replace $m_{\dot{\boldsymbol{\alpha}}}$ with a sample of the posterior GP. Note that unlike GP-dPHS as described in (Tan et al., 2024), this version of GP-dPHS is still infinite-dimensional at this stage. Its discretization is achieved using the Partitioned Finite Element Method (PFEM) in Section 4. This procedure consisting in first studying/controlling a dPHS and then discretizing is known as late lumping, as opposed to first discretizing it and then controlling it (early lumping), see (Marko et al., 2018). Finally, this approach ensures *discretization invariance* over the estimation of $\mathcal{H}$ and the related dynamics, as described in e.g. (Lassas and Siltanen, 2004; Stuart, 2010).

3. Discretization of the GP-dPHS with PFEM : PFEM-GP-dPHS

To perform numerical simulations, we need to discretize the GP $f_{\dot{\boldsymbol{\alpha}}}$, or more precisely, write a GP $f_{\underline{\dot{\alpha}}}$ on the discrete coefficients of the discretization of $\dot{\boldsymbol{\alpha}}$. To achieve this, we combine the GP-dPHS method with the partitioned finite element method (PFEM) (Cardoso-Ribeiro et al., 2021) which has the advantage of being structure-preserving, in so far as the power balance of the dPHS is preserved at the discrete level, see equation (9). We call this the PFEM-GP-dPHS method, which we illustrate below on a dissipative nonlinear wave equation in 1D.

3.1. Dissipative nonlinear wave equation in dPHS form

The first step is to write the 1D wave equation in dPHS form. This equation with Neumann boundary control is given by ($\mathcal{O} = [0, \ell]$) for $t \geq 0, x \in \mathcal{O}$

$$\begin{cases} \rho(x)\,\partial_{tt}^2 w(t,x) - \partial_x\left(s(x,\partial_x w(t,x))\,\partial_x w(t,x)\right) + \nu(x)\,\partial_t w(t,x) &= 0, \\ -s(0,\partial_x w(t,0))\partial_x w(t,0) &= u_L(t), \\ s(\ell,\partial_x w(t,\ell))\partial_x w(t,\ell) &= u_R(t). \end{cases} \tag{7}$$

We set the energy variable $\alpha_q := \partial_x w$ the strain and $\alpha_p := \rho\partial_t w$ the linear momentum. The Hamiltonian $\mathcal{H}$ of this system can be defined by $\mathcal{H}(\alpha_q,\alpha_p) := \frac{1}{2}\int_0^\ell \int_0^{\alpha_q(t,x)} s(x,\alpha_q')\alpha_q'\,d\alpha_q' + \frac{\alpha_p(t,x)^2}{\rho(x)}\mathrm{d}x$. Computing the variational derivatives of $\mathcal{H}$ with respect to the energy variables leads to the co-energy variables $e_q := \delta_{\alpha_q}\mathcal{H} = s(.,\alpha_q)\alpha_q$, the stress, and $e_p := \delta_{\alpha_p}\mathcal{H} = \rho^{-1}\alpha_p$, the velocity. With Newton's second law and Schwarz's lemma, we get the port-Hamiltonian system representing a damped vibrating string with Neumann boundary control and Dirichlet boundary observation:

$$\begin{pmatrix}\partial_t\alpha_q\\ \partial_t\alpha_p\end{pmatrix} = \underbrace{\begin{bmatrix}0 & \partial_x\\ \partial_x & -\nu\end{bmatrix}}_{\mathcal{J}_\mathcal{R}=\mathcal{J}-\mathcal{R}}\begin{pmatrix}e_q\\ e_p\end{pmatrix}, \qquad \begin{cases} e_q(t,0) &= u_L(t), \quad y_L(t) = e_p(t,0), \\ e_q(t,\ell) &= u_R(t), \quad y_R(t) = e_p(t,\ell). \end{cases} \tag{8}$$

The following power balance can be computed for this system:

$$\frac{d}{dt}\mathcal{H}(t) = -\int_0^\ell \nu(x)\,e_p(t,x)^2\mathrm{d}x + u_L(t)y_L(t) + u_R(t)y_R(t)\,. \tag{9}$$

3.2. The Partitioned Finite Element Method (PFEM)

Let us apply the partitioned finite elements method (PFEM) (Cardoso-Ribeiro et al., 2021) on the above example. Let φ_q and φ_p be smooth test functions. One can write the weak formulation and integrate the second line by parts to make the controls u_L and u_R appear:

$$\begin{cases} \int_0^\ell \dot{\alpha_q}(t,x)\varphi_q(x)\mathrm{d}x &= \int_0^\ell \partial_x e_p(t,x)\varphi_q(x)\mathrm{d}x, \\ \int_0^\ell \dot{\alpha_p}(t,\cdot)\varphi_p\mathrm{d}x &= -\int_0^\ell e_q(t,.)\partial_x\varphi_p\mathrm{d}x - \int_0^\ell \nu e_p(t,.)\varphi_p\mathrm{d}x + u_R(t)\varphi_p(\ell) + u_L(t)\varphi_p(0), \\ y_L(t) &= e_p(t,0), \\ y_R(t) &= e_p(t,\ell). \end{cases}$$

Now, let $(\varphi_q^i)_{1\le i\le N_q}$ and $(\varphi_p^j)_{1\le j\le N_p}$ be two finite element families of approximation functions for α_q and α_p respectively, so that $\alpha_q(t,x) \simeq \underline{\varphi_q}^\top(x)\underline{\alpha_q}(t)$, $\alpha_p(t,x) \simeq \underline{\varphi_p}^\top(x)\underline{\alpha_p}(t)$ (likewise for $\dot{\alpha_q}$ and $\dot{\alpha_p}$) and $e_q(t,x) \simeq \underline{\varphi_q}^\top(x)\underline{e_q}(t)$, $e_p(t,x) \simeq \underline{\varphi_p}^\top(x)\underline{e_p}(t)$. We next replace the variables by their approximations, and write the system in matrix form

$$\begin{cases} \underbrace{\begin{bmatrix} M_q & 0 \\ 0 & M_p \end{bmatrix}}_{M} \underbrace{\begin{pmatrix} \dot{\underline{\alpha_q}}(t) \\ \dot{\underline{\alpha_p}}(t) \end{pmatrix}}_{\dot{\underline{\alpha}}} &= \underbrace{\begin{bmatrix} 0 & D \\ -D^\top & -R_{22} \end{bmatrix}}_{J-R} \underbrace{\begin{pmatrix} \underline{e_q}(t) \\ \underline{e_p}(t) \end{pmatrix}}_{\underline{e}} + \underbrace{\begin{bmatrix} 0 & 0 \\ B_L & B_R \end{bmatrix}}_{G} \underbrace{\begin{pmatrix} u_L(t) \\ u_R(t) \end{pmatrix}}_{\underline{u}}, \\ \underline{y}(t) := \begin{pmatrix} y_L(t) \\ y_R(t) \end{pmatrix} &= \begin{bmatrix} 0 & B_L^\top \\ 0 & B_R^\top \end{bmatrix} \begin{pmatrix} \underline{e_q}(t) \\ \underline{e_p}(t) \end{pmatrix}, \end{cases} \tag{10}$$

with square matrices $(M_q)_{ij} := \int_0^\ell \varphi_q^j(x)\varphi_q^i(x)\mathrm{d}x$, $(M_p)_{k\ell} := \int_0^\ell \varphi_p^\ell(x)\varphi_p^k(x)\mathrm{d}x$, $(R_{22})_{k\ell} := \int_0^\ell \nu(x)\varphi_p^\ell(x)\varphi_p^k(x)\mathrm{d}x$, and rectangular matrices $(D)_{i\ell} := \int_0^\ell \partial_x\varphi_p^\ell(x)\varphi_q^i(x)\mathrm{d}x$, $(B_L)_k := \varphi_p^k(0)$ and $(B_R)_k := \varphi_p^k(\ell)$. To close this system, we need to approximate the constitutive relations. Write them in their weak form:

$$\begin{cases} \int_0^\ell e_q(t,x)\varphi_q(x)\mathrm{d}x &= \int_0^\ell s(x,\alpha_q(t,x))\alpha_q(t,x)\varphi_q(x)\mathrm{d}x, \\ \int_0^\ell e_p(t,x)\varphi_p(x)\mathrm{d}x &= \int_0^\ell \frac{1}{\rho(x)}\alpha_p(t,x)\varphi_p(x)\mathrm{d}x. \end{cases} \tag{11}$$

Then the matrix form of the discrete weak formulation of the constitutive relations is

$$\begin{cases} M_q\underline{e_q}(t) &= M_s\underline{\alpha_q}(t), \\ M_p\underline{e_p}(t) &= M_\rho\underline{\alpha_p}(t), \end{cases} \tag{12}$$

where $(M_s)_{ij} := \int_0^\ell s(x,\alpha_q(t,x))\varphi_q^j(x)\varphi_q^i(x)\mathrm{d}x$ and $(M_\rho)_{k\ell} := \int_0^\ell \frac{1}{\rho(x)}\varphi_p^\ell(x)\varphi_p^k(x)\mathrm{d}x$.

To retrieve a finite-dimensional pHs in equation (10), we can define the discrete Hamiltonian $\mathcal{H}^d$ as the evaluation of $\mathcal{H}$ on the approximation of the state variables $\mathcal{H}^d(\underline{\alpha_q},\underline{\alpha_p}) := \frac{1}{2}\underline{\alpha_q}^\top M_s(\alpha_q)\underline{\alpha_q} + \frac{1}{2}\underline{\alpha_p}^\top M_\rho\underline{\alpha_p}$, and can write that

$$\begin{bmatrix} \underline{e_q} \\ \underline{e_p} \end{bmatrix} = M^{-1} \begin{bmatrix} \nabla_{\underline{\alpha_q}} \\ \nabla_{\underline{\alpha_p}} \end{bmatrix} \mathcal{H}^d(\underline{\alpha_q},\underline{\alpha_p}). \tag{13}$$

The main perk of PFEM is that, whatever the chosen mesh refinement, we have the exact discrete power balance given by (compare with equation (9)):

$$\frac{d}{dt}\mathcal{H}^d(t) = -\underline{e_p}^\top(t)R_{22}\,\underline{e_p}(t) + u_L(t)y_L(t) + u_R(t)y_R(t)\,.$$

Numerical simulations of many dPHS models using PFEM can be performed using the `SCRIMP` environment (Ferraro et al., 2024), see `https://g-haine.github.io/scrimp/`. In Appendices C.1 and C.2, we present an example of a 2D dPHS along with its discretization using PFEM.

3.3. Expressing PFEM as an affine transformation of the Hamiltonian functional

Assume that the Hamiltonian functional $\mathcal{H}$ is unknown, and that the parameter ν is given. A first way to identify $\mathcal{H}$ would be to use the GP-PHS method from (Beckers et al., 2022) by setting a GP prior on some discretized Hamiltonian $\mathcal{H}^d$ (early lumping). Unfortunately, this approach leads to a number of hyperparameters proportional to the number of degrees of freedom used for the discretization as we would have to define a set of hyperparameter for each degree of freedom. To solve this issue, we use the GP-dPHS method by first setting a function-indexed GP prior $\widehat{\mathcal{H}}$ on the true Hamiltonian functional $\mathcal{H}$ itself : $\widehat{\mathcal{H}} \sim \mathcal{GP}(m(\boldsymbol{\alpha}), k(\boldsymbol{\alpha}, \boldsymbol{\alpha}'))$ (late lumping), see Section 2.3. $f_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha}) = \widehat{\mathcal{J}_R}\, \delta_{\boldsymbol{\alpha}} \widehat{\mathcal{H}}(\boldsymbol{\alpha})$ is a function-valued GP, see Section 2.3. Explicit mean and kernel are described in Section 4. We next choose the hyperparameters of m and k as continuous functions themselves, which enables us to discretize them independently from the PFEM discretization of a given dPHS, see Section 4. We will use piecewise linear approximations but other choices can be considered such as splines, trigonometric sums or wavelets.

We then apply the PFEM method on the dPHS $\dot{\boldsymbol{\alpha}} = f_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha})$. We show below that PFEM can be viewed as applying a linear transformation on $f_{\dot{\boldsymbol{\alpha}}}(\boldsymbol{\alpha})$, thus yielding a GP $f_{\underline{\dot{\alpha}}}(\underline{\alpha}, \underline{u}) \in \mathbb{R}^{N_q+N_p}$ which is vector-valued but also vector-indexed by the *pair* $(\underline{\alpha}, \underline{u}) \in \mathbb{R}^{N_q+N_p} \times \mathbb{R}^2$; see Section 3.2 for the introduction of $\underline{\alpha}$ and $\underline{u}$.

We thus express $\underline{\dot{\alpha}}$ as combination of affine transformations applied to $\widehat{\mathcal{H}}$. For this, we start from the left-hand side of equations (11) and (12) to link $\underline{e}$ and $\widehat{\mathcal{H}}$, yielding

$$M\underline{e}(\boldsymbol{\alpha}) = \int_0^\ell \Phi(x)(\delta_{\boldsymbol{\alpha}}\widehat{\mathcal{H}})(\boldsymbol{\alpha})(x)\mathrm{d}x\,, \quad \Phi(x) = \begin{bmatrix} \underline{\varphi_q}(x) & 0_{N_q,1} \\ 0_{N_p,1} & \underline{\varphi_p}(x) \end{bmatrix}. \tag{14}$$

Above we have $\Phi \in \mathbb{R}^{(N_q+N_p)\times 2}$. We next substitute $\boldsymbol{\alpha}$ with its finite dimensional approximation $\Phi^T\underline{\alpha} \simeq \boldsymbol{\alpha}$ and use equation (10) which links $\underline{\dot{\alpha}}$ and $\underline{e}$, to find

$$f_{\underline{\dot{\alpha}}}(\underline{\alpha}, \underline{u}) = \underline{\dot{\alpha}} = M^{-1}(J-R)M^{-1}\int_0^\ell \Phi(x)(\delta_{\boldsymbol{\alpha}}\widehat{\mathcal{H}})(\Phi^T\underline{\alpha})(x)\mathrm{d}x + G\underline{u}\,. \tag{15}$$

Equation (15) thus expresses $f_{\underline{\dot{\alpha}}}(\underline{\alpha}, \underline{u})$ as an affine transformation of $\widehat{\mathcal{H}}$. and we can derive the GP followed by $f_{\underline{\dot{\alpha}}}(\underline{\alpha}, \underline{u})$. The prior of this so called PFEM-GP-dPHS depends on the hyperparameters of the prior $\widehat{\mathcal{H}}$, as well as the parameter ν. As a function of $\underline{\alpha}$, the discrete output $\underline{y}$ is also a GP since

$$\underline{y}(\underline{\alpha}) = G^\top M^{-1}\int_0^\ell \Phi(x)(\delta_{\boldsymbol{\alpha}}\widehat{\mathcal{H}})(\Phi^T\underline{\alpha})(x)\mathrm{d}x\,. \tag{16}$$

After the GP $\widehat{\mathcal{H}}$ is conditioned on measurement data, we obtain a posterior vector-valued GP whose (posterior) expectation is denoted $m_{\underline{\dot{\alpha}}}(\underline{\alpha}, \underline{u})$. Finally, we emulate the true dPHS by solving $\underline{\dot{\alpha}}(t) = m_{\underline{\dot{\alpha}}}(\underline{\alpha}(t), \underline{u}(t))$ with $\underline{\alpha}(0) = \underline{\alpha}_0 \in \mathcal{X}$.

4. Prior of the PFEM-GP-dPHS

4.1. Selecting a prior for the Hamiltonian functional

Following equation (15) a prior on $\widehat{\mathcal{H}}$ should be defined to obtain the prior of $f_{\underline{\dot{\alpha}}}$. For this (Li et al., 2025) proposed using a zero mean and a squared exponential (SE) kernel. However, in most physical systems, the Hamiltonian usually goes to infinity when a variable goes to infinity. This behaviour is absolutely not captured by GP with zero mean and SE kernel, because sampled functions from a GP with a constant zero mean value and SE kernel are necessary bounded. To remedy this, we propose to use a GP prior with a quadratic mean and a SE kernel. This will first allow the sampled Hamiltonians to go to infinity when a variable goes to infinity and second, as in physical systems there are often quadratic dependencies of the Hamiltonian on some of the variables, the mean will approximate the quadratic part of the Hamiltonian, and the kernel will focus on the non-quadratic part of the Hamiltonian, allowing better modelling of the system. Note that the quadratic part of the Hamiltonian corresponds to the linear part of the dPHS, hence the kernel will mainly focus on approximating the nonlinear part of the dPHS. Thus the scalar prior mean m and kernel k selected for the GP of $\widehat{\mathcal{H}}$ of our example, such that $\widehat{\mathcal{H}} \sim \mathcal{GP}(m, k)$, are given by

$$m(\boldsymbol{\alpha}) = \int_0^\ell m_q(x)\alpha_q^2(x)\,dx + \int_0^\ell m_p(x)\alpha_p^2(x)\,dx, \tag{17}$$

$$k(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \sigma_f^2 \exp\left[-\frac{1}{2}\left(\int_0^\ell \frac{(\alpha_q(x) - \alpha_q'(x))^2}{\ell_q^2(x)}\,dx + \int_0^\ell \frac{(\alpha_p(x) - \alpha_p'(x))^2)}{\ell_p^2(x)}\,dx\right)\right], \tag{18}$$

where the functions m_q, m_p, ℓ_q and ℓ_p are nonnegative hyperparameters of the GP $\widehat{\mathcal{H}}$, and $\sigma_f \in \mathbb{R}^+$ is also a hyperparameter of the GP. In Section 5 the functions m_q, m_p, ℓ_q and ℓ_p are discretized using piecewise linear functions; a discretization based on 3[rd] order polynomials is studied in Appendix B. Note that this quadratic mean is a form a physics-informed prior, as the Hamiltonians associated to many dPHS have a quadratic term, e.g. the one associated to the kinetic energy in classical and quantum mechanics. A more detailed study of the role and the impact of the quadratic mean is given in Appendix A.

4.2. Derivation of the prior over $f_{\underline{\dot{\alpha}}}$

To obtain the prior of $f_{\underline{\dot{\alpha}}}$, each affine transformation appearing in equation (15) is applied to the GP of $\widehat{\mathcal{H}}$. First we apply the variational derivative which we decompose as $\delta_{\boldsymbol{\alpha}} = [\delta_{\alpha_q}\ \delta_{\alpha_p}]^\top$. This yields a GP prior over $\boldsymbol{e}(\boldsymbol{\alpha}) \sim \mathcal{GP}(m_e(\boldsymbol{\alpha}), k_e(\boldsymbol{\alpha}, \boldsymbol{\alpha}'))$ with $m_e(\boldsymbol{\alpha}) := \delta_{\boldsymbol{\alpha}} m(\boldsymbol{\alpha})$ and $k_e(\boldsymbol{\alpha}, \boldsymbol{\alpha}') := \delta_{\boldsymbol{\alpha}}\delta_{\boldsymbol{\alpha}'} k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$. Explicitly we have the two formulas

$$m_e(\boldsymbol{\alpha}) = \begin{bmatrix} m_q\alpha_q \\ m_p\alpha_p \end{bmatrix}, \tag{19}$$

$$\frac{k_e(\boldsymbol{\alpha}, \boldsymbol{\alpha}')}{k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')} = \begin{bmatrix} \frac{1}{\ell_q^2}\mathcal{I} - \frac{1}{\ell_q^2}(\alpha_q - \alpha_q') \otimes \frac{1}{\ell_q^2}(\alpha_q - \alpha_q') & -\frac{1}{\ell_q^2}(\alpha_q - \alpha_q') \otimes \frac{1}{\ell_p^2}(\alpha_p - \alpha_p') \\ -\frac{1}{\ell_p^2}(\alpha_p - \alpha_p') \otimes \frac{1}{\ell_q^2}(\alpha_q - \alpha_q') & \frac{1}{\ell_p^2}\mathcal{I} - \frac{1}{\ell_p^2}(\alpha_p - \alpha_p') \otimes \frac{1}{\ell_p^2}(\alpha_p - \alpha_p') \end{bmatrix}, \tag{20}$$

Note that above, we have a function-valued mean and a symmetric p.s.d. operator-valued kernel. Following Section 3.3, multiply by Φ, integrate and replace $\boldsymbol{\alpha}$ by its finite dimensional approximation $\Phi^\top \underline{\alpha}$. This yields a GP prior over $\underline{e}(\underline{\alpha})$ given by $\underline{e}(\underline{\alpha}) \sim \mathcal{GP}(m_{\underline{e}}, k_{\underline{e}})$ where $m_{\underline{e}}(\underline{\alpha}) = M^{-1}\int_0^\ell \Phi(x) m_e(\underline{\alpha})(x)\mathrm{d}x$ and where the symmetric p.s.d matrix-valued kernel is $k_{\underline{e}}(\underline{\alpha}, \underline{\alpha}') = M^{-1}\int_0^\ell \Phi(x)(k_e(\underline{\alpha}, \underline{\alpha}')\Phi^\top)(x)\mathrm{d}x M^{-1}$. Their expressions are given by

$$m_{\underline{e}}(\underline{\alpha}) = \begin{bmatrix} (M_{m,q}\underline{\alpha_q})^\top & (M_{m,p}\underline{\alpha_p})^\top \end{bmatrix}^\top, \tag{21}$$

with $(M_{m,q})_{ij} := \int_0^\ell m_q \varphi_q^i \varphi_q^j \mathrm{d}x$ and $(M_{m,p})_{ij} := \int_0^\ell m_p \varphi_p^i \varphi_p^j \mathrm{d}x$ and

$$\frac{k_{\underline{e}}(\underline{\alpha}, \underline{\alpha}')}{k(\Phi^\top\underline{\alpha}, \Phi^\top\underline{\alpha}')} = M_k \begin{bmatrix} I_{N_q} - (\underline{\alpha_q} - \underline{\alpha_q'})(\underline{\alpha_q} - \underline{\alpha_q'})^\top & -(\underline{\alpha_q} - \underline{\alpha_q'})(\underline{\alpha_p} - \underline{\alpha_p'})^\top \\ -(\underline{\alpha_p} - \underline{\alpha_p'})(\underline{\alpha_q} - \underline{\alpha_q'})^\top & I_{N_p} - (\underline{\alpha_p} - \underline{\alpha_p'})(\underline{\alpha_p} - \underline{\alpha_p'})^\top \end{bmatrix} M_k^\top, \tag{22}$$

with $(M_{k,q})_{ij} := \int_0^\ell \ell_q^{-2}\varphi_q^i\varphi_q^j \mathrm{d}x$, $(M_{k,p})_{ij} := \int_0^\ell \ell_p^{-2}\varphi_p^i\varphi_p^j \mathrm{d}x$ and $M_k = M^{-1}\mathrm{Diag}(M_{k,q}, M_{k,p})$. Then from here, to obtain the GP prior over $f_{\underline{\dot{\alpha}}}$, we apply the matrix multiplication $(J-R)$ to the GP $\underline{e}$ and add the mean $G\underline{u}$ so that finally

$$f_{\underline{\dot{\alpha}}}(\underline{\alpha}, \underline{u}) \sim \mathcal{GP}(M^{-1}(J-R)m_{\underline{e}}(\underline{\alpha}) + G\underline{u},\ M^{-1}(J-R)k_{\underline{e}}(\underline{\alpha}, \underline{\alpha}')(J-R)^\top M^{-1}). \tag{23}$$

The results given in Sections 3, 4 and 5 are generalized to a 2D dPHS in Appendix C.

5. Numerical results

We test PFEM-GP-dPHS on the dPHS described by equation (8). We decompose the Young's Modulus term s into the sum of a quadratic term $T(x) = 2 - 4x(1-x)$ and a spatially modulated non-quadratic term $c(x)f(\alpha_q(t,x))$. Here, $c(x) = 2x(x-1)^2$ is the spatial modulator and $f(u) = \exp(-u^2)$ is the non-quadratic function. We thus have $s(x, \alpha_q(t,x)) = T(x) + c(x)f(\alpha_q(t,x))$. The linear density is given by $\rho(x) = 3 - 2.5x^2$. The Hamiltonian functional can be written as

$$\mathcal{H}(\boldsymbol{\alpha}) = \mathcal{H}(\alpha_q, \alpha_p) := \frac{1}{2}\int_0^\ell c(x)\int_0^{\alpha_q(t,x)} f(\alpha_q')\alpha_q'\, d\alpha_q' + T(x)\alpha_q^2(t,x) + \frac{\alpha_p(t,x)^2}{\rho(x)}\mathrm{d}x. \tag{24}$$

Above, we have two quadratic terms depending on the parameters T and ρ^{-1} and one non-quadratic term depending on the parameter c. For simplicity, we take $\nu = 0$ so that only the Hamiltonian parameters T, ρ and c are supposed unknown. In the initial state, we set $\underline{\alpha}_0 = \underline{0}$, and we use $\underline{u}(t) = (u_L(t), u_R(t)) = (\sin(\pi t), 0)$. α_q and α_p are discretized on a 21 or 41-point uniform mesh on $\mathbb{P}^1$ Lagrange elements, resulting in 42 or 82 degrees of freedom ($N_q = N_p \in \{21, 41\}$), resulting in a finite-dimensional PHS with high dimension $d \in \{42, 82\}$. The hyperparameters m_q, m_p, ℓ_q^{-1} and ℓ_p^{-1} are discretized on a mesh of $\mathbb{P}^1$ Lagrange elements with mesh size of $\Delta x \in \{0.1, 0.2\}$. Accounting for σ_f and σ_{noise}, this yields a hyperparameter θ of size 26 or 46 (note that different meshes for each hyperparameter functions could be used). In Appendix B, we approximate the hyperparameter functions with polynomials of degree 3, resulting in 18 hyperparameters. Note that using an early lumping approach would result in 166 hyperparameters.

The training data is obtained by first simulating the dPHS described above for $t_f = 20s$ with the `SCRIMP` environment (Ferraro et al., 2024) with time step $\Delta t = 0.01s$ and number of simulation

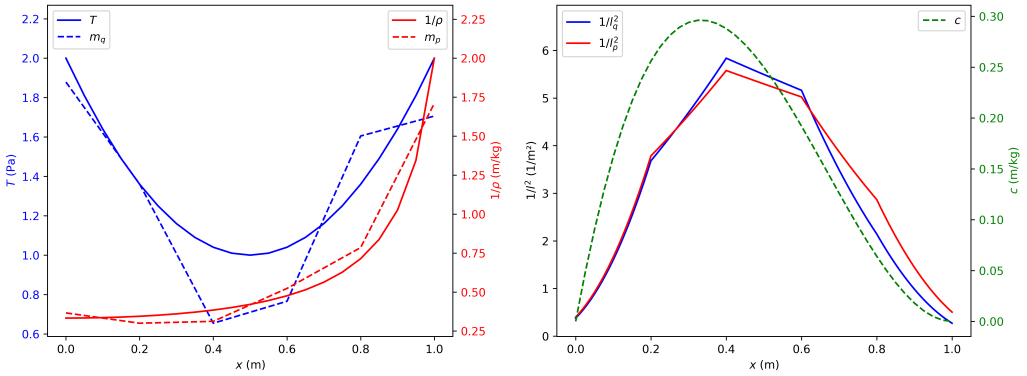

Figure 1: Left : estimation of the GP quadratic mean functions m_q and m_p using piecewise linear functions, versus true physical parameters T and ρ^{-1}. Right : estimation of the GP functional lengthscales ℓ_q^{-2} and ℓ_p^{-2} using piecewise linear functions, versus the function c. Here $N_q = N_p = 21$, $N_s = 35$ and the hyperparameter mesh size is $\Delta x = 0.2$ resulting in 26 hyperparameters.

time steps $N_t = 2000$. In this single simulation we select for $0 \leq t \leq 10s$, $N_s = 35$ uniformly spaced time stamps t_i ($0 \leq i < N_s$) and the corresponding true $\underline{\alpha}_i$ and $\dot{\underline{\alpha}}_i$, resulting in $N_s \times (N_q + N_p) = 1470$ training points. The estimation of the hyperparameters is done by minimizing the negative log marginal likelihood (NLML) using the L-BFGS-B method of `scipy` along with the NLML gradient. After this minimization step, we estimate $f_{\dot{\underline{\alpha}}}$ with its conditional expectation with reference to the training data. We next simulate the resulting dPHS using $m_{\dot{\underline{\alpha}}}$, the posterior mean of $f_{\dot{\underline{\alpha}}}$ conditioned on the training data, by solving $\dot{\underline{\alpha}}(t) = m_{\dot{\underline{\alpha}}}(\underline{\alpha}(t), \underline{u}(t))$ with $\underline{\alpha}(0) = \underline{\alpha}_0$. A discussion on the computational cost of this procedure is given in Appendix D.2.

Given the Hamiltonian $\mathcal{H}(\boldsymbol{\alpha})$ in equation (24) as well as the mean and kernel functions in equations (17) and (18), we expect the mean function $m(\boldsymbol{\alpha})$ to learn the quadratic part of $\mathcal{H}(\boldsymbol{\alpha})$, and the kernel $k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ to learn the nonquadratic part of $\mathcal{H}(\boldsymbol{\alpha})$.

The estimated hyperparameter functions after successful NLML minimisation are given in Figure 1. Figure 1 (left) shows that m_q and m_p approximately follow their respective counterparts T and ρ^{-1}, indicating that the quadratic part of the true Hamiltonian has been approximated by the quadratic mean of the GP prior. Figure 1 (right) shows that the shape of the squared inverse correlation lengths ℓ_q^{-2} and ℓ_p^{-2} follows the shape of the spatial modulator c. We explain this result as follow. Interpreting $\ell_p(x)$ and $\ell_q(x)$ as lengthscales at "dimension x", we expect that the Hamiltonian is well approximated by the quadratic mean at locations ("dimensions x") where its nonlinearity is mild, i.e. $c(x) \simeq 0$. Hence we expect that the tuned kernel k should not contribute at such locations x. Ensuring this latter property corresponds to having infinite lengthscales $\ell_p(x)$ and $\ell_q(x)$, which is the case in Figure 1 (right) at locations where $c(x) \simeq 0$.

Figure 7 and 8 in Appendix D.1 describe simulated states using the approximate evolution equation $\dot{\underline{\alpha}}(t) = m_{\dot{\underline{\alpha}}}(\underline{\alpha}(t), \underline{u}(t))$, versus the `SCRIMP` output. Those figures provide visual evidence that on this example, the approximate dPHS $\dot{\underline{\alpha}}(t) = m_{\dot{\underline{\alpha}}}(\underline{\alpha}(t), \underline{u}(t))$ obtained using the PFEM-GP-dPHS framework provides a reasonable approximation of the true dPHS (equation (5)).

Figure 2 studies the behaviour of our method as a function of the discretization of the functions m_q, m_p, ℓ_q^{-1} and ℓ_p^{-1}. First recall that we have at our disposal a single `SCRIMP` output $(\alpha_q^{\text{SCP}}, \alpha_p^{\text{SCP}})$ with fixed measurement points (up to $10s$ in the simulation) as described above. We discretize the

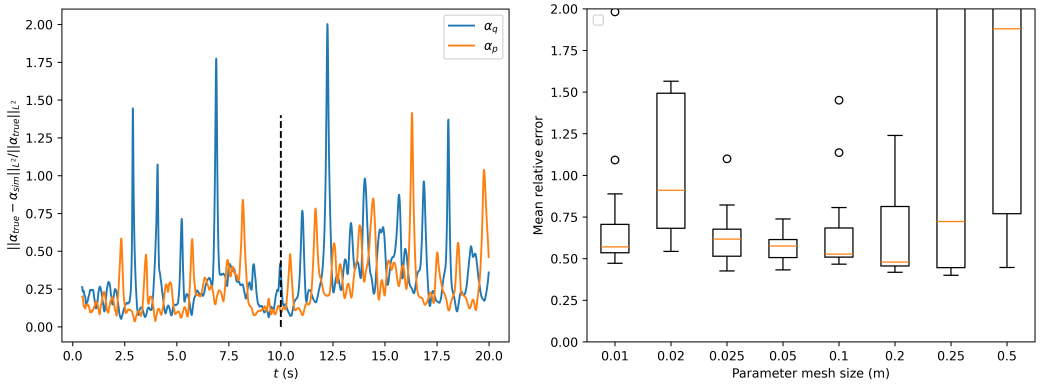

Figure 2: Study of the behaviour of PFEM-GP-dPHS as a function of the hyperparameter mesh size.

functions m_q, m_p, ℓ_q^{-1} and ℓ_p^{-1} on the same grid with a chosen mesh size $\Delta x \in \{0.01, 0.02, 0.025, 0.05, 0.1, 0.2, 0.25, 0.5\}$. Accounting for σ_f and σ_{noise}, this yields a hyperparameter θ of size $2 + 4N_{\Delta x}$, where $N_{\Delta x} = 1 + \Delta x^{-1} \in \{101, 51, 41, 21, 11, 6, 5, 3\}$. Given this mesh size, we run $N_{\text{opt}} = 20$ L-BFGS-B NLML minimisations, with an initial guess chosen at random in $[1, 2]^{2+4N_{\Delta x}}$. This yields 20 different optimized values θ_i^*, $1 \leq i \leq N_{\text{opt}}$. This in turn yields N_{opt} posterior expectations $m^i_{\underline{\dot{\alpha}}}$, N_{opt} approximated dPHS equations $\underline{\dot{\alpha}}(t) = m^i_{\underline{\dot{\alpha}}}(\underline{\alpha}(t), \underline{u}(t))$ and N_{opt} solutions (α_q^i, α_p^i). For each i, we compare α_p^i and the `SCRIMP` output α_p^{SCP} : at each time step $t_j \in \{0, \dots, N_t\}$, we compute the relative L^2 spatial error $e_{i,j} = \|\alpha_p^i(t_j) - \alpha_p^{\text{SCP}}(t_j)\|_{L^2} / \|\alpha_p^{\text{SCP}}(t_j)\|_{L^2}$ (likewise for α_q), and the spatial error averaged over time $e_i = \frac{1}{N_t} \sum_{j=0}^{N_t} e_{i,j}$ for α_p. Figure 2 (left) shows an example of $(e_{i,1}, \dots, e_{i,N_t})$ for $\Delta x = 0.1$. Running this whole procedure we obtain, for each mesh size Δx, and a sample $(e_1, \dots, e_{N_{\text{opt}}})$. Figure 2 (right) depicts histograms of these data sets. This figure shows that our method works best when the hyperparameter mesh size is either 0.05 or 0.1, the former also being the mesh size of the reference `SCRIMP` simulation (21 mesh points on the spatial domain $[0, 1]$). This figure also shows that the L-BFGS-B optimization is the most robust for those step sizes. For the other step sizes, an average relative error higher than 1 is not uncommon, meaning that the null predictor performs better than PFEM-GP-dPHS in those cases. In Appendix E, we discuss the use of the posterior variance provided by PFEM-GP-dPHS for active learning purposes.

6. Conclusion

By combining the physics-based prior of GP-dPHS and the PFEM discretization method, we were able to obtain a new GP prior that preserves the passivity of a given dPHS at the discrete level. Using a late lumping approach allowed us to introduce functional hyperparameters that can be later discretized independently from PFEM. By discretizing them as piecewise linear functions, PFEM-GP-dPHS is shown to be capable of learning a 1D nonlinear wave equation. The choice of a quadratic mean in the GP prior over the Hamiltonian functional allowed us to better model the Hamiltonian, allowing the identification of quadratic components of the true Hamiltonian functional (linear part of the dPHS) and separating them from non-quadratic contributions. Future works include the further study of our framework in the presence of dissipation ($\nu \neq 0$) as well as nD dPHS, both cases being compatible with the `SCRIMP` Python package.

Acknowledgments

Iain Henderson would like to thank the DRRP (Direction de la Recherche et des Ressources Pédagogiques) of ISAE-Supaéro for funding the internship of Florian Courteville. The authors would like to the reviewers for their helpful comments which improved the quality of the paper.

References

Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, pages vol. 4, no. 3, pp. 195–266, 2012. URL https://doi.org/10.1561/2200000036.

Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496:112549, 2024. URL https://doi.org/10.1016/j.jcp.2023.112549.

Thomas Beckers, Jacob Seidman, Paris Perdikaris, and George J. Pappas. Gaussian process port-Hamiltonian systems: Bayesian learning with physics prior. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1447–1453, 2022. URL https://doi.org/10.1109/CDC51059.2022.9992733.

Flávio Luiz Cardoso-Ribeiro, Denis Matignon, and Laurent Lefèvre. Partitioned finite element method for power-preserving discretization of open systems of conservation laws. *IMA Journal of Mathematical Control and Information*, 38:493–533, 2021. URL https://doi.org/10.1093/imamci/dnaa038.

Florian Courteville. Gaussian processes for infinite-dimensional port-Hamiltonian systems : Modelling, discretisation, simulation and application to Schrödinger equation. Master's thesis, KTH, Physics, 2025. URL https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A2000500&dswid=540.

Giuseppe Ferraro, Michel Fournié, and Ghislain Haine. Simulation and control of interactions in multi-physics, a Python package for port-Hamiltonian systems. *IFAC-PapersOnLine*, 58(6):119–124, 2024. URL https://doi.org/10.1016/j.ifacol.2024.08.267.

Matti Lassas and Samuli Siltanen. Can one use total variation prior for edge-preserving Bayesian inversion? *Inverse Problems*, 20(5):1537–1563, 2004. URL https://doi.org/10.1088/0266-5611/20/5/013.

Peilun Li, Kaiyuan Tan, and Thomas Beckers. NAPI-MPC: Neural accelerated physics-informed MPC for nonlinear PDE systems. In Necmiye Ozay, Laura Balzano, Dimitra Panagou, and Alessandro Abate, editors, *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pages 1230–1242. PMLR, 04–06 Jun 2025. URL https://proceedings.mlr.press/v283/li25d.html.

Lukas Marko, Jan-Frederik Mennemann, Lukasz Jadachowski, Wolfgang Kemmetmüller, and Andreas Kugi. Early-and late-lumping observer designs for long hydraulic pipelines: Application to pumped-storage power plants. *International Journal of Robust and Nonlinear Control*, 28(7): 2759–2779, 2018. URL https://doi.org/10.1002/rnc.4049.

Mahboubi Negareh, Xie Junyao, and Huang Biao. Physics-guided transfer learning for Bayesian optimization of chemical port-Hamiltonian systems. *Computers and Chemical Engineering*, page 109331, 2025. URL https://doi.org/10.1016/j.compchemeng.2025.109331.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. URL https://doi.org/10.7551/mitpress/3206.001.0001.

Olivier Roustant. Metamodeling with Gaussian processes. *HAL Id: hal-04509167*, 2024. URL https://hal.science/hal-04509167v2.

Andrew M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010. URL https://doi.org/10.1017/S0962492910000061.

Kaiyuan Tan, Peilun Li, and Thomas Beckers. Physics-constrained learning of PDE systems with uncertainty quantified port-Hamiltonian models. In Alessandro Abate, Mark Cannon, Kostas Margellos, and Antonis Papachristodoulou, editors, *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, volume 242 of *Proceedings of Machine Learning Research*, pages 1753–1764. PMLR, 15–17 Jul 2024. URL https://proceedings.mlr.press/v242/tan24a.html.

Kaiyuan Tan, Peilun Li, Jun Wang, and Thomas Beckers. Plug-and-play physics-informed learning using uncertainty quantified port-Hamiltonian models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10980–10986, 2025. URL 10.1109/ICRA55743.2025.11127683.

Arjan van der Schaft and Bernhard Maschke. Hamiltonian formulation of distributed-parameter systems with boundary energy flow. *Journal of Geometry and Physics*, 42:166–194, 2002. URL https://doi.org/10.1016/S0393-0440(01)00083-3.

PFEM-GP-dPHS : a finite element framework for combining Gaussian processes and infinite-dimensional port-Hamiltonian systems

Supplemental material

Appendix A. Study of the role of the quadratic mean

Kernel regression methods are universal approximators and thus the benefits of using a quadratic mean should be further examined.

Study of PFEM-GP-dPHS without using a quadratic mean We run the same workflow as the one we used to obtain Figure 2, without using quadratic mean functions. We also restrict the discretization step sizes for the hyperparameter functions $(\ell_q^{-1}, \ell_p^{-1})$ to $\Delta x \in \{0, 05, 0.1, 0.2, 0.25, 0.5\}$. The resulting histograms are given in Figure 3.

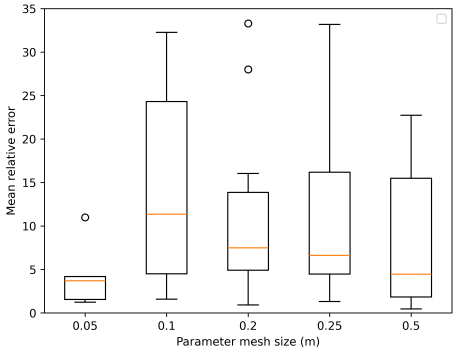

Figure 3: Study of the behaviour of PFEM-GP-dPHS as a function of the hyperparameter mesh size, without using a quadratic mean.

Numerical experiments show that without using a quadratic mean, the NLML minimisation step rarely converges to a satisfying solution $(\ell_q^{-1}, \ell_p^{-1}, \sigma_f, \sigma_{\text{noise}})$. Quantitatively, we can see in Figure 3 that, even for $\Delta x = 0.05$, we often obtain a time-averaged relative L^2 error larger than 1. This provides evidence that, given a fixed learning set, using a quadratic mean greatly increases the approximation quality of the Hamiltonian and more importantly, of the Hamiltonian dynamics (equations (5)). In fact, given the high dimensionality of the discretized dPHS from Section 5 ($d = N_p + N_q = 42$ or 82) and the relatively high number of training points (840), we expect that from a computational perspective, obtaining a reliable discretized dPHS may become close to intractable without using a quadratic mean. To finish, this issue will become even more prohibitive for dPHS with multiple space dimensions.

Enforcing the boundedness of the solution $\boldsymbol{\alpha}$ in the phase space On a theoretical level, ensuring that $\mathcal{H}(\boldsymbol{\alpha}) \to +\infty$ (and that $\mathcal{H}$ is bounded from below) when $\|\boldsymbol{\alpha}\| \to +\infty$ ensures that the sublevel sets $L_s := \{\boldsymbol{\alpha} : \mathcal{H}(\boldsymbol{\alpha}) \leq s\}$ are bounded for all $s \in \mathbb{R}$. This implies that the solutions to the associated Hamiltonian dynamics $\dot{\boldsymbol{\alpha}} = (\mathcal{J} - \mathcal{R})(\delta_{\boldsymbol{\alpha}}\mathcal{H})(\boldsymbol{\alpha})$ remain bounded in phase space, if the dPHS is well-posed in suitable function spaces. More precisely, integrating equation (5), using that

$\mathcal{R}$ is nonnegative definite and using the Cauchy-Schwarz inequality,

$$\begin{aligned}\mathcal{H}(\boldsymbol{\alpha}(t)) &= \mathcal{H}(\boldsymbol{\alpha}_0) + \int_0^t \left(- \int_{\mathcal{O}} \boldsymbol{e}(\boldsymbol{\alpha}(s)).\mathcal{R}\boldsymbol{e}(\boldsymbol{\alpha}(s)) + \langle u_\partial(s), y_\partial(s)\rangle_{\partial\mathcal{O}} \right) \mathrm{d}s \\ &\leq \mathcal{H}(\boldsymbol{\alpha}_0) + \int_0^t \langle u_\partial(s), y_\partial(s)\rangle_{\partial\mathcal{O}} \mathrm{d}s \leq \underbrace{\mathcal{H}(\boldsymbol{\alpha}_0) + \int_0^t \|u_\partial(s)\|_{L^2(\partial\mathcal{O})} \|y_\partial(s)\|_{L^2(\partial\mathcal{O})} \mathrm{d}s}_{=:C(\boldsymbol{\alpha}_0, u_\partial, t)}.\end{aligned}$$

Above, $\boldsymbol{\alpha}_0$ is the initial data and u_∂ is the control. If the dPHS is well-posed in suitable function spaces then $\|y(s)\|_{L^2(\partial\mathcal{O})}$ is a continuous function of $s, \boldsymbol{\alpha}_0$ and u_∂, and $(\boldsymbol{\alpha}_0, u_\partial, t) \mapsto C(\boldsymbol{\alpha}_0, u_\partial, t)$ is a continuous function. Thus, $\boldsymbol{\alpha}(t) \in L_{C(\boldsymbol{\alpha}_0, u_\partial, t)}$ which is a bounded set in the phase space whose growth in size is determined by the growth of $C(\boldsymbol{\alpha}_0, u_\partial, t)$. Ensuring this boundedness property at the level of the GP prior over $\mathcal{H}$ can be understood as a form of physics-informed GP model (although not all (d)PHS enjoy this boundedness property, such as the Schrödinger equation with a finite potential well). In any case, this boundedness property is usually not ensured when using standard kernels alone. For example, if the kernel k is of the form $k(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = k_S(\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|)$ where $k_S(h) \to 0$ when $h \to +\infty$, then regressors built from it will typically be of the form $\boldsymbol{\alpha} \mapsto \sum_{k=1}^N a_j k_S(\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_j\|)$, which also goes to 0 when $\|\boldsymbol{\alpha}\| \to +\infty$. This implies that the set $\{\boldsymbol{\alpha} : \mathcal{H}(\boldsymbol{\alpha}) \leq s\}$ is unbounded for all $s > 0$, potentially allowing the solution $\boldsymbol{\alpha}$ to go to infinity for new sets of inputs $(\boldsymbol{\alpha}_0, u_\partial)$.

To ensure the boundedness property above without using a quadratic mean, an alternative would be to use a non-stationary kernel e.g. of the form $k(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \sigma(\|\boldsymbol{\alpha}\|)\sigma(\|\boldsymbol{\alpha}'\|)k_S(\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|)$ where σ is a preselected function such that, for instance, $\sigma(h)k_S(h) \sim_{+\infty} Ch^\beta$ for some chosen $C, \beta > 0$. Still, this model is less "physics-informed" than the one with a quadratic mean, following the description given in Section 4.1.

Appendix B. PFEM-GP-dPHS results using polynomial approximations for the GP functional hyperparameters

We consider the same test case as the one described in Section 5. We now seek the functions m_q, m_p, ℓ_q^{-1} and ℓ_p^{-1} in the space of polynomial functions of order 3 instead of the space of piecewise linear functions. This results in $4 \times 4 + 2 = 18$ hyperparameters in total, as opposed to e.g. 26 hyperparameters estimated to obtain Figure 1. The obtained estimations are given in Figure 4, which is a counterpart of Figure 1.

Figure 4 (left) shows that ρ^{-1} is more or less well-estimated by m_p. However, the function T is not well approximated by m_q. As a result, the quadratic part of the Hamiltonian is not identified correctly by the prior mean with tuned hyperparameters. Contrarily to the piecewise linear case, this implies that the kernel k must be active even at locations where the quadratic part of the Hamiltonian is dominant. In particular, there is no reason for the correlation lengths to follow the variations of c, as shown in Figure 4 (right). Yet, Figure 7 (right) provides visual evidence that, even in this case where the kernel needs to compensate for an ill-estimated quadratic mean, the PFEM-GP-dPHS framework is still capable of providing a good estimation of α_p.

Possible extensions of this study include using other bases, such as the Fourier basis or wavelets, and studying their performance as a function of their discretization mesh size.

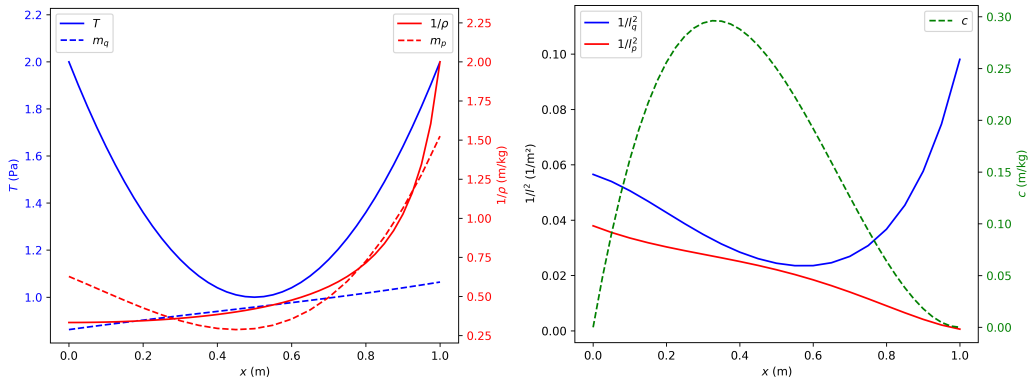

Figure 4: Left : estimation of the GP quadratic mean functions m_q and m_p using polynomial functions of order 3, versus the true physical parameters T and ρ^{-1}. Right : estimation of the GP lengthscale functions ℓ_q^{-1} and ℓ_p^{-1} using polynomial functions of order 3, versus the function c.

Appendix C. Extension to 2D dPHS

Extension of the PFEM-GP-dPHS method to the 2D case, and generally higher dimension case, is possible using the `SCRIMP` package. We show here the calculation and numerical result for the nonlinear 2D wave equation, following the same workflow as the 1D case in Sections 3, 4 and 5.

C.1. Nonlinear 2D wave equation in dPHS form

We consider the direct extension our 1D exemple, the 2D wave equation with Neumann boundary controls on a bounded rectangle $\mathcal{O}_{2D} := (0, L) \times (0, \ell)$. The deflection of the membrane from the equilibrium w satisfies

$$\begin{cases} \rho(x)\,\partial_{tt}^2 w(t,x) - \operatorname{div}\left(\overline{\overline{h}}(x, \mathbf{grad}\,(w(t,x))) \cdot \mathbf{grad}\,(w(t,x))\right) = 0, \quad t \geq 0, x \in \mathcal{O}_{2D}, \\ \qquad \boldsymbol{n} \cdot \overline{\overline{h}}(s, \mathbf{grad}\,(w(t,s))) \cdot \mathbf{grad}\,(w(t,s)) = u_N(t,s), \quad t \geq 0, s \in \Gamma = \delta\mathcal{O}_{2D}, \end{cases} \tag{25}$$

We set the energy variable $\boldsymbol{\alpha}_q := \mathbf{grad} w \in \mathbb{R}^2$ the strain, and $\alpha_p := \rho \partial_t w \in \mathbb{R}$ the linear momentum. Note that the main difference with the 1D case is that $\boldsymbol{\alpha}_q$ is now a vector-valued variable.

The term $\overline{\overline{h}}(x, \boldsymbol{\alpha}_q(t,x)) = \overline{\overline{T}}(x) + \overline{\overline{c}}(x)\, g(\boldsymbol{\alpha}_q(t,x))$ is made of a linear term (Elasticity tensor) $\overline{\overline{T}}(x) \in \mathbb{R}^{2\times 2}$, and a nonlinear term $\overline{\overline{c}}(x)\, g(\boldsymbol{\alpha}_q(.,x))$ with $\overline{\overline{c}}(x) \in \mathbb{R}^{2\times 2}$ and $g(\boldsymbol{u}) = \exp(-\boldsymbol{u}\cdot\boldsymbol{u})$. One can express the total mechanical energy $\mathcal{H}$, the Hamiltonian, as

$$\mathcal{H}(t) = \mathcal{H}(\boldsymbol{\alpha}_q(t,x), \alpha_p(t,x)) := \underbrace{\frac{1}{2}\int_{\mathcal{O}_{2D}} \frac{\alpha_p(t,x)^2}{\rho(x)} \mathrm{d}x}_{\text{Kinetic energy}} + \underbrace{\frac{1}{2}\int_{\mathcal{O}_{2D}} \boldsymbol{\alpha}_q^\top(t,x)\overline{\overline{T}}(x)\boldsymbol{\alpha}_q(t,x)\mathrm{d}x}_{\text{Quadratic potential energy}}$$

$$+ \underbrace{\frac{1}{2}\int_{\mathcal{O}_{2D}} \int_{0<y<\alpha_{q1}(t,x),\, 0<z<\alpha_{q2}(t,x)} g\left(\begin{bmatrix} y \\ z \end{bmatrix}\right) \begin{bmatrix} y & z \end{bmatrix} \overline{\overline{c}}(x) \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}z \end{bmatrix}}_{\text{Nonlinear term}}. \tag{26}$$

The equation above is a 2D generalization of equation (24). The co-energy variables are, as in the 1D case $\boldsymbol{e}_q := \delta_{\boldsymbol{\alpha}_q}\mathcal{H} = \overline{\overline{h}} \cdot \boldsymbol{\alpha}_q$ the stress, and $e_p := \delta_{\alpha_p}\mathcal{H} = \frac{\alpha_p}{\rho}$ the velocity.

Using Newton's second law and Schwarz's lemma, we obtain port-Hamiltonian system representing a vibrating membrane with Neumann boundary controls :

$$\begin{pmatrix} \partial_t \boldsymbol{\alpha}_q \\ \partial_t \alpha_p \end{pmatrix} = \begin{bmatrix} 0 & \mathbf{grad} \\ \mathrm{div} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{e}_q \\ e_p \end{pmatrix}, \qquad \left\{ \begin{array}{rcll} \boldsymbol{e}_q(t,s)\cdot \boldsymbol{n} & = & u(t,s), & t\geq 0,\ s\in\Gamma, \\ y(t,s) & = & e_p(t,s), & t\geq 0,\ s\in\Gamma. \end{array} \right. \tag{27}$$

The power balance satisfied by the Hamiltonian is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}(\boldsymbol{\alpha}_q(t), \alpha_p(t)) = \underbrace{\langle y(t,\cdot), u(t,\cdot)\rangle_\Gamma}_{\text{power flowing through } \Gamma}. \tag{28}$$

C.2. PFEM

Now let apply the PFEM on this 2D problem. Let $\boldsymbol{\varphi}_q$ and φ_p be smooth test functions on $\mathcal{O}_{2D}$, and ψ be smooth test functions on Γ . One can write the weak formulation of the Dirac structure as follows :

$$\left\{ \begin{array}{rcl} \int_{\mathcal{O}_{2D}} \partial_t \boldsymbol{\alpha}_q(t,x)\cdot\boldsymbol{\varphi}_q(x)\mathrm{d}x & = & \int_{\mathcal{O}_{2D}} \mathbf{grad}\left(e_p(t,x)\right)\cdot\boldsymbol{\varphi}_q(x)\mathrm{d}x, \\ \int_{\mathcal{O}_{2D}} \partial_t \alpha_p(t,x)\varphi_p(x)\mathrm{d}x & = & \int_{\mathcal{O}_{2D}} \mathrm{div}\left(\boldsymbol{e}_q(t,x)\right)\varphi_p(x)\mathrm{d}x, \\ \langle y,\psi\rangle_\Gamma & = & \langle e_p,\psi\rangle_\Gamma . \end{array} \right. \tag{29}$$

Integrating the second line by parts makes the control u and the observation y appear:

$$\int_{\mathcal{O}_{2D}} \partial_t \alpha_p(t,x)\varphi_p(x)\mathrm{d}x = -\int_{\mathcal{O}_{2D}} \boldsymbol{e}_q(t,x)\cdot\mathbf{grad}\left(\varphi_p(x)\right)\mathrm{d}x + \langle \varphi_p, u\rangle_\Gamma \tag{30}$$

Let $(\boldsymbol{\varphi}_q^i)_{1\leq i\leq N_q} \subset \left(L^2(\mathcal{O}_{2D})\right)^2$ and $(\varphi_p^k)_{1\leq k\leq N_p} \subset H^1(\mathcal{O}_{2D})$ be two finite families of approximations so that $\boldsymbol{\alpha}_q(t,x) \simeq \underline{\varphi_q}^\top(x)\underline{\alpha_q}(t)$, $\alpha_p(t,x) \simeq \underline{\varphi_p}^\top(x)\underline{\alpha_p}(t)$ (likewise for $\dot{\boldsymbol{\alpha}}_q$ and $\dot{\alpha}_p$) and $\boldsymbol{e}_q(t,x) \simeq \underline{\varphi_q}^\top(x)\underline{e_q}(t)$, $e_p(t,x) \simeq \underline{\varphi_p}^\top(x)\underline{e_p}(t)$. One thing to remark is that compared to the 1D case, $\underline{\varphi_q}^\top = [\boldsymbol{\varphi}_q^1, \cdots, \boldsymbol{\varphi}_q^{N_q}] \in \left(L^2(\mathcal{O}_{2D})\right)^{2\times N_q}$ is now a matrix. We denote also $(\psi^m)_{1\leq m\leq N} \subset H^{\frac{1}{2}}(\Gamma)$. In particular, the latter choices imply that the duality brackets at the boundary reduce to simple L^2 scalar products. We next replace the variables by their approximations, and write the system in matrix form

$$\left\{ \begin{array}{rcl} \underbrace{\begin{bmatrix} M_q & 0 \\ 0 & M_p \end{bmatrix}}_{M} \underbrace{\begin{pmatrix} \underline{\dot{\alpha}_q}(t) \\ \underline{\dot{\alpha}_p}(t) \end{pmatrix}}_{\underline{\dot{\alpha}}} & = & \underbrace{\begin{bmatrix} 0 & D \\ -D^\top & 0 \end{bmatrix}}_{J-R} \underbrace{\begin{pmatrix} \underline{e_q}(t) \\ \underline{e_p}(t) \end{pmatrix}}_{\underline{e}} + \underbrace{\begin{bmatrix} 0 \\ B \end{bmatrix}}_{G} \underline{u}, \\ \underline{y}(t) & = & \begin{bmatrix} 0 & B^\top \end{bmatrix} \begin{pmatrix} \underline{e_q}(t) \\ \underline{e_p}(t) \end{pmatrix}, \end{array} \right. \tag{31}$$

with square matrices $(M_q)_{ij} := \int_{\mathcal{O}_{2D}} \boldsymbol{\varphi}_q^j(x)\cdot\boldsymbol{\varphi}_q^i(x)\mathrm{d}x$, $(M_p)_{k\ell} := \int_{\mathcal{O}_{2D}} \varphi_p^\ell(x)\varphi_p^k(x)\mathrm{d}x$ and $(M_\Gamma)_{k\ell} := \int_\Gamma \psi^\ell(s)\psi^k(s)\mathrm{d}s$, and rectangular matrices $(D)_{i\ell} := \int_{\mathcal{O}_{2D}} \mathbf{grad}\left(\varphi_p^\ell(x)\right)\cdot\boldsymbol{\varphi}_q^i(x)\mathrm{d}x$ and $(B)_{k\ell} := \int_\Gamma \varphi_p^\ell(s)\psi^k(s)\mathrm{d}s$. The system is closed by approximating the constitutive relations like in 1D by writing them in their weak form:

$$\left\{\begin{array}{rcl} \int_{\mathcal{O}_{2D}} \boldsymbol{e}_q(t,x)\cdot\boldsymbol{\varphi}_q(x)\mathrm{d}x &=& \int_{\mathcal{O}_{2D}} \left(\overline{\overline{h}}(x,\boldsymbol{\alpha}_q(t,x))\,\boldsymbol{\alpha}_q(t,x)\right)\cdot\boldsymbol{\varphi}_q(x)\mathrm{d}x, \\ \int_{\mathcal{O}_{2D}} e_p(t,x)\varphi_p(x)\mathrm{d}x &=& \int_{\mathcal{O}_{2D}} \frac{1}{\rho(x)}\alpha_p(t,x)\varphi_p(x)\mathrm{d}x. \end{array}\right. \tag{32}$$

Then the matrix form of the discrete weak formulation of the constitutive relations is

$$\left\{\begin{array}{rcl} M_q\underline{e_q}(t) &=& M_h\underline{\alpha_q}(t), \\ M_p\underline{e_p}(t) &=& M_\rho\underline{\alpha_p}(t), \end{array}\right. \tag{33}$$

where $(M_h)_{ij} := \int_{\mathcal{O}_{2D}} \boldsymbol{\varphi}_q^j(x)\cdot\overline{\overline{h}}(x,\boldsymbol{\alpha}_q(t,x))\cdot\boldsymbol{\varphi}_q^i(x)\mathrm{d}x$ and $(M_\rho)_{k\ell} := \int_{\mathcal{O}_{2D}} \frac{1}{\rho(x)}\varphi_p^\ell(x)\varphi_p^k(x)\mathrm{d}x$.

C.3. PFEM-GP-dPHS

To write the associated PFEM-GP-dPHS as in Section 4, we use a scalar prior mean m and kernel k selected for the GP of $\widehat{\mathcal{H}}$, such that $\widehat{\mathcal{H}} \sim \mathcal{GP}(m,k)$, with

$$m(\boldsymbol{\alpha}) = \int_{\mathcal{O}_{2D}} \boldsymbol{\alpha}_q^\top(x)A_q(x)\boldsymbol{\alpha}_q(x)\,dx + \int_{\mathcal{O}_{2D}} m_p(x)\alpha_p^2(x)\,dx, \tag{34}$$

$$k(\boldsymbol{\alpha},\boldsymbol{\alpha}') = \sigma_f^2\exp\Bigg[-\frac{1}{2}\Bigg(\int_{\mathcal{O}_{2D}} (\boldsymbol{\alpha}_q(x)-\boldsymbol{\alpha}_q'(x))^\top L_q^{-2}(x)(\boldsymbol{\alpha}_q(x)-\boldsymbol{\alpha}_q'(x))\,dx + \int_{\mathcal{O}_{2D}} \frac{(\alpha_p(x)-\alpha_p'(x))^2)}{\ell_p^2(x)}\,dx\Bigg)\Bigg], \tag{35}$$

where the functions m_p and ℓ_p are 2D nonnegative hyperparameters, and the PSD-matrix-valued function $A_q(x) = \left[[m_{q1}(x)\ m_{q3}(x)]^\top\ \ [m_{q3}(x)\ m_{q2}(x)]^\top\right]$ and diagonal matrix-valued function $L_q(x) = \mathrm{Diag}(\ell_{q1}(x),\ell_{q2}(x)) \in \mathbb{R}^{2\times 2}$ are matrix hyperparameters of the GP $\widehat{\mathcal{H}}$.

Next, we adapt the affine transformation of equation (14) we need to obtain the GP distribution of e:

$$M\underline{e}(\boldsymbol{\alpha}) = \int_{\mathcal{O}_{2D}} \underbrace{\begin{bmatrix} \underline{\varphi_q}(x) & 0_{N_q,1} \\ 0_{N_p,2} & \underline{\varphi_p}(x) \end{bmatrix}}_{\Phi(x)} (\delta_{\boldsymbol{\alpha}}\widehat{\mathcal{H}})(\boldsymbol{\alpha})(x)\mathrm{d}x, \tag{36}$$

where now $\Phi \in \mathbb{R}^{(N_q+N_p)\times 3}$. We apply the variational derivative $\delta_{\boldsymbol{\alpha}} = [\delta_{\boldsymbol{\alpha}_q},\delta_{\alpha_p}]^\top$ to obtain the prior GP distribution over e

$$m_e(\boldsymbol{\alpha}) = \begin{bmatrix} A_q\boldsymbol{\alpha}_q \\ m_p\alpha_p \end{bmatrix}, \tag{37}$$

$$\frac{k_e(\boldsymbol{\alpha},\boldsymbol{\alpha}')}{k(\boldsymbol{\alpha},\boldsymbol{\alpha}')} = \begin{bmatrix} \mathrm{Diag}\left(\frac{1}{\ell_{q1}^2}\mathcal{I},\frac{1}{\ell_{q2}^2}\mathcal{I}\right) & 0 \\ 0 & \frac{1}{\ell_p^2}\mathcal{I} \end{bmatrix} \tag{38}$$

$$- \begin{bmatrix} L_q^{-2}(\boldsymbol{\alpha}_q-\boldsymbol{\alpha}_q')\otimes L_q^{-2}(\boldsymbol{\alpha}_q-\boldsymbol{\alpha}_q')^\top & L_q^{-2}(\boldsymbol{\alpha}_q-\boldsymbol{\alpha}_q')\otimes\frac{1}{\ell_p^2}(\alpha_p-\alpha_p') \\ \frac{1}{\ell_p^2}(\alpha_p-\alpha_p')\otimes L_q^{-2}(\boldsymbol{\alpha}_q-\boldsymbol{\alpha}_q')^\top & \frac{1}{\ell_p^2}(\alpha_p-\alpha_p')\otimes\frac{1}{\ell_p^2}(\alpha_p-\alpha_p') \end{bmatrix} \tag{39}$$

Finally, we multiply by Φ, integrate and replace $\boldsymbol{\alpha}$ by its finite-dimensional approximation $\Phi^\top \underline{\alpha}$ to obtain the prior over $\underline{e}(\underline{\alpha})$

$$m_{\underline{e}}(\underline{\alpha}) = \begin{bmatrix} (M_{m,q}\underline{\alpha_q})^\top & (M_{m,p}\underline{\alpha_p})^\top \end{bmatrix}^\top, \tag{40}$$

with $(M_{m,q})_{ij} := \int_{\mathcal{O}_{2D}} \boldsymbol{\varphi}_q^{i\top} A_q \boldsymbol{\varphi}_q^j \mathrm{d}x$ and $(M_{m,p})_{ij} := \int_{\mathcal{O}_{2D}} m_p \varphi_p^i \varphi_p^j \mathrm{d}x$ and

$$\frac{k_{\underline{e}}(\underline{\alpha}, \underline{\alpha}')}{k(\Phi^\top \underline{\alpha}, \Phi^\top \underline{\alpha}')} = M_k \begin{bmatrix} I_{N_q} - (\underline{\alpha_q} - \underline{\alpha_q'})(\underline{\alpha_q} - \underline{\alpha_q'})^\top & -(\underline{\alpha_q} - \underline{\alpha_q'})(\underline{\alpha_p} - \underline{\alpha_p'}) \\ -(\underline{\alpha_p} - \underline{\alpha_p'})(\underline{\alpha_q} - \underline{\alpha_q'})^\top & I_{N_p} - (\underline{\alpha_p} - \underline{\alpha_p'})(\underline{\alpha_p} - \underline{\alpha_p'})^\top \end{bmatrix} M_k^\top, \tag{41}$$

where $M_k = M^{-1}\mathrm{Diag}(M_{k,q}, M_{k,p})$, where the submatrices $M_{k,q}$ and $M_{k,p}$ are given by $(M_{k,q})_{ij} := \int_{\mathcal{O}_{2D}} \boldsymbol{\varphi}_q^{i\top} L_q^{-2} \boldsymbol{\varphi}_q^j \mathrm{d}x$ and $(M_{k,p})_{ij} := \int_{\mathcal{O}_{2D}} \ell_p^{-2} \varphi_p^i \varphi_p^j \mathrm{d}x$.

Finally to obtain the GP prior over $f_{\underline{\alpha}}$, like in 1D we apply the matrix multiplication $(J - R)$ to the GP $\underline{e}$ and add the mean $G\underline{u}$.

C.4. Generalization to higher dimensions

From here, we can remark that the only difference between the 1D PFEM-GP-dPHS and 2D PFEM-GP-dPHS lies only in the matrix $M_{m,q}$,$M_{m,p}$,$M_{k,q}$ and $M_{k,p}$, as equations (21) and (40), and also equations (22) and (41), are the same. From here, it is easy to see than for any dPHS over a space $\mathcal{O}_{nD}$ of dimension nD and with vector-valued energy variables of size n_q and n_p, the associated PFEM-GP-dPHS will use the mean and kernel functions given in equations (21) and (22), but with

$$\begin{aligned}
(M_{m,q})_{ij} &:= \int_{\mathcal{O}_{nD}} \boldsymbol{\varphi}_q^{i\top}(x) A_q(x) \boldsymbol{\varphi}_q^j(x) \mathrm{d}x, \\
(M_{m,p})_{ij} &:= \int_{\mathcal{O}_{nD}} \boldsymbol{\varphi}_p^{i\top}(x) A_p(x) \boldsymbol{\varphi}_p^j(x) \mathrm{d}x, \\
(M_{k,q})_{ij} &:= \int_{\mathcal{O}_{nD}} \boldsymbol{\varphi}_q^{i\top}(x) L_q^{-2}(x) \boldsymbol{\varphi}_q^j(x) \mathrm{d}x, \\
(M_{k,p})_{ij} &:= \int_{\mathcal{O}_{nD}} \boldsymbol{\varphi}_p^{i\top}(x) L_p^{-2}(x) \boldsymbol{\varphi}_p^j(x) \mathrm{d}x,
\end{aligned}$$

where A_q and A_p are PSD-symmetric matrix-valued function of size respectively n_q and n_p and L_q and L_p are diagonal matrix-valued function of size respectively n_q and n_p. From here, those new hyperparameter functions can, like in 1D, be discretized using their own finite element method, or even other kind of decomposition.

C.5. Numerical results

We test PFEM-GP-dPHS on the dPHS described by equation (27), using the same numerical workflow as for the 1D equation in Section 5. The space $\mathcal{O}_{2D}$ is meshed with 22 elements. $\boldsymbol{\alpha}_q$ is discretized with $\mathbb{P}^1$ Lagrange elements giving $N_q = 132$ degrees of freedom, and α_p is discretized with $\mathbb{P}^2$ Lagrange elements giving $N_q = 57$ degrees of freedom. We also discretize the boundary terms u and y with 1D $\mathbb{P}^1$ Lagrange elements, and use a simulation time step $\Delta t = 0.01$s. The parameters of the dPHS are taken constant with $\rho^{-1} = 0.25$, $\overline{\overline{T}} = \begin{bmatrix} [5\ 2]^\top & [2\ 7]^\top \end{bmatrix}$ and

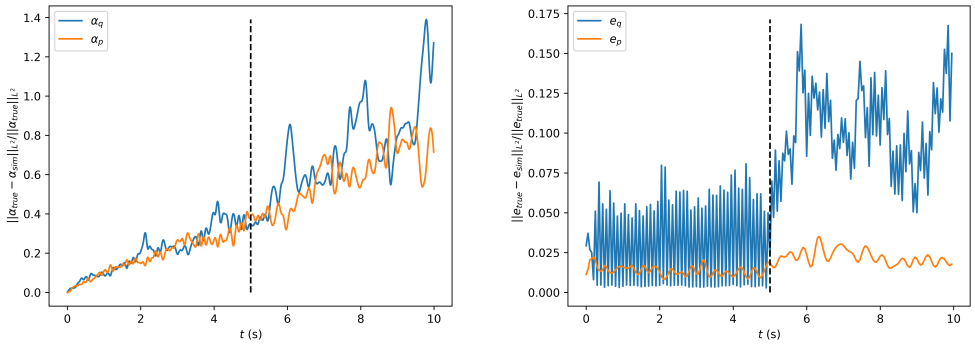

Figure 5: (Left) Evolution of the relative error of the $\boldsymbol{\alpha}_q$ and α_p trajectories over time. (Right) Relative error of the posterior mean of $\boldsymbol{e}_q$ and e_p calculated at each point of the true trajectory $\boldsymbol{\alpha}(t)$. The dotted black line represents the time where we stop using time step for input data.

$\overline{\overline{c}} = \left[[2\ 1]^\top\ [1\ 2.5]^\top\right]$. The hyperparameters are also taken to be constant over the space, giving us as hyperparameters $A_q = \left[[m_{q1}\ m_{q3}]^\top\ [m_{q3}\ m_{q2}]^\top\right]$, $L_q = \text{Diag}(\ell_{q1}, \ell_{q2})$, m_p, ℓ_p^{-1}, σ_f and the noise, thus 9 hyperparameters. We use SCRIMP to generate 10s of simulation with all controls u fixed to 0 and with initial condition $\boldsymbol{\alpha}_q = [0.1x(x-2)y(y-1), -0.1x(x-2)y(y-1)]^\top$ and $\alpha_p = 3^{(-20((x-0.5)(x-0.5)+(y-0.5)(y-0.5)))}$. Then we select $N_s = 55$ time steps within 0 to 5 seconds to compute our training data, resulting in $N_{train} = (N_q + N_p) \times N_s = 10395$ training points. Using the same method as for the 1D case, we present in Figure 5 and 6 the result of a successful regression. Here we obtained $A_q = \left[[0.609\ 0.606]^\top\ [0.606\ 0.741]^\top\right]$, $L_q = \left[[2.97\ 0]^\top\ [0\ 2.05]^\top\right]$, $m_p = 0.126$, $\ell_p = 16.10$ and $\sigma_f = 0.879$. Figure 5 (left) shows that the relative errors of the calculated trajectory versus the true trajectory of SCRIMP α^{SCP}, $e^{\text{rel}}_{mj} = \|\alpha_m(t_j) - \alpha^{\text{SCP}}_m(t_j)\|_{L^2}/\|\alpha^{\text{SCP}}_m(t_j)\|_{L^2}$ $(m = q, p)$, slowly increase but stay below 50% within the time interval during which we measured training data. After the 5s mark, we can see that the relative error rises faster as the trajectory reaches parts of the $\boldsymbol{\alpha}$ space on which the GP has not been trained. However, this divergent behaviour stays relatively slow, which can be explained by the right part of Figure 5. Here, we see that the relative error of the posterior mean of the GP $\underline{e}$, $\tilde{m}_{\underline{e}}$ over the true trajectory, $e^{\text{rel,e}}_{mj} = \|\tilde{m}_{\underline{e},m}(\alpha^{\text{SCP}}(t_j)) - e^{\text{SCP}}_m(\alpha^{\text{SCP}}(t_j))\|_{L^2}/\|e^{\text{SCP}}_m(\alpha^{\text{SCP}}(t_j))\|_{L^2}$ $(m = q, p)$, is low before 5 seconds as it is where the training points are located, but also not very high after 5 seconds. This prevents the trajectory from diverging too fast from the SCRIMP simulation for "large" times, and also underlines the capacity of the PFEM-GP-dPHS to approximate the Hamiltonian in places of the $\boldsymbol{\alpha}$ space with no training points in it. Figure 6 shows a representation of the estimated and true energy variable at $t = 1$ (α_{q1} and α_p), to have a qualitative appreciation of the quality of the regression.

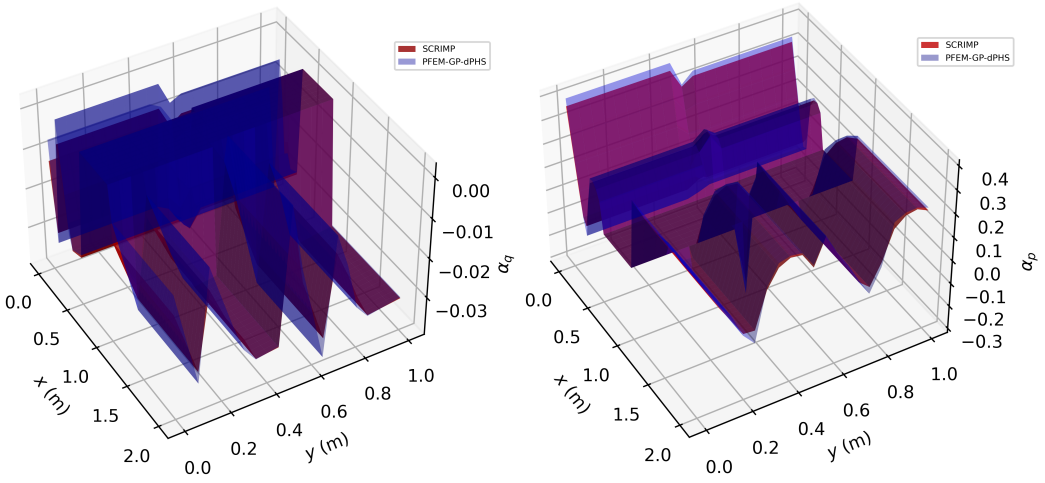

Figure 6: Representation of the true and simulated (left) α_{q1} and (right) α_p at time $t = 1s$ for for the 2D case.

Appendix D. Computational costs and additional 1D simulation figures

D.1. Additional 1D simulation figures

Figures 7 and 8 present additional 1D simulation outputs, showing the trajectory of α_p or α_q over time for the SCRIMP simulation and the predicted trajectory after regression with the PFEM-GP-dPHS method. Figure 7 shows two successful predicted trajectories using 2 different decomposition methods: piecewise linear functions and 3rd order polynomial functions. The analysis of this result is done in Appendix B. Figure 8 illustrates the behaviour of PFEM-GP-dPHS as a function of the number of finite elements: using a finer finite element mesh for the energy variables allows to obtain smoother trajectories after regression.

D.2. Computational costs

The computational cost of the method is dominated by the inversion of the square kernel Gram matrix of size $(N_q + N_p)N_s$ using its Cholesky decomposition, for the calculation of the NLML resulting in cost of $O(((N_q + N_p)N_s)^3)$. Moreover, when calculating the gradient of the NLML, one needs one matrix multiplication of matrix of size $(N_q + N_p)N_s$ per hyperparameters of type ℓ_q, ℓ_p, σ_f and σ_{noise}. This gives a gradient cost dominated by $O((N_{\theta q} + N_{\theta p} + 2)((N_q + N_p)N_s)^3)$ where $N_{\theta q}$ and $N_{\theta p}$ are the number of hyperparameters of type ℓ_q and ℓ_p. This shows the advantage of using late lumping : using early lumping would imply $N_{\theta q} = N_q$ and $N_{\theta p} = N_p$ whereas the late lumping approach allows us to use lower dimensional discretizations of the functional hyperparameters. This reduces the number of hyperparameters to optimize, reduces the computational complexity of NLML optimization, and accelerates the whole computational procedure. Using an 8 year old personal laptop with a processor `Intel Core i7-6700HQx8`, 20 minutes were needed to obtain the results of Figure 1, 1 hour for Figure 7 (right), 10 minutes for the polynomial approximation of Figure 4 and 2 hours and 30 minutes for the 2D case of Figure 5.

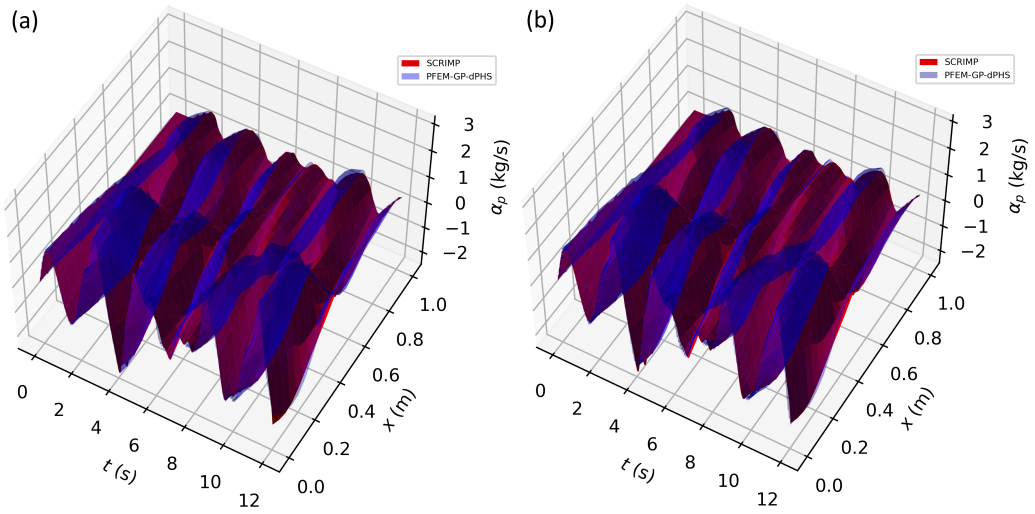

Figure 7: True α_p trajectory obtained with `SCRIMP` plotted with the estimated α_p trajectory obtained using the posterior mean of the PFEM-GP-dPHS and discretizing hyperparameters with (left) piecewise linear functions (right) 3 order polynomial functions.

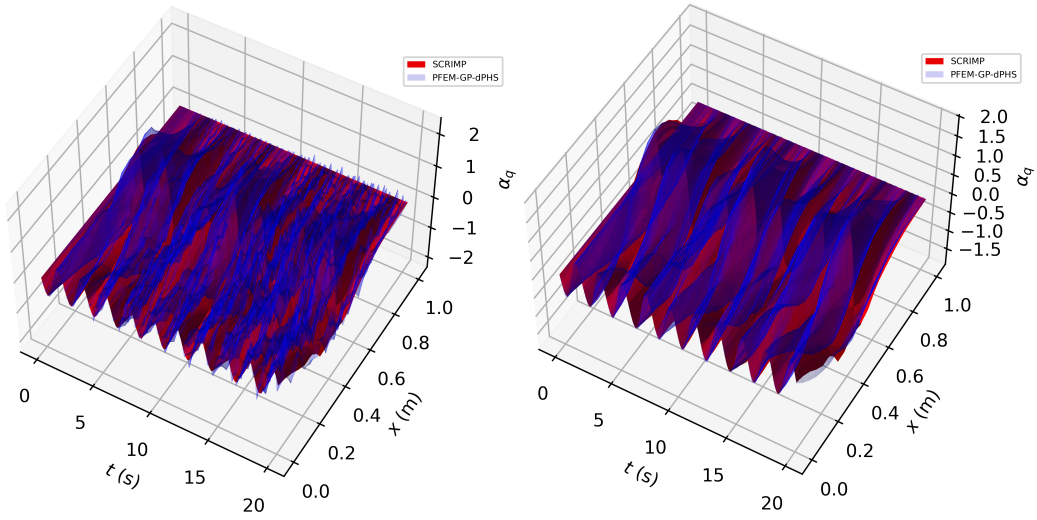

Figure 8: True α_q trajectory obtained with `SCRIMP` plotted with the estimated α_q trajectory obtained using the posterior mean of the PFEM-GP-dPHS and discretizing hyperparameters. Left: a result using the same parameters and hyperparameters as those of Figure 1 ($N_q = N_p = 21$ and $\Delta x = 0.2$). Right: a result using finer parameters, $N_q = N_p = 41$ and $\Delta x = 0.1$ resulting in 46 hyperparameters ($N_s = 35$ for both). This shows that using a larger number of finite elements (right) for α_q and α_p allows to obtain smoother simulated trajectories.

Appendix E. Exploiting the posterior variance provided by PFEM-GP-dPHS

Figure 9 shows an example of the *absolute* L^2 errors $e^{\text{abs}}_{i,j} = \|\alpha^i_p(t_j) - \alpha^{\text{SCP}}_p(t_j)\|_{L^2}$ for some example i as a function of time, as opposed to Figure 2 (left) which shows *relative* errors (see

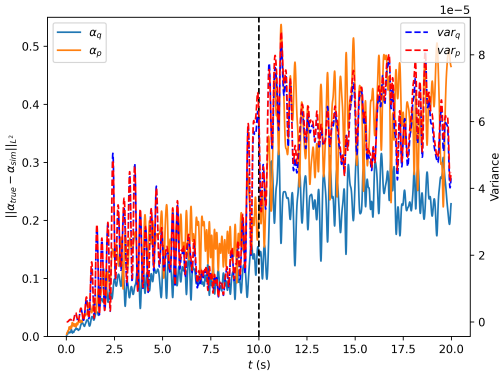

Figure 9: Evolution of $\mathrm{var}_q(t)$, $\mathrm{var}_p(t)$ and the L^2 absolute errors as functions of time.

Section 5 for further details on the definition of the quantities above). Figure 9 also shows the quantities $\mathrm{var}_q(t)$ and $\mathrm{var}_p(t)$, which are given in equations (42) and (43) below :

$$\mathrm{var}_q(t) := \mathrm{Tr}\Big(\mathbb{C}\mathrm{ov}\big[\widetilde{\underline{e}}_q(\underline{\alpha}(t))\big]\Big) = \mathbb{E}\big[\|\widetilde{\underline{e}}_q(\underline{\alpha}(t)) - \mu_q(t)\|_2^2\big], \quad \mu_q(t) = \mathbb{E}[\widetilde{\underline{e}}_q(\underline{\alpha}(t))]\,, \tag{42}$$

$$\mathrm{var}_p(t) := \mathrm{Tr}\Big(\mathbb{C}\mathrm{ov}\big[\widetilde{\underline{e}}_p(\underline{\alpha}(t))\big]\Big) = \mathbb{E}\big[\|\widetilde{\underline{e}}_p(\underline{\alpha}(t)) - \mu_p(t)\|_2^2\big], \quad \mu_p(t) = \mathbb{E}[\widetilde{\underline{e}}_p(\underline{\alpha}(t))]\,. \tag{43}$$

Above, $(\widetilde{\underline{e}}_q(\underline{\alpha}))_{\underline{\alpha}\in\mathbb{R}^{N_q+N_p}}$ is the GP obtained by conditioning the GP $\underline{e}_q(\underline{\alpha})$ on the training data. Recall that $(\underline{e}(\underline{\alpha}))_{\underline{\alpha}\in\mathbb{R}^{N_q+N_p}}$ is the vector-valued GP with mean and kernel functions given in equations (21) and (22) respectively, and $\underline{e}(\underline{\alpha}) = \big(\underline{e}_q(\underline{\alpha}), \underline{e}_p(\underline{\alpha})\big)$. The equality $\mathrm{Tr}(\mathbb{C}\mathrm{ov}[\widetilde{\underline{e}}_q(\underline{\alpha}(t))]) = \mathbb{E}[\|\widetilde{\underline{e}}_q(\underline{\alpha}(t)) - \mu_q(t)\|_2^2]$ (likewise for $\widetilde{\underline{e}}_p$) are justified at the end of this section. Figure 9 shows that the absolute error plots behave approximately like the quantities $\mathrm{var}_q(t)$ and $\mathrm{var}_p(t)$. In particular, all four curves increase after the 10s landmark, which corresponds to the time at which we stopped collecting measurement data in the `SCRIMP` simulation to train our PFEM-GP-dPHS model. The knowledge of this posterior variance can be used to perform adaptive learning for our PFEM-GP-dPHS model, as dexcribed in (Courteville, 2025, Section 5.4) in a (finite-dimensional) GP-PHS context.

Note that in Figure 9 shows error plots about the energy variables $(\underline{\alpha}_q, \underline{\alpha}_p)$, while the variance plots concern the *coenergy* variables $(\underline{e}_q, \underline{e}_p)$ and not $(\underline{\alpha}_q, \underline{\alpha}_p)$. This is because we have no GP model over $\underline{\alpha} = (\underline{\alpha}_q, \underline{\alpha}_p)$; they are the inputs of the GP $(\underline{e}(\underline{\alpha}))_{\underline{\alpha}\in\mathbb{R}^{N_q+N_p}}$! However, after training the GP $\underline{e}(\underline{\alpha})$ and during a PFEM-GP-dPHS simulation, the variance of the conditioned GP $\widetilde{\underline{e}}$ *can* be computed online at points $\underline{\alpha}(t)$, for any $t \geq 0$.

We finish this appendix with a proof of the equalities given in equations (42) and (43). For this, consider a random vector $X \in \mathbb{R}^N$ with mean m and covariance matrix Σ, then using that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ if A and B have suitable dimensions,

$$\begin{aligned}\mathrm{Tr}\,\Sigma &= \mathrm{Tr}\,\mathbb{E}[(X-m)(X-m)^\top] = \mathbb{E}\Big[\mathrm{Tr}\Big((X-m)(X-m)^\top\Big)\Big] \\ &= \mathbb{E}\Big[\mathrm{Tr}\Big((X-m)^\top(X-m)\Big)\Big] = \mathbb{E}\big[\mathrm{Tr}\big(\|X-m\|_2^2\big)\big] = \mathbb{E}\big[\|X-m\|_2^2\big]\,.\end{aligned}$$

To obtain the equalities in equations (42) and (43), use the result above with $(N, X) = (N_q, \widetilde{\underline{e}}_q(\underline{\alpha}(t))$ and $(N, X) = (N_p, \widetilde{\underline{e}}_p(\underline{\alpha}(t)))$.