

A Reduced Order Modeling Method with Variable-Separation-Based Domain Decomposition for Parametric Dynamical Systems

Yuming Ba¹, Liang Chen^{2,3}, Yaru Chen^{2,3,*} and Qiuqi Li^{2,3}

¹*School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China.*

²*School of Mathematics, Hunan University, Changsha 410082, China.*

³*Hunan Provincial Key Laboratory of Intelligent Information Processing and Applied Mathematics, Changsha 410082, China.*

Received 3 June 2025; Accepted (in revised version) 28 October 2025.

Abstract. This paper proposes a model order reduction method for a class of parametric dynamical systems. Using a temporal Fourier transform, we reformulate these systems into complex-valued elliptic equations in the frequency domain, containing frequency variables and parameters inherited from the original model. To reduce the computational cost of the frequency-variable elliptic equations, we extend the variable-separation-based domain decomposition method to the complex-valued context, resulting in an offline-online procedure for solving the parametric dynamical systems. At the offline stage, separate representations of the solutions for the interface problem and the subproblems are constructed. At the online stage, the solutions of the parametric dynamical systems for new parameter values can be directly derived by utilizing the separate representations and implementing the inverse Fourier transform. The proposed approach is capable of being highly efficient because the online stage is independent of the spatial discretization. Finally, we present three specific instances of parametric dynamical systems to demonstrate the effectiveness of the proposed method.

AMS subject classifications: 35R60, 37M99, 65P99

Key words: Parametric dynamical system, model order reduction, domain decomposition, variable-separation method, Fourier transformation.

1. Introduction

Parametric dynamical systems have been widely adopted in science and engineering to model complex real-world problems characterized by various uncertainties. These uncertainties may stem from multiple sources, including physical properties, geometric configurations, initial conditions, and boundary conditions. The numerical simulation of such

*Corresponding author. *Email addresses:* yumingba@gpnu.edu.cn (Y. Ba), chl@hnu.edu.cn (L. Chen), yrchen@hnu.edu.cn (Y. Chen), qli28@hnu.edu.cn (Q. Li)

parametric systems poses substantial computational challenges, especially in the case of repeated parameter evaluation for design optimization, control analysis, and uncertainty quantification. Although traditional high-fidelity numerical approaches such as finite element method (FEM), finite difference method, and finite volume method can provide accurate solutions, they often entail extreme-scale computations with prohibitive computational costs.

To overcome arising computational bottlenecks in the simulation of parametric dynamical systems, a variety of innovative numerical approaches have been developed over past few decades. In particular, the model order reduction (MOR) technique has emerged as a powerful tool — cf. Refs. [1, 2, 16, 18, 33, 42, 46]. It is used to construct reduced-order models by approximating solutions in a low-dimensional subspace, which significantly reduces the computational cost. A prominent MOR approach is the reduced basis (RB) method — c.f. [12, 17, 19, 20, 24, 25], which follows an offline-online computational strategy. In the offline phase, high-fidelity solutions for a carefully selected parameter set are precomputed and stored. The online phase then efficiently approximates solutions for new parameters via a linear combination of these precomputed solutions. In this method, the key challenge lies in constructing an optimal set of basis functions that accurately capture the essential features of the full order model. Having the reduced basis functions obtained, one can use intrusive or non-intrusive approach to construct a reduced order model. Non-intrusive MOR shows the flexibility and efficiency because the full order numerical systems are not required. To enhance the applicability of MOR to nonlinear systems and describe relationship between the inputs and outputs, MOR methods based on various machine or deep learning techniques have been proposed [7, 27, 49, 50]. These methods roughly contain data-driven machine learning [32, 38, 53], physical-informed machine learning [7, 27, 49], and physics-data combined machine learning [15, 41]. Among these methods, the most effective and easiest approach is the data-driven machine learning, which usually converges rapidly because the loss functions are often simple. The physically-informed machine learning can demonstrate strong theoretical constraints and inductive biases by integrating governing physical rules and domain knowledge into the learning process. The physics-data combined machine learning has been used to study nonlinear dynamical systems in small-data regimes [15]. It has adopted a step-by-step training scheme, which seamlessly integrates the governing physical laws and the limited labeled data into feedforward neural networks.

An alternative approach for effective reducing the computational cost of dynamical systems is the frequency-domain method, which relies on two key transformations — viz. the Fourier transform [10, 11, 14, 28, 29] and the Laplace transform [30, 31] in the time dimension. The Fourier transform stands out as a significant discovery in mathematical sciences, playing a crucial role in modern scientific and technological advancements, which has been widely applied in the analysis of continuous-time systems, including signal and image processing, analog circuit analysis, and communication systems. The fundamental principle behind this approach is to convert time-dependent PDEs into frequency-dependent elliptic problems, which are stationary in time but parameterized by frequency. The time-domain solution is then reconstructed by applying an inverse Fourier transform to the frequency-

domain solutions. A major advantage of this framework is its inherent suitability for parallel computation — i.e. each frequency can be solved independently as the resulting frequency-domain equations are decoupled. For the frequency-domain equations that require numerous solutions, the MOR technique has demonstrated its effectiveness [1, 8, 23]. A low-order Galerkin proper orthogonal decomposition basis is employed to approximate solutions for frequency-domain equations in [45]. It adopts the Gaussian quadrature rule based on Legendre-Gauss-Lobatto (LGL) points to enable a precise numerical integration of the inverse Fourier transformation.

Motivated by these developments, we focus on the frequency-domain method based on the Fourier transform. Specifically, we reformulate time-dependent parametric PDEs as a family of frequency-dependent parametric elliptic equations. Although this transformation eliminates the time variable, the resulting equations remain dependent on spatial discretization, and must be solved repeatedly for multiple frequencies to recover the full solution via inverse transformation. However, for large number of frequencies the overall computational cost is prohibitively high. Hence, the reduction of the model order is essential for improving efficiency. As one of the MOR methods, the variable-separation (VS) method [34] has been verified to be able to provide accurate and efficient approximate solutions to stochastic PDEs. Then, Jiang and Li [26] has extended the VS to stochastic saddle point problems. For nonlinear PDEs with random input, the VS method also very efficient [35]. Moreover, it has been used in non-overlapping domain decomposition methods (DDM) for stochastic PDEs [5] and in parametric dynamical systems [6]. To improve the computational efficiency of the frequency-variable parametric elliptic equations, we consider a variable-separation-based domain decomposition (DD-VS) method [5] instead of VS method. It is well-suited to heterogeneous problems and enables efficient online computations.

The DD-VS method is developed based on the framework of non-overlapping DDM [3, 9, 21, 36, 41, 43, 47, 51, 52]. In this approach, the computational domain is split into non-overlapping subdomains, where the degree of freedom in each subdomain are classified into interior and interface parts. Applying block Gaussian elimination, one reduces global system to a Schur complement system with interface unknowns only [22, 37, 44]. This step constitutes the core and the most technically challenging part of the entire method. The DD-VS method constructs a mapping between the input parameters and the interface system of the parametric PDEs. It adopts an offline-online decomposition strategy to improve computational efficiency. In the offline phase, the extended VS method is used to transform the interface problem into a reduced-order algebraic system. A surrogate model is then built for this reduced interface system. In the online phase, the surrogate model is utilized to rapidly evaluate new parameter samples. The online computation is highly efficient, as its cost is entirely independent of the spatial discretization.

To apply the DD-VS method to frequency-dependent elliptic equations, it is first necessary to extend the VS method to handle complex-valued problems, since the solutions of such equations, as well as their associated interface and subdomain problems, are inherently complex-valued. To address this, we decompose the complex-valued problem into its real and imaginary parts, and formulate a coupled real-valued system accordingly.

By following the standard procedures of the VS method developed for real-valued elliptic equations, we construct separate affine surrogate models for the real and imaginary components. This extension allows the DD-VS method to be effectively applied to complex-valued frequency-dependent elliptic problems, while preserving computational efficiency in both the interface and subdomain problems during the online phase.

By combining the complex-valued DD-VS method with Fourier transformation, we develop a reduced order method for time-dependent problems, which can be regarded as an extension of the DD-VS method. The process begins with applying the Fourier transform to reformulate the time-dependent problems into a set of frequency-variable elliptic equations. Subsequently, utilizing the complex-valued DD-VS method, we construct efficient surrogate models for both the interface and subdomain problems during the offline phase. The resulting online stage is highly efficient — viz. for any new input parameter, the computation involves only evaluating the parametric coefficients of the surrogate model and performing the inverse Fourier transform. Importantly, the online phase is completely independent of the spatial discretization and requires no additional time-domain simulation, thereby enabling efficient solutions of time-dependent problems.

This paper is organized as follows. Section 2 contains necessary notation and preliminaries and introduces the Fourier transform for the time variable. Section 3 introduces a VS method for parametric complex-valued systems and the domain decomposition method for deterministic complex-valued PDEs. In Section 4, we present a new DD-VS method with key steps and procedures. Three numerical examples in Section 5 demonstrate the performance and computational advantages of the method proposed. Finally, we draw conclusions and offer some remarks on the method and its potential applications.

2. Preliminaries

Let Ω be the set representing the parameter domain. We use $\xi \in \Omega$ to represent a vector of parametric inputs. Given an open and bounded spatial domain $D \subset \mathbb{R}^d$ ($d = 1, 2, 3$) with Lipschitz continuous boundary ∂D , we consider the parametric dynamical system

$$\begin{aligned} \frac{\partial u}{\partial t}(\mathbf{x}, t; \xi) &= \mathcal{F}(u(\mathbf{x}, t; \xi); \xi), & \mathbf{x} \in D, & \quad t \in [0, T], \quad \xi \in \Omega, \\ u(\mathbf{x}, 0; \xi) &= u_0(\mathbf{x}; \xi), & \mathbf{x} \in D, & \quad \xi \in \Omega, \\ \mathcal{B}(u(\mathbf{x}, t; \xi)) &= g(\mathbf{x}, t; \xi), & \mathbf{x} \in \partial D, & \quad t \in [0, T], \quad \xi \in \Omega, \end{aligned}$$

where T is the end time of the computation, \mathcal{F} the spatial differential operator, \mathcal{B} the boundary condition operator, $u(\cdot, t; \xi)$ the system solution, $g(\cdot, t; \xi)$ the boundary term, and $u_0(\cdot; \xi)$ the initial function. To simplify the presentation, we illustrate our methodology using the following parametric parabolic partial differential equation as a representative example:

$$\frac{\partial u}{\partial t}(\mathbf{x}, t; \xi) - \nabla \cdot (c(\mathbf{x}; \xi) \nabla u(\mathbf{x}, t; \xi)) = f(\mathbf{x}, t; \xi), \quad \mathbf{x} \in D, \quad t \in [0, T], \quad \xi \in \Omega, \quad (2.1a)$$

$$u(\mathbf{x}, 0; \xi) = u_0(\mathbf{x}; \xi), \quad \mathbf{x} \in D, \quad \xi \in \Omega, \quad (2.1b)$$

$$u(\mathbf{x}, t; \xi) = g(\mathbf{x}, t; \xi), \quad \mathbf{x} \in \partial D, \quad t \in [0, T], \quad \xi \in \Omega, \quad (2.1c)$$

where $c(\mathbf{x}; \xi)$ is the diffusion coefficient and $f(\mathbf{x}, t; \xi)$ the source term.

Let

$$\mathcal{V} := L^2(D; \mathbb{C}) = \left\{ \hat{u}(\mathbf{x}) \mid \int_D |\hat{u}(\mathbf{x})|^2 dx < +\infty \right\}.$$

Besides, the inner product on \mathcal{V} and the associated norm are defined by

$$\langle \hat{u}(\mathbf{x}), \hat{v}(\mathbf{x}) \rangle := \int_D \hat{u}(\mathbf{x}) \overline{\hat{v}(\mathbf{x})} dx, \quad \|\hat{u}(\mathbf{x})\| := \sqrt{\langle \hat{u}(\mathbf{x}), \hat{u}(\mathbf{x}) \rangle}.$$

2.1. The frequency-domain method

To tackle the dynamical system (2.1), we apply the Fourier transformation to get the following set of complex-valued elliptic equations depending on the frequency ω :

$$\begin{aligned} i\omega \hat{u}(\mathbf{x}, \omega; \xi) - \nabla \cdot (c(\mathbf{x}; \xi) \nabla \hat{u}(\mathbf{x}, \omega; \xi)) &= \hat{f}(\mathbf{x}, \omega; \xi), & \mathbf{x} \in D, & \quad \xi \in \Omega, \\ \hat{u}(\mathbf{x}, \omega; \xi) &= \hat{g}(\mathbf{x}, \omega; \xi), & \mathbf{x} \in \partial D, & \quad \xi \in \Omega, \end{aligned} \quad (2.2)$$

where $i = \sqrt{-1}$ is the imaginary unit. The Fourier transform $\hat{u}(\mathbf{x}, \omega; \xi)$ of a function $u(\mathbf{x}, t; \xi)$ in time and its inverse are

$$\begin{aligned} \hat{u}(\mathbf{x}, \omega; \xi) &:= \int_{-\infty}^{\infty} u(\mathbf{x}, t; \xi) \exp(-i\omega t) dt, \\ u(\mathbf{x}, t; \xi) &:= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{u}(\mathbf{x}, \omega; \xi) \exp(i\omega t) d\omega. \end{aligned}$$

For the Fourier transformation, the function u is extended by zero outside of the interval $[0, T]$ with the functions f and g being treated identically to derive Eq. (2.2).

The weak formulation of (2.2) is: For all $\omega \in \mathbb{R}$ and $\xi \in \Omega$, find $\hat{u}(\cdot, \omega; \xi)$ such that $\hat{u} = \hat{g}$ on ∂D and

$$i\omega \langle \hat{u}(\cdot, \omega; \xi), \hat{v} \rangle + \langle c(\cdot; \xi) \nabla \hat{u}(\cdot, \omega; \xi), \nabla \hat{v} \rangle = \langle \hat{f}(\cdot, \omega; \xi), \hat{v} \rangle \quad \text{for all } \hat{v} \in \mathcal{V}. \quad (2.3)$$

Consider the finite element approximation of problem (2.3) in an n -dimensional subspace $\mathcal{V}_h \subset \mathcal{V}$. If $\{\psi_j\}_{j=1}^n$ is a set of basis functions of the space \mathcal{V}_h , then the solution $\hat{u}(\mathbf{x}, \omega; \xi)$ can be approximated as

$$\hat{u}(\mathbf{x}, \omega; \xi) \approx \hat{u}_h(\mathbf{x}, \omega; \xi) := \sum_{j=1}^n \hat{u}_j(\omega; \xi) \psi_j(\mathbf{x}),$$

where

$$\hat{u}_j(\omega; \xi) := \hat{u}_j^{\text{Re}}(\omega; \xi) + i\hat{u}_j^{\text{Im}}(\omega; \xi), \quad j = 1, \dots, n$$

is a complex-valued function. For the approximation of the Fourier inverse, we employ the Gaussian quadrature based on the LGL-points with an appropriate interval $[0, \omega^*]$ for a sufficiently large ω^* , where $\hat{u}(\cdot, \omega; \cdot)$ is negligible for $|\omega| > \omega^*$. Let $\{\omega_j\}_{j=1}^{N_\omega}$ be the set of LGL-points on the interval $[0, \omega^*]$. Then, the time-variable approximate $u(\mathbf{x}, t; \xi)$ to problem (2.1) is

$$u(\mathbf{x}, t; \xi) = \frac{1}{\pi} \operatorname{Re} \left(\sum_{j=1}^{N_\omega} \hat{u}_h(\mathbf{x}, \omega_j; \xi) \exp(i\omega_j t) w_j \right), \quad (2.4)$$

where w_j is the Gaussian quadrature weight of the LGL-points ω_j .

Note that, for different values of ω and ξ , one has to solve (2.3) repetitively to obtain the solution $\hat{u}_h(\mathbf{x}, \omega; \xi)$. This process requires a large amount of computation when dealing with a large number of parameters ω and ξ , especially for more complex problems. To reduce the computational cost, the frequency variable ω will be treated as a one-dimensional parameter in this paper. We aim to construct the reduced order model of $\hat{u}_h(\mathbf{x}, \omega; \xi)$. With this reduced order model, (2.4) can be computed in a computationally economical fashion, and the implementation details will be elaborated in the following sections.

3. VS and Domain Decomposition Methods

In this section, we will present in detail the VS method and the domain decomposition method for the complex-valued problem (2.3), where the VS method will be utilized in the next section to construct surrogate models for domain decomposed problems.

3.1. VS method for complex-valued problems

As already mentioned, our goal is the construction of a reduced order model when ω is considered as one of the parameters. In this section, the VS method will be extended to derive a separate representation of the solutions $\hat{u}(\mathbf{x}, \omega; \xi)$ for the complex-valued elliptic problem (2.3). For convenience, define $\boldsymbol{\mu} := (\omega, \xi)$, $\tilde{\Omega} := \mathbb{R} \times \Omega$. Then the equality (2.3) can be rewritten as

$$a(\hat{u}, \hat{v}; \boldsymbol{\mu}) = b(\hat{v}; \boldsymbol{\mu}), \quad (3.1)$$

where $a(\cdot, \cdot; \boldsymbol{\mu})$ and $b(\cdot; \boldsymbol{\mu})$ are complex-valued bilinear and linear forms on \mathcal{V} , respectively. Assume that $a(\cdot, \cdot; \boldsymbol{\mu})$ and $b(\cdot; \boldsymbol{\mu})$ are affine with respect to $\boldsymbol{\mu}$, i.e.

$$\begin{aligned} a(\hat{u}, \hat{v}; \boldsymbol{\mu}) &= \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu}) a_j(\hat{u}, \hat{v}) + i\gamma(\boldsymbol{\mu}) m(\hat{u}, \hat{v}) & \text{for all } \hat{u}, \hat{v} \in \mathcal{V}, \quad \boldsymbol{\mu} \in \tilde{\Omega}, \\ b(\hat{v}; \boldsymbol{\mu}) &= \sum_{j=1}^{m_b} \left(\beta_j^{\operatorname{Re}}(\boldsymbol{\mu}) b_j^{\operatorname{Re}}(\hat{v}) + i\beta_j^{\operatorname{Im}}(\boldsymbol{\mu}) b_j^{\operatorname{Im}}(\hat{v}) \right) & \text{for all } \hat{v} \in \mathcal{V}, \quad \boldsymbol{\mu} \in \tilde{\Omega}, \end{aligned} \quad (3.2)$$

where $\gamma(\boldsymbol{\mu}) = \omega$, $m(\hat{u}, \hat{v}) = \langle \hat{u}, \hat{v} \rangle$, $\alpha_j(\boldsymbol{\mu}) : \tilde{\Omega} \rightarrow R$ is $\boldsymbol{\mu}$ -dependent function and $a_j : \mathcal{V} \times \mathcal{V} \rightarrow R$ is a bilinear form independent of $\boldsymbol{\mu}$, for each $j = 1, \dots, m_a$, $\beta_j^{\operatorname{Re}}(\boldsymbol{\mu}) : \tilde{\Omega} \rightarrow R$ and

$\beta_j^{\text{Im}}(\boldsymbol{\mu}) : \tilde{\Omega} \rightarrow R$ are $\boldsymbol{\mu}$ -dependent functions, $b_j^{\text{Re}} : \mathcal{V} \rightarrow R$ and $b_j^{\text{Im}} : \mathcal{V} \rightarrow R$, $j = 1, \dots, m_b$ are linear forms independent of $\boldsymbol{\mu}$. Our analysis shows that affine decompositions play a crucial role in enabling the offline-online decomposition.

When (3.2) holds, the matrix form of Eq. (2.3) in a finite element space \mathcal{V}_h is given by

$$\left(\sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu}) A_j + i\gamma(\boldsymbol{\mu}) M \right) \hat{\mathbf{u}}(\boldsymbol{\mu}) = \sum_{j=1}^{m_b} \left(\beta_j^{\text{Re}}(\boldsymbol{\mu}) F_j^{\text{Re}} + i\beta_j^{\text{Im}}(\boldsymbol{\mu}) F_j^{\text{Im}} \right), \quad (3.3)$$

where $\hat{\mathbf{u}}(\boldsymbol{\mu}) = \hat{\mathbf{u}}^{\text{Re}}(\boldsymbol{\mu}) + i\hat{\mathbf{u}}^{\text{Im}}(\boldsymbol{\mu})$, and

$$\begin{aligned} (A_j)_{kl} &= a_j(\psi_k, \psi_l), & M_{kl} &= \langle \psi_k, \psi_l \rangle, \\ (\hat{\mathbf{u}}^{\text{Re}}(\boldsymbol{\mu}))_k &= \hat{u}_k^{\text{Re}}(\boldsymbol{\mu}), & (\hat{\mathbf{u}}^{\text{Im}}(\boldsymbol{\mu}))_k &= \hat{u}_k^{\text{Im}}(\boldsymbol{\mu}), \\ (F_j^{\text{Re}})_k &= b_j^{\text{Re}}(\psi_k), & (F_j^{\text{Im}})_k &= b_j^{\text{Im}}(\psi_k), \quad 1 \leq k, l \leq n. \end{aligned}$$

Now, we introduce the VS method for the problem (3.3) to obtain a separate approximation of the solution — i.e.

$$\hat{\mathbf{u}}(\boldsymbol{\mu}) \approx \hat{\mathbf{u}}_N(\boldsymbol{\mu}) := \sum_{j=1}^N \left(\zeta_j^{\text{Re}}(\boldsymbol{\mu}) \mathbf{c}_j^{\text{Re}} + i\zeta_j^{\text{Im}}(\boldsymbol{\mu}) \mathbf{c}_j^{\text{Im}} \right), \quad (3.4)$$

where $\zeta_j^{\text{Re}}(\boldsymbol{\mu})$ and $\zeta_j^{\text{Im}}(\boldsymbol{\mu})$ are real-valued parametric functions, and \mathbf{c}_j^{Re} and \mathbf{c}_j^{Im} are real-valued vectors independent of $\boldsymbol{\mu}$, with $N (\ll n)$ denoting the number of the separate terms. We define the error of the VS method by

$$\mathbf{e}(\boldsymbol{\mu}) := \mathbf{e}^{\text{Re}}(\boldsymbol{\mu}) + i\mathbf{e}^{\text{Im}}(\boldsymbol{\mu}) = \hat{\mathbf{u}}(\boldsymbol{\mu}) - \hat{\mathbf{u}}_{k-1}(\boldsymbol{\mu}), \quad (3.5)$$

where $\hat{\mathbf{u}}_{k-1}(\boldsymbol{\mu}) \equiv 0$ at $k = 1$. Substituting (3.5) into the Eq. (3.3) yields

$$-\gamma(\boldsymbol{\mu}) M \mathbf{e}^{\text{Im}}(\boldsymbol{\mu}) + \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu}) A_j \mathbf{e}^{\text{Re}}(\boldsymbol{\mu}) = \mathbf{r}_k^{\text{Re}}(\boldsymbol{\mu}), \quad (3.6)$$

$$\gamma(\boldsymbol{\mu}) M \mathbf{e}^{\text{Re}}(\boldsymbol{\mu}) + \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu}) A_j \mathbf{e}^{\text{Im}}(\boldsymbol{\mu}) = \mathbf{r}_k^{\text{Im}}(\boldsymbol{\mu}), \quad (3.7)$$

where

$$\mathbf{r}_k^{\text{Re}}(\boldsymbol{\mu}) := \begin{cases} \sum_{j=1}^{m_b} \beta_j^{\text{Re}}(\boldsymbol{\mu}) F_j^{\text{Re}}, & k = 1, \\ \sum_{j=1}^{m_b} \beta_j^{\text{Re}}(\boldsymbol{\mu}) F_j^{\text{Re}} + \gamma(\boldsymbol{\mu}) M \hat{\mathbf{u}}_{k-1}^{\text{Im}}(\boldsymbol{\mu}) - \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu}) A_j \hat{\mathbf{u}}_{k-1}^{\text{Re}}(\boldsymbol{\mu}), & k \geq 2, \end{cases} \quad (3.8)$$

$$\mathbf{r}_k^{\text{Im}}(\boldsymbol{\mu}) := \begin{cases} \sum_{j=1}^{m_b} \beta_j^{\text{Im}}(\boldsymbol{\mu}) F_j^{\text{Im}}, & k = 1, \\ \sum_{j=1}^{m_b} \beta_j^{\text{Im}}(\boldsymbol{\mu}) F_j^{\text{Im}} - \gamma(\boldsymbol{\mu}) M \hat{\mathbf{u}}_{k-1}^{\text{Re}}(\boldsymbol{\mu}) - \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu}) A_j \hat{\mathbf{u}}_{k-1}^{\text{Im}}(\boldsymbol{\mu}), & k \geq 2. \end{cases} \quad (3.9)$$

At the k -th step we choose $\boldsymbol{\mu}_k$ as follows:

$$\boldsymbol{\mu}_k = \begin{cases} \text{chosen randomly in } \Xi, & k = 1, \\ \operatorname{argmax}_{\boldsymbol{\mu} \in \Xi} |\mathbf{r}_k(\boldsymbol{\mu})|, & k \geq 2, \end{cases}$$

where Ξ is a collection of a finite number of samples in $\tilde{\Omega}$ and $\mathbf{r}_k(\boldsymbol{\mu}) = \mathbf{r}_k^{\operatorname{Re}}(\boldsymbol{\mu}) + i\mathbf{r}_k^{\operatorname{Im}}(\boldsymbol{\mu})$. Solving the Eqs. (3.6)-(3.7) with $\boldsymbol{\mu} = \boldsymbol{\mu}_k$, we derive $\mathbf{c}_k^{\operatorname{Re}} = \mathbf{e}^{\operatorname{Re}}(\boldsymbol{\mu}_k)$ and $\mathbf{c}_k^{\operatorname{Im}} = \mathbf{e}^{\operatorname{Im}}(\boldsymbol{\mu}_k)$ in (3.4).

Let $\tilde{\mathbf{e}}^{\operatorname{Re}}(\boldsymbol{\mu}) = \zeta_k^{\operatorname{Re}}(\boldsymbol{\mu})\mathbf{c}_k^{\operatorname{Re}}$ and $\tilde{\mathbf{e}}^{\operatorname{Im}}(\boldsymbol{\mu}) = \zeta_k^{\operatorname{Im}}(\boldsymbol{\mu})\mathbf{c}_k^{\operatorname{Im}}$. By taking $\mathbf{e}^{\operatorname{Re}} = \tilde{\mathbf{e}}^{\operatorname{Re}}$ and $\mathbf{e}^{\operatorname{Im}} = \tilde{\mathbf{e}}^{\operatorname{Im}}$ in (3.6)-(3.7), the linear systems involving the unknowns $\zeta_k^{\operatorname{Re}}(\boldsymbol{\mu})$ and $\zeta_k^{\operatorname{Im}}(\boldsymbol{\mu})$ are as follows:

$$-\gamma(\boldsymbol{\mu})M\mathbf{c}_k^{\operatorname{Im}}\zeta_k^{\operatorname{Im}}(\boldsymbol{\mu}) + \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})A_j\mathbf{c}_k^{\operatorname{Re}}\zeta_k^{\operatorname{Re}}(\boldsymbol{\mu}) = \mathbf{r}_k^{\operatorname{Re}}(\boldsymbol{\mu}), \quad (3.10)$$

$$\gamma(\boldsymbol{\mu})M\mathbf{c}_k^{\operatorname{Re}}\zeta_k^{\operatorname{Re}}(\boldsymbol{\mu}) + \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})A_j\mathbf{c}_k^{\operatorname{Im}}\zeta_k^{\operatorname{Im}}(\boldsymbol{\mu}) = \mathbf{r}_k^{\operatorname{Im}}(\boldsymbol{\mu}). \quad (3.11)$$

Taking the dot product of (3.10) with $\mathbf{c}_k^{\operatorname{Re}}$ and (3.11) with $\mathbf{c}_k^{\operatorname{Im}}$ yields the affine representations of $\zeta_k^{\operatorname{Re}}(\boldsymbol{\mu})$ and $\zeta_k^{\operatorname{Im}}(\boldsymbol{\mu})$ as

$$\begin{aligned} \zeta_k^{\operatorname{Re}}(\boldsymbol{\mu}) &= \left(\sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Im}})^T A_j \mathbf{c}_k^{\operatorname{Im}} \cdot (\mathbf{c}_k^{\operatorname{Re}})^T \mathbf{r}_k^{\operatorname{Re}}(\boldsymbol{\mu}) + \gamma(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Re}})^T M \mathbf{c}_k^{\operatorname{Im}} \cdot (\mathbf{c}_k^{\operatorname{Im}})^T \mathbf{r}_k^{\operatorname{Im}}(\boldsymbol{\mu}) \right) \\ &\quad \times \left(\sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Re}})^T A_j \mathbf{c}_k^{\operatorname{Re}} \cdot \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Im}})^T A_j \mathbf{c}_k^{\operatorname{Im}} \right. \\ &\quad \left. + \gamma(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Re}})^T M \mathbf{c}_k^{\operatorname{Im}} \cdot \gamma(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Im}})^T M \mathbf{c}_k^{\operatorname{Re}} \right)^{-1}, \end{aligned} \quad (3.12)$$

$$\begin{aligned} \zeta_k^{\operatorname{Im}}(\boldsymbol{\mu}) &= \left(\sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Re}})^T A_j \mathbf{c}_k^{\operatorname{Re}} \cdot (\mathbf{c}_k^{\operatorname{Im}})^T \mathbf{r}_k^{\operatorname{Im}}(\boldsymbol{\mu}) - \gamma(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Im}})^T M \mathbf{c}_k^{\operatorname{Re}} \cdot (\mathbf{c}_k^{\operatorname{Re}})^T \mathbf{r}_k^{\operatorname{Re}}(\boldsymbol{\mu}) \right) \\ &\quad \times \left(\gamma(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Re}})^T M \mathbf{c}_k^{\operatorname{Im}} \cdot \gamma(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Im}})^T M \mathbf{c}_k^{\operatorname{Re}} \right. \\ &\quad \left. + \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Im}})^T A_j \mathbf{c}_k^{\operatorname{Im}} \cdot \sum_{j=1}^{m_a} \alpha_j(\boldsymbol{\mu})(\mathbf{c}_k^{\operatorname{Re}})^T A_j \mathbf{c}_k^{\operatorname{Re}} \right)^{-1}, \end{aligned} \quad (3.13)$$

where both $\mathbf{r}_k^{\operatorname{Re}}(\boldsymbol{\mu})$ and $\mathbf{r}_k^{\operatorname{Im}}(\boldsymbol{\mu})$ are affine functions of $\boldsymbol{\mu}$, as defined by the Eqs. (3.8)-(3.9).

The iteration procedure ends when $|\mathbf{r}_k(\boldsymbol{\mu}_k)|$ is small enough. The above procedure of the VS method for the complex-valued elliptic problem is summarized in Algorithm 3.1. We emphasize that the coefficients $\{\mathbf{c}_j^{\operatorname{Re}}\}_{j=1}^N$ and $\{\mathbf{c}_j^{\operatorname{Im}}\}_{j=1}^N$ (independent of $\boldsymbol{\mu}$) can be calculated

Algorithm 3.1 VS Method for the Complex-Valued Elliptic Problem

Input: The complex-valued elliptic problem (3.3), a set of samples $\Xi \in \tilde{\Omega}$, and the error tolerance ε .

Output: The approximation $\hat{\mathbf{u}}_N(\boldsymbol{\mu}) := \sum_{j=1}^N (\zeta_j^{\text{Re}}(\boldsymbol{\mu})\mathbf{c}_j^{\text{Re}} + i\zeta_j^{\text{Im}}(\boldsymbol{\mu})\mathbf{c}_j^{\text{Im}})$.

Step 1. Initialize the iteration counter $k = 1$, a random $\boldsymbol{\mu}_1 \in \Xi$.

Step 2. Calculate \mathbf{c}_k^{Re} and \mathbf{c}_k^{Im} by solving (3.6) and (3.7) with $\boldsymbol{\mu} = \boldsymbol{\mu}_k$; compute $\zeta_k^{\text{Re}}(\boldsymbol{\mu})$ and $\zeta_k^{\text{Im}}(\boldsymbol{\mu})$ by (3.12) and (3.13).

Step 3. Update $\Xi = \Xi \setminus \boldsymbol{\mu}_k$, and take $\hat{\mathbf{u}}_k(\boldsymbol{\mu}) := \sum_{j=1}^k (\zeta_j^{\text{Re}}(\boldsymbol{\mu})\mathbf{c}_j^{\text{Re}} + i\zeta_j^{\text{Im}}(\boldsymbol{\mu})\mathbf{c}_j^{\text{Im}})$.

Step 4. Set $k \rightarrow k + 1$.

if $|\mathbf{r}(\boldsymbol{\mu})| \geq \varepsilon$ and $\Xi \neq \emptyset$ **then**

 Choose $\boldsymbol{\mu}_k \in \arg\max_{\boldsymbol{\mu} \in \Xi} |\mathbf{r}_k(\boldsymbol{\mu})|$, and go to Step 2;

else

 Set $N = k$ and terminate.

end if

during the offline stage of the VS method and used directly in the online stage. Then we only need to calculate the parametric functions in Eq. (3.4) for any $\boldsymbol{\mu} \in \tilde{\Omega}$ during the online stage. This procedure is highly efficient, since it only involves the separate representation (3.4).

Although the VS method allows constructing reduced-order models for frequency-domain problems, its application to time-domain dynamical problems in complex geometries remains challenging. In particular, employing the Fourier transform in such scenarios often necessitates a large N to achieve sufficient approximation accuracy, leading to high computational costs. To address this limitation and enhance computational efficiency, we propose incorporating a domain decomposition strategy.

3.2. Domain decomposition method for complex-valued problems

In this section, we introduce the non-overlapping domain decomposition method for complex-valued problems. For real-valued problems, the reader can consult Refs. [39, 48]. Here the complex-valued problem (2.2) is considered for a fixed $\bar{\boldsymbol{\mu}} \in \tilde{\Omega}$.

Suppose that the domain D is split into N_s non-overlapping subdomains $\{D_j\}_{j=1}^{N_s}$, i.e. $D = \cup_{j=1}^{N_s} \overline{D_j}$ and $D_j \cap D_k = \emptyset$ if $j \neq k$. Restricting the weak formulation (3.1) to subdomain D_j , we write

$$a(\hat{\mathbf{u}}, \hat{\mathbf{v}}; \bar{\boldsymbol{\mu}})_j = b(\hat{\mathbf{v}}; \bar{\boldsymbol{\mu}})_j \quad \text{for all } \hat{\mathbf{v}} \in \mathcal{V}_j,$$

where \mathcal{V}_j is the Hilbert space \mathcal{V} restricted to the subdomain D_j . The finite element discretization of the above equation leads the local linear system

$$A^j \hat{\mathbf{u}}^j = \mathbf{f}^j, \tag{3.14}$$

where A^j , \mathbf{f}^j , and $\hat{\mathbf{u}}^j$ are respectively the local system matrix, local load vector, and local unknown solution. All of those are complex-valued — i.e.

$$A^j := A^{j,\text{Re}} + iA^{j,\text{Im}}, \quad \hat{\mathbf{u}}^j := \hat{\mathbf{u}}^{j,\text{Re}} + i\hat{\mathbf{u}}^{j,\text{Im}}, \quad \mathbf{f}^j := \mathbf{f}^{j,\text{Re}} + i\mathbf{f}^{j,\text{Im}}.$$

Similar to the real-valued case, the local system (3.14) is singular due to the lack of boundary conditions. To address this, we decompose $\hat{\mathbf{u}}^j$ into two parts — viz. the interface part $\hat{\mathbf{u}}_\Gamma^j$, where nodes are shared by two or more adjacent subdomains, and the interior part $\hat{\mathbf{u}}_I^j$ belonging to the subdomain D_j . As a result, the Eq. (3.14) can be written as

$$\begin{bmatrix} A_{II}^{j,\text{Re}} + iA_{II}^{j,\text{Im}} & A_{I\Gamma}^{j,\text{Re}} + iA_{I\Gamma}^{j,\text{Im}} \\ A_{\Gamma I}^{j,\text{Re}} + iA_{\Gamma I}^{j,\text{Im}} & A_{\Gamma\Gamma}^{j,\text{Re}} + iA_{\Gamma\Gamma}^{j,\text{Im}} \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{u}}_I^{j,\text{Re}} + i\hat{\mathbf{u}}_I^{j,\text{Im}} \\ \hat{\mathbf{u}}_\Gamma^{j,\text{Re}} + i\hat{\mathbf{u}}_\Gamma^{j,\text{Im}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_I^{j,\text{Re}} + i\mathbf{f}_I^{j,\text{Im}} \\ \mathbf{f}_\Gamma^{j,\text{Re}} + i\mathbf{f}_\Gamma^{j,\text{Im}} \end{Bmatrix}.$$

The system shows that once the interface unknowns $\hat{\mathbf{u}}_\Gamma^j$ are determined, the interior unknowns $\hat{\mathbf{u}}_I^j$ can be determined by solving the interior problem

$$\left[A_{II}^{j,\text{Re}} + iA_{II}^{j,\text{Im}} \right] \left\{ \hat{\mathbf{u}}_I^{j,\text{Re}} + i\hat{\mathbf{u}}_I^{j,\text{Im}} \right\} = \left\{ \mathbf{f}_I^{j,\text{Re}} + i\mathbf{f}_I^{j,\text{Im}} \right\} - \left[A_{I\Gamma}^{j,\text{Re}} + iA_{I\Gamma}^{j,\text{Im}} \right] \left\{ \hat{\mathbf{u}}_\Gamma^{j,\text{Re}} + i\hat{\mathbf{u}}_\Gamma^{j,\text{Im}} \right\}.$$

Let S_j and F_j be respectively the local Schur complement matrix and the corresponding right-hand side vector — i.e.

$$\begin{aligned} S_j &= \left[A_{\Gamma\Gamma}^{j,\text{Re}} + iA_{\Gamma\Gamma}^{j,\text{Im}} \right] - \left[A_{\Gamma I}^{j,\text{Re}} + iA_{\Gamma I}^{j,\text{Im}} \right] \left[A_{II}^{j,\text{Re}} + iA_{II}^{j,\text{Im}} \right]^{-1} \left[A_{I\Gamma}^{j,\text{Re}} + iA_{I\Gamma}^{j,\text{Im}} \right], \\ F_j &= \left\{ \mathbf{f}_\Gamma^{j,\text{Re}} + i\mathbf{f}_\Gamma^{j,\text{Im}} \right\} - \left[A_{\Gamma I}^{j,\text{Re}} + iA_{\Gamma I}^{j,\text{Im}} \right] \left[A_{II}^{j,\text{Re}} + iA_{II}^{j,\text{Im}} \right]^{-1} \left\{ \mathbf{f}_I^{j,\text{Re}} + i\mathbf{f}_I^{j,\text{Im}} \right\}. \end{aligned}$$

Then the global interface unknowns $\hat{\mathbf{u}}_\Gamma$ can be obtained by solving the global interface problem

$$S\hat{\mathbf{u}}_\Gamma = F, \tag{3.15}$$

where

$$S = \sum_{j=1}^{N_s} R_j^T S_j R_j, \quad F = \sum_{j=1}^{N_s} R_j^T F_j.$$

The restriction matrix R_j acts as a scatter operator that maps the global interface unknowns $\hat{\mathbf{u}}_\Gamma$ to the local interface unknowns $\hat{\mathbf{u}}_\Gamma^j$, satisfying the relation $R_j \hat{\mathbf{u}}_\Gamma = \hat{\mathbf{u}}_\Gamma^j$. This operator enables the consistent transfer of interface information from the global domain to each local subdomain within the domain decomposition framework.

Note that the primary computational cost of the construction of S and F arises from inverting the complex-valued matrices $A_{II}^{j,\text{Re}} + iA_{II}^{j,\text{Im}}$, $j = 1, \dots, N_s$. In this work, we investigate the application of the domain decomposition method to parametric dynamical systems. Using the Fourier transform, these systems can be converted into complex-valued parametric elliptic problems. For each random input $\boldsymbol{\mu} \in \tilde{\Omega}$, all the matrices mentioned are $\boldsymbol{\mu}$ -dependent, which leads to significant computational challenges when solving the interface system.

4. Domain Decomposition Based on Variable-Separation

In this section, we will first present an offline-online method for the interface problem of the complex-valued elliptic problem. This builds a relation between random input and

global interface solution — cf. Sections 4.1-4.2. After that we present separate representations of the subproblems solutions. According to the Eq. (3.15), the interface problem for the complex-valued elliptic equation can be written as

$$S(\boldsymbol{\mu})\hat{\mathbf{u}}_\Gamma(\boldsymbol{\mu}) = F(\boldsymbol{\mu}),$$

where

$$S(\boldsymbol{\mu}) = \sum_{j=1}^{N_s} R_j^T S_j(\boldsymbol{\mu}) R_j, \quad F(\boldsymbol{\mu}) = \sum_{j=1}^{N_s} R_j^T F_j(\boldsymbol{\mu}),$$

and

$$S_j(\boldsymbol{\mu}) = \left[A_{\Gamma\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma\Gamma}^{j,\text{Im}}(\boldsymbol{\mu}) \right] - \left[A_{\Gamma I}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma I}^{j,\text{Im}}(\boldsymbol{\mu}) \right] \left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu}) \right]^{-1} \left[A_{I\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{I\Gamma}^{j,\text{Im}}(\boldsymbol{\mu}) \right], \quad (4.1)$$

$$F_j(\boldsymbol{\mu}) = \left\{ \mathbf{f}_\Gamma^{j,\text{Re}}(\boldsymbol{\mu}) + i\mathbf{f}_\Gamma^{j,\text{Im}}(\boldsymbol{\mu}) \right\} - \left[A_{\Gamma I}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma I}^{j,\text{Im}}(\boldsymbol{\mu}) \right] \left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu}) \right]^{-1} \left\{ \mathbf{f}_I^{j,\text{Re}}(\boldsymbol{\mu}) + i\mathbf{f}_I^{j,\text{Im}}(\boldsymbol{\mu}) \right\}. \quad (4.2)$$

To simplify the presentation of the method, we split the domain D into two subdomains D_1 and D_2 and denote their interface by Γ instead of Γ_{12} . Let us point out that two-subdomain analysis extends naturally to multiple number of subdomains. The corresponding interface problem for the complex-valued elliptic equation takes the following form:

$$(S_1(\boldsymbol{\mu}) + S_2(\boldsymbol{\mu}))\hat{\mathbf{u}}_\Gamma(\boldsymbol{\mu}) = F_1(\boldsymbol{\mu}) + F_2(\boldsymbol{\mu}). \quad (4.3)$$

4.1. Affine representations of S and F

To enhance the computational efficiency, we develop an affine reformulation strategy for (4.3) that guarantees the affine parameter dependence of both $S(\boldsymbol{\mu})$ and $F(\boldsymbol{\mu})$. The implementation details of this approach are presented in this section. Using the affine decomposition (3.2), we write the parametric matrices in Eqs. (4.1)-(4.2) as

$$\begin{aligned} A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu}) A_{II}^{jn}, & A_{II}^{j,\text{Im}}(\boldsymbol{\mu}) &= \gamma^j(\boldsymbol{\mu}) M_{II}^j, \\ A_{I\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu}) A_{I\Gamma}^{jn}, & A_{I\Gamma}^{j,\text{Im}}(\boldsymbol{\mu}) &= \gamma^j(\boldsymbol{\mu}) M_{I\Gamma}^j, \\ A_{\Gamma I}^{j,\text{Re}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu}) A_{\Gamma I}^{jn}, & A_{\Gamma I}^{j,\text{Im}}(\boldsymbol{\mu}) &= \gamma^j(\boldsymbol{\mu}) M_{\Gamma I}^j, \\ A_{\Gamma\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu}) A_{\Gamma\Gamma}^{jn}, & A_{\Gamma\Gamma}^{j,\text{Im}}(\boldsymbol{\mu}) &= \gamma^j(\boldsymbol{\mu}) M_{\Gamma\Gamma}^j, \end{aligned} \quad (4.4)$$

where the matrices $A_{II}^{jn}, A_{I\Gamma}^{jn}, A_{\Gamma I}^{jn}, A_{\Gamma\Gamma}^{jn}$ and $M_{II}^j, M_{I\Gamma}^j, M_{\Gamma I}^j, M_{\Gamma\Gamma}^j$ do not depend on $\boldsymbol{\mu}$. Note that the first term of $S_j(\boldsymbol{\mu})$ has the structure affine with respect to $\boldsymbol{\mu}$ as required. Next, we discuss how to obtain the affine approximation of the term

$$-\left[A_{\Gamma I}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma I}^{j,\text{Im}}(\boldsymbol{\mu})\right]\left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu})\right]^{-1}\left[A_{I\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{I\Gamma}^{j,\text{Im}}(\boldsymbol{\mu})\right]$$

in (4.1).

Step 1. Construct the low-rank approximation of

$$X^j(\boldsymbol{\mu}) = \left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu})\right]^{-1}\left[A_{I\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{I\Gamma}^{j,\text{Im}}(\boldsymbol{\mu})\right] \quad (4.5)$$

such as

$$X_N^j(\boldsymbol{\mu}) := \sum_{n=1}^{N_{S_j}} \left(\phi_n^{j,\text{Re}}(\boldsymbol{\mu})X_n^{j,\text{Re}} + i\phi_n^{j,\text{Im}}(\boldsymbol{\mu})X_n^{j,\text{Im}}\right). \quad (4.6)$$

Indeed, under the assumption (4.4), one can rewrite (4.5) as

$$\left(\sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu})A_{II}^{jn} + i\gamma^j(\boldsymbol{\mu})M_{II}^j\right)X^j(\boldsymbol{\mu}) = \left(\sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu})A_{I\Gamma}^{jn} + i\gamma^j(\boldsymbol{\mu})M_{I\Gamma}^j\right).$$

Let n_Γ be the number of the interface unknowns. The unknown parametric matrix $X^j(\boldsymbol{\mu})$ can be expressed as $X^j(\boldsymbol{\mu}) = [x_1^j(\boldsymbol{\mu}), x_2^j(\boldsymbol{\mu}), \dots, x_{n_\Gamma}^j(\boldsymbol{\mu})]$, and each element $x_k^j(\boldsymbol{\mu})$, $k = 1, \dots, n_\Gamma$ can be obtained by solving

$$\left(\sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu})A_{II}^{jn} + i\gamma^j(\boldsymbol{\mu})M_{II}^j\right)x_k^j(\boldsymbol{\mu}) = \delta_k^j(\boldsymbol{\mu}), \quad (4.7)$$

where $\delta_k^j(\boldsymbol{\mu})$ denotes the k -th column of the matrix $(\sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu})A_{I\Gamma}^{jn} + i\gamma^j(\boldsymbol{\mu})M_{I\Gamma}^j)$.

Applying Algorithm 3.1, we can derive the low-rank approximation for each component $x_k^j(\boldsymbol{\mu})$. Then, the approximate solution $X_N^j(\boldsymbol{\mu})$ in (4.6) is constructed by rearranging the low-rank approximations of $\{x_k^j(\boldsymbol{\mu})\}_{k=1}^{n_\Gamma}$.

Step 2. Assemble affine expression for

$$\mathcal{X}^j(\boldsymbol{\mu}) = -\left[A_{\Gamma I}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma I}^{j,\text{Im}}(\boldsymbol{\mu})\right]\left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu})\right]^{-1}\left[A_{I\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{I\Gamma}^{j,\text{Im}}(\boldsymbol{\mu})\right].$$

Using the low-rank representation of $X^j(\boldsymbol{\mu})$ and the Eq. (4.4), we write

$$\begin{aligned} \mathcal{X}^j(\boldsymbol{\mu}) &\approx -\left[\sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu})A_{\Gamma I}^{jn} + i\gamma^j(\boldsymbol{\mu})M_{\Gamma I}^j\right]\left[\sum_{n=1}^{N_{S_j}} \left(\phi_n^{j,\text{Re}}(\boldsymbol{\mu})X_n^{j,\text{Re}} + i\phi_n^{j,\text{Im}}(\boldsymbol{\mu})X_n^{j,\text{Im}}\right)\right] \\ &= \sum_{n=1}^{(m_{a_j}+1)N_{S_j}} \left(\eta_n^{j,\text{Re}}(\boldsymbol{\mu})\boldsymbol{x}_n^{j,\text{Re}} + i\eta_n^{j,\text{Im}}(\boldsymbol{\mu})\boldsymbol{x}_n^{j,\text{Im}}\right), \end{aligned} \quad (4.8)$$

where

$$\begin{aligned}\eta_n^{j,\text{Re}}(\boldsymbol{\mu}) &= \alpha^{jn_1}(\boldsymbol{\mu})\phi_{n_2}^{j,\text{Re}}(\boldsymbol{\mu}), & \mathcal{X}_n^{j,\text{Re}} &= -A_{\Gamma I}^{jn_1}X_{n_2}^{j,\text{Re}}, \\ \eta_n^{j,\text{Im}}(\boldsymbol{\mu}) &= \alpha^{jn_1}(\boldsymbol{\mu})\phi_{n_2}^{j,\text{Im}}(\boldsymbol{\mu}), & \mathcal{X}_n^{j,\text{Im}} &= -A_{\Gamma I}^{jn_1}X_{n_2}^{j,\text{Im}}, \quad n = 1, 2, \dots, m_{a_j}N_{S_j}, \\ \eta_n^{j,\text{Re}}(\boldsymbol{\mu}) &= \gamma^j(\boldsymbol{\mu})\phi_{n_2}^{j,\text{Im}}(\boldsymbol{\mu}), & \mathcal{X}_n^{j,\text{Re}} &= M_{\Gamma I}^jX_{n_2}^{j,\text{Im}}, \\ \eta_n^{j,\text{Im}}(\boldsymbol{\mu}) &= \gamma^j(\boldsymbol{\mu})\phi_{n_2}^{j,\text{Re}}(\boldsymbol{\mu}), & \mathcal{X}_n^{j,\text{Im}} &= -M_{\Gamma I}^jX_{n_2}^{j,\text{Re}}, \quad n = m_{a_j}N_{S_j} + 1, \dots, (m_{a_j} + 1)N_{S_j},\end{aligned}$$

where $n_1 = 1, 2, \dots, m_{a_j}$ and $n_2 = 1, 2, \dots, N_{S_j}$.

Subsequently, the low-rank representation of $S_j(\boldsymbol{\mu})$ takes the form

$$\begin{aligned}S_j(\boldsymbol{\mu}) &= \left[A_{\Gamma \Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma \Gamma}^{j,\text{Im}}(\boldsymbol{\mu}) \right] + \mathcal{X}^j(\boldsymbol{\mu}) \\ &\approx \sum_{n=1}^{m_{a_j}} \alpha^{jn}(\boldsymbol{\mu})A_{\Gamma \Gamma}^{jn} + i\gamma^j(\boldsymbol{\mu})M_{\Gamma \Gamma}^j + \sum_{n=1}^{(m_{a_j}+1)N_{S_j}} \left(\eta_n^{j,\text{Re}}(\boldsymbol{\mu})\mathcal{X}_n^{j,\text{Re}} + i\eta_n^{j,\text{Im}}(\boldsymbol{\mu})\mathcal{X}_n^{j,\text{Im}} \right),\end{aligned}$$

so that the low-rank representation of $S(\boldsymbol{\mu})$ is

$$S(\boldsymbol{\mu}) = S_1(\boldsymbol{\mu}) + S_2(\boldsymbol{\mu}) \approx \sum_{n=1}^{m_s} \left(\hat{\eta}_n^{\text{Re}}(\boldsymbol{\mu})\hat{\mathcal{X}}_n^{\text{Re}} + i\hat{\eta}_n^{\text{Im}}(\boldsymbol{\mu})\hat{\mathcal{X}}_n^{\text{Im}} \right), \quad (4.9)$$

where the sum term comes from stacking the variables and sorting the corresponding indices via a single one, and

$$m_s = m_{a_1} + m_{a_2} + (m_{a_1} + 1)N_{S_1} + (m_{a_2} + 1)N_{S_2}.$$

Note that m_s is the number of the real part of $S(\boldsymbol{\mu})$, and the number of its imaginary part is equal to $2 + (m_{a_1} + 1)N_{S_1} + (m_{a_2} + 1)N_{S_2}$, which is less than or equal to m_s . For convenience, we also use m_s to denote the number of the imaginary part, since for the imaginary part, the terms with the index $n = 3 + (m_{a_1} + 1)N_{S_1} + (m_{a_2} + 1)N_{S_2}, \dots, m_s$ can be set to 0. Algorithm 4.1 outlines the assembling process of $S(\boldsymbol{\mu})$.

Algorithm 4.1 The Assemble Process of $S(\boldsymbol{\mu})$

Input: The complex-valued matrices $A_{II}^j(\boldsymbol{\mu})$, $A_{I\Gamma}^j(\boldsymbol{\mu})$, $A_{\Gamma I}^j(\boldsymbol{\mu})$, $A_{\Gamma\Gamma}^j(\boldsymbol{\mu})$, $j = 1, 2$.

Output: The low-rank representation $S(\boldsymbol{\mu}) = \sum_{n=1}^{m_s} (\hat{\eta}_n^{\text{Re}}(\boldsymbol{\mu})\hat{\mathcal{X}}_n^{\text{Re}} + i\hat{\eta}_n^{\text{Im}}(\boldsymbol{\mu})\hat{\mathcal{X}}_n^{\text{Im}})$.

Step 1. Get the approximation of

$$X^j(\boldsymbol{\mu}) = \left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu}) \right]^{-1} \left[A_{I\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{I\Gamma}^{j,\text{Im}}(\boldsymbol{\mu}) \right]$$

by solving Eq. (4.7) with Algorithm 3.1.

Step 2. Assemble the affine expression of

$$-\left[A_{\Gamma I}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma I}^{j,\text{Im}}(\boldsymbol{\mu}) \right] \left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu}) \right]^{-1} \left[A_{I\Gamma}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{I\Gamma}^{j,\text{Im}}(\boldsymbol{\mu}) \right]$$

by (4.8).

- Step 3. Assemble $S_j(\boldsymbol{\mu})$ based on (4.4) and the expression derived in Step 2.
 Step 4. Assemble the low-rank representation of $S(\boldsymbol{\mu})$ by (4.9).
-

Similarly, with the assumption (3.2) of affine decomposition, we have

$$\begin{aligned} \mathbf{f}_I^{j,\text{Re}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{b_j}} \beta^{jn,\text{Re}}(\boldsymbol{\mu}) \mathbf{f}_I^{jn,\text{Re}}, & \mathbf{f}_I^{j,\text{Im}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{b_j}} \beta^{jn,\text{Im}}(\boldsymbol{\mu}) \mathbf{f}_I^{jn,\text{Im}}, \\ \mathbf{f}_\Gamma^{j,\text{Re}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{b_j}} \beta^{jn,\text{Re}}(\boldsymbol{\mu}) \mathbf{f}_\Gamma^{jn,\text{Re}}, & \mathbf{f}_\Gamma^{j,\text{Im}}(\boldsymbol{\mu}) &= \sum_{n=1}^{m_{b_j}} \beta^{jn,\text{Im}}(\boldsymbol{\mu}) \mathbf{f}_\Gamma^{jn,\text{Im}}, \end{aligned} \quad (4.10)$$

where $\mathbf{f}_I^{jn,\text{Re}}$, $\mathbf{f}_I^{jn,\text{Im}}$, $\mathbf{f}_\Gamma^{jn,\text{Re}}$ and $\mathbf{f}_\Gamma^{jn,\text{Im}}$ are independent of $\boldsymbol{\mu}$. Following the assemble strategy of $S(\boldsymbol{\mu})$, one can derive the low-rank representation for $F(\boldsymbol{\mu})$ as follows:

$$F(\boldsymbol{\mu}) = F_1(\boldsymbol{\mu}) + F_2(\boldsymbol{\mu}) \approx \sum_{n=1}^{m_F} (\hat{\rho}_n^{\text{Re}}(\boldsymbol{\mu}) \hat{F}_n^{\text{Re}} + i \hat{\rho}_n^{\text{Im}}(\boldsymbol{\mu}) \hat{F}_n^{\text{Im}}), \quad (4.11)$$

where

$$m_F = m_{b_1} + m_{b_2} + (m_{a_1} + 1)N_{F_1} + (m_{a_2} + 1)N_{F_2}.$$

The detailed procedure of assembling $F(\boldsymbol{\mu})$ is presented in Algorithm 4.2.

Algorithm 4.2 The Assemble Process of $F(\boldsymbol{\mu})$

Input: The complex-valued matrices $A_{II}^j(\boldsymbol{\mu})$, $A_{\Gamma I}^j(\boldsymbol{\mu})$ and vectors $\mathbf{f}_I^j(\boldsymbol{\mu})$, $\mathbf{f}_\Gamma^j(\boldsymbol{\mu})$, $j = 1, 2$.

Output: The low-rank representation $F(\boldsymbol{\mu}) = \sum_{n=1}^{m_F} (\hat{\rho}_n^{\text{Re}}(\boldsymbol{\mu}) \hat{F}_n^{\text{Re}} + i \hat{\rho}_n^{\text{Im}}(\boldsymbol{\mu}) \hat{F}_n^{\text{Im}})$.

- Step 1. Get the approximation of $[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu})]^{-1} \{\mathbf{f}_I^{j,\text{Re}}(\boldsymbol{\mu}) + i\mathbf{f}_I^{j,\text{Im}}(\boldsymbol{\mu})\}$ by Algorithm 3.1.
 Step 2. Assemble the affine expression of

$$-\left[A_{\Gamma I}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{\Gamma I}^{j,\text{Im}}(\boldsymbol{\mu})\right] \left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + iA_{II}^{j,\text{Im}}(\boldsymbol{\mu})\right]^{-1} \left\{\mathbf{f}_I^{j,\text{Re}}(\boldsymbol{\mu}) + i\mathbf{f}_I^{j,\text{Im}}(\boldsymbol{\mu})\right\}.$$

- Step 3. Assemble $F_j(\boldsymbol{\mu})$ based on (4.10) and the expression derived in Step 2.
 Step 4. Assemble the low-rank representation of $F(\boldsymbol{\mu})$ by (4.11).
-

4.2. Reduced model for the interface problem

Based on the separate approximations of $S(\boldsymbol{\mu})$ and $F(\boldsymbol{\mu})$ derived in Section 4.1, the interface problem (4.3) can be expressed as

$$\sum_{n=1}^{m_S} (\hat{\gamma}_n^{\text{Re}}(\boldsymbol{\mu}) \hat{\mathcal{X}}_n^{\text{Re}} + i \hat{\gamma}_n^{\text{Im}}(\boldsymbol{\mu}) \hat{\mathcal{X}}_n^{\text{Im}}) \hat{\mathbf{u}}_\Gamma(\boldsymbol{\mu}) = \sum_{n=1}^{m_F} (\hat{\rho}_n^{\text{Re}}(\boldsymbol{\mu}) \hat{F}_n^{\text{Re}} + i \hat{\rho}_n^{\text{Im}}(\boldsymbol{\mu}) \hat{F}_n^{\text{Im}}), \quad (4.12)$$

which is a linear complex-valued system with n_Γ unknowns. Although solving (4.12) requires much less computation effort than that of the original model (4.3), this may not be a very small-scale problem. This persistence stems from the fact that the reformulated model (4.12) inherits the discretization-dependent characteristics of the original full-order system. Therefore, we again employ the VS method from Section 3.1 to derive the following reduced model representation of the interface problem:

$$\hat{\mathbf{u}}_\Gamma(\boldsymbol{\mu}) \approx \sum_{k=1}^{N_\Gamma} \left(\zeta_\Gamma^{k,\text{Re}}(\boldsymbol{\mu}) \mathbf{c}_\Gamma^{k,\text{Re}} + i \zeta_\Gamma^{k,\text{Im}}(\boldsymbol{\mu}) \mathbf{c}_\Gamma^{k,\text{Im}} \right), \quad (4.13)$$

where N_Γ is the number of the separate terms, $\mathbf{c}_\Gamma^{k,\text{Re}}$ and $\mathbf{c}_\Gamma^{k,\text{Im}}$, $k = 1, \dots, N_\Gamma$ are n_Γ -dimension vectors. Using this separate representation for new random inputs, we can directly obtain the solution to the interface problem without solving the Eq. (4.3) or the Eq. (4.12) again. This approach has the potential of very high efficiency, since it only requires computing the coefficients $\zeta_\Gamma^{k,\text{Re}}(\boldsymbol{\mu})$ and $\zeta_\Gamma^{k,\text{Im}}(\boldsymbol{\mu})$ for new parameter values while reusing the precomputed spatial basis $\mathbf{c}_\Gamma^{k,\text{Re}}$ and $\mathbf{c}_\Gamma^{k,\text{Im}}$.

4.3. Reduced model for subdomain problems

As discussed in Section 3.2, once having the solution $\hat{\mathbf{u}}_\Gamma^j(\boldsymbol{\mu})$ of the interface problem, the interior solution $\hat{\mathbf{u}}_I^j(\boldsymbol{\mu})$ in the subdomain D_j can be obtained by solving the complex-valued equation

$$\begin{aligned} & \left[A_{II}^{j,\text{Re}}(\boldsymbol{\mu}) + i A_{II}^{j,\text{Im}}(\boldsymbol{\mu}) \right] \left\{ \hat{\mathbf{u}}_I^{j,\text{Re}}(\boldsymbol{\mu}) + i \hat{\mathbf{u}}_I^{j,\text{Im}}(\boldsymbol{\mu}) \right\} \\ & = \left\{ \mathbf{f}_I^{j,\text{Re}}(\boldsymbol{\mu}) + i \mathbf{f}_I^{j,\text{Im}}(\boldsymbol{\mu}) \right\} - \left[A_{IT}^{j,\text{Re}}(\boldsymbol{\mu}) + i A_{IT}^{j,\text{Im}}(\boldsymbol{\mu}) \right] \left\{ \hat{\mathbf{u}}_\Gamma^{j,\text{Re}}(\boldsymbol{\mu}) + i \hat{\mathbf{u}}_\Gamma^{j,\text{Im}}(\boldsymbol{\mu}) \right\}. \end{aligned}$$

To reduce the computational cost, we apply the VS method in order to construct an efficient surrogate model. Using the affine decomposition of the matrices $A_{II}^{j,\text{Re}}(\boldsymbol{\mu})$, $A_{II}^{j,\text{Im}}(\boldsymbol{\mu})$, $A_{IT}^{j,\text{Re}}(\boldsymbol{\mu})$, $A_{IT}^{j,\text{Im}}(\boldsymbol{\mu})$ in (4.4) and the vectors $\mathbf{f}_I^{j,\text{Re}}(\boldsymbol{\mu})$, $\mathbf{f}_I^{j,\text{Im}}(\boldsymbol{\mu})$ in (4.10) and the affine surrogate model of the interface solutions $\hat{\mathbf{u}}_\Gamma^{j,\text{Re}}(\boldsymbol{\mu})$, $\hat{\mathbf{u}}_\Gamma^{j,\text{Im}}(\boldsymbol{\mu})$ in Eq. (4.13), we get

$$\hat{\mathbf{u}}_I^j(\boldsymbol{\mu}) \approx \sum_{k=1}^{N_I} \left(\zeta_I^{jk,\text{Re}}(\boldsymbol{\mu}) \mathbf{c}_I^{jk,\text{Re}} + i \zeta_I^{jk,\text{Im}}(\boldsymbol{\mu}) \mathbf{c}_I^{jk,\text{Im}} \right),$$

where N_I is the number of the separate terms, $\mathbf{c}_I^{jk,\text{Re}}$ and $\mathbf{c}_I^{jk,\text{Im}}$, $k = 1, \dots, N_I$ are vectors independent of $\boldsymbol{\mu}$.

In summary, for time-dependent problems, after computing the corresponding solutions using the preconstructed surrogate models for both the interface problem and subproblems with new parameter inputs, we can obtain the temporal solution through an inverse Fourier transform. The entire computational process is independent of the spatial discretization of the original system and requires no time-stepping iterations, thus demonstrating remarkable efficiency.

Notably, the online stage of the proposed approach can be much more efficient, as the surrogate models of the interface problem and subproblems (as well as their real and imaginary parts) allow parallel computation across various frequency values.

Remark 4.1. The proposed method can be extended to non-affine, nonlinear, and multi-physical parameterized problems. For the non-affine case, the VS method [34] for a multi-variable function and the empirical interpolation method can be used to obtain the affine form. In multi-physical parameterized problems, we can use the domain decomposition to reduce a global multi-physical problem to a local single-physical problem. Then we perform the VS method for each local problem. To obtain the Fourier transform of the nonlinear case, the convolution property will be used.

5. Numerical Experiments

Here we present three numerical examples. The corresponding numerical results demonstrate the applicability and efficiency of the method proposed. In Section 5.1, the method is applied to the heat equation. After that we apply it to the reaction diffusion equation in Sections 5.2.1 and 5.2.2 with different settings. All numerical experiments were run in Python on a Dell workstation with Intel Xeon W-2295 CPU @3.00GHz and 64GB of RAM.

To quantify the accuracy of the proposed method, the average relative error ϵ_u for the solution of the original dynamical systems is defined by

$$\epsilon_u := \frac{1}{M} \sum_{k=1}^M \frac{\|u(\mathbf{x}, t; \xi_k) - u_N(\mathbf{x}, t; \xi_k)\|_{L^2([0, T]; \mathcal{V}_h)}}{\|u(\mathbf{x}, t; \xi_k)\|_{L^2([0, T]; \mathcal{V}_h)}}, \quad (5.1)$$

where M is the number of samples, $u_N(\mathbf{x}, t; \xi_k)$ the approximate solution obtained by the proposed method which is denoted FT-DD-VS, and $u(\mathbf{x}, t; \xi_k)$ the reference solution. In the following examples, the reference solution is calculated by FEM in space and the backward Euler scheme in time, denoted as FEM-BE.

Furthermore, we define the average relative error $\epsilon_{\hat{\mathbf{u}}}$ for the complex-valued equations as

$$\epsilon_{\hat{\mathbf{u}}} := \frac{1}{M} \sum_{k=1}^M \frac{|\hat{\mathbf{u}}(\boldsymbol{\mu}_k) - \hat{\mathbf{u}}_N(\boldsymbol{\mu}_k)|}{|\hat{\mathbf{u}}(\boldsymbol{\mu}_k)|}, \quad (5.2)$$

where $\hat{\mathbf{u}}_N(\boldsymbol{\mu}_k)$ is the approximate solution obtained by the VS method and $\hat{\mathbf{u}}(\boldsymbol{\mu}_k)$ the reference solution determined by FEM.

5.1. Heat equation

On the domain $D = [0, 1] \times [0, 1]$, we consider the following heat equation:

$$\frac{\partial u}{\partial t}(\mathbf{x}, t; \xi) - \nabla \cdot (c(\mathbf{x}; \xi) \nabla u(\mathbf{x}, t; \xi)) = f(\mathbf{x}, t; \xi),$$

where

$$\begin{aligned} u(\mathbf{x}, 0; \xi) &= 0, \quad (\mathbf{x}, \xi) \in D \times \Omega, \\ u(\mathbf{x}, t; \xi) &= 0, \quad (\mathbf{x}, t; \xi) \in \partial D \times [0, T] \times \Omega. \end{aligned} \quad (5.3)$$

The analytical solution of this problem is

$$u(\mathbf{x}, t; \xi) = \frac{1}{\pi} \frac{t}{t^2 + 1} \sin \pi x_1 \sin \pi x_2,$$

so that the source function f has the form

$$f(\mathbf{x}, t; \xi) = \left(\frac{1}{\pi} \frac{1-t^2}{(t^2+1)^2} + 2\pi \frac{t}{t^2+1} c(\mathbf{x}; \xi) \right) \sin \pi x_1 \sin \pi x_2.$$

The original domain D is divided into the subdomains $D_1 = [0, 0.5] \times [0, 1]$ and $D_2 = [0.5, 1] \times [0, 1]$, and the coefficient $c(\mathbf{x}; \xi)$ is defined as

$$c(\mathbf{x}; \xi) = \begin{cases} \xi_1, & \mathbf{x} \in D_1, \\ 2\xi_2, & \mathbf{x} \in D_2. \end{cases}$$

Therefore, the source function has different representations in subdomains D_1 and D_2 . Note that $\xi := (\xi_1, \xi_2) \in [1, 2]^2$ and $T = 1$. The reference solution is obtained by FEM in space with the mesh size $h = 0.02$ and the backward Euler scheme in time with the step size $\tau = 5 \times 10^{-4}$. Besides, $|\Xi|$ denotes the cardinality of the sample set Ξ and is set to $|\Xi| = 10$ for the offline stage in Algorithm 3.1. In this problem, we take $\omega^* = 20$ and $N_\omega = 20$, which suffices for determining an effective approximation.

First, we quantify the assembly errors for S and F in Step 1, which coincide with their affine approximation errors. Taking S_1 as an example, Fig. 1 plots the average relative error versus the number of separate terms N_{S_1} , computed as (5.2) with 10^3 samples. Note that the error diminishes monotonically as N_{S_1} increases. This implies that the VS method effectively approximates complex-valued problems. To construct an efficient surrogate model for the interface problem, we select $N_{S_1} = 4$, ensuring the relative approximation error of S_1 remains below 10^{-6} . Correspondingly, we set $N_{S_2} = 4$, $N_{F_1} = 4$, and $N_{F_2} = 4$.

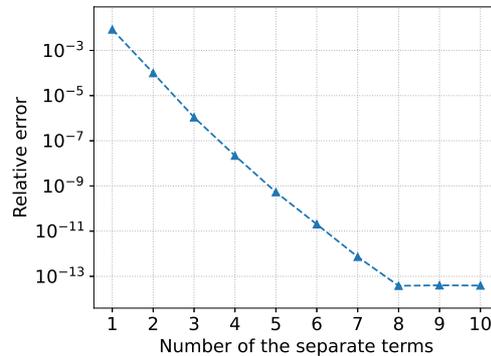


Figure 1: Average relative error versus the number of separate terms N_{S_1} .

Based on the affine approximations of S and F , Fig. 2 shows the average relative error versus the number of separate terms for interface problem N_Γ and subdomain problems N_I . The errors ϵ_{u_Γ} and ϵ_{u_I} are calculated by formula (5.2) with 10^3 samples.

We find that the relative error of subdomain problems decreases rapidly and stabilizes when $N_I \geq 4$. The results demonstrate that the VS method yields accurate approximations for both interface and subdomain solutions, a capability critical for ensuring the reliability of our proposed method in parametric dynamical systems.

Fig. 2 shows that the number of separate terms $N_\Gamma = 1$ for the interface problem and $N_I = 4$ for the subdomain problems yields a surrogate model with the desired relative error while maintaining computational efficiency. Under this configuration ($N_\Gamma = 1$ and $N_I = 4$), we compute the relative error of the original dynamical system by (5.1). Fig. 3(a) illustrates the relative errors of the first 100 samples. Along with the temporal evolution, the relative error decreases — cf. Fig. 3(b). Notice that the proposed method can provide a good approximation for the solution of the original problem.

Fig. 4 presents the mean solutions at $t = 1$, computed by the proposed method and the FEM-BE scheme in two subdomains. The top row shows the reference solutions, while

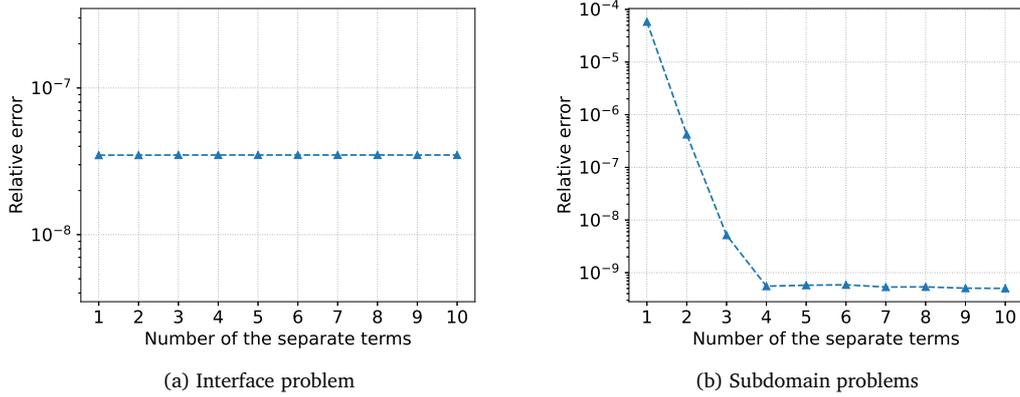


Figure 2: Average relative error versus the number of separate terms.

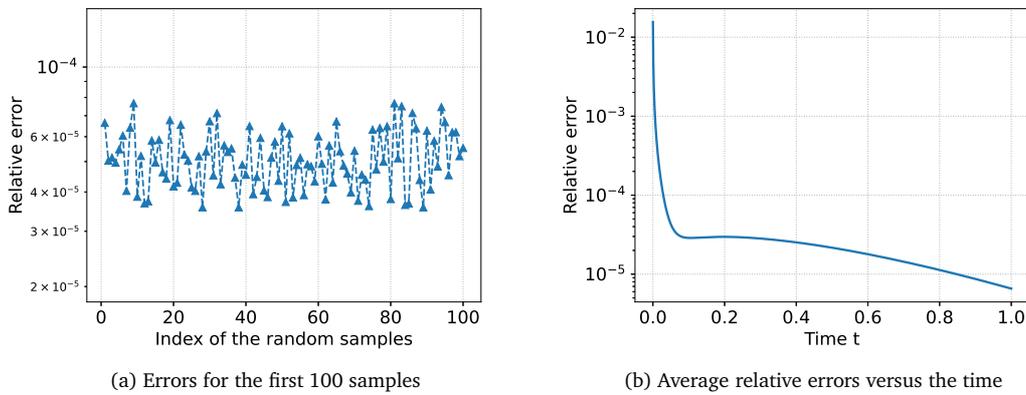


Figure 3: Comparison of the errors.

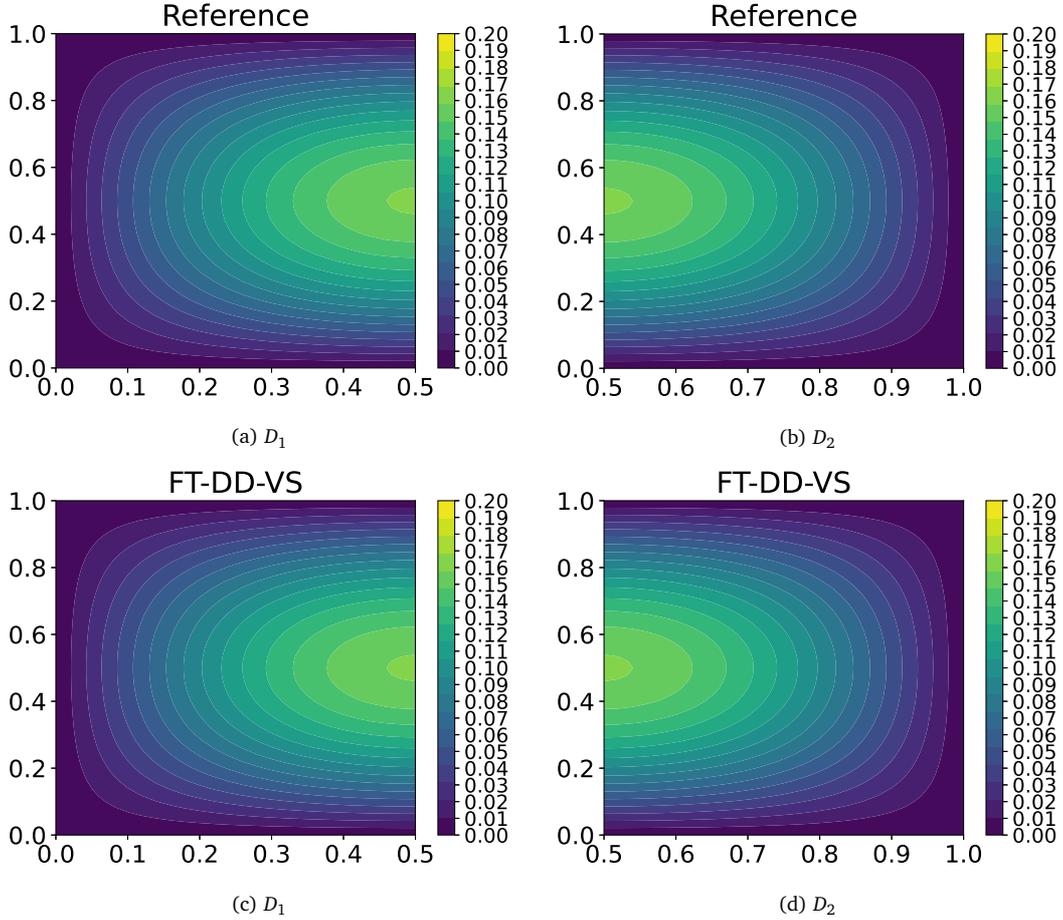


Figure 4: Comparison of the solutions in subdomains for the FEM-BE and FT-DD-VS methods.

the bottom row displays the corresponding results obtained using the proposed approach. It can be observed that the solution profiles and magnitudes produced by the proposed method closely match those of the reference.

Table 1 presents CPU time based on $M = 10^3$ random samples, which includes the offline CPU time (\mathcal{T}_{off}), online CPU time (\mathcal{T}_{on}), total CPU time (\mathcal{T}_{tot}), and the average total CPU time (T_{tot}). The magnitude of the average total CPU time by the proposed method is much smaller than that of the FEM-BE method. It demonstrates that the proposed method can achieve a good trade-off in both approximation accuracy and computational efficiency for this numerical experiment.

Table 1: Comparison of the CPU time for FT-DD-VS and FEM-BE.

Algorithm	\mathcal{T}_{off}	\mathcal{T}_{on}	\mathcal{T}_{tot}	T_{tot}
FT-DD-VS	$2.90 \times 10^2 s$	$9.39 \times 10^2 s$	$1.23 \times 10^3 s$	1.23s
FEM-BE	\	\	$3.98 \times 10^5 s$	$3.98 \times 10^2 s$

5.2. Reaction diffusion equation

In this subsection, we consider the reaction diffusion equation

$$\frac{\partial u}{\partial t}(\mathbf{x}, t; \xi) = \nabla \cdot (c_1(\mathbf{x}; \xi) \nabla u(\mathbf{x}, t; \xi)) - c_2(\mathbf{x}; \xi) u(\mathbf{x}, t; \xi) + f(\mathbf{x}, t; \xi) \quad (5.4)$$

with the initial and boundary conditions (5.3). The original domain $D = [0, 1] \times [0, 1]$ is divided into two subdomains $D_1 = [0, 0.5] \times [0, 1]$ and $D_2 = [0.5, 1] \times [0, 1]$.

5.2.1. The first scenario

First, we consider the source function

$$f(\mathbf{x}, t; \xi) = e^{-t^2} e^{5(x_1+x_2)^2},$$

and the coefficients

$$c_1(\mathbf{x}; \xi) = \begin{cases} 100\xi_1, & \mathbf{x} \in D_1, \\ 10\xi_2, & \mathbf{x} \in D_2, \end{cases} \quad c_2(\mathbf{x}; \xi) = \begin{cases} \xi_3, & \mathbf{x} \in D_1, \\ 0.1\xi_4, & \mathbf{x} \in D_2, \end{cases}$$

where $\xi := (\xi_1, \xi_2, \xi_3, \xi_4) \in [1, 2]^4$. The reference solution is calculated by FEM in space with the mesh size $h = 0.05$ and the backward Euler scheme in time with the step size $\tau = 5 \times 10^{-4}$. We take $\omega^* = 15$, $N_\omega = 15$, and $|\Xi| = 10$ for training in the offline stage of Algorithm 3.1. In this problem, we select $N_{S_1} = N_{S_2} = N_{F_1} = N_{F_2} = 4$ and focus on the results of the interface problem and the subproblems.

Fig. 5 plots the average relative errors of the interface problem and the subproblems against the number of the separate terms N , where the average relative errors are computed based on 10^3 samples. The relative error decreases as the number of separate terms increases, as shown in Fig. 5(a) for the interface problem. For the subdomain problems, the relative error stabilizes when the number of separate terms is no less than 2. From the

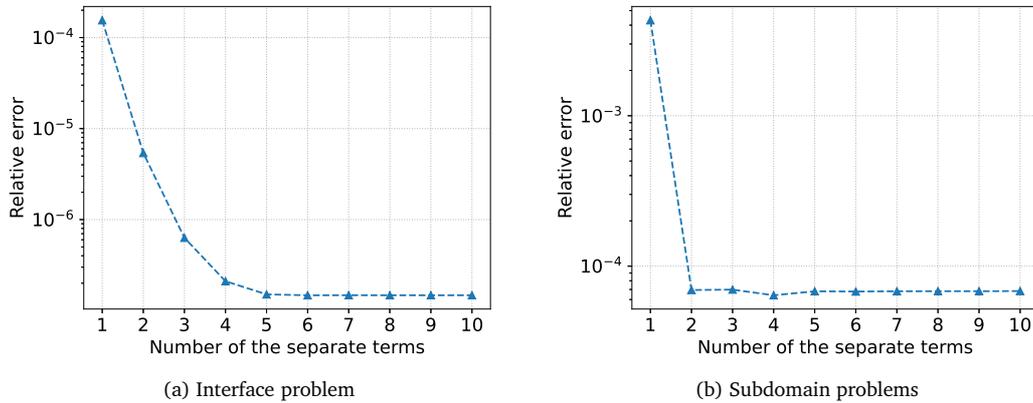


Figure 5: Average relative error versus the number of separate terms.

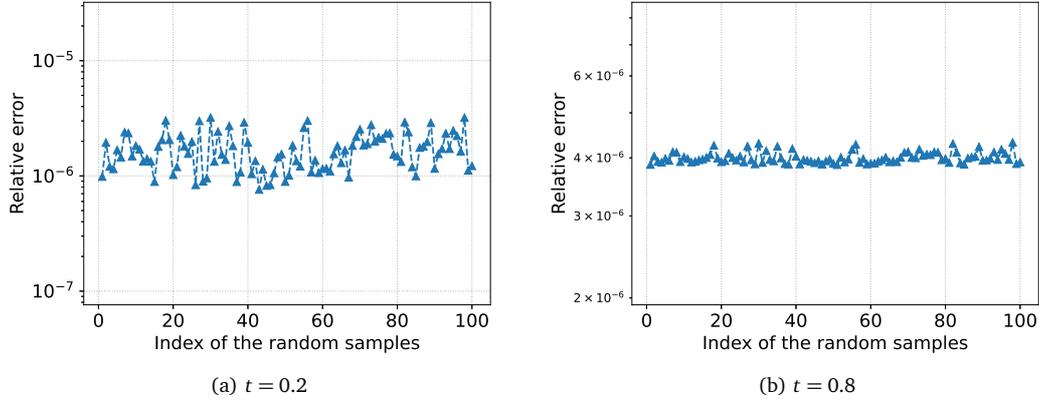


Figure 6: Errors for the first 100 samples.

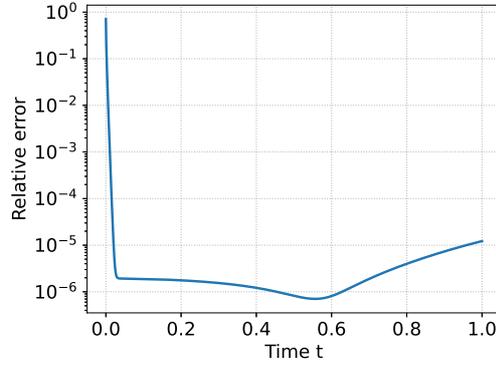


Figure 7: Average relative errors versus time.

figures, we can see that as the number of the separate terms increases, the approximation first becomes more accurate and then stable.

To illustrate the individual relative error of the approximation in Fig. 6, we set $N_T = 3$ and $N_I = 2$. Fig. 6 demonstrates the relative errors of the first 100 samples at $t = 0.2$ and $t = 0.8$. The range of the relative errors against the sample index at $t = 0.2$ is wider than that at $t = 0.8$. Besides, we also plot the average relative errors versus time in Fig. 7. Here, the relative error first decreases and then increases one order of magnitude compared to that at $t = 0.6$. This illustrates the proposed method can provide a good approximation for the original problem (5.4) after $t = 0.02$ with the relative error less than 10^{-5} .

Finally, we present the CPU times based on $M = 10^3$ random samples for both the proposed method and the FEM-BE method in Table 2. The average total CPU time is reduced

Table 2: Comparison of the CPU time for FT-DD-VS and FEM-BE.

Algorithm	\mathcal{T}_{off}	\mathcal{T}_{on}	\mathcal{T}_{tot}	T_{tot}
FT-DD-VS	$2.87 \times 10^1 s$	$2.48 \times 10^2 s$	$2.77 \times 10^2 s$	$2.77 \times 10^{-1} s$
FEM-BE	\	\	$7.56 \times 10^4 s$	$7.56 \times 10^1 s$

from $7.56 \times 10^1 s$ to $2.77 \times 10^{-1} s$ using the proposed method. In summary, the proposed method significantly reduces computational time compared to the FEM-BE method, while maintaining high accuracy.

5.2.2. The second scenario

In this example we set

$$f(\mathbf{x}, t; \xi) = \frac{1-t^2}{(1+t^2)^2} e^{5(x_1+x_2)^2},$$

$$c_1(\mathbf{x}; \xi) = \begin{cases} 0, & \mathbf{x} \in D_1, \\ 10\xi_1, & \mathbf{x} \in D_2, \end{cases}$$

$$c_2(\mathbf{x}; \xi) = \begin{cases} 100\xi_2, & \mathbf{x} \in D_1, \\ 0, & \mathbf{x} \in D_2, \end{cases}$$

and $\xi := (\xi_1, \xi_2) \in [3, 4]^2$. The reference solution is calculated using FEM in space with a mesh size $h = 0.05$ and the backward Euler scheme in time with a step size $\tau = 10^{-3}$. In this problem, we take $\omega^* = 15$, $N_\omega = 15$, $|\Xi| = 10$, and set $N_{S_1} = N_{S_2} = N_{F_1} = N_{F_2} = 4$.

Fig. 8 presents the average relative errors of the interface problem and the subproblems against the number of the separate terms N . The average relative errors are calculated from 10^3 random samples. The figures show that the approximation accuracy for the interface problem improves as N increases. However, the error of the subproblems first decreases and then plateaus as N increases. Based on the results, we set $N_\Gamma = N_I = 4$ to perform the Fourier inversion and ensure the approximation accuracy.

To illustrate the approximation accuracy, the relative errors of the first 100 samples along with the errors versus the time are plotted in Fig. 9. The results in Fig. 9(a) demonstrate that the proposed method provides accurate approximations for the first 100 random samples, because the curve of relative errors stay below the threshold of 10^{-3} . In Fig. 9(b),

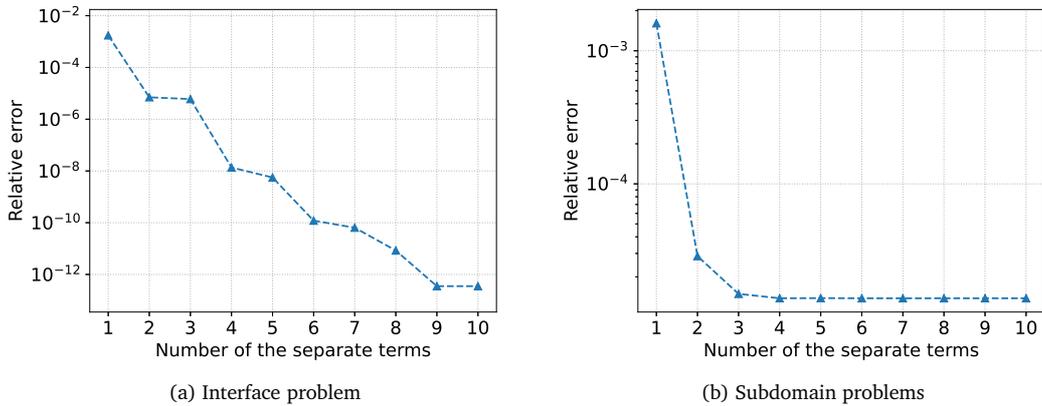


Figure 8: Average relative error versus the number of separate terms.

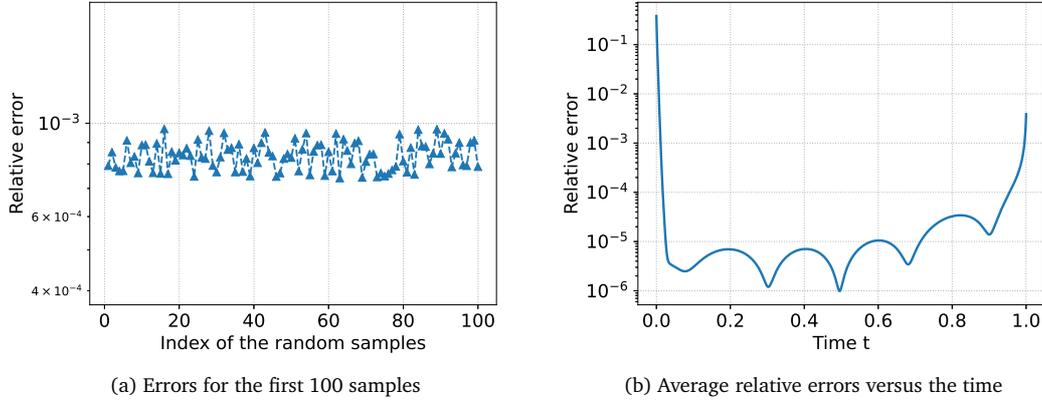


Figure 9: Comparison of the errors.

we can see that the relative errors within $t \in [0.1, 0.9]$ exhibits a clear periodic variation pattern. Moreover, the average relative errors lies in the interval $[10^{-6}, 10^{-4}]$ when $t \in [0.02, 0.96]$. In this example, the proposed method shows a good performance for $t \in [0.02, 0.96]$.

Finally, Table 3 compares the proposed method with the FEM-BE approach in terms of the computational cost. We find that the CPU time of the proposed method is orders of magnitude smaller than that of the FEM-BE method. The results demonstrate that the proposed method maintains high accuracy while achieving significant computational time reduction compared to the FEM-BE approach.

Table 3: Comparison of the CPU time for FT-DD-VS and FEM-BE.

Algorithm	\mathcal{T}_{off}	\mathcal{T}_{on}	\mathcal{T}_{tot}	T_{tot}
FT-DD-VS	$2.81 \times 10^1 s$	$2.02 \times 10^2 s$	$2.30 \times 10^2 s$	$2.30 \times 10^{-1} s$
FEM-BE	\	\	$3.69 \times 10^4 s$	$3.69 \times 10^1 s$

6. Conclusions

We propose a model order reduction method for parametric dynamical systems, in which time-dependent problems are first transformed into frequency-variable elliptic equations using the Fourier transform. The converted equations become time-independent, allowing for parallel computation across different frequency values. To construct an efficient and accurate separable approximation for the resulting complex-valued elliptic equations, we employ the variable-separation-based domain decomposition method. This approach leverages domain decomposition to convert the solution of the complex elliptic problem into solving interface problem and subproblems, making it well-suited for handling complex scenarios. Subsequently, the VS method is applied to establish surrogate models for both the interface problem and the subproblems. Based on this framework, the solution to

the time-dependent problem is obtained by applying the inverse Fourier transform to the elliptic equation solutions. A key advantage of the method is its ability to decouple offline and online computations, ensuring that the online phase remains entirely independent of spatial discretization, thus achieving high efficiency. As a result, the overall computational cost of the proposed method is significantly lower than that of standard finite element method, a fact supported by numerical experiments. Nevertheless, the Fourier transform has inherent limitations, since the functions to be transformed should satisfy certain assumptions. Consequently, this method faces significant constraints when applied to time-dependent problems. In future work, we aim to explore domain decomposition directly on time-dependent problems to achieve efficient computation, and focus on extending to general parameter distributions by adapting the subdomain partitioning strategy.

Acknowledgments

The authors would like to thank the two anonymous referees for their helpful comments that improved the quality of the manuscript.

This work was supported by the National Key R & D Program of China (Grant No. 2021YFA001300), by the National Natural Science Foundation of China (Grant Nos. 12401567, 12271150, 12471405), by the Hunan Provincial Natural Science Foundation of China (Grant No. 2023JJ10001), by the Science and Technology Innovation Program of Hunan Province (Grant No. 2022RC1190), and by the Hunan Provincial Innovation Foundation for Postgraduate (Grant No. CX20250499).

References

- [1] P. Benner, S. Gugercin and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev. **57**, 483–531 (2015).
- [2] S.L. Brunton and J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press (2019).
- [3] T.F. Chan, *Analysis of preconditioners for domain decomposition*, SIAM J. Numer. Anal. **24**, 382–390 (1987).
- [4] T.F. Chan and T.P. Mathew, *Domain decomposition algorithms*, Acta Numer. **3**, 61–143 (1994).
- [5] L. Chen, Y. Chen, Q. Li and Z. Zhang, *Stochastic domain decomposition based on variable-separation method*, Comput. Methods Appl. Mech. Engrg. **418**, Paper No. 116538 (2024).
- [6] L. Chen, Y. Chen, Q. Li and T. Zhou, *A dynamical variable-separation method for parameter-dependent dynamical systems*, SIAM J. Sci. Comput. **47**, A1783–A1808 (2025).
- [7] W. Chen, Q. Wang, J.S. Hesthaven and C. Zhang, *Physics-informed machine learning for reduced-order modeling of nonlinear problems*, J. Comput. Phys. **446**, Paper No. 110666 (2021).
- [8] Y. Chen, J.S. Hesthaven, Y. Maday and J. Rodríguez, *Certified reduced basis methods and output bounds for the harmonic Maxwell’s equations*, SIAM J. Sci. Comput. **32**, 970–996 (2010).
- [9] Y. Chen, J. Jakeman, C. Gittelsohn and D. Xiu, *Local polynomial chaos expansion for linear differential equations with high dimensional random inputs*, SIAM J. Sci. Comput. **37**, A79–A102 (2015).
- [10] J. Douglas, J.E. Santos and D. Sheen, *Approximation of scalar waves in the space-frequency domain*, Math. Models Meth. Appl. Sci. **4**, 509–531 (1994).

- [11] J. Douglas, J.E. Santos, D. Sheen and L.S. Bennethum, *Frequency domain treatment of one-dimensional scalar waves*, Math. Model Meth. Appl. Sci. **3**, 171–194 (1993).
- [12] M. Drohmann, B. Haasdonk and M. Ohlberger, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput. **34**, A937–A969 (2012).
- [13] M. Esmaily and D. Jia, *A stabilized formulation for the solution of the incompressible unsteady stokes equations in the frequency domain*, J. Comput. Phys. **473**, Paper No. 111736 (2023).
- [14] X. Feng and D. Sheen, *An elliptic regularity estimate for a problem arising from the frequency domain treatment of waves*, Trans. Ams. Math. Soc. **346**, 475–487 (1994).
- [15] J. Fu, D. Xiao, R. Fu, C. Li, C. Zhu, R. Arcucci and I.M. Navon, *Physics-data combined machine learning for parametric reduced-order modelling of nonlinear dynamical systems in small-data regimes*, Comput. Methods Appl. Mech. Engrg. **404**, Paper No. 115771 (2023).
- [16] R. Ghaffari and F. Ghoreishi, *Reduced collocation method for time-dependent parametrized partial differential equations*, Bull. Iran. Math. Soc. **45**, 1487–1504 (2019).
- [17] M.A. Grepl and A.T. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations*, M2AN Math. Model. Numer. Anal. **39**, 157–181 (2005).
- [18] M. Guo and J.S. Hesthaven, *Data-driven reduced order modeling for time-dependent problems*, Comput. Methods Appl. Mech. Engrg. **345**, 75–99 (2019).
- [19] B. Haasdonk and M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, ESAIM Math. Model. Numer. Anal. **42**, 277–302 (2008).
- [20] B. Haasdonk and M. Ohlberger, *Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition*, Math. Comput. Model. Dyn. Syst. **17**, 145–161 (2011).
- [21] M. Hadigol, A. Doostan, H.G. Matthies and R. Niekamp, *Partitioned treatment of uncertainty in coupled domain problems: A separated representation approach*, Comput. Methods Appl. Mech. Engrg. **274**, 103–124 (2014).
- [22] M. Heinkenschloss and H. Nguyen, *Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems*, SIAM J. Sci. Comput. **28**, 1001–1028 (2006).
- [23] M.W. Hess and P. Benner, *Fast evaluation of time-harmonic Maxwell's equations using the reduced basis method*, IEEE Trans. Microw. Theory Tech. **61**, 2265–2274 (2013).
- [24] J.S. Hesthaven, C. Pagliantini and G. Rozza, *Reduced basis methods for time-dependent problems*, Acta Numer. **31**, 265–345 (2022).
- [25] J.S. Hesthaven, G. Rozza and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer (2016).
- [26] L. Jiang and Q. Li, *Model reduction method using variable-separation for stochastic saddle point problems*, J. Comput. Phys. **354**, 43–66 (2018).
- [27] Y. Kim, Y. Choi, D. Widemann and T. Zohdi, *A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder*, J. Comput. Phys. **451**, Paper No. 110841 (2022).
- [28] C.O. Lee, J. Lee and D. Sheen, *Frequency domain formulation of linearized Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg. **187**, 351–362 (2000).
- [29] C.O. Lee, J. Lee, D. Sheen and Y. Yeom, *A frequency-domain parallel method for the numerical approximation of parabolic problems*, Comput. Methods Appl. Mech. Engrg. **169**, 19–29 (1999).
- [30] J. Lee and D. Sheen, *An accurate numerical inversion of Laplace transforms based on the location*

- of their poles, *Comput. Math. Appl.* **48**, 1415–1423 (2004).
- [31] J. Lee and D. Sheen, *A parallel method for backward parabolic problems based on the Laplace transformation*, *SIAM J. Numer. Anal.* **44**, 1466–1486 (2006).
- [32] S. Lee and D. You, *Data-driven prediction of unsteady flow over a circular cylinder using deep learning*, *J. Fluid Mech.* **879**, 217–254 (2019).
- [33] K. Li, T.Z. Huang, L. Li and S. Lanteri, *Non-intrusive reduced-order modeling of parameterized electromagnetic scattering problems using cubic spline interpolation*, *J. Sci. Comput.* **87**, 1–29 (2021).
- [34] Q. Li and L. Jiang, *A novel variable-separation method based on sparse and low rank representation for stochastic partial differential equations*, *SIAM J. Sci. Comput.* **39**, A2879–A2910 (2017).
- [35] Q. Li and P. Zhang, *A variable-separation method for nonlinear partial differential equations with random inputs*, *SIAM J. Sci. Comput.* **42**, A723–A750 (2020).
- [36] Q. Liao and K. Willcox, *A domain decomposition approach for uncertainty analysis*, *SIAM J. Sci. Comput.* **37**, A103–A133 (2015).
- [37] J. Mandel, *Balancing domain decomposition*, *Comm. Numer. Methods Engrg.* **9**, 233–241 (1993).
- [38] M. Mohebbujaman, L.G. Rebholz and T. Iliescu, *Physically constrained data-driven correction for reduced-order modeling of fluid flows*, *Int. J. Numer. Meth. Fl.* **89**, 103–122 (2019).
- [39] L. Mu and G. Zhang, *A domain decomposition model reduction method for linear convection-diffusion equations with random coefficients*, *SIAM J. Sci. Comput.* **41**, A1984–A2011 (2019).
- [40] A. Nouy and O.P. Le Maître, *Generalized spectral decomposition for stochastic nonlinear problems*, *J. Comput. Phys.* **228**, 202–235 (2009).
- [41] X. Pan and D. Xiao, *Domain decomposition for physics-data combined neural network based parametric reduced order modelling*, *J. Comput. Phys.* **519**, Paper No. 113452 (2024).
- [42] T.P. Sapsis and P.F.J. Lermusiaux, *Dynamically orthogonal field equations for continuous stochastic dynamical systems*, *Phys. D* **238**, 2347–2360 (2009).
- [43] A. Sarkar, N. Benabbou and R. Ghanem, *Domain decomposition of stochastic PDEs: Theoretical formulations*, *Internat. J. Numer. Methods Engrg.* **77**, 689–701 (2009).
- [44] J. Schöberl, *Efficient contact solvers based on domain decomposition techniques*, *Comput. Math. Appl.* **42**, 1217–1228 (2001).
- [45] J.K. Seo and B.C. Shin, *Reduced-order modeling using the frequency-domain method for parabolic partial differential equations*, *AIMS Math.* **8**, 15255–15268 (2023).
- [46] M.K. Sleeman and M. Yano, *Goal-oriented model reduction for parametrized time-dependent nonlinear partial differential equations*, *Comput. Methods Appl. Mech. Engrg.* **388**, Paper No. 114206 (2022).
- [47] W. Subber and S. Loisel, *Schwarz preconditioners for stochastic elliptic PDEs*, *Comput. Methods Appl. Mech. Engrg.* **272**, 34–57 (2014).
- [48] W. Subber and A. Sarkar, *A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner*, *J. Comput. Phys.* **257**, 298–317 (2014).
- [49] R. Swischuk, L. Mainini, B. Peherstorfer and K. Willcox, *Projection-based model reduction: Formulations for physics-based machine learning*, *Comput. Fluids* **179**, 704–717 (2019).
- [50] D. Xiao, *Error estimation of the parametric non-intrusive reduced order model using machine learning*, *Comput. Methods Appl. Mech. Engrg.* **355**, 513–534 (2019).
- [51] D. Xiao, F. Fang, C.E. Heaney, I.M. Navon and C.C. Pain, *A domain decomposition method for the non-intrusive reduced order modelling of fluid flow*, *Comput. Methods Appl. Mech. Engrg.* **354**, 307–330 (2019).

- [52] D. Xiao, C.E. Heaney, F. Fang, L. Mottet, R. Hu, D.A. Bistrrian, E. Aristodemou, I.M. Navon and C.C. Pain, *A domain decomposition non-intrusive reduced order model for turbulent flows*, *Comput. Fluids* **182**, 15–27 (2019).
- [53] J. Xu and K. Duraisamy, *Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics*, *Comput. Methods Appl. Mech. Engrg.* **372**, Paper No. 113379 (2020).