Journal of Computational and Applied Mathematics 442 (2024) 115673

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A new treatment of boundary conditions in PDE solution with Galerkin methods via Partial Integral Equation framework

Yulia T. Peet *, Matthew M. Peet

School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287, USA

ARTICLE INFO

Keywords:
Partial Differential Equations
Boundary conditions
Galerkin methods
Chebyshev polynomials

ABSTRACT

We present a new mathematical framework for solution of Partial Differential Equations (PDEs), which is based on an exact transformation of the underlying PDE that removes the boundary constraints from the solution state and moves them into the dynamics of the equivalent transformed equation. The framework is based on a Partial Integral Equation (PIE) representation of a PDE or a system of PDEs, where Partial Integral Equation does not require boundary conditions on its solution state. The PDE-PIE framework allows for a development of a generalized and consistent treatment of boundary conditions in constructing spectrally convergent solution approximations to a broad class of linear PDEs with non-constant coefficients governed by non-periodic boundary conditions, including, e.g., Dirichlet, Neumann and Robin boundaries, among others. The significance of this result is that a solution to almost any linear PDE in a form of a function series approximation can now be systematically constructed, irrespective of the boundary conditions. Furthermore, in many cases, the resulting ODE system for the expansion coefficients can now be integrated analytically in time, which allows us to obtain solution approximations to a broad class of unsteady PDEs with unprecedented accuracy. We present several PDE solution examples in one spatial variable implemented with the developed PIE-Galerkin methodology using both analytical and numerical integration in time. We also present comparison of the PIE methods with some classical direct PDE solution methods, further demonstrating advantages and potential limitations of the PIE approach. The developed framework can be naturally extended to multiple spatial dimensions and, potentially, to nonlinear problems.

Science is a Differential Equation. Religion is a Boundary Condition. – Alan Turing (1912–1954).

1. Introduction

Models in physical, biological and engineering sciences are frequently represented by Partial Differential Equations (PDEs) [1–3]. Efficient and accurate methods for analytical and numerical solution of PDEs are crucial for modeling, analysis and control of fundamental processes in these systems. Development of generalized techniques for treatment of PDEs in mathematical and computational sciences has been, however, hampered by the need to enforce boundary conditions.

Boundary conditions are the auxiliary constraints on the solution, its partial derivatives, or a combination of thereof, whose imposition is required to guarantee a unique solution to a PDE [4,5]. To enforce boundary conditions, a solution function is typically split into a homogeneous part that satisfies a specified type of boundary conditions but with zero values, and an inhomogeneous

* Corresponding author.
E-mail addresses: ypeet@asu.edu (Y.T. Peet), mpeet@asu.edu (M.M. Peet).

https://doi.org/10.1016/j.cam.2023.115673
Received 10 February 2023; Received in revised form 23 October 2023
Available online 29 November 2023

part that must satisfy non-zero boundary conditions [6,7]. For the inhomogeneous part, any appropriately smooth function defined on the solution domain that conforms to the boundary values but not necessarily satisfies the PDE can be used, which will result in a modification of the right-hand side of the PDE. However, it is the search for a homogeneous solution, which is required to satisfy *both* the PDE and the homogeneous boundary conditions, that represents the biggest challenge and has hindered a development of a unifying theoretical framework for solving PDE equations for more than two centuries.

The easiest way to impose boundary conditions is to seek a solution to a PDE in terms of functions that already satisfy the boundary conditions, which is done in the so-called Galerkin methods [8]. Unfortunately, such basis functions are readily available only for a limited class of problems, e.g., the ones with periodic boundary conditions, for which Fourier methods based onto expansion into harmonic bases offer an elegant, efficient, and generalizable approach to the solution of PDEs with periodic boundaries [9]. For boundary conditions other than periodic, the picture is more obscure. An unfortunate fact to accept is that there are no convenient basis functions (viz. harmonic functions or classical orthogonal polynomials) that satisfy general, non-periodic boundary conditions. This yields, in a classical PDE analysis framework, three options: (1) construct more sophisticated basis functions from the primary ones that do satisfy boundary conditions [10,11], (2) enforce boundary conditions discretely on the expansion coefficients [12–14], (3) enforce boundary conditions in a weak form, by introducing penalty terms or Lagrange multipliers into the variational form of the equations [15–17]. The problem with the first approach is that it leads to a complicated basis that depends on the order of equations and on the boundary conditions [10,11,18,19], limiting the generalizability of approach. The second option, which is typically used in conjunction with either tau methods [12,14] or nodal/collocation methods [13,20,21], is inherently tied to a discretization, and thus has limited options for providing generalized close-form solutions that are useful for analysis and control of continuous models [22–24]. Additionally, a discrete enforcement of boundary conditions requires an ad-hoc modification of the discrete matrix operators, which can lead to ill-conditioned matrices and affect stability and accuracy [9,25,26]. Weak enforcement of the boundary conditions attempts to circumvent the above deficiencies [27,28]. However, this method introduces a tunable penalty parameter, which is not known from the first principles, problem-dependent, and leads to a lack of robustness of the solution [28–30]. Moreover, a weak imposition of boundary conditions forfeits the possibility of satisfying the conservation laws in a strong form, which, in some cases, e.g. for hyperbolic systems, is highly desirable [31–33].

In this paper, we present a conceptually new approach to address the problems associated with the enforcement of boundary conditions in the solution of PDEs. Specifically, we exploit a novel Partial Integral Equation (PIE) framework for representation of Partial Differential Equations [24]. In this framework, PIEs can be used to equivalently represent the solution of PDEs, yet require no boundary conditions on their solution variables. This is due to the fact that solutions of the PIE equations are expressed using a so-called "fundamental state", which consists of specially constructed functions that include derivatives of the primary solution; these functions lie in a space of L_2 square-integrable functions and require no boundary conditions. Instead, the effect of boundary conditions (both homogeneous and inhomogeneous) is *analytically* incorporated into the PIE dynamics through construction of the corresponding partial-integral operators. This integral representation essentially acts to move the boundary conditions from the realm of "religion" (artificial constraints on a solution) to the realm of "science" (integro-differential equations). Significantly, by solving PIEs, we are now free to represent the solution using any choice of an approximation space without the need to impose the boundary conditions on the solution functions in that space!

Since the idea of solving boundary value problems by relating the boundary condition functions to the interior solution resonates with several other techniques in mathematics, here we contrast our approach with the popular methods of Green functions [34–36] and boundary integral equations (BIE) [37–39]. Both Green functions and BIE approaches require a knowledge of the fundamental solutions of the corresponding differential operator, while no such a-priori knowledge is required in the current approach. Note that the "fundamental state" in a PIE is completely different from the "fundamental solution", which is a response of a linear differential operator to an impulse forcing [40,41]. In a classical Green functions approach, the acquired fundamental solutions are also required to satisfy homogeneous boundary conditions. In a BIE formulation, this requirement is relaxed, and solution satisfying the desired boundary conditions is formulated as a continuous superposition of arbitrary fundamental solutions, giving rise to an integral equation for the distribution density on the boundary of the domain [37–39]. Both these approaches are fundamentally different from the methodology presented in this paper, since, first of all, the integral operators act on the domain boundary in BIEs, and they act on the domain interior in PIEs, and, second, the PIE formulation does not require any a-priori knowledge of the fundamental solutions, which are only available for certain equations [34–36], and, for the case of non-constant coefficients, only approximately [42–44].

Several other approaches utilize a spatial integration of PDEs to eliminate function derivatives from a solution as a means to arrive at better-conditioned and more compact discrete matrix operators resulting from an integration as opposed to a differentiation procedure [45–47]. However, these approaches do not eliminate the boundary conditions and still have to enforce them on a solution, which is typically done at a discrete level by modifying the corresponding rows of discrete matrix operators to represent the algebraic constraints on the expansion coefficients [45,46], similar to the corresponding differentiation tau or collocation techniques.

In this regard, it is also useful to mention the Fokas method [48], which seeks to propose a unified transform procedure for solving initial–boundary value problems. The method involves performing joint Fourier-type integral transforms of the PDE together with initial and boundary conditions in space and time, solving for a global relation, and performing an inverse Fourier transform, which involves taking an indefinite integral over specified contours in a complex half-plane. This approach, however, is associated with certain difficulties as applied to a general case: first, it relies on the existence of a Lax pair [49], which can only be formulated for certain equations [48,50]; second, extension to a finite-size domain is challenging in that it yields an integrand which is no longer analytic, and requires evaluation of the residues at the complex poles, which may lack convergence and complicate the computation [51,52].

Previous research on PDEs and their numerical solution briefly surveyed above laid out a groundbreaking work in development of innovative numerical approaches for PDE discretizations [6,8,53] and established theories for analyzing their accuracy and stability [32,53,54]. However, generalized approaches of treating boundary conditions during the solution of PDEs on a firm theoretical basis are still lacking, limited to either an ad-hoc modification of the resulting discrete operators [12,45,46], or an ad-hoc modification of the governing equations [14,29,30]. We propose to remedy this situation by introducing the PIE formulation, which allows us to treat boundary conditions in a unified and consistent manner, prior to discretization, in a way, which is supported by theory and does not rely on an ad-hoc modification of either the underlying PDEs or the resulting solution. Apart from yielding solutions to a large class of unsteady PDEs with an unprecedented accuracy, as shown in this work, PIE representation, via an analytical embedment of the boundary constraints into the governing equation dynamics, enables a development of provably optimal approaches for stability analysis and control of PDE systems [55–57], which were previously unattainable. While PIE representation itself is decoupled from discretization [24,58], a discrete approximation of PIEs in a stable and accurate manner is the subject of the current paper. While other choices are possible, we focus on Galerkin methods as they allow us to analytically compute the action of the partial integral operators on the basis functions, thus limiting discretization errors. We are further motivated by developing numerical approximations that are spectrally convergent (in space, for smooth solutions); we choose Chebyshev polynomials of the first kind as suitable candidate basis functions for this purpose, which also possess convenient and fast spectral transform routines [8,47,59].

The objective of this paper is to present a theoretical formulation and a numerical implementation of the PIE-Galerkin-Chebyshev (PGC) methodology for linear PDEs with non-constant coefficients under generalized (non-periodic) boundary conditions. The developed numerical methodology based on the transformation of the PDE to PIE equations is unique and warrants the originality of the presented methodology. All the data obtained with this novel numerical method is new and has not been presented before. The current work aims to answer the following research questions: (1) Can stability and convergence of the PGC methodology be theoretically established and numerically verified? (2) What is the influence of a temporal integration on the overall solution accuracy in the PGC methods? (3) What are the specific advantages, if any, of the developed PGC methodology as applied to a solution of parabolic and hyperbolic PDE systems?

The current paper is organized as follows. In Section 2, we present a general formulation of the PIE framework for linear PDEs with non-constant coefficients in one spatial dimension, where we also extend an original representation in [24] to include inhomogeneous boundary conditions. In Section 3, we introduce a Galerkin approach based on Chebyshev polynomials of the first kind for a solution of the PDE equations in the PIE framework, and present the corresponding stability and convergence proofs for the PIE-Galerkin approach. In Section 4, we show numerical examples and compare our developed PDE-PIE approach with some conventional direct PDE solution methods, followed by discussion and conclusions in Section 5.

2. Partial Integral Equations PIE framework

2.1. Standardized PDE representation

The assumptions of the methodology developed in the current paper are as follows: (1) PDE (or a system of PDEs) has one temporal and one spatial dimension; (2) PDE (or a system of PDEs) is linear with the coefficients that are constant or functions of space; (3) PDE problem is well-posed [1,2].

We now define some notation. We solve a Partial Differential Equation (PDE), or a coupled system of PDEs, on a spatio-temporal domain $(x,t) \in ([a,b] \times \mathbb{R}^+)$. Let $L_2[a,b]^n$ be the space of $\mathbb{R}^n$-valued square-integrable functions in a Lebesgue sense defined on $[a,b]$, equipped with an inner product. We use the notation $H_k[a,b]^n$ to denote a Sobolev subspace of $L_2[a,b]^n$ defined as $\left\{\mathbf{u} \in L_2[a,b]^n : \frac{\partial^q \mathbf{u}}{\partial x^q} \in L_2[a,b]^n, \ \forall\, q \le k\right\}$. C^k will denote the space of functions with k continuous derivatives. $I_n \in \mathbb{R}^{n\times n}$ is used to denote the identity matrix, while 0_n denotes a zero vector of size n. It is implied that, for all the solution states $\mathbf{u}(x,t)$, a partial derivative with respect to time exists for $t \in \mathbb{R}^+$.

We now consider a PDE or a system of PDEs, which satisfies the aforementioned assumptions, in its "state-space" representation [24,60],

$$\begin{bmatrix}\mathbf{u}_0(x,t)\\ \mathbf{u}_1(x,t)\\ \mathbf{u}_2(x,t)\end{bmatrix}_t = A_0(x)\begin{bmatrix}\mathbf{u}_0(x,t)\\ \mathbf{u}_1(x,t)\\ \mathbf{u}_2(x,t)\end{bmatrix} + A_1(x)\begin{bmatrix}\mathbf{u}_1(x,t)\\ \mathbf{u}_2(x,t)\end{bmatrix}_x + A_2(x)\left[\mathbf{u}_2(x,t)\right]_{xx} + \mathbf{f}(x,t), \tag{2.1}$$

with boundary conditions,

$$B\begin{bmatrix}\mathbf{u}_1(a,t)\\ \mathbf{u}_1(b,t)\\ \mathbf{u}_2(a,t)\\ \mathbf{u}_2(b,t)\\ \mathbf{u}_{2x}(a,t)\\ \mathbf{u}_{2x}(b,t)\end{bmatrix} = \mathbf{h}(t) \in C^1(R^+), \tag{2.2}$$

and initial conditions

$$\begin{bmatrix}\mathbf{u}_0(x,0)\\ \mathbf{u}_1(x,0)\\ \mathbf{u}_2(x,0)\end{bmatrix} = \boldsymbol{\beta}^h(x) \in X^h, \tag{2.3}$$

with the space X^h defined below in Eq. (2.5).

Here, $A_0(x) : \mathbb{R} \to \mathbb{R}^{ns\times ns}, A_1(x) : \mathbb{R} \to \mathbb{R}^{ns\times(n_1+n_2)}, A_2(x) : \mathbb{R} \to \mathbb{R}^{ns\times n_2}$ are the bounded matrix-valued real functions. We introduce a functional space X of dimension $ns = n_0 + n_1 + n_2$, such that

$$X := \left\{ \begin{bmatrix} \mathbf{u}_0(x,t) \\ \mathbf{u}_1(x,t) \\ \mathbf{u}_2(x,t) \end{bmatrix} \in \begin{bmatrix} L_2[a,b]^{n_0} \\ H_1[a,b]^{n_1} \\ H_2[a,b]^{n_2} \end{bmatrix}, t \in \mathbb{R}^+ \right\}. \tag{2.4}$$

Furthermore, we denote a subset $X^h \subset X$ that contains functions satisfying the boundary conditions (2.2) as

$$X^h := \left\{ \begin{bmatrix} \mathbf{u}_0(x,t) \\ \mathbf{u}_1(x,t) \\ \mathbf{u}_2(x,t) \end{bmatrix} \in X \cap B \begin{bmatrix} \mathbf{u}_1(a,t) \\ \mathbf{u}_1(b,t) \\ \mathbf{u}_2(a,t) \\ \mathbf{u}_2(b,t) \\ \mathbf{u}_{2x}(a,t) \\ \mathbf{u}_{2x}(b,t) \end{bmatrix} = \mathbf{h}(t),\ t \in \mathbb{R}^+ \right\}. \tag{2.5}$$

We say that a solution,

$$\mathbf{u}^h(x,t) = \begin{bmatrix} \mathbf{u}_0^h(x,t) \\ \mathbf{u}_1^h(x,t) \\ \mathbf{u}_2^h(x,t) \end{bmatrix} \in X^h, \tag{2.6}$$

to Eq. (2.1) with boundary (2.2) and initial (2.3) conditions is in its *primary* state. Here, a superscript h denotes a dependency of the solution on the boundary conditions. Note that, for a well-posed problem, we demand that initial conditions (2.3) satisfy boundary conditions at $t = 0$, i.e. $\boldsymbol{\beta}^h(x) \in X^h\big|_{t=0}$.

To arrive at an Eq. (2.1) for any given linear PDE, a set containing an original scalar-valued dependent variable $v(x,t)$ of a PDE (or a vector-valued dependent variable $\mathbf{v}(x,t)$ for a system of coupled PDEs) and their partial derivatives must be transformed into its corresponding state-space form, where the functions $\mathbf{u}_0(x,t) \in L_2[a,b]^{n_0}$ admit no partial spatial derivatives, the functions $\mathbf{u}_1(x,t) \in H_1[a,b]^{n_1}$ admit only first-order partial spatial derivatives, and the functions $\mathbf{u}_2(x,t) \in H_2[a,b]^{n_2}$ admit up to second-order partial spatial derivatives. Note that the functions $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2\}$ in a state-space form are generally vector-valued, even if the original dependent variable $v(x,t)$ was a scalar [24,60]. Matrix $B \in \mathbb{R}^{n_b\times 2n_b}$ is the boundary conditions matrix, $n_b = n_1 + 2n_2$ is the number of boundary conditions required for a well-posed problem, and $\mathbf{h}(t) \in \mathbb{R}^{n_b}$ is the vector of the boundary condition values. According to a decomposition of the functions into its state-space form, the functions $\mathbf{u}_0(x,t)$ admit no boundary conditions, functions $\mathbf{u}_1(x,t)$ admit one boundary condition per each scalar component, and functions $\mathbf{u}_2(x,t)$ admit two boundary conditions per each scalar component. Since these boundary conditions can be prescribed either on the left or the right ends of the domain or, in general, contain boundary constraints that couple the two ends, a boundary conditions matrix B has $2n_b$ columns. Most 1D PDEs can be formulated using this standardized representation, with multiple examples on how to accomplish this transformation for various linear PDE models given in our previous work [24,60], and in the numerical examples presented in this paper.

2.2. Conversion to a Partial Integral Equation (PIE) representation

2.2.1. Some useful preliminaries

Peet [24] have introduced a framework for converting PDE equations in the form of (2.1) to a Partial Integral Equation (PIE) form. The original formulation is, however, restricted to a homogeneous case, i.e. a zero forcing function $\mathbf{f}(x,t)$, and homogeneous boundary conditions (2.2) given by $\mathbf{h}(t) = 0$. Here, we extend the previous result to inhomogeneous boundary conditions in (2.2) defined by an arbitrary vector $\mathbf{h}(t) \in C^1(\mathbb{R}^+)^{2n_b}$, and an arbitrary forcing function $\mathbf{f}(x,t) \in L_2^{ns}$ in Eq. (2.1). We will try to minimize the repetition of the proofs that already appeared in [24,61], and will refer the reader to these two manuscripts, whenever possible.

For the homogeneous boundary conditions, we have the following lemma [24].

Lemma 2.1. *If $\mathbf{h}(t) = 0$, i.e. boundary conditions are homogeneous, X^0 is a linear subspace of X.*

Proof. We show the following properties of X^0 that makes it a linear subspace:

1. The zero element $0_{ns} \in X^0$, since $0_{ns} \in X$, and 0_{ns} satisfies (2.2) with $\mathbf{h}(t) = 0$.
2. X^0 is closed under addition and scalar multiplication, since X is closed under addition and scalar multiplication, and these operations preserve homogeneous boundary conditions. □

Note that, for inhomogeneous boundary conditions, $\mathbf{h}(t) \neq 0$, X^h is not a linear subspace, since, for one, it does not contain a zero vector. Instead, it corresponds to an affine space isomorphic to X^0 that is obtained from X^0 by a translation transformation, as will be discussed later.

Given a primary state defined by (2.6), we now introduce a *fundamental state* as

$$\mathbf{u}_f(x,t) = \begin{bmatrix} \mathbf{u}_{f0}(x,t) \\ \mathbf{u}_{f1}(x,t) \\ \mathbf{u}_{f2}(x,t) \end{bmatrix} = \begin{bmatrix} \mathbf{u}_0(x,t) \\ \mathbf{u}_{1x}(x,t) \\ \mathbf{u}_{2xx}(x,t) \end{bmatrix} \in \begin{bmatrix} (L_2[a,b])^{n_0} \\ (L_2[a,b])^{n_1} \\ (L_2[a,b])^{n2} \end{bmatrix}, t \in \mathbb{R}^+. \tag{2.7}$$

Note that the fundamental state solution is in $L_2[a,b]^{ns}$ space, and thus, it does not admit boundary constraints, which is reflected in the fact that the superscript h is now omitted from the notation. It can be seen, that the fundamental state is related to the primary state by the following differentiation operation

$$\mathbf{u}_f(x,t) = \mathcal{D}\,\mathbf{u}^h(x,t), \tag{2.8}$$

where the differentiation operator $\mathcal{D}$ has the form

$$\mathcal{D} := \begin{bmatrix} I_{n_0} & & \\ & I_{n_1}\,\partial_x & \\ & & I_{n_2}\,\partial_x^2 \end{bmatrix}. \tag{2.9}$$

Note that, in general, a map $\mathcal{D} : X \to L_2^{ns}$ is non-injective, since there can be multiple elements of X mapped into the same fundamental state $\mathbf{u}_f(x,t)$, differing by boundary conditions.

We now proceed with invoking the following lemma proven in [24].

Lemma 2.2. *Suppose that $u \in H_2[a,b]$. Then for any $x \in [a,b]$,*

$$u(x) = u(a) + \int_a^x u_x(s)ds \tag{2.10}$$

$$u_x(x) = u_x(a) + \int_a^x u_{xx}(s)ds \tag{2.11}$$

$$u(x) = u(a) + u_x(a)(x-a) + \int_a^x (x-s)u_{xx}(s)ds \tag{2.12}$$

Proof. See the manuscript [24] for a proof. □

Next, we define the boundary conditions vectors as

$$\mathbf{u}_{bf}(t) = \begin{bmatrix} \mathbf{u}_1(a,t) \\ \mathbf{u}_1(b,t) \\ \mathbf{u}_2(a,t) \\ \mathbf{u}_2(b,t) \\ \mathbf{u}_{2x}(a,t) \\ \mathbf{u}_{2x}(b,t) \end{bmatrix}, \ \mathbf{u}_{bc}(t) = \begin{bmatrix} \mathbf{u}_1(a,t) \\ \mathbf{u}_2(a,t) \\ \mathbf{u}_{2x}(a,t) \end{bmatrix}, \tag{2.13}$$

where $\mathbf{u}_{bf}(t)$ corresponds to a full set of boundary conditions, and $\mathbf{u}_{bc}(t)$ corresponds to a "core" set of boundary conditions [61]. Note that, under this definition, boundary constraint (2.2) reads as $B\mathbf{u}_{bf}(t) = \mathbf{h}(t)$.

We now have to introduce the notation to define a partial-integral operator of a specific form, which will be referred to as a 3-PI operator.

Definition 1. If $N_0 : [a,b] \to \mathbb{R}^{n\times n}$, $N_1 : [a,b]^2 \to \mathbb{R}^{n\times n}$, $N_2 : [a,b]^2 \to \mathbb{R}^{n\times n}$ are bounded matrix-valued functions, we define a 3-PI operator $\mathcal{P} : L_2^n[a,b] \to L_2^n[a,b]$ as

$$\begin{aligned}(\mathcal{P}\mathbf{u})(x) &:= \left(\mathcal{P}_{\{N_0,N_1,N_2\}}\mathbf{u}\right)(x) := N_0(x)\mathbf{u}(x) \\ &+ \int_a^x N_1(x,s)\mathbf{u}(s)\,ds + \int_a^b N_2(x,s)\mathbf{u}(s)ds,\end{aligned} \tag{2.14}$$

where N_0 defines a multiplier operator and N_1, N_2 define the kernels of the integral operators.

Our definition is slightly different from the one presented in [24] in that a last term here is defined as an integral from a to b, while it is defined as an integral from x to b in [24], however, with the appropriate modification of the integral kernels, the two definitions are equivalent. It is proven in [24] that 3-PI operators are closed under addition, scalar multiplication and composition, and thus form an algebra.

We now define two specific 3-PI operators, which will be instrumental for conversion of the PDEs into the PIE framework, as will be seen below.

$$\begin{aligned}\mathcal{T} &:= \mathcal{P}_{\{G_0,G_1,G_2\}}, \qquad \mathcal{A} := \mathcal{P}_{\{H_0,H_1,H_2\}}, \\ H_0(x) &= A_0(x)G_0 + A_1(x)G_3 + A_{20}(x), \\ H_1(x,s) &= A_0(x)G_1(x,s) + A_1(x)G_4, \\ H_2(x,s) &= A_0(x)G_2(x,s) + A_1(x)G_5(s), \\ A_{20}(x) &= \begin{bmatrix}0 & 0 & A_2(x)\end{bmatrix},\end{aligned} \tag{2.15}$$

where $A_i(x)$, $i = 0 \dots 2$, are as defined in Eq. (2.1), $G_i(x,s), i = 0 \dots 5$, are given in the Appendix A. With this definition, only $G_2(x,s)$ and $H_2(x,s)$ operators depend on the boundary conditions matrix B, and the other operators will stay invariant for a given PDE regardless of the choice of the boundary conditions.

2.2.2. PIE representation

We are now ready to prove the following theorem.

Theorem 2.3. *If the matrix*

$$B_T = BT \tag{2.16}$$

is invertible, where T is given by

$$T := \begin{bmatrix} I_{n_1} & 0 & 0 \\ I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & I_{n_2} & (b-a)I_{n_2} \\ 0 & 0 & I_{n_2} \\ 0 & 0 & I_{n_2} \end{bmatrix}, \tag{2.17}$$

then for any $\mathbf{u}^h(x,t) \in X^h$ there exists a unique fundamental state $\mathbf{u}_f(x,t) \in L_2^{ns}$ given by (2.8), such that $\mathbf{u}^h(x,t)$ can be obtained from $\mathbf{u}_f(x,t)$ by a transformation

$$\mathbf{u}^h(x,t) = K(x)B_T^{-1}\mathbf{h}(t) + \mathcal{T}\mathbf{u}_f(x,t), \tag{2.18}$$

with $\mathcal{T}$ as defined in (2.15), and $K(x)$ given in Appendix A. Furthermore, for any $\mathbf{u}_f(x,t) \in L_2^{ns}$, $\mathbf{u}^h(x,t)$ obtained via (2.18) is in X^h.

Proof. Suppose $\mathbf{u}^h(x,t) \in X^h$. Define the corresponding fundamental state $\mathbf{u}_f(x,t)$ via (2.8). Clearly, $\mathbf{u}_f(x,t) \in L_2^{ns}$. Using Lemma 2.2, we can express $\mathbf{u}_{bf}(t)$ through $\mathbf{u}_{bc}(t)$ (see Eq. (2.13)) and the fundamental state $\mathbf{u}_f(x,t)$ given by (2.8) as

$$\mathbf{u}_{bf}(t) = T\mathbf{u}_{bc}(t) + \mathcal{P}_{\{0,0,Q\}}\mathbf{u}_f(x,t), \tag{2.19}$$

where T is given by (2.17), and Q is defined in Appendix A. Analogously, the primary state $\mathbf{u}^h(x,t)$ can be expressed through $\mathbf{u}_{bc}(t)$ and $\mathbf{u}_f(x,t)$ as

$$\mathbf{u}^h(x,t) = K(x)\mathbf{u}_{bc}(t) + \mathcal{P}_{\{G_0,G_1,0\}}\mathbf{u}_f(x,t), \tag{2.20}$$

where G_0, G_1 are as defined in Appendix A. Using (2.19), the boundary constraint (2.2) can be expressed as

$$B\mathbf{u}_{bf}(t) = B_T\mathbf{u}_{bc}(t) + B\mathcal{P}_{\{0,0,Q\}}\mathbf{u}_f(x,t), \tag{2.21}$$

from where, since $B\mathbf{u}_{bf}(t) = \mathbf{h}(t)$, we have

$$B_T\mathbf{u}_{bc}(t) + B\mathcal{P}_{\{0,0,Q\}}\mathbf{u}_f(x,t) = \mathbf{h}(t). \tag{2.22}$$

Using the assumption of invertibility of B_T, we may now express the core boundary condition vector as

$$\begin{aligned} \mathbf{u}_{bc}(t) &= B_T^{-1}\mathbf{h}(t) - B_T^{-1}B\mathcal{P}_{\{0,0,Q\}}\mathbf{u}_f(x,t) \\ &= B_T^{-1}\mathbf{h}(t) - \mathcal{P}_{\{0,0,B_T^{-1}BQ\}}\mathbf{u}_f(x,t). \end{aligned} \tag{2.23}$$

Substituting (2.23) into (2.20), we get

$$\begin{aligned} &\mathbf{u}^h(x,t) = K(x)B_T^{-1}\mathbf{h}(t) - \mathcal{P}_{\{K,0,0\}}\mathcal{P}_{\{0,0,B_T^{-1}BQ\}}\mathbf{u}_f(x,t) + \\ &\mathcal{P}_{\{G_0,G_1,0\}}\mathbf{u}_f(x,t) = K(x)B_T^{-1}\mathbf{h}(t) + \mathcal{P}_{\{G_0,G_1,G_2\}}\mathbf{u}_f(x,t), \end{aligned} \tag{2.24}$$

which concludes the proof of the first part of the theorem. Note that the addition rule, scalar multiplication rule and the composition rule for the 3-PI operators [24] were used in this proof.

Conversely, let $\mathbf{u}_f(x,t)$ be in L_2^{ns}. It is proven in [61] that $\mathcal{T}\mathbf{u}_f(x,t) \in X^0$. Therefore, $\mathcal{T}\mathbf{u}_f(x,t) \in X$, since $X^0 \subset X$. It is easy to see that $K(x)B_T^{-1}\mathbf{h}(t) \in H_\infty^{ns}$, therefore $K(x)B_T^{-1}\mathbf{h}(t) \in X$, and $\mathbf{u}^h(x,t) \in X$. We now only need to show that $\mathbf{u}^h(x,t)$ satisfies boundary conditions (2.2). We may evaluate the value of components $\mathbf{u}_1^h(x,t)$, $\mathbf{u}_2^h(x,t)$ from (2.18) using the definition of $K(x)$ and $\mathcal{T}$. Correspondingly, we have

$$\mathbf{u}_1^h(x,t) = \begin{bmatrix} I_{n_1} & 0 & 0 \end{bmatrix} B_T^{-1}\mathbf{h}(t) - \begin{bmatrix} 0 & I_{n_1} & 0 \end{bmatrix} \mathcal{P}_{\{0,G_1,G_2\}}\mathbf{u}_f(x,t), \tag{2.25}$$

$$\mathbf{u}_2^h(x,t) = \begin{bmatrix} 0 & I_{n_2} & (x-a)I_{n_2} \end{bmatrix} B_T^{-1}\mathbf{h}(t) - \begin{bmatrix} 0 & 0 & I_{n_2} \end{bmatrix} \mathcal{P}_{\{0,G_1,G_2\}}\mathbf{u}_f(x,t). \tag{2.26}$$

Furthermore, differentiating (2.26) with respect to x, we get

$$\mathbf{u}_{2x}^h(x,t) = \begin{bmatrix} 00 & I_{n_2} \end{bmatrix} B_T^{-1}\mathbf{h}(t) - \frac{\partial}{\partial x}\left(\begin{bmatrix} 0 & 0 & I_{n_2} \end{bmatrix} \mathcal{P}_{\{0,G_1,G_2\}}\mathbf{u}_f(x,t) \right). \tag{2.27}$$

Now, evaluating (2.25), (2.26), (2.27) at $x = a$ nullifies the contribution of $\mathcal{P}_{\{0,G_1,0\}}$ operator and gives us the boundary conditions vector $\mathbf{u}_{bc}(t)$ as

$$\mathbf{u}_{bc}(t) = \begin{bmatrix} \mathbf{u}_1^h(a,t) \\ \mathbf{u}_2^h(a,t) \\ \mathbf{u}_{2x}^h(a,t) \end{bmatrix} = \begin{bmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & I_{n_2} \end{bmatrix} B_T^{-1}\mathbf{h}(t) - B_T^{-1} B \mathcal{P}_{\{0,0,Q\}}\mathbf{u}_f(x,t), \tag{2.28}$$

see also [61]. Now, multiplying both sides of (2.28) by B_T shows that $B_T\mathbf{u}_{bc}(t) + B\mathcal{P}_{\{0,0,Q\}}\mathbf{u}_f(x,t) = \mathbf{h}(t)$, which, by identity (2.21) proves that the primary state $\mathbf{u}^h(x,t)$ constructed via the transformation (2.18) satisfies the boundary conditions. □

We also have the following corollary that further establishes the properties of the transformation (2.18).

Corollary 2.4. *A transformation $L_2^{ns} \to X^h$ defined by Eq. (2.18) is a surjection.*

Proof. Since, by Theorem 2.3, for every $\mathbf{u}^h(x,t) \in X^h$ there exists $\mathbf{u}_f(x,t) \in L_2^{ns}$ that can be mapped into $\mathbf{u}^h(x,t)$, this shows that (2.18) is a surjection. □

Another corollary allows to view the transformation (2.18) as a sequence of a linear and an affine transformation.

Corollary 2.5. *A transformation $L_2^{ns} \to X^h$ defined by Eq. (2.18) consists of a sequence of transformations $L_2^{ns} \underbrace{\to}_{\mathcal{T}} X^0 \underbrace{\to}_{\mathcal{R}} X^h$ where the transformation $\mathcal{T} : L_2^{ns} \to X^0$ is a unitary map, and a transformation $\mathcal{R} : X^0 \to X^h$ is an affine isomorphism defined by a translation.*

Proof. Denote $\mathbf{u}^0(x,t) = \mathcal{T}\mathbf{u}_f(x,t)$. From [24,61], we see that $\mathbf{u}^0(x,t) \in X^0$. Since X^0 is a special case of X^h with $\mathbf{h}(t) = 0$, Corollary 2.4 shows that $\mathcal{T} : L_2^{ns} \to X^0$ is a surjection (an alternative proof can be found in [61]. Since, by Lemma 2.1, X^0 is a linear subspace, an inner product can be defined. Refs. [24,61] further show that $\mathcal{T}$ preserves the inner products, and thus is a unitary map.

Now, we define $R(x,t) = K(x)B_T^{-1}\mathbf{h}(t)$, such that $\mathcal{R} : X^0 \to X^h$ is given by $\mathbf{u}^h(x,t) = \mathbf{u}^0(x,t) + R(x,t)$, which is an affine transformation of translation. Given a specific vector of boundary conditions $\mathbf{h}(t)$ that fixes X^h, a translation function $R(x,t)$ is uniquely defined. We now show that $\mathcal{R}$ is isomorphism. Let $\mathbf{u}^h(x,t)$ be in X^h. Theorem 2.3 shows that $\mathbf{u}^0(x,t) = \mathbf{u}^h(x,t) - R(x,t)$ is in X^0, and thus $\mathcal{R} : X^0 \to X^h$ is a surjection. Now, we have to show that R is also an injection. Suppose there are two elements in X^0, $\mathbf{u}_1^0(x,t)$ and $\mathbf{u}_2^0(x,t)$ that are mapped into a single element $\mathbf{u}^h(x,t)$. We then have $\mathbf{u}_1^0(x,t) = \mathbf{u}^h(x,t) - R(x,t)$, and $\mathbf{u}_2^0(x,t) = \mathbf{u}^h(x,t) - R(x,t)$. Since $R(x,t)$ is a unique function for every X^h, this shows that $\mathbf{u}_1^0(x,t) = \mathbf{u}_2^0(x,t)$, and thus $\mathcal{R}$ is an injection. Hence, $\mathcal{R}$ is an isomorphism, as desired. □

We are now ready to state the final result concerning the conversion of PDEs with inhomogeneous boundary conditions to the PIE framework.

Theorem 2.6. *The function $\mathbf{u}^h(x,t) \in X^h$ satisfies the PDE Eq. (2.1) with boundary conditions (2.2) and initial conditions (2.3) if and only if the corresponding fundamental state function $\mathbf{u}_f(x,t) = \mathcal{D}\mathbf{u}^h(x,t) \in L_2^{ns}$ satisfies the following PIE equation*

$$\mathcal{T}\frac{\partial \mathbf{u}_f(x,t)}{\partial t} = \mathcal{A}\,\mathbf{u}_f(x,t) + \mathbf{g}(x,t), \tag{2.29}$$

with $\mathbf{g}(x,t)$ given by

$$\begin{aligned} \mathbf{g}(x,t) &= A_0(x)K(x)B_T^{-1}\,\mathbf{h}(t) \\ &+ A_1(x)VB_T^{-1}\mathbf{h}(t) - K(x)B_T^{-1}\frac{d\,\mathbf{h}(t)}{d\,t} + \mathbf{f}(x,t), \end{aligned} \tag{2.30}$$

initial conditions $\mathbf{u}_f(x,0) = \boldsymbol{\beta}_f(x)$, where $\boldsymbol{\beta}_f(x) = \mathcal{D}\boldsymbol{\beta}^h(x)$, and the 3-PI operators $\mathcal{T}$, $\mathcal{A}$ as defined by (2.15), $K(x)$ and V as defined in Appendix A. Moreover, $\mathbf{u}^h(x,t)$ is related to $\mathbf{u}_f(x,t)$ by the transformation (2.18).

Proof. Suppose $\mathbf{u}^h(x,t) \in X^h$ satisfies the PDE (2.1) with boundary conditions (2.2) and initial conditions (2.3). Since $\mathbf{u}_f(x,t) = \mathcal{D}\mathbf{u}^h(x,t)$, it immediately follows that $\mathbf{u}_f(x,0) = \mathcal{D}\mathbf{u}^h(x,0)$, i.e. $\boldsymbol{\beta}_f(x) = \mathcal{D}\boldsymbol{\beta}^h(x)$. Using the definition of the PDE (2.1) and defining an auxiliary differentiation operator $\mathcal{D}_1$ as

$$\mathcal{D}_1 := \begin{bmatrix} 0_{n_1\times n_0} & I_{n_1}\partial_x & 0 \\ 0 & 0 & I_{n_2}\,\partial_x \end{bmatrix}, \tag{2.31}$$

we get

$$\begin{aligned} \frac{\partial \mathbf{u}^h(x,t)}{\partial t} &= \mathcal{P}_{\{A_0,0,0\}}\mathbf{u}^h(x,t) + \mathcal{P}_{\{A_1,0,0\}}\,\mathcal{D}_1\,\mathbf{u}^h(x,t) \\ &+ \mathcal{P}_{\{A_{20},0,0\}}\,\mathcal{D}\mathbf{u}^h(x,t) + \mathbf{f}(x,t), \end{aligned} \tag{2.32}$$

with A_{20} defined in (2.15). To evaluate $\mathcal{D}\mathbf{u}^h(x,t)$, Eq. (2.8) can be used, while $\mathcal{D}_1\mathbf{u}^h(x,t)$ can be obtained from

$$\mathcal{D}_1\mathbf{u}^h(x,t) = \mathcal{D}_1\mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t) + \mathcal{D}_1\mathcal{T}\mathbf{u}_f(x,t), \tag{2.33}$$

where the notation $\tilde{K}(x) = K(x)B_T^{-1}$ is used. Substituting (2.18), (2.8) and (2.33) into (2.32), we obtain

$$\begin{aligned}\frac{\partial \mathbf{u}^h(x,t)}{\partial t} &= \mathcal{P}_{\{A_0,0,0\}}\mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t) + \mathcal{P}_{\{A_0,0,0\}}\mathcal{T}\mathbf{u}_f(x,t)\\ &+ \mathcal{P}_{\{A_1,0,0\}}\mathcal{D}_1\mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t) + \mathcal{P}_{\{A_1,0,0\}}\mathcal{D}_1\mathcal{T}\mathbf{u}_f(x,t)\\ &+ \mathcal{P}_{\{A_{20},0,0\}}\mathcal{D}\mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t) + \mathcal{P}_{\{A_{20},0,0\}}\mathcal{D}\mathcal{T}\mathbf{u}_f(x,t) + \mathbf{f}(x,t).\end{aligned} \tag{2.34}$$

Separating homogeneous and inhomogeneous terms in the right-hand side, we have

$$\frac{\partial \mathbf{u}^h(x,t)}{\partial t} = H(x,t) + I(x,t), \tag{2.35}$$

where

$$H(x,t) = \mathcal{P}_{\{A_0,0,0\}}\mathcal{T}\mathbf{u}_f(x,t) + \mathcal{P}_{\{A_1,0,0\}}\mathcal{D}_1\mathcal{T}\mathbf{u}_f(x,t) + \mathcal{P}_{\{A_{20},0,0\}}\mathbf{u}_f(x,t), \tag{2.36}$$

$$I(x,t) = \mathcal{P}_{\{A_0,0,0\}}\mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t) + \mathcal{P}_{\{A_1,0,0\}}\mathcal{D}_1\mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t) + \mathbf{f}(x,t). \tag{2.37}$$

The homogeneous term, as shown in [24], reduces to

$$H(x,t) = \mathcal{P}_{\{H_0,H_1,H_2\}}\mathbf{u}_f(x,t) = \mathcal{A}\,\mathbf{u}_f(x,t). \tag{2.38}$$

Finally, taking a partial derivative with respect to time of Eq. (2.18), we have

$$\frac{\partial \mathbf{u}^h(x,t)}{\partial t} = K(x)B_T^{-1}\frac{d\,\mathbf{h}(t)}{d\,t} + \mathcal{T}\,\frac{\partial \mathbf{u}_f(x,t)}{\partial t}. \tag{2.39}$$

Combining Eqs. (2.35)–(2.39) leads to (2.29)–(2.30).

Conversely, suppose $\mathbf{u}_f(x,t) \in L_2^{ns}$ satisfies the PIE Eq. (2.29)–(2.30) with initial conditions $\mathbf{u}_f(x,0) = \boldsymbol{\beta}_f(x)$. Define $\mathbf{u}^h(x,t)$ according to the transformation (2.18). By Theorem 2.3, $\mathbf{u}^h(x,t) \in X^h$, and thus satisfies boundary conditions (2.2). Rearrange the PIE equation as

$$\mathcal{T}\,\frac{\partial \mathbf{u}_f(x,t)}{\partial t} + K(x)B_T^{-1}\frac{d\,\mathbf{h}(t)}{d\,t} = \mathcal{A}\,\mathbf{u}_f(x,t) + I(x,t), \tag{2.40}$$

with $I(x,t)$ as defined in (2.37). The left-hand side of (2.40) is equal to $\partial\,\mathbf{u}^h(x,t)/\partial\,t$, according to (2.39). Recognizing that, by (2.38), $\mathcal{A}\,\mathbf{u}_f(x,t) = H(x,t)$, and using Eqs. (2.36) and (2.37), the right-hand side of (2.40) becomes

$$\begin{aligned}H(x,t) + I(x,t) &= \mathcal{P}_{\{A_0,0,0\}}\left(\mathcal{T}\mathbf{u}_f(x,t) + \mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t)\right)\\ &+ \mathcal{P}_{\{A_1,0,0\}}\left(\mathcal{D}_1\mathcal{T}\mathbf{u}_f(x,t) + \mathcal{D}_1\mathcal{P}_{\{\tilde{K},0,0\}}\mathbf{h}(t)\right)\\ &+ \mathcal{P}_{\{A_{20},0,0\}}\mathbf{u}_f(x,t) + \mathbf{f}(x,t).\end{aligned} \tag{2.41}$$

Using (2.8), (2.18) and (2.33), the right-hand side of (2.41) reduces to

$$\begin{aligned}&H(x,t) + I(x,t)\\ &= \mathcal{P}_{\{A_0,0,0\}}\mathbf{u}^h(x,t) + \mathcal{P}_{\{A_1,0,0\}}\,\mathcal{D}_1\,\mathbf{u}^h(x,t) + \mathcal{P}_{\{A_{20},0,0\}}\,\mathcal{D}\,\mathbf{u}^h(x,t) + \mathbf{f}(x,t),\end{aligned} \tag{2.42}$$

which is equivalent to the right-hand side of the PDE Eq. (2.1), showing that $\mathbf{u}^h(x,t)$ indeed satisfies the original PDE. □

2.2.3. Note on invertibility of B_T

Theorem 2.3 relies on the condition of invertibility of the B_T matrix. It was proven in [24] that the necessary and sufficient condition for the inverse of B_T to exist is for B to: (1) have a row rank of n_b, and (2) have a row space that has a trivial intersection with the row space of $T^\perp$, where $T^\perp$ defines an orthogonal complement to a column space of T. This leads to an exclusion of the boundary conditions that are a linear combination of

$$\mathbf{u}_1(a,t) - \mathbf{u}_1(b,t) = \mathbf{h}_1(t), \tag{2.43}$$

$$\mathbf{u}_2(a,t) + (b-a)\mathbf{u}_{2x}(a,t) - \mathbf{u}_2(b,t) = \mathbf{h}_2^{(1)}(t), \tag{2.44}$$

$$\mathbf{u}_{2x}(a,t) - \mathbf{u}_{2x}(b,t) = \mathbf{h}_2^{(2)}(t), \tag{2.45}$$

from the set of the boundary conditions, for which B_T is invertible. Note that the excluded boundary conditions involve periodic boundary conditions on the state $\mathbf{u}_1(x,t)$, periodic boundary conditions on the derivatives of the state $\mathbf{u}_2(x,t)$, and Neumann–Neumann conditions on the state $\mathbf{u}_2(x,t)$, among others. In general, such boundary conditions are ill-posed for the boundary value problems, however, a unique solution might exist to initial–boundary value problems [62]. In a PIE framework, the problem with B_T invertibility for these boundary conditions arises from the fact that now a fundamental state needs to have an additional constraint

in order to satisfy these boundary conditions, implying that the fundamental state is no longer minimal. For example, with the periodic boundary condition on $\mathbf{u}_1$, we have a constraint that the integral of its derivative over the domain must be equal to zero. If this derivative enters the fundamental state, as would be the case for $\mathbf{u}_{1x}$, this additional constraint, since it is not embedded into the PIE dynamics, may not be satisfied.

To remedy this situation, it is possible to redefine a fundamental state to be free of constraints, and embed the corresponding constraints into the PIE operators. This can be formally accomplished by performing an SVD decomposition of the B_T matrix, introducing an auxiliary state vector $\mathbf{u}_n(t) \in \mathbb{R}^r$, where r is the rank deficiency of B_T, and modifying the PIE equations accordingly [63]. While this is generally possible, such modification will not be considered here, and we will assume that B_T matrix is invertible, with the use of appropriate boundary conditions.

3. Solution of the PDEs in the PIE framework: PIE-Galerkin- Chebyshev (PGC) approximation

3.1. Spatial treatment

We are now interested in finding a solution $\mathbf{u}_f(x,t) \in L_2[a,b]^{ns}$ to the PIE equation (2.29) with the initial conditions $\mathbf{u}_f(x,0) = \boldsymbol{\beta}_f(x)$, whose corresponding primary state $\mathbf{u}^h(x,t)$ given by (2.18) satisfies the original PDE equation (2.1), according to Theorem 2.6. Since $\mathbf{u}_f(x,t) \in L_2[a,b]^{ns}$, we are free to choose any approximation space without needing to worry about satisfying boundary conditions. We choose Chebyshev polynomials of the first kind as the approximation functions [8,59]. Since Chebyshev polynomials are defined on the $[-1,1]$ domain, we need to map our original PDE from $x = [a,b]$ onto a computational domain $x^{(c)} = [-1,1]$, which can be readily accomplished by a linear transformation $x^{(c)} = \frac{2x-(b+a)}{b-a}$, with the inverse map $x = \frac{b-a}{2}x^{(c)} + \frac{b+a}{2}$. With a slight abuse of notation, in what follows, we will assume that the corresponding PIE equation is defined on $x \in [-1,1]$ domain, following a prior mapping if necessary.

In accordance with (2.7), (2.8), and (2.9), we can write for each sub-component $\mathbf{u}_{fp}(x,t)$ of $\mathbf{u}_f(x,t)$, $p = 0,1,2$,

$$\mathbf{u}_{fp}(x,t) = \frac{\partial^p \mathbf{u}_p(x,t)}{\partial x^p}. \tag{3.1}$$

Therefore, with each component $u_{fi}(x,t)$, $i = 1 \dots ns$, of the vector $\mathbf{u}_f(x,t)$, we can associate an index

$$p = p(i), \tag{3.2}$$

defined as a "minimum smoothness" required from the original $u_i(x,t)$ function to enter the PDE (2.1). We now look for solutions $u_{fi}(x,t) \in \mathbb{P}[-1,1]^{N-p(i)}$ for each corresponding $u_{fi}(x,t)$ component, where $\mathbb{P}[-1,1]^{N-p(i)}$ is the space of all polynomial functions of degree $N - p(i)$ or less on the domain $[-1,1]$, i.e. we approximate

$$\hat{u}_{fi}(x,t) = \sum_{k=0}^{N-p(i)} a_{ik}(t) T_k(x), \tag{3.3}$$

where $T_k(x)$ are the Chebyshev polynomials of the first kind [8,59], and $a_{ik}(t) \in C^1(R^+)$ are the corresponding time-dependent Chebyshev coefficients, where the subscript i denotes their affiliation with a particular solution component $\hat{u}_{fi}(x,t)$. The approximation for the vector-valued function $\hat{\mathbf{u}}_f(x,t)$ can then be compactly written as

$$\hat{\mathbf{u}}_f(x,t) = \sum_{i=1}^{ns} \sum_{k=0}^{N-p(i)} a_{ik}(t) \boldsymbol{\phi}_{ik}(x), \tag{3.4}$$

where the vector-valued Chebyshev basis functions $\boldsymbol{\phi}_{ik}(x) : \mathbb{R} \to \mathbb{R}^{ns}$ can be defined as

$$\boldsymbol{\phi}_{ik}(x) = \underbrace{\begin{bmatrix} 0 & \cdots & \cdots & T_k(x) & \cdots & 0 \end{bmatrix}^T}_{ns}, \tag{3.5}$$

where $T_k(x)$ is in the ith position of the vector $\boldsymbol{\phi}_{ik}(x)$ for each vector-valued basis function distinguished by a fixed $i = 1 \dots ns$, $k = 0 \dots N - p(i)$. We denote the polynomial space spanned by the vector-valued basis functions $\boldsymbol{\phi}_{ik}(x)$ as $Y^{N_p} := \mathbb{P}[-1,1]^{N_p}$, where $N_p = n_0 N \times n_1(N-1) \times n_2(N-2)$, so that the composite vector-valued approximation $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$.

We introduce the same approximation for the lumped inhomogeneous term $\mathbf{g}(x,t)$, see (2.30), i.e. we write

$$\hat{\mathbf{g}}(x,t) = \sum_{i=1}^{ns} \sum_{k=0}^{N-p(i)} b_{ik}(t) \boldsymbol{\phi}_{ik}(x), \tag{3.6}$$

where $b_{ik}(t)$ are the corresponding Chebyshev coefficients associated with the inhomogeneous term, $\hat{\mathbf{g}}(x,t) \in Y^{N_p}$.

With the expansion (3.4), the action of the 3-PI operator $\mathcal{T}$ on the function approximation $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$ can be written as

$$\mathcal{T}\hat{\mathbf{u}}_f(x,t) = \sum_{i=1}^{ns} \sum_{k=0}^{N-p(i)} a_{ik}(t) \mathcal{T}\boldsymbol{\phi}_{ik}(x) = \sum_{i=1}^{ns} \sum_{k=0}^{N-p(i)} a_{ik}(t) \,\mathrm{Col}_i(\mathcal{T}) T_k(x), \tag{3.7}$$

where the notation $\mathrm{Col}_i(\mathcal{T})$ stands for the ith column of the matrix operator $\mathcal{T}$. We now have the following lemma.

Lemma 3.1. *The product $\mathcal{T}_{mn}T_k(x)$, where $\mathcal{T}_{mn}$ is an element of the matrix operator $\mathcal{T}$, $T_k(x)$ is a Chebyshev polynomial function, can be evaluated according to the following rules:*

1. *For $m \leq n_0$,*

$$\mathcal{T}_{mn}T_k(x) = \delta_{mn}T_k(x). \tag{3.8}$$

2. *For $n_0 < m \leq n_0 + n_1$,*

$$\mathcal{T}_{mn}T_k(x) = b^{(1)}_{0kmn}T_0(x) + b^{(1)}_{1kmn}T_1(x) + \delta_{mn}\left(c^-_{k-1}T_{k-1}(x) + c^+_{k+1}T_{k+1}(x)\right), \tag{3.9}$$

where

$$c^-_k = \begin{cases} 0, & k \leq 1 \\ -\frac{1}{2k}, & k \geq 2 \end{cases} \qquad c^+_k = \begin{cases} 0, & k \leq 1 \\ \frac{1}{2k}, & k \geq 2 \end{cases} \tag{3.10}$$

3. *For $m > n_0 + n_1$,*

$$\begin{aligned}\mathcal{T}_{mn}T_k(x) &= b^{(2)}_{0kmn}T_0(x) + b^{(2)}_{1kmn}T_1(x) \\ &\quad + \delta_{mn}\left(d^-_{k-2}T_{k-2}(x) + d_kT_k(x) + d^+_{k+2}T_{k+2}(x)\right),\end{aligned} \tag{3.11}$$

where

$$d^-_k = \begin{cases} 0, & k \leq 1 \\ \frac{1}{4k(k+1)}, & k \geq 2 \end{cases} \qquad d^+_k = \begin{cases} 0, & k \leq 1 \\ \frac{1}{2k(k-1)}, & k = 2 \\ \frac{1}{4k(k-1)}, & k \geq 3 \end{cases} \tag{3.12}$$

$$d_k = \begin{cases} 0, & k \leq 1 \\ -\frac{1}{2(k^2-1)}, & k \geq 2, \end{cases}$$

where δ_{mn} is a Kronecker delta function, $b^{(i)}_{jkmn}$, $i = 1, 2$, $j = 0, 1$ are real constants, generally dependent on the boundary conditions, and $c^-_k, c^+_k, d^-_k, d_k, d^+_k$ are real constants independent of the boundary conditions. Constants that depend on the boundary conditions can be found, given a particular PIE operator, following the polynomial integration rules established in the proof.

Proof. The proof of this lemma is included in the Appendix B. □

As a consequence of this result, it can be concluded that the action of $\mathcal{T}$ on functions that belong to the polynomial subspaces, keeps them in the polynomial subspaces, which allows us to evaluate the action of a partial-integral operator $\mathcal{T}$ on the polynomial functions analytically, using the formulas presented in Lemma 3.1. In fact, we can now prove the following lemma.

Lemma 3.2. *If $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$, $N_p = n_0N \times n_1(N-1) \times n_2(N-2)$, $t \in \mathbb{R}^+$, $N \geq 2$, the corresponding function approximation $\hat{\mathbf{u}}^h(x,t)$ to the primary solution*

$$\hat{\mathbf{u}}^h(x,t) = K(x)B_T^{-1}\mathbf{h}(t) + \mathcal{T}\hat{\mathbf{u}}_f(x,t) \tag{3.13}$$

is in the space $\mathbb{P}^{Nns}, t \in \mathbb{R}^+$, i.e. all the components of the primary vector-valued solution are in $\mathbb{P}^N$. Furthermore, for $\hat{\mathbf{u}}^h(x,t) \in \mathbb{P}^{Nns}$, the corresponding fundamental state approximation

$$\hat{\mathbf{u}}_f(x,t) = \mathcal{D}\hat{\mathbf{u}}^h(x,t) \tag{3.14}$$

is in Y^{N_p}.

Proof. Suppose $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$. We first note that $K(x)B_T^{-1}\mathbf{h}(t) \in \mathbb{P}^{1\cdot ns}$, which, for $\mathcal{T}\hat{\mathbf{u}}_f(x,t) \in \mathbb{P}^{Nns}$, keeps the composite function in $\mathbb{P}^{Nns}$. We now proceed to show that $\mathcal{T}\hat{\mathbf{u}}_f(x,t) \in \mathbb{P}^{Nns}$. Denote

$$\hat{\mathbf{u}}_f(x,t) = \begin{bmatrix} \hat{\mathbf{u}}_{f0}(x,t) \\ \hat{\mathbf{u}}_{f1}(x,t) \\ \hat{\mathbf{u}}_{f2}(x,t) \end{bmatrix}, \hat{\mathbf{u}}^0(x,t) = \mathcal{T}\hat{\mathbf{u}}_f(x,t) = \begin{bmatrix} \hat{\mathbf{u}}^0_0(x,t) \\ \hat{\mathbf{u}}^0_1(x,t) \\ \hat{\mathbf{u}}^0_2(x,t) \end{bmatrix}, \tag{3.15}$$

where $\hat{\mathbf{u}}_{fp}(x,t)$ is the polynomial approximation of $\mathbf{u}_{fp}(x,t) \in L_2^{n_p}$, and $\hat{\mathbf{u}}^0_p(x,t)$ is the polynomial approximation of $\mathbf{u}^0_p(x,t)$, respectively, $p = 0, 1, 2$, $\mathbf{u}^0(x,t) \in X^0$. Noting the structure of the matrix functions G_0, G_1 and G_2, it is easily seen that

$$\begin{bmatrix} \hat{\mathbf{u}}^0_0(x,t) \\ \hat{\mathbf{u}}^0_1(x,t) \\ \hat{\mathbf{u}}^0_2(x,t) \end{bmatrix} = \mathcal{P}_{\{G_0,0,0\}}\begin{bmatrix} \hat{\mathbf{u}}_{f0}(x,t) \\ 0 \\ 0 \end{bmatrix} + \mathcal{P}_{\{0,G_1,0\}}\begin{bmatrix} 0 \\ \hat{\mathbf{u}}_{f1}(x,t) \\ \hat{\mathbf{u}}_{f2}(x,t) \end{bmatrix} + \mathcal{P}_{\{0,0,G_2\}}\begin{bmatrix} 0 \\ \hat{\mathbf{u}}_{f1}(x,t) \\ \hat{\mathbf{u}}_{f2}(x,t) \end{bmatrix}. \tag{3.16}$$

The first term in the right-hand side of Eq. (3.16) shows that the first n_0 components of $\hat{\mathbf{u}}_f(x,t)$ are mapped into the first n_0 components of $\hat{\mathbf{u}}^0(x,t)$, with the corresponding $\mathbb{P}^N \to \mathbb{P}^N$ mapping according to (3.8). Since the matrix G_1 is block-diagonal, and according to (3.9), (3.11), the second term of (3.16) maps the second group of n_1 components between the vectors $\hat{\mathbf{u}}_f(x,t)$ and $\hat{\mathbf{u}}^0(x,t)$ as $\mathbb{P}^{N-1} \to \mathbb{P}^N$, and the third group of n_2 components as $\mathbb{P}^{N-2} \to \mathbb{P}^N$. The last entry of Eq. (3.16) corresponds to an integral over an entire domain, and thus, as shown in the proof of Lemma 3.1, produces only the outputs in $\mathbb{P}^0$ or $\mathbb{P}^1$.

Now, let $\hat{\mathbf{u}}^h(x,t)$ be in $\mathbb{P}^{Nns}$. According to the structure of the differentiation operator $\mathcal{D}$, see Eq. (2.9), it is easy to see that $\mathcal{D}\hat{\mathbf{u}}^h(x,t) \in Y^{n_0 N \times n_1(N-1) \times n_2(N-2)}$, which concludes the proof. □

Define the polynomial space $\mathbb{P}^h$ as the space of functions from $\mathbb{P}^{Nns}$ that satisfy the boundary conditions on $[-1,1]$ domain, i.e.

$$\mathbb{P}^h := \left\{ \begin{bmatrix} \hat{\mathbf{u}}_0(x,t) \\ \hat{\mathbf{u}}_1(x,t) \\ \hat{\mathbf{u}}_2(x,t) \end{bmatrix} \in \mathbb{P}^{Nns} \cap B \begin{bmatrix} \hat{\mathbf{u}}_1(-1,t) \\ \hat{\mathbf{u}}_1(1,t) \\ \hat{\mathbf{u}}_2(-1,t) \\ \hat{\mathbf{u}}_2(1,t) \\ \hat{\mathbf{u}}_{2x}(-1,t) \\ \hat{\mathbf{u}}_{2x}(1,t) \end{bmatrix} = \mathbf{h}(t),\ t \in \mathbb{R}^+ \right\} \tag{3.17}$$

The following important theorem allows us to establish the approximation properties of the primary solution $\hat{\mathbf{u}}^h(x,t)$ of the PDE (2.1), given by (3.13).

Theorem 3.3. *For every $\hat{\mathbf{u}}^h(x,t) \in \mathbb{P}^h$, with $N \geq 2$, there exists a corresponding approximation to a fundamental state $\hat{\mathbf{u}}_f(x,t) = \mathcal{D}\hat{\mathbf{u}}^h(x,t)$, $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$, $N_p = n_0 N \times n_1(N-1) \times n_2(N-2), t \in \mathbb{R}^+$, that is mapped into $\hat{\mathbf{u}}^h(x,t)$ according to the transformation (3.13). Moreover, for every $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$, $\hat{\mathbf{u}}^h(x,t)$ defined by (3.13) is in $\mathbb{P}^h$.*

Proof. Let $\hat{\mathbf{u}}^h(x,t) \in \mathbb{P}^h$. Therefore, $\hat{\mathbf{u}}^h(x,t) \in \mathbb{P}^{Nns}$. Suppose $\hat{\mathbf{u}}_f(x,t)$ satisfies Eq. (3.14). By Lemma 3.2, $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$. Moreover, due to Theorem 2.3, we have that, since $\mathbb{P}^h \subset X^h$, and $Y^{N_p} \subset L_2^{ns}$, $\hat{\mathbf{u}}_f(x,t)$ defined by (3.14) is mapped into $\hat{\mathbf{u}}^h(x,t)$ according to the transformation (2.18), which is equivalent to (3.13).

Now, consider any $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$. Again, by Lemma 3.2, $\hat{\mathbf{u}}^h(x,t)$ defined by the transformation (3.13) is in $\mathbb{P}^{Nns}$. We are left to prove that $\hat{\mathbf{u}}^h(x,t)$ satisfies the boundary conditions at $a = -1, b = 1$. Since $\hat{\mathbf{u}}_f(x,t) \in L_2^{ns}[-1,1]$, Theorem 2.3 ensures that $\hat{\mathbf{u}}^h(x,t)$ obtained via (3.13), which is equivalent to (2.18), is in $X^h[-1,1]$, i.e. satisfies the aforementioned boundary conditions, which concludes the proof. □

According to Theorem 3.3, $\hat{\mathbf{u}}^h(x,t)$ can be decomposed into a corresponding polynomial approximation as

$$\hat{\mathbf{u}}^h(x,t) = \sum_{i=1}^{ns} \sum_{k=0}^{N} a_{ik}^h(t) \boldsymbol{\psi}_{ik}(x), \tag{3.18}$$

where $a_{ik}^h(t)$ are the Chebyshev coefficients, and $\boldsymbol{\psi}_{ik}(x) \in \mathbb{P}^{Nns}$ are the basis functions defined as

$$\boldsymbol{\psi}_{ik}(x) = \underbrace{\begin{bmatrix} 0 \cdots & \cdots & T_k(x) & \cdots & 0 \end{bmatrix}^T}_{ns}, \tag{3.19}$$

where $T_k(x)$ is in the ith position of the vector $\boldsymbol{\psi}_{ik}(x)$, $i = 1 \ldots ns$, $k = 0 \ldots N$.

We note that the property given by Theorem 3.3 could be established due to the fact that the 3-PI operator $\mathcal{T}$ is invariant under a projection onto the polynomial subspace $\mathbb{P}^N$. Such invariance would not necessarily hold true for another choice of an approximation space (which does not contain a polynomial basis).

To represent the operator $\mathcal{A} = \mathcal{P}_{\{H_0,H_1,H_2\}}$ in the right-hand side of Eq. (2.29), which contains the functions $A_0(x)$, $A_1(x)$, and $A_2(x)$ in the Galerkin–Chebyshev approximation framework, we decompose the functions $A_j(x)$, $j = 0,1,2$, into the Chebyshev series as

$$A_j(x) = \sum_{m=0}^{\infty} A_{jm} T_m(x), \tag{3.20}$$

where A_{jm} are the matrix-valued coefficients for a particular function $A_j(x)$. Correspondingly, the kernel functions H_j, $j = 0,1,2$, in $\mathcal{P}_{\{H_0,H_1,H_2\}}$ can be decomposed into the matrix-valued Chebyshev expansion series as

$$H_0(x) = \sum_{m=0}^{\infty} H_{jm} T_m(x), \tag{3.21}$$

$$H_j(x,s) = \sum_{m=0}^{\infty} \sum_{i=0}^{1} A_{im} T_m(x) G_{j+3i}(x,s), \quad j = 1,2. \tag{3.22}$$

To apply the operator $\mathcal{A} = \mathcal{P}_{\{H_0,H_1,H_2\}}$ to $\hat{\mathbf{u}}_f(x,t)$ given by (3.4), we first note that

$$H_0(x)T_k(x) = \sum_{m=0}^{\infty} H_{0m} T_m(x) T_k(x) = \sum_{m=0}^{\infty} \frac{1}{2} H_{0m} \left(T_{m+k}(x) + T_{|m-k|}(x) \right). \tag{3.23}$$

For the integral kernels, we note that

$$\int H_j(x,s)\,T_k(s)\,ds = \sum_{m=0}^{\infty}\sum_{i=0}^{1} A_{im}T_m(x)\int G_{j+3i}(s)T_k(s)\,ds, \tag{3.24}$$

where $j = 1,2$, upon which the integrals in the right-hand side of Eq. (3.24) can be computed using the integration rules for the Chebyshev polynomials [8,64] as outlined in Appendix B.

We proceed with applying a method of weighted residuals to Eq. (2.29), i.e., we introduce a space of test functions $\hat{\mathbf{v}}(x) \in Z^{N_p}$, and search for $\hat{\mathbf{u}}_f(x,t) \in Y^{N_p}$, $t \in R^+$, such that

$$\left(\mathcal{T}\,\frac{\partial \hat{\mathbf{u}}_f(x,t)}{\partial t}, \hat{\mathbf{v}}(x)\right) = \left(\mathcal{A}\,\hat{\mathbf{u}}_f(x,t) + \hat{\mathbf{g}}(x,t), \hat{\mathbf{v}}(x)\right),\ \forall \hat{\mathbf{v}}(x) \in Z^{N_p}, \tag{3.25}$$

with $(\hat{\mathbf{u}}(x,t), \hat{\mathbf{v}}(x))$, $t \in \mathbb{R}^+$, denoting an inner product on a Hilbert space defined as

$$(\hat{\mathbf{u}}(x,t), \hat{\mathbf{v}}(x)) = \int_{-1}^{1} \hat{\mathbf{u}}^T(x,t)\hat{\mathbf{v}}(x)w(x)dx,\ w(x) = \frac{1}{\sqrt{1-x^2}}, \tag{3.26}$$

where $w(x)$ is the weight function [8,59]. Following Galerkin approach, we set $Z^{N_p} = Y^{N_p}$. Evaluating an inner product in (3.25) for each $\hat{\mathbf{v}}(x) = \boldsymbol{\phi}_{mn}(x) \in Y^{N_p}$, $m = 1 \dots ns$, $n = 0 \dots N - p(m)$, and using the orthogonality of the Chebyshev polynomials with respect to this weight function [8,59], a set of N_d linear ordinary differential equations (ODEs) is obtained for N_d unknown Chebyshev coefficients $a_{ik}(t)$ in (3.4), $N_d = n_0(N+1) \times n_1 N \times n_2(N-1)$, which can be written in a matrix form as

$$M\,\frac{d\,\mathbf{a}(t)}{d\,t} = A\,\mathbf{a}(t) + \mathbf{b}(t), \tag{3.27}$$

with initial conditions $\mathbf{a}(0) = \boldsymbol{a}_0$ consisting of Chebyshev coefficients of $\hat{\mathbf{u}}_f(x,0)$. Here, $\mathbf{a}(t) \in \mathbb{R}^{N_d}$ is the vector of the Chebyshev expansion coefficients of the unknown function $\hat{\mathbf{u}}_f(x,t)$ via (3.4), and $\mathbf{b}(t) \in \mathbb{R}^{N_d}$ is the vector of known Chebyshev coefficients coming from the series expansion of the lumped inhomogeneous term (2.30) via (3.6). To form the $\mathbf{a}(t)$ and $\mathbf{b}(t)$ vectors, we stack $N - p(i)$ Chebyshev coefficients $a_{ik}(t)$, $b_{ik}(t)$, corresponding to each component i, prior to proceeding to the next component, i.e. the entries $a_j(t)$, $b_j(t)$ of $\mathbf{a}(t)$, $\mathbf{b}(t)$ can be expressed as $a_{(i-1)ns+k+1}(t) = a_{ik}(t)$, $i = 1 \dots ns, k = 0 \dots N - p(i)$, same for $b_j(t)$. Matrices $M \in \mathbb{R}^{N_d \times N_d}$, $A \in \mathbb{R}^{N_d \times N_d}$ are the matrices consisting of the entries of the discretized $\mathcal{T}$ and $\mathcal{A}$ operators, respectively, multiplying the corresponding components of the $\mathbf{a}(t)$ vector.

To recover the primary solution $\hat{\mathbf{u}}^h(x,t)$ approximated as (3.18), we take an inner product of (3.13) with each of the basis function $\boldsymbol{\psi}_{ik}(x) \in \mathbb{P}^{Nns}$ to yield

$$\mathbf{a}^h(t) = \left(Z_1 K(0) + Z_2(K(1) - 1)\right) B_T^{-1}\mathbf{h}(t) + \tilde{M}\,\mathbf{a}(t), \tag{3.28}$$

where $\mathbf{a}^h(t) \in \mathbb{R}^{Nns}$, $Z_k \in \mathbb{R}^{(N+1)ns \times ns}$, $k = 1,2$ are zero matrices with one in the positions $\{k + (l-1)(N+1) \times l\}$, $l = 1 \dots ns$, and the matrix $\tilde{M} \in \mathbb{R}^{(N+1)ns \times N_d}$ is the corresponding non-square discrete representation of the operator $\mathcal{T}$.

3.2. Stability and convergence of a semi-discrete approximation

This section concerns the stability and convergence estimates of a semi-discrete PGC formulation, namely, when a temporal variable is not discretized. For the sake of brevity, we will consider the scalar case, while extension to the vector-valued case is straightforward. Since Eq. (2.29) can represent both parabolic and hyperbolic systems, we consider the most conservative situation and, instead of assuming coercivity [8,65], we assume a weaker non-positivity property [8] associated with the integral operators $\mathcal{A}, \mathcal{T}$ as

$$(\mathcal{A}\,u_f, \mathcal{T}\,u_f) \le 0 \text{ for all } u_f \in L_2[-1,1], \tag{3.29}$$

with the inner product defined as in (3.26), and its discrete counterpart

$$(\mathcal{A}\,\hat{u}_f, \mathcal{T}\hat{u}_f)_N \le 0 \text{ for all } \hat{u}_f \in \mathbb{P}[-1,1]^N \text{ and for all } N > 0, \tag{3.30}$$

where the inner product in the left-hand side of (3.30) is defined as

$(\mathcal{A}\,\hat{u}_f, \mathcal{T}\hat{u}_f)_N = (R_N(\mathcal{A}\,\hat{u}_f), R_N(\mathcal{T}\hat{u}_f))$, with $R_N : L_2 \to \mathbb{P}^N$ being a projection operator. The following theorem concerns the stability of Galerkin approximation of the PIE equation (2.29).

Theorem 3.4. *Denote $\widehat{\mathcal{T}u_f} = R_{N-p}(\mathcal{T}u_f)$, where $p = 0, 1$ or 2 as defined in (3.2). Under the assumption (3.30), the following inequality holds*

$$\|\widehat{\mathcal{T}\hat{u}_f(t)}\|^2 \le C(t)\left(\|\widehat{\mathcal{T}\hat{u}_f(0)}\|^2 + \int_0^t \|\hat{g}(s)\|^2\,ds\right) \text{ for all } t \ge 0, \tag{3.31}$$

with the constant $C(t)$ independent of N, which yields stability of approximation (3.25).

Proof. Estimate (3.31) is readily obtained from (3.25) by using $\hat{v} = \widehat{\mathcal{T}\hat{u}_f(t)}$ as a test function, assumption (3.30), Cauchy–Schwarz inequality to bound the inner product $(\hat{g}, \widehat{\mathcal{T}\hat{u}_f}) \le \|\hat{g}\|\,\|\widehat{\mathcal{T}\hat{u}_f}\|$, algebraic inequality $ab \le 1/(4\epsilon)\,a^2 + \epsilon\,b^2$ with $\epsilon = 1/2$, and, subsequently, invoking Gronwall's lemma [8,66,67], yielding $C(t) = \exp(t)$. □

The following theorem establishes the convergence properties of the PGC methodology.

Theorem 3.5. *If (3.30) is satisfied, the following convergence estimate holds*

$$\|u^h(t) - \widehat{\hat{u}^h(t)}\| \leq C(N-p)^{p-m}\Bigg\{ \|u_f(t)\| + exp\left(\frac{t}{2}\right)\left(\int_0^t \left(\|\dot{u}_f(s)\|^2 + \|u_f(s)\|^2 + \|g(s)\|^2\right) ds\right)^{1/2}\Bigg\} \text{ for all } t \geq 0, \tag{3.32}$$

where p is a minimum smoothness of the primary solution as in Theorem 3.4, m is the actual number of square-integrable spatial derivatives of the primary solution, and a dot symbol denotes a partial derivative with respect to time.

Proof. From (2.18), (3.13), we have $\|u^h(t) - \widehat{\hat{u}^h(t)}\| = \|\mathcal{T}u_f(t) - \widehat{\mathcal{T}\hat{u}_f(t)}\|$. To obtain a convergence estimate, we define an error function $e(x,t) = R_{N-p}u_f(x,t) - \hat{u}_f(x,t)$. Taking an inner product of (2.29) with $\widehat{\mathcal{T}e}$, substituting $\widehat{\mathcal{T}e}$ as a test function in (3.25), and a subsequent algebraic manipulation, the following evolution equation for the error can be obtained:

$$\frac{1}{2}\frac{d}{dt}\left(\widehat{\mathcal{T}e(x,t)}, \widehat{\mathcal{T}e(x,t)}\right) = \left(\widehat{\mathcal{A}e(x,t)}, \widehat{\mathcal{T}e(x,t)}\right) + \left(R(x,t), \widehat{\mathcal{T}e(x,t)}\right), \tag{3.33}$$

where the residual term $R(x,t)$ is given by

$$\begin{aligned} R(x,t) = &-\left(\mathcal{T}\dot{u}_f(x,t) - \widehat{\mathcal{T}\dot{u}_f(x,t)}\right) + \left(\mathcal{A}u_f(x,t) - \widehat{\mathcal{A}u_f(x,t)}\right) + (g(x,t) - \hat{g}(x,t)) \\ &- \left(\widehat{\mathcal{T}\dot{u}_f(x,t)} - \mathcal{T}R_{N-p}\dot{u}_f(x,t)\right) + \left(\widehat{\mathcal{A}u_f(x,t)} - \mathcal{A}R_{N-p}u_f(x,t)\right), \end{aligned} \tag{3.34}$$

where the last two terms in (3.34) are the errors due to non-commutativity of the integration and projection operators. Applying assumption (3.30) to the first term in the right-hand side of (3.33), bounding the inner product $\left(R(x,t), \widehat{\mathcal{T}e(x,t)}\right)$ the same way we bounded $(\hat{g}, \widehat{\mathcal{T}\hat{u}_f})$ in Theorem 3.4 and using the Gronwall's lemma, we obtain

$$\|\widehat{\mathcal{T}e(t)}\|^2 \leq \exp(t)\left(\|\widehat{\mathcal{T}e(0)}\|^2 + \int_0^t \|R(s)\|^2 ds\right) \text{ for all } t \geq 0. \tag{3.35}$$

We can bound the residual term by noting that, by the properties of the Chebyshev approximation [8,59], $\|\mathcal{T}\dot{u}_f(t) - \widehat{\mathcal{T}\dot{u}_f(t)}\| \leq C_1(N-p)^{-m}\|\mathcal{T}\dot{u}_f(t)\| \leq C_T(N-p)^{-m}\|\dot{u}_f(t)\|$, $\|\mathcal{A}u_f(t) - \widehat{\mathcal{A}u_f(t)}\| \leq C_2(N-p)^{-m}\|\mathcal{A}u_f(t)\| \leq C_A(N-p)^{-m}\|u_f(t)\|$ due to a boundedness of the integral operators $\mathcal{T}$, $\mathcal{A}$. Additionally, $\|g(t) - \hat{g}(t)\| \leq C_3(N-p)^{-m}\|g(t)\|$. For the commutation error, we have

$$\begin{aligned} &\|\widehat{\mathcal{T}\dot{u}_f(x,t)} - \mathcal{T}R_{N-p}\dot{u}_f(x,t)\| \leq \\ &\|\widehat{\mathcal{T}\dot{u}_f(x,t)} - \mathcal{T}\dot{u}_f(x,t)\| + \|\mathcal{T}\left(\dot{u}_f(x,t) - R_{N-p}\dot{u}_f(x,t)\right)\| \leq C_4(N-p)^{p-m}\|\dot{u}_f(t)\|, \end{aligned} \tag{3.36}$$

and, analogously, for the $\left(\widehat{\mathcal{A}u_f(x,t)} - \mathcal{A}R_{N-p}u_f(x,t)\right)$ term.

Since $\mathcal{T}u_f - \widehat{\mathcal{T}\hat{u}_f} = \mathcal{T}\left(u_f - R_{N-p}u_f\right) + \left(\mathcal{T}R_{N-p}u_f - \widehat{\mathcal{T}R_{N-p}u_f}\right) + \widehat{\mathcal{T}e}$, and noting that $e(0) = 0$ in the current definition, we obtain the desired estimate (3.32). □

Note that the estimate (3.32) implies an exponential convergence of a semi-discrete approximation (3.18) with N for smooth solutions.

3.3. Temporal treatment

If M is invertible, Eq. (3.27) can be rewritten as

$$\frac{d\,\mathbf{a}(t)}{d\,t} = \tilde{A}\,\mathbf{a}(t) + \tilde{B}\,\mathbf{b}(t), \tag{3.37}$$

where $\tilde{A} = M^{-1}A$, and $\tilde{B} = M^{-1}$. Invertibility of M generally follows from its block-diagonal structure and well-posedness of the boundary conditions. If M is not invertible, Eq. (3.27) would admit linear in time eigensolutions irrespective of the right-hand side, and this situation will not be considered here.

We now define several approaches to the time integration of (3.37).

3.3.1. Exact integration

The following lemma establishes an exact solution to the matrix Eq. (3.37).

Lemma 3.6. *The solution to the matrix Eq. (3.37) with initial conditions $\mathbf{a}(0) = \boldsymbol{a}_0$ is given by [68,69]*

$$\mathbf{a}(t) = e^{\tilde{A}t}\,\mathbf{a}_0 + \int_0^t e^{\tilde{A}(t-s)}\,\tilde{B}\,\mathbf{b}(s)\,ds. \tag{3.38}$$

Proof. Proof can be found in [68,69]. □

Upon substitution $\tilde{A} = M^{-1}A$, and $\tilde{B} = M^{-1}$ into (3.38), we recover an exact solution to Eq. (3.27) in our original notation

$$\mathbf{a}(t) = e^{M^{-1}A\,t}\,\mathbf{a}_0 + \int_0^t e^{M^{-1}A\,(t-s)}\,M^{-1}\,\mathbf{b}(s)\,ds. \tag{3.39}$$

Evaluating Eq. (3.39) in practice is, however, challenging, due to a necessity of computing an integral over the matrix exponentials, which is computationally expensive for reasonably large matrices. Evaluation of the integral can, however, be simplified, if the matrix $\tilde{A} = M^{-1}A$ is diagonalizable as $\tilde{A} = S\,\Lambda\,S^{-1}$, in which case Eq. (3.39) becomes

$$\mathbf{a}(t) = S\,e^{\Lambda\,t}S^{-1}\,\mathbf{a}_0 + S\int_0^t e^{\Lambda(t-s)}S^{-1}M^{-1}\,\mathbf{b}(s)\,d\,s. \tag{3.40}$$

This allows us to split the integral in (3.40) that involves matrix exponentials into a summation of the scalar integrals of the form

$$I_{kl} = \int_0^t e^{\lambda_k(t-s)}b_l(s)ds, \tag{3.41}$$

where λ_k and $b_l(t)$ for $k, l = 1 \dots N_d$, are the eigenvalues of $\tilde{A}$ (diagonal entries of Λ) and components of the vector $\mathbf{b}(t)$, respectively. Consequently, the entire vector-valued integral $\mathbf{I} = \int_0^t e^{\Lambda(t-s)}S^{-1}M^{-1}\,\mathbf{b}(s)\,d\,s$ in (3.40) can be evaluated componentwise as $I_k = \sum_{l=1}^{N_d} I_{kl}[S^{-1}M^{-1}]_{kl}$, where I_k is the kth component of $\mathbf{I}$, $[S^{-1}M^{-1}]_{kl}$ is the corresponding entry of the matrix $S^{-1}M^{-1}$ in the kth row and lth column, and summation over k is not implied. Furthermore, if inputs $\mathbf{b}(t)$ are separable into m time-dependent entries $\mathbf{b}(t) = \sum_{l=1}^{m}\boldsymbol{\alpha}_l b_l(t)$, $m < N_d$, $\boldsymbol{\alpha}_l$ are the vectors independent of time, the evaluation of the integral in (3.40) reduces to a computation of $m\,N_d$ integrals of the form (3.41), and the reconstruction process yields $\int_0^t e^{\Lambda(t-s)}S^{-1}M^{-1}\,\mathbf{b}(s)\,d\,s = \sum_{l=1}^{m} D_l\,S^{-1}M^{-1}\boldsymbol{\alpha}_l$, where D_l is a diagonal matrix that, for each l, consists of the corresponding I_{kl} values, such that $D_l = \mathrm{diag}(I_{kl})$, $k = 1 \dots N_d$. If the inputs $\mathbf{b}(t)$ are such that the integrals in (3.41) can be evaluated analytically, the described procedure yields an exact temporal integration of Eq. (3.37).

3.3.2. Alternative exact integration

While Eq. (3.40) and its analytical evaluation via the approach described above provides a robust solution whenever M is invertible, the ODE system (3.37) is stable, and matrix $\tilde{A} = M^{-1}A$ is diagonalizable, in some cases, we can further reduce the errors associated with the inversion of the matrix M by employing the alternative form of the solution to (3.27) given by the following lemma.

Lemma 3.7. *If matrix M is diagonalizable as $M = S\,\Lambda\,S^{-1}$ and does not have zero eigenvalues, a solution to Eq. (3.27) with initial conditions $\mathbf{a}(0) = \boldsymbol{a}_0$ is given by*

$$\mathbf{a}(t) = S\,e^{\Lambda^{-1}\,S^{-1}\,A\,S\,t}\,S^{-1}\mathbf{a}_0 + S\int_0^t e^{\Lambda^{-1}\,S^{-1}\,A\,S\,(t-s)}\Lambda^{-1}\,S^{-1}\mathbf{b}(s)\,d\,s. \tag{3.42}$$

Proof. Upon substituting $M = S\,\Lambda\,S^{-1}$ into Eq. (3.27), multiplying both sides by S^{-1}, and defining $\mathbf{z} = S^{-1}\mathbf{a}$, Eq. (3.27) reduces to

$$\Lambda\frac{d\,\mathbf{z}(t)}{d\,t} = S^{-1}A\,S\,\mathbf{z}(t) + S^{-1}\mathbf{b}(t). \tag{3.43}$$

Upon multiplying Eq. (3.43) by the inverse of Λ, the solution given by (3.42) follows immediately from (3.38) and substitution $\mathbf{a} = S\,\mathbf{z}$. □

Note that for the PDEs with constant coefficients, A would be a multiple of an identity matrix, so that $\Lambda^{-1}\,S^{-1}\,A\,S$ is by itself diagonal. Alternatively, its diagonalization similar to a procedure described in Section 3.3.1 needs to be performed for an analytical evaluation of (3.42).

Unfortunately, the eigenvalues of $M^{-1}A$ are different from the eigenvalues of $\Lambda^{-1}\,S^{-1}\,A\,S$, which can render the evaluation of the integral in (3.42) unstable, even if the integral in (3.40) is stable. This approach, therefore, cannot be advocated as a general-purpose solution. However, for diffusive problems that are inherently robust, integration via (3.42) significantly reduces approximation errors associated with the matrix inversion in (3.40). Since one of the purposes of this study is to demonstrate strong spatial convergence properties of the PGC approximation, ideally decoupled from the temporal errors, we intend to use (3.42) whenever possible.

3.3.3. Gauss integration

The analytical integration procedure described above will fail if

- $\tilde{A} = M^{-1}A$ is not diagonalizable, so that (3.39) cannot be reduced to (3.40).
- Inhomogeneous inputs $\mathbf{b}(t)$ have a functional form that does not allow for an analytical evaluation of the integral in (3.40) or (3.42).

In this case, the integral in (3.39) can be approximated numerically. In this work, we use a high-order Gauss integration to accomplish this. In particular, the total time interval is partitioned into N_{int} sub-intervals, and a Gauss–Lobatto quadrature with N_g points (including end points) is used for each time interval. This approach alleviates the problems associated with the analytical integration mentioned above, and also avoids some difficulties attributed to the classical time stepping procedures. First, it does not suffer from

the CFL-type instabilities and the associated time step restrictions of the classical time stepping schemes. As long as the ODE system is physically stable (that is, it does not possess eigenvalues with positive real parts), the Gauss integration approach remains stable. Second, Gauss integration does not require a sequential approach and can, in principle, be leveraged for developing efficient time-parallel algorithms [70,71]. Regarding its practical implementation, in some cases it was found beneficial to use a non-uniform distribution of time intervals, with their clustering towards the end of the time period t, defined by a geometric progression with a specified ratio r. Within each time interval, the Gauss–Lobatto (GL) integration with the nodes specified by GL quadrature is used.

3.3.4. Backward differentiation formula

While the above approaches associated with the approximation of the exact solution to an ODE system (3.37) in the form (3.39) provide inherently low errors due to their high temporal accuracy, their applicability can be sometimes limited. To compare analytical and Gauss integration approaches to the conventional time stepping schemes as applied to the ODE Eq. (3.37) and to show the effect of the temporal discretization errors on the solution convergence, we also implement a backward differentiation formula (BDF) for the time integration. Backward differentiation formula of order k (BDFk) is an implicit time integration scheme, which, as applied to (3.37), is given by

$$\sum_{p=0}^{k} \beta_p \, \mathbf{a}^{n-p} = \Delta t \, (\tilde{A} \, \mathbf{a}^n + \tilde{B} \, \mathbf{b}^n), \tag{3.44}$$

where Δt is the time step, and the vectors with the superscript n correspond to their values at the discrete time level t^n. BDF schemes of the order 3 and 4, denoted as BDF3 and BDF4, respectively, are considered here. The corresponding BDF coefficients β_p for these two schemes can be found, e.g., in [6,72].

3.4. Software

To enable utilization of the presented numerical methodology in a generalized and automated manner, the authors have developed a new general-purpose open-source software PIESIM implemented in MATLAB, available for download at http://control.asu.edu/pietools. PIESIM leverages the capabilities of the open-source package PIETOOLS, previously developed by the authors [73], which introduces a new *opvar* class of objects for efficient construction, manipulation and optimization of the Partial Integral operators in MATLAB. Within the context of the presented numerical methodology, PIETOOLS converts a user-defined PDE problem into a PIE framework and constructs the corresponding 3-PI operators, while PIESIM discretizes the operators, computes a numerical solution of the PIE problem using the PGC approach, and transforms the PIE solution back to represent a required solution of the original PDE problem. The developed software emerges as a general-purpose *high-order* PDE solver which ensures an automated high-order treatment of arbitrary boundary conditions and is written in a user-friendly manner that requires no user intervention apart from declaring an original PDE problem. PIESIM/PIETOOLS utilize a functional graphical user interface for an easy and intuitive declaration of the PDE problem and the boundary conditions, while the PDE-PIE transformation, discretization, simulation and post-processing are handled automatically by the solver. PIESIM/PIETOOLS can also be used for stability analysis and control of PDEs, DDEs, and coupled PDE-ODE systems and their verification [57,58,63]. All numerical examples below were solved using PIESIM.

4. Numerical results

This section demonstrates verification and application of the presented numerical methodology to canonical PDE equation problems.

4.1. Verification, data processing and error analysis

Numerical examples presented in this section serve verification purposes of the developed methodology. For this purpose, in each example, we compute the L_2 error on the basis of the analytical solution, defined as

$$L_2(\mathbf{u})(t) = \sqrt{\frac{1}{\|\Omega\|} \int_{\Omega} \left(\mathbf{u}(x,t) - \hat{\mathbf{u}}(x,t)\right)^2}, \tag{4.1}$$

where $\mathbf{u}(x,t)$ is the analytical solution, $\hat{\mathbf{u}}(x,t)$ is its numerical approximation, Ω is the computational domain, and $\|\Omega\|$ is its volume (length in one dimension). The error in (4.1) consists of a sum of a spatial error, which is due to a projection of the infinite-dimensional PDE solution onto a space of polynomials of degree N in (3.18), and a temporal error, which is due to a temporal integration of the ODE Eq. (3.27) for the expansion coefficients. The error due to a polynomial projection (spatial error) can be controlled by the polynomial degree used (N). Bound on the spatial error in the developed PGC formulation arises directly from the convergence proof in Theorem 3.5 and, in fact, is theoretically shown to decrease exponentially with N for smooth solutions. This propety is verified numerically in the subsequent examples by investigating the L_2 error (4.1) behavior versus N. The temporal error (due to the integration rule) can be controlled by the choice of the temporal scheme and by the time step, Δt. Bound on the temporal error can be approximated as $C(\Delta t)^k$, where k is the accuracy of the temporal scheme [74].

Table 1
Analytical and numerical estimates of the temporal error bound in Example 1a based on $\Delta t = 10^{-3}$ and a value of empirical constant $C = 100$.

Numerical Scheme	Estimate temporal error	Actual final error
BDF3	10^{-7}	10^{-7}
BDF4	10^{-10}	10^{-10}
Gauss	N/A	10^{-12}
Analytical Eq. (3.40)	N/A	10^{-12}
Analytical Eq. (3.42)	10^{-16}	10^{-15}

4.2. Numerical examples with PIEs: Parabolic problems

4.2.1. Example 1: Diffusion equation

Example 1a: Constant viscosity. We consider a diffusion equation

$$u_t = \nu\, u_{xx}, \tag{4.2}$$

whit ν a scalar, defined on a domain $x \in [-1, 1]$. In terms of a standardized representation given in Section 2.1, Eq. (4.2) yields $A_0(x) = A_1(x) = 0$, $A_2(x) = \nu$, $n_0 = n_1 = 0$, $n_2 = 1$. Furthermore, $u_2(x,t) = u(x,t)$ is a primary state, while from (2.7), $u_{f2}(x,t) = u_{xx}(x,t)$ is a fundamental state. We consider Dirichlet–Neumann boundary conditions, defined as $u(-1,t) = h_1(t), u_x(1,t) = h_2(t)$, with the corresponding boundary conditions matrix

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.3}$$

With this value of B, the 3-PI operators $\mathcal{T}$ and $\mathcal{A}$ in (2.15) for Eq. (4.2) are parameterized by $G_0 = 0, G_1(x,s) = x - s, G_2(x,s) = -x - 1$, and $H_0(x) = \nu$, $H_1 = H_2 = 0$, respectively, so that the corresponding PIE representation of (4.2) reads

$$\int_{-1}^{x} (x-s)\, \dot{u}_{f2}(s,t)\, ds - (x+1) \int_{-1}^{1} \dot{u}_{f2}(s,t)\, ds = \nu\, u_{f2}(x,t), \tag{4.4}$$

where a dot above the function denotes a partial derivative in time.

Applying the discretization procedure described in Section 3, we obtain a discrete $N_d \times N_d$ matrix M, which, given that $n_0 = n_1 = 0$, $n_2 = 1$, reduces to a $(N-1) \times (N-1)$ matrix, while the matrix $A = \nu \cdot I$. The graphical illustration of the physical process governed by the considered PDE problem with $\nu = 0.5$ is given in Fig. 1. The solution decays in time due to the influence of diffusion process. The solution and the convergence plots with N for this test case with different time integration approaches are presented for $\nu = 0.5$, time step $\Delta t = 10^{-3}$, and $t = 0.1$ in Fig. 2.

As discussed in Section 4.1, the L_2 error estimated by Eq. (4.1) consists of a spatial and a temporal error. Fig. 2 demonstrates that the error for all the schemes initially decays exponentially with N while the exponential decay stops at some finite value of N. This is due to the temporal error beginning to dominate the spatial error and to halt the convergence. We can estimate the bound on a temporal error invoking the arguments presented in Section 4.1 and compare it to the actual final error for this example in Table 1 (we define the actual final error as the error attained by each numerical scheme at the highest value of N investigated, which is $N = 48$ in this study). Since precise value of a constant in the temporal error bound estimate is unknown, we estimate it based on a value of $C = 100$, which gives a good agreement with the actual final error. Note that the temporal error in the Gauss integration and in the analytical integration based on Eq. (3.40) is difficult to estimate, since this error is associated with the inaccuracies of computing the matrix exponential containing the matrix inverse; however, the numerical example shows this error to be on the order of 10^{-12}. The important implication of this analysis is that: (1) The plots demonstrate the exponential convergence of the error with N for all the integration schemes before the temporal error starts to dominate, as predicted by the theoretical analysis in Section 3.1, and (2) PIE methodology with the analytical evaluation of the integral based on (3.42) provides an unprecedentedly low error close to a machine precision $O(10^{-15})$ in a solution of an unsteady PDE problem with time-dependent inputs. The analytical approach is followed behind just slightly by a Gauss integration of Eq. (3.39), with a final error of $O(10^{-12})$. Other numerical examples discussed below demonstrate a similar error behavior, albeit a constant in the temporal error bounds varies.

Example 1b: Variable viscosity. We consider a diffusion equation with a variable viscosity,

$$u_t = \nu(x)\, u_{xx}, \tag{4.5}$$

with $\nu(x) = x$.

We use the domain $x \in [0, 2]$ to ensure a non-negative value of viscosity for a physically stable solution. We define initial conditions as $u(x,0) = -x^2$, boundary conditions as Dirichlet–Dirichlet with $u(0,t) = 0$, $u(2,t) = -4t - 4$, so that the boundary conditions matrix is

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.6}$$

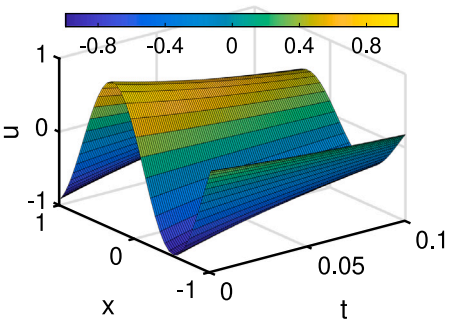
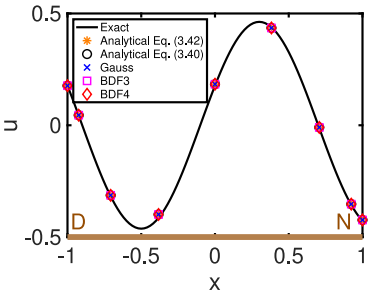
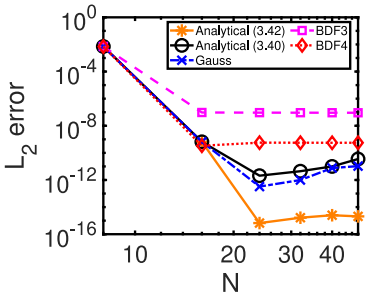


Fig. 1. Spatio-temporal behavior of the solution to the Example 1a for $(x,t) \in [-1,1] \times [0,0.1)$: diffusion problem with Dirichlet-Neumann boundary conditions and a constant viscosity $\nu = 0.5$. The solution decays in time due to the influence of diffusion process.



(a) Solution plot at $N = 8$. Black solid line, exact solution; symbols, numerical solutions (see the legend).



(b) L_2 error versus the polynomial order N.

Fig. 2. Solution (a) and convergence plots (b) for Example 1a: diffusion equation with constant viscosity $\nu = 0.5$ and Dirichlet–Neumann boundary conditions at a time $t = 0.1$. In (a): a brown thick line represents a solution domain, and the letters "D, N" on the left and right indicate the specified boundary conditions. Orange solid line with asterisks, analytical evaluation of Eq. (3.42); black solid line with circles, analytical evaluation of Eq. (3.40); blue dash-dotted line with crosses, Gauss integration of Eq. (3.39) with $N_g = 10$ and $N_{int} = 10$ non-uniform time intervals with the geometric progression ratio $r = 0.25$; magenta dashed line with squares, BDF3 with $\Delta t = 10^{-3}$; red dotted line with diamonds, BDF4 with $\Delta t = 10^{-3}$.

In this case, the analytical solution exists, which is given by $u(x,t) = -2\,x\,t - x^2$. When the physical domain does not coincide with $[-1,1]$, a mapping of the physical domain $x \in [a,b]$ into the computational domain $x^{(c)} \in [-1,1]$ must be performed as discussed in Section 3.1. In this case, for the 3-PI operators, we have $G_0^{(c)} = 0, G_1^{(c)}(x^{(c)},s^{(c)}) = x^{(c)} - s^{(c)}, G_2^{(c)}(x^{(c)},s^{(c)}) = \frac{1}{2}(x^{(c)}+1)(s^{(c)}-1)$, and $H_0^{(c)}(x^{(c)}) = x^{(c)} + 1$, $H_1^{(c)} = H_2^{(c)} = 0$, where the superscript (c) indicates that the functions and 3-PI operators are evaluated in the computational domain. The solution and the convergence plots for this test case are presented for $t = 0.1$ in Fig. 3. Note, since the exact solution is a second-order polynomial, which is resolved with Chebyshev approximation starting with $N = 2$, the error at the lowest N investigated ($N = 8$) is already at a machine precision in this test case.

4.2.2. *Example 2: Euler–Bernoulli beam*

Euler–Bernoulli beam model is represented by a fourth-order PDE

$$u_{tt} = -c\,u_{xxxx}, \tag{4.7}$$

on the domain $x \in [0,L]$, where L is the length of the beam, $c = EI/\mu$, E is the elastic modulus, I is the second moment of area of the beam's cross-section, and μ is the mass per unit length. In a cantilevered state described by the boundary conditions

$$u(0,t) = u_x(0,t) = u_{xx}(L,t) = u_{xxx}(L,t) = 0, \tag{4.8}$$

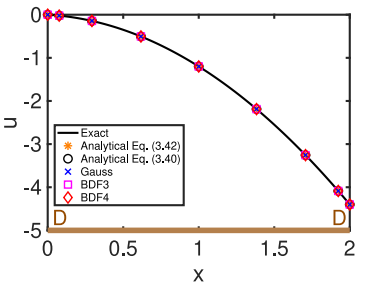
a free vibration solution exists given by the following harmonic modes $u_n(x,t) = \mathrm{Re}\left[\tilde{u}_n(x)\, e^{-i\omega_n t}\right]$ [75], with eigenmodes

$$\tilde{u}_n(x) = A_n[\cosh(\beta_n x) - \cos(\beta_n x) + \frac{\cos(\beta_n x) + \cosh(\beta_n x)}{\sin(\beta_n x) + \sinh(\beta_n x)}(\sin(\beta_n x) - \sinh(\beta_n x))], \tag{4.9}$$

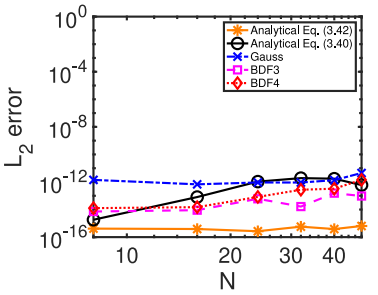
and the eigenvalues β_n being a solution of the following eigenvalue problem

$$\cosh(\beta_n L)\cos(\beta_n L) + 1 = 0, \tag{4.10}$$

and the vibration frequencies defined as $\omega_n = \beta_n^2\sqrt{EI/\mu} = \beta_n^2\sqrt{c}$.



(a) Solution plot at $N = 8$. Solid line, exact solution; symbols, numerical solutions (see the legend).



(b) L_2 error versus the polynomial order N.

Fig. 3. Solution (a) and convergence plots (b) for Example 1b: diffusion equation with variable viscosity $\nu(x) = x$ with Dirichlet-Dirichlet boundary conditions at a time $t = 0.1$. In (a): a brown thick line represents a solution domain, and the letters "D, D" on the left and right indicate the specified boundary conditions. Orange solid line with asterisks, analytical evaluation of Eq. (3.42); black solid line with circles, analytical evaluation of Eq. (3.40); blue dash-dotted line with crosses, Gauss integration of Eq. (3.39) with $N_g = 10$ and $N_{int} = 10$ non-uniform time intervals with the geometric progression ratio $r = 0.25$; magenta dashed line with squares, BDF3 with $\Delta t = 10^{-3}$; red dotted line with diamonds, BDF4 with $\Delta t = 10^{-3}$.

To cast Eq. (4.7) into a state-space representation of (2.1), we define the following states $v_1(x,t) = u_t(x,t)$, $v_2(x,t) = u_{xx}(x,t)$, so that (4.7) transforms into

$$\mathbf{v}_t = \underbrace{\begin{bmatrix} 0 & -c \\ 1 & 0 \end{bmatrix}}_{A_2} \mathbf{v}_{xx}, \tag{4.11}$$

where the state vector $\mathbf{v} = [v_1 \ v_2]^T$, $n_0 = n_1 = 0, n_2 = 2$, which represents an example of a vector-valued state. Thus, the fundamental state is $\mathbf{v}_f = [v_{1xx} \quad v_{2xx}]^T$, $A_0 = A_1 = 0$, and A_2 is as given by Eq. (4.11). For the boundary conditions defined by (4.8), the last two equations can be restated in terms of the state $v_2(x,t)$ as $v_2(L,t) = 0, v_{2x}(L,t) = 0$. The first two boundary conditions can be differentiated in time to give boundary constraints for the state $v_1(x,t)$ as $v_1(0,t) = 0, v_{1x}(0,t) = 0$. With these, the boundary conditions matrix B reads

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{B} \begin{bmatrix} v_1(0,t) \\ v_2(0,t) \\ v_1(L,t) \\ v_2(L,t) \\ v_{1x}(0,t) \\ v_{2x}(0,t) \\ v_{1x}(L,t) \\ v_{2x}(L,t) \end{bmatrix} = 0. \tag{4.12}$$

To reconstruct the original variable $u(x,t)$ from the state-space variables $v_1(x,t)$, $v_2(x,t)$, we can utilize Eq. (2.12) to recover $u(x,t)$ from its second-derivative $u_{xx}(x,t) = v_2(x,t)$. In the PIE framework, this effectively can be done by a transformation (2.18) applied to $v_2(x,t)$, with $\mathcal{T} = \{0, x-s, 0\}$, $K(x)B_T^{-1} = \begin{bmatrix} 1 & x-a \end{bmatrix}$, and $\mathbf{h}(t) = \begin{bmatrix} u(a,t) & u_x(a,t) \end{bmatrix}^T$, with $a = 0$.

In the following, we choose $L = 2$ and keep our solution domain at $x^{(c)} \in [-1, 1]$ while recovering the original solution in $x \in [0, 2]$ by the transformation $x = x^{(c)} + 1$. The solution and the convergence plots for the second to fourth eigenmodes of a cantilever beam are shown in Fig. 4 at $t = 0.1$ obtained with $c = 2$, $\Delta t = 10^{-3}$. To compute these solutions, we set the initial conditions corresponding to an eigenmode shape (4.9) with the amplitude $A_n = 1$ for each eigenmode, which is an exact solution at $t = 0$. It can be noted that the first (not shown here) and the second eigenmodes are well captured with $N = 8$. The third eigenmode has a slight deviation near the free boundary at $N = 8$, but a correct shape starting with $N = 16$. The fourth eigenmode shows a vastly incorrect deflection with $N = 8$, while recovering a correct shape starting with $N = 16$. Note that the tolerance in solving a nonlinear eigenvalue problem (4.10) must be set to a very low value (10^{-16} was used in the current work) to obtain these convergence plots, otherwise the convergence will be limited by the value of the set tolerance.

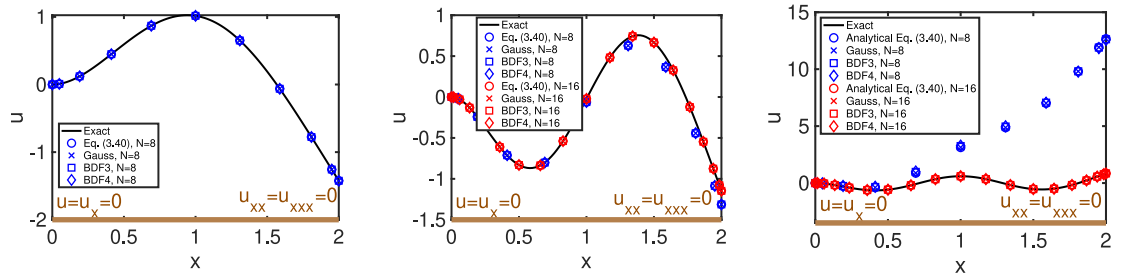
4.3. Numerical examples with PIEs: hyperbolic problems

4.3.1. Example 3: Transport equation

Example 3a: Propagating Gaussian bump. Here, we consider a transport equation of the form

$$u_t + c\, u_x = 0, \tag{4.13}$$

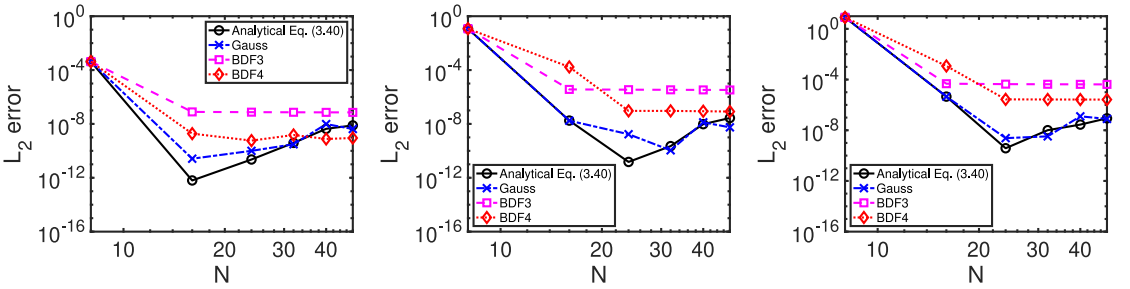
on the domain $x \in [-1, 1]$, with $A_0(x) = 0, A_1(x) = -c, A_2(x) = 0$. We have $n_0 = n_2 = 0$, $n_1 = 1$, leading to a primary state $u_1(x,t) = u(x,t)$, and a fundamental state $u_{f1}(x,t) = u_x(x,t)$. The transport equation admits solutions in the form of right- (for



(a) Second eigenmode. $N = 8$ (blue); black, exact solution.

(b) Third eigenmode. $N = 8$ (blue); $N = 16$ (red); black, exact solution.

(c) Fourth eigenmode. $N = 8$ (blue); $N = 16$ (red); black, exact solution.



(d) Second eigenmode.

(e) Third eigenmode.

(f) Fourth eigenmode.

Fig. 4. Solution (a)–(c) and convergence plots (d)–(e) for Example 2: Euler–Bernoulli beam equation with $c = 2$ at a time $t = 0.1$. In (a)–(c): a brown thick line represents a solution domain, with the specified boundary conditions at the left and right ends of the beam indicated in brown letters. Black solid line with circles, analytical evaluation of Eq. (3.40); blue dash-dotted line with crosses, Gauss integration of Eq. (3.39) with $N_g = 10$ and $N_{int} = 10$ non-uniform time intervals with the geometric progression ratio $r = 0.25$; magenta dashed line with squares, BDF3 with $\Delta t = 10^{-3}$; red dotted line with diamonds, BDF4 with $\Delta t = 10^{-3}$.

$c > 0$), or left- (for $c < 0$) propagating waves. We consider a test case of a propagating Gaussian bump given by the exact solution $u(x,t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-ct-\mu}{\sigma})^2}$, with the corresponding initial condition and a Dirichlet boundary condition. For $c > 0$, we specify a Dirichlet boundary condition at the left at $x = -1$. The matrix B in this case reduces to $B = [1 \quad 0]$, $K(x) = 1$, $K(x)B_T^{-1} = 1$, and the 3-PI operators are $G_0 = 0, G_1 = 1, G_2 = 0$, and $H_0 = -c, H_1 = H_2 = 0$. Choosing $\sigma = 0.2$, $\mu = 0$ and $c = 4$, the solution and the convergence plots are presented in Fig. 5 at a time $t = 0.1$. As with the Euler–Bernoulli beam example, it is seen that the Gaussian bump is not well resolved with $N = 8$ points, while a correct solution profile is recovered starting at $N = 16$.

Example 3b: Long-term stability: Traveling Sine wave. We demonstrate long term stability and strong conservation properties of the PGC methodology on the example of a traveling sine wave in the form of $u(x,t) = \sin(x - c\,t)$, where initial conditions $u(x,0) = \sin(x)$ and boundary conditions $u(-1,t) = \sin(-1 - c\,t)$ are specified. The results of a long-time integration at $t = 100$ and $c = 4$ are presented in Fig. 6. It is seen that the traveling sine wave is well recovered with $N = 8$ points, and the solution is perfectly conserved even after $t = 100$ time units.

4.3.2. Example 4: Wave equation

Example 4a: Dirichlet-Neumann boundary conditions. We now proceed to solving a wave equation of the form

$$u_{tt} = c^2\, u_{xx} \tag{4.14}$$

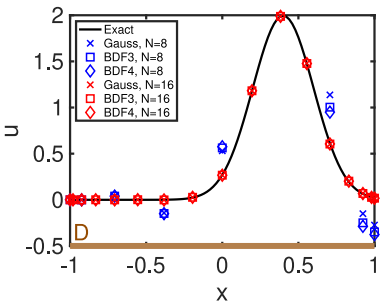
on the domain $x \in [-1, 1]$ with Dirichlet-Neumann boundary conditions $u(-1,t) = h_1(t), u_x(1,t) = h_2(t)$ and initial conditions

$$u(x,0) = f(x), u_t(x,0) = g(x). \tag{4.15}$$

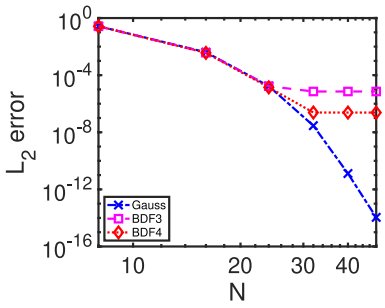
The exact solution to the wave equation is given by the d'Alembert's formula [1,2] and depends on the initial conditions for both the function $u(x,0)$ and its time derivative $u_t(x,0)$,

$$u(x,t) = \frac{1}{2}\left[f(x - ct) + f(x + ct)\right] + \frac{1}{2c}\int_{x-ct}^{x+ct} g(\xi)\, d\,\xi, \tag{4.16}$$

where the functions $f(x)$ and $g(x)$ come from the initial conditions (4.15). Thus, in general, the solution to the wave equation consists of the left- and right-propagating waves. However, in certain situations, depending on the initial conditions, one of the waves can cancel out due to a contribution from the initial conditions on the time derivative, which results in a single left- or right-traveling wave solution.

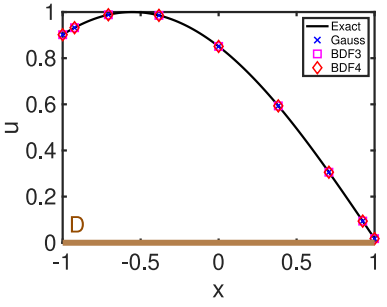


(a) Solution plot at $N = 8$ (blue) and $N = 16$ (red). Exact solution is in black.

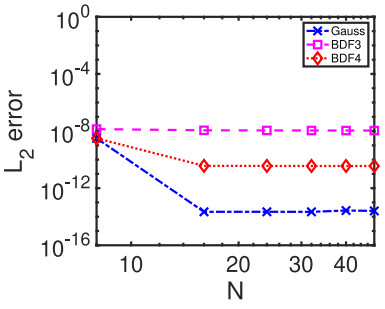


(b) L_2 error versus the polynomial order N.

Fig. 5. Solution (a) and convergence plots (b) for Example 3a: transport equation for a propagating Gaussian bump with $c = 4$, $\sigma = 0.2$, $\mu = 0$ and Dirichlet boundary condition at the left at a time $t = 0.1$. In (a): a brown thick line represents a solution domain, and the letter "D" on the left indicates the specified boundary condition. Blue dash-dotted line with crosses, Gauss integration of Eq. (3.39) with $N_g = 100$ and $N_{int} = 1$; magenta dashed line with squares, BDF3 with $\Delta t = 10^{-3}$; red dotted line with diamonds, BDF4 with $\Delta t = 10^{-3}$.



(a) Solution plot at $N = 8$. Solid line, exact solution; symbols, numerical solutions (see the legend).



(b) L_2 error versus the polynomial order N.

Fig. 6. Solution (a) and convergence plots (b) for Example 3b: transport equation for a traveling sine wave with $c = 4$ and Dirichlet boundary condition at the left at a time $t = 100$. In (a): a brown thick line represents a solution domain, and the letter "D" on the left indicates the specified boundary condition. Blue dash-dotted line with crosses, Gauss integration of Eq. (3.39) with $N_g = 100$ and $N_{int} = 100$ uniform time intervals; magenta dashed line with squares, BDF3 with $\Delta t = 10^{-3}$; red dotted line with diamonds, BDF4 with $\Delta t = 10^{-3}$. This example demonstrates long-term stability and conservation properties of the method.

To reduce a wave equation to its standardized state-space form given by (2.1), we introduce two states $v_1(x,t) = u_t(x,t)$, $v_2(x,t) = u_x(x,t)$, with the corresponding boundary conditions on the states $v_1(-1,t) = g_1'(t), v_2(1,t) = g_2(t)$, i.e., in terms of the new state vector $\mathbf{v} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$, we have Dirichlet boundary conditions (albeit at different ends) on both (first-order) states. With this state vector, Eq. (4.14) now looks

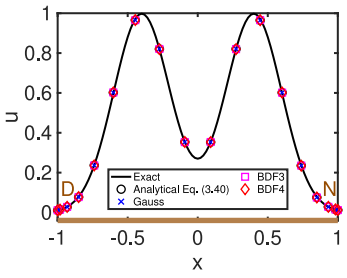
$$\mathbf{v}_t = \underbrace{\begin{bmatrix} 0 & c^2 \\ 1 & 0 \end{bmatrix}}_{A_1} \mathbf{v}_x. \tag{4.17}$$

The fundamental state is, therefore, $\mathbf{v}_f = \begin{bmatrix} v_{1x} & v_{2x} \end{bmatrix}^T$, and we have $n_0 = n_2 = 0$, $n_1 = 2$. As in the Euler–Bernoulli beam example, to recover the original variable $u(x,t)$ from a state-space variable $u_x(x,t)$, we need to perform an additional transformation $u(x,t) = \mathcal{T} u_x(x,t) + K(x) B_T^{-1} \mathbf{h}(t)$, with $\mathcal{T} = \{0,1,0\}$, $K(x) B_T^{-1} = 1, \mathbf{h}(t) = u(-1,t)$ which corresponds to the formula (2.11).

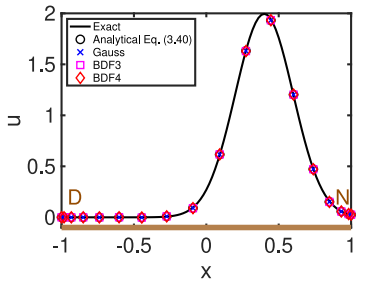
As discussed above, the exact solution to the wave equation depends on the initial conditions on both the functions $u(x,t)$ and $u_t(x,t)$. We first show how, depending on the initial conditions on the derivative $u_t(x,0)$, the same initial shape in a form of a Gaussian bump given by the function $u(x,0) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$, can either propagate in one direction, or split in half and give rise to the left- and right-propagating waves.

Splitting case. According to the d'Alembert's formula (4.16), a splitting case is realized if the initial time derivative $u_t(x,0) = g(x) = 0$, and we have the following exact solution

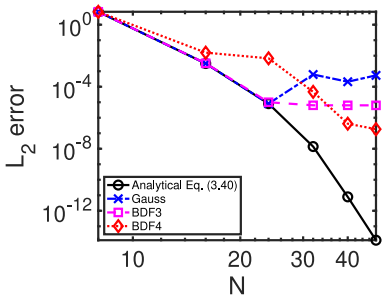
$$u(x,t) = \frac{1}{2\sigma\sqrt{2\pi}} \left[e^{-\frac{1}{2}(\frac{x-ct-\mu}{\sigma})^2} + e^{-\frac{1}{2}(\frac{x+ct-\mu}{\sigma})^2} \right]. \tag{4.18}$$



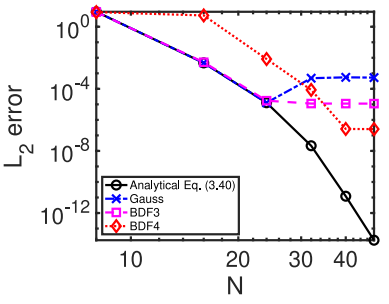
(a) Solution plot at $N = 16$ for the splitting case.



(b) Solution plot at $N = 16$ for the right-propagating case.



(c) L_2 error versus the polynomial order N for the splitting case.



(d) L_2 error versus the polynomial order N for the right-propagating case.

Fig. 7. Solution (a),(b) and convergence plots (c),(d) for Example 4a: wave equation for a Gaussian bump with Dirichlet-Neumann boundary conditions with $c = 4$, $\sigma = 0.2$, $\mu = 0$ at a time $t = 0.1$ for the splitting case (left), and the right-propagating case (right). In (a),(b): a brown thick line represents a solution domain, and the letters "D, N" on the left and right indicate the specified boundary conditions. Black solid line with circles, analytical evaluation of Eq. (3.40); blue dash-dotted line with crosses, Gauss integration of Eq. (3.39) with $N_g = 100$ and $N_{int} = 1$; magenta dashed line with squares, BDF3 with $\Delta t = 10^{-3}$; red dotted line with diamonds, BDF4 with $\Delta t = 10^{-3}$.

Right-propagating case. In this case, the initial time derivative is specified as

$$u_t(x,0) = g(x) = c\left(\frac{x - c\,t - \mu}{\sigma^2}\right) \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-ct-\mu}{\sigma}\right)^2}, \tag{4.19}$$

and the exact solution is

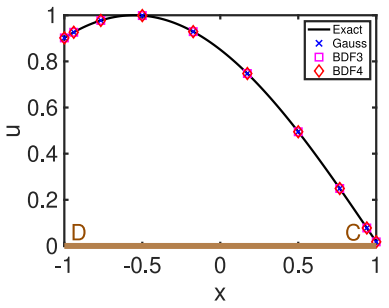
$$u(x,t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-ct-\mu}{\sigma}\right)^2}. \tag{4.20}$$

Choosing $\sigma = 0.2$, $\mu = 0$, and $c = 4$, the numerical solution obtained with the PGC framework and the convergence plots are shown in Fig. 7 at $t = 0.1$ with $\Delta t = 10^{-3}$ for both splitting and right-propagating cases.

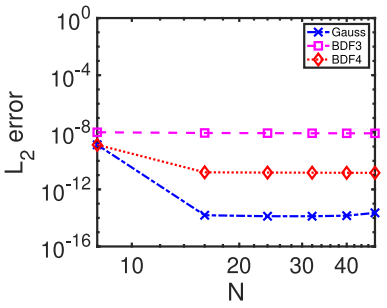
Example 4b: Dirichlet-characteristic boundary conditions. Since the exact value of the function derivative at the domain outflow is typically not available, we are now considering a characteristic, or a "non-reflecting", boundary condition at the right end of the domain given by a characteristic equation $u_t + c\,u_x = 0$, while keeping a Dirichlet boundary condition at the left end of the domain. The advantage of the PIE framework is that the characteristic boundary condition, which is an optimum choice for an outflow boundary condition in hyperbolic problems [76,77], can now be enforced exactly in a strong form. For that, the matrix B is given by

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & c \end{bmatrix}. \tag{4.21}$$

The implemented built-in characteristic outflow boundary condition demonstrates robustness and ensures conservation and long term stability of the solution to the wave equation with all the time stepping schemes considered. The results for the traveling sine wave of the form $u(x,t) = \sin(x - ct)$ with $c = 4$ are presented in Fig. 8 for the time $t = 100$. As with the transport equation, no numerical dissipation or dispersion of the solution is observed at time $t = 100$.



(a) Solution plot at $N = 8$. Solid line, exact solution; symbols, numerical solutions (see the legend).



(b) L_2 error versus the polynomial order N.

Fig. 8. Solution (a) and convergence plots (b) for Example 4b: wave equation for a traveling sine wave with Dirichlet-Characteristic boundary conditions with $c = 4$ at a time $t = 100$. In (a): a brown thick line represents a solution domain, and the letters "D, C" on the left and right indicate the specified boundary conditions. Blue dash-dotted line with crosses, Gauss integration of Eq. (3.39) with $N_g = 100$ and $N_{int} = 100$ uniform time intervals; magenta dashed line with squares, BDF3 with $\Delta t = 10^{-3}$; red dotted line with diamonds, BDF4 with $\Delta t = 10^{-3}$. This example demonstrates long-term stability and strong conservation properties of the PGC method with the enforcement of characteristic outflow boundary conditions for the wave equation through the PIE framework.

4.4. Comparison of PIEs with direct PDE solution methods

In this section, we compare the presented numerical methodology based on a transformation of a PDE to a PIE system with some classical methods of a direct PDE solution. To keep the comparison fair, we discretize the PDE system with a Chebyshev-tau approach [8,12], so that both the PIE and PDE discretizations are spectrally-convergent in space (for smooth solutions) and employ the same set of basis functions, which are Chebyshev polynomials of the first kind. The difference is in satisfying the boundary conditions: in PIE methodology, the boundary conditions are embedded into the equation dynamics by the PIE transformation, leaving the solution (in its fundamental state) to be free of constraints; in a direct PDE solution, the boundary conditions need to be satisfied at the solution level, which is accomplished in the tau method by constraining the corresponding Chebyshev expansion coefficients through the algebraic relations [8,12,14]. We present a comparison for the two representative problems: a parabolic problem represented by a diffusion equation of Example 1a, and a hyperbolic problem represented by a wave equation of Example 4b.

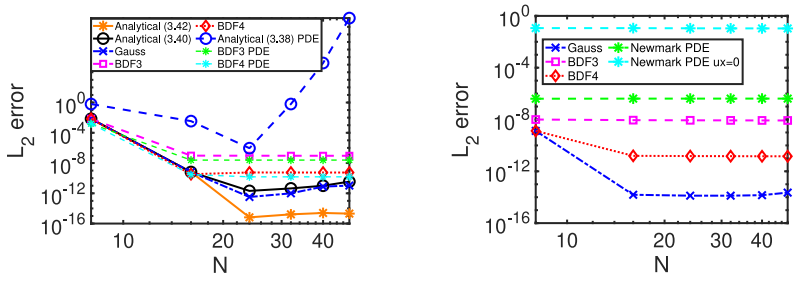
4.4.1. Comparison case 1: Diffusion equation

We consider a numerical setup of Example 1a and discretize the PDE diffusion Eq. (4.2) with a constant viscosity with the Chebyshev-tau methodology by projecting the solution $u(x,t)$ onto a set of Chebyshev polynomial basis functions $T_k(x)$ up to a degree N as $\hat{u}(x,t) = \sum_{k=0}^{N} a_k(t)T_k(x)$, where $\hat{u}(x,t)$ represents the corresponding orthogonal projection of $u(x,t)$ onto the polynomial basis. After applying a method of weighted residuals to Eq. (4.2) in a similar manner as was done with Eq. (3.25), we obtain a system of ODEs as

$$\frac{d\,\mathbf{a}(t)}{d\,t} = D_2\mathbf{a}(t), \tag{4.22}$$

where $\mathbf{a}(t)$ is the vector of Chebyshev coefficients, and D_2 is the Chebyshev differentiation matrix of the second order [8,59]. Eq. (4.22) can then be solved in time by any temporal integration method of choice. To enforce the boundary conditions, the last n_b rows of the system (4.22), where n_b is the number of boundary conditions, need to be replaced with the corresponding algebraic constraint equations on the expansion coefficients $a_k(t)$. For example, we have $\sum_{k=0}^{N} a_k(t)T_k(x_b) = u(x_b,t)$ for a Dirichlet constraint, and $\sum_{k=0}^{N}(D_1\mathbf{a}(t))_k T_k(x_b) = u_x(x_b,t)$ for a Neumann constraint, where D_1 is the Chebyshev differentiation matrix of the first order, $x_b = \pm 1$ is the location of the boundary in question, $T_k(1) = 1$, and $T_k(-1) = (-1)^k$.

A comparison of the PIE and PDE solution convergence properties for this example is shown in Fig. 9(a). First of all, we note that since boundary conditions in the direct PDE solution approaches are no longer embedded into the equation dynamics, the integral in Eqs. (3.38)–(3.42) vanishes. However, the matrix D_2 in (4.22), being a differentiation matrix on the expansion coefficients, is both rank-deficient and non-diagonalizable. As a result, computing a matrix exponential in Eq. (3.38) results in highly inaccurate and unstable results that blow up with high N. This precludes direct PDE methods from a possibility of providing solutions to unsteady partial-differential equations that are integrated exactly in time, contrary to the PIE methods, which are capable of providing such solutions. As a result, direct PDE solution approaches will be dominated by the errors resulting from a temporal integration of the solution even when a high-order spatial discretization is employed. PIE methods have a potential to overcome this limitation. We remark that a discrete time integration of Eqs. (3.37) and (4.22) with BDF schemes results in a similar accuracy between the PDE and the PIE methods.



(a) Heat equation (b) Wave equation

Fig. 9. Comparison of PIE and PDE solution methods with Chebyshev–Galerkin and Chebyshev-tau approximations, respectively, for (a) heat equation with $\nu = 0.5$, and (b) wave equation with $c = 4$. L_2 error versus the polynomial order N is presented for different temporal integration schemes. The plots corresponding to the PDE methods are labeled as "PDE" in the legend; all other plots correspond to PIE methods. In (a): both PDE and PIE methods utilize Dirichlet boundary conditions on the left, and Neumann on the right. In (b): PIE methods use Dirichlet on the left (inflow) and characteristic on the right (outflow) boundary conditions, and PDE methods use Dirichlet on the left (inflow) and Neumann on the right (outflow) boundary conditions. The plot labeled "Newmark PDE" uses u_x from the exact solution, while 'Newmark PDE $u_x = 0$" sets the outflow gradient to zero.

4.4.2. *Comparison case 2: Wave equation*

Here, we assess the ability of the PDE and the PIE methods to perform long-time integration of the wave equation, which is a stringent test for numerical stability and accuracy due to a non-dissipative feature of the solution and the necessity to satisfy the conservation laws [31–33]. We consider a sine wave propagation problem of Example 4b, with an exact solution $u(x,t) = \sin(x - c\,t)$ with $c > 0$. For a direct PDE solution, we discretize the wave Eq. (4.14) using a Chebyshev polynomial approximation in space, as in Section 4.4.1, yielding the ODE system for the Chebyshev coefficients

$$\frac{d^2\mathbf{a}(t)}{d\,t^2} = c^2 D_2 \mathbf{a}(t). \tag{4.23}$$

Previously employed BDF schemes, which approximate the first temporal derivative, are not immediately applicable to Eq. (4.23), which is a second order in time. Thus, we invoke a Newmark β-method [78,79] to discretize Eq. (4.23), which is a standard approach for a temporal integration of second-order equations. With the employed parameters $(\beta, \gamma) = (0.25, 0.5)$, the method is implicit and second-order accurate [78]. For both PDE and PIE methods, we specify Dirichlet boundary conditions at the left end of the domain. At the right end, for the PIE formulation, we use characteristic outflow conditions $(u_t + c u_x = 0)$ as described in Example 4b in Section 4.3.2. For the PDE Newmark method, a Neumann boundary condition with $u_x(1,t) = \cos(1 - c\,t)$, corresponding to the exact derivative, is originally employed at the right end of the domain. An imposition of the characteristic outflow condition in the Newmark method was infeasible. Since the displacement coefficients $(\mathbf{a})$ are updated first, and the velocity coefficients $(\mathbf{a}_t)$ are updated next, this results in an explicit treatment of the characteristic outflow condition, which leads to instability. A comparison of the PIE and PDE solutions is presented in Fig. 9(b). We see that the accuracy of the PDE solution is inferior in this example due to a lower accuracy of the Newmark scheme as compared to the other methods. Furthermore, an exact spatial derivative of the solution at the outflow is typically not known; a standard way (unless characteristic condition is employed) is to specify a zero gradient at the outflow, i.e. $u_x = 0$. Comparison of the outflow with $u_x = 0$ in a PDE method is also presented in Fig. 9(b). One can see that the accuracy of the outflow condition with $u_x = 0$ is significantly low, with the error of the order of $O(1)$ independent on N. This example demonstrates the beneficial properties of the PIE methodology in enforcing physically-relevant boundary conditions in a consistent manner. For example, a strong imposition of physically-consistent characteristic outflow condition with the PIE formulation leads to a long-term stability, conservation, and a high accuracy of the solution to the wave equation; outflow conditions have long been known as a grand challenge in development of accurate and stable numerical methodologies of physically-relevant problems [76,77].

4.4.3. *Comparison of CPU time and operation count*

Comparison of the operation count taken by different time integration schemes in the PIE and PDE methods is presented in Table 2. The comparison is performed in MATLAB R2023b using the function "*socFunctionAnalyzer*" available in a System on Chip (SoC) Blockset of MATLAB.

While operation count represents an important metric, it may be misleading, since the operations themselves may range in terms of their computational complexity. For example, the Gauss method shows on the order of $O(10^2)$ more operations than the BDF methods; however, a majority of the operations in the Gauss method are composed of fast integer additions and subtractions. Likewise, PIE BDF methods show approximately 30%–40% higher operation count than PDE BDF methods. However, one needs to consider that PIE methods operate on smaller matrices for the equivalent approximation accuracy of the primary solution, since PIEs search for the fundamental state (composed of the function derivatives), which resides in a lower-degree polynomial subspace.

To account for these considerations, we compare the CPU time taken by different numerical schemes in PIE and PDE methods in Fig. 10. Timing comparison is performed utilizing the "*tic-toc*" function in MATLAB R2023b running on a desktop with a 2.3 GHz 8-Core Intel Core i9 processor. First, it can be noted that, despite of the aforementioned issues, the operation count in Table 2 is a fairly accurate indicator of the actual timing performance scaling between different methods. Second, it can be remarked that analytical

Table 2
Comparison of the operation count between different schemes in PIE and PDE methods in MATLAB R2023b. Operation count is invariant of the polynomial degree N, since matrix operations are counted separately from scalar operations. The number of time steps $N_{steps} = 100$ is used in all the methods.

Numerical Scheme	Total operations	Scalar multiplications	Matrix multiplications	Other
PIE BDF3	9816	2231	1372 matrix N − 1	6213
PIE BDF4	10 303	2236	1552 matrix N − 1	6515
PIE Gauss	1 052 699	50 046	3942 matrix N − 1	998 711
PDE BDF3	7087	1484	786 matrix N + 1	4817
PDE BDF4	7545	1475	972 matrix N + 1	5098
PDE Newmark	10 364	1502	602 matrix N + 1	8260

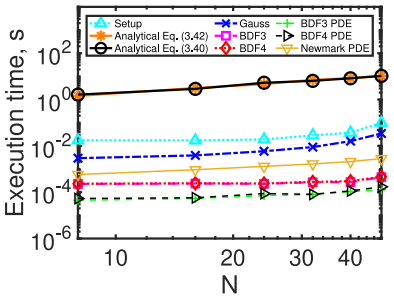


Fig. 10. CPU time taken by different numerical schemes for the PIE and PDE methods versus the polynomial degree of approximation, N. The plots corresponding to the PDE methods are labeled as "PDE" in the legend; all other plots correspond to PIE methods. Comparison is performed in MATLAB R2023b on a desktop with a 2.3 GHz 8-Core Intel Core i9 processor. The number of time steps $N_{steps} = 100$ is used in all the methods.

methods that require a symbolic integration in MATLAB take considerably longer, which is not unexpected. High computational cost of these methods is however compensated by an unprecedented accuracy close to machine precision that they yield across a large class of PDE problems as demonstrated in the examples in Section 4. Use of these methods thus can be recommended only if an extremely high accuracy is required in critical applications. Gauss methods, which feature significantly faster performance, yet provide the errors almost as small as the analytical methods do, and certainly smaller than the conventional time discretization techniques, may be a reasonable compromise.

5. Discussion and conclusions

The current paper presents a novel methodology for a solution of Partial Differential Equations, in which the boundary constraints are removed from the solution state and are embedded into the equation dynamics via an analytical transformation that transforms a PDE into an equivalent PIE representation. Unlike in a PDE, whose solution does not necessarily satisfy the boundary conditions and needs to be specifically adjusted in order to fit the constraints, any solution of a PIE (together with a well-defined inverse PIE-PDE solution map) automatically satisfies the boundary conditions by construction! This property of the PIE representation allows one to minimize the errors associated with the imposition of the boundary conditions during the solution procedure and develop automated and generalizable methodologies for solution of a large class of PDE systems irrespective of the nature of the boundary conditions.

While the presented framework is certainly appealing from the theoretical perspective, the current paper seeks to assess what are the specific advantages of this framework, if any, as applied to a numerical solution of PDEs, and how it compares with the existing direct PDE solution methods. For this purpose, we choose to discretize the PIE equation with Galerkin projection techniques employing Chebyshev polynomials of the first kind as the basis functions. This allows for a comparison of the developed PIE methodology with a direct PDE solution procedure that employs classical Chebyshev-tau methods [8,14,59].

Comparison of the PIE and PDE solution properties demonstrates several specific advantages of the PIE methodology: (1) With PIEs, an analytical integration in time of the resulting ODE system for the function expansion coefficients is possible, which allows one to obtain an unprecedentedly low error, close to machine precision, while solving unsteady PDE equations with generic boundary conditions. This, to the authors' knowledge, so far has not been possible with other techniques. The possibility of an analytical integration is due to the fact that the matrix operator in the ODE with PIEs results from the integral operators and is of full rank, while in a PDE solution it results from the differential operators and is of reduced rank. The ill-conditioning of the ODE matrix operator with the PDE approaches precludes an analytical evaluation of the temporal behavior of the solution. (2) PIE methodology shows clear advantages in regard to a treatment of the outflow boundary conditions, which are known to be a bottleneck in development of stable, accurate and conservative methods for solution of hyperbolic and convection-dominated problems [76,77]. (3) Due to an existence of a well-defined analytical PDE-to-PIE conversion procedure, the entire PDE-PIE solution process with its benefits of enabling a strong enforcement of the boundary conditions through the PIE framework can be automated. The methodology was implemented in a newly developed software PIESIM, which emerges as a new generalized high-order PDE solver capable of

providing robust and accurate solutions to a large class of PDE problems with no user intervention apart from declaring an original PDE problem.

The advantages of PIEs, especially in a possibility of obtaining exceptionally accurate solutions featuring a near zero error, come with certain limitations. For example, an analytical integration in time, employed using a symbolic integration capability of MATLAB, is fairly slow, and can only be recommended if the size of the problem is relatively small. In this respect, a numerical approximation of the exact time integral in Eq. (3.39) with a high-accurate Gauss quadrature can be recommended as a viable alternative. Gauss integration is significantly faster than an analytical integration, and nonetheless yields several orders of magnitude lower errors than the conventional discrete time stepping techniques, as demonstrated in numerical examples presented in this paper. We remark that the aforementioned Gauss integration is not possible with direct PDE solvers, since representation of the boundary inputs in the equation dynamics is not explicitly defined, as in a PIE formulation. Gauss integration enabled by the PIE methodology emerges as an interesting alternative to conventional, sequential, time stepping algorithms and potentially offers a possibility of developing efficient time-parallel techniques for PDE solutions [70,71]. Future work shall be devoted to further understanding of the properties of the Gauss integration, its accuracy with respect to a choice of the quadrature points, and the influence of the structure of the matrix operators on the result, as well as further improvements in computational efficiency. We remark that solution of both PDE and PIE problems with conventional time stepping approaches results in a similar computational efficiency. Another limitation of the PIE methodology is that it currently relies on the new *opvar* class of objects for constructing and manipulating the PI operators [24,73], which is presently tied to MATLAB. Portability and extension to other programming languages will be addressed in a future work.

Analytical treatment of boundary conditions through embedding their influence on the physical processes into the equation dynamics is useful not only for developing advanced PDE solution techniques, but also for stability analysis and control of PDEs, previously obfuscated by the need to enforce boundary constraints on the PDE solution state [24,63]. Thus, the developed numerical discretization procedure of PIE systems is also useful for designing and simulating real-time controllers for physical systems governed by PDE models [55,56]. Future work will involve extension of the numerical discretization methodology based on PIEs to problems with multiple spatial dimensions and to nonlinear problems. In fact, some preliminary work regarding these topics is already on the way [80,81]. As an outlook and the future perspectives of the developed methodology, we envision that PIE representation may be especially useful in developing accurate and stable numerical techniques that involve coupling of physical processes across the interfaces, such as in domain decomposition methods [82,83] and in multi-physics problems, e.g., fluid–structure interaction [79,84]. Available numerical techniques for these problems often suffer from instabilities related to numerical difficulties associated with the enforcement of continuity of variables across interfaces [82,84,85]. PIE methodology, which can address this problem at a continuous level, i.e. prior to discretization, may be beneficial. This hypothesis will be further explored in the future work.

To conclude, the most important result of the presented research is the verified advantage in accuracy of the developed PDE-PIE solution methodology as applied to linear unsteady PDEs with variable coefficients. The existing issues related to the difficulties of imposing boundary conditions in convection-dominated and multi-physics problems can be effectively solved by the presented methodology. By providing robust and accurate solutions to a wide class of PDE equation models in a generalized and automated manner, the developed methodology will enable solutions to pressing technological problems by offering advanced simulation, analysis and control capabilities for a wide range of applications in engineering and science.

Data availability

Data will be made available on request.

Acknowledgments

This work has been supported by the National Science Foundation grant numbers NSF CMMI-1935453 and NSF CAREER CBET-1944568.

Appendix A. Definition of 3-PI operators in the PIE representation

This appendix gives a definition of the functions $G_i(x,s), i = 0 \dots 5$, appearing in the composition of 3-PI operators in (2.15).

$$
\begin{aligned}
G_0 &= \begin{bmatrix} I_{n_0} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, G_1(x,s) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_{n_1} & 0 \\ 0 & 0 & (x-s)I_{n_2} \end{bmatrix}, \\
G_2(x,s) &= -K(x)B_T^{-1}BQ(s), \\
G_3 &= \begin{bmatrix} 0 & I_{n_1} & 0 \\ 0 & 0 & 0 \end{bmatrix}, G_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & I_{n_2} \end{bmatrix}, \\
G_5(s) &= -VB_T^{-1}BQ(s), \\
T &= \begin{bmatrix} I_{n_1} & 0 & 0 \\ I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & I_{n_2} & (b-a)I_{n_2} \\ 0 & 0 & I_{n_2} \\ 0 & 0 & I_{n_2} \end{bmatrix}, Q(s) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_{n_1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (b-s)I_{n_2} \\ 0 & 0 & 0 \\ 0 & 0 & I_{n_2} \end{bmatrix},
\end{aligned}
\tag{A.1}
$$

$$K(x) = \begin{bmatrix} 0 & 0 & 0 \\ I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & (x-a)I_{n_2} \end{bmatrix}, V = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & I_{n_2} \end{bmatrix},$$

with matrix B_T defined in (2.16).

Appendix B. Proof of Lemma 3.1

Proof.

1. To prove the first case: if $\mathcal{T}_{mn}$ is such that $m \leq n_0$, according to the structure of G_0, G_1 and G_2, it must have the form $\mathcal{T}_{mn} = \mathcal{P}_{\{\delta_{mn},0,0\}}$, and thus it is easily computed that $\mathcal{T}_{mn}T_k(x) = \delta_{mn}T_k(x)$.
2. To prove the second case, we first need to recall some useful recursive relations for Chebyshev polynomials [8,64]:

$$\int T_k(x)\,dx = \begin{cases} T_1(x) + C_0, & k = 0 \\ \frac{1}{4}\left[T_0(x) + T_2(x)\right] + C_1, & k = 1 \\ \frac{1}{2}\left[\frac{T_{k+1}(x)}{k+1} - \frac{T_{k-1}(x)}{k-1}\right] + C_k, & k \geq 2 \end{cases} \tag{B.1}$$

$$x\,T_k(x) = \begin{cases} T_1(x), & k = 0 \\ \frac{1}{2}\left[T_{k-1}(x) + T_{k+1}(x)\right], & k \geq 1 \end{cases} \tag{B.2}$$

Let us now consider $\mathcal{T}_{mn}$ such that $n_0 < m \leq n_0 + n_1$. According to the structure of G_0, G_1 and G_2, it has the form of $\mathcal{T}_{mn} = \mathcal{P}_{\{0,\delta_{mn},G_{2mn}\}}$, such that

$$\mathcal{P}_{\{0,\delta_{mn},G_{2mn}\}}T_k(x) = \delta_{mn}\int_{-1}^{x} T_k(s)\,d\,s + \int_{-1}^{1} G_{2mn}(x,s)T_k(s)\,d\,s. \tag{B.3}$$

The first integral in the right-hand side can be evaluated according to (B.1). Let us now consider the second integral. According to the composition of the operator G_2, its general entry G_{2mn} would be of the form $G_{2mn} = \beta_{0mn} + \beta_{1mn}\,s + \beta_{2mn}\,x + \beta_{3mn}\,xs$, where $\beta_{jmn}, j = 0 \ldots 3$, are some real constants. Taking an integral yields

$$\begin{aligned} \int_{-1}^{1} G_{2mn}(x,s)T_k(s)\,d\,s &= \int_{-1}^{1} \left(\beta_{0mn} + \beta_{1mn}\,s + \beta_{2mn}\,x + \beta_{3mn}\,xs\right) T_k(s)\,d\,s \\ &= \int_{-1}^{1} \left(\beta_{0mn} + \beta_{1mn}\,s\right) T_k(s)\,d\,s + x\int_{-1}^{1} \left(\beta_{2mn} + \beta_{3mn}\,s\right) T_k(s)\,d\,s. \end{aligned} \tag{B.4}$$

The two integrals in (B.4) evaluate to $\gamma_{jkmn}T_0(x)$, due to the constant limits of integration, where γ_{jkmn}, $j = 0, 1$, are some real constants for the first and second integrals, respectively. The multiplication by x in the second integral produces the result $x \cdot \gamma_{1kmn}T_0(x) = \gamma_{1kmn}T_1(x)$. Combining the two integral contributions, (B.3) can be rewritten as

$$\mathcal{P}_{\{0,\delta_{mn},G_{2mn}\}}T_k(x) = \gamma_{0kmn}T_0(x) + \gamma_{1kmn}T_1(x) \tag{B.5}$$

$$+ \delta_{mn}\begin{cases} T_1(x) - T_1(-1), & k = 0 \\ \frac{1}{4}\left[T_0(x) + T_2(x)\right] - \frac{1}{4}\left[T_0(-1) + T_2(-1)\right], & k = 1 \\ \frac{1}{2}\left[\frac{T_{k+1}(x)}{k+1} - \frac{T_{k-1}(x)}{k-1}\right] - \frac{1}{2}\left[\frac{T_{k+1}(-1)}{k+1} - \frac{T_{k-1}(-1)}{k-1}\right], & k \geq 2 \end{cases} \tag{B.6}$$

$$= b^{(1)}_{0\,kmn}T_0(x) + b^{(1)}_{1kmn}T_1(x) + \delta_{mn}\begin{cases} \frac{1}{2}\left[\frac{T_{k+1}(x)}{k+1}\right], & k = 1, 2 \\ \frac{1}{2}\left[\frac{T_{k+1}(x)}{k+1} - \frac{T_{k-1}(x)}{k-1}\right], & k \geq 3, \end{cases} \tag{B.7}$$

since $T_k(-1) = (-1)^k = (-1)^k T_0(x)$, leading to (3.9), (3.10).
3. For the third case, we have that $\mathcal{T}_{mn}$, $m > n_0 + n_1$ has the form of $\mathcal{T}_{mn} = \mathcal{P}_{\{0,\delta_{mn}(x-s),G_{2mn}\}}$ and

$$\mathcal{P}_{\{0,\delta_{mn}(x-s),G_{2mn}\}}T_k(x) = \delta_{mn}\int_{-1}^{x} (x-s)T_k(s)\,d\,s + \int_{-1}^{1} G_{2mn}(x,s)T_k(s)\,d\,s. \tag{B.8}$$

The last integral in Eq. (B.8) is evaluated analogously to the previous case. The first integral yields

$$\int_{-1}^{x} (x-s)T_k(s)\,d\,s = x\int_{-1}^{x} T_k(s)\,d\,s - \int_{-1}^{x} s\,T_k(s)\,d\,s. \tag{B.9}$$

Considering the first contribution, we have

$$
\begin{aligned}
&x\int_{-1}^{x} T_k(s)\,d\,s \\
&\quad= x\begin{cases} T_1(x)-T_1(-1), & k=0 \\ \frac{1}{4}\left[T_0(x)+T_2(x)\right]-\frac{1}{4}\left[T_0(-1)+T_2(-1)\right], & k=1 \\ \frac{1}{2}\left[\frac{T_{k+1}(x)}{k+1}-\frac{T_{k-1}(x)}{k-1}\right]-\frac{1}{2}\left[\frac{T_{k+1}(-1)}{k+1}-\frac{T_{k-1}(-1)}{k-1}\right], & k\geq 2 \end{cases} \\
&\quad= \tilde{\alpha}_{1k}T_1(x)+x\begin{cases} T_1(x), & k=0 \\ \frac{1}{4}\left[T_0(x)+T_2(x)\right], & k=1 \\ \frac{1}{2}\left[\frac{T_{k+1}(x)}{k+1}-\frac{T_{k-1}(x)}{k-1}\right], & k\geq 2, \end{cases} \\
&\quad= \tilde{\alpha}_{1k}T_1(x)+\begin{cases} \frac{1}{2}\left[T_0(x)+T_2(x)\right], & k=0 \\ \frac{1}{4}\left[T_1(x)+\frac{1}{2}\left[T_1(x)+T_3(x)\right]\right], & k=1 \\ \frac{1}{2}\left[\frac{\frac{1}{2}[T_k(x)+T_{k+2}(x)]}{k+1}-\frac{\frac{1}{2}[T_{k-2}(x)+T_k(x)](x)}{k-1}\right], & k\geq 2, \end{cases} \\
&\quad= \tilde{\tilde{\alpha}}_{0k}T_0(x)+\tilde{\tilde{\alpha}}_{1k}T_1(x)+\begin{cases} \frac{1}{2}\left[\frac{T_{k+2}(x)}{k+1}\right], & k=0 \\ \frac{1}{4}\left[\frac{T_{k+2}(x)}{k+1}\right], & k=1 \\ \frac{1}{4}\left[\frac{T_{k+2}(x)}{k+1}-\frac{2T_k(x)}{k^2-1}\right], & k=2,3 \\ \frac{1}{4}\left[\frac{T_{k+2}(x)}{k+1}-\frac{2T_k(x)}{k^2-1}-\frac{T_{k-2}(x)}{k-1}\right], & k\geq 4. \end{cases}
\end{aligned}
\tag{B.10}
$$

Considering the second contribution, we have

$$
\begin{aligned}
&-\int_{-1}^{x} s\,T_k(s)\,d\,s=-\int_{-1}^{x} ds\begin{cases} T_1(s), & k=0 \\ \frac{1}{2}\left[T_{k-1}(s)+T_{k+1}(s)\right], & k\geq 1 \end{cases} \\
&\quad= \tilde{\beta}_{0k}T_0(x)-\begin{cases} \frac{1}{4}\left[T_0(x)+T_2(x)\right], & k=0 \\ \frac{1}{2}T_1(x)+\frac{1}{4}\left[\frac{T_3(x)}{3}-T_1(x)\right], & k=1 \\ \frac{1}{8}\left[T_0(x)+T_2(x)\right]+\frac{1}{4}\left[\frac{T_4(x)}{4}-\frac{T_2(x)}{2}\right], & k=2 \\ \frac{1}{4}\left[\frac{T_k(x)}{k}-\frac{T_{k-2}(x)}{k-2}\right]+\frac{1}{4}\left[\frac{T_{k+2}(x)}{k+2}-\frac{T_k(x)}{k}\right], & k\geq 3 \end{cases} \\
&\quad= \tilde{\tilde{\beta}}_{0k}T_0(x)+\tilde{\tilde{\beta}}_{1k}T_1(x)-\begin{cases} \frac{1}{2}\left[\frac{T_{k+2}(x)}{k+2}\right] & k=0 \\ \frac{1}{4}\left[\frac{T_{k+2}(x)}{k+2}\right], & 1\leq k\leq 3 \\ \frac{1}{4}\left[\frac{T_{k+2}(x)}{k+2}-\frac{T_{k-2}(x)}{k-2}\right], & k\geq 4 \end{cases}
\end{aligned}
\tag{B.11}
$$

Combining Eqs. (B.4), (B.8), (B.10) and (B.11) yields Eq. (3.11) with (3.12).

Dependence of the constants $b^{(i)}_{jkmn}$, $i=1,2$, $j=0,1$, on the boundary conditions comes from the dependence of the operator entries G_{2mn} on the boundary conditions defined by the matrix B. This concludes the proof. □

References

[1] H. Brezis, F. Browder, Partial differential equations in the 20th century, Adv. Math. 135 (1) (1998) 76–144.
[2] A.N. Tikhonov, A.A. Samarskii, Equations of Mathematical Physics, Dover Publications, 2011.
[3] J.D. Logan, Applied Partial Differential Equations, Springer, 2015.
[4] R.C. McOwen, Partial Differential Equations: Methods and Applications, Prentice Hall, 2003.
[5] J. Nordström, A roadmap to well-posed and stable problems in computational physis, J. Sci. Comput. 71 (2017) 365–385.
[6] M.O. Deville, P.F. Fischer, E.H. Mund, High-Order Methods for Incompressible Fluid Flow, Cambridge University Press, Cambridge, UK, 2002.
[7] V. Grigoryan, Partial differential equations, 2010, web.math.ucsb.edu/~grigoryan/124A.pdf.
[8] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer–Verlag, 1988.
[9] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM Press, 1977.
[10] J. Shen, Efficient spectral-Galerkin method I. Direct solvers for the second and fourth order equations using Legendre polynomials, SIAM J. Numer. Anal. 15 (6) (1994) 1489–1505.

[11] J. Shen, A new dual-Petrov-Galerkin method for third and higher odd-order differential equations: application to the KdV equation, SIAM J. Numer. Anal. 41 (5) (2003) 1595–1619.
[12] D.B. Haidvogel, T. Zang, The accurate solution of Poisson's equation by expansion in Chebyshev polynomials, J. Comput. Phys. 30 (2) (1979) 167–180.
[13] C. Canuto, Boundary conditions in Chebyshev and Legendre methods, SIAM J. Numer. Anal. 23 (4) (1986) 815–831.
[14] H.I. Siyyam, M.I. Syam, An accurate solution of the Poisson equation by the Chebyshev-Tau method, J. Comput. Appl. Math. 85 (1) (1997) 1–10.
[15] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, in: Abhandlungen Aus Dem Mathematischen Seminar Der Universität Hamburg, Vol. 36, Springer, 1971, pp. 9–15.
[16] Y. Bazilevs, T.J.R. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, Comput. & Fluids 36 (1) (2007) 12–26.
[17] V. Jovanovic, S. Koshkin, The Ritz method for boundary problems with essential conditions as constraints, Adv. Math. Phys. 2016 (2016).
[18] B.-Y. Guo, J. Shen, L.-L. Wang, Generalized Jacobi polynomials/functions and their applications, Appl. Numer. Math. 59 (5) (2009) 1011–1028.
[19] X. Yu, Z. Wang, H. Li, Jacobi-Sobolev orthogonal polynomials and spectral methods for elliptic boundary value problems, Commun. Appl. Math. Comput. 1 (2019) 283.
[20] P.F. Fischer, An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations, J. Comput. Phys. 133 (1) (1997) 84–101.
[21] G. Karniadakis, S.J. Sherwin, Spectral/Hp Element Methods for Computational Fluid Dynamics, Oxford University Press, USA, 2005.
[22] A. Smyshlyaev, M. Krstic, Backstepping observers for a class of parabolic PDEs, Systems Control Lett. 54 (7) (2005) 613–625.
[23] E. Fridman, Y. Orlov, An LMI approach to H^∞ boundary control of semilinear parabolic and hyperbolic systems, Automatica 45 (9) (2009) 2060–2066.
[24] M.M. Peet, A partial integral equation (PIE) representation of coupled linear PDEs and scalable stability analysis using LMIs, Automatica 125 (2021) 109473, 1–14.
[25] D. Lehotzky, T. Insperger, A pseudospectral tau approximation for time delay systems and its comparison with other weighted-residual-type methods, Internat. J. Numer. Methods Engrg. 108 (6) (2016) 588–613.
[26] E. Boström, Boundary Conditions for Spectral Simulations of Atmospheric Boundary Layers (Ph.D. thesis), KTH Royal Institute of Technology, 2017.
[27] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, E. Rank, Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method, Internat. J. Numer. Methods Engrg. 95 (10) (2013) 811–846.
[28] M. Vymazal, D. Moxey, C.D. Cantwell, S.J. Sherwin, R.M. Kirby, On weak Dirichlet boundary conditions for elliptic problems in the continuous Galerkin method, J. Comput. Phys. 394 (2019) 732–744.
[29] J. Freund, R. Stenberg, On weakly imposed boundary conditions for second order problems, in: Proceedings of the Ninth Int. Conf. Finite Elements in Fluids, Venice, 1995, pp. 327–336.
[30] M. Juntunen, R. Stenberg, Nitsche's method for general boundary conditions, Math. Comp. 78 (267) (2009) 1353–1374.
[31] R.J. LeVeque, Numerical Methods for Conservation Laws, Vol. 214, Springer, 1992.
[32] E. Tadmor, A review of numerical methods for nonlinear partial differential equations, Bull. Amer. Math. Soc. 49 (4) (2012) 507–554.
[33] H. Bansal, S. Weiland, L. Iapichino, W.H.A. Schilders, N. van de Wouw, Structure-preserving spatial discretization of a Two-Fluid model, in: IEEE-CDC, 2020, pp. 5062–5067.
[34] I. Stakgold, M.J. Holst, Green's Functions and Boundary Value Problems, John Wiley & Sons, 1979.
[35] G.F. Roach, Green's Functions, second ed., Cambridge University Press, Cambridge, Great Britain, 1982.
[36] Green's function library, 2020, http://www.greensfunction.unl.edu/home/index.html.
[37] K.E. Atkinson, The numerical solution of boundary integral equations, in: Institute of Mathematics and Its Applications Conference Series, Vol. 63, Oxford University Press, 1997, pp. 223–260.
[38] O. Marin, K. Gustavsson, A.-K. Tornberg, A highly accurate boundary treatment for confined Stokes flow, Comput. & Fluids 66 (2012) 215–230.
[39] C. Carvalho, S. Khatri, A.D. Kim, Asymptotic approximations for the close evaluation of double-layer potentials, SIAM J. Sci. Comput. 42 (1) (2020) A504–A533.
[40] Fundamental solution, in: Encyclopaedia of Mathematics, Kluwer, 1994.
[41] P. Kythe, Fundamental Solutions for Differential Operators and Applications, Springer Science & Business Media, 2012.
[42] S.G. Johnson, Notes on Green's functions in inhomogeneous media, 2010, math.mit.edu/~stevenj/18.303/inhomog-notes.pdf.
[43] D.-J. van Manen, J.O.A. Robertsson, A. Curtis, Modeling of wave propagation in inhomogeneous media, Phys. Rev. Lett. 94 (16) (2005) 164301.
[44] F.J. Sánchez-Sesma, R. Madariaga, K. Irikura, An approximate elastic two-dimensional Green's function for a constant-gradient medium, Geophys. J. Int. 146 (1) (2001) 237–248.
[45] L. Greengard, Spectral integration and two-point boundary value problems, SIAM J. Numer. Anal. 28 (4) (1991) 1071–1080.
[46] M. Hiegemann, Chebyshev matrix operator method for the solution of integrated forms of linear ordinary differential equations, Acta Mech. 122 (1997) 231–242.
[47] T.A. Driscoll, Automatic spectral collocation for integral, integro-differential, and integrally reformulated differential equations, J. Comput. Phys. 229 (17) (2010) 5980–5998.
[48] A.S. Fokas, A unified transform method for solving linear and certain nonlinear PDEs, Proc. R. Soc. Lond. Ser. A 453 (1997) 1411–1443.
[49] A.S. Fokas, Lax pairs and a new spectral method for linear and integrable nonlinear PDEs, Selecta Math. 4 (1998) 31–68.
[50] P.A. Treharne, A.S. Fokas, Initial-boundary value problems for linear PDEs with variable coefficients, Math. Proc. Cambridge Philos. Soc. 143 (2007) 221–242.
[51] B. Deconinck, T. Trogdon, V. Vasan, The method of Fokas for solving linear partial differential equations, SIAM Rev. 56 (1) (2014) 159–186.
[52] E. Kesici, B. Pelloni, T. Pryer, D. Smith, A numerical implementation of the unified Fokas transform for evolution problems on a finite interval, EJAM 29 (3) (2018) 543–567.
[53] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, SIAM, 2004.
[54] P.D. Lax, R.D. Richtmyer, Survey of the stability of linear finite difference equations, Comm. Pure Appl. Math. 9 (2) (1956) 267–293.
[55] S. Shivakumar, A. Das, S. Weiland, M.M. Peet, Duality and H_∞ optimal control of coupled ODE-PDE systems, in: 59th Conference on Decision in Control (CDC), 2020.
[56] A. Das, S. Shivakumar, S. Weiland, M.M. Peet, H_∞ optimal estimation for linear coupled PDE systems, in: 2019 IEEE 58th Conference on Decision and Control (CDC), IEEE, 2019, pp. 262–267.
[57] C. Edwards, Y. Peet, Linear stability of plane Poiseuille flow in the sense of Lyapunov, in: Proceedings of 62nd IEEE Conference on Decision and Control, Marina Bay Sands, Singapore, 2023.
[58] M.M. Peet, Representation of networks and systems with delay: DDEs, DDFs, ODE–PDEs and PIEs, Automatica 127 (2021) 109508.
[59] L.N. Trefethen, Approximation Theory and Approximation Practice, SIAM, 2013.
[60] M.M. Peet, A new state-space representation for coupled PDEs and scalable Lyapunov stability analysis in the SOS framework, in: Proc. IEEE Conf. on Decision and Control, 2018.
[61] M.M. Peet, A partial integral equation (PIE) representation of coupled linear PDEs and scalable stability analysis using LMIs, 2018, arxiv.org/abs/1812.06794.
[62] D.A. Smith, Well-posed two-point initial-boundary value problems with arbitrary boundary conditions, Math. Proc. Cambridge Philos. Soc. 152 (2012) 473–496.

[63] S. Shivakumar, A. Das, S. Weiland, M. Peet, Extension of the Partial Integral Equation representation to GPDE input-output systems, in: Transactions on Automatic Control, 2023, submitted for publication.
[64] P. Moin, Fundamentals of Engineering Numerical Analysis, Cambridge University Press, 2001.
[65] N. Bressan, A. Quarteroni, Analysis of Chebyshev collocation methods for parabolic equations, SIAM J. Numer. Anal. 23 (6) (1986) 1138–1154.
[66] C. Canuto, A. Quarteroni, Error estimates for spectral and pseudospectral approximations of hyperbolic equations, SIAM J. Numer. Anal. 19 (3) (1982) 629–642.
[67] E. Tadmor, Spectral Methods for Hyperbolic Problems, in: Lecture Notes Delivered at Ecole Des Ondes, Inria-Rocquencort, France, 1994.
[68] G.F. Dullerud, F. Paganini, A Course in Robust Control Theory, Springer–Verlag, 2000.
[69] R.L. Williams II., D.A. Lawrence, Linear State-Space Control Systems, John Wiley & Sons, Inc., 2007.
[70] M. Emmett, M. Minion, Toward an efficient parallel in time method for partial differential equations, Commun. Appl. Math. Comput. Sci. 7 (1) (2012) 105–132.
[71] B.W. Ong, J.B. Schroder, Applications of time parallelization, Comput. Vis. Sci. 23 (2020) 1–15.
[72] Y.T. Peet, P.F. Fischer, Stability analysis of interface temporal discretization in grid overlapping methods, SIAM J. Numer. Anal. 50 (6) (2012) 3375–3401.
[73] S. Shivakumar, A. Das, M.M. Peet, PIETOOLS: A MATLAB toolbox for manipulation and optimization of partial integral operators, in: Proceedings of 2020 American Control Conference (ACC), Denver, CO, USA, 2020, pp. 2667–2672.
[74] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, Vol. 61, SIAM, 1998.
[75] E. Volterra, E. Zachmanoglou, Dynamics of Vibrations, Charles E. Merrill Books, 1965.
[76] D.H. Rudy, J.C. Strikwerda, A nonreflecting outflow boundary condition for subsonic Navier-Stokes calculations, J. Comput. Phys. 36 (1) (1980) 55–70.
[77] T. Colonius, Modeling artificial boundary conditions for compressible flow, Annu. Rev. Fluid Mech. 36 (2004) 315–345.
[78] N.M. Newmark, A method of computation for structural dynamics, J. Eng. Mech. Div. 85 (3) (1959) 67–94.
[79] Y. Xu, Y.T. Peet, Verification and convergence study of a spectral-element numerical methodology for fluid-structure interaction, J. Comput. Phys.: X 10 (2021) 100084.
[80] D.S. Jagt, M.M. Peet, A PIE representation of coupled linear 2D PDEs and stabiity analysis using LPIs, in: Proc. 2022 American Control Conference (ACC), Atlanta, USA, 2022.
[81] D. Jagt, P. Seiler, M. Peet, A PIE representation of scalar quadratic PDEs and global stability analysis using SDP, in: Proc. IEEE Conference on Decision Control, 2023.
[82] C.N. Dawson, Q. Du, T.F. Dupont, A finite difference domain decomposition algorithm for numerical solution of the heat equation, Math. Comp. 57 (195) (1991) 63–71.
[83] B. Smith, P. Bjørstad, W. Gropp, Domain Decomposition: Parallel Methods for Elliptic Partial Differential Equations, Cambridge University Press, 2004.
[84] C. Förster, W.A. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, Comput. Methods Appl. Mech. Engrg. 196 (7) (2007) 1278–1293.
[85] M.A. Fernández, J. Mullaert, M. Vidrascu, Generalized Robin–Neumann explicit coupling schemes for incompressible fluid-structure interaction: Stability analysis and numerics, Internat. J. Numer. Methods Engrg. 101 (3) (2015) 199–229.