

# Variational Formulations for Explicit Runge-Kutta Methods

Judit Muñoz-Matute<sup>a</sup>, David Pardo<sup>a,b,c</sup>, Victor M. Calo<sup>d,e</sup>,  
Elisabete Alberdi<sup>a</sup>

<sup>a</sup>*University of the Basque Country (UPV/EHU), Leioa, Spain*

<sup>b</sup>*BCAM-Basque Center for Applied Mathematics, Bilbao, Spain*

<sup>c</sup>*IKERBASQUE, Basque Foundation for Science, Bilbao, Spain*

<sup>d</sup>*Applied Geology, Western Australian School of Mines, Faculty of Science and Engineering, Curtin University, Perth, WA, Australia 6845*

<sup>e</sup>*Mineral Resources, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Kensington, WA, Australia 6152*

---

## Abstract

Variational space-time formulations for partial differential equations have been of great interest in the last decades, among other things, because they allow to develop mesh-adaptive algorithms. Since it is known that implicit time marching schemes have variational structure, they are often employed for adaptivity. Previously, Galerkin formulations of explicit methods were introduced for ordinary differential equations employing specific inexact quadrature rules. In this work, we prove that the explicit Runge-Kutta methods can be expressed as discontinuous-in-time Petrov-Galerkin methods for the linear diffusion equation. We systematically build trial and test functions that, after exact integration in time, lead to one, two, and general stage explicit Runge-Kutta methods. This approach enables us to reproduce the existing time-domain (goal-oriented) adaptive algorithms using explicit methods in time.

*Keywords:* linear diffusion equation, discontinuous Petrov-Galerkin formulations, dynamic meshes, Runge-Kutta methods

---

## 1. Introduction

Adaptive algorithms for partial differential equations (PDEs) produce optimal grids that seek to minimize the computational cost. For time dependent problems, it is common to employ time-marching schemes and adapt the time step size and/or the spatial mesh size employing a posteriori error

estimates [11, 12, 38]. Alternatively, there exist adaptive strategies based on space-time finite element methods (FEM). In some algorithms, the authors assume a space-time tensor-product of the trial and test functions. Then, employing discontinuous-in-time basis functions, the space-time FEM can be reinterpreted as a time-marching scheme [43, 45]. Here, the approximation orders (in space and time) as well as the mesh size and the time-step size can be adapted [46, 50]. That is, space-time FEM can build unstructured space-time meshes [1, 2, 19, 36].

To perform goal-oriented adaptivity [6, 7, 10, 44, 47, 51], we require a space-time variational formulation of the problem. In this kind of algorithms, we represent the error in the quantity of interest as an integral over the whole space-time domain that is subsequently expressed as a sum of local element contributions, which we use for adaptivity. A full space-time variational formulation allows such representation [17]. However, the most used space-time variational formulations for PDEs lead to implicit methods in time when they are thought as time marching schemes. For that reason, adaptive strategies based in space-time FEM as well as time-domain goal-oriented adaptive processes employ implicit methods in time.

Our focus is to design goal-oriented adaptive algorithms employing explicit methods in time since in many instances they can be computationally cheaper than implicit ones. Nevertheless, explicit methods are conditionally stable and the Courant-Friedrichs-Lewy (CFL) condition must be satisfied to ensure stability. To cope with this, in [37] we proposed an explicit-in-time goal-oriented adaptive algorithm employing the Forward Euler method that adapts locally the time grid based on the CFL condition.

To solve time-dependent PDEs, we commonly discretize independently the spatial and temporal variables (also called semidiscretization or method of lines) [42]. First, the spatial variable is discretized by the finite element method to obtain a system of ODEs. The resulting system is subsequently solved employing time stepping schemes. The alternative idea of using variational space-time methods was well established in the late eighties and early nineties [4, 20, 25, 34]. Hughes and Hulbert [30, 31] proposed a stabilized space-time FEM for hyperbolic problems. They showed that the oscillations present in the solutions were considerably reduced by employing space-time variational formulations rather than using semidiscretizations. Nowadays, it is well known that some low-order space-time FEM are algebraically equivalent to some semidiscretizations [5]. For example, the discontinuous Galerkin method using constant functions in time (usually denoted by dG(0)) leads to the Backward Euler method. The continuous Petrov-Galerkin method in time with linear trial and constant test functions (denoted cGP(1)) is

equivalent to the Crank-Nicholson method. Recently, higher order dG(k)- and cGP(k)-methods have been developed and analyzed for parabolic and hyperbolic problems [3, 21, 33, 35].

There also exist variational formulations of time marching schemes in the context of Ordinary Differential Equations (ODEs) [8]. Delfour et. al. [14, 15] and Hulme [32] showed that it is possible to obtain classical schemes like Runge-Kutta methods by employing Galerkin methods for initial value problems together with quadrature formulas. Estep and French [22, 23] derived error bounds for the continuous and discontinuous Galerkin methods to efficiently adapt the time step size. More recently, Estep and Stuart studied the dynamical behavior of discontinuous Galerkin methods for ODEs in [24] and Tang et. al. provided in [48] a unified framework of finite element methods in time. In [26, 48], the variational formulation of symplectic time integrators are studied for Hamiltonian systems. There also exist some works describing variational formulations of explicit methods for ODEs. In [52], the authors derived some low order explicit Runge-Kutta methods by selecting specific quadrature rules and test functions. However, they claimed that with the formulation they proposed, it is not possible to reproduce second-order Runge-Kutta methods with null weights in the Butcher tableau ( $b_j = 0$ ). In [13], Collins et. al. also proposed a variational formulation for explicit schemes by inserting some operators in the equation together with quadrature rules. They also derived an a posteriori error estimation of explicit schemes in the goal-oriented approach. In this case, the errors committed by using the operators and the quadratures must be included in the error representation. In both settings [13, 52], it is possible to recover some explicit Runge-Kutta methods but only on a case-by-case basis and not in a constructive way.

In this work, we propose a constructive method to derive variational formulations of explicit Runge-Kutta schemes of any stage. We apply it to parabolic problems, but its extension to other linear and some nonlinear problems is straightforward. Given a Butcher tableau, we systematically build the corresponding trial and test basis functions integrating analytically in time. In particular, we characterize all the second-order and two-stage explicit Runge-Kutta methods. As we fully define the trial and test spaces, we can naturally represent the error in the quantity of interest in the same way as for implicit methods [6, 44]. Such error representation allows us to design explicit-in-time (and consequently cheaper) goal-oriented adaptive algorithms. The presented formulation is also useful to build new time-stepping schemes of Runge-Kutta type or more general ones, and also to extend the existing space discretizations like IGA [27, 29, 49], DPG [16] and

Trefftz [18], to time domain problems.

First, we derive a discontinuous-in-time Petrov-Galerkin formulation of the problem that allows us to use discontinuous trial and test functions in time. In order to obtain an explicit method, we define the jumps of the solution using a downwind approximation across each time interface instead of an upwind approximation as in classical discontinuous Galerkin methods in time. We prove that by selecting piecewise-constant trial and test functions, we recover the Forward Euler method. This construction supports dynamic meshes in space, i.e, we allow different spatial discretizations per time interval. In order to obtain square mass matrices, we displace in time the spatial discrete spaces of the test space with respect to the trial space. This displacement leads to a Petrov-Galerkin method both in space and time.

For a general number of Runge-Kutta stages, we define families of piecewise polynomials that are discontinuous-in-time for both trial and test spaces. By substituting them into the variational formulation and treating the coefficients of the polynomials as unknowns, we obtain integrals of the products of trial and test functions. Then, we establish some conditions that these time integrals must satisfy. First, we state the necessary orthogonality conditions needed to obtain an explicit method. We also define non-orthogonality conditions by matching the remaining integrals with the entries of the Butcher's tableau that define the Runge-Kutta methods. Finally, performing analytic integration, we obtain a system of nonlinear equations. By solving this system, we obtain the coefficients of the trial and test functions for any stage Runge-Kutta method. We define the corresponding trial and test spaces as the span of these functions. However, for a large number of stages (above 5) the system becomes hard to solve.

This article is organized as follows. Section 2 describes the strong and weak formulations of the linear diffusion equation we use to develop the theory. In Section 3, we derive a discontinuous-in-time Petrov-Galerkin formulation of the problem. In Section 4, we build the trial and test functions for the one, two, and general  $s$ -stage Runge-Kutta methods, providing some examples. Expert readers may directly go to Section 4.3, while those interested on low-order methods only, may have a more accessible presentation by reading Sections 4.1 and 4.2. In Section 5, we explain how to apply our constructions to goal-oriented error estimation. Section 6 describes the conclusions and the possible extensions of this work. Finally, in Appendix A, we express in matrix form the nonlinear system of equations we need to solve to obtain any explicit Runge-Kutta method. Appendix B provides a MATLAB code to solve it.

## 2. Model Problem

In this section, we state both the strong and weak formulations of the model problem we employ to develop the discontinuous-in-time Petrov-Galerkin formulations.

### 2.1. Strong formulation

Let  $\Omega \subset \mathbb{R}^d$ , where  $d \in \{1, 2, 3\}$ , and  $I = (0, T] \subset \mathbb{R}$ . We consider the *linear heat (diffusion) equation*

$$\begin{cases} u_t - \Delta u = f & \text{in } \Omega \times I, \\ u = 0 & \text{on } \partial\Omega \times I, \\ u(0) = u_0 & \text{in } \Omega, \end{cases} \quad (1)$$

where  $u_t := \partial u / \partial t$ ,  $\Delta u = \operatorname{div}(\operatorname{grad}(u))$  is the Laplacian of  $u$  and  $\partial\Omega$  denotes the boundary of the spatial domain  $\Omega$ . The solution  $u(\mathbf{x}, t)$  of (1) could represent the temperature distribution in a body. The source term  $f(\mathbf{x}, t)$  and the initial temperature distribution  $u_0(\mathbf{x})$  are given data. For arbitrary Dirichlet (geometric) boundary conditions, we can modify the source term accordingly, thus making (1) a general statement.

### 2.2. Weak formulation

In order to obtain the weak formulation of (1), we multiply the diffusion equation by test functions  $v$  of a suitable space  $\mathcal{V}$  and we integrate over the whole domain  $\Omega \times I$

$$\int_I \int_{\Omega} \left( v u_t - v \Delta u \right) d\Omega dt = \int_I \int_{\Omega} v f d\Omega dt, \quad \forall v \in \mathcal{V}.$$

Integrating by parts in space the diffusion term and selecting test functions vanishing on  $\partial\Omega$  we obtain

$$\int_I \int_{\Omega} \left( v u_t + \nabla v \cdot \nabla u \right) d\Omega dt = \int_I \int_{\Omega} v f d\Omega dt, \quad \forall v \in \mathcal{V}. \quad (2)$$

A sufficient condition for the above integrals to make sense is if all factors in the above products are in  $L^2$  in both space and time. For the space integrals, taking into account the diffusion term in (2), it seems natural that  $u$  and  $v$  should be in

$$V := H_0^1(\Omega) := \{u \in L^2(\Omega) \mid \nabla u \in L^2(\Omega), u = 0 \text{ on } \partial\Omega\},$$

and therefore, to guarantee integrability of the weak formulation,  $u_t$  and  $f$  should belong to  $V' := H^{-1}(\Omega)$ , which is the dual space of  $V$ . In time, it is enough to ensure that all the functions in (2) are in  $L^2$ .

In the remaining of this article, we omit the spatial dependence of the functions, i.e., we write  $u(t)$  instead of  $u(\mathbf{x}, t)$ . We consider  $u(t)$  and  $v(t)$  as functions in time that take values in suitable Hilbert spaces [41], so in view of the sufficient integrability conditions discussed in the previous paragraph, we construct the following test space

$$\mathcal{V} := L^2(I; V) = \left\{ u : I \longrightarrow V \mid u \text{ is } V\text{-measurable and } \int_I \|u(t)\|_V^2 dt < +\infty \right\},$$

which is the space of all integrable functions in time that take values in  $V$ .

On the other hand, for the solution, we need  $u \in \mathcal{V}$  and  $u_t \in \mathcal{V}' := L^2(I; V')$ , so we define the following trial space

$$\mathcal{U} := \{u \in \mathcal{V} \mid u_t \in \mathcal{V}'\}.$$

Finally, assuming that  $f \in \mathcal{V}'$  and  $u_0 \in L^2(\Omega)$  and imposing the initial condition in weak form, the weak formulation of problem (1) reads: Find  $u \in \mathcal{U}$  such that

$$\begin{cases} \int_I \langle v, u_t \rangle dt + \int_I (\nabla v, \nabla u) dt = \int_I \langle v, f \rangle dt, \quad \forall v \in \mathcal{V}, \\ (\hat{v}, u(0)) = (\hat{v}, u_0), \quad \forall \hat{v} \in L^2(\Omega), \end{cases} \quad (3)$$

where  $\langle \cdot, \cdot \rangle$  denotes the duality pairing between the spaces  $V$  and  $V'$ , and  $(\cdot, \cdot)$  is the inner product in  $L^2(\Omega)$ . We now define

$$B(v, u) := \int_I \langle v, u_t \rangle dt + \int_I (\nabla v, \nabla u) dt. \quad (4)$$

### 3. Discontinuous-in-time Petrov-Galerkin formulation

In this section, we derive *discontinuous-in-time Petrov-Galerkin* formulations that are equivalent to explicit Runge-Kutta methods in time.

We perform a partition of the time interval  $\bar{I} = [0, T]$  as

$$0 = t_0 < t_1 < \dots < t_{m-1} < t_m = T,$$

and denote  $I_k := (t_{k-1}, t_k)$ ,  $\tau_k := t_k - t_{k-1}$ ,  $\forall k = 1, \dots, m$  and  $\tau := \max_{1 \leq k \leq m} \tau_k$ .

Now, we select the following semi-discrete spaces

$$\begin{aligned} \mathcal{U}_\tau &:= \{u \in L^2(I; V) \mid u|_{(t_{k-1}, t_k)} \in P_q([t_{k-1}, t_k]; V), \forall k = 1, \dots, m, u(T) \in V\}, \\ \mathcal{V}_\tau &:= \{v \in L^2(I; V) \mid v|_{(t_{k-1}, t_k]} \in P_r((t_{k-1}, t_k]; V), \forall k = 1, \dots, m, v(0) \in V\}, \end{aligned} \quad (5)$$

where  $P_r(I_k; V)$  denotes the space of polynomials of degree less than or equal to  $r$  taking values in  $V$ . The functions in  $\mathcal{U}_\tau$  and  $\mathcal{V}_\tau$  could be discontinuous at each time step  $t_k$ . Moreover, the trial functions are right-discontinuous while the test functions are left-discontinuous in time [48] (see Figure 1). Therefore, both  $\mathcal{U}_\tau$  and  $\mathcal{V}_\tau$  are subspaces of  $\mathcal{V}$ . However,  $\mathcal{U}_\tau \not\subset \mathcal{U}$  (because functions in  $\mathcal{U}_\tau$  are discontinuous) and  $\mathcal{U} \not\subset \mathcal{U}_\tau$  (since functions in  $\mathcal{U}_\tau$  are piecewise polynomials).

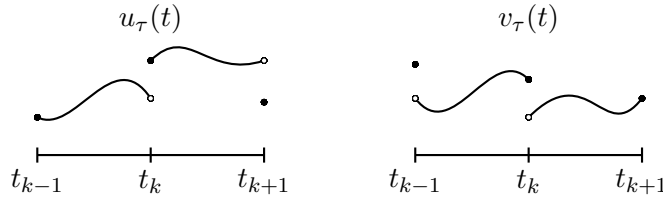


Figure 1: Discontinuous trial and test functions.

As we consider discontinuous functions in time and different trial and test spaces, we need a discontinuous-in-time Petrov-Galerkin formulation of problem (3). We define the jump of a function  $v$  at each time interface  $t_k$  as  $\llbracket v \rrbracket^k = v(t_k^+) - v(t_k^-)$ , where  $v(t_k^\pm) := \lim_{\varepsilon \rightarrow 0^+} v(t_k \pm \varepsilon)$ . Now, we integrate by parts in time the bilinear form (4) over each subinterval  $I_k$

$$- \int_{I_k} \langle v_t, u \rangle dt + \int_{I_k} (\nabla v, \nabla u) dt + (v(t_k^-), u(t_k^-)) - (v(t_{k-1}^+), u(t_{k-1}^+)).$$

At each time interface, instead of performing an upwind approximation of the solution like in the standard DG approach, we perform a *downwind approximation* in order to obtain an explicit method. That is, we substitute  $u(t_k^-)$  by  $u(t_k^+)$  in the above formula. Then, integrating by parts again, we obtain

$$\int_{I_k} \langle v, u_t \rangle dt + \int_{I_k} (\nabla v, \nabla u) dt + (v(t_k^-), u(t_k^+)) - (v(t_k^-), u(t_k^-)). \quad (6)$$

We define two subspaces  $\tilde{\mathcal{U}}_\tau \subset \mathcal{U}_\tau$  and  $\tilde{\mathcal{V}}_\tau \subset \mathcal{V}_\tau$  as

$$\begin{aligned} \tilde{\mathcal{U}}_\tau &:= \{u \in L^2(I; V) \mid u|_{[t_{k-1}, t_k]} \in P_q^{incu}([t_{k-1}, t_k]; V), \forall k = 1, \dots, m, u(T) \in V\}, \\ \tilde{\mathcal{V}}_\tau &:= \{v \in L^2(I; V) \mid v|_{(t_{k-1}, t_k]} \in P_r^{incv}((t_{k-1}, t_k]; V), \forall k = 1, \dots, m, v(0) \in V\}, \end{aligned} \quad (7)$$

where  $P_q^{incu}(I_k; V)$  (and  $P_r^{incv}(I_k; V)$ ) are (possibly) incomplete spaces of polynomials of degree less than or equal to  $q$  (and  $r$ , respectively) taking

values in  $V$ . We define explicitly these incomplete spaces of polynomials in the next section.

Now, summing the expression (6) over all intervals, and adding the initial condition, we obtain the following discontinuous-in-time Petrov-Galerkin formulation of (3):

$$\begin{cases} \text{Find } u_\tau \in \tilde{\mathcal{U}}_\tau \subset \mathcal{U}_\tau \text{ such that} \\ B_{DG}(v_\tau, u_\tau) = F(v_\tau), \quad \forall v_\tau \in \tilde{\mathcal{V}}_\tau \subset \mathcal{V}_\tau, \end{cases} \quad (8)$$

where  $B_{DG}(v, u) := B(v, u) + \sum_{k=1}^m \left( v(t_k^-), \llbracket u \rrbracket^k \right) + (v(0), u(0))$  is a bilinear form that admits discontinuous-in-time functions and

$$F(v) := \int_I \langle v, f \rangle dt + (v(0), u_0).$$

In space, we employ either a *spectral element method (SEM)* or a finite element method (FEM) [40]. We allow dynamic meshes in space so we define a finite dimensional subspace of  $V$  for each time step, i.e.,  $V_h^k \subset V$ ,  $\forall k = 0, \dots, m$ , where  $h$  is the largest element diameter of each dynamic mesh.

Finally, we introduce two new subspaces  $\tilde{\mathcal{U}}_{\tau h} \subset \tilde{\mathcal{U}}_\tau$  and  $\tilde{\mathcal{V}}_{\tau h} \subset \tilde{\mathcal{V}}_\tau$ . Here, to obtain the Forward Euler method, we assume that the basis functions generating the subspaces  $\tilde{\mathcal{U}}_{\tau h}$  and  $\tilde{\mathcal{V}}_{\tau h}$  can be expressed as Cartesian product of space and time functions. However, in more general cases, we lose such structure in order to obtain compatible schemes allowing dynamic meshes in space. We define the fully discrete problem as:

$$\begin{cases} \text{Find } u_{\tau h} \in \tilde{\mathcal{U}}_{\tau h} \subset \tilde{\mathcal{U}}_\tau \text{ such that} \\ B_{DG}(v_{\tau h}, u_{\tau h}) = F(v_{\tau h}), \quad \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h} \subset \tilde{\mathcal{V}}_\tau. \end{cases} \quad (9)$$

We will determine subspaces  $\tilde{\mathcal{U}}_{\tau h}$  and  $\tilde{\mathcal{V}}_{\tau h}$  in the next section so we recover explicit Runge-Kutta methods for (9).

#### 4. Explicit Runge-Kutta methods

In this section, to facilitate the understanding of the construction we propose in this article, we first derive the Forward Euler method. Then, we construct the subspaces for some two-stage and second-order explicit Runge-Kutta methods and finally, we generalize the process to any s-stage explicit Runge-Kutta method.



#### 4.1. Forward Euler method

In (5), we select piecewise constant functions in time, i.e.  $q = r = 0$  and we define the subspaces in (9) as

$$\begin{aligned}\tilde{\mathcal{U}}_{\tau h} &:= \{u \in L^2(I; V) \mid u|_{[t_{k-1}, t_k]} \in P_0([t_{k-1}, t_k]; V_h^{k-1}), \forall k = 1, \dots, m, u(T) \in V_h^m\}. \\ \tilde{\mathcal{V}}_{\tau h} &:= \{v \in L^2(I; V) \mid v|_{(t_{k-1}, t_k]} \in P_0((t_{k-1}, t_k]; V_h^k), \forall k = 1, \dots, m, v(0) \in V_h^0\}.\end{aligned}\tag{10}$$

In particular, the subspaces defined in (7) coincide with the spaces defined in (5) with  $q = r = 0$ , i.e.,  $\tilde{\mathcal{U}}_{\tau} = \mathcal{U}_{\tau}$  and  $\tilde{\mathcal{V}}_{\tau} = \mathcal{V}_{\tau}$ .

We express the solution of (9) as follows

$$u_{\tau h}(t) = \sum_{k=1}^{m+1} u_h^{k-1} \phi^{k-1}(t),\tag{11}$$

where  $u_h^{k-1} \in V_h^{k-1}$ ,  $\forall k = 1, \dots, m+1$  and the trial functions are

$$\phi^{k-1}(t) = \begin{cases} 1, & t \in [t_{k-1}, t_k), \\ 0, & \text{elsewhere,} \end{cases} \quad \phi^m(t) = \begin{cases} 1, & t = t_m, \\ 0, & \text{elsewhere.} \end{cases}\tag{12}$$

We select the following test functions

$$v_h^k \varphi^k(t), \quad \forall k = 0, \dots, m,\tag{13}$$

where  $v_h^k \in V_h^k$ ,  $\forall k = 0, \dots, m$  and

$$\varphi^0(t) = \begin{cases} 1, & t = t_0, \\ 0, & \text{elsewhere,} \end{cases} \quad \varphi^k(t) = \begin{cases} 1, & t \in (t_{k-1}, t_k], \\ 0, & \text{elsewhere.} \end{cases}\tag{14}$$

In this section, we assume that the trial and test functions can be expressed as a Cartesian product of functions in time and space. We will show in the next sections that, in order to allow dynamic meshes in space, we lose such Cartesian product structure of the test space  $\tilde{\mathcal{V}}_{\tau}$ . We commit a slight abuse of notation by omitting the constants in (11) because we can express each function  $u_h^k$  as a linear combination of basis functions in  $V_h^k$ ,  $\forall k = 0, \dots, m$ .

**Theorem 1.** (Forward Euler method) *Selecting trial and test functions as in (11) and (13), problem (9) leads to the following scheme*

$$\left\{ \begin{array}{l} \text{Find } u_h^0 \in V_h^0 \text{ and } u_h^k \in V_h^k, \forall k = 1, \dots, m, \text{ such that} \\ (v_h^k, u_h^k) = (v_h^k, u_h^{k-1}) - \tau_k (\nabla v_h^k, \nabla u_h^{k-1}) + \int_{I_k} \langle v_h^k, f \rangle dt, \forall v_h^k \in V_h^k, \\ (v_h^0, u_h^0) = (v_h^0, u_0), \forall v_h^0 \in V_h^0, \end{array} \right. \quad (15)$$

which is an explicit method that is a variant of the Forward Euler method in time.

*Proof.* From the first test function of (13),  $v_{\tau h}(t) = v_h^0 \varphi^0(t)$ , and substituting it into (9), we obtain

$$(v_h^0, u_h^0) = (v_h^0, u_0), \quad \forall v_h^0 \in V_h^0,$$

which is the  $L^2$ -projection in space of the initial condition  $u_0$  on  $V_h^0 \subset V$ .

Then, for  $k = 1, \dots, m$ , each test function of the form  $v_{\tau h}(t) = v_h^k \varphi^k(t)$  has local support in  $(t_{k-1}, t_k]$ , so problem (9) reads

$$\int_{I_k} \langle v_{\tau h}, u_{\tau h, t} \rangle dt + \int_{I_k} (\nabla v_{\tau h}, \nabla u_{\tau h}) dt + (v_{\tau h}(t_k^-), \llbracket u_{\tau h} \rrbracket^k) = \int_{I_k} \langle v_{\tau h}, f \rangle dt.$$

Now, as both trial and test functions are piecewise constant in time, each term of the previous formula becomes

$$\begin{aligned} \int_{I_k} \langle v_{\tau h}, u_{\tau h, t} \rangle dt &= (v_h^k, u_h^{k-1}) \int_{I_k} \varphi^k \frac{\partial}{\partial t} \phi^{k-1} dt = 0, \\ \int_{I_k} (\nabla v_{\tau h}, \nabla u_{\tau h}) dt &= (\nabla v_h^k, \nabla u_h^{k-1}) \int_{I_k} \varphi^k \phi^{k-1} dt = \tau_k (\nabla v_h^k, \nabla u_h^{k-1}), \\ (v_{\tau h}(t_k^-), \llbracket u_{\tau h} \rrbracket^k) &= (v_h^k \varphi^k(t_k^-), u_h^k \phi^k(t_k^+) - u_h^{k-1} \phi^{k-1}(t_k^-)) = (v_h^k, u_h^k - u_h^{k-1}), \\ \int_{I_k} \langle v_{\tau h}, f \rangle dt &= \int_{I_k} \langle v_h^k, f \rangle dt. \end{aligned}$$

Finally, we obtain

$$(v_h^k, u_h^k) = (v_h^k, u_h^{k-1}) - \tau_k (\nabla v_h^k, \nabla u_h^{k-1}) + \int_{I_k} \langle v_h^k, f \rangle dt.$$

□

Scheme (15) is the Forward Euler method in time except for the source term. A standard difference between variational forms and difference methods is that variational forms include an integral measure rather than a point-wise sample of the forcing terms. In space, we can then employ the spectral element method, which leads to a diagonal mass matrix for arbitrary dimensional problems using arbitrary geometrical mappings [40].

**Remark 1.** *To obtain an expression whose form is identical to the classical Forward Euler method, we can interpolate the source term as*

$$f(\mathbf{x}, t) = \sum_{k=1}^{m+1} f^{k-1}(\mathbf{x}) \phi^{k-1}(t),$$

where we identify  $f^k(\mathbf{x})$  with  $f(\mathbf{x}, t_k)$ . Then, the source term in (15) becomes

$$\int_{I_k} \langle v_h^k, f \rangle dt = \int_{I_k} \langle v_h^k, f^{k-1} \phi^{k-1} \rangle dt = \tau_k \langle v_h^k, f^{k-1} \rangle.$$

**Remark 2.** *In the subspaces defined in (10), as  $v_h^k \in V_h^k$ ,  $\forall k = 0, \dots, m$ , the spatial discrete spaces in the test space  $\mathcal{V}_{\tau h}$  are displaced in time with respect to the trial space  $\tilde{\mathcal{U}}_{\tau h}$ , which leads to a Petrov-Galerkin method. This is needed to obtain invertible square mass matrices on the left-hand-side of (15). Figure 2 illustrates this displacement of the spaces.*

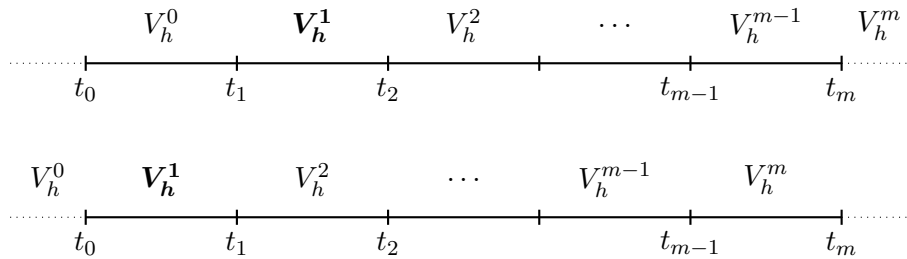


Figure 2: Illustration of the displacement in time of the trial (top) and test (bottom) discrete spaces for the lowest order case  $q = r = 0$ .

#### 4.2. Two-stage Runge-Kutta methods

In this section, following a similar derivation as in Section 4.1, we build trial and test functions that generate the subspaces in (9) to obtain equivalent methods to some two-stage second-order Runge-Kutta methods.

We introduce a test function for the initial condition  $v_h \varphi^0(t)$  with  $v_h \in V_h^0$  and being  $\varphi^0(t)$  the function defined in (14). We also introduce two test functions per temporal element  $I_k$

$$v_{h,1} \varphi_1^k(t), v_{h,2} \varphi_2^k(t), \quad (16)$$

where  $v_{h,1} \in V_h^k$ ,  $v_{h,2} \in V_h^{k-1}$ ,  $\forall k = 1, \dots, m$ . Then, we express the solution of (9) as

$$u_{\tau h}(t) = \sum_{k=1}^m u_{h,1}^{k-1} \phi_1^{k-1}(t) + u_{h,2}^{k-1} \phi_2^{k-1}(t) + u_h^m \phi^m(t), \quad (17)$$

where  $u_{h,1}^{k-1}, u_{h,2}^{k-1} \in V_h^{k-1}$ ,  $\forall k = 1, \dots, m$ ,  $u_h^m \in V_h^m$  and  $\phi^m(t)$  is the function defined in (12).

As before, the trial and test functions are piecewise polynomials defined in  $I_k$  and globally discontinuous across the time interfaces. We define the subspaces in (9) following a similar construction as in [44]

$$\begin{aligned} \tilde{\mathcal{U}}_{\tau h} &:= \{u \in L^2(I; V) \mid u|_{[t_{k-1}, t_k]} \in \tilde{\mathcal{U}}_{\tau h}^{k-1}, \forall k = 1, \dots, m, u(T) \in V_h^m\}, \\ \tilde{\mathcal{V}}_{\tau h} &:= \{v \in L^2(I; V) \mid v|_{(t_{k-1}, t_k]} \in \tilde{\mathcal{V}}_{\tau h}^k, \forall k = 1, \dots, m, v(0) \in V_h^0\}, \end{aligned}$$

where we define the following discrete subspaces in each interval

$$\begin{aligned} \tilde{\mathcal{U}}_{\tau h}^{k-1} &:= \text{span} \left\{ u_{h,1}^{k-1} \phi_1^{k-1}(t), u_{h,2}^{k-1} \phi_2^{k-1}(t) \mid u_{h,1}^{k-1}, u_{h,2}^{k-1} \in V_h^{k-1} \right\}, \\ \tilde{\mathcal{V}}_{\tau h}^k &:= \text{span} \left\{ v_{h,1} \varphi_1^k(t), v_{h,2} \varphi_2^k(t) \mid v_{h,1} \in V_h^k, v_{h,2} \in V_h^{k-1} \right\}. \end{aligned}$$

With this construction, we have that  $\tilde{\mathcal{U}}_{\tau h}^{k-1} \subset P_q^{\text{inc}_u}([t_{k-1}, t_k]; V)$  and  $\tilde{\mathcal{V}}_{\tau h}^k \subset P_r^{\text{inc}_v}((t_{k-1}, t_k]; V)$  and therefore,  $\tilde{\mathcal{U}}_{\tau h} \subset \tilde{\mathcal{U}}_{\tau}$  and  $\tilde{\mathcal{V}}_{\tau h} \subset \tilde{\mathcal{V}}_{\tau}$ . Here, we set the orders of the polynomials equal to the number of stages, i.e.,  $p = r = 2$ .

**Remark 3.** *In order for the discrete system to make sense and obtain invertible square mass matrices at each stage, we need the test functions (16) to satisfy*

$$v_{h,1} \in V_h^k, v_{h,2} \in V_h^{k-1},$$

*so while both test functions are polynomials in time, in space they belong to different spaces. Figure 3 illustrates this choice of the test functions.*

We assume that each trial function is associated with a coefficient (as in Figure 4). Thus, we impose

$$\begin{cases} \phi_1^{k-1}(t_{k-1}^+) = 1, \\ \phi_2^{k-1}(t_{k-1}^+) = 0, \end{cases} \quad (18)$$

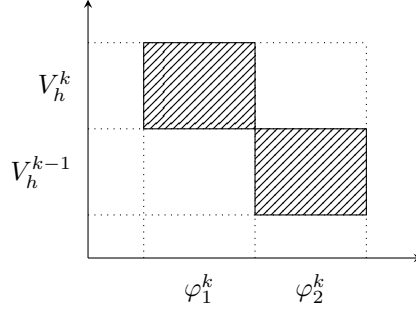


Figure 3: Non-Cartesian space-time structure of the test functions inside each time interval  $I_k$ .

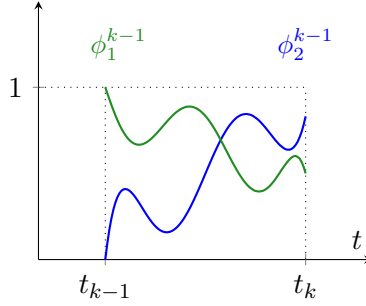


Figure 4: Trial functions of arbitrary order inside each element  $I_k$ .

so  $u_{h,1}^{k-1} \in V_h^{k-1}$  is the value of  $u_{\tau h}(t)$  at  $t_{k-1}^+$ .

We first substitute (17) and the first test function  $v_h \varphi^0(t)$  into (9) and we obtain

$$(v_h, u_{h,1}^0) = (v_h, u_0), \quad \forall v_h \in V_h^0.$$

Now, if we substitute (17) into (9), as each test function has local support in  $I_k$ , we have that

$$B_{DG}(v_{h,j} \varphi_j^k, u_{h,1}^{k-1} \phi_1^{k-1}) + B_{DG}(v_{h,j} \varphi_j^k, u_{h,2}^{k-1} \phi_2^{k-1}) = F(v_{h,j} \varphi_j^k), \quad (19)$$

where  $j \in \{1, 2\}$ . In the above equation, we know that the jump of  $u_{\tau h}(t)$  at  $t_k$  is

$$\begin{aligned} \llbracket u_{\tau h} \rrbracket^k &= u_{\tau h}(t_k^+) - u_{\tau h}(t_k^-) \\ &= u_{h,1}^k \phi_1^k(t_k^+) + u_{h,2}^k \phi_2^k(t_k^+) - u_{h,1}^{k-1} \phi_1^{k-1}(t_k^-) - u_{h,2}^{k-1} \phi_2^{k-1}(t_k^-) \\ &= u_{h,1}^k - u_{h,1}^{k-1} \phi_1^{k-1}(t_k^-) - u_{h,2}^{k-1} \phi_2^{k-1}(t_k^-), \end{aligned}$$

$\forall k = 1, \dots, m-1$  and, in particular, for the last time step, we have:

$$\begin{aligned} \llbracket u_{\tau h} \rrbracket^m &= u_{\tau h}(t_m^+) - u_{\tau h}(t_m^-) \\ &= u_h^m \phi^m(t_m^+) - u_{h,1}^{m-1} \phi_1^{m-1}(t_m^-) - u_{h,2}^{m-1} \phi_2^{m-1}(t_m^-) \\ &= u_h^m - u_{h,1}^{m-1} \phi_1^{m-1}(t_m^-) - u_{h,2}^{m-1} \phi_2^{m-1}(t_m^-). \end{aligned}$$

Equivalently, we write equation (19) as

$$\begin{aligned} & \left( v_{h,j}, u_{h,1}^{k-1} \right) \int_{I_k} \varphi_j^k \frac{\partial}{\partial t} \phi_1^{k-1} dt + \left( v_{h,j}, u_{h,2}^{k-1} \right) \int_{I_k} \varphi_j^k \frac{\partial}{\partial t} \phi_2^{k-1} dt \\ & + \left( \nabla v_{h,j}, \nabla u_{h,1}^{k-1} \right) \int_{I_k} \varphi_j^k \phi_1^{k-1} dt + \left( \nabla v_{h,j}, \nabla u_{h,2}^{k-1} \right) \int_{I_k} \varphi_j^k \phi_2^{k-1} dt \\ & + \left( v_{h,j} \varphi_j^k(t_k^-), u_{h,1}^k - u_{h,1}^{k-1} \phi_1^{k-1}(t_k^-) - u_{h,2}^{k-1} \phi_2^{k-1}(t_k^-) \right) = \int_{I_k} \left\langle v_{h,j} \varphi_j^k, f \right\rangle dt. \end{aligned} \tag{20}$$

Now, reorganizing the terms in (20) we obtain

$$\begin{aligned} & \left( v_{h,j}, u_{h,1}^{k-1} \right) \left( \int_{I_k} \varphi_j^k \frac{\partial}{\partial t} \phi_1^{k-1} dt - \varphi_j^k(t_k^-) \phi_1^{k-1}(t_k^-) \right) \\ & + \left( v_{h,j}, u_{h,2}^{k-1} \right) \left( \int_{I_k} \varphi_j^k \frac{\partial}{\partial t} \phi_2^{k-1} dt - \varphi_j^k(t_k^-) \phi_2^{k-1}(t_k^-) \right) \\ & + \left( \nabla v_{h,j}, \nabla u_{h,1}^{k-1} \right) \int_{I_k} \varphi_j^k \phi_1^{k-1} dt \\ & + \left( \nabla v_{h,j}, \nabla u_{h,2}^{k-1} \right) \int_{I_k} \varphi_j^k \phi_2^{k-1} dt \\ & + (v_{h,j}, u_{h,1}^k) \varphi_j^k(t_k^-) = \int_{I_k} \left\langle v_{h,j} \varphi_j^k, f \right\rangle dt. \end{aligned} \tag{21}$$

We build the trial and test functions to guarantee the satisfaction of some design conditions. We need the following orthogonality conditions in order to obtain an explicit method:

$$\begin{cases} \varphi_2^k(t_k^-) = 0, \\ \int_{I_k} \varphi_2^k \phi_2^{k-1} dt = 0. \end{cases} \tag{22}$$

In (21) we have two equations (one per test function) and  $u_{h,1}^{k-1}$  is a known value from the previous step. By imposing (22), we can calculate

explicitly  $u_{h,2}^{k-1}$  from  $u_{h,1}^{k-1}$  and then compute  $u_{h,1}^k$  from  $u_{h,1}^{k-1}$  and  $u_{h,2}^{k-1}$ . To obtain a Runge-Kutta method, we need to impose further conditions on the system. Indeed, the general expression of the two-stage and second-order explicit Runge-Kutta method we want to obtain is

$$\begin{aligned} (v_{h,1}, u_{h,1}^k) - (v_{h,1}, u_{h,1}^{k-1}) + \frac{\tau_k}{2\alpha} (\nabla v_{h,1}, \nabla u_{h,2}^{k-1}) \\ + \left(1 - \frac{1}{2\alpha}\right) \tau_k (\nabla v_{h,1}, \nabla u_{h,1}^{k-1}) = \int_{I_k} \langle v_{h,1} \varphi_1^k, f \rangle dt, \quad (23) \\ (v_{h,2}, u_{h,2}^{k-1}) - (v_{h,2}, u_{h,1}^{k-1}) + \alpha \tau_k (\nabla v_{h,2}, \nabla u_{h,1}^{k-1}) = \int_{I_k} \langle v_{h,2} \varphi_2^k, f \rangle dt, \end{aligned}$$

where  $\alpha \in \mathbb{R} - \{0\}$  and its corresponding Butcher tableau is described in Table 1.

0	0	0
$\alpha$	$\alpha$	0
	$1 - \frac{1}{2\alpha}$	$\frac{1}{2\alpha}$

Table 1: Butcher tableau for the two-stage and second-order Runge-Kutta method.

In order to obtain (23) from (21), in addition to the orthogonality conditions (22), we need to impose also the following conditions from the first equation of (23):

$$\left\{ \begin{aligned} \varphi_1^k(t_k^-) &= 1, \\ \int_{I_k} \varphi_1^k \frac{\partial}{\partial t} \phi_2^{k-1} dt - \varphi_1^k(t_k^-) \phi_2^{k-1}(t_k^-) &= 0, \\ \int_{I_k} \varphi_1^k \frac{\partial}{\partial t} \phi_1^{k-1} dt - \varphi_1^k(t_k^-) \phi_1^{k-1}(t_k^-) &= -1, \\ \int_{I_k} \varphi_1^k \phi_2^{k-1} dt &= \frac{\tau_k}{2\alpha}, \\ \int_{I_k} \varphi_1^k \phi_1^{k-1} dt &= \left(1 - \frac{1}{2\alpha}\right) \tau_k, \end{aligned} \right. \quad (24)$$

and also from the second equation of (23):

$$\left\{ \begin{array}{l} \int_{I_k} \varphi_2^k \frac{\partial}{\partial t} \phi_2^{k-1} dt - \varphi_2^k(t_k^-) \phi_2^{k-1}(t_k^-) = 1, \\ \int_{I_k} \varphi_2^k \frac{\partial}{\partial t} \phi_1^{k-1} dt - \varphi_2^k(t_k^-) \phi_1^{k-1}(t_k^-) = -1, \\ \int_{I_k} \varphi_2^k \phi_1^{k-1} dt = \alpha \tau_k. \end{array} \right. \quad (25)$$

Finally, from (18), (22), (24) and (25), the conditions that the trial and test functions must satisfy to obtain the general 2-stage Runge-Kutta method (23) are

$$\begin{array}{ll} \varphi_1^k(t_k^-) = 1, & \varphi_2^k(t_k^-) = 0, \\ \phi_1^{k-1}(t_{k-1}^+) = 1, & \phi_2^{k-1}(t_{k-1}^+) = 0, \\ \int_{I_k} \varphi_1^k \frac{\partial}{\partial t} \phi_1^{k-1} dt - \phi_1^{k-1}(t_k^-) = -1, & \int_{I_k} \varphi_1^k \frac{\partial}{\partial t} \phi_2^{k-1} dt - \phi_2^{k-1}(t_k^-) = 0, \\ \int_{I_k} \varphi_2^k \frac{\partial}{\partial t} \phi_1^{k-1} dt = -1, & \int_{I_k} \varphi_2^k \frac{\partial}{\partial t} \phi_2^{k-1} dt = 1, \\ \int_{I_k} \varphi_1^k \phi_1^{k-1} dt = \left(1 - \frac{1}{2\alpha}\right) \tau_k, & \int_{I_k} \varphi_1^k \phi_2^{k-1} dt = \frac{\tau_k}{2\alpha}, \\ \int_{I_k} \varphi_2^k \phi_1^{k-1} dt = \alpha \tau_k, & \int_{I_k} \varphi_2^k \phi_2^{k-1} dt = 0. \end{array} \quad (26)$$

For the trial and test functions, which are polynomials of arbitrary order, conditions (26) become a system of nonlinear equations. In particular, if we select linear trial and test functions, we obtain no solutions. However, if we select quadratic functions, we have a system of 12 nonlinear equations with 12 unknowns, which has two solutions. We solve the resulting system in the master element  $[0, 1]$  using the MATLAB code we describe in Appendix B (see Table 2), and we obtain different sets of trial and test functions depending on the value of  $\alpha$ .

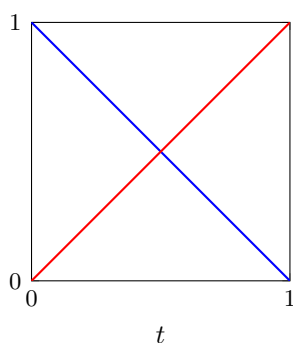
*Example 1: Explicit trapezoidal rule*

When  $\alpha = 1$ , (23) is equivalent to the *explicit trapezoidal rule* [28]. Figure 5 shows the trial and test functions of both solutions over the master element.

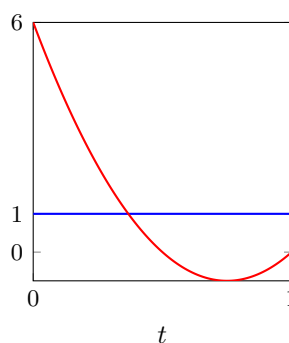


	Solution 1	Solution 2
$\phi_1^{k-1}(t)$	$-\frac{1}{\alpha}t + 1$	$-\frac{6}{\alpha}t^2 + \frac{3}{\alpha}t + 1$
$\phi_2^{k-1}(t)$	$\frac{1}{\alpha}t$	$\frac{6}{\alpha}t^2 - \frac{3}{\alpha}t$
$\varphi_1^k(t)$	1	1
$\varphi_2^k(t)$	$12\alpha t^2 - 18\alpha t + 6\alpha$	$-2\alpha t + 2\alpha$

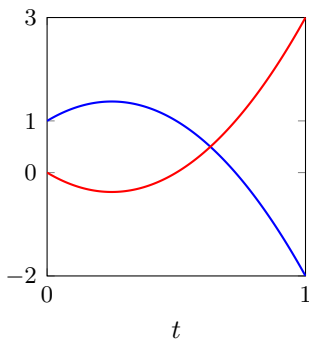
Table 2: Trial and test functions defined over the master element  $[0, 1]$  that lead to the two-stage Runge-Kutta method.



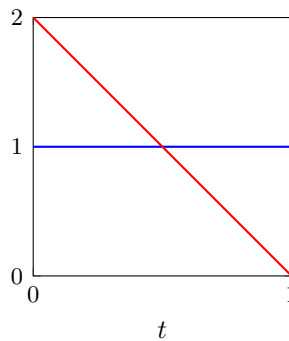
(a) Trial functions of the first solution.



(b) Test functions of the first solution.



(c) Trial functions of the second solution.

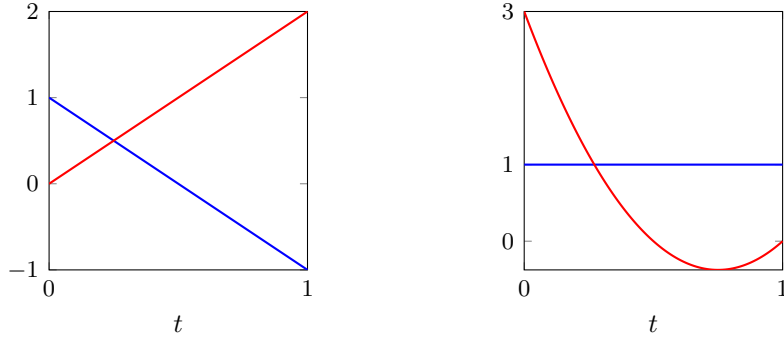


(d) Test functions of the second solution.

Figure 5: Trial and test functions over the master element  $[0, 1]$  when  $\alpha = 1$ .

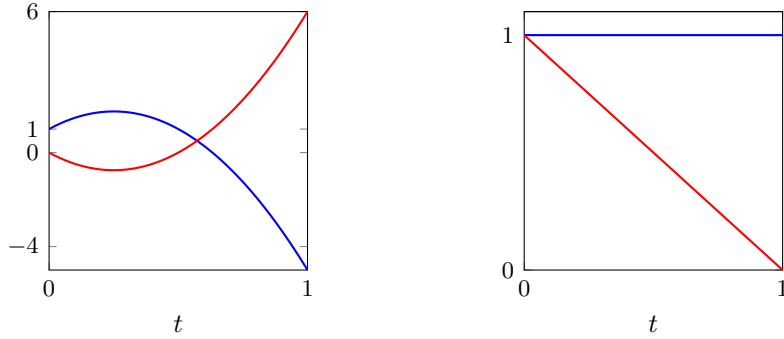
*Example 2: Explicit midpoint rule*

When  $\alpha = \frac{1}{2}$ , we obtain the *explicit midpoint rule* [28]. Figure 6 shows the trial and test functions of both solutions over  $[0, 1]$ .



(a) Trial functions of the first solution.

(b) Test functions of the first solution.



(c) Trial functions of the second solution.

(d) Test functions of the second solution.

Figure 6: Trial and test functions over the master element  $[0, 1]$  when  $\alpha = \frac{1}{2}$ .

**Remark 4.** *As in Section 4.1, to obtain expressions whose form is identical to standard Runge-Kutta methods, we can interpolate the source term in the trial space as*

$$\begin{aligned} \int_{I_k} \langle v_{h,1} \varphi_1^k, f \rangle dt &= \int_{I_k} \langle v_{h,1} \varphi_1^k, f_1^{k-1} \phi_1^{k-1} + f_2^{k-1} \phi_2^{k-1} \rangle dt \\ &= \left(1 - \frac{1}{2\alpha}\right) \tau_k \langle v_{h,1}, f_1^{k-1} \rangle + \frac{\tau_k}{2\alpha} \langle v_{h,1}, f_2^{k-1} \rangle, \end{aligned}$$

$$\int_{I_k} \langle v_{h,2} \varphi_2^k, f \rangle dt = \int_{I_k} \langle v_{h,2} \varphi_2^k, f_1^{k-1} \phi_1^{k-1} + f_2^{k-1} \phi_2^{k-1} \rangle dt = \alpha \tau_k \langle v_{h,2}, f_1^{k-1} \rangle,$$

*which is the general two-stage Runge-Kutta method in time. Here, we identify  $f_1^{k-1}(\mathbf{x}) := f(\mathbf{x}, t_{k-1})$  and  $f_2^{k-1}(\mathbf{x}) := f(\mathbf{x}, t_{k-1} + \alpha \tau_k)$ .*

In these examples, we set  $q = r = 2$  in (7) and we have two possibilities for the subspaces in problem (9). In the first solution of Table 2, the trial

space  $\tilde{\mathcal{U}}_{\tau h}$  is the complete space of linear polynomials, while in the second solution it is an incomplete space of quadratic polynomials that only reproduces constant functions in time. Both solutions are nodally equivalent to the same two-stage and second-order explicit Runge-Kutta method. We will focus on the first solution in Table 2 so we can rely on approximation theory results inside the temporal intervals.

#### 4.3. General $s$ -stage Runge-Kutta Methods

In this section, we generalize the constructions of Sections 4.1 and 4.2, to the general  $s$ -stage explicit Runge-Kutta Method.

The general method we want to obtain is of the form

$$\begin{aligned} (v_{h,1}, u_{h,1}^k) - (v_{h,1}, u_{h,1}^{k-1}) + \tau_k \sum_{i=1}^s b_i (\nabla v_{h,1}, \nabla u_{h,i}^{k-1}) &= \tau_k \sum_{i=1}^s b_i \langle v_{h,1}, f_i^{k-1} \rangle, \\ (v_{h,i}, u_{h,i}^{k-1}) - (v_{h,i}, u_{h,i}^{k-1}) + \tau_k \sum_{j=1}^{i-1} a_{ij} (\nabla v_{h,i}, \nabla u_{h,j}^{k-1}) &= \tau_k \sum_{j=1}^{i-1} a_{ij} \langle v_{h,i}, f_j^{k-1} \rangle, \end{aligned} \quad (27)$$

$\forall i = 2, \dots, s$ , where

$$f_i^{k-1}(\mathbf{x}) := f(\mathbf{x}, t_{k-1} + c_i \tau_k), \quad \forall i = 1, \dots, s. \quad (28)$$

The coefficients  $a_{ij}$ ,  $b_i$ ,  $c_i$ , with  $i, j \in \{1, \dots, s\}$ , are the ones corresponding to the Butcher tableau (see Table 3) [9]. As (27) is an explicit method, we have that

$$a_{ij} = 0, \forall j \geq i.$$

0	0	0	0	...	0
$c_2$	$a_{21}$	0	0	...	0
$c_3$	$a_{31}$	$a_{32}$	0	...	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$
$c_s$	$a_{s1}$	$a_{s2}$	...	$a_{s,s-1}$	0
	$b_1$	$b_2$	...	$b_{s-1}$	$b_s$

Table 3: Butcher tableau.

We consider  $s$  trial functions per time interval

$$\phi_1^{k-1}(t), \dots, \phi_s^{k-1}(t),$$

so we express the solution in (9) as

$$u_{\tau h}(t) = \sum_{k=1}^m \sum_{i=1}^s u_{h,i}^{k-1} \phi_i^{k-1}(t) + u_h^m \phi^m(t).$$

We also consider  $s$  test functions per time interval

$$v_{h,1}\varphi_1^k(t), v_{h,2}\varphi_2^k(t), \dots, v_{h,s}\varphi_s^k(t),$$

$\forall k = 1, \dots, m$  and we define the subspaces in (9) as

$$\begin{aligned} \tilde{\mathcal{U}}_{\tau h} &:= \{u \in L^2(I; V) \mid u|_{[t_{k-1}, t_k]} \in \tilde{\mathcal{U}}_{\tau h}^{k-1}, \forall k = 1, \dots, m, u(T) \in V_h^m\}, \\ \tilde{\mathcal{V}}_{\tau h} &:= \{v \in L^2(I; V) \mid v|_{(t_{k-1}, t_k]} \in \tilde{\mathcal{V}}_{\tau h}^k, \forall k = 1, \dots, m, v(0) \in V_h^0\}, \end{aligned}$$

where we define

$$\begin{aligned} \tilde{\mathcal{U}}_{\tau h}^{k-1} &:= \text{span} \left\{ u_{h,i}^{k-1} \phi_i^{k-1}(t) \mid u_{h,i}^{k-1} \in V_h^{k-1}, \forall i = 1, \dots, s \right\}, \\ \tilde{\mathcal{V}}_{\tau h}^k &:= \text{span} \left\{ v_{h,1} \varphi_1^k(t), v_{h,j} \varphi_j^k(t) \mid v_{h,1} \in V_h^k, v_{h,j} \in V_h^{k-1}, \forall j = 2, \dots, s \right\}. \end{aligned}$$

**Remark 5.** *Again, to properly define solvable discrete systems, we seek to obtain square mass matrices in (27). Thus, we need test functions satisfying*

$$v_{h,1} \in V_h^k, v_{h,j} \in V_h^{k-1}, \quad \forall j = 2, \dots, s.$$

Figure 7 illustrates this choice of the test functions.

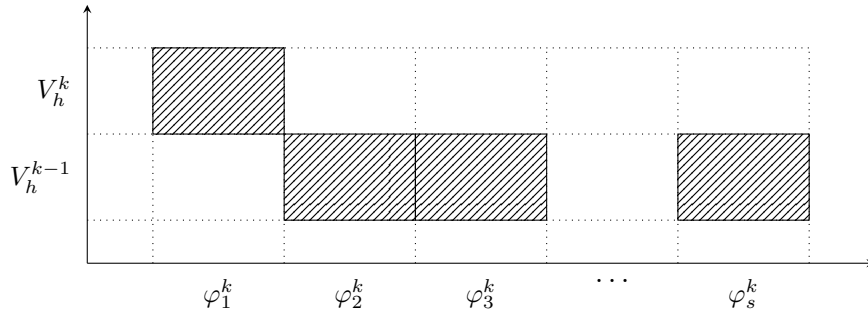


Figure 7: Non-Cartesian space-time structure of the test functions inside each time interval  $I_k$ .

As before, to obtain an expression whose form is identical to the classical Runge-Kutta methods, we can interpolate the source term using the trial functions as

$$f(\mathbf{x}, t) = \sum_{k=1}^{m+1} \sum_{i=1}^s f_i^{k-1}(\mathbf{x}) \phi_i^{k-1}(t).$$

Following the same logic as in Section 4.2, we generalize conditions (26) to  $s$ -stages as follows

$$\begin{aligned} \varphi_1^k(t_k^-) &= 1, & \varphi_j^k(t_k^-) &= 0, \quad \forall j = 2, \dots, s, \\ \phi_1^{k-1}(t_{k-1}^+) &= 1, & \phi_i^{k-1}(t_{k-1}^+) &= 0, \quad \forall i = 2, \dots, s, \\ \int_{I_k} \varphi_1^k \frac{\partial}{\partial t} \phi_1^{k-1} dt - \phi_1^{k-1}(t_k^-) &= -1, & \int_{I_k} \varphi_1^k \frac{\partial}{\partial t} \phi_i^{k-1} dt - \phi_i^{k-1}(t_k^-) &= 0, \quad \forall i = 2, \dots, s, \\ \int_{I_k} \varphi_j^k \frac{\partial}{\partial t} \phi_1^{k-1} dt &= -1, & \int_{I_k} \varphi_j^k \frac{\partial}{\partial t} \phi_i^{k-1} dt &= \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}, \quad \forall i, j = 2, \dots, s, \\ \int_{I_k} \varphi_1^k \phi_i^{k-1} dt &= \tau_k b_i, \quad \forall i = 1, \dots, s, \\ \int_{I_k} \varphi_j^k \phi_i^{k-1} dt &= \tau_k a_{ji}, \quad \forall i = 1, \dots, s, \quad \forall j = 2, \dots, s. \end{aligned} \tag{29}$$

Appendix A gives an explicit matrix form for (29).

**Remark 6.** We can extend this construction to ODEs of the form  $\dot{u} = f(t, u(t))$ ,  $u(0) = u_0$  by interpolating the right-hand-side as follows

$$f(t, u(t)) \simeq \sum_{k=1}^{m+1} \sum_{i=1}^s f(t_{k-1} + \tau_k c_i, u_{h,i}^{k-1}) \phi_i^{k-1}(t),$$

where  $\text{span}\{\phi_i^{k-1}\}_{i=1}^s$  is a complete space of polynomials.

*Example 1: Three-stage Runge-Kutta*

We calculate the trial and test functions of the three-stage and third order Runge-Kutta method that has the Butcher tableau as in Table 4 [9].

If we consider cubic polynomials ( $q = r = 3$ ) in (7), we obtain four possible solutions for the trial and test basis functions in the subspaces (9): two of them with real coefficients (see Table 5) and the remaining two solutions with complex conjugate coefficients (see Table 6). In Table 6,  $z_j$  and  $\bar{z}_j$  denote the following complex numbers and their conjugates.

0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0
1	-1	2	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 4: Butcher tableau of the three-stage Runge-Kutta method.

$$\begin{aligned}
z_1 &= (9 - i\sqrt{66})/7, & z_6 &= (34 + 3i\sqrt{66})/7, \\
z_2 &= (11 - 2i\sqrt{66})/7, & z_7 &= (16 + i\sqrt{66})/7, \\
z_3 &= (12 + i\sqrt{66})/7, & z_8 &= (30 + i\sqrt{66})/7, \\
z_4 &= (5 + i\sqrt{66})/7, & z_9 &= (89 + 6i\sqrt{66})/7, \\
z_5 &= (2 - i\sqrt{66})/7, & z_{10} &= (39 + 2i\sqrt{66})/7.
\end{aligned}$$

Here, the four solutions are nodally equivalent to the three stage and third order explicit Runge-Kutta method defined in Table 4. However, the utility of the complex solutions is unknown for us and an open area of research. Figure 8 shows the trial and test functions of the first real solution over the master element  $[0, 1]$ . In this case, the trial space  $\tilde{\mathcal{U}}_{\tau h}$  is a complete space of quadratic polynomials.

	Solution 1	Solution 2
$\phi_1^{k-1}(t)$	$\frac{1}{2}t^2 - 2t + 1$	$110t^3 - 130t^2 + 30t + 1$
$\phi_2^{k-1}(t)$	$-t^2 + 2t$	$-120t^3 + 140t^2 - 32t$
$\phi_3^{k-1}(t)$	$\frac{1}{2}t^2$	$10t^3 - 10t^2 + 2t$
$\varphi_1^k(t)$	1	1
$\varphi_2^k(t)$	$-30t^3 + 60t^2 - 36t + 6$	$\frac{3}{2}t^2 - 3t + \frac{3}{2}$
$\varphi_3^k(t)$	$420t^3 - 780t^2 + 408t - 48$	$-6t^2 + 6t$

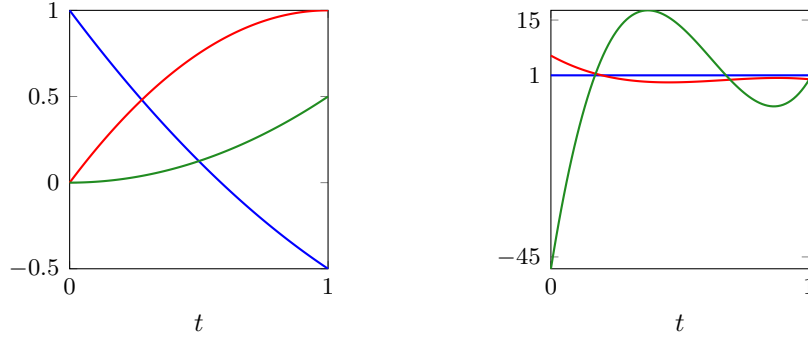
Table 5: Trial and test functions with real coefficients over the master element  $[0, 1]$  that lead to a three-stage Runge-Kutta method.

*Example 2: Four-stage Runge-Kutta*

Now, we consider the four-stage and fourth order Runge-Kutta method with the Butcher tableau as in Table 7 [9]. In order to reduce the number of unknowns, and taking into account the solutions we obtained for one and

	Solution 3	Solution 4
$\phi_1^{k-1}(t)$	$\frac{10}{3}z_1t^3 - 2z_2t^2 - z_3t + 1$	$\frac{10}{3}\bar{z}_1t^3 - 2\bar{z}_2t^2 - \bar{z}_3t + 1$
$\phi_2^{k-1}(t)$	$-\frac{20}{3}z_1t^3 + 4z_2t^2 + 2z_4t$	$-\frac{20}{3}\bar{z}_1t^3 + 4\bar{z}_2t^2 + 2\bar{z}_4t$
$\phi_3^{k-1}(t)$	$\frac{10}{3}z_1t^3 - 2z_2t^2 + z_5t$	$\frac{10}{3}\bar{z}_1t^3 - 2\bar{z}_2t^2 + \bar{z}_5t$
$\varphi_1^k(t)$	1	1
$\varphi_2^k(t)$	$-10z_1t^3 + 6z_6t^2 - 9z_7t + z_8$	$-10z_1t^3 + 6z_6t^2 - 9z_7t + z_8$
$\varphi_3^k(t)$	$40z_1t^3 - 24z_6t^2 + 6z_9t - 2z_{10}$	$40z_1t^3 - 24z_6t^2 + 6z_9t - 2z_{10}$

Table 6: Trial and test functions with complex coefficients over the master element  $[0, 1]$  that lead to a three-stage Runge-Kutta method.



(a) Trial functions of the first real solution. (b) Test functions of the first real solution.

Figure 8: Trial and test functions over the master element  $[0, 1]$  for the real solutions of the three-stage Runge-Kutta method.

two stages, we solve the nonlinear system considering non-symmetric choices for the order of polynomials in (7):  $q = 3$ ,  $r = 4$  and  $q = 4$ ,  $r = 3$ . Table 8 shows two real solutions for this method and Figure 9 shows the solution for which the trial space  $\tilde{\mathcal{U}}_{\tau h}$  is a complete space of cubic polynomials.

## 5. Goal-oriented error estimation

The construction presented in this paper could be used to represent the error in the quantity of interest in order to design explicit in time goal-oriented adaptive algorithms.

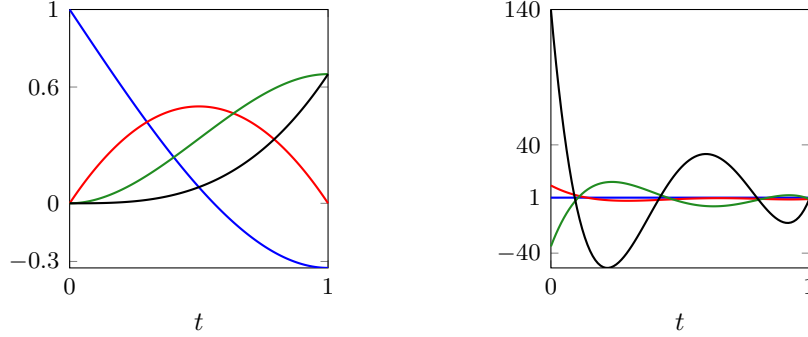
First, note that problem (3) holds for every subspace of  $\mathcal{V}$ . So selecting

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 7: Butcher tableau of the four-stage Runge-Kutta method.

	Solution 1	Solution 2
$\phi_1^{k-1}(t)$	$\frac{2}{3}t^3 - 2t + 1$	$-\frac{2800}{3}t^4 + 1610t^3 - 810t^2 + \frac{320}{3}t + 1$
$\phi_2^{k-1}(t)$	$-2t^2 + 2t$	$\frac{4480}{3}t^4 - 2520t^3 + 1230t^2 - \frac{470}{3}t$
$\phi_3^{k-1}(t)$	$-\frac{4}{3}t^3 + 2t^2$	$-\frac{1820}{3}t^4 + 980t^3 - 450t^2 + \frac{160}{3}t$
$\phi_4^{k-1}(t)$	$\frac{2}{3}t^3$	$\frac{140}{3}t^4 - 70t^3 + 30t^2 - \frac{10}{3}t$
$\varphi_1^k(t)$	1	1
$\varphi_2^k(t)$	$140t^4 - 350t^3 + 300t^2 - 100t + 10$	$-2t^3 + 6t^2 - 6t + 2$
$\varphi_3^k(t)$	$-910t^4 + 2170t^3 - 1725t^2 + 500t - 35$	$2t^3 - 3t^2 + 1$
$\varphi_4^k(t)$	$4480t^4 - 10360t^3 + 7890t^2 - 2150t + 140$	$-6t^2 + 6t$

Table 8: Trial and test functions with real coefficients over the master element  $[0, 1]$  that lead to a four-stage Runge-Kutta method.



(a) Trial functions of the first real solution. (b) Test functions of the first real solution.

Figure 9: Trial and test functions over the master element  $[0, 1]$  for the real solutions of the four-stage Runge-Kutta method.

$\mathcal{V}_\tau \subset \mathcal{V}$  and adding both equations in (3), we have that

$$B(v_\tau, u) + (v_\tau(0), u(0)) = B_{DG}(v_\tau, u), \quad \forall v_\tau \in \mathcal{V}_\tau,$$



where  $B_{DG}(\cdot, \cdot)$  is the bilinear form defined in (8). Since  $u \in \mathcal{U}$  is a globally continuous function in time, the jumps at each time interface are zero. Therefore, we consider the following primal problems:

$$\begin{cases} \text{Find } u \in \mathcal{U} \text{ and } u_{\tau h} \in \tilde{\mathcal{U}}_{\tau h} \text{ such that} \\ B_{DG}(v_\tau, u) = F(v_\tau), \quad \forall v_\tau \in \mathcal{V}_\tau \subset \mathcal{V}, \\ B_{DG}(v_{\tau h}, u_{\tau h}) = F(v_{\tau h}), \quad \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}. \end{cases} \quad (30)$$

Now, as we explained in Section 3,  $\mathcal{U}_\tau \not\subset \mathcal{U}$  and  $\mathcal{U} \not\subset \mathcal{U}_\tau$ . Then, we define the following space

$$\widehat{\mathcal{U}}_\tau := \{u \in \mathcal{V} \mid u|_{I_k} \in L^2(I_k; V), \quad u_{t|I_k} \in L^2(I_k; V'), \quad \forall k = 1, \dots, m\},$$

which is the minimum subspace of  $\mathcal{V}$  containing  $\mathcal{U}_\tau$  and  $\mathcal{U}$  [39].

In the goal-oriented approach, given a linear functional  $L(\cdot) : \mathcal{U} \subset \mathcal{V} \rightarrow \mathbb{R}$  of the form

$$L(v) = \int_I \langle v, g \rangle dt + (v(T), z_T), \quad (31)$$

the following dual problem is introduced: Find  $z \in \mathcal{U}$  such that

$$\begin{cases} - \int_I \langle v, z_t \rangle dt + \int_I (\nabla v, \nabla z) dt = \int_I \langle v, g \rangle dt, \quad \forall v \in \mathcal{V}, \\ (\hat{v}, z(T)) = (\hat{v}, z_T), \quad \forall \hat{v} \in L^2(\Omega), \end{cases} \quad (32)$$

where  $g \in \mathcal{V}'$  and  $z_T \in L^2(\Omega)$  are given functions. It is well known that due to integration by parts in time, the dual problem runs backwards in time [5].

Similarly as for the primal problem, (32) holds for every subspace of  $\mathcal{V}$ . So selecting  $\widehat{\mathcal{U}}_\tau \subset \mathcal{V}$ , adding both equations of (32) and taking into account that the jumps of  $z$  at each time interface are zero, we consider:

$$\begin{cases} \text{Find } z \in \mathcal{U} \text{ such that} \\ B_{DG}^*(\hat{v}_\tau, z) = L(\hat{v}_\tau), \quad \forall \hat{v}_\tau \in \widehat{\mathcal{U}}_\tau \subset \mathcal{V}, \end{cases} \quad (33)$$

where  $B_{DG}^*(\cdot, \cdot)$  is the adjoint operator of  $B_{DG}(\cdot, \cdot)$  obtained after integration by parts in time. Now, we define the error of the primal problem as  $e = u - u_{\tau h}$  and we represent the error in the quantity of interest in the following theorem.

**Theorem 2.** (*Error representation*) Let  $B_{DG}(\cdot, \cdot)$  be the bilinear form defined in Eq. (8),  $L(\cdot)$  the linear output functional defined in Eq. (31) and  $e = u - u_{\tau h}$  the error of the primal problem. It holds that

$$L(e) = B_{DG}(z - v_{\tau h}, e) = F(z - v_{\tau h}) - B_{DG}(z - v_{\tau h}, u_{\tau h}), \quad \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}.$$

*Proof.* We have that  $u \in \mathcal{U} \subset \widehat{\mathcal{U}}_\tau$  and  $u_{\tau h} \in \widetilde{\mathcal{U}}_{\tau h} \subset \mathcal{U}_\tau \subset \widehat{\mathcal{U}}_\tau$  so  $e = u - u_{\tau h} \in \widehat{\mathcal{U}}_\tau$ . Substituting  $e$  into (33) and integrating by parts in time, we obtain

$$L(e) = B_{DG}^*(e, z) = B_{DG}(z, e). \quad (34)$$

Now, as  $\widetilde{\mathcal{V}}_{\tau h}$  is a subspace of  $\mathcal{V}_\tau$ , the first equation of (30) also holds for all functions in  $\widetilde{\mathcal{V}}_{\tau h}$ . Therefore, substituting  $v_{\tau h}$  in (30) and subtracting both equations, we obtain the following consistency result (commonly called Galerkin orthogonality of the error)

$$B_{DG}(v_{\tau h}, e) = 0, \quad \forall v_{\tau h} \in \widetilde{\mathcal{V}}_{\tau h}. \quad (35)$$

Finally, subtracting (35) from (34) and employing the definition of problem (30), we obtain

$$L(e) = B_{DG}(z - v_{\tau h}, e) = F(z - v_{\tau h}) - B_{DG}(z - v_{\tau h}, u_{\tau h}), \quad \forall v_{\tau h} \in \widetilde{\mathcal{V}}_{\tau h}. \quad (36)$$

□

Formula (36) is the error in the quantity of interest represented as an integral over the whole space-time domain  $\Omega \times I$ . This quantity is usually bounded by the sum of local element contributions which will drive the goal-oriented adaptive process. However, (36) involves the analytical solutions of the primal and dual problems that are usually unknown. There exist several strategies in the literature [5, 7, 17, 44, 47] to define computable local error contributions of equation (36). For example, following [37, 47], we can employ the Forward Euler method (constant trial and test functions in time) to solve both primal and dual problems, where we solve the dual problem in a finer mesh in space  $z \sim z_{\tau \frac{h}{2}}$ . Then, we can localize the error estimator to the basis of each spatial mesh  $V_{\frac{h}{2}}^k, \forall k = 0, \dots, m$ . Another option is to follow a similar strategy to [6]. We can solve the primal and dual problems ( $u_{\tau h}$  and  $z_{\tau h}$ ) with a second-order Runge-Kutta method (linear trial and incomplete quadratic test functions). Then, we could perform a local high-order post-processing both in space and time of  $z_{\tau h}$  to approximate  $z$ , and choose  $v_{\tau h}$  as the constant-in-time nodal interpolation of  $z_{\tau h}$ .

In conclusion, with the construction of the variational formulation and corresponding goal-oriented error representation described in this paper, we could reproduce most existing goal-oriented adaptive algorithms using *explicit* Runge-Kutta methods in time to solve the primal and dual problems.

*Example: Effectivity index*

In order to verify numerically that the error representation (36) can be employed for goal-oriented adaptivity, we calculate the effectivity index

$$I_{eff} = \frac{Est}{L(e)}, \quad (37)$$

where  $L(e)$  is the true error in the quantity of interest and  $Est$  is the estimated error when we approximate the exact primal and dual solutions in (36). It is desirable to have an effectivity index close to 1.

In (1), we consider the test case where  $d = 1$ ,  $\Omega = (0, 1)$ ,  $T = 0.05$ ,  $f = 0$ , and  $u_0(x) = \sin(\pi x)$ . Then, the exact solution is

$$u(x, t) = e^{-\pi t} \sin(\pi x).$$

We select the quantity of interest defined in (31) with  $g = 0$  and  $z_T = \sin(\pi x)$ , i.e.,

$$L(u) = (u(T), z_T).$$

In (36), we choose  $z_{\tau h}$  as the solution of the dual problem in a coarse mesh and we approximate the exact primal and dual solutions in a finer mesh in space and time. Then, we obtain

$$L(e) \sim Est = B_{DG}(e_{\tau h}, \varepsilon_{\tau h}),$$

where

$$e_{\tau h} = u_{\frac{\tau}{2} \frac{h}{2}} - u_{\tau h}, \quad \varepsilon_{\tau h} = z_{\frac{\tau}{2} \frac{h}{2}} - z_{\tau h}.$$

In Table 9, we show the effectivity index (37) when performing global space-time refinements. In order to fulfill the CFL condition, when we double the number of space elements in space we increase the number of time steps by four. For the discretization in space, we employ a FEM with piecewise linear functions. In time, we solve both primal and dual problems with the Forward Euler method and the explicit trapezoidal rule (or Heun's method, a two-stage explicit Runge-Kutta method). Figure 10 shows the convergence of the true and the estimated errors for both methods in time.

Elements in space	Time steps	$I_{eff}$ Forward Euler	$I_{eff}$ Explicit Trapezoidal
4	20	0.70978	0.75706
8	80	0.70366	0.75173
16	320	0.70212	0.75043

32	1,280	0.70174	0.75011
64	5,120	0.70164	0.75003
128	20,480	0.70162	0.75001

Table 9: Effectivity index for the Forward Euler method and the explicit trapezoidal rule when performing uniform space-time refinements.

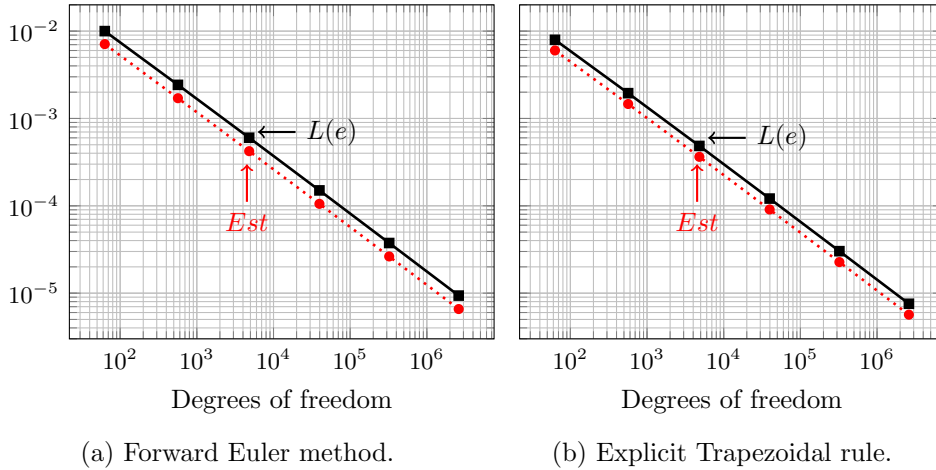


Figure 10: Convergence of the true and the estimated errors for the Forward Euler method (left) and the explicit trapezoidal rule (right).

We observe that, for this test case, the effectivity index for the Forward Euler method is  $I_{eff} \sim 0.7$  and for the explicit trapezoidal rule is  $I_{eff} \sim 0.75$ .

We now select an example with larger variations in time. We consider the test case where  $d = 1$ ,  $\Omega = (0, 1)$ ,  $T = 0.05$  and the initial condition  $u_0$  and the source  $f$  in such a way that the exact solution is

$$u(x, t) = \cos(100\pi t) \sin(\pi x).$$

In Table 10, we show the effectivity index (37) when performing global space-time refinements. Figure 11 shows the convergence of the true and the estimated errors for the Forward Euler and the explicit trapezoidal methods.

Elements in space	Time steps	$I_{eff}$ Forward Euler	$I_{eff}$ Explicit Trapezoidal
4	20	0.50483	0.73029
8	80	0.50073	0.73033
16	320	0.49943	0.73909

32	1,280	0.49908	0.74624
64	5,120	0.499	0.74896
128	20,480	0.49897	0.74973

Table 10: Effectivity index for the Forward Euler method and the explicit trapezoidal rule when performing uniform space-time refinements.

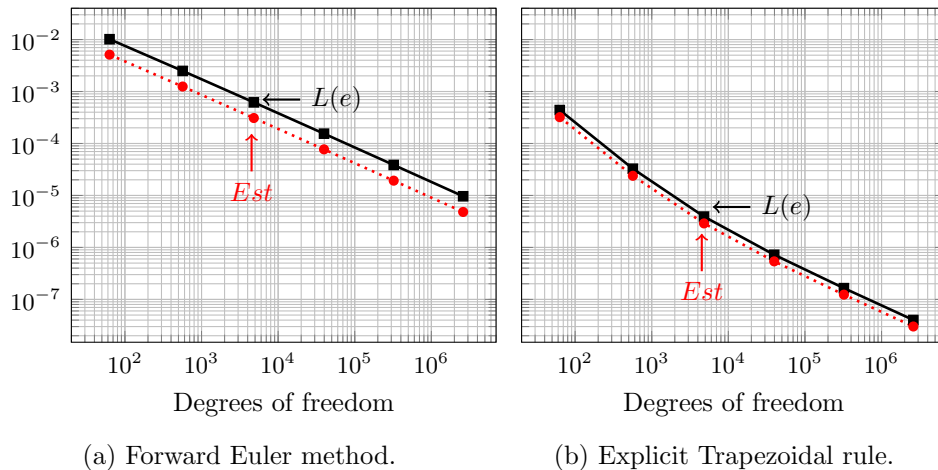


Figure 11: Convergence of the true and the estimated errors for the Forward Euler method (left) and the explicit trapezoidal rule (right).

We conclude that, in this example, the effectivity index for the Forward Euler method is  $I_{eff} \sim 0.5$  and for the explicit trapezoidal rule is  $I_{eff} \sim 0.75$ . Therefore, it seems the error representation (36) could be successfully employed to construct and apply explicit-in-time goal-oriented adaptive algorithms.

## 6. Conclusions

We propose a discontinuous-in-time Petrov-Galerkin formulation of the linear diffusion equation that, after exact integration in time, leads to explicit Runge-Kutta methods. We define families of piecewise polynomials for trial and test functions for any stage Runge-Kutta method where the trial and test spaces are defined as the span of certain polynomial functions. We provide explicit examples for Runge-Kutta methods of up to four stages. When the trial functions are polynomials of order  $p$  in time, then the test space is formed by incomplete polynomial spaces of order  $p + 1$ . Alternatively, we can define the test space to be a complete polynomial space of

order  $p$  and the resulting trial space will contain some polynomials of order  $p+1$ , but in this case we lose approximability of the solution. Methods with more than two stages result in systems which have solutions with complex coefficients for the polynomial basis, thus leading to equivalent Runge-Kutta methods of the appropriate order.

We present a constructive method to systematically build trial and test functions corresponding to explicit Runge-Kutta methods in time. Employing this formulation, we can naturally represent the error in the quantity of interest. Therefore, this variational structure reproduces the existing goal-oriented space-time adaptive algorithms but employing explicit-in-time Runge-Kutta schemes. A limitation of our method is that, for a large number of stages, we end up with large nonlinear systems of equations that are difficult to solve. However, following the same construction, we could fix the basis functions of the trial space (being or not a complete space of piecewise-discontinuous polynomials). Then, we would obtain a linear system for the coefficients of the test functions. Finally, defining the minimum orthogonality conditions to obtain an explicit method, we can compute test functions that might lead to new families of explicit methods (not necessarily Runge-Kutta ones).

As future work, we plan to design explicit-in-time goal-oriented adaptive strategies for parabolic problems. We will also analyze the stability of the new time marching schemes arising from our Galerkin construction in order to build more stable explicit methods. The presented formulation could be useful to study the variational structure of other implicit and explicit methods such as Adams-Bashforth, Adams-Moulton or Backward Differentiation Formulas (BDF). Finally, we will explore ideas from the DPG [16] community to the construction presented in this article.

## Appendix A. Matrix form of the nonlinear system

In this section we express (29) in matrix form. We consider, for example,  $s$ -stages and trial and test functions of order  $s$  over the master element  $[0, 1]$

$$\left\{ \begin{array}{l} \phi_1(t) = c_{10} + c_{11}t + \dots + c_{1s}t^s, \\ \vdots \\ \phi_s(t) = c_{s0} + c_{s1}t + \dots + c_{ss}t^s, \\ \varphi_1(t) = d_{10} + d_{11}t + \dots + d_{1s}t^s, \\ \vdots \\ \varphi_s(t) = d_{s0} + d_{s1}t + \dots + d_{ss}t^s. \end{array} \right.$$

To simplify notation, we collect the entries into the following matrices

$$C := \begin{bmatrix} c_{10} & c_{11} & \cdots & c_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{s0} & c_{s1} & \cdots & c_{ss} \end{bmatrix},$$

$$D := \begin{bmatrix} d_{10} & d_{11} & \cdots & d_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ d_{s0} & d_{s1} & \cdots & d_{ss} \end{bmatrix},$$

thus, we can write conditions (29) in matrix form as

$$\begin{cases} C \mathbf{e}_{1,s+1} = \mathbf{e}_{1,s}, \\ D \mathbf{1}_{s+1} = \mathbf{e}_{1,s}, \\ DAC^T - BC^T = E, \\ DFC^T = G, \end{cases} \quad (\text{A.1})$$

where

$$\begin{aligned} \mathbf{e}_{1,s+1} &:= [1 \ 0 \ \cdots \ 0]^T, \quad \mathbf{1}_{s+1} := [1 \ 1 \ \cdots \ 1]^T, \\ A &:= \begin{bmatrix} 0 & 1 & \cdots & 1 & 1 \\ 0 & 1/2 & \cdots & (s-1)/s & s/(s+1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1/s & \cdots & 1/2 & s/(2s-1) \\ 0 & 1/(s+1) & \cdots & (s-1)/(2s-1) & 1/2 \end{bmatrix}, \\ B &:= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad E := \begin{bmatrix} -1 & 0 & \cdots & 0 \\ -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & 1 \end{bmatrix}, \end{aligned}$$

$$F := \begin{bmatrix} 1 & 1/2 & \cdots & 1/s & 1/(s+1) \\ 1/2 & 1/3 & \cdots & 1/(s+1) & 1/(s+2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1/s & 1/(s+1) & \cdots & 1/(2s-1) & 1/2s \\ 1/(s+1) & 1/(s+2) & \cdots & 1/2s & 1/(2s+1) \end{bmatrix},$$

$$G := \begin{bmatrix} b_1 & b_2 & \cdots & b_s \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \cdots & a_{ss} \end{bmatrix}.$$

We compute the entries in the matrices  $A$  and  $F$  from

$$\begin{aligned} A &= \int_0^1 \begin{bmatrix} 1 \\ t \\ \vdots \\ t^{s-1} \\ t^s \end{bmatrix} \begin{bmatrix} 0 & 1 & \cdots & (s-1)t^{s-2} & st^{s-1} \end{bmatrix} dt \\ &= \int_0^1 \begin{bmatrix} 0 & 1 & \cdots & (s-1)t^{s-2} & st^{s-1} \\ 0 & t & \cdots & (s-1)t^{s-1} & st^s \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & t^{s-1} & \cdots & (s-1)t^{2s-3} & st^{2s-2} \\ 0 & t^s & \cdots & (s-1)t^{2s-2} & st^{2s-1} \end{bmatrix} dt \\ &= \begin{bmatrix} 0 & 1 & \cdots & 1 & 1 \\ 0 & 1/2 & \cdots & (s-1)/s & s/(s+1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1/s & \cdots & 1/2 & s/(2s-1) \\ 0 & 1/(s+1) & \cdots & (s-1)/(2s-1) & 1/2 \end{bmatrix}, \end{aligned}$$



$$\begin{aligned}
F &= \int_0^1 \begin{bmatrix} 1 \\ t \\ \vdots \\ t^{s-1} \\ t^s \end{bmatrix} \begin{bmatrix} 1 & t & \dots & t^{s-1} & t^s \end{bmatrix} dt \\
&= \int_0^1 \begin{bmatrix} 1 & t & \dots & t^{s-1} & t^s \\ t & t^2 & \dots & t^s & t^{s+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t^{s-1} & t^s & \dots & t^{2s-2} & t^{2s-1} \\ t^s & t^{s+1} & \dots & t^{2s-1} & t^{2s} \end{bmatrix} dt \\
&= \begin{bmatrix} 1 & 1/2 & \dots & 1/s & 1/(s+1) \\ 1/2 & 1/3 & \dots & 1/(s+1) & 1/(s+2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1/s & 1/(s+1) & \dots & 1/(2s-1) & 1/2s \\ 1/(s+1) & 1/(s+2) & \dots & 1/2s & 1/(2s+1) \end{bmatrix}.
\end{aligned}$$

## Appendix B. MATLAB Code

In this section we provide a MATLAB code to solve system (29) in matrix form for a general number of stages.

```

1 %Script to calculate the trial and test functions of
  Runge-Kutta methods
2 %Import data
3 [n_fun, p_trial, p_test, V_triv, V_deriv, V_grad]=data;
4 %Initialize the solution
5 S_trial=cell(n_fun, p_trial+1);
6 S_test=cell(n_fun, p_test+1);
7 %Write the coefficients
8 coef_trial=sym('c%d%d', [n_fun p_trial+1]);
9 coef_test=sym('d%d%d', [n_fun p_test+1]);

```

```

10 %Write the conditions
11 [C_triv , C_deriv , C_grad]=conditions(n_fun , coef_trial ,
    coef_test , p_trial , p_test);
12 %Solve the nonlinear system
13 [S_trial {1:end} , S_test {1:end}]=solve([C_triv C_deriv
    C_grad]==[V_triv V_deriv V_grad] , [coef_trial
    coef_test]);

1 function [n_fun , p_trial , p_test , V_triv , V_deriv , V_grad]
    = data()
2 %Function to write the data
3 %Number of trial and test functions and their order
4 n_fun=2;
5 p_trial=2;
6 p_test=2;
7 %The value of the conditions. For example:
8 %2-stage Runge-Kutta method
9 V_triv=[1 1 ;0 0];
10 V_deriv=[-1 0;-1 1];
11 alpha=1;
12 V_grad=[1-1/(2*alpha) 1/(2*alpha); alpha 0];
13 %Forward Euler method
14 %V_triv=[1 1];
15 %V_deriv=-1;
16 %V_grad=1;
17 end

1 function [C_triv , C_deriv , C_grad]= conditions(n_fun ,
    coef_trial , coef_test , p_trial , p_test)
2 %Function to write the conditions in matrix form the
    trial and test functions must satisfy
3 C_trial=coef_trial*eye(p_trial+1,1);
4 C_test=coef_test*ones(p_test+1,1);
5 C_triv=[C_trial C_test];
6
7 C=repmat((1:p_test+1)',1,p_trial+1)+repmat(0:p_trial ,
    p_test+1,1);
8 C=1./C;
9 C_grad=coef_test*C*coef_trial.';

```

```

10
11 B=[ones(1,p_trial+1);zeros(n_fun-1,p_trial+1)];
12 A=repmat(0:p_trial,p_test+1,1).*[zeros(p_test+1,1)C
    (:,1:end-1)];
13 C_deriv=coef_test*A*coef_trial.-B*coef_trial.';
14 end

```

## Acknowledgments

All authors have received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 777778 (MATHROCKS).

David Pardo, Elisabete Alberdi and Judit Muñoz-Matute were partially funded by the Basque Government Consolidated Research Group Grant IT649-13 on “Mathematical Modeling, Simulation, and Industrial Applications (M2SI)” and the Projects of the Spanish Ministry of Economy and Competitiveness with reference MTM2016-76329-R (AEI/FEDER, EU), and MTM2016-81697-ERC.

David Pardo has also received funding from the BCAM “Severo Ochoa” accreditation of excellence SEV-2013-0323 and the Basque Government through the BERC 2014-2017 program.

Victor M. Calo was partially funded by the CSIRO Professorial Chair in Computational Geoscience at Curtin University, the Mega-grant of the Russian Federation Government (N 14.Y26.31.0013) and the Deep Earth Imaging Enterprise Future Science Platforms of the Commonwealth Scientific Industrial Research Organisation, CSIRO, of Australia. Additional support was provided at Curtin University by The Institute for Geoscience Research (TIGeR) and by the Curtin Institute for Computation.

Judit Muñoz-Matute has received funding from the University of the Basque Country (UPV/EHU) grant No. PIF15/346.

All authors want to thank Prof. Alexandre Ern from CERMICS at École des Ponts ParisTech for the very fruitful discussions on the topic.

## References

- [1] R. Abedi, R. B. Haber, S. Thite, and J. Erickson. An h-adaptive spacetime-discontinuous Galerkin method for linear elastodynamics. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(6):619–642, 2006.

- [2] R. Abedi, B. Petracovici, and R. B. Haber. A space-time discontinuous Galerkin method for linearized elastodynamics with element-wise momentum balance. *Computer Methods in Applied Mechanics and Engineering*, 195(25):3247–3273, 2006.
- [3] N. Ahmed and V. John. Adaptive time step control for higher order variational time discretizations applied to convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 285:83–101, 2015.
- [4] A. Aziz and P. Monk. Continuous finite elements in space and time for the heat equation. *Mathematics of Computation*, 52(186):255–274, 1989.
- [5] W. Bangerth, M. Geiger, and R. Rannacher. Adaptive Galerkin finite element methods for the wave equation. *Computational Methods in Applied Mathematics*, 10(1):3–48, 2010.
- [6] W. Bangerth and R. Rannacher. Adaptive finite element techniques for the acoustic wave equation. *Journal of Computational Acoustics*, 9(2):575–591, 2001.
- [7] M. Besier and R. Rannacher. Goal-oriented space-time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow. *International Journal for Numerical Methods in Fluids*, 70(9):1139–1166, 2012.
- [8] C. L. Bottasso. A new look at finite elements in time: a variational interpretation of Runge-Kutta methods. *Applied Numerical Mathematics*, 25(4):355–368, 1997.
- [9] J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2008.
- [10] J. H. Chaudhry, D. Estep, V. Ginting, J. N. Shadid, and S. Tavener. A posteriori error analysis of IMEX multi-step time integration methods for advection–diffusion–reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 285:730–751, 2015.
- [11] Z. Chen and J. Feng. An adaptive finite element algorithm with reliable and efficient error control for linear parabolic problems. *Mathematics of Computation*, 73(247):1167–1193, 2004.

- [12] N. Collier, H. Radwan, L. Dalcin, and V. M. Calo. Diffusive wave approximation to the shallow water equations: computational approach. *Procedia Computer Science*, 4:1828–1833, 2011.
- [13] J. Collins, D. Estep, and S. Tavener. A posteriori error analysis for finite element methods with projection operators as applied to explicit time integration techniques. *BIT Numerical Mathematics*, 55(4):1017–1042, 2015.
- [14] M. Delfour and F. Dubeau. Discontinuous polynomial approximations in the theory of one-step, hybrid and multistep methods for nonlinear ordinary differential equations. *Mathematics of Computation*, 47(175):169–189, 1986.
- [15] M. Delfour, W. Hager, and F. Trochu. Discontinuous Galerkin methods for ordinary differential equations. *Mathematics of Computation*, 36(154):455–473, 1981.
- [16] L. F. Demkowicz and J. Gopalakrishnan. An overview of the discontinuous Petrov Galerkin method. In *Recent developments in discontinuous Galerkin finite element methods for partial differential equations*, pages 149–180. Springer, 2014.
- [17] P. Díez and G. Calderón. Goal-oriented error estimation for transient parabolic problems. *Computational Mechanics*, 39(5):631–646, 2007.
- [18] H. Egger, F. Kretzschmar, S. M. Schnepp, and T. Weiland. A space-time discontinuous Galerkin-Trefftz method for time dependent Maxwell’s equations. *SIAM Journal on Scientific Computing*, 37(5):B689–B711, 2015.
- [19] J. Erickson, D. Guoy, J. M. Sullivan, and A. Üngör. Building spacetime meshes over arbitrary spatial domains. *Engineering with Computers*, 20(4):342–353, 2005.
- [20] K. Eriksson and C. Johnson. Error estimates and automatic time step control for nonlinear parabolic problems, I. *SIAM Journal on Numerical Analysis*, 24(1):12–23, 1987.
- [21] A. Ern, I. Smears, and M. Vohralík. Guaranteed, locally space-time efficient, and polynomial-degree robust a posteriori error estimates for high-order discretizations of parabolic problems. *SIAM Journal on Numerical Analysis*, 55(6):2811–2834, 2017.

- [22] D. Estep. A posteriori error bounds and global error control for approximation of ordinary differential equations. *SIAM Journal on Numerical Analysis*, 32(1):1–48, 1995.
- [23] D. Estep and D. French. Global error control for the continuous Galerkin finite element method for ordinary differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 28(7):815–852, 1994.
- [24] D. Estep and A. Stuart. The dynamical behavior of the discontinuous Galerkin method and related difference schemes. *Mathematics of Computation*, 71(239):1075–1103, 2002.
- [25] D. A. French. A space-time finite element method for the wave equation. *Computer Methods in Applied Mechanics and Engineering*, 107(1):145–157, 1993.
- [26] E. Gagarina, V. R. Ambati, S. Nurijanyan, J. J. van der Vegt, and O. Bokhove. On variational and symplectic time integrators for Hamiltonian systems. *Journal of computational physics*, 306:370–389, 2016.
- [27] H. Gómez, V. M. Calo, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4333–4352, 2008.
- [28] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.
- [29] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.
- [30] T. J. R. Hughes and G. M. Hulbert. Space-time finite element methods for elastodynamics: formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66(3):339–363, 1988.
- [31] G. M. Hulbert and T. J. R. Hughes. Space-time finite element methods for second-order hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, 84(3):327–348, 1990.

- [32] B. L. Hulme. One-step piecewise polynomial Galerkin methods for initial value problems. *Mathematics of Computation*, 26(118):415–426, 1972.
- [33] S. Hussain, F. Schieweck, and S. Turek. Higher-order Galerkin time discretizations and fast multigrid solvers for the heat equation. *Journal of Numerical Mathematics*, 19(1):41–61, 2011.
- [34] C. Johnson. Discontinuous Galerkin finite element methods for second order hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 107(1):117–129, 1993.
- [35] U. Köcher and M. Bause. Variational space-time methods for the wave equation. *Journal of Scientific Computing*, 61(2):424–453, 2014.
- [36] S. T. Miller and R. B. Haber. A spacetime discontinuous Galerkin method for hyperbolic heat conduction. *Computer Methods in Applied Mechanics and Engineering*, 198(2):194–209, 2008.
- [37] J. Muñoz-Matute, V. M. Calo, D. Pardo, E. Alberdi, and K. G. van der Zee. Explicit-in-time goal-oriented adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 347:176–200, 2019.
- [38] S. Nicaise and N. Soualem. A posteriori error estimates for a non-conforming finite element discretization of the heat equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(2):319–348, 2005.
- [39] N. Parés, P. Díez, and A. Huerta. Bounds of functional outputs for parabolic problems. Part I: Exact bounds of the discontinuous Galerkin time discretization. *Computer Methods in Applied Mechanics and Engineering*, 197(19):1641–1660, 2008.
- [40] C. Pozrikidis. *Introduction to finite and spectral element methods using MATLAB*. CRC Press, 2005.
- [41] S. Salsa. *Partial differential equations in action: from modelling to theory*. Springer Science & Business Media, 2008.
- [42] W. E. Schiesser. *The numerical method of lines: integration of partial differential equations*. Elsevier, 2012.
- [43] F. Schieweck. A-stable discontinuous Galerkin-Petrov time discretization of higher order. *Journal of Numerical Mathematics*, 18(1):25–57, 2010.

- [44] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM Journal on Scientific Computing*, 30(1):369–393, 2008.
- [45] D. Schötzau and C. Schwab. An hp a priori error analysis of the DG time-stepping method for initial value problems. *Calcolo*, 37(4):207–232, 2000.
- [46] D. Schötzau and C. Schwab. Time discretization of parabolic problems by the hp-version of the discontinuous Galerkin finite element method. *SIAM Journal on Numerical Analysis*, 38(3):837–875, 2000.
- [47] G. Şimşek, X. Wu, K. van der Zee, and E. van Brummelen. Duality-based two-level error estimation for time-dependent PDEs: application to linear and nonlinear parabolic equations. *Computer Methods in Applied Mechanics and Engineering*, 288:83–109, 2015.
- [48] W. Tang and Y. Sun. Time finite element methods: a unified framework for numerical discretizations of ODEs. *Applied Mathematics and Computation*, 219(4):2158–2179, 2012.
- [49] P. Vignal, N. Collier, L. Dalcin, D. Brown, and V. M. Calo. An energy-stable time-integrator for phase-field models. *Computer Methods in Applied Mechanics and Engineering*, 316:1179–1214, 2017.
- [50] T. Werder, K. Gerdes, D. Schötzau, and C. Schwab. hp-discontinuous Galerkin time stepping for parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 190(49):6685–6708, 2001.
- [51] X. Wu, K. G. van der Zee, G. Simsek, and E. Van Brummelen. A posteriori error estimation and adaptivity for nonlinear parabolic equations using IMEX-Galerkin discretization of primal and dual equations. *SIAM Journal on Scientific Computing*, 40(5):A3371–A3399, 2018.
- [52] S. Zhao and G.-W. Wei. A unified discontinuous Galerkin framework for time integration. *Mathematical methods in the applied sciences*, 37(7):1042–1071, 2014.