Explicit-in-Time Goal-Oriented Adaptivity

Judit Muñoz-Matute[a], Victor M. Calo[b,c], David Pardo[a,d,e],
Elisabete Alberdi[a], Kristoffer G. van der Zee[f]

[a] *University of the Basque Country (UPV/EHU), Leioa, Spain*
[b] *Applied Geology, Western Australian School of Mines, Faculty of Science and Engineering, Curtin University, Perth, WA, Australia 6845*
[c] *Mineral Resources, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Kensington, WA, Australia 6152*
[d] *BCAM-Basque Center for Applied Mathematics, Bilbao, Spain*
[e] *IKERBASQUE, Basque Foundation for Science, Bilbao, Spain*
[f] *School of Mathematical Sciences, University of Nottingham, University Park, NG7 2RD Nottingham, United Kingdom*

Abstract

Goal-oriented adaptivity is a powerful tool to accurately approximate physically relevant solution features for Partial Differential Equations. In time dependent problems, we seek to represent the error in the quantity of interest as an integral over the whole space-time domain. A full space-time variational formulation allows such representation. Most authors employ implicit time marching schemes to perform goal-oriented adaptivity as it is known that they can be reinterpreted as Galerkin methods. In this work, we consider variational forms for explicit methods in time. We derive an appropriate error representation and propose a goal-oriented adaptive algorithm in space. For that, we derive the Forward Euler method in time employing a discontinuous-in-time Petrov-Galerkin formulation. In terms of time domain adaptivity, we impose the Courant-Friedrichs-Lewy condition to ensure the stability of the method. We provide some numerical results applied to the diffusion and the advection-diffusion equations to show the performance of the proposed explicit-in-time goal-oriented adaptive algorithm.

Keywords: linear advection-diffusion equation, goal-oriented adaptivity, explicit methods in time, error representation, Finite Element Method

1. Introduction

Adaptive processes for Partial Differential Equations (PDEs) are essential tools to produce optimal grids in order to obtain accurate solutions while minimizing the computational cost. In time-domain problems, there exist several adaptive algorithms based on space-time Finite Element Methods (FEM). For example, tent-pitching strategies [1–4] where space-time FEM is employed to build unstructured meshes. On the other hand, there are some algorithms that are often based on time marching schemes like $hp-$adaptive methods in both space and time [5–7] or goal-oriented adaptive algorithms [8, 9]. These methods assume that the trial and test basis functions have space-time tensor-product structure and selecting certain discontinuous-in-time basis functions, the space-time FEM becomes a time-marching scheme. However, to date only implicit methods in time are available in the literature, where their variational structure is exploited [10, 11]. Herein, our goal is to design a goal-oriented adaptive algorithm for time-dependent PDEs employing explicit methods in time as they are usually computationally cheaper than the implicit ones.

Goal-oriented adaptive algorithms address the accuracy requirements of some engineering problems, where it is crucial to accurately approximate some physical features of the solution [12–17]. These relevant features are called quantities of interest and they are usually represented by output functionals computed from the solution. The objective of these adaptive algorithms is to reduce the error in the quantity of interest [18–22]. For that, a dual problem is defined where the output functional becomes the source. Then, an error in the quantity of interest is represented as an integral over the whole domain in terms of the errors of the direct and dual problems. Finally, this error representation is bounded above by the sum of local element contributions that drive the goal-oriented adaptive process.

In time dependent problems, most authors discretize the primal and dual problems employing the Method of Lines (MoL) [23–26]. Here, starting from a variational formulation in space, the spatial variable is discretized employing the FEM. Then, the resulting system of Ordinary Differential Equations (ODEs) is solved employing numerical methods for ODEs like Backward Euler or Crank-Nicholson [27, 28]. However, for time-domain goal-oriented adaptivity, a full space-time variational formulation is needed to represent the error in the quantity of interest as an integral over the whole space-time domain [29]. Therefore, to discretize them using the MoL it is essential to know their equivalence with the corresponding space-time FEM. It is known that some low-order FEM in time are algebraically equivalent to some numerical methods for ODEs [30–32]. For instance, if we select piecewise constant-in-time functions for trial and test in the space-time variational formulation, we obtain the Backward Euler method in time. Employing piecewise linear basis function for trial and constants for test, the space-time FEM is equivalent to the Crank-Nicholson method in time.

The aforementioned methods are implicit in time so that is the reason why most authors employ implicit methods in time for goal-oriented adaptivity. In [10, 33, 34], goal-oriented adaptive strategies are explained for wave propagation phenomena; in [29, 35, 36] for parabolic problems; in [37, 38] the error representation is derived for structural transient dynamics; and finally, in [39, 40] the goal-oriented approach is extended to nonlinear problems. Recently, in [41], the authors expressed some multi-step Implicit-Explicit (IMEX) schemes as Galerkin methods in time for PDEs by using specific quadrature rules for time integration. Finally, in [42], we proved that explicit Runge-Kutta methods can be reinterpreted as discontinuous Petrov-Galerkin methods for linear parabolic problems. For any explicit Runge-Kutta method, we described an algorithm that builds the corresponding trial and test basis functions employing exact integration in time.

In this work, we derive a space-time error representation of the quantity of interest and develop a goal-oriented adaptive process in space employing explicit methods in time following the ideas presented in [42]. We allow dynamic meshes in space, that is, we consider different spatial discretizations per time interval. Then, we select piecewise constant trial and test functions. Finally, displacing in time the spatial discrete spaces of the test space with respect to those associated with the trial space, we show that we obtain the Forward Euler method. This displacement involves the use of different trial and test spaces, leading to a Petrov-Galerkin method. Moreover, as we select piecewise constant functions in time, we need a discontinuous Galerkin formulation, therefore, we employ a discontinuous Petrov-Galerkin formulation of the advection-diffusion equation. To discretize the dual problem, we shift the trial and test spaces from the primal problem and we also obtain the Forward Euler method but running backwards in time.

As we are solving the primal and dual problems with explicit methods in time, the Courant-Friedrichs-Lewy (CFL) condition [43] must be satisfied to ensure the stability of the discretization. Our adaptive strategy ensures the CFL condition by construction, as we start with a time step size that is in the limit of the CFL condition. We further assume (as it occurs in many applications) that once the CFL condition has been satisfied, most of the error is due to an inadequate space discretization. Thus, our adaptive scheme focuses on performing optimal goal-oriented space refinements followed by a simple refinement condition in time intended to recover the CFL condition after those space refinements. Therefore, employing the error in space of the primal and dual problems, we obtain the error representation that drives the goal-oriented adaptivity in space. Higher order explicit methods in time could be employed for time discretization employing the basis functions presented in [42]. While the focus of this article is performing local adaptivity in space, the space-time variational formulation of the problem also allows to perform local space-time refinements. We will study this possibility in the future.

The outline of this article is the following: Section 2 describes the strong and weak formulations of the linear advection-diffusion equation. In Section 3 we describe the discretization that leads to explicit methods in time for the primal problem. Section 4 introduces the dual problem and the discretization we use to obtain explicit methods running backwards in time. Section 5 shows the error representation in the goal-oriented framework. In Section 6, we describe the goal-oriented algorithm in space and complemented with CFL-based refinements in time. Section 7 presents the numerical results obtained with the explicit-in-time goal-oriented adaptive algorithm. Finally, Section 8 summarizes the conclusions and possible extensions of this work.

2. Model Problem

In this section, we describe the strong and weak formulations of the model problem we employ to develop the explicit in time goal-oriented adaptive method.

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 advection-diffusion equation*

$$\begin{cases} u_{,t} - \nabla \cdot (\nu \nabla u) + \beta \cdot \nabla u = f & \text{in } \Omega \times I, \\ u = 0 & \text{on } \partial\Omega \times I, \\ u(0) = u_0 & \text{in } \Omega, \end{cases} \tag{1}$$

where $u_{,t} := \partial u / \partial t$ and $\partial\Omega$ denotes the boundary of the spatial domain Ω. The source term $f(\mathbf{x}, t)$, the initial temperature distribution $u_0(\mathbf{x})$, the diffusion coefficient $\nu(\mathbf{x})$ and the velocity field $\beta(\mathbf{x})$ are given data.

We assume that β is a bounded divergence-free vector field

$$\nabla \cdot \beta = 0,$$

and ν is a bounded above and strictly positive function

$$0 < \nu_1 \leq |\nu(\mathbf{x})| \leq \nu_2, \ \forall \mathbf{x} \in \Omega.$$

For simplicity, we omit the spatial dependence of the functions, that is, we write $u(t)$ instead of $u(\mathbf{x}, t)$.

2.2. Weak formulation

We need a full space-time variational formulation of problem (1) in order to properly define a dual problem and to subsequently represent the error in the quantity of interest as an integral over the whole space-time domain.

From the diffusion term in (1) and the boundary conditions, we consider the following space

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

Therefore, a simple way to ensure sufficient integrability of the weak formulation consists of imposing that u_t and f should belong to the dual space of V, i.e., $V' = H^{-1}(\Omega)$. Now, we introduce 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. We denote as $\mathcal{V}' := L^2(I; V')$ the dual space of $\mathcal{V}$.

For the solution, we need $u \in \mathcal{V}$ but also $u_t \in \mathcal{V}'$, so we define the following trial space

$$\mathcal{U} := \{u \in \mathcal{V} \mid u_{,t} \in \mathcal{V}'\},$$

which is the space of all functions that are L^2 in time and also their derivatives, therefore, all functions in $\mathcal{U}$ are continuous in time.

Now, we multiply the advection-diffusion equation by the test functions $v \in \mathcal{V}$ and we integrate over the entire domain $\Omega \times I$. We also impose the initial condition in weak form. Finally, assuming that $f \in \mathcal{V}$ and $u_0 \in L^2(\Omega)$, the weak solution of problem (1) belongs to $\mathcal{U}$ and satisfies

$$\begin{aligned} \int_I \langle u_{,t}, v\rangle \, dt + \int_I (\nu \nabla u, \nabla v) \; dt + \int_I (\beta \cdot \nabla u, v) \; dt = \int_I \langle f, v\rangle \, dt,\ \forall v \in \mathcal{V}, \\ (u(0), w) = (u_0, w),\ \forall w \in L^2(\Omega). \end{aligned} \tag{2}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the duality pairing between the spaces V and V', $(\cdot, \cdot)$ is the inner product in $L^2(\Omega)$ and we also denote

$$B(u, v) := \int_I \langle u_{,t}, v\rangle \, dt + \int_I (\nu \nabla u, \nabla v) \; dt + \int_I (\beta \cdot \nabla u, v) \; dt. \tag{3}$$

3. Discretization

In this paper, we use finite elements for the spatial discretization. In time, we employ explicit methods and in order to derive them from (2), we need a *discontinuous Petrov-Galerkin formulation* of the problem.

3.1. Semidiscretization in time

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

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

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

We select the following subspace of $\mathcal{V}$

$$\mathcal{V}_\tau := \{v \in L^2(I; V) \mid v_{|_{(t_{k-1}, t_k]}} \in P_r((t_{k-1}, t_k]\,; V),\ \forall k = 1, \ldots, m,\ v(0) \in V\} \subset \mathcal{V},$$

where $P_r(I_k; V)$ is the space of all polynomials with degree less or equal than r on the interval I_k with values in V. The functions in $\mathcal{V}_\tau$ could be discontinuous at each time step t_k, $\forall k = 1, \ldots, m$. By selecting $\mathcal{V}_\tau$ for trial and test spaces and employing a standard discontinuous Galerkin formulation of the problem, we obtain implicit methods in time for different values of r [10].

However, we want to obtain explicit methods in time and, for that, we need to find the approximated solution of u in a space different from $\mathcal{V}_\tau$

$$\mathcal{U}_\tau := \{u \in L^2(I; V) \mid u_{|_{[t_{k-1}, t_k)}} \in P_r([t_{k-1}, t_k)\,; V),\ \forall k = 1, \ldots, m,\ u(T) \in V\} \subset \mathcal{V},$$

which is a more general space than $\mathcal{U}$, as the functions in $\mathcal{U}$ are continuous in time while functions in $\mathcal{U}_\tau$ could be discontinuous at each time step.

As we select discontinuous-in-time trial and test functions, we need a discontinuous Galerkin formulation of problem (2). Moreover, as the trial and test spaces $\mathcal{U}_\tau$ and $\mathcal{V}_\tau$ are different, we derive a discontinuous Petrov-Galerkin formulation of problem (2).

First, we define the jump of a function v at instant t_k as $[\![v]\!]^k := v(t_k^+) - v(t_k^-)$, where

$$v(t_k^+) := \lim_{s \longrightarrow 0^+} v(t_k + s), \ \ v(t_k^-) := \lim_{s \longrightarrow 0^+} v(t_k - s).$$

We integrate by parts in time over each subinterval I_k the bilinear form (3) and obtain

$$\int_{I_k} \Big(-\langle u, v_{,t} \rangle + (\nu \nabla u, \nabla v) + (\beta \cdot \nabla u, v) \Big) dt + (u(t_k^-), v(t_k^-)) - (u(t_{k-1}^+), v(t_{k-1}^+)).$$

Now, we perform a *downwind* approximation of the solution in each time interface, i.e., we substitute $u(t_k^-)$ by $u(t_k^+)$ in the above expression. Finally, integrating by parts in time again, we have

$$\int_{I_k} \Big(\langle u_{,t}, v\rangle + (\nu \nabla u, \nabla v) + (\beta \cdot \nabla u, v) \Big) dt + (u(t_k^+), v(t_k^-)) - (u(t_k^-), v(t_k^-)). \tag{4}$$

Therefore, summing up expression (4) for all time intervals and adding the initial condition, the discontinuous Petov-Galerkin formulation of (2) we propose is

$$\left\{ \begin{array}{l} \text{Find } u_\tau \in \mathcal{U}_\tau \text{ such that} \\ B_{_{DG}}(u_\tau, v_\tau) = F(v_\tau) \;\; \forall v_\tau \in \mathcal{V}_\tau, \end{array} \right. \tag{5}$$

where u_τ denotes the semidiscrete approximation of u and

$$\begin{aligned} B_{_{DG}}(u,v) :&= \sum_{k=1}^{m} \int_{I_k} \Big(\langle u_{,t}, v\rangle + (\nu \nabla u, \nabla v) + (\beta \cdot \nabla u, v) \Big) dt + \\ &+ \sum_{k=1}^{m} \Big([\![u]\!]^k, v(t_k^-) \Big) + (u(0^+), v(0^-)), \\ F(v) :&= \int_I \langle f, v\rangle \, dt + (u_0, v(0^-)), \end{aligned}$$

and $B_{_{DG}}(u,v) = B(u,v) + \sum_{k=1}^{m} \Big([\![u]\!]^k, v(t_k^-) \Big) + (u(0^+), v(0^-))$. Thus, the bilinear form $B_{_{DG}}(\cdot,\cdot)$ is a generalization of $B(\cdot,\cdot)$ for discontinuous-in-time functions. As the solution of (2) is continuous in time, then $[\![u]\!]^k = 0$, $\forall k = 1, \ldots, m$, so, it also satisfies problem (5).

3.2. Semidiscretization in space

We consider dynamic meshes in space so we define one discrete space per time step, i.e., $V_h^k \subset V$, $\forall k = 0, \ldots, m$. We also define the fully discrete trial and test spaces

$$\mathcal{U}_{\tau h} := \{ u \in L^2(I;V) \mid u_{|_{[t_{k-1}, t_k)}} \in P_r([t_{k-1}, t_k)\,; V_h^{k-1}), \; \forall k = 1, \ldots, m, \; u(T) \in V_h^m \} \subset \mathcal{U}_\tau,$$

$$\mathcal{V}_{\tau h} := \{ v \in L^2(I;V) \mid v_{|_{(t_{k-1}, t_k]}} \in P_r((t_{k-1}, t_k]\,; V_h^{k}), \; \forall k = 1, \ldots, m, \; v(0) \in V_h^0 \} \subset \mathcal{V}_\tau.$$

The functions of the trial space $\mathcal{U}_{\tau h}$ in I_k take values in V_h^{k-1}, while functions of the test space $\mathcal{V}_{\tau h}$ take values in V_h^k. Figure 1 illustrates this displacement of the spaces.

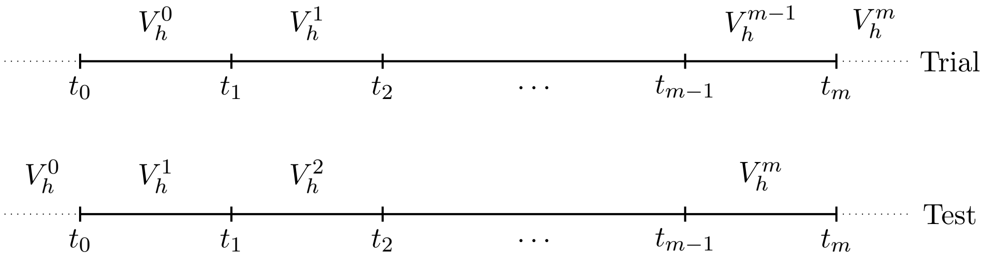

Figure 1: Illustration of the displacement in time of the trial (top) and test (bottom) discrete spaces.

Then, we define the fully discrete problem as

$$\left\{\begin{array}{l} \text{Find } u_{\tau h} \in \mathcal{U}_{\tau h} \text{ such that} \\ B_{DG}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}) \ \ \forall v_{\tau h} \in \mathcal{V}_{\tau h}. \end{array}\right. \tag{6}$$

3.3. Forward Euler method (r=0)

In this section, we show that selecting piecewise constant functions in time ($r = 0$), implies that problem (6) is algebraically equivalent to the Forward Euler method. Selecting different polynomials spaces in time for trial $\mathcal{U}_{\tau h}$ and test $\mathcal{V}_{\tau h}$, we obtain higher order explicit Runge-Kutta methods, as described in [42].

First, we assume that the trial and test basis functions exhibit a Cartesian product structure between time t and space $\mathbf{x}$. Then, we approximate the solution $u_{\tau h}$ as (see Figure 2)

$$u_{\tau h}(\mathbf{x}, t) = \sum_{k=1}^{m} u_h^{k-1}(\mathbf{x})\phi^{k-1}(t) + u_h^m(\mathbf{x})\chi^m(t), \tag{7}$$

where $u_h^k \in V_h^k, \ \forall k = 0, \ldots, m$, and

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

Similarly, we select the following test functions (see Figure 2)

$$v_h^0(\mathbf{x})\chi^0(t), \ v_h^k(\mathbf{x})\varphi^k(t), \ \ \forall k = 1, \ldots, m,$$

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

$$\chi^0(t) = \begin{cases} 1, \ t = 0, \\ 0, \ \text{elsewhere}, \end{cases} \qquad \varphi^k(t) = \begin{cases} 1, \ t \in (t_{k-1}, t_k], \\ 0, \ \text{elsewhere}. \end{cases}$$

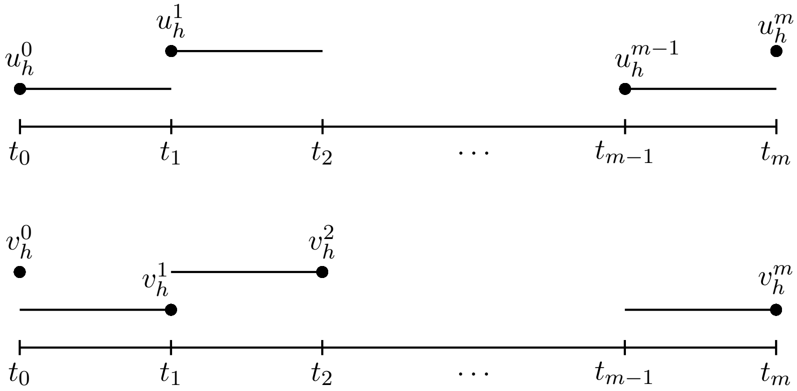

Figure 2: Trial and test functions for the primal problem when $r = 0$.

Since the test functions have local support in $(t_{k-1}, t_k]$ and are constant in time, we can split (6) into m local-in-time problems

$$\int_{I_k} \left\langle u_{\tau h,t}, v_h^k \right\rangle dt + \int_{I_k} \left(\nu \nabla u_{\tau h}, \nabla v_h^k) + (\beta \cdot \nabla u_{\tau h}, v_h^k) \right) dt + \left([\![u_{\tau h}]\!]^k, v_h^k \right) = \int_{I_k} \left\langle f, v_h^k \right\rangle dt, \tag{8}$$

$\forall k = 1, \ldots, m$ and for the initial condition we have $(u_h^0, v_h^0) = (u_0, v_h^0)$.

As the trial functions are constants in time, we have $u_{\tau h,t} = 0$. Also $u_{\tau h}(t_k^+) = u_h^k$ and $u_{\tau h}(t_k^-) = u_h^{k-1}$. Then, problems (8) become

$$\tau_k \left(\nu \nabla u_h^{k-1}, \nabla v_h^k) + (\beta \cdot \nabla u_h^{k-1}, v_h^k) \right) + \left(u_h^k - u_h^{k-1}, v_h^k \right) = \int_{I_k} \left\langle f, v_h^k \right\rangle dt, \tag{9}$$

$\forall k = 1, \ldots, m$.

Finally, we can write (6) as

$$\begin{cases} \text{Find } u_h^k \in V_h^k, \ \forall k = 1, \ldots, m, \text{ such that} \\ \\ \left(u_h^k, v_h^k \right) = \left(u_h^{k-1}, v_h^k \right) - \tau_k \left(\nu \nabla u_h^{k-1}, \nabla v_h^k \right) - \\ \qquad - \tau_k \left(\beta \cdot \nabla u_h^{k-1}, v_h^k \right) + \int_{I_k} \left\langle f, v_h^k \right\rangle dt, \ \forall v_h^k \in V_h^k, \\ \\ \left(u_h^0, v_h^0 \right) = \left(u_0, v_h^0 \right), \ \forall v_h^0 \in V_h^0, \end{cases} \tag{10}$$

which is the Forward Euler method in time with a modified source term.

To obtain the classical Forward Euler method, we can project the source term into the trial space as follows

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

where $f^k(\mathbf{x}) := f(\mathbf{x}, t_k)$. Then, as the functions $\phi^{k-1}(t)$ are constants over I_k, the source term in (10) becomes

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

As (10) is an explicit method, it is conditionally stable, which means that the time step size is constrained by the spatial resolution to keep the method stable. Thus, we ensure that the Courant-Friedrichs-Lewy (CFL) condition [43] is satisfied to recover proper stability of the method.

4. Output functionals and dual problem

We consider a linear functional $L : \mathcal{U} \subset \mathcal{V} \longrightarrow \mathbb{R}$, which is called the *output functional* and represents a physical quantity of interest computed from the solution.

We consider linear output functionals of the form

$$L(v) = \int_I \langle g, v \rangle \, dt + (z_{_T}, v(T)),$$

where $g \in \mathcal{V}'$ and $z_{_T} \in L^2(\Omega)$ are given functions.

We now introduce a *dual problem* that gives information about how the error committed in every space-time point affects the error in the quantity of interest

$$\begin{aligned} -\int_I \langle z_{,t}, v \rangle \, dt + \int_I (\nu \nabla z, \nabla v) dt - \int_I (\beta \cdot \nabla z, v) dt =& \int_I \langle g, v \rangle \, dt, \ \forall v \in \mathcal{V}, \\ (z(T), w) =& (z_T, w), \ \forall w \in L^2(\Omega), \end{aligned} \tag{11}$$

where we denote as

$$B^*(z,v) := -\int_I \langle z_{,t}, v \rangle \, dt + \int_I (\nu \nabla z, \nabla v) dt - \int_I (\beta \cdot \nabla z, v) dt,$$

the adjoint operator of $B(\cdot,\cdot)$.

Dual problem (11) is often employed to represent the error in the quantity of interest [44]. For this particular case, the corresponding strong formulation of (11) is

$$\begin{cases} -z_{,t} - \nabla \cdot (\nu \nabla z) - \beta \cdot \nabla z = g & \text{in} \quad \Omega \times I, \\ z = 0 & \text{in} \quad \partial\Omega \times I, \\ z(T) = z_{_T} & \text{in} \quad \Omega, \end{cases} \tag{12}$$

where (12) states that the dual problem runs backwards in time.

4.1. Discretization of the dual problem

In the same way as for the primal problem, we solve the dual problem (11) employing an explicit method in time. The Discontinuous Galerkin formulation of problem (11) we propose is

$$\begin{cases} \text{Find } z_\tau \in \mathcal{V}_\tau \text{ such that} \\ B^*_{_{DG}}(z_\tau, v) = L(v) \;\; \forall v \in \mathcal{U}_\tau, \end{cases} \tag{13}$$

where $B^*_{_{DG}}(\cdot,\cdot)$ is the resulting bilinear form after integrating by parts the time derivative and the space advection terms of the form $B_{_{DG}}(\cdot,\cdot)$ (see Appendix B for details)

$$\begin{aligned} B^*_{_{DG}}(z, v) :=& \sum_{k=1}^{m} \int_{I_k} \Big(- \langle z_{,t}, v \rangle + (\nu \nabla z, \nabla v) - (\beta \cdot \nabla z, v) \Big) dt \\ & - \sum_{k=1}^{m} \Big([\![z]\!]^{k-1}, v(t_{k-1}^+) \Big) + (z(T^-), v(T^+)). \end{aligned}$$

Now, we select the same subspaces as in the primal problem and we define the fully discrete problem

$$\begin{cases} \text{Find } z_{\tau h} \in \mathcal{V}_{\tau h} \text{ such that} \\ B^*_{_{DG}}(z_{\tau h}, v_{\tau h}) = L(v_{\tau h}), \;\; \forall v_{\tau h} \in \mathcal{U}_{\tau h}. \end{cases} \tag{14}$$

In (14), we shift the trial and test spaces for the discrete dual problem (14) with respect to those of the discrete primal problem (6).

4.2. Forward Euler method backwards in time (r=0)

We show that selecting constant functions in time ($r = 0$), problem (14) is also algebraically equivalent to the Forward Euler method but running backwards in time.

We assume that the trial and test functions have tensor product structure and we approximate the solution $z_{\tau h}$ as (see Figure 3)

$$z_{\tau h}(\mathbf{x},t) = z_h^0(\mathbf{x})\chi^0(t) + \sum_{k=1}^{m} z_h^k(\mathbf{x})\varphi^k(t), \tag{15}$$

where $z_h^k \in V_h^k, \ \forall k = 0,\dots,m$.

We select the following test functions (see Figure 3)

$$v_h^{k-1}(\mathbf{x})\phi^{k-1}(t), \ \ \forall k = 1,\dots,m, \ \ v_h^m(\mathbf{x})\chi^m(t),$$

where $v_h^k \in V_h^k, \ \ \forall k = 0,\dots,m$. Here, functions $\varphi^k(t)$, $\phi^{k-1}(t)$, $\chi^0(t)$ and $\chi^m(t)$ are the same as those we defined for the primal problem in Section 3.3.

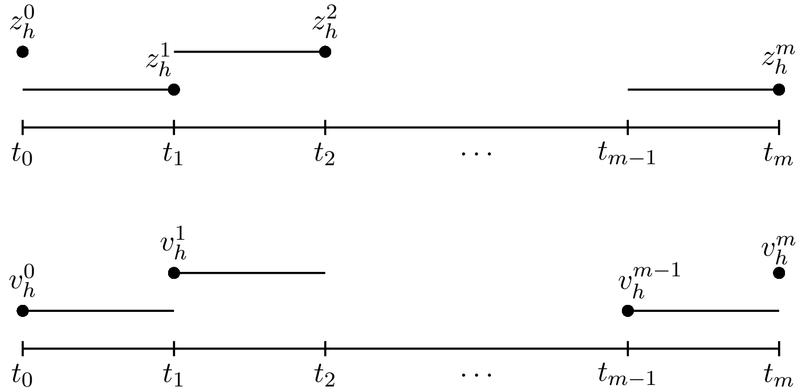

Figure 3: Trial and test functions for the dual problem when $r = 0$.

Since the test functions have local support in $[t_{k-1}, t_k)$ and are constant in time, we can split (14) in m local-in-time problems

$$\begin{aligned}\int_{I_k} -\left\langle z_{\tau h,t}, v_h^{k-1}\right\rangle dt + \int_{I_k}\left(\nu\nabla z_{\tau h}, \nabla v_h^{k-1}) - (\beta\cdot\nabla z_{\tau h}, v_h^{k-1})\right)dt - \left([\![z_{\tau h}]\!]^{k-1}, v_h^{k-1}\right) = \\ = \int_{I_k}\left\langle g, v_h^{k-1}\right\rangle dt,\end{aligned} \tag{16}$$

$\forall k = 1,\dots,m$ and we have the final condition $(z_h^m, v_h^m) = (z_{\scriptscriptstyle T}, v_h^m)$.

As before, the trial functions are constant in time. Thus, we have $z_{\tau h,t} = 0$. Since $z_{\tau h}(t_{k-1}^{+}) = z_h^k$ and $z_{\tau h}(t_{k-1}^{-}) = z_h^{k-1}$, problems (16) become

$$\tau_k\left(\nu\nabla z_h^k, \nabla v_h^{k-1}) - (\beta\cdot\nabla z_h^k, v_h^{k-1})\right) - \left(z_h^k - z_h^{k-1}, v_h^{k-1}\right) = \int_{I_k}\left\langle g, v_h^{k-1}\right\rangle dt. \tag{17}$$

Finally, the fully discrete problem (14) becomes

$$\left\{\begin{aligned} &\text{Find } z_h^{k-1}\in V_h^{k-1},\ \forall k = m,\ldots,1, \text{ such that}\\ &\left(z_h^{k-1}, v_h^{k-1}\right) = \left(z_h^k, v_h^{k-1}\right) - \tau_k\left(\nu\nabla z_h^k, \nabla v_h^{k-1}\right) + \tau_k\left(\beta\cdot\nabla z_h^k, v_h^{k-1}\right) +\\ &\qquad\qquad + \int_{I_k}\left\langle g, v_h^{k-1}\right\rangle dt,\ \forall v^{k-1}\in V_h^{k-1},\\ &\left(z_h^m, v_h^m\right) = \left(z_{\scriptscriptstyle T}, v_h^m\right),\ \forall v_h^m \in V_h^m. \end{aligned}\right. \tag{18}$$

that is also a variant of the Forward Euler method but running backwards in time.

As for the primal problem, to obtain the classical Forward Euler method, we can project $g(\mathbf{x}, t)$ using

$$g(\mathbf{x}, t) = g^0(\mathbf{x})\chi^0(t) + \sum_{k=1}^{m} g^k(\mathbf{x})\varphi^k(t),$$

where $g^k(\mathbf{x}) := g(\mathbf{x}, t_k)$. Then, as functions $\varphi^k(t)$ are constants over I_k, the source term in (18) becomes

$$\int_{I_k}\left\langle g, v_h^{k-1}\right\rangle dt = \int_{I_k}\left\langle g^k, v_h^{k-1}\right\rangle dt = \tau_k\left\langle g^k, v_h^{k-1}\right\rangle.$$

Remark 1. *We could discretize the primal (2) and dual (11) problems employing higher-order explicit Runge-Kutta methods in time using the basis functions described in [42]. However, as shown in [42], for higher order in time, the spaces in the dual problem (14) are not necessarily the same as those for the primal problem (6).*

5. Error representation

In this section, we represent the error in the quantity of interest for the lowest order case ($r = 0$).

5.1. Continuous error representation

Since (2) and (11) also satisfy (5) and (13), respectively, we consider the following continuous and discrete primal problems

$$\begin{cases} \text{Find } u \in \mathcal{U} \text{ and } u_{\tau h} \in \mathcal{U}_{\tau h} \text{ such that} \\ \qquad B_{DG}(u, v) = F(v), \quad \forall v \in \mathcal{V}, & (19) \\ \qquad B_{DG}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}), \quad \forall v_{\tau h} \in \mathcal{V}_{\tau h}, & (20) \end{cases}$$

and their duals

$$\begin{cases} \text{Find } z \in \mathcal{U} \text{ and } z_{\tau h} \in \mathcal{V}_{\tau h} \text{ such that} \\ \qquad B^*_{DG}(z, v) = L(v), \quad \forall v \in \mathcal{V}, & (21) \\ \qquad B^*_{DG}(z_{\tau h}, v_{\tau h}) = L(v_{\tau h}), \quad \forall v_{\tau h} \in \mathcal{U}_{\tau h}. & (22) \end{cases}$$

We define the errors of the primal and dual problems as

$$e := u - u_{\tau h}, \quad \varepsilon := z - z_{\tau h}.$$

Since $\mathcal{V}_{\tau h}$ is a subspace of $\mathcal{V}$, equation (19) also holds for all functions in $\mathcal{V}_{\tau h}$. Therefore, we can substitute $v_{\tau h}$ into (19). Now, subtracting from (19) equation (20), and using the bilinearity of $B_{DG}(\cdot, \cdot)$, we express the *Galerkin orthogonality* as

$$B_{DG}(e, v_{\tau h}) = 0, \quad \forall v_{\tau h} \in \mathcal{V}_{\tau h}. \tag{23}$$

In the following theorem we represent the error in the quantity of interest as an integral over the whole domain $\Omega \times I$.

Theorem 1. *(Continuous error representation). It holds that*

$$L(e) = B_{DG}(e, \varepsilon). \tag{24}$$

Proof. Equation (21) holds for the error of the primal problem $e = u - u_{\tau h}$ because $u \in \mathcal{U} \subset \mathcal{V}$ and $u_{\tau h} \in \mathcal{U}_{\tau h} \subset \mathcal{U}_{\tau} \subset \mathcal{V}$. Therefore, $e \in \mathcal{V}$ and substituting it in (21), we obtain

$$L(e) = B^*_{DG}(z, e),$$

integrating by parts the time derivative and the advection space term, we have

$$L(e) = B_{DG}(e, z).$$

Finally, the Galerkin orthogonality (23) is satisfied for $z_{\tau h} \in \mathcal{V}_{\tau h}$. Thus,

$$L(e) = B_{DG}(e, z) - B_{DG}(e, z_{\tau h}) = B_{DG}(e, \varepsilon).$$

□

5.2. Discrete error representation

In the error representation (24), we need the exact solutions u and z of the primal and dual problems, respectively. Since they are unavailable, we approximate them numerically. We obtain such approximations by, for instance, enriching the subspaces $\mathcal{U}_{\tau h}$ and $\mathcal{V}_{\tau h}$, and selecting

$$u \simeq u_{\tau \frac{h}{2}} \in \mathcal{U}_{\tau \frac{h}{2}}, \quad z \simeq z_{\tau \frac{h}{2}} \in \mathcal{V}_{\tau \frac{h}{2}},$$

where $\mathcal{U}_{\tau \frac{h}{2}}$ and $\mathcal{V}_{\tau \frac{h}{2}}$ are the subspaces obtained from splitting in half each spatial element of $\mathcal{U}_{\tau h}$, $\mathcal{V}_{\tau h}$, respectively. We do not consider a finer space in time since in our adaptive strategy we are only interested on representing the error induced by a poor space discretization.

We define the following discrete errors

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

We approximate the exact error as

$$e = u - u_{\tau h} \sim u_{\tau \frac{h}{2}} - u_{\tau h} = e_{\tau h},$$

$$\varepsilon = z - z_{\tau h} \sim z_{\tau \frac{h}{2}} - z_{\tau h} = \varepsilon_{\tau h}.$$

We define the approximated errors at each time step $k = 0, \ldots, m$ as

$$e_h^k = u_{\frac{h}{2}}^k - u_h^k, \;\; \varepsilon_h^k = z_{\frac{h}{2}}^k - z_h^k.$$

We now focus on reducing the error in the quantity of interest coming from the spatial discretization $L\left(e_{\tau h}\right)$, so we need a discrete version of Theorem 1.

We solve the primal and dual problems at two levels of discretization. Then we have

$$\left\{\begin{array}{l}\text{Find } u_{\tau\frac{h}{2}} \in \mathcal{U}_{\tau\frac{h}{2}} \text{ and } u_{\tau h} \in \mathcal{U}_{\tau h} \text{ such that}\\[2ex] \qquad B_{DG}(u_{\tau\frac{h}{2}}, v_{\tau\frac{h}{2}}) = F(v_{\tau\frac{h}{2}}), \;\; \forall v_{\tau\frac{h}{2}} \in \mathcal{V}_{\tau\frac{h}{2}}, \qquad (25)\\[2ex] \qquad B_{DG}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}), \;\; \forall v_{\tau h} \in \mathcal{V}_{\tau h}, \qquad (26)\end{array}\right.$$

$$\left\{\begin{array}{l}\text{Find } z_{\tau\frac{h}{2}} \in \mathcal{V}_{\tau\frac{h}{2}} \text{ and } z_{\tau h} \in \mathcal{V}_{\tau h} \text{ such that}\\[2ex] \qquad B^*_{DG}(z_{\tau\frac{h}{2}}, v_{\tau\frac{h}{2}}) = L(v_{\tau\frac{h}{2}}), \;\; \forall v_{\tau\frac{h}{2}} \in \mathcal{U}_{\tau\frac{h}{2}}, \qquad (27)\\[2ex] \qquad B^*_{DG}(z_{\tau h}, v_{\tau h}) = L(v_{\tau h}), \;\; \forall v_{\tau h} \in \mathcal{U}_{\tau h}. \qquad (28)\end{array}\right.$$

As the discrete solutions are piecewise constant in time, and taking into account the discretization we select in (7) and (15), we simplify the bilinear forms $B_{DG}(\cdot,\cdot)$ and $B^*_{DG}(\cdot,\cdot)$ at the discrete level as

$$\begin{aligned} B_{DG}(u_{\tau h}, v_{\tau h}) &= \sum_{k=1}^{m} \left(u_h^k - u_h^{k-1}, v_h^k\right) + \tau_k \left(\nu \nabla u_h^{k-1}, \nabla v_h^k\right) + \\ &+ \tau_k \left(\beta \cdot \nabla u_h^{k-1}, v_h^k\right) + \left(u_h^0, v_h^0\right), \\ B^*_{DG}(z_{\tau h}, v_{\tau h}) &= \sum_{k=1}^{m} \left(z_h^{k-1} - z_h^k, v_h^{k-1}\right) + \tau_k \left(\nu \nabla z_h^k, \nabla v_h^{k-1}\right) - \\ &- \tau_k \left(\beta \cdot \nabla z_h^k, v_h^{k-1}\right) + \left(z_h^m, v_h^m\right). \end{aligned} \tag{29}$$

Since $\mathcal{V}_{\tau h}$ is a subspace of $\mathcal{V}_{\tau\frac{h}{2}}$ and following the same argument as for the continuous level, we express the *discrete Galerkin orthogonality* as

$$B_{DG}(e_h, v_{\tau h}) = 0, \;\; \forall v_{\tau h} \in \mathcal{V}_{\tau h}. \tag{30}$$

Theorem 2. *(Discrete error representation). It holds that*

$$L\left(e_{\tau h}\right) = B_{DG}\left(e_{\tau h}, \varepsilon_{\tau h}\right). \tag{31}$$

Proof. Since $e_{\tau h} = u_{\tau\frac{h}{2}} - u_{\tau h} \in \mathcal{U}_{\tau\frac{h}{2}}$, we have that

$$L\left(e_{\tau h}\right) = B^*_{DG}\left(z_{\tau\frac{h}{2}}, e_{\tau h}\right),$$

and equivalently

$$L\left(e_{\tau h}\right)=\sum_{k=1}^{m}\left(z_{\frac{h}{2}}^{k-1}-z_{\frac{h}{2}}^{k}, e_{h}^{k-1}\right)+\tau_k\left(\nu \nabla z_{\frac{h}{2}}^{k}, \nabla e_{h}^{k-1}\right)-\tau_k\left(\beta \cdot \nabla z_{\frac{h}{2}}^{k}, e_{h}^{k-1}\right)+\left(z_{\frac{h}{2}}^{m}, e_{h}^{m}\right). \tag{32}$$

Now, we reorganize the first and the last terms of (32)

$$\begin{aligned} \sum_{k=1}^{m}\left(z_{\frac{h}{2}}^{k-1}-z_{\frac{h}{2}}^{k}, e_{h}^{k-1}\right)+\left(z_{\frac{h}{2}}^{m}, e_{h}^{m}\right) &= \\ =\sum_{k=0}^{m-1}\left(z_{\frac{h}{2}}^{k}, e_{h}^{k}\right)-\sum_{k=1}^{m}\left(z_{\frac{h}{2}}^{k}, e_{h}^{k-1}\right)+\left(z_{\frac{h}{2}}^{m}, e_{h}^{m}\right) &= \\ =\sum_{k=1}^{m}\left(z_{\frac{h}{2}}^{k}, e_{h}^{k}\right)-\sum_{k=1}^{m}\left(z_{\frac{h}{2}}^{k}, e_{h}^{k-1}\right)+\left(z_{\frac{h}{2}}^{0}, e_{h}^{0}\right) &= \\ =\sum_{k=1}^{m}\left(z_{\frac{h}{2}}^{k}, e_{h}^{k}-e_{h}^{k-1}\right)+\left(z_{\frac{h}{2}}^{0}, e_{h}^{0}\right)&, \end{aligned}$$

and integrating by parts in space the advection term in (32)

$$L\left(e_{\tau h}\right)=\sum_{k=1}^{m}\left(z_{\frac{h}{2}}^{k}, e_{h}^{k}-e_{h}^{k-1}\right)+\tau_k\left(\nu \nabla z_{\frac{h}{2}}^{k}, \nabla e_{h}^{k-1}\right)+\tau_k\left(z_{\frac{h}{2}}^{k}, \beta \cdot \nabla e_{h}^{k-1}\right)+(z_{\frac{h}{2}}^{0}, e_{h}^{0}).$$

Then, we have

$$L\left(e_{\tau h}\right)=B_{DG}\left(e_{\tau h}, z_{\tau \frac{h}{2}}\right),$$

and finally, applying the discrete Galerkin orthogonality (30) with $z_{\tau h} \in \mathcal{V}_{\tau h}$ we obtain

$$L\left(e_{\tau h}\right)=B_{DG}\left(e_{\tau h}, z_{\tau \frac{h}{2}}\right)-B_{DG}\left(e_{\tau h}, z_{\tau h}\right)=B_{DG}(e_{\tau h}, \varepsilon_{\tau h}).$$

□

Remark 2. *In problems (10) and (18), if we employ the Spectral Element Method in space we obtain diagonal mass matrices for each time step and it is computationally cheaper to solve than employing the Finite Element Method. However, the Spectral Element Method does not integrate exactly the L^2-terms of the operator $B_{DG}(\cdot,\cdot)$, and therefore, we loose the equality of Theorem 2. Nonetheless, we could still use it to approximate the error representation.*

6. Goal-oriented adaptivity in space for explicit in time methods

In this section, we describe an explicit goal-oriented adaptive algorithm that performs local refinements in space and global refinements in time based on the CFL condition.

6.1. Goal-oriented adaptivity in space

The goal-oriented adaptive strategy in space we propose is based on the error representation (31)

$$\begin{aligned} L\left(e_{\tau h}\right) = \left(e_h^0, \varepsilon_h^0\right) + \sum_{k=1}^{m} \Big(e_h^k - e_h^{k-1}, \varepsilon_h^k \Big) + \\ + \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k \right) + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k \right). \end{aligned} \tag{33}$$

Equivalently, we can express (33) as follows

$$\begin{aligned} L\left(e_{\tau h}\right) = \sum_{i=1}^{n_0} \left(e_h^0, \varepsilon_h^0\right)_{\Omega_i^0} + \sum_{k=1}^{m} \sum_{i=1}^{n_k} \Big(e_h^k - e_h^{k-1}, \varepsilon_h^k \Big)_{\Omega_i^k} + \\ + \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k \right)_{\Omega_i^k} + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k \right)_{\Omega_i^k}, \end{aligned} \tag{34}$$

where $\{\Omega_i^k\}_{i=1,\ldots,n_k}, \forall k = 0, \ldots, m$ is a partition of the spatial domain Ω at $t = t_k$ and $(\cdot, \cdot)_{\Omega_i^k}$ is the restriction of the inner product in $L^2(\Omega)$ to each element Ω_i^k.

Finally, applying the triangle inequality in (34), we obtain the following upper bound of the error in the quantity of interest

$$\begin{aligned} \left|L\left(e_{\tau h}\right)\right| \leq \sum_{i=1}^{n_0} \left| \left(e_h^0, \varepsilon_h^0\right)_{\Omega_i^0} \right| + \sum_{k=1}^{m} \sum_{i=1}^{n_k} \Bigg| \Big(e_h^k - e_h^{k-1}, \varepsilon_h^k \Big)_{\Omega_i^k} + \\ + \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k \right)_{\Omega_i^k} + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k \right)_{\Omega_i^k} \Bigg|, \end{aligned} \tag{35}$$

which we use to guide the goal-oriented adaptive process.

We define the error estimator of each time step as

$$Est_k := \sum_{i=0}^{n_k} \eta_i^k, \tag{36}$$

$\forall k = 0, \dots, m$, where

$$\eta_i^0 := \left(e_h^0, \varepsilon_h^0\right)_{\Omega_i^0},$$
$$\eta_i^k := \left(e_h^k - e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k} + \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k\right)_{\Omega_i^k} + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k}, \tag{37}$$

are the error estimators of each spatial element Ω_i^k at each time step.

We employ error estimators (36) and (37) to decide which space-time elements we need to refine in space in order to reduce the error in the quantity of interest.

6.2. Time adaptivity based on the CFL condition

We adapt the time mesh to ensure that the CFL condition is satisfied. As explained in Section 3, scheme (10) is explicit so it is conditionally stable. In the same way, scheme (18) is also explicit and, therefore, the CFL condition must be satisfied in both problems in order to ensure the stability of the discretization.

Even if we solve both problems in the same space-time grid, the information is propagated in opposite directions as the dual problem runs backward in time. Figure 4 illustrates the influence area of a space-time point and the direction of the information of each problem. Schemes (10) and (18) must capture those areas of influence so we have to deal with two CFL conditions, one for the primal problem and another for the dual problem.

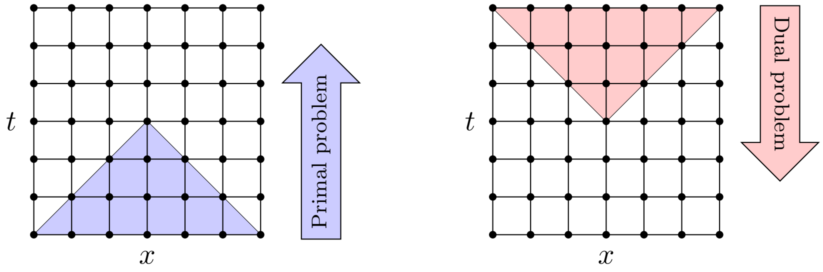

Figure 4: Transmission of the information for the primal and dual problems.

If we denote by h_k the minimum element size of the spatial mesh at t_k, the two CFL conditions are

$$\frac{\tau_k}{h_k^2}\nu < \frac{1}{6} \text{ (dual)}, \quad \frac{\tau_{k+1}}{h_k^2}\nu < \frac{1}{6} \text{ (primal)}, \tag{38}$$

where $\tau_k = t_k - t_{k-1}$ and $\tau_{k+1} = t_{k+1} - t_k$.

Therefore, we adapt the time grid based on the CFL conditions defined in (38). Once the spatial meshes are refined, we identify those time intervals where the CFL conditions are not satisfied and we split them by introducing new synchronous levels of spatial discretization (spatial meshes). Figure 5 shows the adaptive process in time. We have h_k^2 in (38), thus we need to split the time step by four in order to satisfy the conditions.

There are different options to select the spatial meshes we insert in a time interval. In Figure 5 we employ the same spatial mesh as the one associated to t_{k-1}, but it could be the one from t_k or, for example, the union of the spaces associated to both meshes. We obtain similar results in Section 7 by employing different approaches.

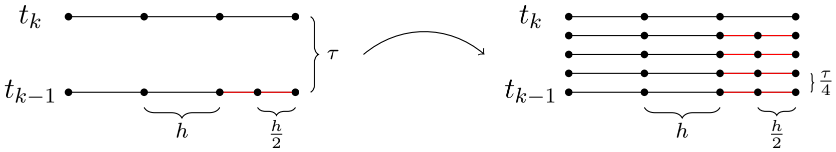

Figure 5: Time refinements based on the CFL condition.

6.3. Algorithm

Algorithm 1 describes our goal-oriented adaptive process in space together with a CFL-based adaptive process in time.

The input arguments of Algorithm 1 are the time grid $\{\tau_k\}_{k=1,\dots,m}$, the spatial mesh at each time step $\{\mathcal{M}_h^k\}_{k=0,\dots,m}$, the tolerance *tol* for the error in the quantity of interest and two parameters $\theta, \lambda \in [0,1]$.

We first calculate the primal solutions $u_{\tau h}$ and $u_{\tau \frac{h}{2}}$ forward in time and we compute the dual solutions $z_{\tau h}$ and $z_{\tau \frac{h}{2}}$ backwards in time. Then, we estimate Est_k and η_i^k and for all spatial meshes satisfying $|Est_k| \geq \theta \cdot \max\limits_{0\leq k\leq m} |Est_k|$, we refine those elements in space that satisfy $|\eta_i^k| \geq \lambda \cdot \max\limits_{1\leq i\leq n_k} |\eta_i^k|$. Finally, we refine those time intervals where the CFL conditions (38) are not satisfied. The process ends when the relative error

$$QoI := \frac{|L\left(e_{\tau h}\right)|}{\left|L\left(u_{\tau \frac{h}{2}}\right)\right|} \cdot 100,$$

is lower than a user-prescribed fixed tolerance *tol*.

Algorithm 1 Goal-oriented in space and CFL-based in time adaptivity

```
1: **Input:** $\{\tau_k\}_{k=1,\dots,m}$, $\{\mathcal{M}_h^k\}_{k=0,\dots,m}$, $tol$, $\theta$, $\lambda$
2: Compute $u_{\tau h}$, $u_{\tau \frac{h}{2}}$ and $e_{\tau h}$
3: Compute $z_{\tau h}$, $z_{\tau \frac{h}{2}}$ and $\varepsilon_{\tau h}$
4: Estimate $QoI$
5: **while** $QoI > tol$ **do**
6:     **for** $k = 0$ to $m$ and $i = 1$ to $n$ **do**
7:         Compute $\eta_i^k$ and $Est_k$
8:     **end for**
9:     **for** $k = 0$ to $m$ and $i = 1$ to $n$ **do**
10:        **if** $|Est_k| \geq \theta \cdot \max\limits_{0\leq k\leq m} |Est_k|$ and $|\eta_i^k| \geq \lambda \cdot \max\limits_{1\leq i\leq n_k} |\eta_i^k|$ **then**
11:            Refine element $\Omega_i^k$ of $\mathcal{M}_h^k$
12:        **end if**
13:    **end for**
14:    **for** $k = 1$ to $m$ **do**
15:        **if** CFL-conditions are not satisfied in $I_k$ **then**
16:            Refine interval $I_k$
17:        **end if**
18:    **end for**
19:    Compute $u_{\tau h}$, $u_{\tau \frac{h}{2}}$ and $e_{\tau h}$
20:    Compute $z_{\tau h}$, $z_{\tau \frac{h}{2}}$ and $\varepsilon_{\tau h}$
21:    Estimate $QoI$
22: **end while**
```

Remark 3. *In Algorithm 1 we have two meshes at each time step: the fine mesh and the coarse mesh, $\mathcal{M}_{\frac{h}{2}}^k$ and $\mathcal{M}_h^k$, respectively. The errors e_h^k and ε_h^k are computed in the fine mesh $\mathcal{M}_{\frac{h}{2}}^k$, whereas the estimators η_i^k are calculated over each element of the coarse mesh $\mathcal{M}_h^k$.*

Remark 4. *In Algorithm 1 (line 10) we employ the maximum strategy [45]. Graphically, we can interpret this method as in Figure 6. First, we order, for example, all the estimators in time from highest to lowest. Then, we draw a horizontal line at $\theta \cdot \max\limits_{0\leq k\leq m} |Est_k|$. Finally, we refine those spatial meshes whose estimators are above the line.*

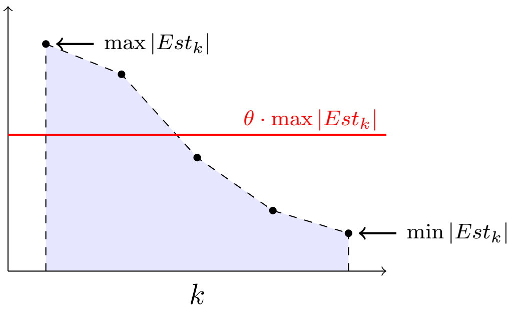

Figure 6: Maximum strategy.

7. Numerical results

7.1. Diffusion problem

We consider $d = 1$, $\Omega = (0, 1)$, $T = 2$, $\beta = 0$, $f(x, t) = (1 + \pi^2 t)\sin(\pi x)$ and $u(x, 0) = 0$. We select a discontinuous diffusion coefficient

$$\nu(x) = \begin{cases} 0.01, & x \in [0.25, 0.75], \\ 0.001, & \text{elsewhere}, \end{cases}$$

and the following output functional

$$L(u) = \int_I \int_{\Omega_0} u(x, t) dx dt,$$

where $\Omega_0 = (0, 0.25) \cup (0.75, 1)$.

Figure 7 shows the primal (2) and dual (11) reference solutions. In Figure 8 we can see the relative error in the quantity of interest and the upper bound (35) when we perform four uniform space-time refinements. We start with 2^3 elements in space and 40 time steps for the first iteration. Then, we split four times each time interval per uniform refinement in space in order to satisfy the CFL condition. We calculate the total number of degrees of freedom as the sum of the degrees of freedom of each spatial mesh, i.e.,

$$DoF_{_T} = \sum_{k=0}^{m} DoF(\Omega^k).$$

In Algorithm 1 we set 40 time steps, 2^3 elements in space, $\theta = \lambda = 0.3$ and $tol = 0.01\%$. In the first iteration, we select the number of time steps in such a way that we obtain limit values of the CFL conditions. Then, the algorithm is going to refine those time intervals where the CFL violated.

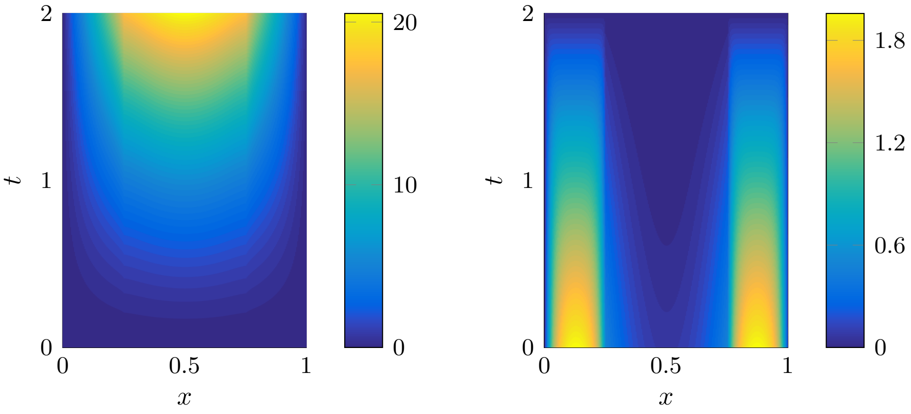

Figure 7: Colormap of the primal (2) (left) and dual (11) (right) problems.

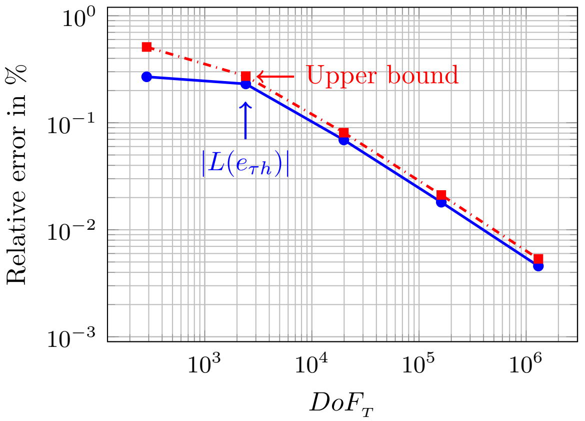

Figure 8: Error in the quantity of interest and upper bound (35) when we perform uniform space-time refinements.

Figure 9 exhibits the adapted space-time mesh after 28 iterations and Figure 10 shows the relative error in the quantity of interest and the upper bound (35). As a comparison, for uniform refinements we need more than 10^5 degrees of freedom to achieve a relative error of 0.01%, while for the adaptive method we need around $5 \cdot 10^3$.

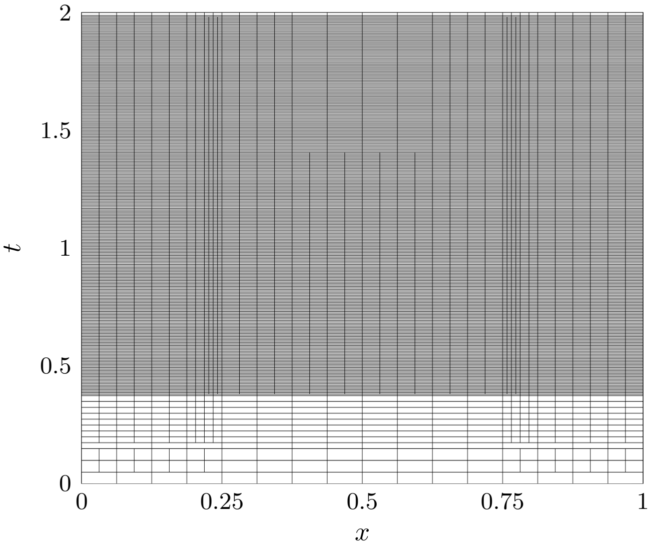

Figure 9: Adapted space-time mesh.

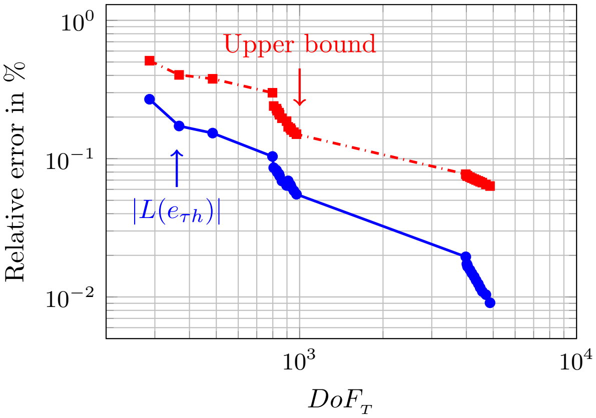

Figure 10: Error in the quantity of interest and upper bound (35) when we perform local refinements in space.

7.2. Advection-diffusion problem

Let $d = 1$, $\Omega = (0,1)$ and $T = 0.5$. We consider problem (1) with $\nu = 0.015$, $\beta = 1.5$, $f(x,t) = 0$,

$$u_0(x) = \begin{cases} 1, & x \in [0.125, 0.375], \\ 0, & \text{elsewhere}, \end{cases}$$

and we select the following output functional

$$L(u) = \int_{I_0} \int_{\Omega_0} u(x,t) dx dt,$$

where $\Omega_0 \times I_0 = (0.75, 1) \times (0.4, 0.5]$ is a subdomain of $\Omega \times I$.

Figure 11 shows the reference solutions of the primal (2) and dual (11) problems. We observe that in the primal problem, the initial condition is propagated. Due to the boundary conditions, a boundary layer is formed in the final time steps in the right endpoint of the spatial domain. We are interested in reducing the error of the solution in such boundary layer. Figure 12 shows the relative error in the quantity of interest and the upper bound (35) when we perform four uniform space-time refinements starting with 15 time steps and 2^3 elements in space.

We again set $\theta = \lambda = 0.3$ and $tol = 0.5\%$ in Algorithm 1. Figure 13 shows the adapted space-time mesh after 37 iterations. Figure 14 exhibits the relative error in the quantity of interest and the upper bound when we perform local refinements in space. We conclude that the convergence ratio in this case is similar to uniform space-time refinements as we are not heavily employing local space-time refinements.

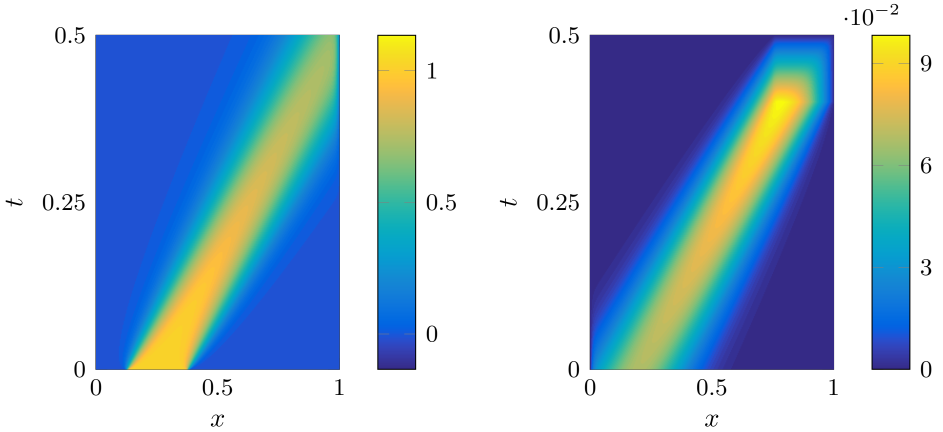

Figure 11: Colormap of the primal (left) and dual (right) problems.

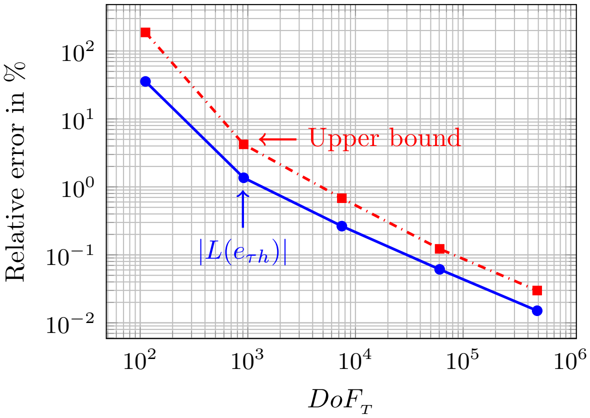

Figure 12: Error in the quantity of interest and upper bound when we perform uniform space-time refinements.

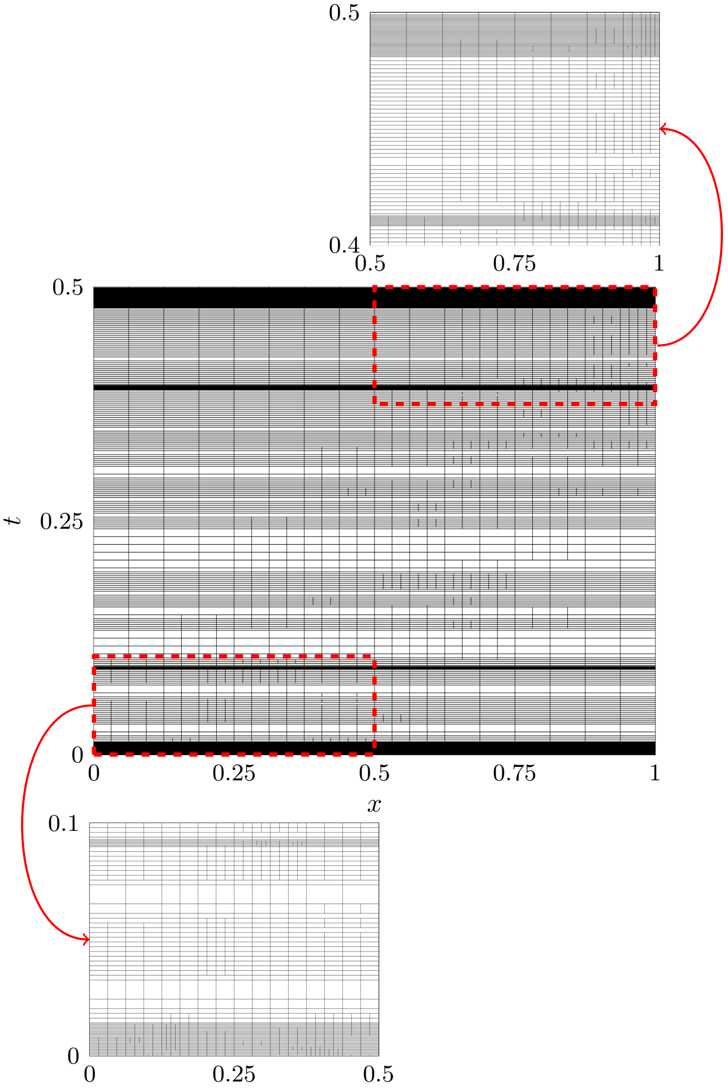

Figure 13: Adapted space-time mesh.

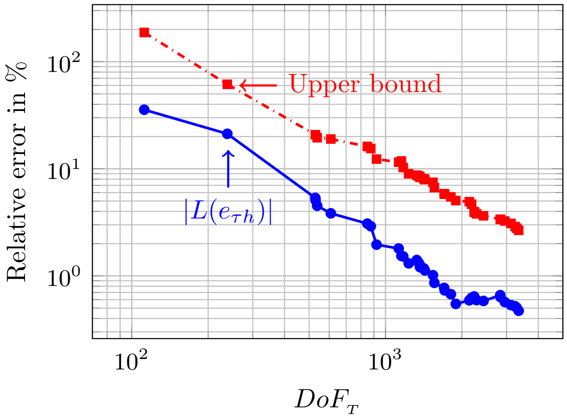

Figure 14: Error in the quantity of interest and upper bound when we perform local refinements in space.

7.3. Symmetric estimator

We consider $d = 1$, $\Omega = (0, 1)$, $T = 0.5$, $\nu = 0.015$, $\beta = 0$, $f(x, t) = 0$ and

$$u_0(x) = \begin{cases} 1, & x \in [0.125, 0.375], \\ 0, & \text{elsewhere}, \end{cases}$$

and we select the following output functional

$$L(u) = \int_{\Omega_0} u(x, T)dx,$$

where $\Omega_0 = (0.625, 0.875) \subset \Omega$. Figure 15 shows the primal and dual reference solutions where both solutions are symmetric and they run in opposite directions in time. Figure 16 exhibits the relative error in the quantity of interest and the upper bound after 4 uniform space-time refinements starting from 2^3 elements in space and 15 time steps.

For Algorithm 1 we again set $\theta = \lambda = 0.3$ and $tol = 3\%$. Employing the operator $B_{DG}(\cdot, \cdot)$ as the adaptive criteria, the algorithm tends to refine the mesh to reduce de error of the primal problem. In the same way, if we employ $B^*_{DG}(\cdot, \cdot)$, the algorithm refines to reduce the error of the dual problem. Therefore, to obtain a symmetric adapted mesh we employ the following symmetric error representation

$$L(e_{\tau h}) = \frac{1}{2} B_{DG}(e_{\tau h}, \varepsilon_{\tau h}) + \frac{1}{2} B^*_{DG}(\varepsilon_{\tau h}, e_{\tau h}). \tag{39}$$

Figures 17 and 18 show the adapted space-time meshes and the relative errors in the quantity of interest, respectively, when we employ $B_{DG}(\cdot,\cdot)$ and $B^*_{DG}(\cdot,\cdot)$ as adaptive criteria. Both graphics in Figure 18 are the same because the problem is symmetric. Figure 19 shows the adapted space-time mesh after 4 iterations and the relative error in the quantity of interest when we employ the symmetric operator (39). We conclude that with estimator $B_{DG}(\cdot,\cdot)$ or $B^*_{DG}(\cdot,\cdot)$, we need 10 iterations to achieve an error of 3% while with symmetric estimator (39) we only need four iterations. Moreover, the space-time mesh obtained employing (39) is symmetric.

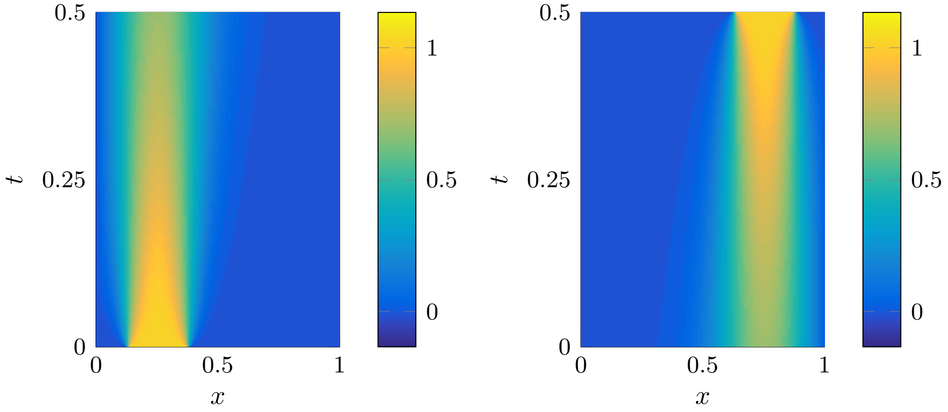

Figure 15: Colormap of the primal (2) (left) and dual (11) (right) problems.

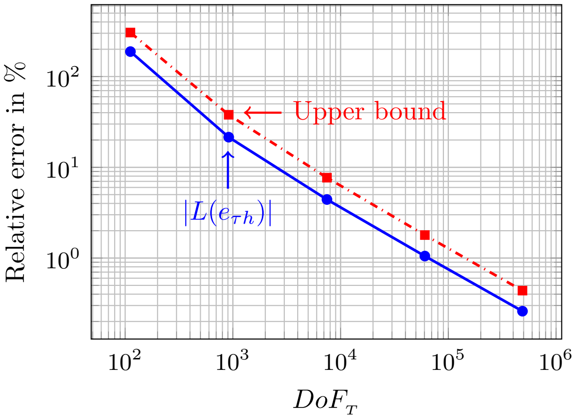

Figure 16: Error in the quantity of interest and upper bound (35) when we perform uniform space-time refinements.

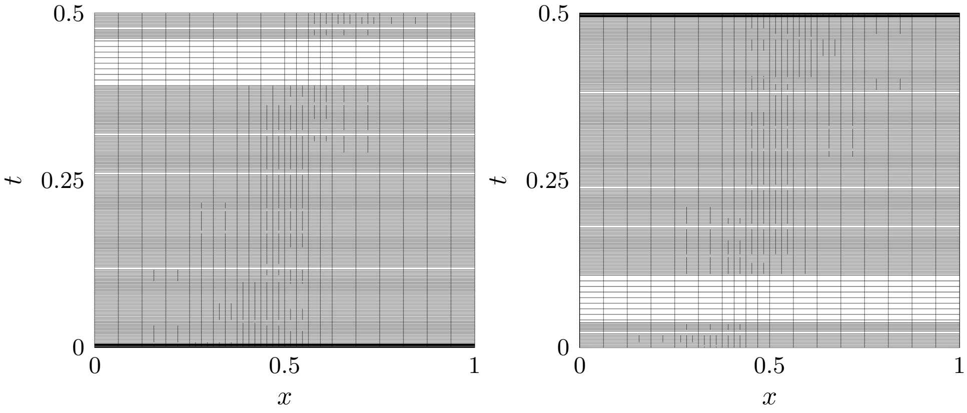

Figure 17: Adapted mesh employing $B_{DG}(\cdot,\cdot)$ (left) and adapted mesh employing $B^*_{DG}(\cdot,\cdot)$ (right).

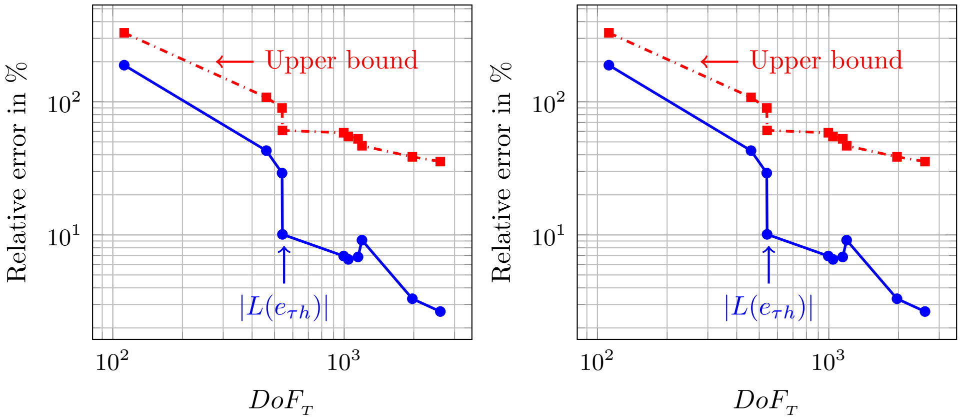

Figure 18: Error in the quantity of interest and upper bounds employing $B_{DG}(\cdot,\cdot)$ (left) and $B^*_{DG}(\cdot,\cdot)$ (right).

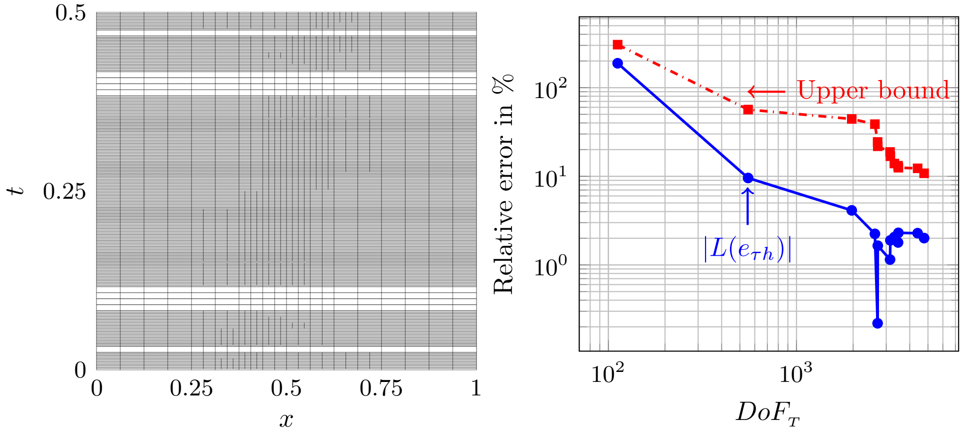

Figure 19: Adapted space-time mesh (left) and error in the quantity of interest employing symmetric estimator (39).

Another strategy could be to select, for each space-time element, the following estimator

$$\mu_i^k := \max\{\eta_i^k, \eta_i^{*,k}\}, \quad \forall k = 0, \dots, m, \tag{40}$$

where η_i^k are the estimators defined in (37) and $\eta_i^{*,k}$ are defined from the dual operator $\forall k = 1, \dots, m$

$$\eta_i^{*,k} : = \left(\varepsilon_h^{k-1} - \varepsilon_h^k, e_h^{k-1}\right)_{\Omega_i^k} + \tau_k \left(\nu \nabla \varepsilon_h^k, \nabla e_h^{k-1}\right)_{\Omega_i^k} - \tau_k \left(\beta \cdot \nabla \varepsilon_h^k, e_h^{k-1}\right)_{\Omega_i^k},$$

$$\eta_i^{*,m} : = \left(\varepsilon_h^m, e_h^m\right)_{\Omega_i^m}.$$

Figure 20 shows the adapted space-time mesh after 5 iterations and the relative error in the quantity of interest when we employ operator (40). The estimator (40) also yields a symmetric mesh. In this example, with estimator (39) we need more than 10^3 degrees of freedom the achieve an error of 3% while with estimator (40) we need less than 10^3 degrees of freedom the reach an error of 1%, although in this last case, we observe an errant behavior of the error in the quantity of interest.

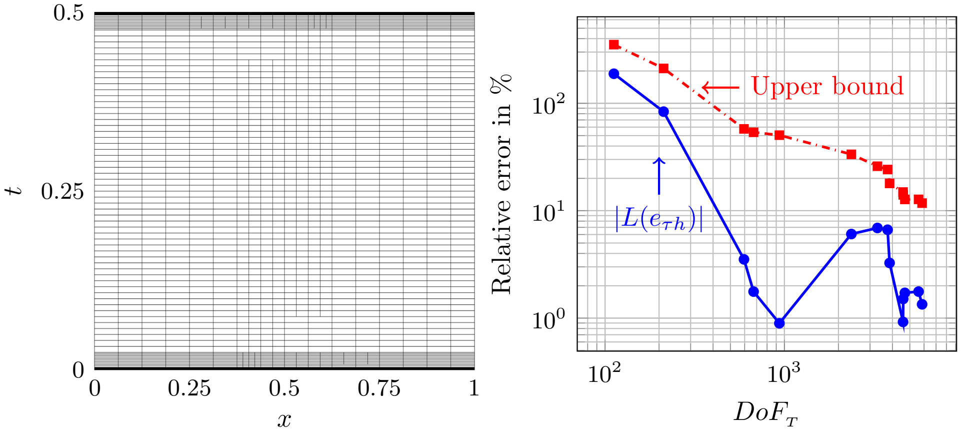

Figure 20: Adapted space-time mesh (left) and error in the quantity of interest employing symmetric estimator (40).

8. Conclusions

We propose an explicit-in-time goal-oriented adaptive algorithm for the linear advection-diffusion equation. We derive the Forward Euler method from a space-time variational formulation of the primal employing constant-in-time trial and test function. For that, we need a discontinuous Petrov-Galerkin formulation. Interchanging the trial and test spaces, we also derive the Forward Euler method for the dual problem but running backwards in time. Then, the error in the quantity of interest is expressed employing the errors in space of the primal and dual problems. The adaptive algorithm we propose performs local goal-oriented adaptive refinements in space. The time grid is adapted locally (for the entire space) based on the CFL condition in order to ensure the stability of the method. We show the performance of the algorithm in one-dimensional diffusion and advection-diffusion problems.

A possible extension of this work is to employ higher-order explicit time-marching schemes to perform space-time goal-oriented adaptivity, as those shown in [42]. Another possible application is to employ the space-time FEM that are equivalent to explicit methods in time to perform local space-time (goal-oriented) refinements.

Acknowledgments

The first four authors have received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-

Curie grant agreement No 777778.

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/AEI.

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.

Kristoffer G. van der Zee has received funding from the School of Mathematical Sciences at University of Nottingham.

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. Jay Gopalakrishnan from Portland State University for the very fruitful discussions on the topic.

Appendix A. Stability

The stability of problems (10) and (18) depends on whether the eigenvalues of the matrix $A := -M^{-1}(K+R)$ (where M, K and R are the mass, stiffness and weak derivative matrices, respectively) are included in the stability region of the time integration method. Figure A.21 shows the stability regions of the explicit Runge-Kutta methods of s stages and order p, when $s = p$. The Forward Euler method is a Runge-Kutta method with $s = p = 1$ [46].

Now, we derive the CFL condition for one space dimension. We assume that $\nu > 0$ and β are constants. We consider uniform spatial and temporal meshes, where h is the size of each element in space and τ is the time step size. If we employ piecewise linear basis functions in space, the elemental matrices are

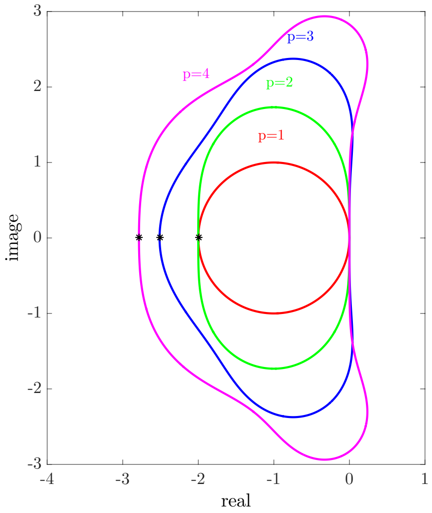

Figure A.21: Stability of the Explicit Runge-Kutta methods when $s = p$ (interior to curves).

$$M_{elem} = h\begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix},\ K_{elem} = \frac{\nu}{h}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},\ R_{elem} = \beta\begin{pmatrix} -1/2 & -1/2 \\ 1/2 & 1/2 \end{pmatrix}.$$

Matrix A has real negative eigenvalues, so it is enough to require that the greatest eigenvalue in module satisfies

$$-2 < \lambda_{max}\tau < 0 \tag{A.1}$$

to ensure that all eigenvalues of A are included in the stability region of both methods. The greatest eigenvalue in module of matrix A is $\lambda_{max} = \dfrac{-12\nu}{h^2}$. Hence, condition (A.1) becomes

$$\frac{\tau\nu}{h^2} < \frac{1}{6},$$

which is the CFL condition of both methods (10) and (18) to ensure the stability in one space dimension.

On the other hand, in the advection-diffusion equation, numerical instabilities occur when the Péclet number is large [47, 48]

$$Pe := \frac{||\beta||h}{2\nu}.$$

Appendix B. Definition of $B^*_{DG}(\cdot,\cdot)$

In this appendix, we explain how we derive $B^*_{DG}(\cdot,\cdot)$ from $B_{DG}(\cdot,\cdot)$. In Section 3 we defined $B_{DG}(\cdot,\cdot)$ as

$$\begin{aligned} B_{DG}(u,v) := \sum_{k=1}^{m}\int_{I_k}\Big(\langle u_{,t},v\rangle + (\nu\nabla u,\nabla v) + (\beta\cdot\nabla u, v)\Big)dt+ \\ + \sum_{k=1}^{m}\Big([\![u]\!]^k, v(t_k^-)\Big) + (u(0^+),v(0^-)), \end{aligned} \tag{B.1}$$

if we integrate by parts in time the first integral of (B.1) we obtain

$$\sum_{k=1}^{m}\int_{I_k} - \langle u, v_{,t}\rangle\, dt + \sum_{k=1}^{m}\Big((u(t_k^-),v(t_k^-)) - (u(t_{k-1}^+),v(t_{k-1}^+))\Big). \tag{B.2}$$

Now, we add to expression (B.2) the jump terms and the initial condition of (B.1) and we have

$$\begin{aligned} \sum_{k=1}^{m}\int_{I_k} - \langle u, v_{,t}\rangle\, dt + \sum_{k=1}^{m}(u(t_k^-),v(t_k^-)) - \sum_{k=1}^{m}(u(t_{k-1}^+),v(t_{k-1}^+)) + \\ + \sum_{k=1}^{m}(u(t_k^+),v(t_k^-)) - \sum_{k=1}^{m}(u(t_k^-),v(t_k^-)) + (u(0^+),v(0^-)). \end{aligned}$$

Simplifying the last expression

$$\sum_{k=1}^{m}\int_{I_k} - \langle u, v_{,t}\rangle\, dt - \sum_{k=1}^{m-1}(u(t_{k-1}^+),v(t_{k-1}^+)) + \sum_{k=0}^{m}(u(t_k^+),v(t_k^-)),$$

Equivalently

$$\sum_{k=1}^{m}\int_{I_k} - \langle u, v_{,t}\rangle\, dt - \sum_{k=1}^{m}(u(t_{k-1}^+),v(t_{k-1}^+)) + \sum_{k=0}^{m-1}(u(t_k^+),v(t_k^-)) + (u(T^+),v(T^-)),$$

and fixing the indices, we obtain

$$\sum_{k=1}^{m}\int_{I_k} - \langle u, v_{,t}\rangle\, dt - \sum_{k=1}^{m}(u(t_{k-1}^+),v(t_{k-1}^+)) + \sum_{k=1}^{m}(u(t_{k-1}^+),v(t_{k-1}^-)) + (u(T^+),v(T^-)).$$

Therefore, we have the following expression

$$\sum_{k=1}^{m}\int_{I_k}\Big(-\langle u,v_{,t}\rangle+(\nu\nabla u,\nabla v)+(\beta\cdot\nabla u,v)\Big)dt-\sum_{k=1}^{m}\Big(u(t_{k-1}^{+}),[\![v]\!]^{k-1})\Big)+(u(T^{+}),v(T^{-})).$$

Finally, integrating by parts in space the advection term and as the advection field is divergence free ($\nabla\cdot\beta=0$), we obtain

$$\sum_{k=1}^{m}\int_{I_k}\Big(-\langle u,v_{,t}\rangle+(\nabla u,\nu\nabla v)-(u,\beta\cdot\nabla v)\Big)dt-\sum_{k=1}^{m}\Big(u(t_{k-1}^{+}),[\![v]\!]^{k-1})\Big)+(u(T^{+}),v(T^{-})).$$

Then, we define the discontinuous Galerkin bilinear form for the dual problem as

$$\begin{aligned}B^{*}_{DG}(z,v):=&\sum_{k=1}^{m}\int_{I_k}\Big(-\langle z_{,t},v\rangle+(\nu\nabla z,\nabla v)-(\beta\cdot\nabla z,v)\Big)dt\\&-\sum_{k=1}^{m}\Big([\![z]\!]^{k-1},v(t_{k-1}^{+})\Big)+(z(T^{-}),v(T^{+})).\end{aligned}$$

References

[1] J. Erickson, D. Guoy, J. M. Sullivan, A. Üngör, Building spacetime meshes over arbitrary spatial domains, Engineering with Computers 20 (4) (2005) 342–353.

[2] R. Abedi, B. Petracovici, 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) (2006) 3247–3273.

[3] S. T. Miller, R. B. Haber, A spacetime discontinuous Galerkin method for hyperbolic heat conduction, Computer Methods in Applied Mechanics and Engineering 198 (2) (2008) 194–209.

[4] J. Gopalakrishnan, P. Monk, P. Sepúlveda, A tent pitching scheme motivated by Friedrichs theory, Computers & Mathematics with Applications 70 (5) (2015) 1114–1135.

[5] D. Schötzau, 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) (2000) 837–875.

[6] D. Schötzau, C. Schwab, An hp a priori error analysis of the DG time-stepping method for initial value problems, Calcolo 37 (4) (2000) 207–232.

[7] T. Werder, K. Gerdes, D. Schötzau, C. Schwab, hp-discontinuous Galerkin time stepping for parabolic problems, Computer Methods in Applied Mechanics and Engineering 190 (49) (2001) 6685–6708.

[8] M. Besier, 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) (2012) 1139–1166.

[9] W. Bangerth, R. Rannacher, Adaptive finite element techniques for the acoustic wave equation, Journal of Computational Acoustics 9 (2) (2001) 575–591.

[10] W. Bangerth, M. Geiger, R. Rannacher, Adaptive Galerkin finite element methods for the wave equation, Computational Methods in Applied Mathematics 10 (1) (2010) 3–48.

[11] U. Köcher, M. Bause, Variational space-time methods for the wave equation, Journal of Scientific Computing 61 (2) (2014) 424–453.

[12] D. Pardo, L. Demkowicz, C. Torres-Verdín, M. Paszynski, Simulation of resistivity logging-while-drilling (LWD) measurements using a self-adaptive goal-oriented hp-finite element method, SIAM Journal on Applied Mathematics 66 (6) (2006) 2085–2106.

[13] L. Chamoin, P. Ladevèze, A non-intrusive method for the calculation of strict and efficient bounds of calculated outputs of interest in linear viscoelasticity problems, Computer Methods in Applied Mechanics and Engineering 197 (9) (2008) 994–1014.

[14] K. Van der Zee, E. Van Brummelen, I. Akkerman, R. De Borst, Goal-oriented error estimation and adaptivity for fluid–structure interaction using exact linearized adjoints, Computer Methods in Applied Mechanics and Engineering 200 (37) (2011) 2738–2757.

[15] J. Alvarez-Aramberri, D. Pardo, Dimensionally adaptive hp-finite element simulation and inversion of 2D magnetotelluric measurements, Journal of Computational Science 18 (2017) 95–105.

[16] V. Darrigrand, D. Pardo, I. Muga, Goal-oriented adaptivity using unconventional error representations for the 1D Helmholtz equation, Computers & Mathematics with Applications 69 (9) (2015) 964–979.

[17] V. Darrigrand, Á. Rodríguez-Rozas, I. Muga, D. Pardo, A. Romkes, S. Prudhomme, Goal-oriented adaptivity using unconventional error representations for the multidimensional Helmholtz equation, International Journal for Numerical Methods in Engineering 113 (1) (2018) 22–42.

[18] R. Becker, R. Rannacher, Weighted a posteriori error control in FE methods, IWR, 1996.

[19] J. T. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, Computers & Mathematics with Applications 41 (5) (2001) 735–756.

[20] S. Prudhomme, J. T. Oden, Computable error estimators and adaptive techniques for fluid flow problems, Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics 25 (2003) 207–268.

[21] D. Pardo, L. Demkowicz, C. Torres-Verdín, L. Tabarovsky, A goal-oriented hp-adaptive finite element method with electromagnetic applications. Part I: Electrostatics, International Journal for Numerical Methods in Engineering 65 (8) (2006) 1269–1309.

[22] D. Pardo, L. Demkowicz, C. Torres-Verdin, M. Paszynski, A self-adaptive goal-oriented hp-finite element method with electromagnetic applications. Part II: Electrodynamics, Computer methods in applied mechanics and engineering 196 (37) (2007) 3585–3597.

[23] W. E. Schiesser, The numerical method of lines: integration of partial differential equations, Elsevier, 2012.

[24] N. Collier, H. Radwan, L. Dalcin, V. M. Calo, Time adaptivity in the diffusive wave approximation to the shallow water equations, Journal of Computational Science 4 (3) (2013) 152–156.

[25] E. Alberdi, J. J. Anza, Solution of the wave-type PDE by numerical damping control multistep methods, Procedia Computer Science 29 (2014) 779–789.

[26] P. Vignal, N. Collier, L. Dalcin, D. Brown, V. M. Calo, An energy-stable time-integrator for phase-field models, Computer Methods in Applied Mechanics and Engineering 316 (2017) 1179–1214.

[27] E. Hairer, S. Nørsett, G. Wanner, Solving ordinary differential equations I. nonstiff problems (1987).

[28] G. Wanner, E. Hairer, Solving ordinary differential equations II, Vol. 1, Springer-Verlag, Berlin, 1991.

[29] P. Díez, G. Calderón, Goal-oriented error estimation for transient parabolic problems, Computational Mechanics 39 (5) (2007) 631–646.

[30] F. Schieweck, A-stable discontinuous Galerkin-Petrov time discretization of higher order, Journal of Numerical Mathematics 18 (1) (2010) 25–57.

[31] S. Hussain, F. Schieweck, S. Turek, Higher-order Galerkin time discretizations and fast multigrid solvers for the heat equation, Journal of Numerical Mathematics 19 (1) (2011) 41–61.

[32] N. Ahmed, 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 (2015) 83–101.

[33] W. Bangerth, R. Rannacher, Adaptive finite element methods for differential equations, Birkhäuser, 2013.

[34] J. Muñoz-Matute, E. Alberdi, D. Pardo, V. M. Calo, Time-domain goal-oriented adaptivity using pseudo-dual error representations, Computer Methods in Applied Mechanics and Engineering 325 (2017) 395–415.

[35] N. Parés, P. Díez, 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) (2008) 1641–1660.

[36] M. Schmich, B. Vexler, Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations, SIAM Journal on Scientific Computing 30 (1) (2008) 369–393.

[37] F. Verdugo, N. Parés, P. Díez, Error assessment in structural transient dynamics, Archives of Computational Methods in Engineering 21 (1) (2014) 59–90.

[38] F. Verdugo, N. Parés, P. Díez, Goal-oriented space-time adaptivity for transient dynamics using a modal description of the adjoint solution, Computational Mechanics 54 (2) (2014) 331–352.

[39] G. Şimşek, X. Wu, K. van der Zee, 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 (2015) 83–109.

[40] X. Wu, K. van der Zee, G. Simsek, H. Van Brummelen, A posteriori error estimation and adaptivity for nonlinear parabolic equations using IMEX-Galerkin discretization of primal and dual equations, arXiv preprint arXiv:1706.04281.

[41] J. H. Chaudhry, D. Estep, V. Ginting, J. N. Shadid, 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 (2015) 730–751.

[42] J. Muñoz-Matute, D. Pardo, V. M. Calo, E. Alberdi, Variational formulations for explicit Runge-Kutta methods, arXiv preprint arXiv:1806.07803.

[43] G. Strang, On the construction and comparison of difference schemes, SIAM Journal on Numerical Analysis 5 (3) (1968) 506–517.

[44] M. Ainsworth, J. T. Oden, A posteriori error estimation in finite element analysis, Vol. 37, John Wiley & Sons, 2011.

[45] R. Verfürth, A review of a posteriori error estimation and adaptive mesh-refinement techniques, John Wiley & Sons Inc, 1996.

[46] D. F. Griffiths, D. J. Higham, Numerical methods for ordinary differential equations: initial value problems, Springer Science & Business Media, 2010.

[47] L. P. Franca, G. Hauke, A. Masud, Revisiting stabilized finite element methods for the advective–diffusive equation, Computer Methods in Applied Mechanics and Engineering 195 (13-16) (2006) 1560–1572.

[48] V. John, E. Schmeyer, Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion, Computer Methods in Applied Mechanics and Engineering 198 (3-4) (2008) 475–494.