



Adaptive time-step control for a monolithic multirate scheme coupling the heat and wave equation

Martyna Soszyńska¹ · Thomas Richter¹

Received: 21 April 2020 / Accepted: 5 March 2021 / Published online: 1 April 2021
© The Author(s) 2021

Abstract

We study the dynamics of a parabolic and a hyperbolic equation coupled on a common interface. We develop time-stepping schemes that can use different time-step sizes for each of the subproblems. The problem is formulated in a strongly coupled (monolithic) space-time framework. Coupling two different step sizes monolithically gives rise to large algebraic systems of equations. There, multiple states of the subproblems must be solved at once. For efficiently solving these algebraic systems, we inherit ideas from the partitioned regime. Therefore we present two decoupling methods, namely a partitioned relaxation scheme and a shooting method. Furthermore, we develop an a posteriori error estimator serving as a mean for an adaptive time-stepping procedure. The goal is to optimally balance the time-step sizes of the two subproblems. The error estimator is based on the dual weighted residual method and relies on the space-time Galerkin formulation of the coupled problem. As an example, we take a linear set-up with the heat equation coupled to the wave equation. We formulate the problem in a monolithic manner using the space-time framework. In numerical test cases, we demonstrate the efficiency of the solution process and we also validate the accuracy of the a posteriori error estimator and its use for controlling the time-step sizes.

Keywords Multirate time-stepping · Galerkin time discretization · A posteriori error estimation · Partitioned solution

Mathematics Subject Classification 65M60 · 65M15 · 65M50 · 74F10

Communicated by Lothar Reichel.

✉ Martyna Soszyńska
martyna.soszynska@ovgu.de

Thomas Richter
thomas.richter@ovgu.de

¹ Otto-von-Guericke Universität Magdeburg, Magdeburg, Germany

1 Introduction

In this work, we are going to examine surface coupled multiphysics problems that are inspired by fluid-structure interaction (FSI) problems [24]. We couple the heat equation with the wave equation through an interface, where the typical FSI coupling conditions or Dirichlet-Neumann type act. Despite its simplicity, each of the subproblems exhibits different temporal dynamics which is also found in FSI. The solution of the heat equation, as a parabolic problem, manifests smoothing properties. Thus, it can be characterized as a problem with slow temporal dynamics. The wave equation, on the other hand, is an example of a hyperbolic equation with highly oscillatory properties.

FSI problems are characterized by two specific difficulties: the coupling of an equation of parabolic type with one of hyperbolic type gives rise to regularity problems at the interface. Further, the added mass effect [6] is present for problems coupling materials of a similar density which calls for strongly coupled discretization and solution schemes. This is the monolithic approach for modeling FSI, in contrast to partitioned approaches, where each of the subproblems is treated and solved as a separate system. The monolithic approach allows for a more rigorous mathematical setting and the use of large time-steps. The partitioned approach allows using fully optimized separate techniques for both of the subproblems. Most realizations for FSI, such as the technique described here, have to be regarded as a blend of both philosophies: while the formulation and discretization are monolithic, ideas of partitioned approaches are borrowed for solving the algebraic problems.

Since FSI problems feature distinct time scales in the two subproblems the use of multirate time-stepping schemes with adapted step sizes for fluid and solid is obvious. For parabolic problems, the concept of multirate time-stepping was discussed in [4, 9, 17]. In the hyperbolic setting, it was considered in [3, 7, 8, 23]. In the context of fluid-structure interactions, such subcycling methods are used in aeroelasticity [22]. There, explicit time integration schemes are used for the flow problem and implicit schemes for the solid problem [10]. In the low Reynolds number regime, common in hemodynamics, the situation is different. Here, implicit and strongly coupled schemes are required by the added mass effect. Hence, large time steps can be applied for the flow problem, but smaller time-steps might be required within the solid. A study on benchmark problems in fluid dynamics (Schäfer, Turek '96 [26]) and FSI presented in [18] shows that FSI problems demand a much smaller step size. However, the problem configuration and the resulting nonstationary dynamics are very similar to oscillating solutions with nearly the same period [25].

We will derive a monolithic variational formulation for FSI like problems that can handle different time-step sizes in the two subproblems. Implicit coupling of two problems with different step sizes will give rise to very large systems where multiple states must be solved at once. In Sect. 3 we will study different approaches for an efficient solution of these coupled systems, a simple partitioned relaxation scheme and a shooting like approach.

Next, in Sect. 4 we present a posteriori error estimators based on the dual weighted residual method [2] for automatically identifying optimal step sizes for the two subproblems. Numerical studies on the efficiency of the time adaptation procedure are presented in Sect. 5.

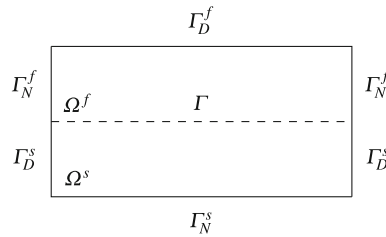


Fig. 1 View of the domain $\Omega = (0, 4) \times (-1, 1)$ split into fluid Ω^f and solid Ω^s along the common interface Γ . On the outer boundary, homogenous Dirichlet values are given on Γ_D^f and Γ_D^s , whereas Neumann conditions are set on Γ_N^f and Γ_N^s

2 Presentation of the model problem

Let us consider the time interval $I = [0, T]$ and two rectangular domains

$$\Omega^f := (0, 4) \times (0, 1), \quad \Omega^s := (0, 4) \times (0, -1)$$

The interface is defined as $\Gamma := \overline{\Omega^f} \cap \overline{\Omega^s} = (0, 4) \times \{0\}$ and the remaining boundaries are shown in Fig. 1. Since our example might be treated as a simplified case of an FSI problem, in the text we will use the corresponding nomenclature. We will refer to the domain Ω^f as the *fluid domain* and the problem defined there as the *fluid problem*. Similarly, we will use *solid domain* and *solid problem* phrases. In this sense, the superscript “f” will always refer to entities connected to the fluid problem and “s” denotes the solid problem.

In the domain Ω^f we pose the heat equation

$$\partial_t v^f - \nu \Delta v^f + \beta \cdot \nabla v^f = g^f \quad \text{in } I \times \Omega^f, \tag{2.1}$$

and in the domain Ω^s we set the wave equation

$$\partial_t v^s - \lambda \Delta u^s - \delta \Delta v^s = g^s, \quad \partial_t u^s = v^s \quad \text{in } I \times \Omega^s, \tag{2.2}$$

written as a first order system. By v^f and v^s we denote the velocities of fluid and solid and by u^s the solid’s displacement. $\nu > 0$ is heat diffusion parameter, $\sqrt{\lambda}$ is the wave propagation speed and $\delta \geq 0$ a damping parameter. By $\beta \in \mathbb{R}^2$ we introduce a transport direction. The two problems are coupled on the interface Γ by the transmission conditions

$$v^f = v^s, \quad \lambda \partial_{\mathbf{n}_s} u^s + \nu \partial_{\mathbf{n}_f} v^f = 0 \quad \text{on } I \times \Gamma \tag{2.3}$$

which are similar to the kinematic and dynamic coupling conditions from fluid-structure interactions, compare [24, Sec. 3.1]. We use symbols \mathbf{n}_f and \mathbf{n}_s to distinguish between normal vectors for different space domains. On the interface it holds $\mathbf{n}_f = -\mathbf{n}_s$. To mimic the similarity to fluid-structure interaction formulations in Arbitrary

Eulerian Lagrangian (ALE) coordinates, see [12], we harmonically extend the solid deformation u^s to the fluid domain where we call it u^f

$$-\Delta u^f = 0 \text{ in } I \times \Omega^f, \quad u^f = u^s \quad \text{on } I \times \Gamma. \quad (2.4)$$

Usually, this artificial deformation defines the ALE domain map used to transfer the fluid problem to ALE coordinates, see [24, Sec. 5.3.5]. In the fluid domain, the left and right boundaries model free inflow and outflow, whereas the upper boundary models a no-slip condition. In the solid domain, the left and right boundary model a fixed solid, whereas the solid is free to move on the lower boundary, resulting in

$$\begin{aligned} u^f = v^f = 0 & \quad \text{on } I \times \Gamma_D^f, & \partial_{\mathbf{n}_f} v^f = \partial_{\mathbf{n}_f} u^f = 0 & \quad \text{on } I \times \Gamma_N^f \\ u^s = v^s = 0 & \quad \text{on } I \times \Gamma_D^s, & \lambda \partial_{\mathbf{n}_s} u^f + \delta \partial_{\mathbf{n}_s} v^s = 0 & \quad \text{on } I \times \Gamma_N^s. \end{aligned} \quad (2.5)$$

At time $t = 0$, all initial values are zero, i.e. $u^f(0) = u^s(0) = v^f(0) = v^s(0) = 0$. The exact values of the parameters read as

$$\nu = 0.001, \quad \beta = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad \lambda = 1000, \quad \delta = 0.1,$$

and they are chosen similar to the configuration of the fluid-structure interaction benchmark problems by Hron and Turek [18], to resemble a similar coupling structure. However, we consider a softer “solid”. The external forces are set to be products of functions of space and time $g^f(\mathbf{x}, t) := h^f(\mathbf{x})f(t)$ and $g^s(\mathbf{x}, t) := h^s(\mathbf{x})f(t)$ where $h^f(\mathbf{x}), h^s(\mathbf{x})$ are space components and $f(t)$ is a time component which models a periodic pulse

$$f(t) = \begin{cases} 1 & z \leq t \leq z + 0.1 \\ 0 & z + 0.1 < t \leq z + 1, \end{cases} \quad \forall z \in \mathbb{Z}.$$

We will consider two different configurations of the right hand side. In Configuration 2.1, the right hand side is concentrated in Ω^f where the space component consists of an exponential function centered around $(\frac{1}{2}, \frac{1}{2})$. For Configuration 2.2 we take a space component concentrated in Ω^s with an exponential function centered around $(\frac{1}{2}, -\frac{1}{2})$.

Configuration 2.1

$$h^f(\mathbf{x}) := e^{-((x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2)}, \quad h^s(\mathbf{x}) := 0,$$

Configuration 2.2

$$h^f(\mathbf{x}) := 0, \quad h^s(\mathbf{x}) := e^{-((x_1 - \frac{1}{2})^2 + (x_2 + \frac{1}{2})^2)}.$$

2.1 Continuous variational formulation

By $H^1(\Omega)$ we denote the space of L^2 -functions with the first weak derivative in $L^2(\Omega)$. To incorporate the Dirichlet boundary conditions on parts of the boundary into trial and test spaces we further define

$$H_0^1(\Omega; \Gamma) = \left\{ v \in H^1(\Omega) \mid v|_\Gamma = 0 \right\},$$

where $H^{-1}(\Omega) = H_0^1(\Omega; \Gamma)^*$ is the dual space. To incorporate the Dirichlet data in the fluid domain we introduce $V^f := H_0^1(\Omega^f; \Gamma_D^f)$ and, in the case of the solid domain, $V^s := H_0^1(\Omega^s; \Gamma_D^s)$. In the space-time domain $I \times \Omega$ we define the family of Hilbert spaces

$$X(V) = \left\{ v \in L^2(I, V) \mid \partial_t v \in L^2(I, V^*) \right\}.$$

For the fluid problem it will hold $(v^f, u^f) \in X^f := (X(V^f))^2$ and for the solid $(v^s, u^s) \in X^s := (X(V^s))^2$.

For any given domain G , which can take the place of the entire domain Ω , the fluid domain Ω^f or the solid domain Ω^s we denote by $(u, v)_G := \int_G u \cdot v \, dx$ the usual L^2 -inner product. Given an element of the dual space $f \in H^{-1}(G)$, we denote by $\langle f, \phi \rangle_{H^{-1}(G) \times H^1(\Omega)} = f(\phi)$ the duality pairing. For $f \in H^{-\frac{1}{2}}(\Gamma)$ we denote by

$$\langle f, \phi \rangle_\Gamma := \langle f, \phi \rangle_{H^{-\frac{1}{2}}(\Gamma) \times H^{\frac{1}{2}}(\Gamma)} = f(\phi)$$

the duality pairing on the interface. Finally, we introduce the abbreviations $(\cdot, \cdot)_f := (\cdot, \cdot)_{\Omega^f}$ and $\langle \cdot, \cdot \rangle_f := \langle \cdot, \cdot \rangle_{H^{-1}(\Omega^f) \times H^1(\Omega^f)}$ as well as the corresponding notation within the solid domain Ω^s .

We acquire the continuous variational formulation by multiplication of (2.1) and (2.4). Afterwards, we integrate in space and time using (2.5). The trial function is given by $\mathbf{U} = (\mathbf{U}^f, \mathbf{U}^s) \in X := X^f \times X^s$, which is further split to $\mathbf{U}^f = (v^f, u^f)$ and $\mathbf{U}^s = (v^s, u^s)$. Similarly, we set the test function as $\Phi = (\Phi^f, \Phi^s)$ and split it to $\Phi^f = (\varphi^f, \psi^f)$ and $\Phi^s = (\varphi^s, \psi^s)$. Given that, we define the space-time variational forms

$$B^f(\mathbf{U})(\Phi^f) := \int_I \langle \partial_t v^f, \varphi^f \rangle_f \, dt + \int_I a^f(\mathbf{U})(\Phi^f) \, dt + (v^f(0), \varphi^f(0))_f, \tag{2.6a}$$

$$B^s(\mathbf{U})(\Phi^s) := \int_I \langle \partial_t v^s, \varphi^s \rangle_s \, dt + \int_I \langle \partial_t u^s, \psi^s \rangle_s \, dt + \int_I a^s(\mathbf{U})(\Phi^s) \, dt + (v^s(0), \varphi^s(0))_s + (u^s(0), \psi^s(0))_s, \tag{2.6b}$$

$$F^f(\Phi^f) := \int_I (g^f, \varphi^f)_f \, dt,$$

$$F^s(\Phi^s) := \int_I (g^s, \varphi^s)_s \, dt$$

with

$$\begin{aligned} a^f(\mathbf{U})(\Phi^f) &:= (v \nabla v^f, \nabla \varphi^f)_f + (\beta \cdot \nabla v^f, \varphi^f)_f + (\nabla u^f, \nabla \psi^f)_f \\ &\quad - \langle \partial_{\mathbf{n}_f} u^f, \psi^f \rangle_\Gamma + \frac{\gamma}{h} \langle u^f - u^s, \psi^f \rangle_\Gamma \\ &\quad - \langle v \partial_{\mathbf{n}_f} v^f, \varphi^f \rangle_\Gamma + \frac{\gamma}{h} \langle v(v^f - v^s), \varphi^f \rangle_\Gamma, \end{aligned} \quad (2.7a)$$

$$\begin{aligned} a^s(\mathbf{U})(\Phi^s) &:= (\lambda \nabla u^s, \nabla \varphi^s)_s + (\delta \nabla v^s, \nabla \varphi^s)_s - (v^s, \psi^s)_s \\ &\quad + \langle v \partial_{\mathbf{n}_f} v^f, \varphi^s \rangle_\Gamma - \langle \delta \partial_{\mathbf{n}_s} v^s, \varphi^s \rangle_\Gamma. \end{aligned} \quad (2.7b)$$

All the Laplacian terms were integrated by parts and the dynamic coupling condition was added. The kinematic coupling condition was incorporated into the fluid problem, while the dynamic condition became a part of the solid problem. The Dirichlet boundary conditions over the interface Γ were formulated in a weak sense using Nitsche's method [21]. The parameter γ can be seen as a penalization parameter enforcing $u^f = u^s$ and $v^f = v^s$ weakly. The parameter $\gamma > 0$ should be large enough to counter-balance different constants, like the one from the inverse estimate. We set $\gamma = 10$, while h is the mesh size. Too small values for γ might cause a discrepancy from the Dirichlet condition, too large values worsen the conditioning of the resulting system. We refer to [5] for the analysis of a full fluid-structure interaction system with Nitsche coupling on the interface.

The compact version of the variational problem presents itself as:

Problem 2.1 Find $\mathbf{U} \in X$ such that

$$\begin{aligned} B^f(\mathbf{U})(\Phi^f) &= F^f(\Phi^f) \\ B^s(\mathbf{U})(\Phi^s) &= F^s(\Phi^s) \end{aligned}$$

for all $\Phi^f \in X^f$ and $\Phi^s \in X^s$.

This coupled heat-wave system carries some similarities to fluid-structure interactions [24] in terms of the parabolic / hyperbolic type of the equations and also in the set of interface coupling conditions of Dirichlet / Neuman type. It can be considered as a further simplification of a linear fluid-structure interaction system coupling the Stokes equations with the Navier-Lame equations, which has been extensively studied [1, 13, 14]. Here, the existence and regularity of the solution is shown, compare [16] for an overview of the results. Zhang and Zuazua [29] analyze the long time behavior of the coupled scalar heat and wave system, similar to our set of equations. However, as further simplification, our model problem contains a strong damping term within the heat equation. Given a sufficiently smooth domain, e.g. C^2 -parametrizable or convex polygonal boundary, right hand side g^f, g^s in L^2 and initial values $u^s \in H^1(\Omega^s)$ and $v^s \in L^2(\Omega^s)$, $v^f \in L^2(\Omega^s)$, $v^f \in L^2(\Omega^f)$, the solution has at least the regularity $v^f \in L^2(I, H^1(\Omega^f))$ and $u^s \in W^{1,\infty}(I, H^1(\Omega^s))$.

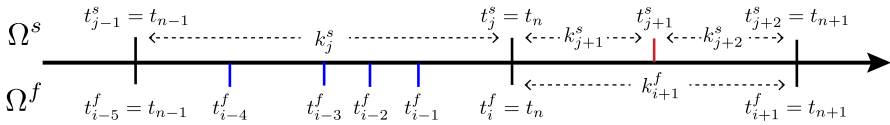


Fig. 2 Visualization of the temporal mesh: macro mesh with steps t_{n-1}, t_n and t_{n+1} and subdivision into two distinct partitionings for fluid and solid, t_i^f and t_j^s , respectively

2.2 Semi-discrete Petrov–Galerkin formulation

One of the main challenges emerging from the discretization of Problem 2.1 is the construction of a satisfactory time interval partitioning. Our main objectives include:

- 1. Allowing for different time-step sizes (possibly non-uniform) in both subproblems**

We introduce two distinct subdivisions I^f and I^s of the interface $I = [0, T]$

$$0 = t_0^f < t_1^f < \dots < t_{N^f}^f = T, \quad 0 = t_0^s < t_1^s < \dots < t_{N^s}^s = T.$$

with $N^f, N^s \in \mathbb{N}$. We will refer to these meshes as *micro time meshes* and we will denote them by I^f and I^s , respectively. Their step sizes $k_n^f := t_n^f - t_{n-1}^f$ and $k_n^s := t_n^s - t_{n-1}^s$ can be non-uniform and fluid- and solid-steps do not have to match.

- 2. Handling coupling conditions**

Based on the micro partitionings I^f and I^s we introduce the coarse time mesh of the interval $I = [0, T]$ that consists of all discrete points in time which are shared by both of the subproblems, i.e.

$$0 = t_0 < t_1 < \dots < t_N = T,$$

with $N \leq \min\{N^f, N^s\}$. For each $n = 0 \dots, N$ there exist two indices $i \in \{0, \dots, N^f\}$ and $j \in \{0, \dots, N^s\}$ such that $t_n = t_i^f = t_j^s$. We will refer to this mesh as the *macro time mesh* and denote it by I_N .

In practice, this mesh structure is generated in a top-down design: we start with a shared macro mesh $I^f = I^s = I_N$ and generate finer meshes by successive refinement of the single steps. After each refinement cycle we identify common nodes $t_i^f = t_j^s$ to generate the new macro mesh. We define the grid sizes as:

$$k^f := \max_{n=1, \dots, N^f} k_n^f, \quad k^s := \max_{n=1, \dots, N^s} k_n^s, \quad k := \max\{k^f, k^s\}.$$

Figure 2 visualizes this construction.

As trial spaces, we choose functions that are continuous in time on the complete interval $I = [0, T]$ and piecewise linear on the subdivisions. In space, they take values

in the corresponding Sobolev spaces V^f and V^s which have been introduced above:

$$X_k^{f,1} = \{v \in C(\bar{I}, L^2(\Omega^f)) \mid v|_{(t_{n-1}^f, t_n^f]} \in \mathcal{P}_1(V^f) \text{ for all } n = 1, \dots, N^f\},$$

$$X_k^{s,1} = \{v \in C(\bar{I}, L^2(\Omega^s)) \mid v|_{(t_{n-1}^s, t_n^s]} \in \mathcal{P}_1(V^s) \text{ for all } n = 1, \dots, N^s\}.$$

We set $X_k^f := (X_k^{f,1})^2$ and $X_k^s := (X_k^{s,1})^2$. The test spaces Y_k^f and Y_k^s are defined likewise, however, these functions are not necessarily continuous on $I = [0, T]$ and they are piecewise constant on each of the micro steps

$$Y_k^{f,0} = \{v \in L^2(\bar{I}, L^2(\Omega^f)) \mid v|_{(t_{n-1}^f, t_n^f]} \in \mathcal{P}_0(V^f) \text{ for all } n = 1, \dots, N^f\},$$

$$Y_k^{s,0} = \{v \in L^2(\bar{I}, L^2(\Omega^s)) \mid v|_{(t_{n-1}^s, t_n^s]} \in \mathcal{P}_0(V^s) \text{ for all } n = 1, \dots, N^s\}.$$

with $Y_k^f := (Y_k^{f,0})^2$ and $Y_k^s := (Y_k^{s,0})^2$. By $\mathcal{P}_r(H)$ we denote the space of piecewise polynomials with degree r and values in H .

With that at hand, we can pose a semi-discrete variational problem:

Problem 2.2 Find $\mathbf{U}_k \in X_k := X_k^f \times X_k^s$ such that:

$$B^f(\mathbf{U}_k)(\Phi_k^f) = F^f(\Phi_k^f)$$

$$B^s(\mathbf{U}_k)(\Phi_k^s) = F^s(\Phi_k^s)$$

for all $\Phi_k^f \in Y_k^f$ and $\Phi_k^s \in Y_k^s$.

Since the operators B^f and B^s are linear, the resulting scheme is equivalent to the Crank-Nicolson scheme up to the numerical quadrature of the right hand sides F^f and F^s , see also [15,28]. Operator B^f depends on the solid solution (v_k^s, u_k^s) and, vice versa the solid operator B^s depends on the fluid solution v_k^f , see (2.6a), (2.7a) and (2.6b), (2.7b). Hence, we have to integrate discrete (so, piecewise linear or piecewise constant) functions from the one micro mesh on the other. To be precise, on the macro step $I_n = (t_{n-1}, t_n]$ it will be necessary to evaluate integrals which couple functions on both discrete and non-matching meshes, like

$$\int_{t_{n-1}}^{t_n} \langle v \partial_{\mathbf{n}^f} v^f, \varphi^s \rangle_{\Gamma}. \tag{2.8}$$

This will ask for a costly interpolation of the fluid velocity v^f to the solid mesh on each macro step, compare Fig. 2. To simplify this coupling, we will transfer discrete functions by means of a simple interpolation into the space of linear functions on the macro mesh, i.e. for $v \in X_k^f$ or $v \in X_k^s$ we define

$$t \in (t_{n-1}, t_n] \quad I_k v(t) := \frac{t_n - t}{k_n} v(t_{n-1}) + \frac{t - t_{n-1}}{k_n} v(t_n) \tag{2.9}$$

which can be easily evaluated on both submeshes. Hereby, (2.8) is approximated by

$$\int_{t_{n-1}}^{t_n} \langle v \partial_{\mathbf{n}_f} v^f, \varphi^s \rangle_\Gamma \approx \int_{t_{n-1}}^{t_n} \langle v \partial_{\mathbf{n}_f} I_k v^f, \varphi^s \rangle_\Gamma. \tag{2.10}$$

3 Decoupling methods

Even though Problem 2.2 is discretized in time, it is still coupled across the interface. That makes solving the subproblems independently impossible. To deal with this obstacle, we chose to use an iterative approach on each of the subintervals I_n and introduce decoupling strategies. Throughout this section, we only consider the semi-discrete problem. Hence, for better readability, we will skip the index “k”. For a fixed time interval I_n the i -th iteration of a decoupling method consists of the following steps:

1. Using the solution of the solid subproblem from the previous iteration $\mathbf{U}^{s,(i-1)}$, we set the boundary conditions on the interface at the time t_n , solve the fluid problem and get the solution $\mathbf{U}^{f,(i)}$.
2. Similarly, we use the solution $\mathbf{U}^{f,(i)}$ for setting the boundary conditions of the solid problem and obtain an intermediate solution $\tilde{\mathbf{U}}^{s,(i)}$.
3. We apply a decoupling function to the intermediate solution $\tilde{\mathbf{U}}^{s,(i)}$ and acquire $\mathbf{U}^{s,(i)}$.

This procedure is visualized by

$$\mathbf{U}^{s,(i-1)} \xrightarrow[\text{subproblem}]{\text{fluid}} \mathbf{U}^{f,(i)} \xrightarrow[\text{subproblem}]{\text{solid}} \tilde{\mathbf{U}}^{s,(i)} \xrightarrow[\text{function}]{\text{decoupling}} \mathbf{U}^{s,(i)}.$$

The main challenge emerges from the transition between $\tilde{\mathbf{U}}^{s,(i)}$ and $\mathbf{U}^{s,(i)}$. In the next subsections, we will present two techniques. The first one is the relaxation method described in Sect. 3.1. The second one, in Sect. 3.2, is the shooting method. We clarify how the intermediate solution $\tilde{\mathbf{U}}^{s,(i)}$ is obtained from $\mathbf{U}^{s,(i-1)}$ by the definition of Problem 3.1.

Problem 3.1 For a given $\mathbf{U}^{s,(i-1)} \in X_k^s$, find $\mathbf{U}^{f,(i)} \in X_k^f$ and $\tilde{\mathbf{U}}^{s,(i)} \in X_k^s$ such that:

$$\begin{aligned} B_n^f \begin{pmatrix} \mathbf{U}^{f,(i)} \\ \mathbf{U}^{s,(i-1)} \end{pmatrix} (\Phi^f) &= F_n^f(\Phi^f) \\ B_n^s \begin{pmatrix} \mathbf{U}^{f,(i)} \\ \tilde{\mathbf{U}}^{s,(i)} \end{pmatrix} (\Phi^s) &= F_n^s(\Phi^s) \end{aligned}$$

for all $\Phi^f \in Y_k^f$ and $\Phi^s \in Y_k^s$. By B_n^f and B_n^s we denote restrictions of forms B^f and B^s to I_n . Forms F_n^f and F_n^s are defined accordingly.

3.1 Relaxation method

The first of the presented methods consists of a simple interpolation operator being an example of a fixed point method. It contains the iterated solution of each of the two sub-problems, taking the interface values from the last iteration of the other problem. For reasons of stability, such explicit partitioned iteration usually requires the introduction of a damping parameter. Here, we only consider fixed damping parameters.

Definition 3.1 (Relaxation Function) Let $\mathbf{U}^{s,(i-1)} \in X_k^s$ and $\tilde{\mathbf{U}}^{s,(i)} \in X_k^s$ be the solid solution of Problem 3.1. Then for $\tau \in [0, 1]$ the relaxation function $R : X_k^s \rightarrow X_k^s$ is defined as:

$$R(\mathbf{U}^{s,(i-1)}) := \tau \tilde{\mathbf{U}}^{s,(i)} + (1 - \tau)\mathbf{U}^{s,(i-1)}$$

Assuming that we already know the value $\mathbf{U}^s(t_{n-1})$, we pose

$$\mathbf{U}^{s,(0)}(t_n) := \mathbf{U}^s(t_{n-1}), \quad \mathbf{U}^{s,(i)}(t_n) := R(\mathbf{U}^{s,(i-1)})(t_n) \text{ for } i = 1, 2, \dots$$

The stopping criterion is based on checking how far the computed solution is from the fixed point. We evaluate the L^∞ norm on the interface degrees of freedom. If for a given tolerance $tol > 0$ we have $\|\tilde{\mathbf{U}}^{s,(i+1)}(t_n) - \mathbf{U}^{s,(i)}(t_n)\|_{L^\infty(\Gamma)} < tol$, we stop the iteration and accept the last approximation.

3.2 Shooting method

Here we present another iterative method where we define a root-finding problem on the interface. We use the Newton method with a matrix-free GMRES method for approximation of the inverse of the Jacobian.

Definition 3.2 (Shooting Function) Let $\mathbf{U}^{s,(i-1)} \in X_k^s$ and $\tilde{\mathbf{U}}^{s,(i)} \in X_k^s$ be the solid solution of Problem 3.1. Then the shooting function $S : X_k^s \rightarrow (L^2(\Gamma))^2$ is defined as:

$$S(\mathbf{U}^{s,(i-1)}) := \left(\mathbf{U}^{s,(i-1)}(t_n) - \tilde{\mathbf{U}}^{s,(i)}(t_n) \right) \Big|_{\Gamma} \tag{3.1}$$

We aim to find the root of function (3.1). To do so, we employ the Newton method

$$S'(\mathbf{U}^{s,(i-1)})\mathbf{d}^{(i)} = -S(\mathbf{U}^{s,(i-1)}).$$

In each iteration of the Newton method, the greatest difficulty causes computing and inverting the Jacobian $S'(\mathbf{U}^{s,(i-1)})$. Instead of approximating all entries of the Jacobian matrix, we consider an approximation of the matrix-vector product only. Since the Jacobian matrix-vector product can be interpreted as a directional derivative, one can assume

$$S'(\mathbf{U}^{s,(i-1)})\mathbf{d}^{(i)} \approx \frac{S(\mathbf{U}^{s,(i-1)} + \varepsilon\mathbf{d}^{(i)}) - S(\mathbf{U}^{s,(i-1)})}{\varepsilon}. \tag{3.2}$$

In principle, the vector $\mathbf{d}^{(i)}$ is not known. Thus, the formula above can not be used for solving the system directly. However, it is possible to use this technique with iterative solvers which only require the computation of matrix-vector products. Because we did not want to assume much structure of the operator (3.2), we chose the matrix-free GMRES method. Such matrix-free Newton-Krylov methods are frequently used if the Jacobian is not available or too costly for evaluation [19]. Once $\mathbf{d}^{(i)}$ is computed, we set

$$\mathbf{U}^{s,(0)}|_F(t_n) := \mathbf{U}^s(t_{n-1})|_F, \quad \mathbf{U}^{s,(i)}|_F(t_n) := \mathbf{U}^{s,(i-1)}|_F(t_n) + \mathbf{d}^{(i)} \text{ for } i = 1, 2, \dots$$

Here, we stop iterating when the L^∞ norm of $S(\mathbf{U}^{s,(i)})$ is sufficiently small and then we accept the last available approximation.

We note that the method presented here is similar to the one presented in [11], where the authors also introduced a root-finding problem on the interface and solved it with a quasi-Newton method. The main difference lies in the approximation of the inverse of the Jacobian. Instead of using a matrix-free linear solver, there the Jacobian is approximated by solving a least-squares problem.

3.3 Numerical comparison of the performance

In Figs. 3 and 4 we present the comparison of the performance of both methods based on the number of micro time-steps. We assumed that both micro and macro time-steps have uniform sizes. We performed the simulations in the case of no micro time-stepping ($N^f = N^s = N$), micro time-stepping in the fluid subdomain ($N^f = 10N$, $N^s = N$) and the solid subdomain ($N^f = N$, $N^s = 10N$). Figure 3 shows results for the right hand side according to Configuration 2.1. Figure 4 corresponds to Configuration 2.2. We investigated one macro time-step $I_2 = [0.02, 0.04]$. We set the relaxation parameter to $\tau = 0.7$. Both methods are very robust concerning the number of micro time-steps. The relaxation method, as expected, has a linear convergence rate. In both cases, despite the nested GMRES method, the performance of the shooting method is much better. For Configuration 2.1, the relaxation method needs 13 iterations to converge. The shooting method needs only 2 iterations of the Newton method (which is the reason why each of the graphs in Fig. 3 displays only two evaluations of the error) and overall requires 6 evaluations of the decoupling function. In the case of Configuration 2.2, both methods need more iterations to reach the same level of accuracy. The number of iterations of the relaxation method increases to 20 while the shooting method needs 3 iterations of the Newton method and 11 evaluations of the decoupling function.

In Figs. 5 and 6 we show the number of evaluations of the decoupling function needed to reach the stopping criteria throughout the complete time interval $I = [0, 1]$ for $N = 50$. Similarly, we performed the simulations in the case of no micro time-stepping, micro time-stepping in the fluid and the solid subdomain. We considered both Configuration 2.1 and 2.2. In the case of Configuration 2.1, the number of evaluations of the decoupling function using the relaxation method varied between 14 and 15. For the shooting function, this value was mostly equal to 6 with a few exceptions when

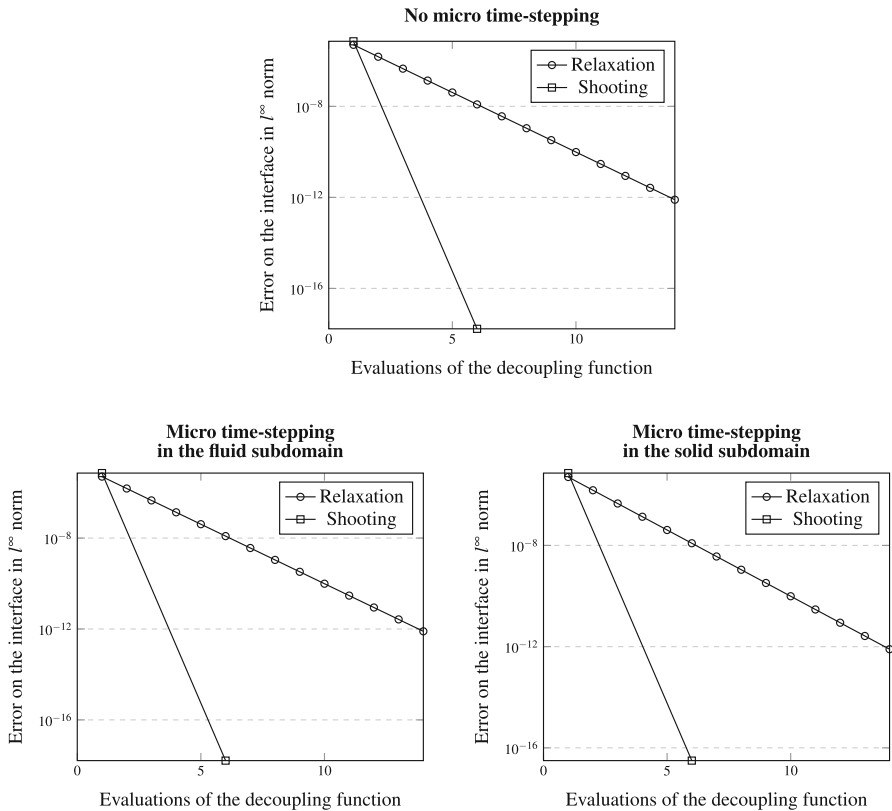


Fig. 3 Performance of decoupling methods for Configuration 2.1 in one macro time-step in the case of $N^f = N^s = N$ (top), $N^f = 10N$ and $N^s = N$ (left), $N^f = N$ and $N^s = 10N$ (right)

only 5 evaluations were needed. For Configuration 2.2, the relaxation method needed between 18 and 21 iterations while for the shooting method it was almost always equal to 11. For each configuration, graphs corresponding to no micro time-stepping and micro time-stepping in the fluid subdomain are the same, while introducing micro time-stepping in the solid subdomain resulted in slight variations. For both decoupling methods, the independence of the performance from the number of micro time-steps extends to the whole time interval $I = [0, 1]$.

4 Goal oriented estimation

In Section 1 we formulated the semi-discrete problem enabling usage of different time-step sizes in fluid and solid subdomains, whereas in Section 2 we presented methods designed to efficiently solve such problems. However, so far the choice of the step sizes was purely arbitrary. In this section, we are going to present an easily localized

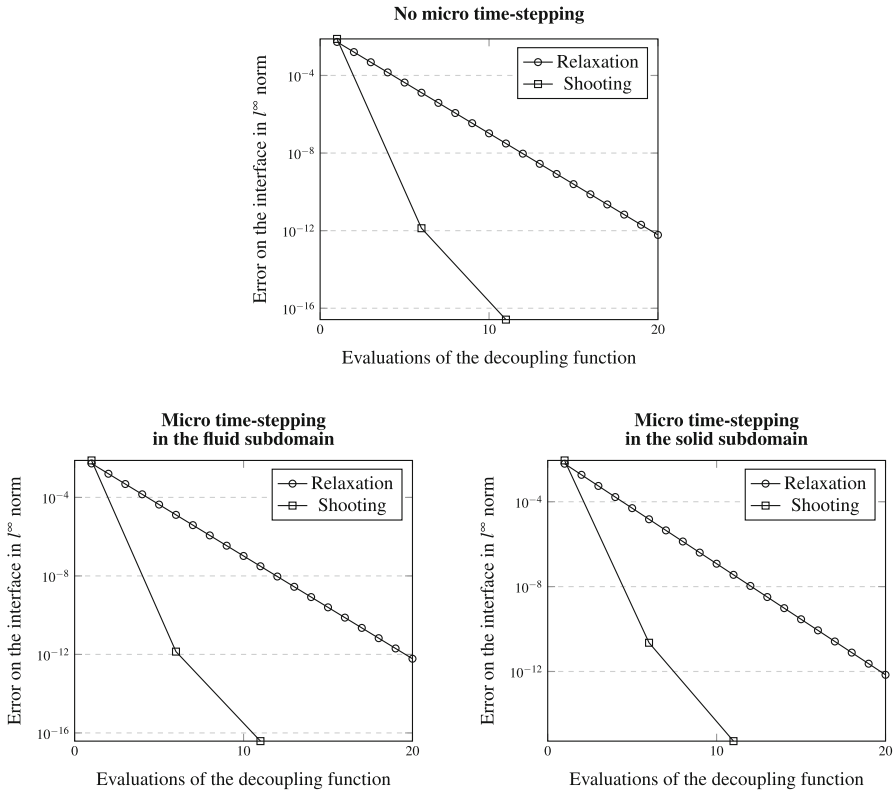


Fig. 4 Performance of decoupling methods for Configuration 2.2 in one macro time-step in the case of $N^f = N^s = N$ (top), $N^f = 10N$ and $N^s = N$ (left), $N^f = N$ and $N^s = 10N$ (right)

error estimator, which can be used as a criterion for the adaptive choice of the time-step size.

For the construction of the error estimator, we used the dual weighted residual (DWR) method [2]. Given a differentiable goal functional $J : X \rightarrow \mathbb{R}$, our aim is finding a way to approximate $J(\mathbf{U}) - J(\mathbf{U}_k)$, where \mathbf{U} is the exact solution (Problem 2.1) and \mathbf{U}_k is the semi-discrete solution (Problem 2.2). The goal functional is split into two parts $J^f : X^f \rightarrow \mathbb{R}$ and $J^s : X^s \rightarrow \mathbb{R}$ which refer to the fluid and solid subdomains, respectively

$$J(\mathbf{U}) := J^f(\mathbf{U}^f) + J^s(\mathbf{U}^s).$$

The DWR method is based on the following constrained optimization problem

$$J(\mathbf{U}) = \min! \quad \text{such that} \quad B(\mathbf{U})(\Phi) = F(\Phi) \text{ for all } \Phi \in X,$$

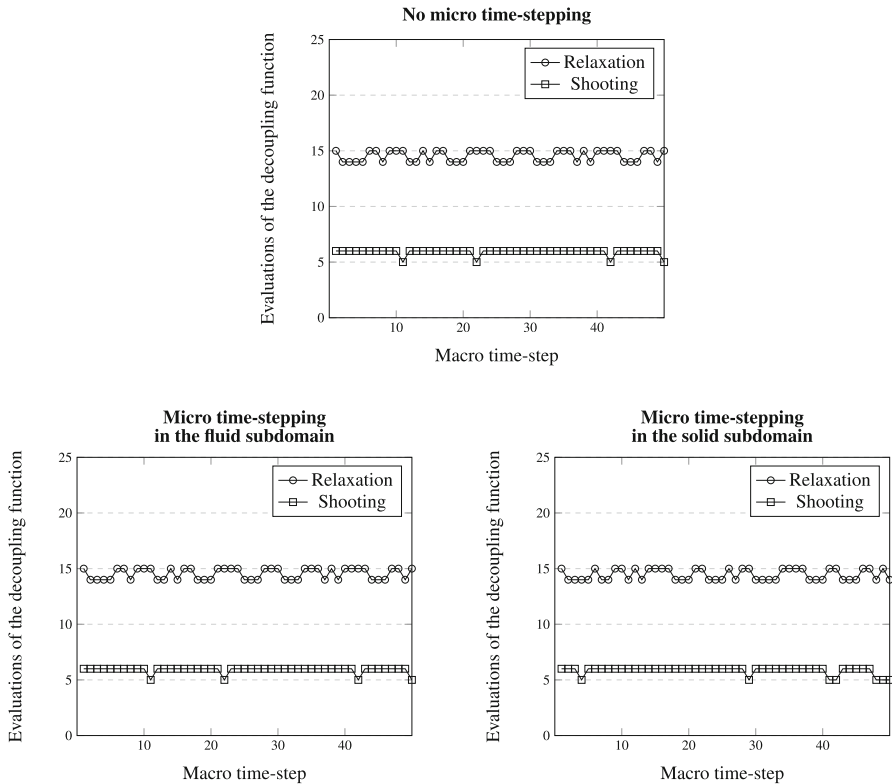


Fig. 5 Number of evaluations of the decoupling functions for Configuration 2.1 needed for convergence on the time interval $I = [0, 1]$ for $N = 50$ in the case of $N^f = N^s = N$ (top), $N^f = 10N$ and $N^s = N$ (left), $N^f = N$ and $N^s = 10N$ (right)

where

$$B(\mathbf{U})(\Phi) := B^f(\mathbf{U})(\Phi^f) + B^s(\mathbf{U})(\Phi^s),$$

$$F(\Phi) := F^f(\Phi^f) + F^s(\Phi^s).$$

Solving this problem corresponds to finding stationary points of a Lagrangian \mathcal{L}

$$\mathcal{L}(\mathbf{U}, \mathbf{Z}) := J(\mathbf{U}) + F(\mathbf{Z}) - B(\mathbf{U})(\mathbf{Z}).$$

Because form B describes a linear problem, finding stationary points of \mathcal{L} is equivalent to solving the following problem:

Problem 4.1 For a given $\mathbf{U} \in X$ being the solution of Problem 2.1, find $\mathbf{Z} \in X$ such that:

$$B(\Psi, \mathbf{Z}) = J'_U(\Psi)$$

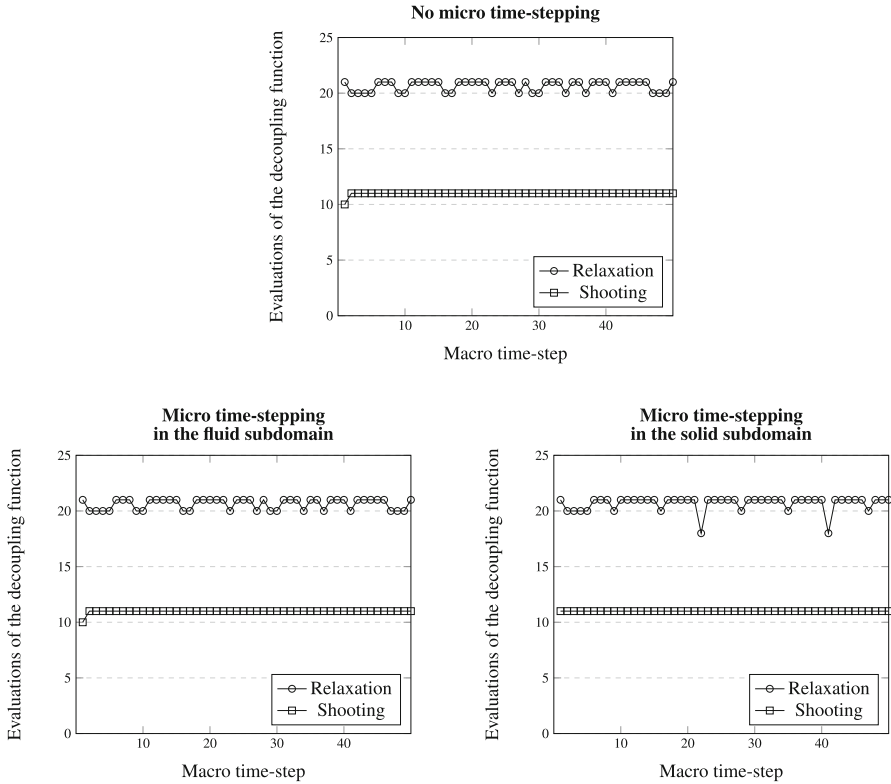


Fig. 6 Number of evaluations of the decoupling functions for Configuration 2.2 needed for convergence on the time interval $I = [0, 1]$ for $N = 50$ in the case of $N^f = N^s = N$ (top), $N^f = 10N$ and $N^s = N$ (left), $N^f = N$ and $N^s = 10N$ (right)

for all $\Psi \in X$.

The solution \mathbf{Z} is called an *adjoint solution*. By $J'_U(\Psi)$ we denote the Gateaux derivative of $J(\cdot)$ at \mathbf{U} in direction of the test function Ψ .

4.1 Adjoint problem

4.1.1 Continuous variational formulation

As the first step in decoupling the Problem 4.1, we would like to split the form B into forms corresponding to fluid and solid subproblems. However, we can not fully reuse the forms (2.7a) and (2.7b) because of the interface terms—the forms have to be sorted regarding test functions. The adjoint solution is written as $\mathbf{Z} = (\mathbf{Z}^f, \mathbf{Z}^s)$, each split to $\mathbf{Z}^f = (z^f, y^f) \in X^f$ and $\mathbf{Z}^s = (z^s, y^s) \in X^s$. Likewise, for the test function, we set $\Psi = (\Psi^f, \Psi^s)$ with $\Psi^f = (\eta^f, \xi^f) \in X^f$ and $\Psi^s = (\eta^s, \xi^s) \in X^s$. Then we

introduce the adjoint forms, sorted by the test functions

$$B(\Psi)(\mathbf{Z}) := \tilde{B}^f(\Psi^f)(\mathbf{Z}) + \tilde{B}^s(\Psi^s)(\mathbf{Z}),$$

we have

$$\begin{aligned} \tilde{B}^f(\Psi^f)(\mathbf{Z}) &:= - \int_I \langle \partial_t z^f, \eta^f \rangle_f dt + \int_I \tilde{a}^f(\Psi^f)(\mathbf{Z}) dt + (\eta^f(T), z^f(T))_f, \\ \tilde{B}^s(\Psi^s)(\mathbf{Z}) &:= - \int_I \langle \partial_t z^s, \eta^s \rangle_s dt - \int_I \langle \partial_t y^s, \xi^s \rangle_s dt + \int_I \tilde{a}^s(\Psi^s)(\mathbf{Z}) dt \\ &\quad + (\eta^s(T), z^s(T))_s + (\xi^s(T), y^s(T))_s \end{aligned}$$

and

$$\begin{aligned} \tilde{a}^f(\Psi^f)(\mathbf{Z}) &:= (v \nabla \eta^f, \nabla z^f)_f + (\beta \cdot \nabla \eta^f, z^f)_f + (\nabla \xi^f, \nabla y^f)_f \\ &\quad - \langle \partial_{\mathbf{n}_f} \xi^f, y^f \rangle_\Gamma + \frac{\gamma}{h} \langle \xi^f, y^f \rangle_\Gamma - \langle v \partial_{\mathbf{n}_f} \eta^f, z^f \rangle_\Gamma \\ &\quad + \frac{\gamma}{h} \langle v \eta^f, z^f \rangle_\Gamma + \langle v \partial_{\mathbf{n}_f} \eta^f, z^s \rangle_\Gamma, \\ \tilde{a}^s(\Psi^s)(\mathbf{Z}) &:= (\lambda \nabla \xi^s, \nabla z^s)_s + (\delta \nabla \eta^s, \nabla z^s)_s - (\eta^s, y^s)_s \\ &\quad - \frac{\gamma}{h} \langle \xi^s, y^f \rangle_\Gamma - \frac{\gamma}{h} \langle v \eta^s, z^f \rangle_\Gamma - \langle \delta \partial_{\mathbf{n}_s} \eta^s, z^s \rangle_\Gamma. \end{aligned}$$

We have applied integration by parts in time which reveals that the adjoint problem runs backward in time. That leads to the formulation of a continuous adjoint variational problem:

Problem 4.2 For a given $\mathbf{U} \in X$ being the solution of Problem 2.1, find $\mathbf{Z} \in X$ such that:

$$\begin{aligned} \tilde{B}^f(\Psi^f)(\mathbf{Z}) &= (J^f)'_{\mathbf{U}}(\Psi^f) \\ \tilde{B}^s(\Psi^s)(\mathbf{Z}) &= (J^s)'_{\mathbf{U}}(\Psi^s) \end{aligned}$$

for all $\Psi^f \in X^f$ and $\Psi^s \in X^s$.

4.1.2 Semi-discrete Petrov–Galerkin formulation

The semi-discrete formulation for the adjoint problem is similar to the one of the primal problem. The main difference lies in the fact that this time trial functions are piecewise constant in time $\mathbf{Z}_k \in Y_k := Y_k^f \times Y_k^s$, while test functions are piecewise linear in time $\Psi_k^f \in X_k^f, \Psi_k^s \in X_k^s$. With that at our disposal, we can formulate a semi-discrete adjoint variational problem:

Problem 4.3 For a given $\mathbf{U} \in X$ being the solution of Problem 2.1, find $\mathbf{Z}_k \in Y_k$ such that:

$$\begin{aligned} \tilde{B}^f(\Psi_k^f)(\mathbf{Z}_k) &= (J^f)'_{\mathbf{U}}(\Psi_k^f), \\ \tilde{B}^s(\Psi_k^s)(\mathbf{Z}_k) &= (J^s)'_{\mathbf{U}}(\Psi_k^s) \end{aligned}$$

for all $\Psi_k^f \in X_k^f$ and $\Psi_k^s \in X_k^s$.

After formulating the problem in a semi-discrete manner, the decoupling methods from Sect. 3 can be applied.

4.2 A posteriori error estimate

We define the primal residual, split into parts corresponding to the fluid and solid subproblems

$$\rho(\mathbf{U})(\Phi) := \rho^f(\mathbf{U})(\Phi^f) + \rho^s(\mathbf{U})(\Phi^s),$$

where

$$\begin{aligned} \rho^f(\mathbf{U})(\Phi^f) &:= F^f(\Phi^f) - B^f(\mathbf{U})(\Phi^f), \\ \rho^s(\mathbf{U})(\Phi^s) &:= F^s(\Phi^s) - B^s(\mathbf{U})(\Phi^s). \end{aligned}$$

Similarly, we establish the adjoint residual resulting from the adjoint problem

$$\rho^*(\mathbf{Z})(\Psi) := \rho^{f,*}(\mathbf{Z})(\Psi^f) + \rho^{s,*}(\mathbf{Z})(\Psi^s)$$

with

$$\begin{aligned} \rho^{f,*}(\mathbf{Z})(\Psi^f) &:= (J^f)'_{\mathbf{U}}(\Psi^f) - \tilde{B}^f(\Psi^f)(\mathbf{Z}), \\ \rho^{s,*}(\mathbf{Z})(\Psi^s) &:= (J^s)'_{\mathbf{U}}(\Psi^s) - \tilde{B}^s(\Psi^s)(\mathbf{Z}) \end{aligned}$$

Becker and Rannacher [2] introduced the a posteriori error representation:

$$\begin{aligned} J(\mathbf{U}) - J(\mathbf{U}_k) &= \frac{1}{2} \min_{\Phi_k \in Y_k} \rho(\mathbf{U}_k)(\mathbf{Z} - \Psi_k) + \frac{1}{2} \min_{\Psi_k \in X_k} \rho^*(\mathbf{Z}_k)(\mathbf{U} - \Phi_k) \\ &\quad + \mathcal{O}(|\mathbf{U} - \mathbf{U}_k|^3, |\mathbf{Z} - \mathbf{Z}_k|^3) \end{aligned} \tag{4.3}$$

This identity can be used to derive an a posteriori error estimate. Two steps of approximation are required: first, the third order remainder is neglected and second, the approximation errors $\mathbf{Z} - \Psi_k$ and $\mathbf{U} - \Phi_k$, the *weights*, are replaced by interpolation errors $\mathbf{Z} - i_k \mathbf{Z}$ and $\mathbf{U} - i_k \mathbf{U}$. These errors are then replaced by discrete reconstructions, since the exact solutions $\mathbf{U}, \mathbf{Z} \in X$ are not available. See [20,27] for a discussion of different reconstruction schemes. Due to these approximation steps, this estimator is

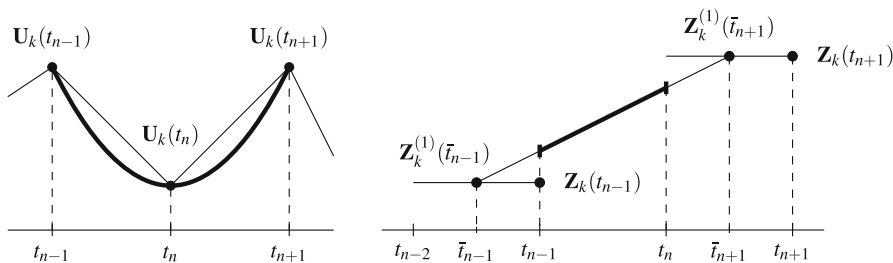


Fig. 7 Reconstruction of the primal solution $U_k^{(2)}$ (left) and the adjoint solution $Z_k^{(1)}$ (right)

not precise and it does not result in rigorous bounds. The estimator consists of a primal and adjoint component. Each of them is split again into a fluid and a solid counterpart

$$\sigma_k := \theta^f + \theta^s + \vartheta^f + \vartheta^s. \tag{4.4}$$

The primal estimators are derived from the primal residuals using U_k and Z_k being the solutions to Problems 2.2 and 4.3, respectively

$$\theta^f := \frac{1}{2} \rho^f(U_k)(Z_k^{f,(1)} - Z_k^f), \quad \theta^s := \frac{1}{2} \rho^s(U_k)(Z_k^{s,(1)} - Z_k^s).$$

The adjoint reconstructions $Z_k^{f,(1)}$ and $Z_k^{s,(1)}$ approximating the exact solution are constructed from Z_k using linear extrapolation (see Fig. 7, right)

$$Z_k^{(1)}(t)|_{I_n} := \frac{t - \bar{t}_{n+1}}{\bar{t}_{n-1} - \bar{t}_{n+1}} Z_k(t_{n-1}) + \frac{t - \bar{t}_{n-1}}{\bar{t}_{n+1} - \bar{t}_{n-1}} Z_k(t_{n+1})$$

with the interval midpoints $\bar{t}_n := \frac{t_n + t_{n-1}}{2}$. To simplify the notation, we defined the reconstruction on the macro time-mesh level. However, we use this idea in the same spirit on each of the micro meshes I_n^f and I_n^s . The adjoint estimators are based on the adjoint residuals

$$\vartheta^f := \frac{1}{2} \rho^{f,*}(Z_k)(U_k^{f,(2)} - U_k^f), \quad \vartheta^s := \frac{1}{2} \rho^{s,*}(Z_k)(U_k^{s,(2)} - U_k^s).$$

The primal reconstructions $U_k^{f,(2)}$ and $U_k^{s,(2)}$ are extracted from U_k using quadratic reconstruction (see Fig. 7, left). We perform the reconstruction on the micro time-mesh level on local patches consisting of two neighboring micro time-steps. In general, the patch structure does not have to coincide with the micro and macro time-mesh structure—two micro time-steps being in the same local patch do not have to be in the same macro time-step. Additionally, we demand two micro time-steps from the same local patch to have the same length.

We compute the effectivity of the error estimate using

$$\text{eff}_k := \frac{\sigma_k}{J(U_{\text{exact}}) - J(U_k)},$$

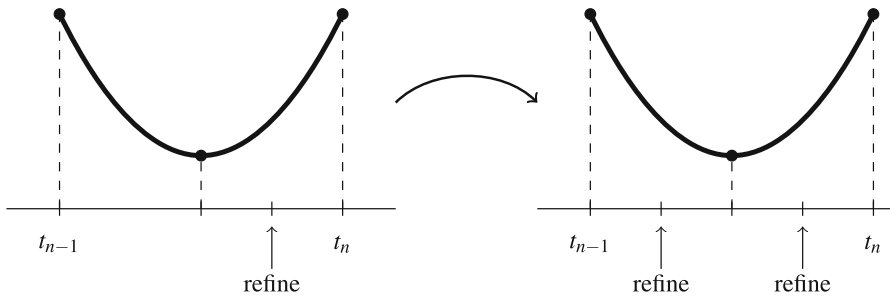


Fig. 8 An example of preserving the local patch structure during the marking procedure: if the time-step is refined, the other time-step belonging to the same patch will also be refined

where $J(\mathbf{U}_{\text{exact}})$ can be approximated by extrapolation in time.

4.3 Adaptivity

The residuals (4.4) can be easily localized by restricting them to a specific subinterval, e.g., the primal fluid-residuals θ^f , restricted to $I_n^f := [t_{n-1}^f, t_n^f]$ is denoted as θ_n^f , see Sect. 2.2 and Fig. 2. All further residuals θ^s, ϑ^f and ϑ^s are restricted in the same fashion. We then can compute an average for each of the components

$$\bar{\sigma}_k := \frac{1}{2N^f} \sum_{n=1}^{N^f} (|\theta_n^f| + |\vartheta_n^f|) + \frac{1}{2N^s} \sum_{n=1}^{N^s} (|\theta_n^s| + |\vartheta_n^s|). \tag{4.5}$$

This way we can obtain satisfactory refining criteria as

$$\begin{aligned} (|\theta_n^f| \geq \bar{\sigma}_k \text{ or } |\vartheta_n^f| \geq \bar{\sigma}_k) &\implies \text{refine } I_n^f = [t_{n-1}^f, t_n^f], \\ (|\theta_n^s| \geq \bar{\sigma}_k \text{ or } |\vartheta_n^s| \geq \bar{\sigma}_k) &\implies \text{refine } I_n^s = [t_{n-1}^s, t_n^s]. \end{aligned} \tag{4.6}$$

Taking into account the time interval partitioning structure, we arrive with the following algorithm:

1. Mark subintervals using the refining criteria (4.6).
2. Adjust the local patch structure—in case only one subinterval from a specific patch is marked, mark the other one as well (see Fig. 8).
3. Perform time refining.
4. Adjust the macro time-step structure—in case within one macro time-step there exist a fluid and a solid micro time-step that coincide, split the macro time-step into two macro time-steps at this point (see Fig. 9).

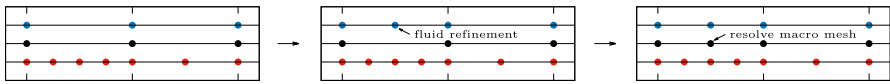


Fig. 9 An example of a splitting mechanism of macro time-steps. On the left, we show the mesh before refinement: middle (in black) the macro nodes, top (in blue) the fluid nodes and bottom (in red) the solid nodes with subcycling. In the center sketch, we refine the first macro interval once within the fluid domain. Since one node is shared between fluid and solid, we refine the macro mesh to resolve subcycling. This final configuration is shown on the right

5 Numerical results

5.1 Fluid subdomain functional

For the first example, we chose to test the derived error estimator on a goal functional concentrated in the fluid subproblem

$$J^f(\mathbf{U}) := \int_0^T v \|\nabla v^f\|_f^2 dt, \quad J^s(\mathbf{U}) := 0.$$

The functional is integrated only over the right half of the fluid subdomain, that is $\tilde{\Omega}^f = (2, 4) \times (0, 1)$. For this example, we also took the right hand side concentrated in the fluid subdomain, presented in Configuration 2.1. As the time interval, we choose $I = [0, 1]$. Then we have

$$(J^f)'_{\mathbf{U}}(\Psi^f) = \int_0^T 2v \left(\nabla v^f, \nabla \eta^f \right)_f dt.$$

Since the functional is nonlinear, we use a 2-point Gaussian quadrature for integration in time. In Table 1 we show the results of the a posteriori error estimator on a sequence of uniform time-meshes. Here, we considered the case without any micro time-stepping, that is the time-step sizes in both fluid and solid subdomains are uniformly equal. That gives a total number of time-steps in the fluid domain equal to $N^f = N$ and $N^s = N$ in the solid domain. Table 1 consists of partial residuals $\theta_k^f, \theta_k^s, \vartheta_k^f$ and ϑ_k^s , overall estimate σ_k , extrapolated errors $\tilde{J} - J(\mathbf{U}_k)$ and effectivities eff_k . The values of the goal functional on the three finest meshes were used for extrapolation in time. As a result, we got the reference value $\tilde{J} = 6.029469 \times 10^{-5}$. Except for the coarsest mesh, the estimator is very accurate and the effectivities are almost 1. On finer meshes, values of θ_k^f and ϑ_k^f are very close to each other which is due to the linearity of the coupled problem [2]. A similar phenomenon happens for θ_k^s and ϑ_k^s . The residuals are concentrated in the fluid subdomain, which suggests the usage of smaller time-step sizes in this space domain.

Table 2 collects the results for another sequence of uniform time-meshes. In this case, each of the macro time-steps in the fluid domain is split into two micro time-steps of the same size. That results in $N^f = 2N$ time-steps in the fluid domain and $N^s = N$ in the solid domain. The performance is still highly satisfactory. The residuals remain mostly concentrated in the fluid subdomain. Additionally, after comparing Tables 1

Table 1 Residuals and effectivities for fluid subdomain functional in case of uniform time-stepping in case $N^f = N^s = N$

N	θ_k^f	θ_k^s	ϑ_k^f	ϑ_k^s	σ_k	$\tilde{J} - J(\mathbf{U}_k)$	eff_k
50	3.62×10^{-8}	5.01×10^{-10}	1.05×10^{-7}	5.03×10^{-10}	1.42×10^{-7}	8.06×10^{-8}	1.76
100	9.66×10^{-9}	1.37×10^{-10}	9.96×10^{-9}	1.40×10^{-10}	1.99×10^{-8}	2.05×10^{-8}	0.97
200	2.48×10^{-9}	3.00×10^{-11}	2.52×10^{-9}	3.02×10^{-11}	5.07×10^{-9}	5.22×10^{-9}	0.97
400	6.28×10^{-10}	9.44×10^{-12}	6.33×10^{-10}	9.56×10^{-12}	1.28×10^{-9}	1.31×10^{-9}	0.98
800	1.58×10^{-10}	2.02×10^{-12}	1.58×10^{-10}	2.06×10^{-12}	3.20×10^{-10}	3.28×10^{-10}	0.98

Table 2 Residuals and effectivities for fluid subdomain functional in case of uniform time-stepping in case $N^f = 2N$ and $N^s = N$

N	θ_k^f	θ_k^s	ϑ_k^f	ϑ_k^s	σ_k	$\tilde{J} - J(\mathbf{U}_k)$	eff_k
50	9.66×10^{-9}	4.99×10^{-10}	9.96×10^{-9}	5.01×10^{-10}	2.06×10^{-8}	2.17×10^{-8}	0.95
100	2.48×10^{-9}	1.37×10^{-10}	2.52×10^{-9}	1.39×10^{-10}	5.28×10^{-9}	5.45×10^{-9}	0.97
200	6.28×10^{-10}	2.99×10^{-11}	6.33×10^{-10}	3.01×10^{-11}	1.32×10^{-9}	1.43×10^{-9}	0.92
400	1.58×10^{-10}	9.44×10^{-12}	1.58×10^{-10}	9.56×10^{-12}	3.35×10^{-10}	3.58×10^{-10}	0.94

Table 3 Residuals and effectivities for fluid subdomain functional in case of adaptive time-stepping

N	N^f	N^s	θ_k^f	θ_k^s	ϑ_k^f	ϑ_k^s	σ_k	$\tilde{J}-J(\mathbf{U}_k)$	eff_k
50	56	50	3.08×10^{-8}	5.01×10^{-10}	3.16×10^{-8}	5.04×10^{-10}	6.34×10^{-8}	6.64×10^{-8}	0.95
50	100	50	9.66×10^{-9}	4.99×10^{-10}	9.96×10^{-9}	5.01×10^{-10}	2.06×10^{-8}	2.17×10^{-8}	0.95
50	110	50	8.21×10^{-9}	4.99×10^{-10}	8.32×10^{-9}	5.02×10^{-10}	1.75×10^{-8}	1.84×10^{-8}	0.95
50	156	50	5.08×10^{-9}	4.99×10^{-10}	5.18×10^{-9}	4.97×10^{-10}	1.13×10^{-8}	1.20×10^{-8}	0.94

and 2, one can see that corresponding values of θ_k^f and ϑ_k^f are the same (value for $N = 800$ in Table 1 and $N = 400$ in Table 2, etc.). Overall, introducing micro time-stepping improves performance and reduces extrapolated error $\tilde{J} - J(\mathbf{U}_k)$ more efficiently.

In Table 3 we present findings in the case of adaptive time mesh refinement. We chose an initial configuration of uniform time-stepping without micro time-stepping for $N = 50$ and applied a sequence of adaptive refinements. On every level of refinement, the total number of time-steps is $N^f + N^s$. One can see that since the error is concentrated in the fluid domain, only time-steps corresponding to this space domain were refined. Again, effectivity gives very good results. The extrapolated error $\tilde{J} - J(\mathbf{U}_k)$ is even more efficiently reduced.

5.2 Solid subdomain functional

For the sake of symmetry, for the second example, we chose a functional concentrated on the solid subdomain

$$J^f(\mathbf{U}) = 0, \quad J^s(\mathbf{U}) = \int_0^T \lambda \|\nabla u^s\|_s^2 dt.$$

Also here the functional is integrated only over the right half of the solid subdomain $\tilde{\Omega}^s = (2, 4) \times (-1, 0)$. This time we set the right hand side according to Configuration 2.2. Again, $\bar{I} = [0, 1]$. The derivative reads as

$$(J^s)'_{\mathbf{U}}(\Psi^s) = \int_0^T 2\lambda (\nabla u^s, \nabla \xi^s)_s dt.$$

A third test case with functional evaluations in both domains gave comparable results without additional information, such that we refrain from adding it to this manuscript. Similarly, Table 4 gathers results for a sequence of uniform meshes without any micro time-stepping ($N^f = N^s = N$). The last three solutions are used for extrapolation in time which gives $\tilde{J} = 3.458826 \times 10^{-4}$. Also for this example, the effectivity is very satisfactory. On the finest discretization, the effectivity slightly declines. This might come from the limited accuracy of the reference value. Once more, on finer meshes, fluid residuals $\theta_k^f, \vartheta_k^f$ and solid residuals $\theta_k^s, \vartheta_k^s$ have similar values. This time, the residuals are concentrated in the solid subdomain and, in this case, the discrepancy is a bit bigger.

In Table 5 we display outcomes for a sequence of uniform meshes where each of the macro time-steps in the solid subdomain is split into two micro time-steps. That gives $N + 2N$ time-steps. Introducing micro time-stepping does not have a negative impact on the effectivity and significantly saves computational effort. Corresponding values of θ_k^s and ϑ_k^s in Tables 4 and 5 are almost the same. Residuals remain mostly concentrated in the solid subdomain.

Following the fluid example, in Table 6 we show calculation results in the case of adaptive time-mesh refinement. Here as well we took the uniform time-stepping without micro time-stepping for $N = 50$ as the initial configuration and the total

Table 4 Residuals and effectivities for solid subdomain functional in case of uniform time-stepping in case $N^f = N^s = N$

N	θ_k^f	θ_k^s	ϑ_k^f	ϑ_k^s	σ_k	$\tilde{J} - J(\mathbf{U}_k)$	eff $_k$
50	2.03×10^{-10}	2.66×10^{-6}	1.93×10^{-10}	1.03×10^{-5}	1.30×10^{-5}	2.49×10^{-5}	0.52
100	4.53×10^{-11}	2.59×10^{-6}	4.26×10^{-11}	2.67×10^{-6}	5.26×10^{-6}	4.77×10^{-6}	1.10
200	1.28×10^{-11}	5.18×10^{-7}	1.26×10^{-11}	5.21×10^{-7}	1.04×10^{-6}	9.80×10^{-7}	1.06
400	3.30×10^{-12}	1.17×10^{-7}	3.29×10^{-12}	1.17×10^{-7}	2.34×10^{-7}	2.23×10^{-7}	1.05
800	8.32×10^{-13}	2.82×10^{-8}	8.32×10^{-13}	2.80×10^{-8}	5.62×10^{-8}	5.07×10^{-8}	1.11

Table 5 Residuals and effectivities for solid subdomain functional in case of uniform time-stepping in case $N^f = N$ and $N^s = 2N$

N	θ_k^f	θ_k^s	ϑ_k^f	ϑ_k^s	σ_k	$\tilde{J} - J(\mathbf{U}_k)$	eff $_k$
50	4.13×10^{-10}	2.61×10^{-6}	1.91×10^{-9}	2.68×10^{-6}	5.29×10^{-6}	4.68×10^{-6}	1.13
100	8.69×10^{-11}	5.20×10^{-7}	-3.72×10^{-11}	5.23×10^{-7}	1.04×10^{-6}	9.54×10^{-7}	1.09
200	1.80×10^{-11}	1.17×10^{-7}	1.40×10^{-12}	1.17×10^{-7}	2.34×10^{-7}	2.16×10^{-7}	1.08
400	3.94×10^{-12}	2.82×10^{-8}	1.87×10^{-12}	2.80×10^{-8}	5.62×10^{-8}	4.90×10^{-8}	1.15

Table 6 Residuals and effectivities for solid subdomain functional in case of adaptive time-stepping

N	N^f	N^s	θ_k^f	θ_k^s	ϑ_k^f	ϑ_k^s	σ_k	$\tilde{J} - J(\mathbf{U}_k)$	eff_k
50	50	88	3.77×10^{-10}	6.57×10^{-6}	6.72×10^{-8}	6.91×10^{-8}	1.35×10^{-5}	1.06×10^{-5}	1.28
50	50	166	5.17×10^{-10}	1.35×10^{-6}	7.16×10^{-8}	1.38×10^{-8}	2.80×10^{-6}	2.52×10^{-6}	1.11
50	50	286	5.80×10^{-10}	4.54×10^{-7}	4.16×10^{-8}	4.56×10^{-7}	9.52×10^{-7}	7.34×10^{-7}	1.30
54	54	400	5.70×10^{-10}	1.19×10^{-7}	4.12×10^{-8}	1.19×10^{-7}	2.81×10^{-7}	1.10×10^{-7}	2.55

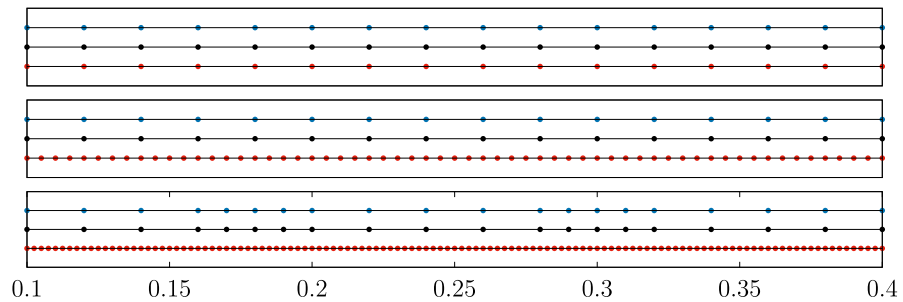


Fig. 10 Adaptive meshes for the solid functional. Top: uniform initial mesh; middle: 2 steps of adaptive refinement; bottom: 4 steps. Each plot shows the macro mesh (middle), the fluid mesh (top, in blue) and the solid mesh (bottom, in red)

number of time-steps is $N^f + N^s$. Except for the last entry, only the time-steps corresponding to the solid domain were refined. On the finest mesh, the effectivity deteriorates. However, adaptive time-stepping is still the most effective in reducing the extrapolated error $\tilde{J} - J(\mathbf{U}_k)$.

Finally, we show in Fig. 10 a sequence of adaptive meshes that result from this adaptive refinement strategy. In the top row, we show the initial mesh with 50 macro steps and no further splitting in fluid and solid. For a better presentation, we only show a small subset of the temporal interval $[0.1, 0.4]$. In the middle plot, we show the mesh after 2 steps of adaptive refinement and in the bottom line after 4 steps of adaptive refinement. Each plot shows the macro mesh, the fluid mesh (above) and the solid mesh (below). As expected, this example leads to a sub-cycling within the solid domain. For a finer approximation, the fluid problem also requires some local refinement. Whenever possible we avoid excessive subcycling by refining the macro mesh as described in Sect. 4.3.

6 Conclusion

In this paper, we have developed a multirate scheme and a temporal error estimate for a coupled problem that is inspired by fluid-structure interactions. The two subproblems, the heat equation and the wave equation feature different temporal dynamics. In this example, balanced approximation properties and stability demands ask for different step sizes.

We introduced a monolithic variational Galerkin formulation for the coupled problem and then used a partitioned framework for solving the algebraic systems. Having different time-step sizes for each of the subproblems couples multiple states in each time-step. That would require an enormous computational effort. To solve this, we discussed two different decoupling methods: first, a simple relaxation scheme that alternates between fluid and solid problem. In the second one, similar to the shooting method, we defined a root-finding problem on the interface and used matrix-free Newton-Krylov method for quickly approximating the zero. Both of the methods were able to successfully decouple our specific example and showed good robustness

concerning different subcycling of the multirate scheme in fluid- or solid-domain. However, the convergence of the shooting method was faster and it required fewer evaluations of the variational formulation.

As the next step, we introduced a goal-oriented error estimate based on the dual weighted residual method to estimate errors with regard to functional evaluations. The monolithic space-time Galerkin formulation allowed to split the residual errors into contributions from the fluid and solid problems. Finally, we established the localization of the error estimator. That let us derive an adaptive refinement scheme for choosing optimal distinct time-meshes for each problem. Several numerical results for two different goal functionals showed very good effectivity of the error estimate.

In future work, it remains to extend the methodology to nonlinear problems, in particular, to fully coupled fluid-structure interactions.

Acknowledgements Both authors acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—314838170, GRK 2297 MathCoRe. TR further acknowledge supported by the Federal Ministry of Education and Research of Germany (Project Number 05M16NMA).

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Avalos, G., Lasiecka, I., Triggiani, R.: Higher regularity of a coupled parabolic hyperbolic fluid-structure interactive system. *Georgian Math. J.* **15**, 402–437 (2008)
2. Becker, R., Rannacher, R.: *An Optimal Control Approach to A Posteriori Error Estimation in Finite Element Methods*, vol. 10, pp. 1–102. Cambridge University Press (2001)
3. Berger, M.: Stability of interfaces with mesh refinement. *Math. Comput.* **45**, 301–318 (1985). <https://doi.org/10.2307/2008126>
4. Blum, H., Lisky, S., Rannacher, R.: A domain splitting algorithm for parabolic problems. *Comput. Arch. Sci. Comput.* **49**(1), 11–23 (1992). <https://doi.org/10.1007/BF02238647>
5. Burman, E., Fernández, M.: An unfitted nitsche method for incompressible fluid-structureinteraction using overlapping meshes. *Comp. Meth. Appl. Mech. Eng.* **279**, 497–514 (2014)
6. Causin, P., Gereau, J., Nobile, F.: Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comp. Meth. Appl. Mech. Eng.* **194**, 4506–4527 (2005)
7. Collino, F., Fouquet, M., Joly, P.: A conservative space-time mesh refinement method for the 1-D wave equation. Part I: construction. *Numer. Math.* **95**, 197–221 (2003). <https://doi.org/10.1007/s00211-002-0446-5>
8. Collino, F., Fouquet, M., Joly, P.: A conservative space-time mesh refinement method for the 1-D wave equation. Part II: analysis. *Numer. Math.* **95**, 223–251 (2003). <https://doi.org/10.1007/s00211-002-0447-4>
9. Dawson, C., Du, Q., Dupont, T.: A finite difference domain decomposition algorithm for numerical solution of the heat equation. *Math. Comput.* **57**, 63–71 (1991). <https://doi.org/10.1090/S0025-5718-1991-1079011-4>

10. De Moerloose, L., Taelman, L., Segers, P., Vierendeels, J., Degroote, J.: Analysis of several subcycling schemes in partitioned simulations of a strongly coupled fluid-structure interaction. *Int. J. Num. Meth. Fluids* **89**(6), 181–195 (2018)
11. Degroote, J., Bathe, K.J., Vierendeels, J.: Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Comput. Struct.* **87**, 793–801 (2009). <https://doi.org/10.1016/j.compstruc.2008.11.013>
12. Donea, J.: An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid-structure interactions. *Comp. Meth. Appl. Mech. Eng.* **33**, 689–723 (1982)
13. Du, Q., Gunzburger, M., Hou, L., Lee, J.: Analysis of a linear fluid-structure interaction problem. *Discrete Continuous Dyn. Syst.* **9**(3), 633–650 (2003)
14. Du, Q., Gunzburger, M., Hou, L., Lee, J.: Semidiscrete finite element approximation of a linear fluid-structure interaction problem. *SIAM J. Numer. Anal.* **42**(1), 1–29 (2004)
15. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Introduction to adaptive methods for differential equations. *Acta Numer.* 105–158 (1995)
16. Failer, L., Meidner, D., Vexler, B.: Optimal control of a linear unsteady fluid-structure interaction problem. *J. Optim. Theory Appl.* **170**, 1–27 (2016)
17. Faille, I., Nataf, F., Willien, F., Wolf, S.: Two local time stepping schemes for parabolic problems. In: *Multiresolution and Adaptive Methods for Convection-Dominated Problems*, ESAIM Proc., vol. 29, pp. 58–72. EDP Sci., Les Ulis (2009). <https://doi.org/10.1051/proc/2009055>
18. Hron, J., Turek, S.: Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In: H.J. Bungartz, M. Schäfer (eds.) *Fluid-Structure Interaction: Modeling, Simulation, Optimization*, Lecture Notes in Computational Science and Engineering, pp. 371–385. Springer (2006)
19. Knoll, D., Keyes, D.: Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J. Comput. Phys.* **193**, 357–396 (2004)
20. Meidner, D., Richter, T.: Goal-oriented error estimation for the fractional step theta scheme. *Comput. Methods Appl. Math.* **14**, 203–230 (2014). <https://doi.org/10.1515/cmam-2014-0002>
21. Nitsche, J.: Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* **36**(1), 9–15 (1971). <https://doi.org/10.1007/BF02995904>
22. Piperno, S.: Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations. *Int. J. Num. Meth. Fluids* **25**, 1207–1226 (1997)
23. Piperno, S.: Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems. *ESAIM Math. Model. Numer. Anal.* **40**, 815–841 (2006). <https://doi.org/10.1051/m2an:2006035>
24. Richter, T.: *Fluid-structure Interactions. Models, Analysis and Finite Elements*, Lecture Notes in Computational Science and Engineering, vol. 118. Springer (2017)
25. Richter, T., Wick, T.: On time discretizations of fluid-structure interactions. In: T. Carraro, M. Geiger, S. Körkel, R. Rannacher (eds.) *Multiple Shooting and Time Domain Decomposition Methods*, Contributions in Mathematical and Computational Science, vol. 9, pp. 377–400. Springer (2015)
26. Schäfer, M., Turek, S.: Benchmark computations of laminar flow around a cylinder. (with support by F. Durst, E. Krause and R. Rannacher). In: E. Hirschel (ed.) *Flow Simulation with High-Performance Computers II*. DFG Priority Research Program Results 1993–1995, no. 52 in *Notes Numer. Fluid Mech.*, pp. 547–566. Vieweg, Wiesbaden (1996)
27. Schmitt, M., Vexler, B.: Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM J. Sci. Comput.* **30**, 369–393 (2008). <https://doi.org/10.1137/060670468>
28. Thomée, V.: *Galerkin Finite Element Methods for Parabolic Problems*, Computational Mathematics, vol. 25. Springer (1997)
29. Zhang, X., Zuazua, E.: Long-time behavior of a coupled heat-wave system arising in fluid-structure interaction. *Arch. Ration. Mech. Anal.* **184**, 49–120 (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.