Time-explicit Hybrid High-Order method for the nonlinear acoustic wave equation

Morgane Steins, Alexandre Ern, Olivier Jamond, Florence Drui

Time-explicit Hybrid High-Order method for the nonlinear acoustic wave equation

Morgane Steins * Alexandre Ern † Olivier Jamond ‡ Florence Drui §

July 28, 2023

Abstract

We devise a fully explicit scheme for the nonlinear acoustic wave equation in its second-order formulation in time, using the HHO method for space discretization and the leapfrog scheme for time integration. The key idea for the explicitation is an iterative procedure to approximate at each time step the static nonlinear coupling between the cell and face unknowns. This procedure is based on a splitting of the HHO stabilization operator and, in the linear case, is proved to converge for a large enough weight of the stabilization uniformly in the mesh size. Increasing the stabilization weight turns out to have a moderate impact on the CFL condition. The numerical experiments demonstrate the computational efficiency of the splitting procedure compared to a semi-implicit scheme for the static coupling between cell and face unknowns.

Mathematics Subjects Classification. 65M22, 65M60, 35L05, 74J30, 65F10.

Keywords. Hybrid high-order methods, wave equation, explicit time scheme, operator splitting.

1 Introduction

Let $\Omega \subset \mathbb{R}^d, d \in \{2,3\}$, denote an open bounded polyhedral domain with Lipschitz boundary $\partial\Omega$, and let $J = (0,\Theta)$ be a time interval with final time $\Theta > 0$. The problem under study is a generic nonlinear acoustic wave equation: Find the potential $u : \Omega \times J \to \mathbb{R}$ such that

$$\partial_t^2 u + \nabla \cdot (\mu(u,\nabla u)^2 \nabla u) = f, \quad \text{in } \Omega,\ \forall t \in J, \tag{1a}$$

$$u|_{t=0} = u_0, \quad \text{in } \Omega, \tag{1b}$$

$$\partial_t u|_{t=0} = v_0, \quad \text{in } \Omega, \tag{1c}$$

with $f : \Omega \times J \to \mathbb{R}$ the source term, $\mu : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}_+$ a nonlinear function representing the speed of sound, and u_0, v_0 the initial data for the potential, u, and the velocity, $\partial_t u$. Homogeneous Dirichlet boundary conditions are enforced on u, and we assume that the initial data satisfies these conditions. The nonlinear acoustic wave equation can be used in the modelling of various physical phenomena, such as seismic studies or sound and water waves. It is also a valuable tool to study numerical schemes before using them on more complex equations such as structural dynamics.

*Université Paris-Saclay, CEA, Service d'études mécaniques et thermiques, 91191, Gif-sur-Yvette, France & CERMICS, Ecole des Ponts, 77455 Marne-la-Vallée 2, France & INRIA, 2 rue Simone Iff, 75589 Paris, France. E-mail address : morgane.steins@cea.fr

†CERMICS, Ecole des Ponts, 77455 Marne-la-Vallée 2, France & INRIA, 2 rue Simone Iff, 75589 Paris, France. E-mail address : alexandre.ern@enpc.fr

‡Université Paris-Saclay, CEA, Service d'études mécaniques et thermiques, 91191, Gif-sur-Yvette, France. E-mail address : olivier.jamond@cea.fr

§Université Paris-Saclay, CEA, Service d'études mécaniques et thermiques, 91191, Gif-sur-Yvette, France. E-mail address : florence.drui@cea.fr

When tackling fast transient or nonlinear phenomena, an explicit time integration is favored. Indeed, an explicit scheme avoids at each time step the inversion of a matrix (other than block-diagonal) in linear problems, and the use of a fixed-point or a Newton algorithm in nonlinear problems. In both cases, these operations are costly in terms of computational time, especially when the simulation requires a large number of time steps. This work focuses on the second-order formulation in time of the nonlinear acoustic wave equation and considers the hybrid high-order (HHO) method for space discretization and the classical leapfrog (central finite difference) scheme for time integration. Other possible approaches for space discretization include conforming finite elements with mass lumping [12, 20, 21, 28] and interior penalty discontinuous Galerkin (DG) with a block-diagonal matrix [29]. A recent application of DG for the simulation of nonlinear sound waves is presented in [5].

The HHO method was introduced for linear diffusion in [25] and for linear elasticity in [24]. It was extended to various applications such as solid mechanics with the Biot problem [7], nonlinear elasticity [8], hyperelasticity [1] and elastoplasticity [2, 3]. The HHO method belongs to the class of hybridizable discontinuous Galerkin (HDG) methods [19], as shown in [17]. Moreover, the HHO method is closely related to nonconforming virtual element methods [6] and shares the same devising principles as weak Galerkin (WG) methods [41], as shown in [16, 26]. Two books were recently devoted to the HHO method [22, 15]. This method offers numerous advantages: support of polyhedral meshes, optimal convergence rates, local conservation principles, and computational efficiency. In particular, the support of polyhedral meshes allows for a natural use of mesh refinement with hanging nodes. The HHO unknowns are polynomials attached to the mesh cells and to the mesh faces. The setting is said to be of equal-order if both unknowns have the same polynomial degree, and of mixed-order if the degree of the cell unknowns is one order higher than that of the face unknowns.

The space discretization of the linear acoustic wave equation by the HHO method has been devised in [9], for both second- and first-order formulations in time. Optimal convergence rates in the space semi-discrete case are established in [11], and an unfitted version of the HHO method is devised in [10]. As shown in [9], combining the classical leapfrog scheme in time with the HHO method for the second-order formulation in time of the wave equation leads to a semi-implicit scheme, where, at each time step, the equation for the cell unknowns is explicit, but there is a static coupling between the face and the cell unknowns. Instead, for the first-order formulation in time, a fully explicit scheme can be devised by combining an explicit Runge–Kutta time-stepping scheme with the HHO method in the mixed-order setting. The same difficulty is encountered when using HDG and WG schemes for space discretization. For instance, for the first-order formulation, both implicit [37, 36] and explicit HDG schemes [40] were devised for the wave equation, with a better computational efficiency for the explicit version reported in [33]. A symplectic HDG formulation is derived in [39]. The second-order formulation in time with HDG discretization is conducted in [18] using a Störmer–Numerov scheme, and leads to a fully implicit scheme (both on the cell and the face unknowns). Similarly, in the WG context, a fully implicit scheme is developed in [30, 31]. The goal of the present work is to derive a fully explicit time discretization of the wave equation in its second-order formulation in time. As shown in the above discussion, this is a novel development for HHO methods. Moreover, leveraging on the close links between HHO, HDG and WG methods, the present work should also be beneficial when using HDG and WG methods. Our main result is an explicitation method built on a suitable splitting of the stabilization operator. In the linear case, we prove the convergence of this procedure provided that the stabilization is scaled by a large enough weight uniformly in the mesh size. This weight has only a moderate impact on the CFL condition. Moreover, numerical experiments illustrate that the splitting procedure is computationally more efficient compared to the semi-implicit scheme.

This work is organized as follows. Section 2 recalls from [9] the principles of the HHO space semi-discretization for the linear acoustic wave equation and extends the method to the present nonlinear setting. Section 3 presents the time discretization of the nonlinear acoustic wave equation with the explicit leapfrog scheme and discusses its algebraic formulation. Section 4 is dedicated to the splitting procedure in the nonlinear case, followed by a stability analysis in the linear case and some numerical tests on the splitting parameters. Finally, Section 5 studies the computational efficiency of the proposed method, first on the linear wave equation and then on two nonlinear model problems: a so-called *p-structure* operator and a vibrating membrane.

2 Space semi-discretization by the HHO method

In this section, the HHO semi-discretization in space of the acoustic wave equation is presented.

2.1 Discrete setting

Standard notation is used for Lebesgue, Sobolev and Bochner spaces. Let $(\cdot,\cdot)_\Omega$ denote the L^2-inner product on Ω and $\|\cdot\|$ the associated norm. Boldface notation is used for vectors and vector-valued fields, and calligraphic notation is used for matrices and matrix-valued fields. Assuming $f \in L^2(J;L^2(\Omega))$, we seek the solution of (1) in $U := H^2(J;L^2(\Omega)) \cap L^2(J;H^1_0(\Omega))$ such that

$$(\partial_t^2 u, w)_\Omega + \big(\mu(u,\nabla u)^2 \nabla u, \nabla w\big)_\Omega = (f,w)_\Omega\,, \quad \forall t \in J,\ \forall w \in H^1_0(\Omega). \tag{2}$$

Let $(\mathcal{T}_h)_{h>0}$ be a sequence of polyhedral meshes of Ω, such that each mesh $\mathcal{T}_h$ covers exactly Ω. For all $h > 0$, let T denote a generic mesh cell in $\mathcal{T}_h$, h_T its diameter and $\boldsymbol{n}_T$ its unit outward normal. We set $h := \max_{T\in\mathcal{T}_h} h_T$ for the mesh size. We say that the $(d-1)$-dimensional set F is a mesh face if there is a hyperplane H_F such that either $F = H_F \cap \partial T_- \cap T_+$ for two distinct mesh cells (and F is called mesh interface) or $F = H_F \cap \partial T_- \cap \partial\Omega$ (and F is called mesh boundary face). The collection of all the mesh faces is denoted $\mathcal{F}_h$. For all $T \in \mathcal{T}_h$, we denote by $\mathcal{F}_T$ the collection of its faces, and for all $F \in \mathcal{F}_T$, we set $\boldsymbol{n}_{TF} := \boldsymbol{n}_T|_F$. The mesh sequence is assumed to be shape-regular. Here, we only consider meshes composed of cells with simple shape (triangles, tetrahedra, quadrangles and hexahedra) so that the classical notion of regularity by Ciarlet is sufficient for our purposes; for mesh regularity with more general shapes, we refer the reader, e.g., to [23, 13, 22, 15]. Mesh regularity implies, in particular, that for all $h > 0$, all $T \in \mathcal{T}_h$ and all $F \in \mathcal{F}_T$, the diameter h_F of F is uniformly comparable to h_T.

Let the integer $k \geqslant 0$ be the polynomial order of the face unknowns and let $l \in \{k, k+1\}$ be the order of the cell unknowns. Recall that the setting is said to be of equal-order if $l = k$ and of mixed-order if $l = k+1$. Let $\mathbb{P}^l_d(T)$ (resp. $\mathbb{P}^k_{d-1}(F)$) denote the set of d-variate (resp. $(d-1)$-variate) polynomials of degree at most l (resp. k) restricted to the cell $T \in \mathcal{T}_h$ (resp. to the face $F \in \mathcal{F}_h$). The linear space composed of all the cell degrees of freedom is denoted $\mathcal{U}_\mathcal{T}^l$, and the linear space composed of all the face degrees of freedom is denoted $\mathcal{U}_\mathcal{F}^k$. These spaces are defined as Cartesian products in the form

$$\mathcal{U}_\mathcal{T}^l := \bigtimes_{T\in\mathcal{T}_h} \mathbb{P}^l_d(T), \quad \mathcal{U}_\mathcal{F}^k := \bigtimes_{F\in\mathcal{F}_h} \mathbb{P}^k_{d-1}(F), \tag{3}$$

and we slightly abuse the notation by viewing an element $w_\mathcal{T} = (w_T)_{T\in\mathcal{T}_h} \in \mathcal{U}_\mathcal{T}^l$ as a function defined a.e. over Ω such that $w_\mathcal{T}|_T := w_T$ for all $T \in \mathcal{T}_h$. The collection of all the cell and face degrees of freedom is the hybrid space

$$\widehat{\mathcal{U}}_h^{l,k} := \mathcal{U}_\mathcal{T}^l \times \mathcal{U}_\mathcal{F}^k. \tag{4}$$

A generic element of $\widehat{\mathcal{U}}_h^{l,k}$ is denoted $\hat{w}_h := (w_\mathcal{T}, w_\mathcal{F}) \in \mathcal{U}_\mathcal{T}^l \times \mathcal{U}_\mathcal{F}^k$ and, in what follows, variables with hats refer to hybrid variables. For a given cell $T \in \mathcal{T}_h$, one also defines a local hybrid space of degrees of freedom

$$\widehat{\mathcal{U}}_T^{l,k} := \mathbb{P}^l_d(T) \times \mathcal{U}_{\partial T}^k, \quad \mathcal{U}_{\partial T}^k := \bigtimes_{F\in\mathcal{F}_T} \mathbb{P}^k_{d-1}(F). \tag{5}$$

Then $\hat{w}_T := (w_T, w_{\partial T} = (w_F)_{F\in\mathcal{F}_T}) \in \widehat{\mathcal{U}}_T^{l,k}$ denotes a generic local hybrid unknown in T, composed of one cell unknown and the collection of the face unknowns for all the faces in $\mathcal{F}_T$. As above, we slightly abuse the notation by viewing an element $w_{\partial T} = (w_F)_{F\in\mathcal{F}_T} \in \mathcal{U}_{\partial T}^k$ as a function defined a.e. over ∂T such that $w_{\partial T}|_F := w_F$ for all $F \in \mathcal{F}_T$. Let $\mathcal{U}_{\mathcal{F},0}^k := \{v_\mathcal{F} \in \mathcal{U}_\mathcal{F}^k,\ \text{s.t. } v_F = 0,\ \forall F \subset \partial\Omega\}$ be the ubspace of face unknowns respecting the homogeneous Dirichlet conditions. The subspace of hybrid unknowns respecting the homogeneous Dirichlet conditions is denoted

$$\widehat{\mathcal{U}}_{h,0}^{l,k} := \mathcal{U}_\mathcal{T}^l \times \mathcal{U}_{\mathcal{F},0}^k. \tag{6}$$

L^2-orthogonal projections onto polynomial spaces are denoted with the symbol Π. For instance, for all $T \in \mathcal{T}_h$, Π_T^l is the projection onto $\mathbb{P}_d^l(T)$ and for all $F \in \mathcal{F}_h$, Π_F^k the projection onto $\mathbb{P}_{d-1}^k(F)$. The L^2-orthogonal projection onto the broken polynomial spaces $\mathcal{U}_\mathcal{T}^l$ and $\mathcal{U}_\mathcal{F}^k$ is denoted by $\Pi_\mathcal{T}^l$ and $\Pi_\mathcal{F}^k$ respectively. Let $(\cdot,\cdot)_T$ and $(\cdot,\cdot)_F$ respectively denote the L^2-inner product in the cell $T \in \mathcal{T}_h$ and a face $F \in \mathcal{F}_h$. For all $v_{\partial T}, w_{\partial T} \in \mathcal{U}_{\partial T}^k$, we also define $(v_{\partial T}, w_{\partial T})_{\partial T} := \sum_{F\in\mathcal{F}_T}(v_F, w_F)_F$.

2.2 Discrete HHO operators

The HHO discretization relies on two key operators: a gradient reconstruction operator and a stabilization operator. Both operators are local, i.e. they are defined independently in every cell $T \in \mathcal{T}_h$. The gradient reconstruction operator builds a gradient in the cell T from the local cell and face unknowns. This operator $\boldsymbol{G}_T^k : \widehat{\mathcal{U}}_T^{l,k} \to \mathbb{P}_d^k(T;\mathbb{R}^d)$ is evaluated by solving the following problem: For all $\hat{v}_T \in \widehat{\mathcal{U}}_T^{l,k}$,

$$(\boldsymbol{G}_T^k(\hat{v}_T), \boldsymbol{q})_T = (\nabla v_T, \boldsymbol{q})_T + (v_{\partial T} - v_T, \boldsymbol{q}\cdot\boldsymbol{n}_T)_{\partial T}\,, \quad \forall \boldsymbol{q} \in \mathbb{P}_d^k(T;\mathbb{R}^d), \tag{7}$$

where $\mathbb{P}_d^k(T;\mathbb{R}^d)$ denotes the set of $\mathbb{R}^d$-valued d-variate polynomials of degree k in the cell T. In practice, each component of the reconstructed gradient is found by inverting the mass matrix associated with a chosen basis of $\mathbb{P}_d^k(T)$. One can also build a potential reconstruction operator $R_T^{k+1} : \widehat{\mathcal{U}}_T^{l,k} \to \mathbb{P}_d^{k+1}(T)$ solving, for all $\hat{w}_T \in \widehat{\mathcal{U}}_T^{l,k}$, the following Neumann problem:

$$(\nabla R_T^{k+1}(\hat{w}_T), \nabla q)_T = (\nabla w_T, \nabla q)_T + (w_{\partial T} - w_T, \nabla q\cdot\boldsymbol{n}_T)_{\partial T}\,, \quad \forall q \in \mathbb{P}_d^{k+1}(T), \tag{8}$$

and the mean-value condition $(R_T^{k+1}(\hat{v}_T), 1)_T = (v_T, 1)_T$. In this case, the computation of $R_T^{k+1}(\hat{v}_T)$ requires inverting the stiffness matrix for a chosen basis of $\mathbb{P}_d^{k+1,0}(T) := \{q \in \mathbb{P}_d^{k+1}(T),\ \text{s.t. } (q,1)_T = 0\}$.

The role of the stabilization is to weakly enforce the matching between cell and face unknowns on each mesh face. Let $T \in \mathcal{T}_h$. For all $\hat{w}_T \in \widehat{\mathcal{U}}_T^{l,k}$, set $\delta_T(\hat{w}_T) := w_{\partial T} - w_T|_{\partial T}$ on ∂T and $\delta_{TF}(\hat{w}_T) := \delta_T(\hat{w}_T)|_F$. In the mixed-order case, the local stabilization operator S_{TF} is defined as

$$S_{TF}(\hat{w}_T) := \Pi_F^k(\delta_{TF}(\hat{w}_T)), \quad \forall \hat{w}_T \in \widehat{\mathcal{U}}_T^{l,k}, \tag{9}$$

and leads to the so-called Lehrenfeld–Schöberl stabilization in the HDG setting (see, e.g. [34, 35]). In the equal-order case, the definition of S_{TF} requires the computation of R_T^{k+1} and writes

$$S_{TF}(\hat{w}_T) := \Pi_F^k\big(\delta_{TF}(\hat{w}_T) + (I - \Pi_T^k)R_T^{k+1}(0, \delta_T(\hat{w}_T))|_F\big), \quad \forall \hat{w}_T \in \hat{\mathcal{U}}_T^{k,k} =: \hat{\mathcal{U}}_T^k. \tag{10}$$

In both equal- and mixed-order settings, we define, for all $\hat{y}_T, \hat{v}_T, \hat{w}_T \in \widehat{\mathcal{U}}_T^{l,k}$, the local stabilization form as

$$\sigma_T(\hat{y}_T; \hat{v}_T, \hat{w}_T) = \gamma\bar{\mu}_T(\hat{y}_T)^2 \sum_{F\in\mathcal{F}_T} \eta_{TF}^{-1}\left(S_{TF}(\hat{v}_T), S_{TF}(\hat{w}_T)\right)_F, \tag{11}$$

with $\bar{\mu}_T(\hat{y}_T)$ an approximation of the local speed of sound in T evaluated using $\hat{y}_T$, the scaling factor η_{TF} equal to h_F or h_T, and $\gamma > 0$ a scaling parameter. Choosing $\eta_{TF} = h_T$ is often more relevant. Finally, in each mesh cell $T \in \mathcal{T}_h$, one defines the local stiffness form b_T, such that, for all $\hat{y}_T, \hat{v}_T, \hat{w}_T \in \widehat{\mathcal{U}}_T^{l,k}$,

$$b_T(\hat{y}_T; \hat{v}_T, \hat{w}_T) := (\mu(y_T, \boldsymbol{G}_T^k(\hat{y}_T))^2 \boldsymbol{G}_T^k(\hat{v}_T), \boldsymbol{G}_T^k(\hat{w}_T))_T. \tag{12}$$

Both forms b_T and σ_T are nonlinear w.r.t. $\hat{y}_T$ and linear w.r.t. $\hat{v}_T$ and $\hat{w}_T$.

Remark 2.1 (Choice of γ)**.** Any positive value can be chosen for the scaling parameter γ. For the plain HHO method, the choice $\gamma = 1$ is made in [24, 25]. In the present setting, this parameter is introduced to tune the scale of the stabilization compared to the stiffness term. If $\gamma \ll 1$, the stabilization is much smaller than the stiffness, and the problem is close to being singular. The case $\gamma \gg 1$ does not entail singularity issues but may lead to larger errors than $\gamma = 1$. Taking larger values of γ also lowers the CFL stability restriction on the time step as we shall see below. Very large values of γ are thus not computationally efficient due to the large number of time steps. We refer the reader to Section 5.1.2 and Appendix A for further insight.

2.3 HHO semi-discretization of the wave equation

The global forms b_h and σ_h are defined, for all $\hat{y}_h, \hat{v}_h, \hat{w}_h \in \widehat{\mathcal{U}}_{h,0}^{l,k}$, as

$$b_h(\hat{y}_h; \hat{v}_h, \hat{w}_h) := \sum_{T\in\mathcal{T}_h} b_T(\hat{y}_T; \hat{v}_T, \hat{w}_T), \quad \sigma_h(\hat{y}_h; \hat{v}_h, \hat{w}_h) := \sum_{T\in\mathcal{T}_h} \sigma_T(\hat{y}_T; \hat{v}_T, \hat{w}_T). \tag{13}$$

The space semi-discrete scheme for the nonlinear wave equation (1) consists of finding $\hat{u}_h := (u_\mathcal{T}, u_\mathcal{F}) \in C^2(\bar{J}; \widehat{\mathcal{U}}_{h,0}^{l,k})$ such that, for all $t \in \bar{J}$ and all $\hat{w}_h := (w_\mathcal{T}, w_\mathcal{F}) \in \widehat{\mathcal{U}}_{h,0}^{l,k}$,

$$(\partial_t^2 u_\mathcal{T}, w_\mathcal{T})_\Omega + b_h(\hat{u}_h(t); \hat{u}_h(t), \hat{w}_h) + \sigma_h(\hat{u}_h(t); \hat{u}_h(t), \hat{w}_h) = (f(t), w_\mathcal{T})_\Omega\,. \tag{14}$$

Notice that the homogeneous Dirichlet boundary condition is enforced by the condition $\hat{u}_h(t) \in \widehat{\mathcal{U}}_{h,0}^{l,k}$ at all times $t \in \bar{J}$. Moreover, the initial conditions translate into conditions on $u_\mathcal{T}$ only:

$$u_\mathcal{T}(0) := \Pi_\mathcal{T}^l(u_0), \quad \partial_t u_\mathcal{T}(0) := \Pi_\mathcal{T}^l(v_0). \tag{15}$$

Notice that initial conditions on the face degrees of freedom are not needed.

Remark 2.2 (Linear case)**.** In this case, the speed of sound μ is assumed to be piecewise constant on a polyhedral partition of Ω, and is assumed to take a constant value denoted by μ_T in each cell T. The dependence of the local forms on the variable $\hat{y}_T$ disappears, so that these forms become bilinear:

$$b_T(\hat{v}_T, \hat{w}_T) = \mu_T(\boldsymbol{G}_T^k(\hat{v}_T), \boldsymbol{G}_T^k(\hat{w}_T))_T, \quad \sigma_T(\hat{v}_T, \hat{w}_T) = \gamma\mu_T^2 \sum_{F\in\mathcal{F}_T} \eta_{TF}^{-1} \left(S_{TF}(\hat{v}_T), S_{TF}(\hat{w}_T)\right)_F, \tag{16}$$

and the global bilinear forms read $b_h(\hat{v}_h, \hat{w}_h) := \sum_{T\in\mathcal{T}_h} b_T(\hat{v}_T, \hat{w}_T)$ and $\sigma_h(\hat{v}_h, \hat{w}_h) := \sum_{T\in\mathcal{T}_h} \sigma_T(\hat{v}_T, \hat{w}_T)$.

3 Time discretization and algebraic formulation

In this section, a time discretization of the space semi-discrete wave equation is presented using the leapfrog scheme. The algebraic formulation is also presented to highlight the implicit coupling between cell and face unknowns.

3.1 Time discretization

Let N be the number of discrete time intervals such that $(t^n)_{0\leqslant n\leqslant N}$ are the discrete time nodes with $t^0 = 0$ and $t^N := \Theta$. For the sake of simplicity, we consider a fixed time step $\Delta t := \frac{\Theta}{N}$. The time discrete unknown $\hat{u}_h^n = (u_\mathcal{T}^n, u_\mathcal{F}^n) \in \widehat{\mathcal{U}}_{h,0}^{l,k}$ is meant to be an approximation of $\hat{u}_h(t^n)$. We also define the projection of a function in $H^1(\Omega)$ onto the space of hybrid degrees of freedom $\widehat{\mathcal{U}}_h^{l,k}$ as

$$\hat{I}_h(v) := ((\Pi_T^l(v))_{T\in\mathcal{T}_h}, (\Pi_F^k(v))_{F\in\mathcal{F}_h}) \in \widehat{\mathcal{U}}_h^{l,k}, \quad \forall v \in H^1(\Omega). \tag{17}$$

Notice that $v \in H_0^1(\Omega)$ implies $\hat{I}_h(v) \in \widehat{\mathcal{U}}_{h,0}^{l,k}$.

A classical explicit scheme for the time integration of the wave equation is the leapfrog (central finite difference) scheme. In the context of the HHO space semi-discretization, this scheme consists of solving, for all $n \geqslant 1$ and all $\hat{w}_h \in \widehat{\mathcal{U}}_{h,0}^{l,k}$,

$$\frac{1}{\Delta t^2}(u_\mathcal{T}^{n+1} - 2u_\mathcal{T}^n + u_\mathcal{T}^{n-1}, w_\mathcal{T})_\Omega + b_h(\hat{u}_h^n; \hat{u}_h^n, \hat{w}_h) + \sigma_h(\hat{u}_h^{n-1}; \hat{u}_h^n, \hat{w}_h) = (f(t^n), w_\mathcal{T})_\Omega\,, \tag{18}$$

with $\hat{u}_h^n$, $\hat{u}_h^{n-1}$ known from prior time steps or given by the initial conditions as follows:

$$u_\mathcal{T}^0 = \Pi_\mathcal{T}^l(u_0) \tag{19a}$$

$$b_h(\hat{u}_h^0;\hat{u}_h^0,(0,w_{\mathcal{F}}))+\sigma_h(\hat{I}_h(u_0);\hat{u}_h^0,(0,w_{\mathcal{F}}))=0, \tag{19b}$$

$$\left(u_{\mathcal{T}}^1,w_{\mathcal{T}}\right)_\Omega=\left(u_{\mathcal{T}}^0+\Delta t\Pi_{\mathcal{T}}^l(v_0),w_{\mathcal{T}}\right)_\Omega+\frac{\Delta t^2}{2}[(f(0),w_{\mathcal{T}})_\Omega-b_h(\hat{u}_h^0;\hat{u}_h^0,(w_{\mathcal{T}},0))-\sigma_h(\hat{u}_h^0;\hat{u}_h^0,(w_{\mathcal{T}},0))], \tag{19c}$$

where (19b) holds for all $w_{\mathcal{F}}\in\mathcal{U}_{\mathcal{F},0}^k$ and (19c) holds for all $w_{\mathcal{T}}\in\mathcal{U}_{\mathcal{T}}^l$. Notice that, in equation (18), the stabilization term is linear w.r.t $\hat{u}_h^n$ and that in the second equation (19b), the stabilization is linear with respect to $\hat{u}_h^0$. This linearization does not influence the accuracy because the speed of sound in the stabilization only aims at equilibrating the magnitude of the stabilization and stiffness terms. More precisely, the approximation of the local speed of sound in a cell $T\in\mathcal{T}_h$ is evaluated as

$$\bar{\mu}_T(u_T^{n-1}):=\mu(u_T^{n-1}(\boldsymbol{x}_T),\boldsymbol{G}_T^k(\hat{u}_T^{n-1})(\boldsymbol{x}_T)), \tag{20}$$

where $\boldsymbol{x}_T$ is the barycenter of T. If the speed of sound does not vary too much across the mesh and during the simulation, a constant value μ_0 can be taken in all the cells and at all the time steps. For the stiffness operator b_h, the above linearization is not considered in order to preserve the second-order convergence rate in time of the discrete scheme.

Remark 3.1 (Linear case)**.** In this case, σ_h does not depend on $\hat{u}_h^{n-1}$, and the fully discrete equation reads for all $n\geqslant 1$ and all $\hat{w}_h\in\widehat{\mathcal{U}}_{h,0}^{l,k}$,

$$\frac{1}{\Delta t^2}(u_{\mathcal{T}}^{n+1}-2u_{\mathcal{T}}^n+u_{\mathcal{T}}^{n-1},w_{\mathcal{T}})_\Omega+b_h(\hat{u}_h^n,\hat{w}_h)+\sigma_h(\hat{u}_h^n,\hat{w}_h)=(f(t^n),w_{\mathcal{T}})_\Omega\,. \tag{21}$$

3.2 Algebraic formulation

Let $N_{\mathcal{T}}:=\dim(\mathcal{U}_{\mathcal{T}}^l)$ and $N_{\mathcal{F}}:=\dim(\mathcal{U}_{\mathcal{F},0}^k)$ and $\{\phi_i\}_{1\leqslant i\leqslant N_{\mathcal{T}}},\{\psi_i\}_{1\leqslant i\leqslant N_{\mathcal{F}}}$ be bases of $\mathcal{U}_{\mathcal{T}}^l$ and $\mathcal{U}_{\mathcal{F},0}^k$ respectively. Let $(\mathsf{U}_{\mathcal{T}}^n,\mathsf{U}_{\mathcal{F}}^n)\in\mathbb{R}^{N_{\mathcal{T}}}\times\mathbb{R}^{N_{\mathcal{F}}}$ be the vector of time-dependent degrees of freedom of the solution $\hat{u}_h^n$ on these bases, and $\mathsf{F}_{\mathcal{T}}^n\in\mathbb{R}^{N_{\mathcal{T}}}$ the vector having components $((f(t^n),\phi_i)_\Omega)_{1\leqslant i\leqslant N_{\mathcal{T}}}$. To alleviate the notation, we use the subscript $_{\mathcal{F}}$ rather than $_{\mathcal{F},0}$ for objects related to $\mathcal{U}_{\mathcal{F},0}^k$. The fully discrete problem (18) can be written in algebraic form by considering the vector-valued nonlinear stiffness operator $\mathsf{B}(\mathsf{U}_{\mathcal{T}}^n,\mathsf{U}_{\mathcal{F}}^n)$ associated with the linear form $b_h(\hat{u}_h^n;\hat{u}_h^n,\cdot)$, with $\mathsf{B}_{\mathcal{T}},\mathsf{B}_{\mathcal{F}}$ respectively collecting the cell and face components. The linearized stabilization bilinear form $\sigma_h(\hat{u}_h^{n-1};\cdot,\cdot)$ leads to a symmetric matrix depending on $\hat{u}_h^{n-1}$ and denoted by $\mathcal{S}^{n-1}$. Altogether, the algebraic formulation reads

$$\frac{1}{\Delta t^2}\begin{bmatrix}\mathcal{M}&0\\0&0\end{bmatrix}\begin{pmatrix}\mathsf{U}_{\mathcal{T}}^{n+1}\\\cdot\end{pmatrix}+\begin{pmatrix}\mathsf{B}_{\mathcal{T}}(\mathsf{U}_{\mathcal{T}}^n,\mathsf{U}_{\mathcal{F}}^n)\\\mathsf{B}_{\mathcal{F}}(\mathsf{U}_{\mathcal{T}}^n,\mathsf{U}_{\mathcal{F}}^n)\end{pmatrix}+\begin{bmatrix}\mathcal{S}_{\mathcal{T}\mathcal{T}}^{n-1}&\mathcal{S}_{\mathcal{T}\mathcal{F}}^{n-1}\\\mathcal{S}_{\mathcal{F}\mathcal{T}}^{n-1}&\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1}\end{bmatrix}\begin{pmatrix}\mathsf{U}_{\mathcal{T}}^n\\\mathsf{U}_{\mathcal{F}}^n\end{pmatrix}=\begin{bmatrix}\mathsf{F}_{\mathcal{T}}^n-\frac{1}{\Delta t^2}\mathcal{M}(\mathsf{U}_{\mathcal{T}}^{n-1}-2\mathsf{U}_{\mathcal{T}}^n)\\0\end{bmatrix}, \tag{22}$$

with $\mathcal{M}$ the cell mass matrix. As a consequence of the structure of the global mass matrix, the face component is replaced by a "$\cdot$" in the acceleration term. The submatrix $\mathcal{S}_{\mathcal{T}\mathcal{T}}^{n-1}$ is block-diagonal, since $\sigma_h(\hat{u}_h^{n-1};\cdot,\cdot)$ does not couple cell degrees of freedom from different cells. Is is useful to notice that in the *mixed-order* case, $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1}$ is also block-diagonal, since it reduces on each face to the mass matrix associated with a local polynomial basis. Finally, the cell mass matrix $\mathcal{M}$ is also block-diagonal.

The resolution of the wave equation discretized with HHO and the leapfrog scheme proceeds in two steps for all $n\geqslant 1$:

1. Compute $\mathsf{U}_{\mathcal{F}}^n$, i.e., solve $\mathsf{B}_{\mathcal{F}}(\mathsf{U}_{\mathcal{T}}^n,\mathsf{U}_{\mathcal{F}}^n)+\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1}\mathsf{U}_{\mathcal{F}}^n=-\mathcal{S}_{\mathcal{F}\mathcal{T}}^{n-1}\mathsf{U}_{\mathcal{T}}^n$ using the second row in (22), where $\mathsf{U}_{\mathcal{T}}^n$ is the data (known from the previous time step or the initial condition) and $\mathsf{U}_{\mathcal{F}}^n$ the unknown;
2. Compute $\mathsf{U}_{\mathcal{T}}^{n+1}$, i.e., solve $\frac{1}{\Delta t^2}\mathcal{M}\mathsf{U}_{\mathcal{T}}^{n+1}=\mathsf{F}_{\mathcal{T}}^n-\frac{1}{\Delta t^2}\mathcal{M}(\mathsf{U}_{\mathcal{T}}^{n-1}-2\mathsf{U}_{\mathcal{T}}^n)-\mathsf{B}_{\mathcal{T}}(\mathsf{U}_{\mathcal{T}}^n,\mathsf{U}_{\mathcal{F}}^n)-\mathcal{S}_{\mathcal{T}\mathcal{T}}^{n-1}\mathsf{U}_{\mathcal{T}}^n-\mathcal{S}_{\mathcal{T}\mathcal{F}}^{n-1}\mathsf{U}_{\mathcal{F}}^n$ using the first row in (22).

The key observation is that (22) is a semi-implicit scheme, and not an explicit scheme. Indeed, the first step above induces a static nonlinear coupling between cell and face unknowns. Solving this nonlinear equation at each time step of the leapfrog scheme is not effective because of the heavy computational cost. This problem is solved in the next section, leading to a fully explicit scheme.

Remark 3.2 (Linear case)**.** In this case, the stabilization matrix does not depend on the previous time step and is simply denoted by $\mathcal{S}$. The stiffness form is bilinear and the operator B becomes linear so that it can be replaced by a matrix product between a stiffness matrix $\mathcal{B}$ and the vector of hybrid unknowns $(\mathsf{U}_{\mathcal{T}}^n, \mathsf{U}_{\mathcal{F}}^n)$. In order to have a more compact notation, we define the complete HHO stiffness matrix as

$$\mathcal{A} := \mathcal{B} + \mathcal{S}. \tag{23}$$

The cell-cell block matrix $\mathcal{A}_{\mathcal{T}\mathcal{T}}$ is block-diagonal, but the face-face block matrix $\mathcal{A}_{\mathcal{F}\mathcal{F}}$ is not block-diagonal since the gradient reconstruction operator couples the degrees of freedom from all the faces of a given cell. Equation (22) translates into the semi-implicit scheme

$$\frac{1}{\Delta t^2}\begin{bmatrix} \mathcal{M} & 0 \\ 0 & 0 \end{bmatrix}\begin{pmatrix} \mathsf{U}_{\mathcal{T}}^{n+1} \\ \cdot \end{pmatrix} + \begin{bmatrix} \mathcal{A}_{\mathcal{T}\mathcal{T}} & \mathcal{A}_{\mathcal{T}\mathcal{F}} \\ \mathcal{A}_{\mathcal{F}\mathcal{T}} & \mathcal{A}_{\mathcal{F}\mathcal{F}} \end{bmatrix}\begin{pmatrix} \mathsf{U}_{\mathcal{T}}^n \\ \mathsf{U}_{\mathcal{F}}^n \end{pmatrix} = \begin{bmatrix} \mathsf{F}_{\mathcal{T}}^n - \frac{1}{\Delta t^2}\mathcal{M}(\mathsf{U}_{\mathcal{T}}^{n-1} - 2\mathsf{U}_{\mathcal{T}}^n) \\ 0 \end{bmatrix}. \tag{24}$$

The first step becomes the resolution of the linear system $\mathcal{A}_{\mathcal{F}\mathcal{F}}\mathsf{U}_{\mathcal{F}}^n = -\mathcal{A}_{\mathcal{F}\mathcal{T}}\mathsf{U}_{\mathcal{T}}^n$, where, as above, $\mathsf{U}_{\mathcal{T}}^n$ is the data and $\mathsf{U}_{\mathcal{F}}^n$ the unknown. This requires the inversion of the sparse matrix $\mathcal{A}_{\mathcal{F}\mathcal{F}}$, either by a direct inversion or by an iterative process.

4 Splitting

In this section, a splitting procedure on the static coupling equation on the faces is devised. The procedure is presented for the nonlinear acoustic wave equation and generic sufficient conditions for convergence are given. A sharper convergence analysis is presented in the linear case. The procedure differs depending on the setting for the polynomial order in the HHO method (mixed- or equal-order).

4.1 Mixed-order setting

We consider first the mixed-order setting since the definition of the stabilization is simpler. The iterative splitting procedure proceeds as follows. For all $n \geqslant 1$, set $u_{\mathcal{F}}^{n,0} = u_{\mathcal{F}}^{n-1}$ and iterate on $m \geqslant 0$ by finding $u_{\mathcal{F}}^{n,m+1} \in \mathcal{U}_{\mathcal{F},0}^k$ such that

$$\sigma_h(\hat{u}_h^{n-1}; (0, u_{\mathcal{F}}^{n,m+1}), (0, w_{\mathcal{F}})) = -b_h((u_{\mathcal{T}}^n, u_{\mathcal{F}}^{n,m}); (u_{\mathcal{T}}^n, u_{\mathcal{F}}^{n,m}), (0, w_{\mathcal{F}})) - \sigma_h(\hat{u}_h^{n-1}; (u_{\mathcal{T}}^n, 0), (0, w_{\mathcal{F}})), \tag{25}$$

for all $w_{\mathcal{F}} \in \mathcal{U}_{\mathcal{F},0}^k$. The algebraic form is as follows: Setting $\mathsf{U}_{\mathcal{F}}^{n,0} := \mathsf{U}_{\mathcal{F}}^{n-1}$, one seeks $\mathsf{U}_{\mathcal{F}}^{n,m+1} \in \mathbb{R}^{N_{\mathcal{F}}}$ such that

$$\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1}\mathsf{U}_{\mathcal{F}}^{n,m+1} = -\mathsf{B}_{\mathcal{F}}(\mathsf{U}_{\mathcal{T}}^n, \mathsf{U}_{\mathcal{F}}^{n,m}) - \mathcal{S}_{\mathcal{F}\mathcal{T}}^{n-1}\mathsf{U}_{\mathcal{T}}^n. \tag{26}$$

This splitting procedure is computationally effective since, as mentioned above, the face-face stabilization submatrix $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1}$ is block-diagonal, the size of each block being $\binom{k+d-1}{d-1}$, i.e., $(k+1)$ for $d = 2$ and $\frac{1}{2}(k+1)(k+2)$ for $d = 3$. The procedure is a fixed-point algorithm, so that its convergence is ensured for $\delta < 1$, where δ is the Lipschitz constant of the vector-valued function $(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1})^{-1}\mathsf{B}_{\mathcal{F}}(\mathsf{U}_{\mathcal{T}}^n, \cdot)$, i.e.,

$$\|(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1})^{-1}\big(\mathsf{B}_{\mathcal{F}}(\mathsf{U}_{\mathcal{T}}^n, X) - \mathsf{B}_{\mathcal{F}}(\mathsf{U}_{\mathcal{T}}^n, Y)\big)\| \leqslant \delta\|X - Y\| \quad \forall X, Y \in \mathbb{R}^{N_{\mathcal{F}}}. \tag{27}$$

Remark 4.1 (Linear case)**.** In this case, the iterative scheme writes, for all $m \geqslant 0$,

$$\sigma_h((0, u_{\mathcal{F}}^{n,m+1}), (0, w_{\mathcal{F}})) = -b_h((u_{\mathcal{T}}^n, u_{\mathcal{F}}^{n,m}), (0, w_{\mathcal{F}})) - \sigma_h((u_{\mathcal{T}}^n, 0), (0, w_{\mathcal{F}})), \tag{28}$$

for all $w_{\mathcal{F}} \in \mathcal{U}_{\mathcal{F},0}^k$, and its algebraic form becomes $\mathcal{S}_{\mathcal{F}\mathcal{F}}\mathsf{U}_{\mathcal{F}}^{n,m+1} = -\mathcal{B}_{\mathcal{F}\mathcal{F}}\mathsf{U}_{\mathcal{F}}^{n,m} - \mathcal{A}_{\mathcal{F}\mathcal{T}}\mathsf{U}_{\mathcal{T}}^n$. The convergence criterion (27) on the Lipschitz constant translates into the algebraic condition

$$\rho(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{-1}\mathcal{B}_{\mathcal{F}\mathcal{F}}) < 1, \tag{29}$$

where $\rho(\mathcal{Q})$ denotes the spectral radius of the matrix $\mathcal{Q}$. This condition does not depend on the time index n since, in the linear case, the stiffness and stabilization matrices do not depend on time.

4.2 Equal-order setting

The equal-order setting does not offer the same simplicity since the stabilization couples together the degrees of freedom of all the faces of a cell. One can, however, draw on the mixed-order setting and split the stabilization form into the mixed-order stabilization form, which leads to a block-diagonal matrix, and the remainder. Specifically, let ζ_T be the local form such that for all $T \in \mathcal{T}_h$, and all $\hat{y}_T \in \widehat{\mathcal{U}}_T^{l,k}$, $v_{\partial T}, w_{\partial T} \in \mathcal{U}_{\partial T}^k$,

$$\begin{aligned}\zeta_T(\hat{y}_T;(0,v_{\partial T}),(0,w_{\partial T})) = \gamma\mu_T(\hat{y}_T)^2 \sum_{F\in\mathcal{F}_T} \eta_{TF}^{-1}\Big\{ &\big((I-\Pi_T^k)R_T^{k+1}(0,v_{\partial T})|_F, v_F\big)_F \\ &+ \big(v_F, (I-\Pi_T^k)R_T^{k+1}(0,w_{\partial T})|_F\big)_F \\ &+ \big(\Pi_F^k(I-\Pi_T^k)R_T^{k+1}(0,v_{\partial T}), \Pi_F^k(I-\Pi_T^k)R_T^{k+1}(0,w_{\partial T})|_F\big)_F\Big\},\end{aligned} \tag{30}$$

and let $\sigma_T^\star$ be the local form defined by

$$\sigma_T^\star(\hat{y}_T;(0,v_{\partial T}),(0,w_{\partial T})) := \gamma\mu_T(\hat{y}_T)^2 \sum_{F\in\mathcal{F}_T} \eta_{TF}^{-1}\,(v_F,w_F)_F\,. \tag{31}$$

Then the equal-order stabilization form writes

$$\sigma_T(\hat{y}_T;(0,v_{\partial T}),(0,w_{\partial T})) = \sigma_T^\star(\hat{y}_T;(0,v_{\partial T}),(0,w_{\partial T})) + \zeta_T(\hat{y}_T;(0,v_{\partial T}),(0,w_{\partial T})). \tag{32}$$

Let us introduce the global forms $\sigma_h^\star(\hat{y}_h;(0,v_\mathcal{F}),(0,w_\mathcal{F})) := \sum_{T\in\mathcal{T}_h}\sigma_T^\star(\hat{y}_T;(0,v_{\partial T}),(0,w_{\partial T}))$ and $\zeta_h(\hat{y}_h;(0,v_\mathcal{F}),(0,w_\mathcal{F})) := \sum_{T\in\mathcal{T}_h}\zeta_T(\hat{y}_T;(0,v_{\partial T}),(0,w_{\partial T}))$, so that $\sigma_h = \sigma_h^\star + \zeta_h$. This leads to the following iterative procedure, with the same initial condition as for the mixed-order setting: For all $m \geqslant 0$, find $u_\mathcal{F}^{n,m+1} \in \mathcal{U}_{\mathcal{F},0}^k$ such that

$$\begin{aligned}\sigma_h^\star(\hat{u}_h^{n-1};(0,u_\mathcal{F}^{n,m+1}),(0,w_\mathcal{F})) = &-b_h((u_\mathcal{T}^n,u_\mathcal{F}^{n,m});(u_\mathcal{T}^n,u_\mathcal{F}^{n,m}),(0,w_\mathcal{F})) \\ &-\zeta_h(\hat{u}_h^{n-1};(0,u_\mathcal{F}^{n,m}),(0,w_\mathcal{F})) - \sigma_h(\hat{u}_h^{n-1};(u_\mathcal{T}^n,0),(0,w_\mathcal{F})),\end{aligned} \tag{33}$$

for all $w_\mathcal{F} \in \mathcal{U}_{\mathcal{F},0}^k$. At the algebraic level, we define two matrices $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star,n-1}$ and $\mathcal{Z}_{\mathcal{F}\mathcal{F}}^{n-1}$ such that $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{n-1} = \mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star,n-1} + \mathcal{Z}_{\mathcal{F}\mathcal{F}}^{n-1}$, $\mathcal{Z}_{\mathcal{F}\mathcal{F}}^{n-1}$ corresponds to the bilinear form $\zeta_h(\hat{u}_h^{n-1};\cdot,\cdot)$ and $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star,n-1}$ to $\sigma_h^\star(\hat{u}_h^{n-1};\cdot,\cdot)$. Then the splitting procedure translates into the following iterative algorithm: For all $m \geqslant 0$, find $\mathsf{U}_\mathcal{F}^{n,m+1} \in \mathbb{R}^{N_\mathcal{F}}$ such that

$$\mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star,n-1}\mathsf{U}_\mathcal{F}^{n,m+1} = -\mathsf{B}_\mathcal{F}(\mathsf{U}_\mathcal{T}^n,\mathsf{U}_\mathcal{F}^{n,m}) - \mathcal{Z}_{\mathcal{F}\mathcal{F}}^{n-1}\mathsf{U}_\mathcal{F}^{n,m} - \mathcal{S}_{\mathcal{F}\mathcal{T}}^{n-1}\mathsf{U}_\mathcal{T}^n. \tag{34}$$

As for the mixed-order setting, the convergence condition is that $\delta < 1$, where δ is the Lipschitz constant of the vector-valued function $(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star,n-1})^{-1}(\mathcal{Z}_{\mathcal{F}\mathcal{F}}^{n-1}(\cdot) + \mathsf{B}_\mathcal{F}(\mathsf{U}_\mathcal{T}^n,\cdot))$, i.e.,

$$\|(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star,n-1})^{-1}(\mathcal{Z}_{\mathcal{F}\mathcal{F}}^{n-1}(X-Y) + \mathsf{B}_\mathcal{F}(\mathsf{U}_\mathcal{T}^n,X) - \mathsf{B}_\mathcal{F}(\mathsf{U}_\mathcal{T}^n,Y))\| \leqslant \delta\|X-Y\|, \quad \forall X,Y \in \mathbb{R}^{N_\mathcal{F}}. \tag{35}$$

Remark 4.2 (Linear case)**.** In this case, the matrices $\mathcal{S}_{\mathcal{F}\mathcal{F}}^\star$ and $\mathcal{Z}_{\mathcal{F}\mathcal{F}}$ are independent of the time index n, and the stability condition (35) reads

$$\rho((\mathcal{S}_{\mathcal{F}\mathcal{F}}^\star)^{-1}(\mathcal{B}_{\mathcal{F}\mathcal{F}} + \mathcal{Z}_{\mathcal{F}\mathcal{F}})) < 1. \tag{36}$$

4.3 Sharper convergence analysis in the linear case

In this section we show that, in the linear case, the convergence criteria (29) and (36) can be achieved by choosing the coefficient γ scaling the stabilization large enough. Moreover, we derive explicit lower bounds on the coefficient γ that are uniform in the mesh size.

Invoking a discrete trace inequality (see, e.g., [27, Lem. 12.8] on simplicial meshes and [23, Lem. 1.46] on more general meshes), we infer that there exists a trace constant C_{tr} independent of h and which only depends on the polynomial order k and the mesh regularity parameter ρ such that

$$\max_{T\in\mathcal{T}_h}\max_{F\in\mathcal{F}_T}\max_{v\in\mathbb{P}_d^k(T)}\frac{|T|^{1/2}\|v\|_F}{|F|^{1/2}\|v\|_T} \leqslant C_{\mathrm{tr}}. \tag{37}$$

Moreover, we introduce a nondimensional geometric constant $\rho_\sharp$ such that

$$\max_{T\in\mathcal{T}_h}\left(\sum_{F\in\mathcal{F}_T}\frac{\eta_{TF}|F|}{|T|}\right)^{\frac12}\leqslant\rho_\sharp, \tag{38}$$

recalling that η_{TF} is the length scale used in the stabilization. The value of $\rho_\sharp$ can be chosen to be independent of h owing to the regularity of the mesh sequence.

Lemma 4.3 (Convergence in the mixed-order setting)**.** *In the linear case, a sufficient condition ensuring the convergence of the iterative algorithm* (28) *in the mixed-order setting is*

$$\gamma>(C_{\mathrm{tr}}\rho_\sharp)^2. \tag{39}$$

Proof. The main idea of the proof is the following equivalence regarding the convergence condition (29):

$$\rho(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{-1}\mathcal{B}_{\mathcal{F}\mathcal{F}})<1\iff\sigma_h((0,w_{\mathcal{F}}),(0,w_{\mathcal{F}}))>b_h((0,w_{\mathcal{F}}),(0,w_{\mathcal{F}})),\quad\forall w_{\mathcal{F}}\in\mathcal{U}_{\mathcal{F},0}^k. \tag{40}$$

Let $w_{\mathcal{F}}\in\mathcal{U}_{\mathcal{F},0}^k$ and let $T\in\mathcal{T}_h$. Recalling that μ_T denotes the constant value taken by μ in the cell $T\in\mathcal{T}_h$, we have

$$\begin{aligned}b_T((0,w_{\partial T}),(0,w_{\partial T}))&=\mu_T\big\|\boldsymbol{G}_T^k(0,w_{\partial T})\big\|_T^2=\mu_T\sum_{F\in\mathcal{F}_T}\big(\boldsymbol{G}_T^k(0,w_{\partial T})\cdot\boldsymbol{n}_{TF},w_F\big)_F\\&\leqslant\sum_{F\in\mathcal{F}_T}\mu_T^{\frac12}\eta_{TF}^{\frac12}\big\|\boldsymbol{G}_T^k(0,w_{\partial T})\cdot\boldsymbol{n}_{TF}\big\|_F\mu_T^{\frac12}\eta_{TF}^{-\frac12}\left\|w_F\right\|_F\\&\leqslant\left(\sum_{F\in\mathcal{F}_T}\mu_T\eta_{TF}\big\|\boldsymbol{G}_T^k(0,w_{\partial T})\cdot\boldsymbol{n}_{TF}\big\|_F^2\right)^{\frac12}\left(\sum_{F\in\mathcal{F}_T}\mu_T\eta_{TF}^{-1}\|w_F\|_F^2\right)^{\frac12},\end{aligned} \tag{41}$$

where we used the definition of the gradient reconstruction operator and the Cauchy–Schwarz inequality. Since $\boldsymbol{G}_T^k(0,w_{\partial T})\cdot\boldsymbol{n}_{TF}\in\mathbb{P}_d^k(T)$ because $\boldsymbol{n}_{TF}$ is a constant vector for all $F\in\mathcal{F}_T$, we can use the discrete inverse trace inequality (37) to infer that

$$\eta_{TF}^{\frac12}\big\|\boldsymbol{G}_T^k(0,w_{\partial T})\cdot\boldsymbol{n}_{TF}\big\|_F\leqslant C_{\mathrm{tr}}\|\boldsymbol{G}_T^k(0,w_{\partial T})\|_T\Big(\frac{\eta_{TF}|F|}{|T|}\Big)^{\frac12},\qquad\forall F\in\mathcal{F}_T.$$

Recognizing the definition of the mixed-order stabilization form without γ in the rightmost sum of (41), we obtain

$$b_T((0,w_{\partial T}),(0,w_{\partial T}))\leqslant C_{\mathrm{tr}}\mu_T^{\frac12}\left\|\boldsymbol{G}_T^k(0,w_{\partial T})\right\|_T\left(\sum_{F\in\mathcal{F}_T}\frac{\eta_{TF}|F|}{|T|}\right)^{\frac12}\left(\frac1\gamma\sigma_T((0,w_{\partial T}),(0,w_{\partial T}))\right)^{\frac12}.$$

Using the definition of $\rho_\sharp$ and that $b_T((0,w_{\partial T}),(0,w_{\partial T}))=\mu_T\big\|\boldsymbol{G}_T^k(0,w_{\partial T})\big\|_T^2$ gives

$$b_T((0,w_{\partial T}),(0,w_{\partial T}))\leqslant\frac{C_{\mathrm{tr}}^2\rho_\sharp^2}{\gamma}\sigma_T((0,w_{\partial T}),(0,w_{\partial T})).$$

Summing over all the mesh cells, we obtain

$$b_h((0,w_{\mathcal{F}}),(0,w_{\mathcal{F}}))\leqslant\frac{C_{\mathrm{tr}}^2\rho_\sharp^2}{\gamma}\sigma_h((0,w_{\mathcal{F}}),(0,w_{\mathcal{F}})).$$

Thus, the splitting converges if $\frac{C_{\mathrm{tr}}^2\rho_\sharp^2}{\gamma}<1$, which proves the claim. □

The equal-order setting does not provide the same simplicity and requires a hypothesis on the spectrum of the matrix $(\mathcal{S}^\star_{\mathcal{F}\mathcal{F}})^{-1}\mathcal{S}_{\mathcal{F}\mathcal{F}}$. Let us define

$$\alpha := \rho((\mathcal{S}^\star_{\mathcal{F}\mathcal{F}})^{-1}\mathcal{Z}_{\mathcal{F}\mathcal{F}}) = \rho((\mathcal{S}^\star_{\mathcal{F}\mathcal{F}})^{-1}\mathcal{S}_{\mathcal{F}\mathcal{F}}) - 1. \tag{42}$$

Notice that the value of α does not depend on γ (since all the involved matrices are proportional to γ) and that the matrices considered in (42) are symmetric.

Lemma 4.4 (Convergence in the equal-order setting)**.** *In the linear case and under the assumption $\alpha < 1$, a sufficient condition for the convergence of the iterative algorithm* (33) *in the equal-order setting is*

$$\gamma > \frac{(C_{\mathrm{tr}}\rho_\sharp)^2}{1-\alpha}. \tag{43}$$

Proof. The following equivalence holds regarding the convergence condition (36):

$$\begin{aligned} &\rho((\mathcal{S}^\star_{\mathcal{F}\mathcal{F}})^{-1}(\mathcal{B}_{\mathcal{F}\mathcal{F}} + \mathcal{Z}_{\mathcal{F}\mathcal{F}})) < 1 \\ &\iff \sigma^\star_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) > b_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) + \zeta_h((0,w_\mathcal{F}),(0,w_\mathcal{F})), \quad \forall w_\mathcal{F} \in \mathcal{U}^k_{\mathcal{F},0}. \end{aligned} \tag{44}$$

Since ζ_h and $\sigma^\star_h$ are symmetric and $\sigma^\star_h$ is nonnegative, the definition of α means that

$$\zeta_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) \leqslant \alpha\sigma^\star_h((0,w_\mathcal{F}),(0,w_\mathcal{F})), \quad \forall w_\mathcal{F} \in \mathcal{U}^k_{\mathcal{F},0}. \tag{45}$$

Considering this, a sufficient condition to obtain the bound announced in (44) is

$$(1-\alpha)\sigma^\star_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) > b_h((0,w_\mathcal{F}),(0,w_\mathcal{F})), \quad \forall w_\mathcal{F} \in \mathcal{U}^k_{\mathcal{F},0}. \tag{46}$$

Indeed, if (46) holds, we infer that

$$b_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) < \sigma^\star_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) - \alpha\sigma^\star_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) \leqslant \sigma^\star_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) - \zeta_h((0,w_\mathcal{F}),(0,w_\mathcal{F})).$$

Lemma 4.3 established that, if $\gamma > (C_{\mathrm{tr}}\rho_\sharp)^2$, then,

$$\sigma^\star_h((0,w_\mathcal{F}),(0,w_\mathcal{F})) > b_h((0,w_\mathcal{F}),(0,w_\mathcal{F})), \quad \forall w_\mathcal{F} \in \mathcal{U}^k_{\mathcal{F},0}. \tag{47}$$

Therefore, taking $\gamma > \frac{(C_{\mathrm{tr}}\rho_\sharp)^2}{1-\alpha}$ ensures the condition (46). This concludes the proof. □

Remark 4.5 (Assumption $\alpha < 1$)**.** This assumption is verified numerically in Section 4.4, see in particular Table 2. Although a proof of this assumption is not available, our numerical experiments indicate that it is reasonable to expect that it holds for polynomial degrees of 0 to 4 and relatively simple mesh shapes.

Remark 4.6 (Local vs. global spectral radius)**.** We notice that the minimal value of γ given in Lemmas 4.3 and 4.4 is derived by reasoning locally on a single mesh cell $T \in \mathcal{T}_h$. Slightly sharper values can be derived by reasoning globally on the mesh and taking into account the homogeneous Dirichlet boundary conditions. This point is further quantified in the next section.

4.4 Numerical study of the stability parameter in the linear case

In this section, we evaluate numerically the influence of the stability parameter γ on the splitting procedure for various polynomial orders. Both equal-order and mixed-order settings are tested, with polynomial orders $k \in \{0,1,2,3,4\}$, as well as various mesh types: Cartesian in two and three dimensions, unstructured quadrangles, structured and unstructured triangles and unstructured tetrahedra. The polynomial order for the HHO method is indicated via the pair of integers (l,k) with l for the cell unknowns and k for the face unknowns.

We first verify that the minimum value of the parameter γ ensuring the convergence of the splitting procedure is bounded, for all the polynomial orders, uniformly in the mesh size. To this purpose, we first

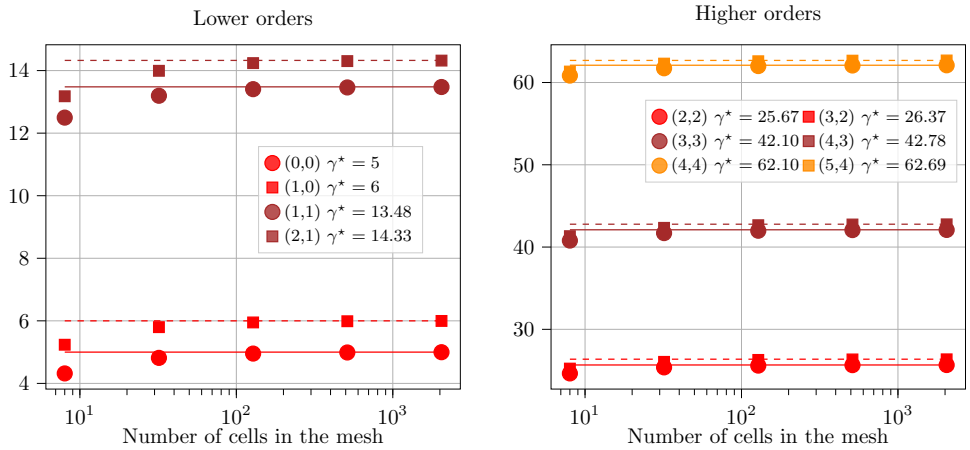


Figure 1: Spectral radius as a function of the mesh size in right-triangular cells with homogeneous Dirichlet conditions in 2D, compared to the reference value on a single cell without boundary conditions (horizontal lines).

consider a series of refined right-triangular meshes of $\Omega := (0,1)^2$ and compute the value of γ via the spectral radius of $\mathcal{S}_{\mathcal{FF}}^{-1}\mathcal{B}_{\mathcal{FF}}$ in the mixed-order setting and the spectral radius of $\mathcal{S}_{\mathcal{FF}}^{-1}(\mathcal{B}_{\mathcal{FF}}+\mathcal{Z}_{\mathcal{FF}})$ in the equal-order setting, with homogeneous Dirichlet boundary conditions. We compare the resulting values to the value $\gamma^\star$ obtained in a single cell without boundary conditions. Figure 1 illustrates, as expected, that $\gamma^\star$ gives a reliable upper bound on γ. This upper bound turns out to be quite sharp even on moderately refined meshes. Therefore, $\gamma^\star$ is a very good minimal value for γ to be used in practice.

Table 1 reports the value of $\gamma^\star$ for a square cell, a right-isoceles triangular cell as well as estimates for general quadrangular and triangular cells belonging to shape-regular sequences of unstructured meshes. The triangular meshes are generated using `gmsh`, and the quadrangular meshes are created from the triangular meshes by merging pairs of adjacent triangles. For these two meshes, the reported value of $\gamma^\star$ is the largest observed value, rounded to the above integer on all the mesh cells and for all the meshes in the sequence. One notices that, in all cases, the value of $\gamma^\star$ increases with the polynomial order. Moreover, for a given polynomial order, the smallest value of $\gamma^\star$ is obtained on square cells, and the largest value on unstructured triangular meshes.

Order (cell, face)	**(0,0)**	**(1,1)**	**(2,2)**	**(3,3)**	**(4,4)**	**(1,0)**	**(2,1)**	**(3,2)**	**(4,3)**	**(5,4)**
Squares	1	5	11	19	29	2	6	12	20	30
Right triangles	5	13.48	25.67	42.10	62.10	6	14.33	26.37	42.78	62.69
Unstructured quadrangles	3	9	19	32	48	4	9	19	32	48
Unstructured triangles	6	15	28	45	65	7	15	28	45	65

Table 1: $\gamma^\star$ computed via the spectral radius of $\mathcal{S}_{\mathcal{FF}}^{-1}\mathcal{B}_{\mathcal{FF}}$ or $\mathcal{S}_{\mathcal{FF}}^{-1}(\mathcal{B}_{\mathcal{FF}}+\mathcal{Z}_{\mathcal{FF}})$ on a single cell (first two lines) and on a shape-regular mesh sequence (last two lines), $k \in \{0,1,2,3,4\}$, equal- and mixed-order settings.

As a second verification, we check the condition $\alpha < 1$ in the equal-order setting on all the previous cases. The value of α appears not to depend on the mesh size when homogeneous Dirichlet boundary conditions are enforced. As before, when unstructured meshes are used, the largest value observed on the mesh sequence is reported. As seen in Table 2, the value of α is in all cases smaller than 1, thereby confirming the assumption made in Lemma 4.4. The lowest-order setting even yields α close to zero. The value of α increases with the polynomial order before it stabilizes for higher orders. For unstructured mesh sequences, the finer meshes are more uniform leading to a smaller value of α. Therefore, the values reported in Table 2 are pessimistic on finer meshes.

A classical result on the leapfrog scheme (24) gives a stability condition on the time step depending on

Order (cell, face)	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)
Squares	≈ 0	≈ 0	0.19	0.26	0.23
Right triangles	≈ 0	0.28	0.38	0.36	0.39
Unstructured quadrangles	0.13	0.21	0.44	0.52	0.61
Unstructured triangles	≈ 0	0.17	0.25	0.24	0.32

Table 2: α, the largest eigenvalue of $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{-1}\mathcal{Z}_{\mathcal{F}\mathcal{F}}$, $k \in \{0, 1, 2, 3, 4\}$, equal- and mixed-order settings.

the spectral radius of the stiffness matrix (see, e.g., [32]). Here, the stiffness matrix to be considered is

$$\mathcal{D}(\gamma) := \mathcal{M}^{-1}(\mathcal{A}_{\mathcal{T}\mathcal{T}}(\gamma) - \mathcal{A}_{\mathcal{T}\mathcal{F}}(\gamma)\mathcal{A}_{\mathcal{F}\mathcal{F}}(\gamma)^{-1}\mathcal{A}_{\mathcal{F}\mathcal{T}}(\gamma)), \tag{48}$$

where $\mathcal{A}(\gamma)$ is the stiffness matrix defined in (23) with the dependence on γ made explicit. The stability condition on Δt then reads

$$\Delta t(\gamma) \leqslant \Delta t^{\text{opt}}(\gamma) := \frac{2}{\sqrt{\rho(\mathcal{D}(\gamma))}}. \tag{49}$$

This allows us to compare stability conditions for different values of γ. In Table 3, we compare values for $\gamma = 1$, i.e. without splitting, to those for $\gamma = \gamma^\star$ from Table 1. The results show that using the splitting procedure leads to a reduction of the time step by at most a factor of two on squares and by at most three on right-isoceles triangles.Moreover, increasing the polynomial degree generally alleviates the tightening of the stability condition except on triangular meshes.

Order (cell, face)	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(1,0)	(2,1)	(3,2)	(4,3)	(5,4)
Squares	1.00	0.75	0.63	0.68	0.60	0.66	0.54	0.52	0.54	0.52
Right triangles	0.37	0.58	0.65	0.54	0.70	0.40	0.36	0.41	0.53	0.63
Unstructured quadrangles	0.60	0.43	0.32	0.37	0.43	0.45	0.41	0.39	0.47	0.51
Unstructured triangles	0.38	0.30	0.42	0.47	0.56	0.25	0.32	0.39	0.44	0.51

Table 3: $\Delta t^{\text{opt}}(\gamma^\star)/\Delta t^{\text{opt}}(1)$, showing the tightening of the stability condition induced by $\gamma^\star$ on 2D meshes, $h = 0.1$, $k \in \{0, 1, 2, 3, 4\}$, equal- and mixed-order settings.

The same studies can be performed on 3D meshes. Only Cartesian hexahedral and unstructured tetrahedral meshes with polynomial orders up to $k \leqslant 2$ are presented. Table 4 summarizes the results. Comparing square and hexahedral meshes, going to 3D does not deteriorate too much the stability condition. On tetrahedral meshes, however, $\gamma^\star$ grows faster with the polynomial degree, and the splitting procedure impacts more strongly the stability condition, by at most a factor of five.

Order (cell, face)	(0,0)	(1,1)	(2,2)	(1,0)	(2,1)	(3,2)
$\gamma^\star$ **hexahedra**	1.83	8.5	17	2.83	9.5	18
$\Delta t^{\text{opt}}(\gamma^\star)/\Delta t^{\text{opt}}(1)$ **hexahedra**	0.57	0.58	0.58	0.5	0.42	0.48
$\gamma^\star$ **tetrahedra**	33	60	90	34	60	90.5
$\Delta t^{\text{opt}}(\gamma^\star)/\Delta t^{\text{opt}}(1)$ **tetrahedra**	0.19	0.17	0.51	0.16	0.26	0.57

Table 4: Value of $\gamma^\star$ and tightening of the stability condition induced by $\gamma^\star$ on 3D meshes (Cartesian hexahedra and unstructured tetrahedra), $h = 0.55$, $k \in \{0, 1, 2\}$, equal- and mixed-order settings.

5 Numerical results

In this section, we present numerical results on both linear and nonlinear wave propagation problems. We start with an analytical linear test case, where we analyze the impact of the number of splitting iterations

and the value of the stabilization parameter γ on the error and on the execution time. We then discuss two nonlinear test cases, in order to compare the splitting procedure with the implicit version in terms of execution time. We also compare the HHO method with first- and second-order finite elements for the space semi-discretization.

All the computations are performed with `MANTA`, a `C++` library developed at CEA Saclay. It relies on `PETSc` for the algebraic solvers, and the computations are spatially and equally distributed on MPI processes. The computations are run on the `TGCC` (*Très Grand Centre de Calcul*) computers, on Intel architecture cores, Skylake@2.7GHz (AVX512) with 3.75 Gb of RAM. All the computations can use the memory of the entire node, composed of 48 cores.

5.1 Linear test cases

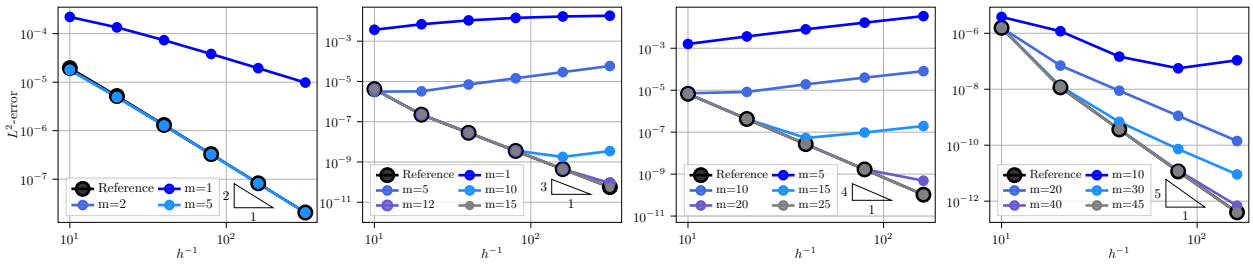
We consider a manufactured solution in $\Omega := (0,1)^2$ with a nonpolynomial behavior in space and a quadratic time behavior. Specifically, we set $\mu := 1$ in (1) and

$$u(x,y,t) := t^2 \sin(\pi x)\sin(\pi y), \tag{50}$$

leading to homogeneous Dirichlet conditions and zero initial conditions. The source term is $f(x,y,t) := 2(\pi^2 t^2 + 1)\sin(\pi x)\sin(\pi y)$. Here, there is no time discretization error, since the time integration scheme is of order 2.

5.1.1 Convergence rates

Figure 2 illustrates the convergence of the L^2-error at the final time $t = 1.0$ with respect to the number of iterations in the splitting procedure on a series of spatially refined square meshes, focusing on the equal-order setting. (The results for the mixed-order setting show the same behavior and are not displayed for brevity.) The time step is computed so that the ratio $\frac{h}{\Delta t}$ remains constant and smaller than the stability condition (49). The reference curve is the error without splitting. The value of γ is determined from Table 1 by setting $\gamma := 1.5\gamma^\star$. For a given number of iterations in the splitting procedure, the error ends up stagnating below a certain value of h, which corresponds to the splitting error being larger than the discretization error. The stagnation value and the mesh size for which stagnation starts both decrease as the number of iterations increases. When the splitting error is sufficiently small, the space discretization error prevails. Moreover, the higher the polynomial order, the more iterations needed to converge. Figure 2 also shows the convergence order of the L^2-error depending on the mesh size h, which is, as expected, h^{k+2}.



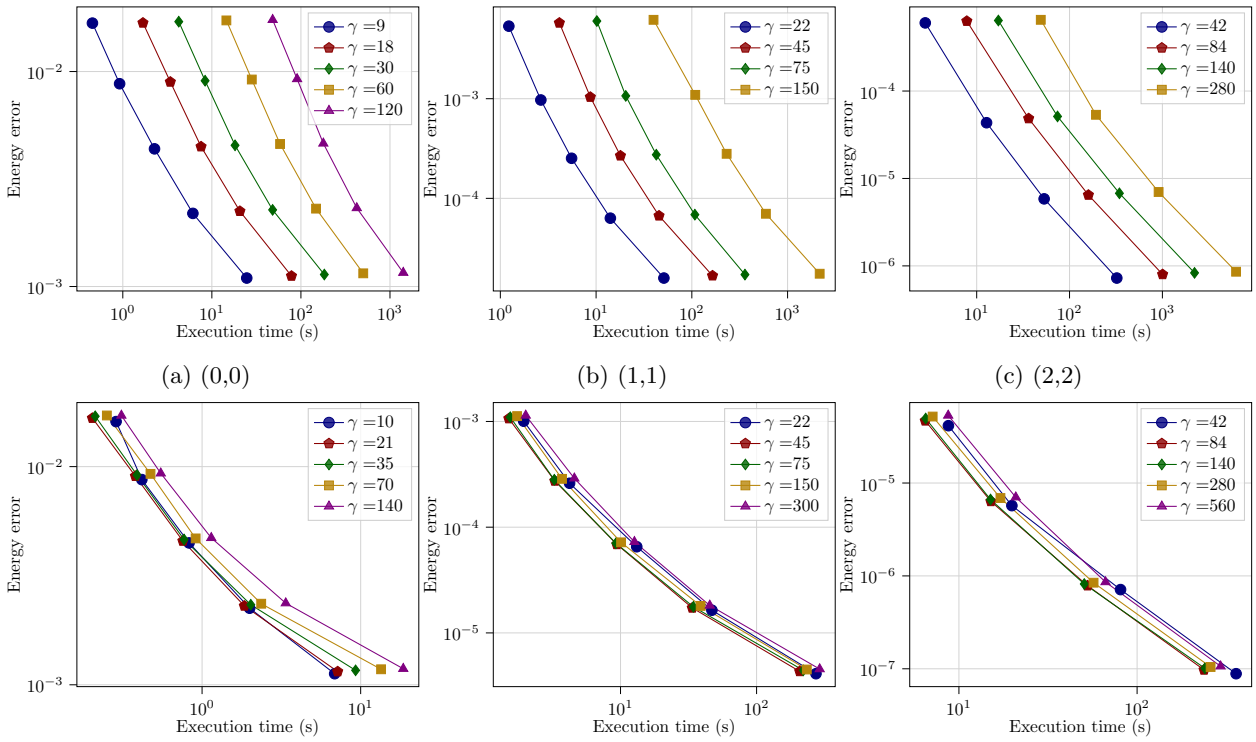
(a) (0,0), $\gamma = 1.5$ (b) (1,1), $\gamma = 7$ (c) (2,2), $\gamma = 20$ (d) (3,3), $\gamma = 25$

Figure 2: Linear wave problem - Convergence of the L^2-error at $t = 1.0$ as a function of the mesh size for varying numbers of splitting iterations m, $k \in \{0,1,2,3\}$, equal-order setting. The reference corresponds to the semi-implicit scheme ($m \to \infty$).

5.1.2 Impact of the stabilization parameter γ

We now focus on the dependence of the error and of the execution time on the value of γ. Since there is no error in time, the error is the sum of two errors, coming from the space discretization on the one hand

and the splitting procedure on the other hand. The splitting error has two sources: the truncation error of the iterative procedure and the error caused by the variation of γ. The previous test case illustrated the convergence of the truncation error. The impact of the value of γ on the error is actually quite moderate (see Appendix A for further insight). Regarding the execution time, we expect that the critical time step decreases as γ increases, thus impacting the computational cost. Further insight into this impact is again provided in Appendix A.



(a) (0,0) (b) (1,1) (c) (2,2)

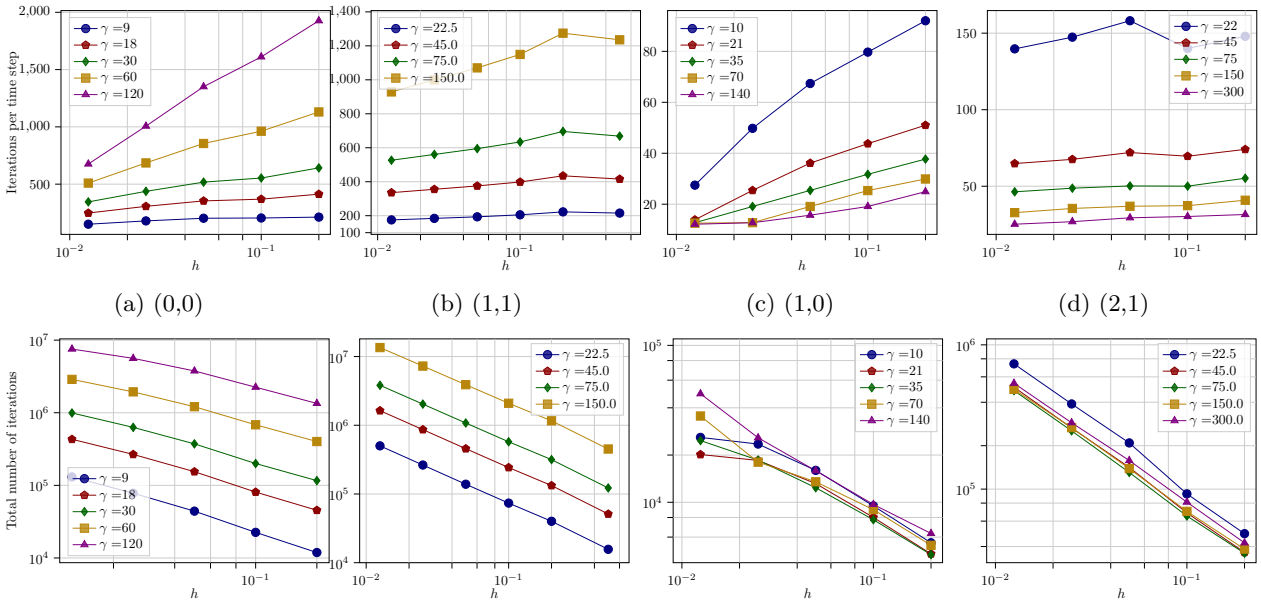
(d) (1,0) (e) (2,1) (f) (3,2)

Figure 3: Linear wave problem - Energy error at $t = 0.1$ as a function of the computation time, $\gamma \in \{1.5\gamma^\star, 3\gamma^\star, 5\gamma^\star, 10\gamma^\star\}$, $k \in \{0, 1, 2\}$, equal- and mixed-order settings.

Figure 3 shows the discrete energy error at $t = 0.1$ as a function of the computation time for the equal- and mixed-order settings, with $k \in \{0, 1, 2\}$. Order 3 is not shown for brevity. The discrete energy error, computed using the projection of the exact solution onto the mesh cells and faces, is the Hilbertian sum of the L^2-error on the cell velocities and the hybrid H^1-error evaluated using the gradient reconstruction. Computations are performed on a sequence of unstructured triangular meshes, obtained by repeated refinements of a coarse initial mesh. This ensures that all the meshes have the same regularity. Computations are run on 16 cores, and the meshes are equally distributed on all the processors. Each marker on the curves of Figure 3 corresponds to the error and execution time on a mesh from the sequence. The scaling parameter γ takes the values $\{1.5\gamma^\star, 3\gamma^\star, 5\gamma^\star, 10\gamma^\star\}$ and, when the computation cost is reasonable, $20\gamma^\star$. For each value of γ, the critical time step $\Delta t^{\text{opt}}(\gamma)$ can be computed via equation (49), and the actual time step is then set to $\Delta t := 0.8\Delta t^{\text{opt}}(\gamma)$. Unlike the experiments in Figure 2, the number of iterations for the splitting procedure is not fixed here. We rather stop the splitting procedure when the relative norm of the increment is smaller than $\epsilon := 10^{-11}$.

The first salient point is that, for a given mesh, the error does not depend on γ. Indeed, all the corresponding markers are almost horizontally aligned. This means that a reasonable increase in γ does not deteriorate the quality of the solution. The second salient point concerns the computation time. Here, the behavior differs between equal- and mixed-order settings. In the equal-order setting, the larger γ, the more

expensive the computation. This could have two origins: a smaller critical time step for larger γ, requiring more time steps to reach the final time, or a higher number of splitting iterations. Figure 4 and Table 5 answer this question. The mean number of splitting iterations for each time step is displayed in Figures 4a and 4b for (0,0) and (1,1), respectively. As γ increases, more iterations are needed for the splitting procedure to converge. Moreover, Table 5 reports the critical time step $\Delta t(\gamma)$ for the mesh size $h = 0.1$ and all polynomial orders. We observe that the critical time step $\Delta t(\gamma)$ decreases when γ increases. When both factors are combined, this leads to a swift increase of the computational cost with γ. On the contrary, in the mixed-order setting, there seems to be an optimal value of γ, for which the computation cost is the smallest. Indeed, the number of iterations per time step decreases as γ increases (see Figures 4c and 4d), but the critical time step decreases as well. For the considered polynomial orders, this optimum is in the interval $\gamma \in [3\gamma^\star, 5\gamma^\star]$. Moreover, comparing Figures 4e and 4f with Figures 4g and 4h highlights that, for a given polynomial order, the total number of iterations in the mixed-order setting is smaller than that in the equal-order setting, by a factor of 10 or more. At the same time, Table 5 shows that the critical time step is just slightly smaller in the mixed-order setting. Combining these two factors makes the mixed-order computations faster than the equal-order ones, on a given mesh and for a given polynomial order.



(a) (0,0) (b) (1,1) (c) (1,0) (d) (2,1)

(e) (0,0) (f) (1,1) (g) (1,0) (h) (1,1)

Figure 4: Linear wave problem - Mean number of splitting iterations per time step (top) and total number of splitting iterations (bottom) in log scale, as a function of the mesh size, $k \in \{0, 1\}$, equal- and mixed-order settings.

Order	(0,0)	(1,1)	(2,2)	(3,3)	(1,0)	(2,1)	(3,2)	(4,3)
$1.5\gamma^\star$	4.61e-2	1.40e-2	3.75e-3	1.12e-3	4.25e-2	7.57e-3	1.75e-3	4.037e-4
$3\gamma^\star$	2.30e-2	8.23e-3	2.33e-3	7.11e-4	2.75e-2	5.09e-3	1.23e-3	2.88e-4
$5\gamma^\star$	1.39e-2	5.51e-3	1.67e-3	5.14e-4	2.06e-2	3.84e-3	9.46e-4	2.23e-4
$10\gamma^\star$	7.07e-3	2.75e-3	1.09e-3	3.34e-4	1.42e-2	2.66e-3	6.58e-4	1.58e-4
$20\gamma^\star$	3.59e-3	1.27e-3	5.56e-4	1.82e-4	9.89e-3	1.86e-3	4.61e-4	1.11e-4

Table 5: Linear wave problem - Critical time step $\Delta t(\gamma)$, $h = 0.1$, $\gamma \in \{1.5\gamma^\star, 3\gamma^\star, 5\gamma^\star, 10\gamma^\star, 20\gamma^\star\}$, $k \in \{0, 1, 2, 3\}$, equal- and mixed-order settings.

The above results allow us to determine the optimal value of γ: $\gamma = \gamma^\star$ in the equal-order setting and

$\gamma \in [3\gamma^\star, 5\gamma^\star]$ in the mixed-order setting. Using these values of γ, we can now compare the error as a function of the computational cost for all polynomial orders. Figure 5 shows the energy error for polynomial degrees $k \in \{0, 1, 2, 3\}$. These results illustrate the fact that, for a given error or a given execution time, taking higher orders on a coarse mesh is more efficient than lower orders on a fine mesh. Moreover, the mixed- and equal-order settings essentially lead to similar curves, but with an offset. Indeed, on a given mesh, the mixed-order setting is faster than the equal-order in order to reach the same error.

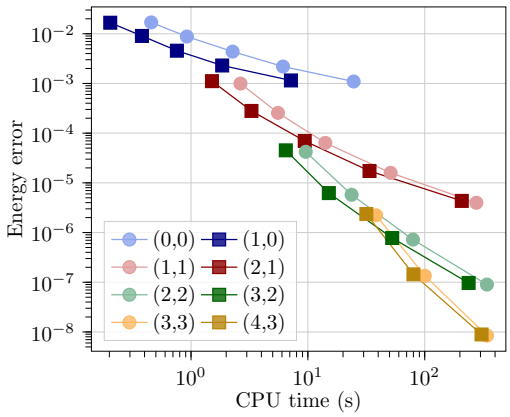


Figure 5: Linear wave problem - Energy error with optimal γ as a function of the computation time, with $(k, \gamma) \in \{(0, 9), (1, 22.5), (2, 42), (3, 67.5)\}$ in the equal-order setting and $(k, \gamma) \in \{(0, 1), (1, 45), (2, 84), (3, 135)\}$ in the mixed-order setting.

5.2 Nonlinear wave equation with p-structure

In this section, we study the splitting algorithm on a nonlinear wave equation where the diffusion operator in space has a so-called p-structure:

$$\partial_t^2 u + \nabla \cdot \left((\mu_0^2 + |\nabla u|^2)^{\frac{p-2}{2}} \nabla u \right) = f, \quad \text{in } \Omega, \forall t \in J, \tag{51}$$

with $\mu_0 \in \mathbb{R}_+$ and $p \in (1, +\infty)$. For the value $p = 2$, (51) corresponds to the linear wave equation. The associated Hamiltonian writes $\mathcal{H}^p(g) := \frac{1}{p}(\mu_0^2 + |g|^2)^{\frac{p}{2}}$ for all $g \in \mathbb{R}^2$, and the nonlinear wave equation (51) can be rewritten as $\partial_t^2 u + \nabla \cdot (\nabla_g \mathcal{H}^p(\nabla u)) = f$. The local nonlinear stiffness form b_T now writes

$$b_T(\hat{y}_T; \hat{v}_T, \hat{w}_T) := \left((\mu_0^2 + |\boldsymbol{G}_T^k(\hat{y}_T)|^2)^{\frac{p-2}{2}} \boldsymbol{G}_T^k(\hat{v}_T), \boldsymbol{G}_T^k(\hat{w}_T) \right)_T. \tag{52}$$

We set $\Omega := (0, 1)^2, \Theta := 0.8, f := 0$ and we enforce homogeneous Dirichlet boundary conditions, null initial condition for u, and the initial velocity $v_0(x, y) := 5 \sin(\pi x) \sin(\pi y)$. In the linear case ($p = 2$), the exact solution is $u(x, y, t) = \frac{5}{\sqrt{2}\pi} \sin(\sqrt{2}\pi t) \sin(\pi x) \sin(\pi y)$. When $p \neq 2$, the exact solution is not known and the error is computed using a reference solution (obtained with high polynomial orders $(4, 3)$ and a fine mesh with $h \approx 0.0003$). If $\mu_0 = 0$, the diffusion operator essentially behaves as the p-Laplace operator. On the contrary, if μ_0 is much larger than $|\nabla u|$, the problem is nearly linear and the behavior does not differ too much from a linear diffusion operator. In order to avoid such cases, the value of μ_0 is chosen so that it is not significantly smaller than $\|\nabla u\|_{L^\infty(\Omega \times J)}$. Here, we take $\mu_0^2 := 0.5$. Moreover, to scale the stabilization, we take the same value $\bar{\mu}_T := 5$ in each cell and at each time step. Based on our simulations (see Figure 6) which cover the range $p \in [1.25, 5]$, this appears to be a reasonable choice for scaling the stabilization.

In our numerical experiments, we consider an unstructured mesh sequence, obtained by repeatedly refining an initial coarse mesh, thus preserving the same regularity in the entire mesh sequence. To compute the errors, we consider ten sensors placed in the triangle $\{x \in [0, 0.5], 0 \leqslant y \leqslant x\}$, and compute the value of the solution, u, the velocity, $\partial_t u$, the acceleration, $\partial_t^2 u$, and of both gradient components, $\partial_x u, \partial_y u$, over the simulation

at these ten sensors. The sensors are placed only in the above triangle by symmetry arguments. In Figure 6, the solution at three sensors is shown for seven values of p. The values for which $p \neq 2$ can be regrouped into pairs of conjugated values (p, p') such that $\frac{1}{p} + \frac{1}{p'} = 1$. Specifically, we choose $p \in \{2.5, 3, 5\}$ and the conjugate values $p' \in \{1.67, 1.5, 1.25\}$. The larger $|p - 2|$, the harder the simulation, and the more refined the mesh needs to be in order to capture the nonlinear variations.

We first make computations with various values of the stability parameter γ and $p = 3$. The minimal value $\gamma^\star$ leading to a converging splitting depends on the polynomial order and on the value of p, and cannot be deduced from the linear study. Table 6 reports the smallest value of γ observed numerically, denoted by $\gamma^{\min}$ in what follows. Since this value is an estimation, it is rounded to the smallest larger integer.

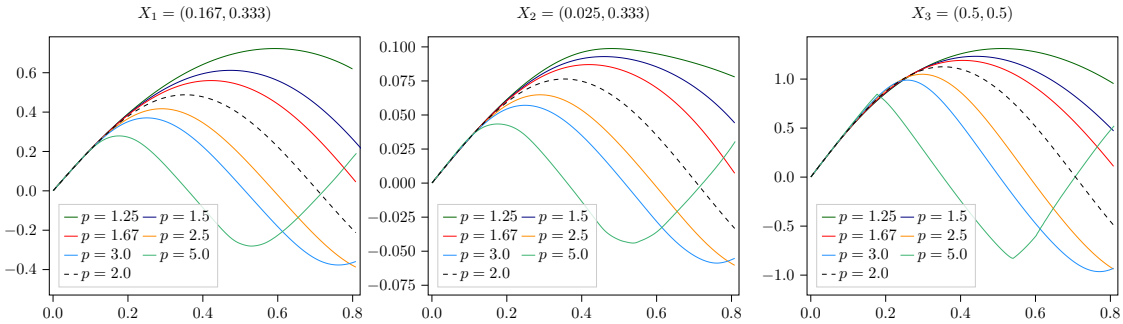
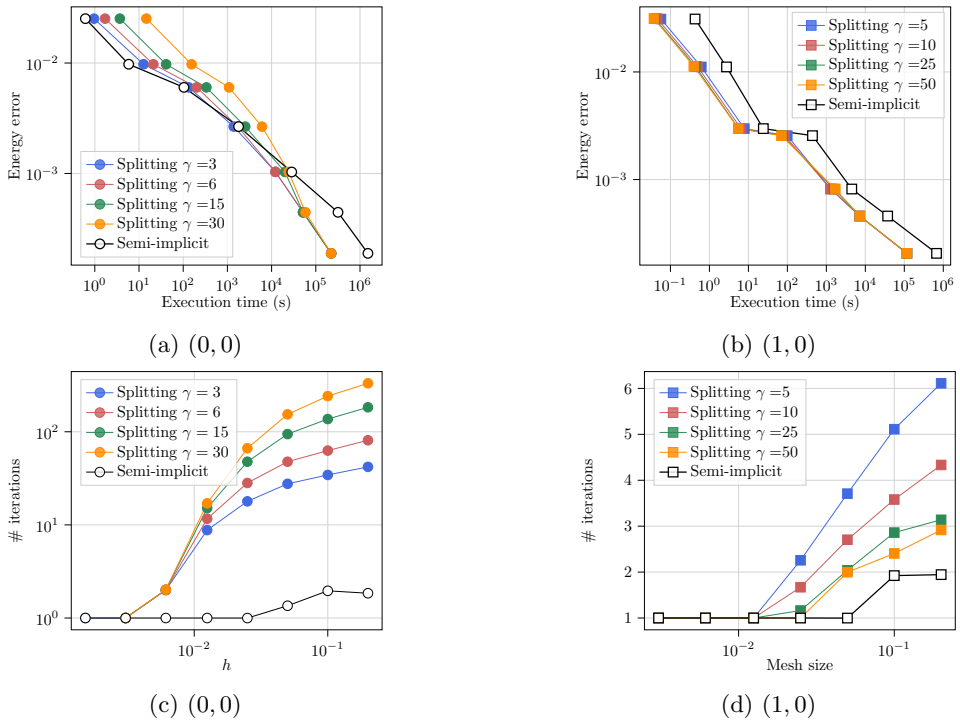


Figure 6: Nonlinear (p-structure) wave problem with $p \in \{1.25, 1.5, 1.67, 2, 2.5, 3, 5\}$ - Solution at sensors placed at $(0.167, 0.333)$, $(0.025, 0.333)$ and $(0.5, 0.5)$ on the time interval $[0, 0.8]$.

	(0,0)	(1,1)	(2,2)	(3,3)	(1,0)	(2,1)	(3,2)	(4,3)
$p = 3$	2	10	20	25	5	20	20	25

Table 6: Nonlinear (p-structure) wave problem with $p = 3$ - Minimal value $\gamma^{\min}$ for the splitting procedure to converge, $k \in \{0, 1, 2, 3\}$, equal- and mixed-order settings.

The objective of the splitting being a reduction of computational costs, we first compare computation times between the splitting procedure and the semi-implicit scheme (based on a Newton solver). Computations are run sequentially. The Newton linear system is solved with a direct solver since experiments with iterative solvers yield similar computation times. The time step is chosen so that the time discretization error and the space discretization error are equilibrated. For all values of γ, this leads to an equilibrated time step that is smaller than the critical time step. Thus, the computation time is not affected by the reduction of the critical time step for larger γ. On the one hand, Figure 7 reports the maximum of the discrete energy error, evaluated using the reference solution as the maximum value over the ten sensor points and all the time steps. On the other hand, Figure 7 also reports the mean number of splitting and Newton iterations per time step for each value of γ and each mesh. The results in Figure 7 deal with the lowest-order cases $(0, 0)$ and $(1, 0)$. The first salient observation is that, as for the linear case, the behavior differs between the equal- and mixed-order settings. In the equal-order setting, the number of iterations is higher when γ increases, whereas there is an optimal value in the mixed-order setting, which is not $\gamma^\star$. In the equal-order setting, the splitting procedure takes more time than the semi-implicit scheme on the coarsest meshes, but is (much) faster on the finest meshes. The gain in computation time (i.e. the ratio of the semi-implicit time over the splitting time) is reported in Table 7. In the mixed-order setting, the splitting procedure is always (much) faster than the semi-implicit scheme, and the gain in computation time increases as the mesh is refined. The number of iterations for the splitting and the semi-implicit scheme decreases when the mesh is refined. This is due to the time step being also refined to satisfy the requirement on error balancing. Hence, the static problem on the face unknowns becomes easier to solve. On the four coarsest meshes, in the equal-order setting, the number of splitting iterations is very large (more than 100 on the coarsest mesh). This explains the not so good performance of the splitting procedure on the coarsest meshes (see Figure 7a). In the mixed-order



(a) $(0,0)$ (b) $(1,0)$

(c) $(0,0)$ (d) $(1,0)$

Figure 7: Nonlinear (p-structure) wave problem with $p = 3$ - Energy error as a function of computation time (top) and mean number of iterations per time step as a function of mesh size (bottom), $\gamma \in \{\gamma^{\min}, 2\gamma^{\min}, 5\gamma^{\min}, 10\gamma^{\min}\}$, polynomial orders $(0,0)$ and $(1,0)$.

Mesh size h	0.0063	0.0031	0.0016
(0,0)	2.34	6.19	6.80
(1,0)	3.44	5.19	5.74

Table 7: Nonlinear (p-structure) wave problem with $p = 3$ - Gain in computation time of the splitting procedure over the semi-implicit scheme on the three finest meshes, $\gamma = \gamma^{\min}$, polynomial orders $(0,0)$ and $(1,0)$.

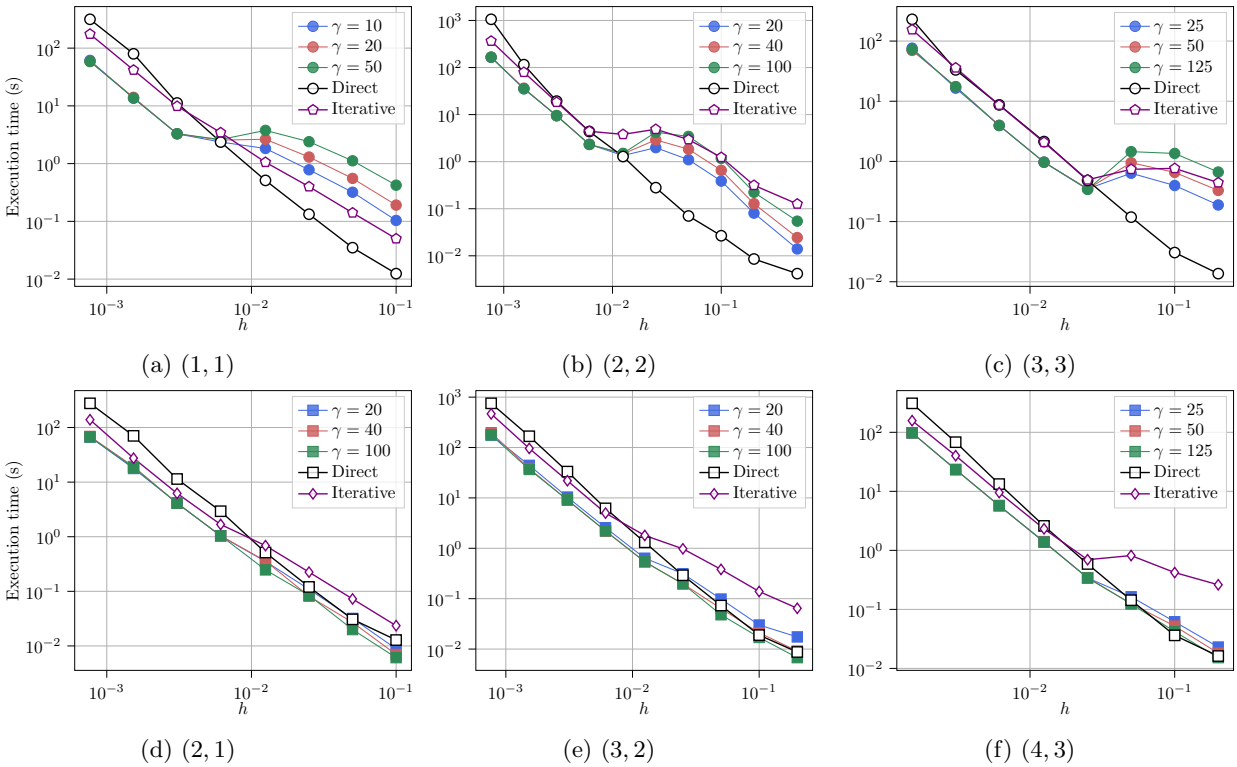
setting, the number of iterations is always quite low (less than 10), which explains the better performances observed in Figure 7b. Since the number of iterations for all the values of γ becomes small on the finest meshes, the value of γ does not impact the execution time in this case.

The above experiments in the linear case and in the lowest-order nonlinear case show that the error does not depend on γ as long as this parameter remains in the range $[1, 150]$. Thus, for higher polynomial degrees, we only focus on the execution time. Figure 8 displays the average execution time of a single time step for each method and each polynomial order on the same family of unstructured triangular meshes as above. Here, we compare the sequential execution times for the splitting procedure with $\gamma \in \{\gamma^{\min}, 2\gamma^{\min}, 5\gamma^{\min}\}$ and the semi-implicit scheme with either a direct solver or a GMRES iterative solver with a Jacobi preconditioner. The gain in computation time is reported in Table 8 only for $\gamma = \gamma^{\min}$ since the value of γ does not impact strongly the execution time on the considered meshes. The gain is given on the four finest meshes, exception made for the polynomial order $k = 3$, for which the semi-implicit scheme on the finest mesh is too expensive with our current implementation. As expected, the equal- and mixed-order settings behave differently on the coarse meshes. In the equal-order setting, the splitting procedure takes more time on the coarse meshes, but this execution time is reduced as the number of iterations diminishes, so that the splitting procedure becomes computationally efficient for mesh sizes smaller than $h \approx 0.006$. In the mixed-order setting, the execution time of the splitting procedure is always smaller or equivalent to the execution time of the semi-implicit

	(1,1)		(2,1)		(2,2)		(3,2)		(3,3)		(4,3)	
Mesh size	direct	gmres	direct	gmres	direct	gmres	direct	gmres	direct	gmres	direct	gmres
0.00613	1.01	1.46	2.84	1.62	1.87	1.89	2.42	1.94	2.19	2.13	2.33	1.60
0.00306	3.41	3.01	2.73	1.49	2.04	1.94	3.21	2.11	2.02	2.13	2.90	1.71
0.00156	5.84	3.06	3.97	1.54	3.24	2.21	3.77	2.17	3.01	2.17	3.16	1.67
0.00078	5.15	2.85	4.13	2.07	6.43	2.23	4.25	2.63	NA	NA	NA	NA

Table 8: Nonlinear (p-structure) wave problem with $p = 3$ - Gain in computation time of the splitting procedure over the semi-implicit scheme on the four finest meshes, $\gamma = \gamma^{\min}$, $k \in \{1, 2, 3\}$, equal- and mixed-order settings.

scheme, but the gain (ratio of execution times) also increases as the mesh is refined. The gain on the four finest meshes is larger than 2 and reaches 6 for polynomial orders (1,1) and (2,2) when a direct solver is employed in the semi-implicit scheme.



(a) $(1, 1)$ (b) $(2, 2)$ (c) $(3, 3)$

(d) $(2, 1)$ (e) $(3, 2)$ (f) $(4, 3)$

Figure 8: Nonlinear (p-structure) wave problem with $p = 3$ - Comparison of execution times of a single time step with splitting procedure and semi-implicit scheme (direct or iterative linear solvers), $\gamma \in \{\gamma^{\min}, 2\gamma^{\min}, 5\gamma^{\min}\}$, $k \in \{1, 2, 3\}$, equal- and mixed-order settings.

We now compare the discrete energy error as a function of the execution time for various polynomial orders using either the HHO method or conforming finite elements for space discretization. We consider $k \in \{0, 1, 2\}$ for HHO in both mixed- and equal-order settings, and $k \in \{1, 2\}$ for finite elements. Based on the above results, we only consider the splitting procedure for HHO simulations, and we choose the best value of γ in terms of computation time, i.e. $\gamma = \gamma^{\min}$ in the equal-order setting and $\gamma = 5\gamma^{\min}$ in the mixed-order setting. Figure 9 first highlights the same behavior as in the linear case when comparing the different HHO orders: higher orders on coarse meshes are more computationally efficient than lower orders on finer meshes. Another salient point is the comparison between mixed- and equal-order settings. Mixed-order settings turns out to be (much) more efficient. For instance, for the same error, the splitting procedure with polynomial

orders (3,2) is ten times faster than the splitting procedure with polynomial orders (2,2). Furthermore, P1 finite elements are more efficient than the lowest equal-order HHO method, broadly equivalent to the lowest mixed-order HHO method, and less efficient than the higher-order HHO methods. The efficiency of P2 finite elements is between the first- and second-order HHO settings and quite close to HHO with polynomial orders (2,1). Thus, for a given target error or time budget, high-order HHO with splitting remains the most effective option.

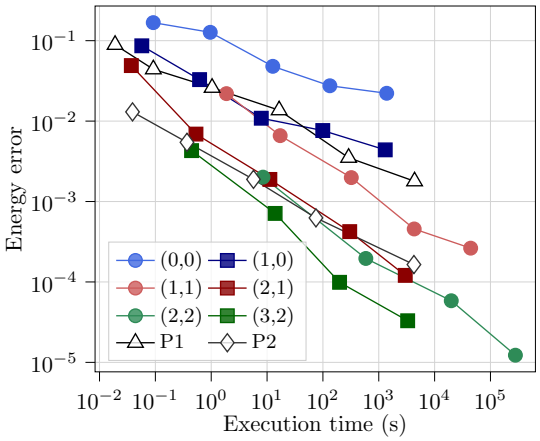


Figure 9: Nonlinear (p-structure) wave problem with $p = 3$ - Discrete energy error as a function of execution time, HHO with $(k, \gamma) \in \{(0, 2), (1, 10), (2, 20)\}$ in the equal-order setting and $(k, \gamma) \in \{(0, 25), (1, 100), (2, 100)\}$ in the mixed-order setting, P1 and P2 finite elements.

The above experiments have so far focused on the case $p = 3$ corresponding to mild nonlinearities. We now investigate other values of p. The same computations as in Figure 8 are performed, but only for $k = 1$. Other polynomial orders yield similar results and are not shown for brevity. The considered values of γ are $\{\gamma^{\min}, 2\gamma^{\min}, 5\gamma^{\min}\}$, unless they are larger than 150. This limit is set to avoid numerical errors induced by larger values of γ. The computational setting is otherwise unchanged. Results are reported in Figure 10. The conclusions are similar to the mildly nonlinear case $p = 3$ indicating that the variation in nonlinearity as quantified by p does not impact the performance of the splitting procedure in the equal-order setting. In the mixed-order setting, $p < 2$ seems to be more favorable to the splitting procedure, whereas with $p > 2$, the gain in computation time is less pronounced.

In conclusion, this p-structure test case highlights the benefits of the splitting procedure on fine meshes compared to the semi-implicit scheme. The results also indicate that, when working on fine meshes, the choice of γ is not sensitive. Indeed, as long as $\gamma > \gamma^{\min}$ and in a reasonable range, here $[\gamma^{\min}, 150]$, the error and the computation time are only marginally affected by the choice of γ. Since $\gamma^{\min}$ can be computed on coarse meshes and extrapolated on finer meshes, the value of γ is easy to be set in practice.

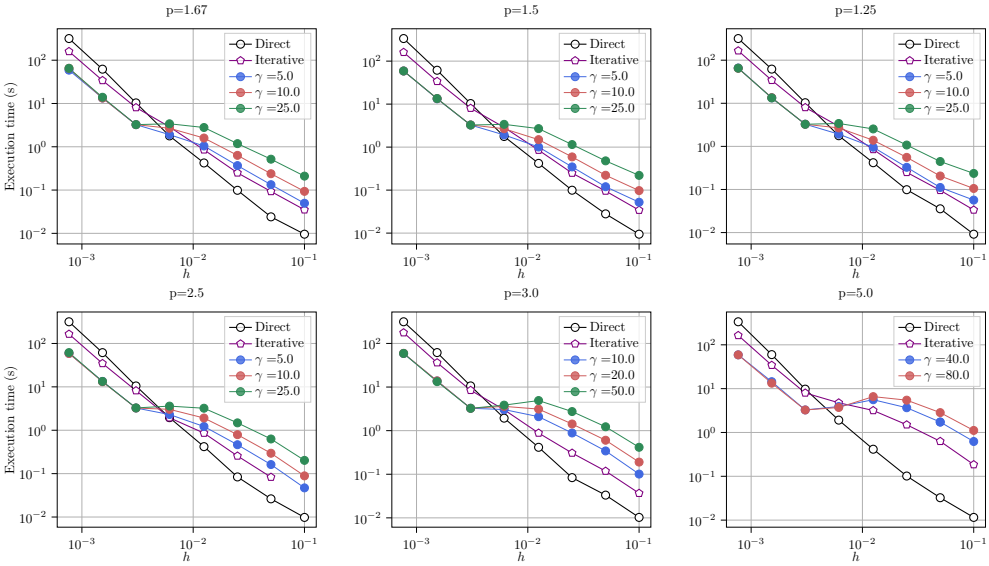
5.3 Nonlinear 2D vibrating membrane

We now consider a model for a 2D vibrating membrane inspired from [14], where a one-dimensional string is considered with two unknowns, the transversal and the longitudinal displacements. Here, we neglect the longitudinal displacement, which leaves us with only the transversal displacement, u. This leads to the following nonlinear wave equation (with zero source term):

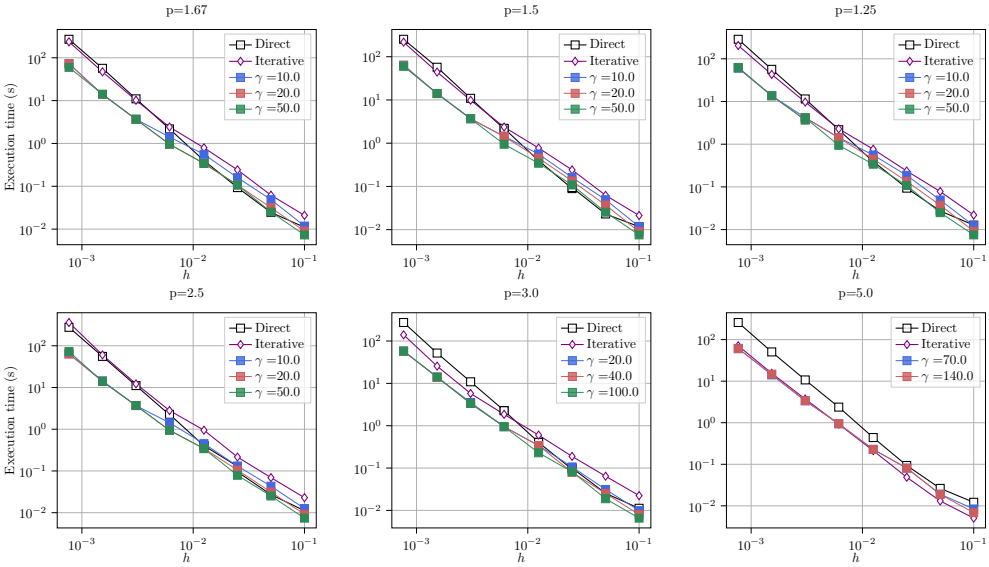
$$\partial_t^2 u - \nabla \cdot \big(\mu(\nabla u)\nabla u\big) = 0, \quad \text{in } \Omega, \forall t \in J, \tag{53}$$

with $\mu : \mathbb{R}^2 \to \mathbb{R}$ such that

$$\mu(g) = 1 - \alpha \frac{1}{\sqrt{|g|^2 + 1}}, \quad \forall g \in \mathbb{R}^2, \forall \alpha \in [0, 1). \tag{54}$$



(a) (1,1)



(b) (2,1)

Figure 10: Nonlinear (p-structure) wave problem with $p \in \{1.25, 1.5, 1.67, 2.5, 3, 5\}$ - Comparison of execution times for a single time step of the splitting procedure and of the semi-implicit scheme (with either a direct or an iterative solver), polynomial orders $(1, 1)$ and $(2, 1)$.

The nonlinear function μ is also considered in the literature in the context of mean-curvature flows. The associated Hamiltonian writes $\mathcal{H}^\alpha(g) := \frac{1}{2}|g|^2 - \alpha\left[\sqrt{|g|^2+1} - 1\right]$ for all $g \in \mathbb{R}^2$, and equation (53) rewrites $\partial_t^2 u - \nabla \cdot (\nabla_g \mathcal{H}^\alpha(\nabla u)) = 0$. If $\alpha = 0$, (53) is equivalent to the linear acoustic wave equation. Moreover, if $\alpha \neq 0$ and if the gradient ∇u becomes very large, the nonlinear part of μ becomes negligible with respect to the linear part. Thus, the most nonlinear behavior is expected for small deformations and α close to one. Typically, $\alpha = 0.8$ is considered to lead to a mildly nonlinear behavior, and $\alpha = 0.99$ to a strongly nonlinear behavior. We consider $\Omega := (0, 1)^2$, and homogeneous Dirichlet boundary conditions on the displacement u

are enforced. We consider a zero initial condition for u and an initial velocity

$$v_0(x,y) := e^{-\pi^2 f_p^2 r(x,y)^2}, \quad \text{with } r(x,y) := (0.5-x)^2 + (0.5-y)^2, \quad f_p := 3.33, \tag{55}$$

which simulates an impact at the center of the domain. The solutions at three sensors located at the points $(0.167, 0.5)$, $(0.333, 0.333)$ and $(0.5, 0.5)$ are displayed in Figure 11 over the time interval $[0,1]$ for $\alpha \in \{0, 0.8, 0.99\}$. As α increases, the wave propagation is slower and the wave front is sharper.

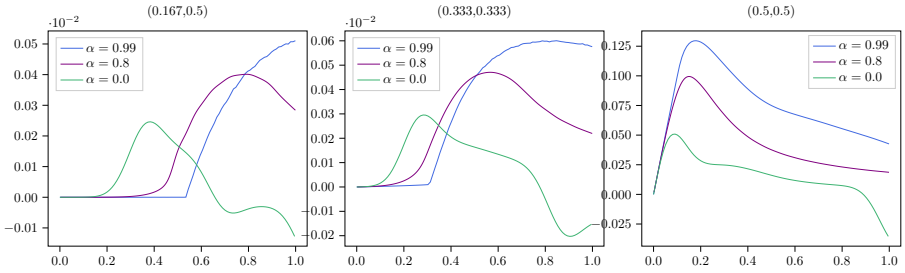


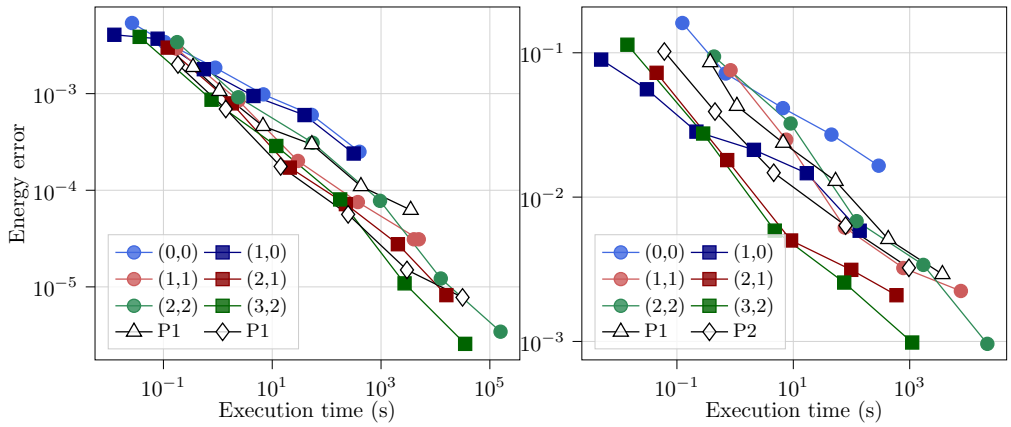
Figure 11: Vibrating membrane with $\alpha \in \{0, 0.8, 0.99\}$ - Reference solution at sensors placed at $(0.167, 0.5)$, $(0.333, 0.333)$ and $(0.5, 0.5)$ over the time interval $[0,1]$.

Figure 12 reports the discrete energy error on the displacement as a function of the computation time for P1 and P2 finite elements as well as the HHO method with polynomial orders $k \in \{0,1,2\}$ in the mixed- and equal-order settings. This error is computed as above, but using now 120 sensors positioned in the triangle $\{x \in [0, 0.5], y \in [x, 0.5]\}$ using the points with barycentric coordinates $(\frac{i_1}{6}, \frac{i_2}{6}, \frac{i_3}{6})$ with $i_1 + i_2 + i_3 = 6$. More points are considered in this test case in order to precisely capture the wave propagation. The same unstructured mesh sequence as for the p-structure test case is considered. The optimal value of γ is used for each polynomial order, namely $\gamma = \gamma^{\min}$ in the equal-order setting and $\gamma = 5\gamma^{\min}$ in the mixed-order setting ($\gamma^{\min}$ being as before the smallest possible integer value of γ observed experimentally that leads to a converging splitting procedure). Table 9 reports the value of $\gamma^{\min}$ and the optimal value of γ used in Figure 12. In the mildly nonlinear case, $\alpha = 0.8$, the curve associated with P1 finite elements is between those associated with the lowest- and the first-order HHO methods. Moreover, the first- and second-order HHO methods and P2 finite elements are almost equivalent in terms of error as a function of the execution time. In the strongly nonlinear case, $\alpha = 0.99$, P1 finite elements are faster than the HHO method with polynomial order (0,0) and deliver a similar performance to that of the other equal-order HHO methods. Moreover, mixed-order settings are (much) faster than equal-order settings and than P2 finite elements. The worse efficiency of the splitting in the equal-order setting can be explained by the results of Figure 13, which displays the mean number of splitting iterations per time step for each polynomial order and $\alpha \in \{0.8, 0.99\}$ as a function of the mesh size h. In the equal-order setting, the number of iterations for $\alpha = 0.99$ is much larger than that for $\alpha = 0.8$, by up to a factor of 10. Instead, in the mixed-order setting, the number of iterations at each time step remains smaller than 10 for both values of α. To sum up, this experiment shows that the HHO method with splitting and high polynomial orders is more efficient than P1 finite elements and better or equivalent to P2 finite elements in the mildly nonlinear case. In the strongly nonlinear case, the HHO method with polynomial orders (2,1) and (3,2) is much faster than P2 finite elements. Hence, in this experiment as well, the mixed- and high-order HHO method is the most efficient choice.

	$(0,0)$	$(1,1)$	$(2,2)$	$(1,0)$	$(2,1)$	(3,2)
$\alpha = 0.8$, ($\gamma^{\min}$, optimal value of γ)	(2,2)	(4,4)	(6,6)	(3,20)	(5,25)	(8,40)
$\alpha = 0.99$, ($\gamma^{\min}$, optimal value of γ)	(2,2)	(4,4)	(6,6)	(3,20)	(5,25)	(8,40)

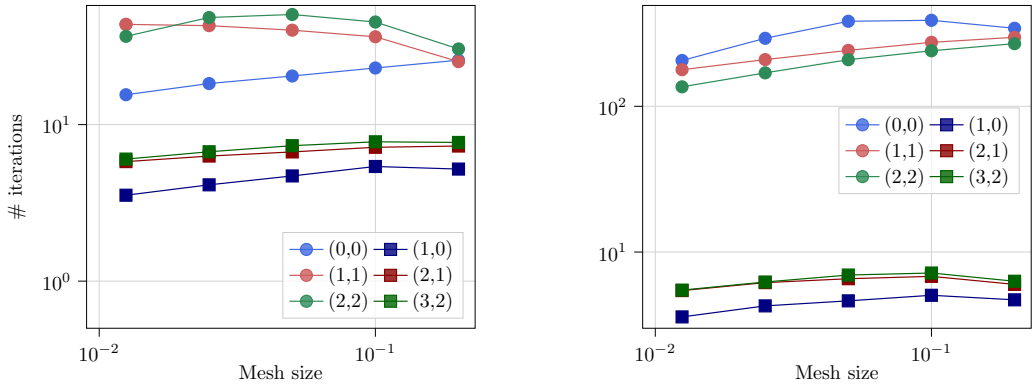
Table 9: Vibrating membrane with $\alpha \in \{0.8, 0.99\}$ - $\gamma^{\min}$ defined as the smallest integer allowing the splitting procedure to converge, and optimal value (in computation time) of γ.

Remark 5.1 (Aitken acceleration)**.** Considering the large number of iterations needed for the splitting procedure to converge in the equal-order setting, it can be interesting to use acceleration techniques to



(a) $\alpha = 0.8$ (b) $\alpha = 0.99$

Figure 12: Vibrating membrane with $\alpha \in \{0.8, 0.99\}$ - Discrete energy error as a function of the computation time, P1 finite elements compared to HHO with $k \in \{0, 1, 2\}$, mixed- and equal-order settings with splitting (only the best value of γ is displayed).



(a) $\alpha = 0.8$ (b) $\alpha = 0.99$

Figure 13: Vibrating membrane with $\alpha \in \{0.8, 0.99\}$ - Mean number of iterations per time step for polynomial orders $k \in \{0, 1, 2\}$, mixed- and equal-order settings, with splitting (only the best value of γ is displayed).

reduce the computational cost. We can consider for instance the Δ^2 Aitken acceleration introduced in [4], see, e.g., [38] for the implementation with vector-valued unknowns. Recalling that $u_{\mathcal{F}}^{n,m}$ denotes the solution at iteration m of the splitting procedure and at time iteration n, we define $\Delta u_{\mathcal{F}}^{n,m} := u_{\mathcal{F}}^{n,m+1} - u_{\mathcal{F}}^{n,m}$, $\Delta^2 u_{\mathcal{F}}^{n,m} := u_{\mathcal{F}}^{n,m+1} - 2u_{\mathcal{F}}^{n,m} + u_{\mathcal{F}}^{n,m-1}$, and

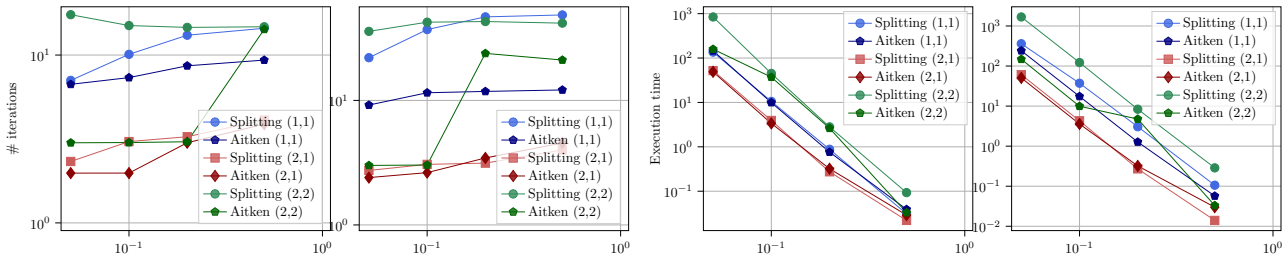
$$\tilde{u}_{\mathcal{F}}^{m} := u_{\mathcal{F}}^{n,m} - \frac{(\Delta u_{\mathcal{F}}^{n,m})^2}{\Delta^2 u_{\mathcal{F}}^{n,m}}. \tag{56}$$

The sequence of $(\tilde{u}_{\mathcal{F}}^{m})_{m \geqslant 0}$ converges faster to the same limit $u_{\mathcal{F}}^{n+1}$ as $(u_{\mathcal{F}}^{n,m})_{m \geqslant 0}$ as long as

$$\lim_{m \to \infty} \frac{u_{\mathcal{F}}^{n,m+1} - u_{\mathcal{F}}^{n+1}}{u_{\mathcal{F}}^{n,m} - u_{\mathcal{F}}^{n+1}} = \lambda \neq 1. \tag{57}$$

Considering the same membrane setting as above, Figures 14a and 14b show the mean number of iterations per time step and Figures 14c and 14d show the computation time, both with and without Aitken's acceleration. The optimal value of γ is the one from Table 9. As expected, Aitken's acceleration is more effective when

the number of splitting iterations is large, i.e. in the equal-order setting and the strongly nonlinear case ($\alpha = 0.99$). Instead, the gain is quite moderate in the mixed-order setting where the number of iterations is very small (less than 4). The equal-order setting with (1,1) benefits from Aitken's acceleration only in the strongly nonlinear case for the same reason. In this case, the splitting procedure with Aitken's acceleration is 30% faster than the splitting procedure without acceleration. In the equal-order setting with polynomial orders (2,2), Aitken's acceleration turns out to be very efficient since the number of iterations is large and does not decrease with mesh refinement. In this case, the gain in computation time is a factor of 5 for the mildly nonlinear case ($\alpha = 0.8$) and more than 10 in the strongly nonlinear case ($\alpha = 0.99$) and on the finest mesh. In conclusion, Aitken's acceleration is an effective tool that helps improve the performance of the equal-order setting in the strongly nonlinear case.



(a) $\alpha = 0.8$ (b) $\alpha = 0.99$ (c) $\alpha = 0.8$ (d) $\alpha = 0.99$

Figure 14: Vibrating membrane with $\alpha \in \{0.8, 0.99\}$ - Mean number of iterations per time step and execution time, polynomial orders (1,1), (2,1), (2,2), splitting procedure with and without Aitken's acceleration, optimal value of γ.

5.4 General conclusions from the numerical tests

The numerical experiments illustrated the effectivity of the splitting procedure. There are four points deserving to be put forward.

1. The behavior of the equal- and mixed-order is different. The mixed-order setting is more efficient than the equal-order setting due to the reduced number of splitting iterations to achieve convergence (Figure 9 for the p-structure case and Figure 12 for the vibrating membrane).

2. The tuning of the parameter γ does not bring any significant difficulty. Indeed, both linear and nonlinear experiments show that, for a value of γ smaller than 100, the quality of the solution is not impacted. All experiments show minimal values of γ ($\gamma^\star$ in the linear case where it can be computed and its approximation $\gamma^{\min}$ in the nonlinear case) smaller than 100. Moreover, the results of Figures 7, 8 and 10 show that, on fine meshes, the value of γ does not impact the computation time. On coarse meshes, both linear and nonlinear experiments give the same optimal choice: $\gamma = \gamma^{\min}$ in the equal-order setting and $\gamma \in [3\gamma^{\min}, 5\gamma^{\min}]$ in the mixed-order setting.

3. In order to increase the accuracy of the solution, it is more efficient to increase the polynomial order than to refine the mesh (see Figure 5 in the linear case and Figures 9 and 12 in the nonlinear case).

4. For nonlinear problems, the splitting procedure combined with a mixed-order setting is always faster than the semi-implicit scheme. The gain in computation time depends on the nonlinearity and increases with mesh refinement. In the equal-order setting, the splitting procedure becomes more efficient than the semi-implicit scheme on refined meshes. Moreover, the HHO method with mixed-order setting, splitting procedure and polynomial order $k \geqslant 1$ is always more efficient than the P1 finite element method. In most cases, equal-order computations are also faster. The splitting procedure with mixed-order and large k is therefore the most effective approach tested in the present work.

A Scaling of the stabilization

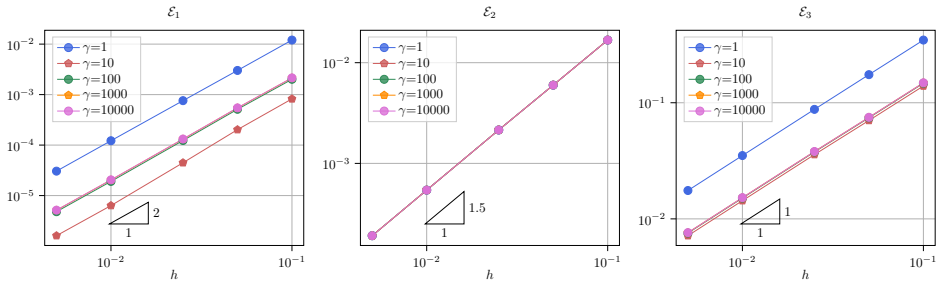
This appendix is dedicated to provide some further insight into the impact of the scaling of the stabilization parameter γ on the HHO method, and in particular to the case $\gamma \to \infty$. We start with a study on the linear diffusion equation, and measure the L^2- and H^1-errors for large values of γ on various meshes. Then, a study of the eigenvalues of the splitting matrices is also performed as $\gamma \to \infty$. Finally, we analyze the stability condition on the time step as a function of γ and its behavior as γ increases.

A.1 Static case

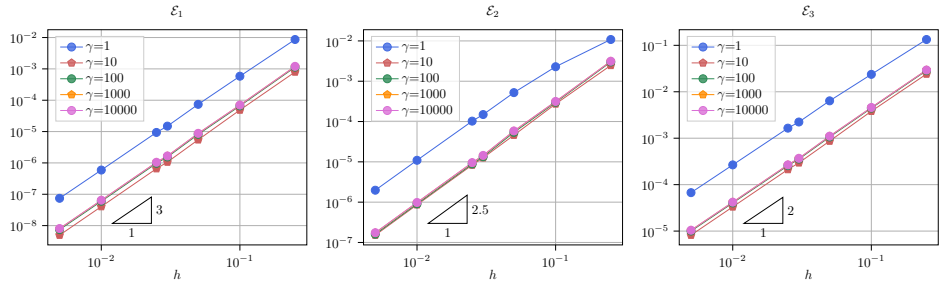
We consider the following problem: Find $u \in H_0^1(\Omega)$ such that $(\mu \nabla u, \nabla w)_\Omega = (f, w)_\Omega$ for all $w \in H_0^1(\Omega)$. The HHO discretization consists of finding $\hat{u}_h \in \widehat{\mathcal{U}}_{h,0}^{l,k}$ such that

$$b_h(\hat{u}_h, \hat{w}_h) + \sigma_h(\hat{u}_h, \hat{w}_h) = (f, w_\mathcal{T})_\Omega \,, \quad \forall \hat{w}_h \in \widehat{\mathcal{U}}_{h,0}^{l,k}. \tag{58}$$

We consider an analytical test case with $u(x,y) := \sin(\pi x)\sin(\pi y)$ obtained using $\mu = 1$ in the domain $\Omega := (0,1)^2$ and the source term $f(x,y) := 2\pi^2 \sin(\pi x)\sin(\pi y)$. We study the L^2-error on the cells, $\mathcal{E}_1 := (\Delta \mathsf{U}_\mathcal{C}^T M_{\mathcal{C}\mathcal{C}} \Delta \mathsf{U}_\mathcal{C})^{\frac{1}{2}}$, on the faces, $\mathcal{E}_2 := h^{-\frac{1}{2}} (\Delta \mathsf{U}_\mathcal{F}^T M_{\mathcal{F}\mathcal{F}} \Delta \mathsf{U}_\mathcal{F})^{\frac{1}{2}}$ (evaluated using the face mass matrix), as well as the (broken) H^1-error, $\mathcal{E}_3 := (\Delta \hat{\mathsf{U}}^T \mathcal{K}(\gamma) \Delta \hat{\mathsf{U}})^{\frac{1}{2}}$, computed using the complete stiffness matrix (this error thus includes the stabilization and is therefore affected by the value of γ). We test the values $\gamma \in \{1, 10, 100, 1000, 10000\}$. Figure 15 shows the three errors for the equal-order setting and $k \in \{0, 1\}$. Other



(a) (0,0)



(b) (1,1)

Figure 15: Errors $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ on the scalar Laplacian as a function of the mesh size for $\gamma \in \{1, 10, 100, 1000, 10000\}$, equal-order setting with $k \in \{0, 1\}$.

polynomial orders are not displayed for brevity, since the results are similar. For the L^2- and H^1-errors, the convergence rates are, as expected, $(k+2)$ and $(k+1)$, respectively. The main observation is that, for a fixed h, the error converges to some value as γ increases. In other words, the limit $\gamma \to \infty$ does not yield a diverging solution, but a suitable solution with optimal convergence rates. Interestingly, this limit solution is reached for a rather small value of γ since the error curves for $\gamma \geqslant 10$ almost overlap. Regarding the L^2-error on the faces, the behavior differs between $k = 0$ and the higher orders. For the higher orders, the behavior

is the same as for the other errors, whereas for the lowest order, the error does not change when γ increases. This indicates that the value on the faces is not affected by the scaling of the stabilization when $k = 0$.

The behavior as $\gamma \to \infty$ can be partially explained by a series expansion in γ. Let us plug the Ansatz $\hat{u}_h = \sum_{n=0}^{\infty} \hat{u}_h^n \gamma^{-n}$ into (58) and consider test functions having only nonzero face components. This yields

$$\gamma \sigma_h(\hat{u}_h^0, (0, w_{\mathcal{F}})) + b_h(\hat{u}_h^0, (0, w_{\mathcal{F}})) + \sigma_h(\hat{u}_h^1, (0, w_{\mathcal{F}})) + \ldots = 0, \quad \forall w_{\mathcal{F}} \in \mathcal{U}_{\mathcal{F},0}^k. \tag{59}$$

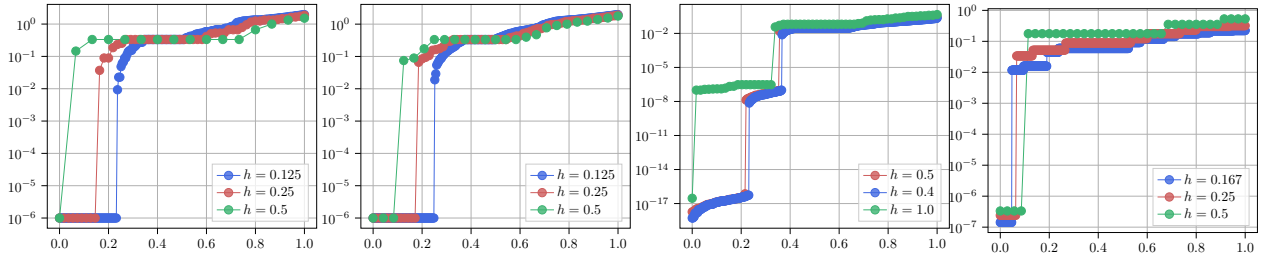
The dominant term in γ gives $\sigma_h(\hat{u}_h^0, w_{\mathcal{F}}) = 0$ for all $w_{\mathcal{F}} \in \mathcal{U}_{\mathcal{F},0}^k$. In the mixed-order setting, this condition reads

$$\sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} \eta_{TF}^{-1} (\Pi_F^k(u_T^0) - u_F^0, w_F) = 0, \quad \forall w_{\mathcal{F}} \in \mathcal{U}_{\mathcal{F},0}^k. \tag{60}$$

Letting $\mathcal{T}_F$ collect the one or two cells adjacent to a face F, this translates into the following direct expression of the face unknowns:

$$u_F^0 = \frac{1}{\#\mathcal{T}_F} \sum_{T \in \mathcal{T}_F} \Pi_F^k(u_T^0). \tag{61}$$

This means that, in the mixed-order setting, when $\gamma \to \infty$, the face unknowns on F are equal to the projection onto $\mathbb{P}_d^k(F)$ of the mean-value of the cell unknowns from the two adjacent cells if F is an interface and of the trace of the cell unknown if F is a boundary face. The equal-order setting does not lead to an explicit expression of the face unknowns in terms of the cell unknowns. Indeed, in this case, the condition $\sigma_h(\hat{u}_h, (0, w_{\mathcal{F}})) = 0$ for all $w_{\mathcal{F}} \in \mathcal{U}_{\mathcal{F},0}^k$ is not sufficient to determine the face unknowns in terms of the cell unknowns, because the spectrum of $\mathcal{S}_{\mathcal{F}\mathcal{F}}$ contains at least one zero eigenvalue as shown in Figure 16. The smallest eigenvalue is nonzero owing to round-off errors. Higher-order polynomials lead to the same behavior and are not displayed for brevity.



(a) Right triangles (b) Unstructured triangles (c) Unstructured tetrahedra (d) Cartesian hexahedra

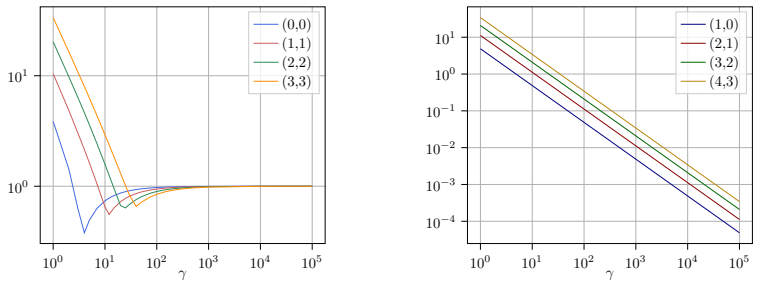
Figure 16: Eigenvalues of $\mathcal{S}_{\mathcal{F}\mathcal{F}}$ on different meshes for polynomial orders (0,0).

A.2 Spectral radius of the iteration matrix as $\gamma \to \infty$

Let us now focus on the spectral radius of $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{-1}\mathcal{B}_{\mathcal{F}\mathcal{F}}$ in the mixed-order setting and of $(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star})^{-1}(\mathcal{B}_{\mathcal{F}\mathcal{F}} + \mathcal{Z}_{\mathcal{F}\mathcal{F}})$ in the equal-order setting. In Section 4.4, a study of the value of $\gamma^{\star}$ as a function of the polynomial order and the mesh regularity has been performed. Figure 17 reports the spectral radius of the iteration matrix as $\gamma \to \infty$ on an unstructured triangular mesh with $h = 0.1$. In the mixed-order setting, the spectral radius of $\mathcal{S}_{\mathcal{F}\mathcal{F}}^{-1}\mathcal{B}_{\mathcal{F}\mathcal{F}}$ goes to zero, whereas the spectral radius of $(\mathcal{S}_{\mathcal{F}\mathcal{F}}^{\star})^{-1}(\mathcal{B}_{\mathcal{F}\mathcal{F}} + \mathcal{Z}_{\mathcal{F}\mathcal{F}})$ converges to one in the equal-order setting. This is the expected behavior.

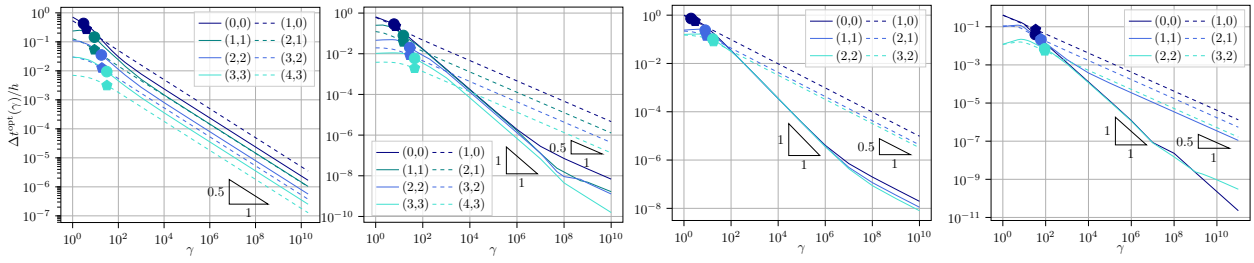
A.3 Stability condition as $\gamma \to \infty$

Equation (49) gives a formula to obtain the critical time step $\Delta t^{\mathrm{opt}}(\gamma)$ as a function of γ. Figure 18 reports the rate $\Delta t^{\mathrm{opt}}(\gamma)/h$ on 2D unstructured quadrangular and triangular meshes, and 3D Cartesian hexahedral and unstructured tetrahedral meshes. For each polynomial order, the value of $\gamma^{\star}$ is indicated by a bullet.



(a) Equal-order (b) Mixed-order

Figure 17: Spectral radius of the iteration matrix as a function of γ, equal-order setting (left) and mixed-order setting (right) on an unstructured triangular mesh with $h = 0.1$.



(a) Quadrangles (b) Triangles (c) Hexahedra (d) Tetrahedra

Figure 18: Ratio $\frac{\Delta t^{\mathrm{opt}}(\gamma)}{h}$ as a function of γ on 2D and 3D unstructured meshes.

As $\gamma \to \infty$, $\Delta t^{\mathrm{opt}}(\gamma)$ go to zero as $\frac{1}{\sqrt{\gamma}}$. Thus, taking $\gamma \gg \gamma^\star$ is not optimal, since this entails very small time steps. Fortunately, Table 1 has already shown that the value of $\gamma^\star$ need not be too large. The fact that $\Delta t^{\mathrm{opt}}(\gamma) \to 0$ as $\gamma \to \infty$ is easily proven using the min-max theorem for the largest eigenvalue. Indeed, since $\mathcal{B}$ is a positive matrix, we have

$$\rho(\mathcal{D}(\gamma)) = \max_{X=(X_{\mathcal{T}},X_{\mathcal{F}})\in\widehat{\mathcal{U}}_{h,0}^{l,k}} \frac{X^T\mathcal{A}X}{X_{\mathcal{T}}\mathcal{M}X_{\mathcal{T}}} \geqslant \max_{X_{\mathcal{T}}\in\mathcal{U}_{\mathcal{T}}^{l}} \frac{X_{\mathcal{T}}^T\mathcal{A}_{\mathcal{T}\mathcal{T}}X_{\mathcal{T}}}{X_{\mathcal{T}}\mathcal{M}X_{\mathcal{T}}} \geqslant \gamma\rho(\mathcal{M}^{-1}\mathcal{S}_{\mathcal{T}\mathcal{T}}).$$

This shows that $\Delta t^{\mathrm{opt}}(\gamma) \leqslant \frac{2}{\sqrt{\gamma\rho(\mathcal{M}^{-1}\mathcal{S}_{\mathcal{T}\mathcal{T}})}} \underset{\gamma\to\infty}{\sim} \frac{1}{\sqrt{\gamma}}$. All the mixed-order curves in Figure 18 show the expected asymptotic behavior. For the equal-order curves, this asymptotic behavior is recovered for the larger values γ, but we also observe that there is a large range of values of γ for which $\Delta t^{\mathrm{opt}}(\gamma) \simeq C\gamma^{-1}$, where C is some constant depending on the polynomial order and the mesh regularity. For triangular meshes with $k = 3$ and for hexahedral meshes with $k = 0$, the asymptotic rate $\gamma^{-\frac{1}{2}}$ is not even reached for $\gamma = 10^{10}$.

References

[1] M. Abbas, A. Ern, and N. Pignet. Hybrid high-order methods for finite deformations of hyperelastic materials. *Comput. Mech.*, 62(4):909–928, 2018.

[2] M. Abbas, A. Ern, and N. Pignet. A hybrid high-order method for finite elastoplastic deformations within a logarithmic strain framework. *Int. J. Numer. Methods Eng.*, 120(3):303–327, 2019.

[3] M. Abbas, A. Ern, and N. Pignet. A hybrid high-order method for incremental associative plasticity with small deformations. *Comp. Meth. Appl. Mech. Eng.*, 346:891–912, 2019.

[4] A. C. Aitken. IV.—Studies in Practical Mathematics. V. On the Iterative Solution of a System of Linear Equations. *P. Roy. Soc. Edinb. A*, 63(1):52–60, 1950.

[5] P. F. Antonietti, I. Mazzieri, M. Muhr, V. Nikolić, and B. Wohlmuth. A high-order discontinuous Galerkin method for nonlinear sound waves. *J. Comput. Phys.*, 415:109484, 2020.

[6] B. Ayuso de Dios, K. Lipnikov, and G. Manzini. The nonconforming virtual element method. *ESAIM Math. Mod. Numer. Anal.*, 50(3), 2016.

[7] D. Boffi, M. Botti, and D. A. Di Pietro. A nonconforming high-order method for the Biot problem on general meshes. *SIAM J. Sci. Comput.*, 38(3):A1508–A1537, 2016.

[8] M. Botti, D. A. Di Pietro, and P. Sochala. A hybrid high-order method for nonlinear elasticity. *SIAM J. Numer. Anal.*, 55(6):2687–2717, 2017.

[9] E. Burman, O. Duran, and A. Ern. Hybrid high-order methods for the acoustic wave equation in the time domain. *Commun. Appl. Math. Comput. (CAMC)*, 4(2):597–633, 2022.

[10] E. Burman, O. Duran, and A. Ern. Unfitted hybrid high-order methods for the wave equation. *Comp. Meth. Appl. Mech. Eng.*, 389:114366, 2022.

[11] E. Burman, O. Duran, A. Ern, and M. Steins. Convergence analysis of hybrid high-order methods for the wave equation. *J. Sci. Comput.*, 87(3):91, 2021.

[12] E. Bécache, P. Joly, and C. Tsogka. An analysis of new mixed finite elements for the approximation of wave propagation problems. *SIAM J. Numer. Anal.*, 37(4):1053–1084, 2000.

[13] A. Cangiani, Z. Dong, E. H. Georgoulis, and P. Houston. *hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes.* SpringerBriefs in Mathematics. Springer, Cham, 2017.

[14] J. Chabassier and P. Joly. Energy preserving schemes for nonlinear hamiltonian systems of wave equations: Application to the vibrating piano string. *Comp. Meth. Appl. Mech. Eng.*, 199(45):2779–2795, 2010.

[15] M. Cicuttin, A. Ern, and N. Pignet. *Hybrid high-order methods. A primer with application to solid mechanics.* SpringerBriefs in Mathematics. Springer, Cham, 2021.

[16] B. Cockburn. Static condensation, hybridization, and the devising of the HDG methods. In *Building bridges: connections and challenges in modern approaches to numerical partial differential equations*, pages 129–177. Springer, 2016.

[17] B. Cockburn, D. A. Di Pietro, and A. Ern. Bridging the hybrid high-order and hybridizable discontinuous Galerkin methods. *ESAIM Math. Mod. Numer. Anal*, 50(3):635–650, 2016.

[18] B. Cockburn, Z. Fu, A. Hungria, L. Ji, M. A. Sánchez, and F.-J. Sayas. Stormer-Numerov HDG methods for acoustic waves. *J. Sci. Comput.*, 75(2):597–624, 2018.

[19] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. on Numer. Anal.*, 47(2):1319–1365, 2009.

[20] G. Cohen, P. Joly, J. E. Roberts, and N. Tordjman. Higher order triangular finite elements with mass lumping for the wave equation. *SIAM J. Numer. Anal.*, 38(6):2047–2078, 2001.

[21] Gary C. Cohen. *Higher-order numerical methods for transient wave equations.* Scientific Computation. Springer-Verlag, Berlin, 2002. With a foreword by R. Glowinski.

[22] D. A. Di Pietro and J. Droniou. *The Hybrid High-Order method for polytopal meshes.* Modeling, Simulation and Applications series. Springer International Publishing, 2020.

[23] D. A. Di Pietro and A. Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*, volume 69. Springer, 2012.

[24] D. A. Di Pietro and A. Ern. A hybrid high-order locking-free method for linear elasticity on general meshes. *Comp. Meth. Appl. Mech. Eng.*, 283:1–21, 2015.

[25] D. A. Di Pietro, A. Ern, and S. Lemaire. An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators. *Comput. Methods Appl. Math.*, pages 461–472, 2014.

[26] Z. Dong and A. Ern. Hybrid high-order and weak Galerkin methods for the biharmonic problem. *SIAM J. Numer. Anal.*, 60(5):2626–2656, 2022.

[27] A. Ern and J.-L. Guermond. *Finite Elements I: Approximation and Interpolation*, volume 72 of *Texts in Applied Mathematics*. Springer, Cham, 2021.

[28] F. X. Giraldo and M. A. Taylor. A diagonal-mass-matrix triangular-spectral-element method based on cubature points. *J. Engrg. Math.*, 56(3):307–322, 2006.

[29] M. J. Grote, A. Schneebeli, and D. Schötzau. Discontinuous Galerkin finite element method for the wave equation. *SIAM J. Numer. Anal.*, 44(6):2408–2431, 2006.

[30] Y. Huang, J. Li, and D. Li. Developing weak Galerkin finite element methods for the wave equation. *Numer. Methods Partial Differ. Equations*, 33(3):868–884, 2017.

[31] P. Jana, N. Kumar, and B. Deka. A systematic study on weak Galerkin finite-element method for second-order wave equation. *Comput. Appl. Math.*, 41(8):Paper No. 359, 25, 2022.

[32] P. Joly. Variational methods for time-dependent wave propagation problems. In *Topics in computational wave propagation*, volume 31 of *Lect. Notes Comput. Sci. Eng.*, pages 201–264. Springer, Berlin, 2003.

[33] M. Kronbichler, S. Schoeder, C. Müller, and W. A. Wall. Comparison of implicit and explicit hybridizable discontinuous galerkin methods for the acoustic wave equation. *Int. J. Numer. Methods Eng.*, 106(9):712–739, 2016.

[34] C. Lehrenfeld. *Hybrid Discontinuous Galerkin methods for solving incompressible flow problems*. PhD thesis, Rheinisch-Westfalischen Technischen Hochschule (RWTH), Aachen, 2010.

[35] C. Lehrenfeld and J. Schöberl. High order exactly divergence-free hybrid discontinuous Galerkin methods for unsteady incompressible flows. *Comp. Meth. Appl. Mech. Eng.*, 307:339–361, 2016.

[36] N. C. Nguyen and J. Peraire. Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. *J. Comput. Phys.*, 231(18):5955–5988, 2012.

[37] N. C. Nguyen, J. Peraire, and B. Cockburn. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *J. Comput. Phys.*, 230, 2011.

[38] I. Ramière and T. Helfer. Iterative residual-based vector methods to accelerate fixed point iterations. *Comput. Math. Appl.*, 70(9):2210–2226, Nov 2015.

[39] M. A. Sánchez, C. Ciuca, N. C. Nguyen, J. Peraire, and B. Cockburn. Symplectic hamiltonian HDG methods for wave propagation phenomena. *J. Comput. Phys.*, 350:951–973, 2017.

[40] M. Stanglmeier, N. C. Nguyen, J. Peraire, and B. Cockburn. An explicit hybridizable discontinuous Galerkin method for the acoustic wave equation. *Comp. Meth. Appl. Mech. Eng.*, 300:748–769, 2016.

[41] J. Wang and X. Ye. A weak Galerkin finite element method for second-order elliptic problems. *J. Comput. Appl. Math.*, 241:103–115, 2013.