Adv Comput Math (2020) 46: 45
https://doi.org/10.1007/s10444-020-09785-z

High-order cut finite elements for the elastic wave equation

Simon Sticko[1] · Gustav Ludvigsson[2] · Gunilla Kreiss[2]

Received: 9 January 2019 / Accepted: 19 March 2020 /
Published online: 6 May 2020

Abstract

A high-order cut finite element method is formulated for solving the elastic wave equation. Both a single domain problem and an interface problem are treated. The boundary or interface is allowed to cut through the background mesh. To avoid problems with small cuts, stabilizing terms are added to the bilinear forms corresponding to the mass and stiffness matrix. The stabilizing terms penalize jumps in normal derivatives over the faces of the elements cut by the boundary/interface. This ensures a stable discretization independently of how the boundary/interface cuts the mesh. Nitsche's method is used to enforce boundary and interface conditions, resulting in symmetric bilinear forms. As a result of the symmetry, an energy estimate can be made and optimal order a priori error estimates are derived for the single domain problem. Finally, numerical experiments in two dimensions are presented that verify the order of accuracy and stability with respect to small cuts.

Keywords Elastic · Wave · Cut · Immersed · Interface

Mathematics Subject Classification 2010 65M60 · 65M85

Communicated by:Ivan Graham

This research was supported by the Swedish Research Council (Grant No. 2014-6088).

✉ Gustav Ludvigsson
gustav.ludvigsson@it.uu.se

Simon Sticko
simon@sticko.se

Gunilla Kreiss
gunilla.kreiss@it.uu.se

[1] Department of Mathematics and Mathematical Statistics, Umeå University,
901 87 Umeå, Sweden

[2] Division of Scientific Computing, Department of Information Technology, Uppsala University,
752 36 Uppsala, Sweden

1 Introduction

The time-dependent elastic wave equation is important in several applications. For example, materials in the Earth's crust can be modeled as linearly elastic, and earthquakes give rise to seismic waves that propagate over large distances. Other applications include non-destructive testing and propagation of waves in beams and other solid structures. There are several types of elastic waves, some propagate through the bulk (s-waves and p-waves), some along surfaces (Rayleigh waves), and others along material interfaces and layers (Stonley and Love waves). In all these examples, material surfaces and interfaces are of profound importance. The dynamics of surface and interface waves are directly influenced by for example the curvature, while bulk waves are influenced by the reflection, transmission, and conversion to other types of waves, that occur at surfaces and material interfaces. For more information on elastic waves, see [2].

High-order accurate methods with explicit time-stepping are especially attractive when solving the time-dependent elastic wave equation. The reason is that high-order methods, in general, have lower work per dispersion error [15], and that information propagates with finite speed, which opens for explicit time-stepping with Courant numbers $ck/h = \mathcal{O}(1)$. Here, c is the wave speed, k is the time step, and h is some measure of the grid size. Seismic waves typically propagate over large distances, which means that simulations of them are severely affected by dispersion errors. Several efficient high-order methods exist. Examples of such methods are discontinuous Galerkin (dG) methods [10, 24] and finite difference methods with the summation by parts (SBP) property [4, 11]. Both of these methods use a computational mesh that conforms to the geometry of the domain. Elastic waves often propagate in media with complicated surface and material interface geometry. High accuracy at surfaces and interfaces is essential for surface and interface waves, but also to accurately capture reflection, transmission, and conversion of bulk waves. However, creating a high-quality mesh that conforms to this geometry can be challenging and computationally expensive.

An alternative is to use a so-called immersed method. Here, boundaries and interfaces do not need to be aligned with the mesh. Thus, immersed methods can avoid complicated mesh generation, and it follows that this approach could potentially be computationally cheaper. This argument is strengthened when the geometry of a boundary or interface changes. This could, for example, be the case if the geometry is unknown and is hard or impossible to measure. One way to find the unknown geometry is to send waves toward it. The incoming waves get reflected, and by measuring the outgoing waves, it is possible to solve an inverse problem to compute the geometry. To solve the inverse problem, one would need to iterate over several different geometries. Since the geometry changes between iterations, a boundary conforming mesh could become very deformed, which would lead to a need for a potentially time-consuming re-meshing procedure. This type of inversion problem has been of interest for some time, see for example [29–31].

When developing an immersed method a few difficulties emerge. One difficulty is how to accurately take the geometry of the domain into account and apply boundary and interface conditions. Another difficulty is that arbitrarily small cuts between the elements in the mesh and the domain may occur. Without careful treatment, small cuts can introduce ill-conditioning of the discrete problem, spurious growth, or severely reduce the time step limit in a time-dependent setting.

Some previous works on immersed methods for the elastic wave equation have been done. A classical way of treating immersed boundary conditions was used in [32]. Here, finite differences are used on a staggered grid and the immersed interface is handled by simply using the local material parameter value in each grid point. In [34], this method is used for treating boundaries connected to vacuum. This is done by modeling the vacuum as having material parameters that are zero, except for the density which is set to a small but non-zero constant. This means that the position of the boundary is approximated with only first-order accuracy, and thus the overall accuracy of the method will be low. Another type of method was presented in [19] for handling interfaces between the acoustic and the elastic wave equations, and in [20] a similar method was used to handle boundary conditions for the elastic wave equation. Here, boundary/interface conditions are treated by extending the boundary/interface values outside of the domain using Taylor expansion. Both second- and fourth-order techniques are used. However, no theoretical stability or accuracy results are presented.

Several high-order immersed methods for other types of problems have been presented in the literature. Examples of such methods are the inverse Lax-Wendroff method [28], the difference potential method [3], and methods based on finite differences [1, 5]. However, as far as we know, none of these methods have been applied to the time-dependent elastic wave equation.

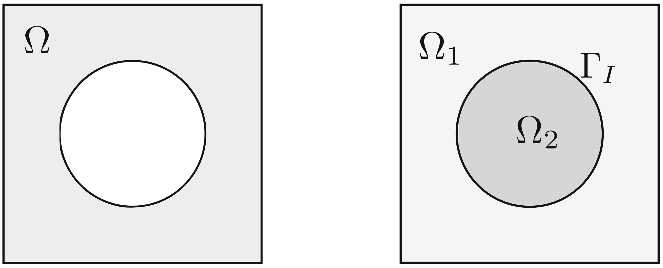
We will use another approach to achieve high accuracy, which is to develop a high-order method of Cut-FEM type. The Cut-FEM technique was originally developed for elliptic problems, see for example the review paper [8] and references therein. In Cut-FEM, Nitsche's method [23] is used to accurately apply boundary and interface conditions at locations that generally do not coincide with element faces. Discretization matrices are constructed by element-wise integration on the background mesh but only considering the regions overlapping with the physical domain. An important ingredient in the method is the handling of the ill-conditioning in the discrete problem, which is related to small cuts. Ghost stabilization, see [7, 9, 21], is a common way to handle this problem, also for higher order settings. It can be understood as enforcing additional continuity by penalizing jumps in derivatives. A significant difficulty in achieving high accuracy is generating sufficiently high-order quadrature rules for the cut elements. In fact, there are few examples in the literature of high-order Cut-FEM. Two of them are [13] and [18], where Stokes equation is considered using high-order piecewise polynomials, but only in the latter work is the method fully accurate for general geometries. There, the quadrature problem is handled by isoparametric mappings, while in the former cuts are assumed to be planar.

The contribution of this paper is an up to fourth-order accurate immersed method of Cut-FEM type for solving the time-dependent elastic wave equation. The spatial discretization builds on the work in [12], where the time-independent elastic equations were solved using the Cut-FEM technique, assuming a parametric piecewise linear description of boundaries and interfaces. In our method, we define the physical domain and interfaces implicitly by level-set functions, and the needed high-order quadrature rules for cut elements are constructed through an algorithm by Saye [25]. This algorithm is applicable to hyperrectangular elements and guarantees positive weights. Our approach is an extension to systems of the work in [26] and [27], where Cut-FEM for the time-dependent scalar wave equation is introduced, and extended to high-order, respectively. For time-dependent problems, the abovementioned ill-conditioning appears in both mass and stiffness matrices, and leads to severe time step limitations for explicit time-stepping, or even spurious temporal growth. To circumvent this, we use the ghost stabilization for both mass and stiffness matrices. Following [12], the stabilization ensures that the eigenvalues of the matrices are bounded from above and below, independently of how the boundary/interface cuts the mesh. In our case, this leads to a much-desired cut-independent CFL condition. This makes it possible to use explicit time-stepping with a time step scaling linearly with the un-cut element size. Furthermore, the method is provably stable and convergent in the semi-discrete setting. To the best of our knowledge, there is in the literature no other fourth-order accurate, immersed method for the time-dependent elastic wave equation with these properties. However, we note that with the ghost stabilization, the condition number of the mass matrix grows when the element order increases, which limits how far beyond fourth-order accuracy this approach can be extended without further modifications.

The present paper is organized as follows. In Section 2, the mathematical problems are stated. These are the elastic wave equation posed on a single domain and as an interface problem. This is followed by a description of the method in Section 3. In Section 4, we present a proof of convergence for the single domain problem and discuss how the proof can be modified to prove a corresponding result for the interface problem. In Section 5, we present numerical results on the order of accuracy and robustness with respect to small cuts. An example with more complicated geometric features illustrates the robustness and applicability of the approach in a dynamic setting. Finally, we end with a discussion in Section 6.

2 Model of the problem

We are interested in the elastic wave equation posed both on a single domain $\Omega \subset \mathbb{R}^d$ (Fig. 1a), and as an interface problem on a composite domain $\Omega = \Omega_1 \cup \Omega_2 \subset \mathbb{R}^d$



(a) Single domain (b) Composite domain for the interface problem

Fig. 1 Considered domains. **a** Single domain. **b** Composite domain for the interface problem

(Fig. 1b). The interface problem is appropriate when we have two materials in contact with each other, which occurs frequently in applications due to, for example, the layered structure of the Earth's crust. On the other hand, the single domain problem is relevant if we have an inclusion of empty space inside another material.

2.1 Single domain problem

Let n denote the outward unit normal to $\partial\Omega$, and assume that $\partial\Omega$ is partitioned such that $\partial\Omega = \Gamma^N \cup \Gamma^D$, with $\Gamma^N \cap \Gamma^D = \emptyset$. Let $T \in (0, \infty)$ denote the end time. The single domain problem reads as follows:

$$\begin{aligned}
\rho\ddot{u} &= \nabla \cdot \sigma(u) + f, && x \in \Omega, && t \in (0, T), && \text{(1a)}\\
\sigma(u) \cdot n &= g^N, && x \in \Gamma^N, && t \in (0, T), && \text{(1b)}\\
u &= g^D, && x \in \Gamma^D, && t \in (0, T), && \text{(1c)}\\
u &= u^0, && x \in \Omega, && t = 0, && \text{(1d)}\\
\dot{u} &= w^0, && x \in \Omega, && t = 0, && \text{(1e)}
\end{aligned}$$

where u is the displacement vector, ρ is the density, and σ is the stress tensor. We shall assume that Γ^D and Γ^N are sufficiently smooth. The boundary condition on Γ^N puts a constraint on the normal stress. If $g^N = 0$, the boundary condition corresponds to the standard boundary condition used at free surfaces. Furthermore, we assume that we are working with a linear, homogeneous, and isotropic material. When this is the case, the stress in the material is given by the following:

$$\sigma_{ij}(u) = 2\mu\epsilon_{ij}(u) + \lambda(\nabla \cdot u)\delta_{ij}, \tag{2}$$

where δ_{ij} is the Kronecker delta function and ϵ is the strain tensor defined as follows:

$$\epsilon_{ij}(u) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{3}$$

In (2), λ and μ are the Lamé parameters, which are material-dependent scalar constants.

2.2 Interface problem

Consider now an interface problem on the domain, Ω, illustrated in Fig. 1b. Let $i \in \{1, 2\}$ index each subdomain, Ω_i. We have two elastic materials, with material parameters ρ_i, λ_i, μ_i. In this case, the problem is given by the following:

$$\begin{aligned}
\rho_i\ddot{u}_i &= \nabla \cdot \sigma(u_i) + f_i, && x \in \Omega_i, && t \in (0, T), && \text{(4a)}\\
\sigma(u_i) \cdot n_i &= g_i^N, && x \in \Gamma_i^N, && t \in (0, T), && \text{(4b)}\\
u_i &= g_i^D, && x \in \Gamma_i^D, && t \in (0, T), && \text{(4c)}\\
[\![u]\!] &= 0, && x \in \Gamma_I, && t \in (0, T), && \text{(4d)}\\
[\![\sigma(u) \cdot n]\!] &= 0, && x \in \Gamma_I, && t \in (0, T), && \text{(4e)}\\
u_i &= u_i^0, && x \in \Omega_i, && t = 0, && \text{(4f)}\\
\dot{u}_i &= w_i^0, && x \in \Omega_i, && t = 0, && \text{(4g)}
\end{aligned}$$

where u_i is the displacement vector in material i and the stress and strain tensors are defined analogously to (2) and (3). We assume that Γ_i^D, Γ_i^N, and Γ^I are sufficiently smooth. Here, n_i is the outward normal to Ω_i and n is the normal pointing from Ω_2 to Ω_1 ($n = n_2$). $[\![\cdot]\!]$ defines the jump over the interface:

$$[\![u]\!] = u_2(x) - u_1(x), \quad x \in \Gamma_I. \tag{5}$$

Since we have several normals defined (n_1, n_2, and n), (4e) can be interpreted in two different ways. To avoid any confusion, we use the convention that the normal is fixed as follows:

$$[\![\sigma(u) \cdot n]\!] = \sigma(u_2) \cdot n - \sigma(u_1) \cdot n, \quad x \in \Gamma_I. \tag{6}$$

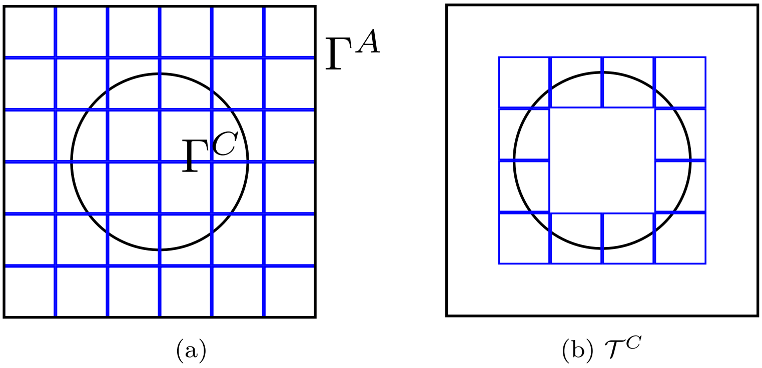
Remark 1 If boundary and/or interface conditions are compatible with a smooth solution it is, in principle, not necessary for the boundary or interface to be smooth. However, some parts of the methodology need to be modified to allow for such geometric non-smoothness. This is further discussed in Remark 5.

3 Numerical method

Let Ω be covered by a background mesh, $\mathcal{T}_B$, as in Fig. 2a. We shall only consider the case when the mesh consists of quadrilaterals that are squares and of the same size. Let h denote their side length. Let the boundary or interface be partitioned as illustrated in Fig. 2a. That is, for the single domain, we assume that $\partial\Omega = \Gamma^A \cup \Gamma^C$ (with $\Gamma^A \cap \Gamma^C = \emptyset$), where Γ^A is aligned with the boundary of the mesh while Γ^C cuts through it. Correspondingly for the interface problem, we assume that $\partial\Omega \cup \Gamma_I = \Gamma^A \cup \Gamma^C$. Let $\mathcal{T}^C$ denote the elements that are intersected by Γ^C:

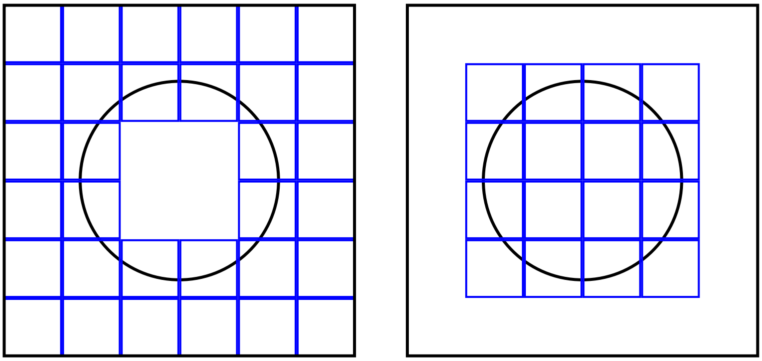
$$\mathcal{T}^C = \{T \in \mathcal{T}_B : T \cap \Gamma^C \neq \emptyset\}, \tag{7}$$

as illustrated in Fig. 2b.



(a) (b) $\mathcal{T}^C$

Fig. 2 **a** Parts of the boundary or interface that are aligned with, Γ^A, or immersed in the mesh, Γ^C. **b** Set of elements, $\mathcal{T}^C$, intersected by Γ^C



(a) $\mathcal{T}$, $\mathcal{T}_1$

(b) $\mathcal{T}_2$

Fig. 3 Smallest set of elements covering **a** Ω, Ω_1 and **b** Ω_2

Let $\mathcal{T}_i$, denote the smallest set of elements in the background mesh covering Ω_i, as illustrated in Fig. 3. In the following, we shall for various entities omit the index, i, when we consider the single domain problem. In particular, for the single domain problem, $\mathcal{T}$ is the smallest set of elements covering Ω. To be precise let

$$\mathcal{T}_i = \{T \in \mathcal{T}_B : T \cap \Omega_i \neq \emptyset\}. \tag{8}$$

Let $\Omega_i^\star$ denote the domain that $\mathcal{T}_i$ covers. That is, let

$$\Omega_i^\star = \bigcup_{T \in \mathcal{T}_i} T. \tag{9}$$

Now introduce the spaces

$$V_h^i = \{v \in [C^0(\Omega_i^\star)]^d : v|_T \in [Q_p(T)]^d,\ T \in \mathcal{T}_i\}. \tag{10}$$

Here, $Q_p(T)$ denotes the pth order Lagrange element, $p \geq 1$, with Gauss-Lobatto nodes over T. For high element orders, Gauss-Lobatto nodes result in a mass matrix with better properties than if equidistant nodes are used [14].

For the single domain problem, we solve for the solution $u_h \in V_h$, while for the interface problem, we solve for the pair $\{u_1, u_2\} \in V_h^1 \times V_h^2$. For the interface problem, this means that the degrees of freedom are doubled over the elements in the set $\mathcal{T}^C$. This doubling guarantees that the analytical solution can be approximated sufficiently well in our finite element space, so that we obtain the order of accuracy that we want. We would like to emphasize that the relevant solution, u_R, to the problem (4) is as follows:

$$u_R(x,t) = \begin{cases} u_1(x,t), & x \in \Omega_1, \\ u_2(x,t), & x \in \Omega_2, \end{cases} \tag{11}$$

and that

$$u_i(x,t), \quad x \in \Omega_i^\star \setminus \Omega_i, \tag{12}$$

is just a smooth extension of the solution in Ω_i to $\Omega_i^\star$.

Since the weak formulations for the single domain and the interface problem are very similar, we discuss their derivation more or less simultaneously. We shall use the following standard inner products:

$$(u, v)_\Omega = \int_\Omega uv d\Omega, \quad \langle u, v\rangle_\Gamma = \int_\Gamma uv d\Gamma, \tag{13}$$

where the subscripts indicate over which domain the integration takes part. If u or v in (13) are tensors, then contraction to a scalar is implied. Note that the angular brackets denote integration over a curve in 2D (or surface in 3D).

By multiplying (1a) or (4a) by a test function, integrating by parts and simplifying (for details see for example [17]), we get the following:

$$\begin{aligned}(\rho_i \ddot{u}_i, v_i)_{\Omega_i} &+2\mu_i(\epsilon(u_i), \epsilon(v_i))_{\Omega_i} + \lambda_i(\nabla\cdot u_i, \nabla\cdot v_i)_{\Omega_i} - \langle \sigma(u_i)\cdot n_i, v_i\rangle_{\partial\Omega_i\setminus\Gamma_i^N} \\ &= (f_i, v_i)_{\Omega_i} + \left\langle g_i^N, v_i\right\rangle_{\Gamma_i^N}, \quad \forall v_i \in V_h^i.\end{aligned} \tag{14}$$

Note that the Dirichlet boundary conditions are consistent with the following terms:

$$-\langle u_i, \sigma(v_i)\cdot n_i\rangle_{\Gamma_i^D} = -\left\langle g_i^D, \sigma(v_i)\cdot n_i\right\rangle_{\Gamma_i^D}, \tag{15}$$

$$2\mu_i\frac{\gamma_D}{h}\langle u_i, v_i\rangle_{\Gamma_i^D} = 2\mu_i\frac{\gamma_D}{h}\left\langle g_i^D, v_i\right\rangle_{\Gamma_i^D}. \tag{16}$$

$$\lambda_i\frac{\gamma_D}{h}\langle u_i\cdot n_i, v_i\cdot n_i\rangle_{\Gamma_i^D} = \lambda_i\frac{\gamma_D}{h}\left\langle g_i^D\cdot n_i, v_i\cdot n_i\right\rangle_{\Gamma_i^D}, \tag{17}$$

To enforce the boundary conditions by Nitsche's method, we add (15)–(17) to (14). Here, γ_D is a constant controlling how strongly the Dirichlet boundary condition is enforced. We now have the following:

$$(\rho_i\ddot{u}_i, v_i)_{\Omega_i} + a_i(u_i, v_i) - \langle\sigma(u_i)\cdot n_i, v_i\rangle_{\partial\Omega_i\setminus(\Gamma_i^D\cup\Gamma_i^N)} = L_i(v_i), \quad \forall v_i \in V_h^i, \tag{18}$$

where

$$a_i(u_i, v_i) = B_i(u_i, v_i) + D_i(u_i, v_i), \tag{19}$$

and

$$L_i(v_i) = (f_i, v_i)_{\Omega_i} + \left\langle g^N, v_i\right\rangle_{\Gamma_i^N} + L_i^D(v_i). \tag{20}$$

In (18), the term B_i corresponds to integration over the "bulk" as follows:

$$B_i(u_i, v_i) = 2\mu_i(\epsilon(u_i), \epsilon(v_i))_{\Omega_i} + \lambda_i(\nabla\cdot u_i, \nabla\cdot v_i)_{\Omega_i}, \tag{21}$$

and the terms D_i and L_i^D enforce the Dirichlet boundary condition over Γ_i^D:

$$\begin{aligned}D_i(u_i, v_i) &= -\langle\sigma(u_i)\cdot n_i, v_i\rangle_{\Gamma_i^D} \\ &\quad -\langle u_i, \sigma(v_i)\cdot n_i\rangle_{\Gamma_i^D} + \frac{\gamma_D}{h}\left(2\mu_i\langle u_i, v_i\rangle_{\Gamma_i^D} + \lambda_i\langle u_i\cdot n_i, v_i\cdot n_i\rangle_{\Gamma_i^D}\right),\end{aligned} \tag{22}$$

$$L_i^D(v_i) = -\left\langle g_i^D, \sigma(v_i)\cdot n_i\right\rangle_{\Gamma_i^D} + \frac{\gamma_D}{h}\left(2\mu_i\left\langle g_i^D, v_i\right\rangle_{\Gamma_i^D} + \lambda_i\left\langle g_i^D\cdot n_i, v_i\cdot n_i\right\rangle_{\Gamma_i^D}\right). \tag{23}$$

Note that the terms (15)–(17) were added in a way so that a_i in (19) is a symmetric bilinear form. Now, (18) is the starting point for the weak formulations for both the single domain and the interface problem. Note also that for the single domain, we have as follows:

$$\partial\Omega \setminus (\Gamma^D \cup \Gamma^N) = \emptyset, \tag{24}$$

while for the interface problem

$$\partial\Omega_i \setminus (\Gamma_i^D \cup \Gamma_i^N) = \Gamma_I. \tag{25}$$

3.1 Stabilizing small cuts

A common problem for immersed methods is robustness with respect to small cuts. In order to understand this problem, consider the single domain. Since Γ^C intersects the mesh in an arbitrary way, an element K may have an arbitrarily small intersection with the domain so that the size of $K \cap \Omega \ll h^d$. For each element, we integrate over $K \cap \Omega$. For the mass matrix, this means that the smallest eigenvalue can be arbitrarily small, and in turn that the condition number can be arbitrarily large. For the stiffness matrix, the problem is even worse. The term (15) that we add to enforce the boundary condition can make some eigenvalues of the stiffness matrix negative, which would make the method unstable.

A suggested way to remedy this problem is to add a stabilizing term, j_i, both to the term that corresponds to the mass matrix and to the term that corresponds to the stiffness matrix:

$$M_i(u_i, v_i) = (\rho_i u_i, v_i)_{\Omega_i} + \gamma_M^i j_i(u_i, v_i), \tag{26}$$

$$A_i(u_i, v_i) = a_i(u_i, v_i) + \frac{\gamma_A^i}{h^2} j_i(u_i, v_i). \tag{27}$$

Here, γ_M^i and γ_A^i are scalar constants that control how much stabilization is added. To explain the definition of j_i, let $\mathcal{F}_i$ denote the faces illustrated in Fig. 4. That is, the faces of $\mathcal{T}^C$ excluding the boundary faces of $\mathcal{T}_i$. To be precise, let

$$\mathcal{F}_i = \{F = T_a \cap T_b : T_a \in \mathcal{T}^C \text{ or } T_b \in \mathcal{T}^C, \quad T_a, T_b \in \mathcal{T}_i\}. \tag{28}$$

We now define the stabilization term as follows:

$$j_i(u, v) = \sum_{F \in \mathcal{F}_i} \sum_{k=1}^{p} \frac{h^{2k+1}}{(2k+1)(k!)^2} \left\langle [\partial_n^k u_i], [\partial_n^k v_i] \right\rangle_F . \tag{29}$$

Here, $\partial_n^k v_i$ denotes the kth derivative in the direction of the face normal, n, and $[\cdot]$ defines the jump over a face F:

$$[u_i] = u_i|_{F_+} - u_i|_{F_-} . \tag{30}$$

In (29), all derivatives up to the polynomial order, p, are included. Note that $[\cdot]$ is different from $[\![\cdot]\!]$ in (5) since we have u_i on both sides of F. The stabilization in (29) was suggested first in [7] and used first for the Poisson equation in [9]. For a nice explanation of why it works, see [21].

With stabilization one can prove, see [12], the following inequalities for the bilinear form M_i:

$$C_L \|v\|^2_{\Omega_i^\star} \leq M_i(v, v) \leq C_U \|v\|^2_{\Omega_i^\star}, \quad \forall v \in V_h^i. \tag{31}$$

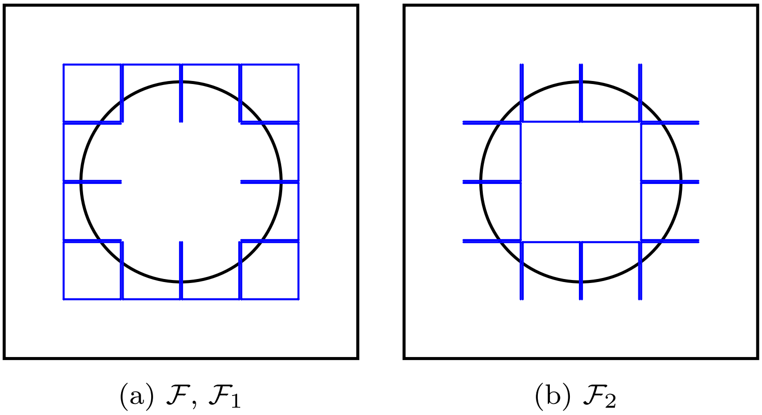


Fig. 4 Set of faces where the stabilization is applied. **a** $\mathcal{F}$, $\mathcal{F}_1$. **b** $\mathcal{F}_2$

In (31), C_L and C_U are positive constants that depend on the element order but not on h. From (31), we immediately get that the eigenvalues of the stabilized mass matrix are bounded independently of how the boundary/interface cuts the mesh. In turn, this bounds the condition number independently of the location of the boundary/interface. Unfortunately (as noted in both [27] and [12]), the constant in the bound increases very fast with the order of the elements. With stabilization, one can also show, see [12], that the bilinear form A is continuous and coercive independently of how the boundary/interface cuts the mesh.

3.2 Weak form for the single domain problem

For the single domain problem, we have that $\partial\Omega = \Gamma^D \cup \Gamma^N$, so by starting from (18) and adding the stabilizing terms, we get the following weak form for the single domain problem: Find u_h so that for each fixed $t \in (0, T]$, $u_h \in V_h$ such that

$$M(\ddot{u}_h, v) + A(u_h, v) = L(v), \quad \forall v \in V_h, \tag{32}$$

where L, M, A were defined in (20), (26), (27) but where we have dropped the subscript i, since there is only a single domain, Ω.

3.3 Weak form for the interface problem

We now want to derive the weak formulation for the interface problem (4). First, let $\kappa_1 > 0$ and $\kappa_2 > 0$ fulfill $\kappa_1 + \kappa_2 = 1$ and let $\{\cdot\}$ to denote the following convex combination:

$$\{v\} = \kappa_1 v_1 + \kappa_2 v_2. \tag{33}$$

By using that $\partial\Omega_i \setminus (\Gamma_i^D \cup \Gamma_i^N) = \Gamma_I, n = n_2 = -n_1$ and the condition (4e), we get that

$$
\begin{aligned}
&\sum_{i=1}^{2} - \langle \sigma(u_i) \cdot n_i, v_i \rangle_{\partial\Omega_i \setminus (\Gamma_i^D \cup \Gamma_i^N)} = \langle \sigma(u_1) \cdot n, v_1 \rangle_{\Gamma_I} - \langle \sigma(u_2) \cdot n, v_2 \rangle_{\Gamma_I} \\
&= \langle (\kappa_1 + \kappa_2)\sigma(u_1) \cdot n, v_1 \rangle_{\Gamma_I} - \langle (\kappa_1 + \kappa_2)\sigma(u_2) \cdot n, v_2 \rangle_{\Gamma_I} \\
&= \langle \kappa_1 \sigma(u_1) \cdot n, v_1 \rangle_{\Gamma_I} - \underbrace{\langle \kappa_1 \sigma(u_2) \cdot n, v_2 \rangle_{\Gamma_I}}_{(4e)} + \underbrace{\langle \kappa_2 \sigma(u_1) \cdot n, v_1 \rangle_{\Gamma_I}}_{(4e)} - \langle \kappa_2 \sigma(u_2) \cdot n, v_2 \rangle_{\Gamma_I} \\
&= \langle \kappa_1 \sigma(u_1) \cdot n, v_1 \rangle_{\Gamma_I} - \overbrace{\langle \kappa_1 \sigma(u_1) \cdot n, v_2 \rangle_{\Gamma_I}} + \overbrace{\langle \kappa_2 \sigma(u_2) \cdot n, v_1 \rangle_{\Gamma_I}} - \langle \kappa_2 \sigma(u_1) \cdot n, v_2 \rangle_{\Gamma_I} \\
&= \langle \kappa_1 \sigma(u_1) \cdot n, v_1 - v_2 \rangle_{\Gamma_I} + \langle \kappa_2 \sigma(u_1) \cdot n, v_1 - v_2 \rangle_{\Gamma_I} \\
&= - \left\langle \{\sigma(u) \cdot n\}, [\![v]\!] \right\rangle_{\Gamma_I} . \qquad (34)
\end{aligned}
$$

Note also that the interface condition (4d) is consistent with the following terms:

$$
- \left\langle [\![u]\!], \{\sigma(v) \cdot n\} \right\rangle_{\Gamma_I} = 0, \qquad (35)
$$

$$
\frac{\gamma_I}{h} \left\langle [\![u]\!], [\![v]\!] \right\rangle_{\Gamma_I} = 0. \qquad (36)
$$

Here, γ_I is a positive constant which will control how strongly the interface condition is enforced. Now, we add (18) for each domain, use (34) and add (35), (36) and stabilization to obtain the finite element method: Find $u = \{u_1, u_2\}$ so that for each fixed $t \in (0, T]$, $u \in V_h^1 \times V_h^2$ such that

$$
\sum_{i=1}^{2} (M_i(\ddot{u}_i, v_i) + A_i(u_i, v_i)) + I(u, v) = \sum_{i=1}^{2} L_i(v_i), \quad \forall v = \{v_1, v_2\} \in V_h^1 \times V_h^2. \qquad (37)
$$

Here, L_i, M_i, and A_i were defined in (20), (26) and (27). The bilinear form I that enforces the interface conditions is given by the following:

$$
I(u, v) = - \left\langle \{\sigma(u) \cdot n\}, [\![v]\!] \right\rangle_{\Gamma_I} - \left\langle [\![u]\!], \{\sigma(v) \cdot n\} \right\rangle_{\Gamma_I} + \frac{\gamma_I}{h} \left\langle [\![u]\!], [\![v]\!] \right\rangle_{\Gamma_I} . \qquad (38)
$$

The method contains several free parameters that need to be chosen. Clearly, the penalty parameters related to the stabilization should scale with the parameters of the materials. We choose to scale them as follows:

$$
\gamma_M^i = \frac{1}{4}\rho_i, \quad \gamma_A^i = \frac{1}{2}\eta_i, \qquad (39)
$$

where

$$
\eta_i = 2\mu_i + \lambda_i. \qquad (40)
$$

We choose the constants related to the interface terms in the following way:

$$
\gamma_I = 20 p^2 \frac{\eta_1 \eta_2}{\eta_1 + \eta_2}, \qquad (41)
$$

where

$$
\kappa_1 = \frac{\eta_2}{\eta_1 + \eta_2}, \quad \kappa_2 = \frac{\eta_1}{\eta_1 + \eta_2}. \qquad (42)
$$

The scaling with respect to η_i is analogous to the choice of parameters for the Poisson interface problem in [8]. The Nitsche parameter related to the Dirichlet boundary condition is chosen as follows:

$$\gamma_D = 5p^2. \tag{43}$$

Here, the scaling with p^2 of γ_D and γ_I in (41) and (43) follows from an inverse inequality. The numerical constants are chosen based on experience. We shall briefly discuss this in Section 6.

Remark 2 Our assumption that the mesh consists of perfectly square elements is not necessary. When the face normals are not aligned with the coordinate axes, we just need to use a slightly more general definition of the face normal derivatives, $\partial_n^k v$, see for example [21].

Remark 3 Although we will not do so in this paper, one possibility is to under-integrate the non-cut elements using a Gauss-Lobatto quadrature with $p+1$ points in each coordinate direction. This choice results in local mass matrices which are diagonal, for non-cut elements that are so far away from the immersed boundary/interface that they are not affected by the added stabilization. This, in turn, makes a large block in the global mass matrix diagonal.

3.4 Imposition of initial conditions

To impose the initial conditions, we first define the stabilized L_2 projection, $\varPi_h u$. For the single domain problem, $\varPi_h u$ is defined as the solution to the following problem: Given u, find $\varPi_h u \in V_h$ such that

$$M(\varPi_h u, v) = (u, v)_\varOmega, \quad \forall v \in V_h. \tag{44}$$

For the interface problem, $\varPi_h u$ is defined analogously as the solution to the following: Given u, find $\varPi_h u = \{\varPi_h u_1, \varPi_h u_2\} \in V_h^1 \times V_h^2$ such that

$$\sum_{i=1}^{2} M_i(\varPi_h u_i, v_i) = \sum_{i=1}^{2} (u_i, v_i)_\varOmega, \quad \forall v = \{v_1, v_2\} \in V_h^1 \times V_h^2. \tag{45}$$

The initial conditions are now imposed as follows:

$$u_h|_{t=0} = \varPi_h u|_{t=0}, \tag{46a}$$

$$\dot{u}_h|_{t=0} = \varPi_h \dot{u}|_{t=0}. \tag{46b}$$

Note that, by setting the discrete initial conditions in this way, the initial conditions of the single domain problem, (1d)–(1e), only need to be defined on $\varOmega$ and not on $\varOmega^\star$.

4 Theory

In this section, we will present some theoretical results, in particular, a proof of convergence for the semi-discrete method for the single domain problem. The proof builds on the results presented in [12] where several time-independent problems were studied. In the analysis, we will use the following norms:

$$\|v\|_M^2 = M(v, v), \quad \|v\|_A^2 = A(v, v), \quad |v|_j^2 = j(v, v), \tag{47}$$

$$|||v|||_h^2 = \|v\|_A^2 + h\|\sigma(v)\|_{\Gamma^D}^2 + \frac{1}{h}(2\mu\|v\|_{\Gamma^D}^2 + \lambda\|v \cdot n\|_{\Gamma^D}^2), \tag{48}$$

where we can note that $|\cdot|_j$ is a semi-norm. Note that these norms only make sense if the argument is defined on Ω^*. We will also use the $\lesssim$ – relation, which we define as follows:

$$a \lesssim b \Leftrightarrow a \leq Cb, \tag{49}$$

where C is some constant that is independent of h.

We also need a bounded extension operator, $\mathbb{E} : H^s(\Omega) \to H^s(\Omega^\star)$. We shall assume that the solution is sufficiently smooth (s is sufficiently high) and that $\partial\Omega$ is sufficiently regular so that

$$j(\mathbb{E}\ddot{u}, v) = 0, \quad \forall v \in V_h. \tag{50}$$

4.1 Ritz projection

To prove convergence, we need a "Ritz-like" projection, which we define as the solution to the following problem: Given u, find $R_h u \in V_h$ such that

$$A(R_h u, v) = a(u, v), \quad \forall v \in V_h. \tag{51}$$

In this section, we will gather some results about the Ritz projection, which will be essential in the analysis to come. For brevity, we will from here on omit the "like" in the Ritz-like projection (51) and simply call it the Ritz projection. As shown in [12], given that γ_D is sufficiently large, A is coercive and continuous with respect to $|||\cdot|||_h$. That is, there exists constants $C_r, C_c > 0$ such that

$$C_r|||v_h|||_h^2 \leq A(v_h, v_h), \quad A(v_h, w_h) \leq C_c|||v_h|||_h|||w_h|||_h, \quad v_h, w_h \in V_h. \tag{52}$$

For simplicity, we will assume that $\Gamma^D \neq \emptyset$. When this holds, $|||\cdot|||_h$ is indeed a norm (i.e., not only a semi-norm) and (51) has a unique solution. However, this assumption can likely be relaxed by looking for the solution in a constrained subspace of V_h.

One should note that this projection is nothing but the solution to the time-independent elasticity problem. To see this, let $\hat{u}(x) = u(x, t_f)$, where t_f is some fixed time, and define $\hat{f}$ so that $\hat{u}$ is the solution to the following:

$$\nabla \cdot \sigma(\hat{u}) = -\hat{f}, \quad x \in \Omega, \tag{53a}$$

$$\hat{u} = g^D(x, t_f), \quad x \in \Gamma^D, \tag{53b}$$

$$\tfrac{\partial \hat{u}}{\partial n} = g^N(x, t_f), \quad x \in \Gamma^N. \tag{53c}$$

This means that $\hat{u}$ will satisfy the following:

$$a(\hat{u}, v) = \hat{L}(v), \tag{54}$$

where $\hat{L}$ is defined as follows:

$$\begin{aligned}\hat{L}(v) \;=\; & (\hat{f}, v) + \left\langle g^N, v\right\rangle_{\Gamma_N} - \left\langle g^D, \sigma(v)\cdot n\right\rangle_{\Gamma^D} \\ & + \frac{\gamma_D}{h}\left(2\mu\left\langle g^D, v\right\rangle_{\Gamma^D} + \lambda\left\langle g^D\cdot n, v\cdot n\right\rangle_{\Gamma^D}\right),\end{aligned} \tag{55}$$

i.e., the same as L in (20) but using the right-hand side data from (53). We can now formulate the finite element method to solve (53) as: Find $\hat{u}_h \in V_h$ such that

$$A(\hat{u}_h, v_h) = \hat{L}(v_h), \quad \forall v_h \in V_h. \tag{56}$$

Now, by subtracting (54) from (56), we can see that the solution $\hat{u}$, to the problem in (53), in fact corresponds to the Ritz projection $R_h u$ in (51). So in principle, the Ritz projection is obtained by solving a linear elasticity problem. This has been treated in detail in [12], where the results presented in Lemma 1were derived.

Lemma 1 *For the Ritz projection, $R_h u$, in (51), the following error estimates hold:*

$$|||R_h u - \mathbb{E}u|||_h \lesssim h^k \|u\|_{H^{k+1}(\Omega)}, \tag{57}$$

$$\|R_h u - u\|_\Omega \lesssim h^{k+1}\|u\|_{H^{k+1}(\Omega)}. \tag{58}$$

Proof See Theorem 4.2 in [12]. □

We will also need the following corollary.

Corollary 1 *For the Ritz projection, $R_h u$, in (51), the following holds:*

$$|\mathbb{E}u - R_h u|_j \lesssim h^{k+1}\|u\|_{H^{k+1}(\Omega)}. \tag{59}$$

Proof From (57) and the definition of $|||\cdot|||_h$ in (48), we get the following:

$$h^{-2}|\mathbb{E}u - R_h u|_j^2 \lesssim |||R_h u - \mathbb{E}u|||_h^2 \lesssim h^{2k}\|u\|^2_{H^{k+1}(\Omega)}, \tag{60}$$

from which (59) follows. □

4.2 A priori analysis

The analysis presented here is similar to the one presented in [26]. We wish to bound the error $u_h - u$, and in doing so, we split the error in two parts:

$$u_h - \mathbb{E}u = e_N + e_R, \tag{61}$$

where $e_N = u_h - R_h u$ and $e_R = R_h u - \mathbb{E}u$. By Lemma 1, we directly get a bound for e_R. To bound e_N, we first aim to find a bound on the "energy" of e_N, which we define as follows:

$$E_{e_N} = \frac{1}{2}(M(\dot{e}_N, \dot{e}_N) + A(e_N, e_N)). \tag{62}$$

To facilitate the proof, we will in this section assume that the discrete initial conditions are imposed using the Ritz projection:

$$u_h|_{t=0} = R_h u|_{t=0}, \tag{63a}$$

$$\dot{u}_h|_{t=0} = R_h \dot{u}|_{t=0}. \tag{63b}$$

Note that (63) is not the same initial conditions as in (46), which are used in the numerical experiments. The reason is that computing the Ritz projection is more involved since it requires that we first compute $\hat{f}$ in (53a) by differentiating u. On the other hand, computing the L_2 projection is simple. Note that both projections approximate the analytical solution with the same order of accuracy. The L_2 projection fulfills the same type of bound as (58):

$$\|\Pi_h u - u\|_\Omega \lesssim h^{k+1} \|u\|_{H^{k+1}(\Omega)}, \tag{64}$$

so in L_2-norm, the error that the two different projections make only differ by a constant. Thus, which projection is used in practice will most likely make little difference. However, the choice (63) makes the analysis simpler since it is equivalent to the following:

$$e_N|_{t=0} = 0, \tag{65}$$

$$\dot{e}_N|_{t=0} = 0, \tag{66}$$

which by the definition of the energy in (62) gives us

$$E_{e_N}|_{t=0} = 0. \tag{67}$$

We are now ready to bound the energy.

Lemma 2 *The following bound holds*

$$E_{e_N}(t) \lesssim h^{2(k+1)}. \tag{68}$$

Proof First, we have that

$$\begin{aligned} M(\ddot{e}_N, v_h) + A(e_N, v_h) &= M(\ddot{u}_h, v_h) + A(u_h, v_h) - M(R_h\ddot{u}, v_h) - A(R_h u, v_h) \\ &= (\rho\ddot{u}, v_h)_\Omega + a(u, v_h) - M(R_h\ddot{u}, v_h) - A(R_h u, v_h) \\ &= (\rho\ddot{u}, v_h)_\Omega - M(R_h\ddot{u}, v_h) \\ &= (\rho\ddot{u}, v_h)_\Omega - M(R_h\ddot{u}, v_h) + \gamma_M j(\mathbb{E}\ddot{u}, v_h) \\ &= M(-\ddot{e}_R, v_h), \end{aligned} \tag{69}$$

where we in the first line used the definition of e_N. When going to the second line, we used the definition of the finite element method in (32) and that the analytical solution satisfies the following:

$$(\rho\ddot{u}, v_h)_\Omega + a(u, v_h) = L(v_h), \quad \forall v_h \in V_h. \tag{70}$$

When going to the third line, we used the definition of the Ritz projection in (51). Finally, we used (50) and the definition of e_R. Now, choosing $v_h = \dot{e}_N$ in (69), we can use the definition of the energy and that M is an inner product (so that Cauchy-Schwarz applies) to get the following:

$$\frac{\mathrm{d}E_{e_N}}{\mathrm{d}t} \leq \|\ddot{e}_R\|_M \|\dot{e}_N\|_M \leq \|\ddot{e}_R\|_M \sqrt{2E_{e_N}}. \tag{71}$$

By using

$$\frac{\mathrm{d}E_{e_N}}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}(\sqrt{E_{e_N}})^2 = 2\sqrt{E_{e_N}}\,\frac{\mathrm{d}}{\mathrm{d}t}\sqrt{E_{e_N}}, \tag{72}$$

we can divide both sides of (71) by $2\sqrt{E_{e_N}}$ and get the following:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\sqrt{E_{e_N}} &\leq \frac{1}{\sqrt{2}}\|\ddot{e}_R\|_M \\ &\leq \frac{1}{\sqrt{2}}\sqrt{\|\rho\ddot{e}_R\|_\Omega^2 + \gamma_M |\mathbb{E}\ddot{u} - R_h\ddot{u}|_j^2} \\ &\leq Ch^{k+1}\|\ddot{u}\|_{H^{k+1}(\Omega)}, \end{aligned} \tag{73}$$

where we in the last line used Lemma 1 and Corollary 1. Integrating and squaring (73) gives the following:

$$E_{e_N}(t) \leq \left(\sqrt{E_{e_N}(0)} + Ch^{k+1}\int_0^t \|\ddot{u}\|_{H^{k+1}(\Omega)} d\tau\right)^2. \tag{74}$$

Finally, using (67) gives us the bound in (68). □

We are now ready to state our a priori error estimates.

Theorem 1 *Let u be the solution to (1) and let u_h be the solution to (32), then at any given time, t, the following a priori error estimates hold*

$$\|u_h - u\|_\Omega \lesssim h^{k+1}, \tag{75}$$

$$\|\nabla u_h - \nabla u\|_\Omega \lesssim h^k. \tag{76}$$

Proof Using the definition of E_{e_N} and Lemma 2, we get the following:

$$\|\dot{e}_N\|_\Omega = \|\dot{u}_h(t) - R_h\dot{u}(t)\|_\Omega \lesssim h^{k+1}, \tag{77}$$

$$\|e_N\|_A = \|u_h(t) - R_h u(t)\|_A \lesssim h^{k+1}. \tag{78}$$

To bound e_N and not $\dot{e}_N$, note that

$$2\|e_N\|_\Omega \frac{\mathrm{d}}{\mathrm{d}t}\|e_N\|_\Omega = \frac{\mathrm{d}}{\mathrm{d}t}\|e_N\|_\Omega^2 = 2(e_N, \dot{e}_N)_\Omega \leq 2\|e_N\|_\Omega \|\dot{e}_N\|_\Omega. \tag{79}$$

Dividing (79) by $2\|e_N\|_\Omega$ and integrating over time gives the following:

$$\|e_N(t)\|_\Omega \leq \int_0^t \|\dot{e}_N(\tau)\|_\Omega\, d\tau, \tag{80}$$

by using (65). Combining (80) with (77) gives us the following:

$$\|u_h(t) - R_h u(t)\|_\Omega \lesssim h^{k+1}. \tag{81}$$

Finally, we use the triangle inequality on (61) and combine (78) and (81) with the bounds on e_R from Lemma 1 to get the estimates (75) and (76). □

Remark 4 Theorem 1 builds on the error estimates of the Ritz-like projection from Lemma 1. This is a direct consequence of the a priori error estimates of the corresponding stationary problem. These were given in [12], but only for the single domain problem. If a priori error estimates were available also for the stationary interface problem, the error estimates of the time-dependent interface problem (4) would follow in the same way.

4.3 Time step restriction

Both of the weak forms, (32) and (37), will discretize to a system of the following form:

$$\mathcal{M}\ddot{\xi} + \mathcal{A}\xi = \mathcal{L}(t), \tag{82}$$

where $\mathcal{M} \in \mathbb{R}^{N\times N}$ is the mass matrix, $\mathcal{A} \in \mathbb{R}^{N\times N}$ is the stiffness matrix, and $\mathcal{L} : \mathbb{R} \to \mathbb{R}^N$.

If we use explicit time-stepping the largest time step, τ, we can take due to stability restrictions will be bounded by the CFL number in the following way:

$$\tau \leq \alpha C_{\mathrm{CFL}} h, \tag{83}$$

where α is a constant which depends on the chosen time-stepping scheme. The CFL number can be computed from the matrices in the discrete system. Let $l_{\max}$ be the largest eigenvalue of the generalized eigenvalue problem: find $l \in \mathbb{R}$, $x \in \mathbb{R}^N$ such that

$$\mathcal{A}x - l\mathcal{M}x = 0. \tag{84}$$

Then, the CFL number is given by the following:

$$C_{\mathrm{CFL}} = \frac{1}{h\sqrt{l_{\max}}}. \tag{85}$$

It is important that the CFL number does not decrease significantly when the smallest cut in the mesh approaches zero. Ideally, the time step restriction should not be more severe than for the standard non-cut finite element method.

4.4 Material parameters

The problem for the single domain contains three material parameters, ρ, λ, and μ. However, by rescaling (see [16]), one can show that the dimensionless equation only depends on the ratio, β, between the Lamé parameters:

$$\beta = \frac{\lambda}{\mu}. \tag{86}$$

Thus, we can without loss of generality assume that the equation is already in dimensionless form and set $\rho = \mu = 1$. Now, we can obtain different physical behaviors by varying λ. For the interface problem, we shall also assume that we are working in dimensionless form. By a corresponding analysis, it is possible to show that we can set $\rho_1 = \mu_2 = 1$ and obtain different physical behavior by varying λ_1, ρ_2, λ_2, and μ_2.

5 Numerical experiments

In this section, we present some numerical examples. First, we investigate if the error converges with the expected order. This is done for the single domain problem in Section 5.1 and for the interface problem in Section 5.2. In Section 5.3, we investigate how the properties of the discretized matrices in (82) change when the smallest cut in the mesh approaches zero. To implement the method, we have used the finite element library deal.II [6]. A level-set function has been used to represent the immersed boundary/interface. On intersected elements, individual quadrature rules need to be generated, and for these, we have used the algorithm from [25].

Remark 5 As long as the immersed boundary/interface is smooth, the corresponding level-set function will be smooth, at least in a neighborhood of the boundary/interface. The boundary/interface can then be well represented by the zero contour of a piecewise polynomial approximation of the level-set function. However, if the boundary/interface is non-smooth, a different representation of the geometry is needed to achieve high accuracy. Also, the quadrature rules on the intersected elements need to take such non-smoothness into account if high accuracy is to be maintained.

Unless stated otherwise, the following material parameters have been used in the experiments below:

$$\begin{aligned} \rho &= \rho_1 = 1, & \rho_2 &= 1.1154, \\ \lambda &= \lambda_1 = 1.1429, & \lambda_2 &= 2.6182, \\ \mu &= \mu_1 = 1, & \mu_2 &= 1.8. \end{aligned} \tag{87}$$

These parameters correspond to material 1 being sandstone and material 2 being granite; these are two of the most common rock types. Note that by using the present model, we have assumed that the materials are linear, homogeneous, and isotropic, which possibly is unrealistic for these types of rock.

For waves in elastic materials, two different wave speeds are of importance. The pressure-, c_p, and shear-wave speed, c_s. These relate to the material parameters as follows:

$$c_p = \sqrt{\frac{\lambda + 2\mu}{\rho}}, \qquad c_s = \sqrt{\frac{\mu}{\rho}}. \tag{88}$$

The parameters in (87) correspond to the following wave speeds:

$$\begin{aligned} c_p &= c_{p,1} = 1.7728, & c_{p,2} &= 2.3611, \\ c_s &= c_{s,1} = 1, & c_{s,2} &= 1.2704. \end{aligned}$$

For time discretization, we have used the explicit fourth-order accurate classical Runge-Kutta after rewriting the second-order system (82) as a first-order system by introducing an auxiliary variable. In the experiment below, a time step

$$\tau = 0.2 \frac{h}{p^2} \left(\max_{\Omega}(c_p) \right)^{-1}, \tag{89}$$

has been used. The condition number of the mass matrix is expected to be large, so a direct solver was used to invert $\mathcal{M}$ during the time-stepping of (82).

5.1 Convergence for the single domain problem

Assume that we have an elastic pressure wave traveling through $\mathbb{R}^2$ in the x-direction:

$$u_1^{\text{in}}(x,t) = \cos(\omega(t - x/c_p)), \quad u_2^{\text{in}} = 0. \tag{90}$$

Here, ω is a constant which we choose as $\omega = \pi$. Let this wave hit a circular inclusion (vacuum inside) with radius, $R = 1$. At the boundary of the inclusion, Γ_N, a homogeneous Neumann boundary condition is enforced. If we consider this problem in all of $\mathbb{R}^2$, the total solution, $\tilde{u}$, will be the sum of the incoming, u^{in}, and reflected wave, u^{ref}:

$$\tilde{u} = u^{\text{in}} + u^{\text{ref}}. \tag{91}$$

The reflected wave can be computed analytically. The total analytical solution (given in [33]) is periodic in time and can be written as a series expansion in Bessel and Hankel functions. In this paper, we truncate the series and use it as our solution $\tilde{u}$. Since the solution is rather complicated, we do not restate the series expansion here, but merely refer the interested reader to [33].

Consider now the single domain problem in (1) posed on the finite domain as in Fig. 1a. We have a finite square domain with side length $L = 2\pi$. As in Fig. 2a, the outer boundary is aligned with the mesh but the inner boundary is not. We want to make the solution, u, on this truncated domain equal to the analytical solution, $\tilde{u}$, on $\mathbb{R}^2$. To achieve this, we set the initial conditions equal to $\tilde{u}$:

$$u|_{t=0} = \tilde{u}|_{t=0}, \quad \left.\frac{\partial u}{\partial t}\right|_{t=0} = \left.\frac{\partial \tilde{u}}{\partial t}\right|_{t=0}, \tag{92}$$

and impose a Dirichlet boundary condition on the outer boundary equal to $\tilde{u}$:

$$u|_{\Gamma_D} = \tilde{u}. \tag{93}$$

We solve this problem until the end time $T = 2$ (corresponding to one period) and compute the L_2-error for decreasing mesh sizes. Snapshots of the solution at the initial time and a quarter of a period later are shown in Fig. 5.

The error in L_2-norm as a function of element size is shown in Fig. 6 for Q_1 to Q_3 elements. The straight lines in the figure denote the expected order of accuracy. We see that the order is a bit low for large h, but when going to finer h, we get the expected order or even slightly higher order than expected.

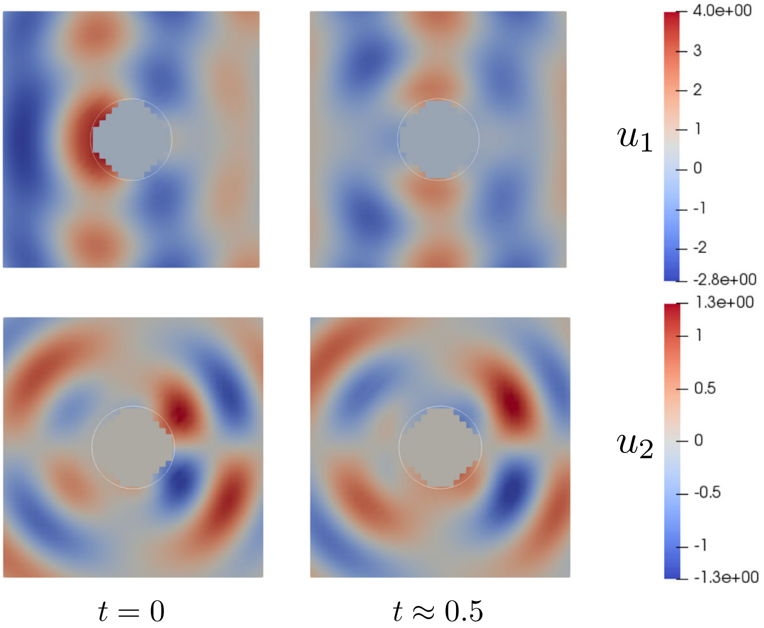


Fig. 5 Snapshots of the solved single domain problem

5.2 Convergence for the interface problem

Consider now a similar setup as in Section 5.1. We have a plane wave of the form (90) traveling through a material in $\mathbb{R}^2$ towards a disc. The material has properties ρ_1, λ_1, μ_1, and the disc has a radius equal to 1. However, instead of vacuum, we replace the

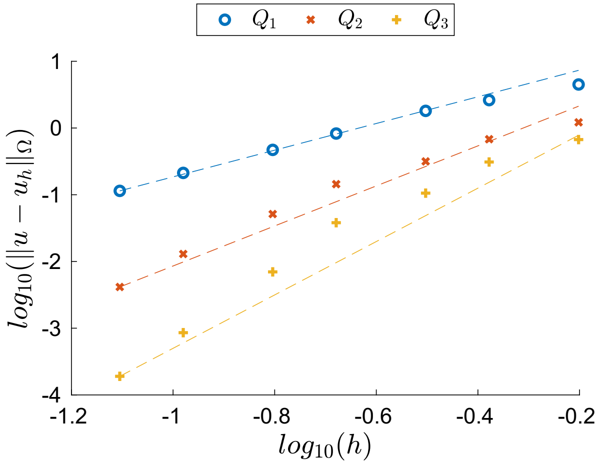


Fig. 6 L_2-error versus element size for the single domain problem, together with straight lines corresponding to the expected order of accuracy

material of the disc by another material with properties ρ_2, λ_2, μ_2. In the same way as before, the reflected wave can be solved for analytically and the total solution, $\tilde{u}$, can be found in [33] in the form of a series expansion. We again truncate the series and use it as our solution.

Now, we solve the interface problem (4) posed on the finite domain in Fig. 1b. Again, we have a square domain with side length 2π. To make the solution of the problem equal to the analytical solution, we again set the initial condition and the outer Dirichlet boundary condition equal to $\tilde{u}$, as in (92)–(93). Snapshots of the solution at two different times are seen in Fig. 7. We see that the displacement in the x-direction looks like the plane wave in (90), but since the wave speed is higher in Ω_2, the plane wave gets distorted.

To verify the convergence, we solve until the end time $T = 2$ (corresponding to one period) and then compute the error. The error in L_2-norm as a function of element size is seen in Fig. 8. We see that the order of accuracy is as expected for Q_1 and Q_2 elements. For Q_3 elements, the order is a bit low for large h, but eventually reaches the expected order when we go to finer h.

5.3 Matrix properties with decreasing cut size

Consider the setup illustrated in Fig. 9a for the single domain and in Fig. 9b for the interface problem. For both setups, we have a rectangular domain on top of a square grid. For the single domain problem in Fig. 9a, the left, bottom, and top boundary are aligned with the mesh, but the right domain boundary intersects the last column of

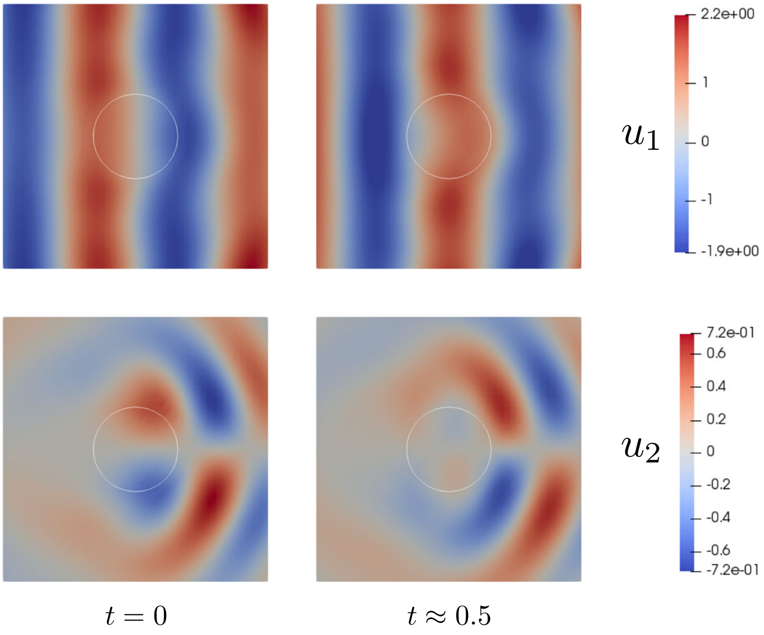


Fig. 7 Snapshots of the solved interface problem

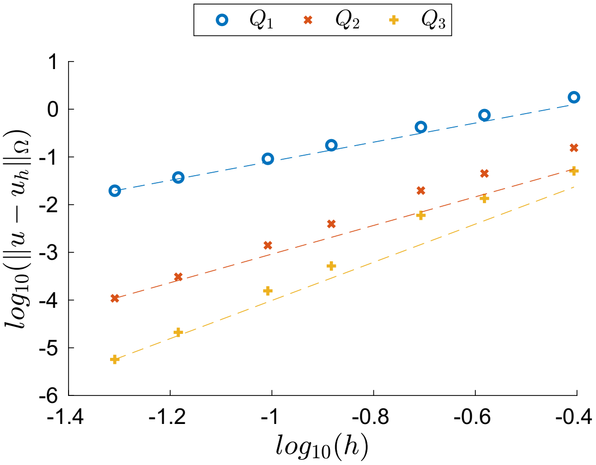
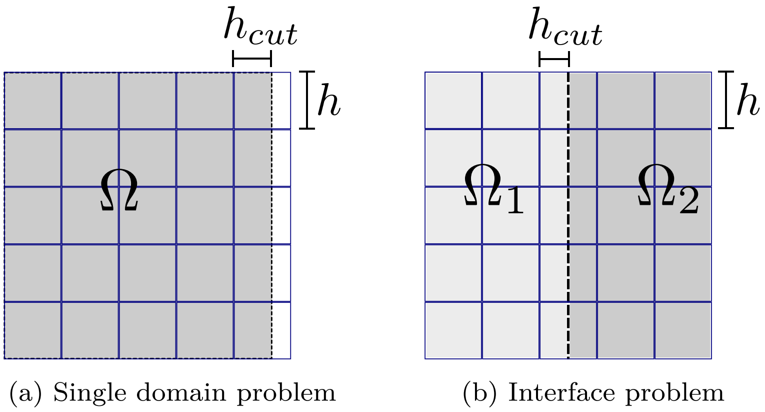


Fig. 8 L_2-error versus element size for the interface problem, together with straight lines corresponding to the expected order of accuracy

elements with a cut of size h_{cut}. For the interface problem, all boundaries are aligned with the mesh boundaries, but the immersed interface intersects the middle column of elements with a cut of size h_{cut}. We are now interested in how the properties of the mass and stiffness matrix change when we vary the size of h_{cut}. In the experiment, we use a background mesh containing 9 × 9 elements, which is slightly finer than what is illustrated in Fig. 9.

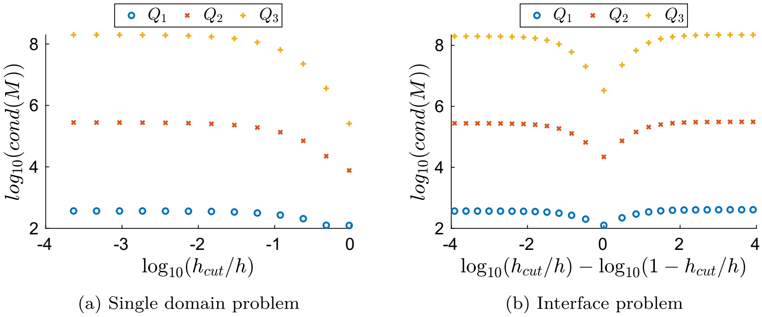
How the condition number of the mass matrix changes is seen in Fig. 10a for the single domain problem. We see that when the cut size is large ($h_{\text{cut}}/h \approx 1$) the condition number is small and initially grows when h_{cut} is decreased. However, as the cut size is decreased further the condition number becomes constant, as expected from the theory. We also see that the constant level increases very fast when we



(a) Single domain problem

(b) Interface problem

Fig. 9 Experiments where elements are intersected with a cut of size h_{cut}. **a** Single domain problem. **b** Interface problem



(a) Single domain problem

(b) Interface problem

Fig. 10 Condition number of the mass matrix when decreasing the size of h_{cut} in Fig. 9. **a** Single domain problem. **b** Interface problem

increase the order of the elements, which is consistent with the results previously presented in [12, 27].

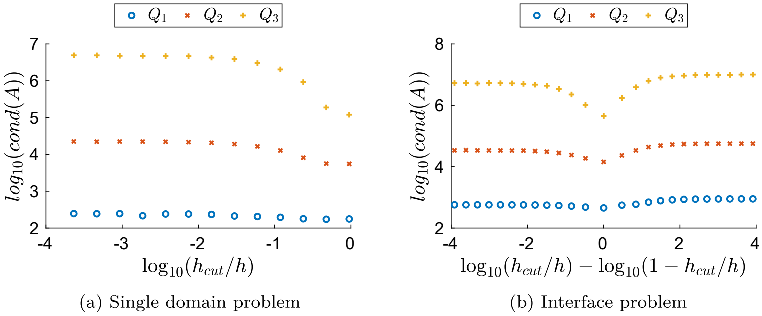
In Fig. 10b, we see the condition number of the mass matrix for the interface problem. Note that we have $f(h_{\text{cut}}/h)$ on the x-axis, where

$$f(x) = \log_{10}(x) - \log_{10}(1 - x). \tag{94}$$

This makes the x-axis "almost logarithmic" as h_{cut}/h approaches both 0 and 1, since $f(x)$ is monotone on the interval (0, 1) and maps (0, 1) to $(-\infty, \infty)$. In Fig. 10b, we see that the behavior is analogous to the single domain problem as h_{cut}/h approaches 0. We also see that the curve is almost mirrored in the point $h_{\text{cut}} = h/2$, and that the curve is not exactly mirrored can be explained by the difference in material parameters.

In the same way, the condition number of the stiffness matrix is seen in Fig. 11a and b. We see that the dependence is similar as for the mass matrix in Fig. 10a and b.

The CFL number computed from (85) is shown in Fig. 12a for the single domain problem and in Fig. 12b for the interface problem. We see in the figures that the CFL number is completely independent of the size of the cut. We also see that the CFL number becomes smaller when we increase the order of the elements. This is also the



(a) Single domain problem

(b) Interface problem

Fig. 11 Condition number of the stiffness matrix when decreasing the size of h_{cut} in Fig. 9. **a** Single domain problem. **b** Interface problem

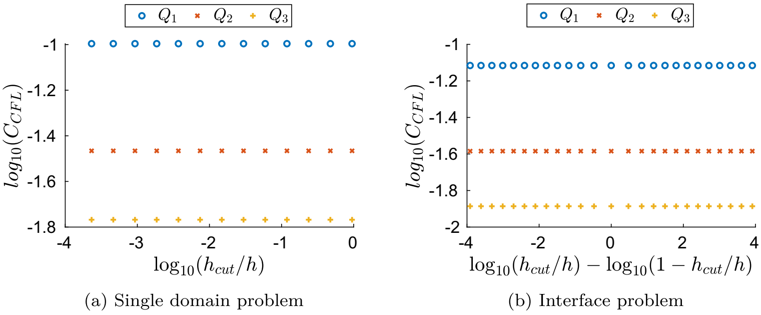


Fig. 12 CFL number when decreasing the size of h_{cut} in Figure 9. **a** Single domain problem. **b** Interface problem

case when using the standard (non-cut) finite element method. Note that the scale is quite different in Figs. 10 and 11 compared to Fig. 12.

5.4 Example with a non-trivial geometry

Finally, we solve the single domain problem (1) with a slightly more complicated geometry, to demonstrate that it is easy to handle more general geometries. We consider a background mesh over $\Omega_0 = [-6, 2] \times [-2, 2]$, and the following domain:

$$\Omega = \{x \in \Omega_0 : \psi(x) < 0\}. \tag{95}$$

In (95), $\psi : \mathbb{R}^d \to \mathbb{R}$ is the following level-set function:

$$\psi(x) = R + R_0 \sin(\omega \arg(x))) - \sqrt{x_1^2 + x_2^2}, \tag{96}$$

where $R = 1$, $R_0 = 0.4$, and $\omega = 4$. The boundaries of Ω_0 are aligned with the mesh, while the boundary described by ψ is immersed in the mesh: $\Gamma^C = \{x : \psi(x) = 0\}$. Let $\lambda = 1$, $\mu = 2$, $\rho = 3$ and let the initial conditions, (1d)–(1e), be zero: $u^0 = 0$, $w^0 = 0$. A homogeneous Dirichlet boundary condition is applied on Γ^C. On the outer boundary, $\partial\Omega_0$, the following Dirichlet boundary condition is applied:

$$g_1^D(x, t) = 0, \tag{97}$$

$$g_2^D(x, t) = \begin{cases} \cos(\pi x_2) e^{-(t-t_c)^2/\sigma^2}, & x_1 = -6, \\ 0, & x_1 \neq -6, \end{cases} \tag{98}$$

where $t_c = 1.5$ and $\sigma = 0.25$. This means that the left boundary generates a pressure wave package in the u_2 component. In Fig. 13, we see a few snapshots of the norm

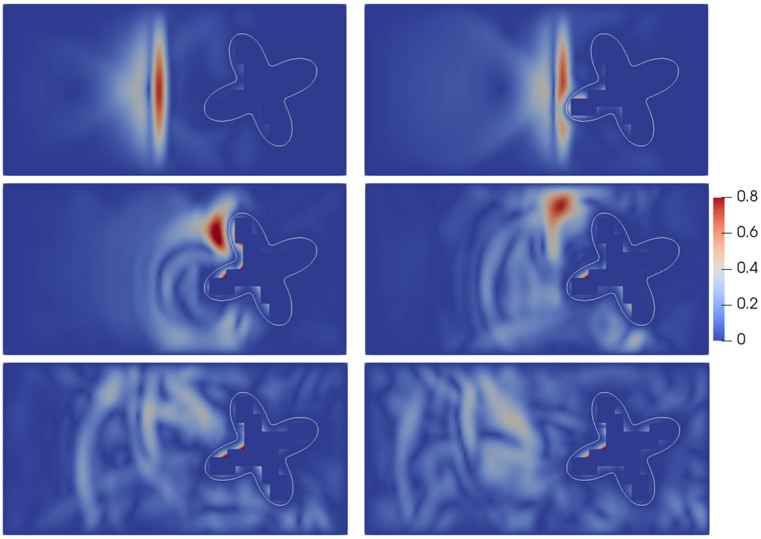


Fig. 13 Norm of the displacement of the solution to the problem described in Section 5.4. At times, $t \approx 5.92,\ 7.10,\ 8.29,\ 9.47,\ 10.66,\ 11.84$ (listing left to right and top-down). Solved with an element size $h = 0.2$ and Q_3 elements.

of the displacement, $\sqrt{u_1^2 + u_2^2}$, in the interval $t \in [0, 12]$. We see that the wave propagates towards the immersed boundary, hits it, and is reflected.

6 Discussion

In this paper, we have described a high-order cut finite element method for the elastic wave equation, which can handle both boundaries and material interfaces without requiring alignment of element boundaries. We have implemented the method in 2 space dimensions using the finite element library [6], but the extension to 3 space dimensions is straightforward. The numerical experiments in Sections 5.1 and 5.2 using elements with piecewise linear, quadratic, and cubic basis functions show that the method converges with orders 2, 3, and 4, which are the orders expected from Theorem 1. From the experiment in Section 5.3, we see that the method is robust when the size of the smallest cut in the mesh approaches zero.

A choice of numerical coefficients in the expressions of γ_M^i, γ_A^i, γ_I, and γ_D, see (39), (42), and (43), has been made. As far as we have seen, the method is not particularly sensitive to this choice. Still, one can wonder what happens when they are chosen differently. If γ_D and γ_I are chosen too small, coercivity is lost and the method becomes unstable, due to eigenvalues of the stiffness matrix becoming negative. This has nothing to do with the method being immersed. The same thing occurs also when symmetric Nitsche techniques are used in non-cut methods. Generally, one wants to choose γ_D and γ_I close to the stability limit. If they are chosen larger than necessary, the CFL number becomes smaller. The influence of the stabilization parameters γ_A^i and γ_M^i on the condition numbers of the mass and stiffness matrix were discussed in [9, 26], for linear P_1 elements. There one could see that the condition numbers had a minimum when either stabilization parameter increased from 0. However, the condition number of either matrix increased rather slowly after passing the minimum. Thus, choosing γ_M^i or γ_A^i slightly larger than necessary does not have a severe effect.

As mentioned earlier, high-order methods are typically attributed to being more efficient for hyperbolic problems. We have not compared efficiency for different orders of the present method, but several aspects would affect the efficiency. When increasing the order of elements, the order of the quadrature must also be increased. Creating quadrature rules on the intersected elements is typically expensive, and using more quadrature points means more work. However, for a fixed geometry this only needs to be done once. Therefore, the question of whether or not it pays off to increase the order likely depends on what algorithm is being used to generate the quadrature and on how long the time interval of interest is. If the time interval is sufficiently long, we do not expect the quadrature to destroy the efficiency of the higher order method. A second aspect is the increased work in the time-stepping of (82), due to stabilization being added to the mass matrix. When Gauss-Lobatto quadrature is used on the non-intersected elements (see Remark 3), a suitable reordering of the degrees of freedom yields a mass matrix which consist of one large diagonal block and a much smaller non-diagonal block. The condition number of the small block will potentially be large when the polynomial order of the elements is high, but as long as the block is small enough to be factorized, we have seen no difficulties. However, if an iterative method must be used, there could be severe loss of efficiency if no suitable preconditioning technique is found. A third aspect is the reduction of the CFL number when increasing the polynomial order of the elements. The reduction is not particular to the cut finite element method but will impact the efficiency.

Future work involves investigating a broader set of materials. The parameters (87) of the two materials used in the experiments for the interface problem are different but do not differ significantly. A future possibility would be to test how more extreme differences in material parameters affect the performance of the method. For the interface problem, the limit $\mu_2 \to 0$ is particularly important. In this case, the material 2 stops being elastic and the problem on Ω_2 becomes equivalent to the acoustic wave equation [22]. One disadvantage of taking the limit $\mu_2 \to 0$ is that the problem on Ω_2 still is a system. Thus, one future research direction would be to consider the problem of the elastic wave equation coupled directly with the acoustic wave equation.

Funding Information Open access funding provided by Uppsala University.

References

1. Abarbanel, S., Ditkowski, A.: Asymptotically stable fourth-order accurate schemes for the diffusion equation on complex shapes. J. Comput. Phys. **133**(2), 279–288 (1997). http://www.sciencedirect.com/science/article/pii/S0021999197956539
2. Achenbach, J.D.: Wave propagation in elastic solids North-Holland (1973)
3. Albright, J., Epshteyn, Y., Medvinsky, M., Xia, Q.: High-order numerical schemes based on difference potentials for 2D elliptic problems with material interfaces. Appl. Numer. Math. **111**, 64–91 (2017). https://doi.org/10.1016/j.apnum.2016.08.017. http://linkinghub.elsevier.com/retrieve/pii/S0168927416301672
4. Appelö, D., Petersson, N.A.: A stable finite difference method for the elastic wave equation on complex geometries with free surfaces. Commun. Comput. Phys. **5**(1), 84–107 (2009)
5. Appelö, D., Petersson, N.A.: A fourth-order accurate embedded boundary method for the wave equation. SIAM J. Sci. Comput. **34**(6), A2982–A3008 (2012). http://epubs.siam.org/doi/abs/10.1137/09077223X
6. Arndt, D., Bangerth, W., Davydov, D., Heister, T., Heltai, L., Kronbichler, M., Maier, M., Pelteret, J.P., Turcksin, B., Wells, D.: The deal II library, version 8.5. J. Numer. Math **25**(3), 137–146 (2017). https://doi.org/10.1515/jnma-2016-1045
7. Burman, E.: Ghost penalty. Comptes Rendus Mathematique **348**(21-22), 1217–1220 (2010). https://doi.org/10.1016/j.crma.2010.10.006
8. Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A.: Cut FEM: discretizing geometry and partial differential equations. International Journal for Numerical Methods in Engineering **104**(7), 472–501 (2015). https://doi.org/10.1002/nme.4823
9. Burman, E., Hansbo, P.: Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. Applied Numerical Mathematics **62**(4), 328–341 (2012). https://doi.org/10.1016/j.apnum.2011.01.008
10. De Basabe Jonás D., Sen Mrinal, K., Wheeler, M.F.: The interior penalty discontinuous Galerkin method for elastic wave propagation: grid dispersion. Geophysical Journal International **175**(1), 83–93 (2008). https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-246X.2008.03915.x
11. Duru, K., Virta, K.: Stable and high order accurate difference methods for the elastic wave equation in discontinuous media. Journal of Computational Physics **279**, 37–62 (2014)
12. Hansbo, P., Larson, M.G., Larsson, K.: Cut finite element methods for linear elasticity problems. In: Bordas, S., Burman, E., Larson, M., Olshanskii, M. (eds.) Geometrically Unfitted Finite Element Methods and Applications, pp. 25–63, Lecture Notes in Computational Science and Engineering, vol. 121. Springer International Publishing, Cham (2017)
13. Johansson, A., Larson, M.G., Logg, A.: High order cut finite element methods for the stokes problem. Advanced Modeling and Simulation in Engineering Sciences **2**(1), 24 (2015)
14. Karniadakis, G.E., Sherwin, S.J.: Spectral/hp Element Methods for CFD. Oxford University Press, New York (1999). http://ebookcentral.proquest.com/lib/uu/detail.action?docID=241558
15. Kreiss, H.O., Oliger, J.: Comparison of accurate methods for the integration of hyperbolic equations. Tellus **24**(3), 199–215 (1972). https://doi.org/10.1111/j.2153-3490.1972.tb01547.x
16. Langtangen, H.P., Pedersen, G.K.: Scaling of Differential Equations. Springer, Berlin Heidelberg New York (2016). http://link.springer.com/10.1007/978-3-319-32726-6
17. Larson, M.G., Bengzon, F.: The Finite Element Method: Theory, Implementation and Applications, Text in Computational Science and Engineering vol. 10. Springer, Berlin Heidelberg (2013)
18. Lederer, P., Pfeiler, C.M., Wintersteiger, C., Lehrenfeld, C.: Higher order unfitted FEM for Stokes interface problems. Proceedings in Applied Mathematics and Mechanics **16**(1), 7–10 (2016). https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.201610003
19. Lombard, B., Piraux, J.: Numerical treatment of two-dimensional interfaces for acoustic and elastic waves. Journal of Computational Physics **195**(1), 90–116 (2004)
20. Lombard, B., Piraux, J., Gélis, C., Virieux, J.: Free and smooth boundaries in 2-D finite-difference schemes for transient elastic waves. Geophysical Journal International **172**(1), 252–261 (2008)
21. Massing, A., Larson, M.G., Logg, A., Rognes, M.E.: A stabilized Nitsche fictitious domain method for the Stokes problem. Journal of Scientific Computing **61**(3), 604–628 (2014). https://doi.org/10.1007/s10915-014-9838-9
22. Mönkölä, S.: Numerical simulation of fluid-structure interaction between acoustic and elastic waves. Ph.D. thesis, University of Jyväskylä, Jyväskylä ISBN 978-951-39-4439-1 (2011)

23. Nitsche, J.: Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg **36**(1), 9–15 (1971). https://doi.org/10.1007/BF02995904
24. Riviere, B., Wheeler, M.F.: Discontinuous finite element methods for acoustic and elastic wave problems. Contemporary Mathematics **329**, 271–282 (2003)
25. Saye, R.I.: High-Order Quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. SIAM Journal on Scientific Computing **37**(2), A993–A1019 (2015). https://doi.org/10.1137/140966290
26. Sticko, S., Kreiss, G.: A stabilized Nitsche cut element method for the wave equation. Computer Methods in Applied Mechanics and Engineering **309**, 364–387 (2016). https://doi.org/10.1016/j.cma.2016.06.001
27. Sticko, S., Kreiss, G.: Higher order cut finite elements for the wave equation. Journal of Scientific Computing **80**(3), 1867–1887 (2019). https://doi.org/10.1007/s10915-019-01004-2
28. Tan, S., Shu, C.W.: Inverse Lax–Wendroff procedure for numerical boundary conditions of hyperbolic equations: survey and new developments. In: Melnik, R., Kotsireas, I. (eds.) Advances in Applied Mathematics, Modeling, and Computational Science, pp. 41–63, Fields Institute Communications, vol. 66. Springer US, Boston (2013)
29. Tarantola, A.: Inversion of seismic reflection data in the acoustic approximation. Geophysics **49**(8), 1259–1266 (1984)
30. Tarantola, A.: Inversion of Travel Times and Seismic Waveforms Seismic Tomography, Pp. 135–157 Springer (1987)
31. Tarantola, A.: Theoretical background for the inversion of seismic waveforms including elasticity and attenuation. Pure and Applied Geophysics **128**(1-2), 365–399 (1988)
32. Virieux, J.: P-sv wave propagation in heterogeneous media: velocity-stress finite-difference method. Geophysics **51**(4), 889–901 (1986)
33. Virta, K., Appelö, D.: Formulae and software for particular solutions to the elastic wave equation in curved geometries (2015)
34. Zahradník, J.í., Moczo, P., Hron, F.e.: Testing four elastic finite-difference schemes for behavior at discontinuities. Bulletin of the Seismological Society of America **83**(1), 107–129 (1993)

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