Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 395 (2022) 114971
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Modeling wave propagation in elastic solids via high-order accurate implicit-mesh discontinuous Galerkin methods

Vincenzo Gulizzi[a,*], Robert Saye[b]

[a] *Center for Computational Sciences and Engineering (CCSE), Lawrence Berkeley National Laboratory, MS 50A-3111, 1 Cyclotron Rd, Berkeley, 94720, CA, United States*
[b] *Mathematics Group, Lawrence Berkeley National Laboratory, MS 50A-1148, 1 Cyclotron Rd, Berkeley, 94720, CA, United States*

Received 4 October 2021; received in revised form 12 January 2022; accepted 5 April 2022
Available online xxxx

Abstract

A high-order accurate implicit-mesh discontinuous Galerkin framework for wave propagation in single-phase and bi-phase solids is presented. The framework belongs to the embedded-boundary techniques and its novelty regards the spatial discretization, which enables boundary and interface conditions to be enforced with high-order accuracy on curved embedded geometries. High-order accuracy is achieved via high-order quadrature rules for implicitly-defined domains and boundaries, whilst a cell-merging strategy addresses the presence of small cut cells. The framework is used to discretize the governing equations of elastodynamics, written using a first-order hyperbolic momentum-strain formulation, and an exact Riemann solver is employed to compute the numerical flux at the interface between dissimilar materials with general anisotropic properties. The space-discretized equations are then advanced in time using explicit high-order Runge–Kutta algorithms. Several two- and three-dimensional numerical tests including dynamic adaptive mesh refinement are presented to demonstrate the high-order accuracy and the capability of the method in the elastodynamic analysis of single- and bi-phases solids containing complex geometries.

Keywords: Embedded-boundary methods; Implicitly-defined meshes; Discontinuous Galerkin methods; High-order accuracy; Elastodynamics

1. Introduction

The propagation of elastic waves is the subject of many research fields in science and engineering, such as geophysics [1], structural health monitoring [2] and metamaterials design [3,4]. In these areas, computational methods are well-established tools, especially when complicated geometries and/or heterogeneous materials are involved. However, a common burden in computational modeling is the generation of a high-quality mesh of the domains of analysis, which often results in the most laborious part of the development of a numerical scheme [5].

Generally, there are two approaches to meshing irregular domains. A common approach is to use body-fitted meshes, whereby the mesh elements are generated to conform to the boundaries of the domain; this approach can be flexible in resolving complex geometrical features but often requires a non-trivial effort to provide high-quality

* Corresponding author.
E-mail address: vgulizzi@lbl.gov (V. Gulizzi).

https://doi.org/10.1016/j.cma.2022.114971
0045-7825/

elements and may become more demanding for moving meshes or dynamic adaptive mesh refinements. On the other hand, one may use embedded-boundary (EB) methods, sometimes also referred to as cut-cell, immersed-boundary or fictitious-domain methods, where a curved geometry is represented on a regular background grid and the boundaries of the grid's cells do not need to conform to the boundaries of the geometry; this facilitates a more algorithmic or automated approach to mesh generation, data storage and adaptive mesh refinement (AMR), see e.g. Ref. [6,7]. Additionally, in EB methods, the majority of the mesh elements are regular elements and their properties (such as the mass matrix, its inverse and/or the interpolation operator for AMR applications) can be routinely, efficiently and accurately handled. However, it is clear that EB methods require additional considerations to handle the presence of curved geometries, especially if high-order accuracy is desired.

One of the most widely employed numerical techniques in solid and structural dynamics is the Finite Element Method (FEM) [8–10]. FEM models are typically based on body-fitted meshing strategies but have also been developed in combination with EB approaches. A notable example is the Finite Cell Method (FCM) [11–14] for two- and three-dimensional problems. In the FCM, the integrand functions used to evaluate all elemental quantities of a standard FEM (such as the mass matrix, the stiffness matrix or the body forces) are multiplied by an indicator function $\psi(\boldsymbol{x})$, which is introduced to define the embedded domain $\mathscr{D}$ as $\psi(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in \mathscr{D}$ and $\psi(\boldsymbol{x}) = \psi_0$, with $0 < \psi_0 \ll 1$, for $\boldsymbol{x} \notin \mathscr{D}$. This approach simplifies the mesh generation procedure but requires suitable integration schemes, or quadrature rules, for those elements where the indicator function jumps from 1 to ψ_0 and vice versa. In Ref. [15], the authors compared different integration approaches for the FCM and concluded that the most effective strategy to compute the integrals was to partition the elements into quadtree or octree subgrids (in 2D or 3D, respectively) suitably refined in proximity of the embedded boundaries. Indeed, the accurate evaluation of the elemental matrices in EB methods is not a simple task, particularly in 3D. The use of high-order quadrature rules for cut cells is one of the key aspects of the method presented in this work, as discussed shortly.

Various modifications of and/or alternatives to the FEM have been proposed in the literature to reduce the meshing effort. A few examples are the extended FEM [16,17] and the extended Ritz method [18], whereby the space of basis functions is enriched to automatically account for the presence of embedded boundaries or interfaces, the Virtual Element Method (VEM) [19], which allows the use of general (also non-convex) polygonal elements, or the Boundary Element Method (BEM) [20,21], which is based on an integral formulation and allows solving the equations of elastodynamics by discretizing the domain's boundaries only.

The discontinuous Galerkin (DG) method has also proven to be a very powerful and flexible numerical technique for solving different classes of PDEs [22,23]. In a DG method, the numerical solution is represented in a space of polynomial basis functions that are discontinuous among the mesh elements. Then, the inter-element continuity, the interface conditions between dissimilar materials and the boundary conditions are enforced in weak sense by introducing suitable boundary integrals. This naturally enables high-order accuracy, the treatment of generally-shaped elements and *hp* AMR with conventional and non-conventional (e.g. polytopic) mesh elements [24–28]. Moreover, unlike other numerical schemes based on continuous approximations, DG methods feature block-structured mass matrices, which are highly desirable in explicit time-stepping schemes and parallel computations as they can be easily inverted on an element-by-element basis. Thus, thanks to its discontinuous nature, the DG method has been employed in combination with the EB approach for high-order accurate solution of elliptic PDEs [29,30], incompressible and compressible fluid flow [31,32], and statics of thin structures with cut-outs [33], among many other applications. However, in the context of solving elastodynamics problems, while the literature offers several DG schemes using body-fitted meshes, which can be classified into those derived from the second-order hyperbolic formulations [34–36], those derived from the first-order hyperbolic formulations [37–40] and unified approaches [41], DG methods using embedded boundaries appear less investigated. Recently, Tavelli et al. [42] proposed a diffused-interface DG scheme for elastic waves where, similar to the FCM, the curved geometry is represented by an indicator function that takes value 1 within the solid and 0 outside the solid; as discussed by the authors, their scheme is high-order accurate far from the embedded boundary but only first-order accurate in the transition region of the indicator function, i.e. in proximity of the embedded boundary. Although high-order accurate embedded-boundary methods for the 2D scalar wave equation [43] and the 2D acoustic equation [44] have been developed, to the best of the authors' knowledge, a high-order accurate embedded-boundary method for elastic wave propagation in two- and three-dimensional geometries has not before been investigated in the literature.

In this work, we present high-order accurate DG methods for elastodynamics in embedded geometries. In prior work, these kinds of DG methods have been successfully employed to model with high-order accuracy free-surface flow and rigid body-fluid interaction in the incompressible regime [30,31,45,46], the static response of

thin multilayered structures with cut-outs [33,47,48], gas dynamics problems [32]; here, the methods are extended to model elastic waves propagating in single- and bi-phase solids characterized by general anisotropic behavior. Geometries are represented via a level set function, whose zero-contour denotes either the curved boundary of a single-phase solid or the interface between the phases of a two-phase solid. A key feature of the proposed approach is the use of high-order quadrature rules for implicitly-defined domains and boundaries stemming from the intersection between the grids and the level set function. These quadrature rules are generated using the algorithm developed in Ref. [49] and enable the resolution of the embedded geometry as well as the enforcement of boundary and interface conditions with high-order accuracy. To avoid the presence of arbitrarily small cut cells, which would lead to overly restrictive times steps and ill-conditioned discrete operators, the framework developed in this work uses a cell-merging technique, whereby those cells with a volume fraction smaller than a user-defined threshold are merged with their neighbors. The DG method is used to discretize the governing equations of elastodynamics in space and an exact Riemann solver [40] is employed to compute the numerical flux at the interface between materials with dissimilar elastic properties. Meanwhile, explicit high-order Runge–Kutta algorithms [23] serve as time integrators.

The paper is organized as follows: Section 2 introduces the momentum-strain formulation of elastodynamics for single- and bi-phase solids considered in this work; Section 3 presents the implicit-mesh discontinuous Galerkin framework including the generation of the implicitly-defined meshes, the weak form of the governing equations and the hp adaptive mesh refinement; Section 4 demonstrates high-order accuracy and the capability of the method by discussing numerical results obtained for several two- and three-dimensional wave propagation problems in single- and bi-phase solids. Conclusions and discussions for further developments are given in Section 5.

2. Elastodynamic formulation

2.1. Geometry description

The geometry of a single-phase or a bi-phase solid is described implicitly by a level set function. Consider a d-dimensional rectangle $\mathscr{R} \subset \mathbb{R}^d$ and its outer boundary $\partial\mathscr{R}$. Consider also a level set function $\varphi : \mathscr{R} \to \mathbb{R}$ and let $\mathscr{R}^- \equiv \{\boldsymbol{x} \in \mathscr{R} : \varphi(\boldsymbol{x}) < 0\}$ be the portion of $\mathscr{R}$ where φ is negative, $\mathscr{R}^+ \equiv \{\boldsymbol{x} \in \mathscr{R} : \varphi(\boldsymbol{x}) > 0\}$ be the portion of $\mathscr{R}$ where φ is positive and $\mathscr{L} \equiv \{\boldsymbol{x} \in \mathscr{R} : \varphi(\boldsymbol{x}) = 0\}$ be the zero-contour of φ. Moreover, let $\partial\mathscr{R}^- \equiv \{\boldsymbol{x} \in \partial\mathscr{R} : \varphi(\boldsymbol{x}) < 0\}$ be the portion of $\partial\mathscr{R}$ where φ is negative and $\partial\mathscr{R}^+ \equiv \{\boldsymbol{x} \in \partial\mathscr{R} : \varphi(\boldsymbol{x}) > 0\}$ be the portion of $\partial\mathscr{R}$ where φ is positive.

Then, we define the domain $\mathscr{D}_\alpha$ and the boundary $\partial\mathscr{D}_\alpha$ of a solid consisting of one phase α as $\mathscr{D}_\alpha \equiv \mathscr{R}^-$ and $\partial\mathscr{D}_\alpha \equiv \mathscr{B}_\alpha \cup \mathscr{L}_\alpha$, respectively, where $\mathscr{B}_\alpha \equiv \partial\mathscr{R}^-$ and $\mathscr{L}_\alpha \equiv \mathscr{L}$. Alternatively, for a bi-phase solid consisting of two distinct phases α and β, we define the domain and the outer boundary of the phase α as $\mathscr{D}_\alpha \equiv \mathscr{R}^-$ and $\mathscr{B}_\alpha \equiv \partial\mathscr{R}^-$, respectively, the domain and the outer boundary of the phase β as $\mathscr{D}_\beta \equiv \mathscr{R}^+$ and $\mathscr{B}_\beta \equiv \partial\mathscr{R}^+$, respectively, and the interface between the two phases as $\mathscr{L}_{\alpha,\beta} \equiv \mathscr{L}$.

As an example, Fig. 1a shows a level set function φ defined over a square, while Fig. 1b shows the domain and the boundaries of the corresponding two-dimensional single-phase solid, and Fig. 1c shows the domains, boundaries and interface of the corresponding two-dimensional bi-phase solid.

2.2. Governing equations

The governing equations of linear elastodynamics can be stated using different formulations, which, on the basis of the chosen set of primary variables, include the displacement formulation, see e.g. [9,10], the velocity-stress formulation, see e.g. [38], or the momentum-strain formulation, see e.g. [40]. Here, we use the momentum-strain formulation since it allows for the consideration of materials with space-dependent constitutive properties. In the following, the governing equations, the initial conditions and the boundary conditions, are written for the domain $\mathscr{D}_\alpha$ and its boundary $\partial\mathscr{D}_\alpha$, but are also valid for the domain $\mathscr{D}_\beta$ and its boundary $\partial\mathscr{D}_\beta$, if one replaces the subscript α with β. However, note that both subscripts α and β appear explicitly in the equations governing the interface conditions for bi-phase solids discussed at the end of this section.

Let $\boldsymbol{v}_\alpha$ and $\boldsymbol{m}_\alpha$ denote the velocity field and the momentum field, respectively, and let $\boldsymbol{\gamma}_\alpha$ and $\boldsymbol{\sigma}_\alpha$ denote the strain field and the stress field, respectively. In $\mathbb{R}^d$, upon defining $N_v \equiv d$ and $N_\sigma \equiv d(d+1)/2$, $\boldsymbol{v}_\alpha$ and $\boldsymbol{m}_\alpha$ are N_v-dimensional vectors, whereas $\boldsymbol{\gamma}_\alpha$ and $\boldsymbol{\sigma}_\alpha$ are N_σ-dimensional vectors containing the strain and the stress

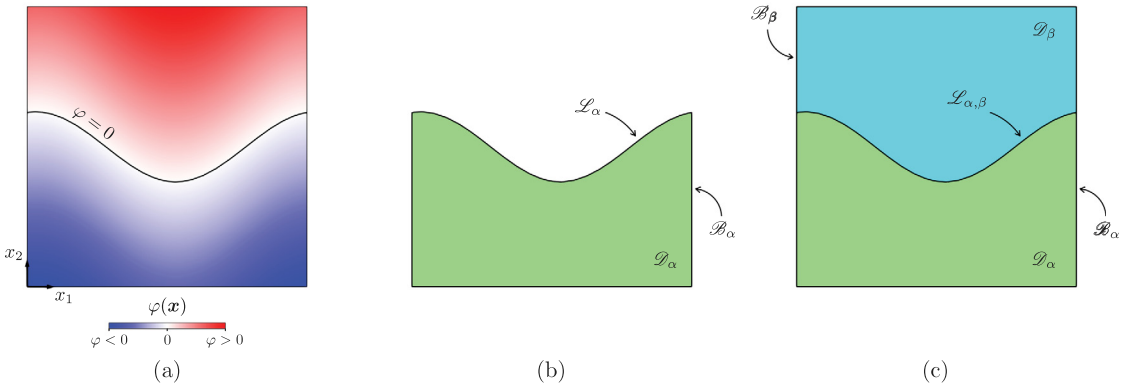

Fig. 1. (a) Sample level set function φ partitioning a square into a region where φ is negative and a region where φ is positive. (b) Single-phase solid and (c) bi-phase solid implicitly defined using the level set function of figure (a).

components, respectively, in Voigt notation [50]; for example, in 2D we have $\boldsymbol{\gamma}_\alpha = (\gamma_{\alpha 11}, \gamma_{\alpha 22}, \gamma_{\alpha 12})^\intercal$, while in 3D we have $\boldsymbol{\gamma} = (\gamma_{\alpha 11}, \gamma_{\alpha 22}, \gamma_{\alpha 33}, \gamma_{\alpha 23}, \gamma_{\alpha 13}, \gamma_{\alpha 12})^\intercal$, where $\bullet^\intercal$ denotes the transpose of $\bullet$.

The domain $\mathscr{D}_\alpha$ is characterized by the density ρ_α, such that $\boldsymbol{m}_\alpha = \rho_\alpha \boldsymbol{v}_\alpha$, and by the (positive-definite) $N_\sigma \times N_\sigma$ matrix $\boldsymbol{c}_\alpha$ of elastic stiffness constants that links the stress and strain via the general Hooke's law $\boldsymbol{\sigma}_\alpha = \boldsymbol{c}_\alpha \boldsymbol{\gamma}_\alpha$. Then, in the momentum-strain formulation, the governing equations of elastodynamics are written as the following first-order hyperbolic system of PDEs

$$\frac{\partial \boldsymbol{U}_\alpha}{\partial t} + \frac{\partial \boldsymbol{F}_{\alpha i}}{\partial x_i} = \boldsymbol{S}_\alpha. \tag{1}$$

In Eq. (1) and in the remainder of the paper, the subscript i takes values in $\{1, \ldots, d\}$ and implies summation when repeated, t is the time, and x_i is the ith coordinate of the d-dimensional space location vector $\boldsymbol{x}$. Additionally, $\boldsymbol{U}_\alpha$, $\boldsymbol{F}_{\alpha i}$ and $\boldsymbol{S}_\alpha$ are N_U-dimensional vectors, with $N_U \equiv N_v + N_\sigma$, representing the conserved variables, the fluxes in the i direction and the source terms, respectively; they are defined as

$$\boldsymbol{U}_\alpha \equiv \begin{pmatrix} \boldsymbol{m}_\alpha \\ \boldsymbol{\gamma}_\alpha \end{pmatrix}, \quad \boldsymbol{F}_{\alpha i} \equiv \begin{pmatrix} -\boldsymbol{I}_i^\intercal \boldsymbol{c}_\alpha \boldsymbol{\gamma}_\alpha \\ -\rho_\alpha^{-1} \boldsymbol{I}_i \boldsymbol{m}_\alpha \end{pmatrix} \quad \text{and} \quad \boldsymbol{S}_\alpha \equiv \begin{pmatrix} \rho_\alpha \boldsymbol{b}_\alpha \\ \boldsymbol{0} \end{pmatrix}, \tag{2}$$

where $\boldsymbol{b}$ is the d-dimensional vector of body forces and the matrices $\boldsymbol{I}_i$ are given by

$$\boldsymbol{I}_1 \equiv \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{I}_2 \equiv \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{3}$$

in 2D, and by

$$\boldsymbol{I}_1 \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{I}_2 \equiv \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{I}_3 \equiv \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{4}$$

in 3D.

Eq. (1) is assumed to be valid for $(t, \boldsymbol{x}) \in \mathscr{T} \times \mathscr{D}_\alpha$, where $\mathscr{T} \equiv [0, T]$ is the time interval and T is the final time, and is supplemented by initial, boundary and interface conditions. Initial conditions are given as

$$\boldsymbol{U}_\alpha = \boldsymbol{U}_{\alpha 0}(\boldsymbol{x}) \quad \text{for } t = 0 \text{ and } \boldsymbol{x} \in \mathscr{D}_\alpha, \tag{5}$$

where $\boldsymbol{U}_{\alpha 0}(\boldsymbol{x})$ contains the known values of $\boldsymbol{U}_\alpha$ at $t = 0$.

Boundary conditions are prescribed at the outer boundary of the domain $\mathscr{D}_\alpha$. Here, we consider three types of boundary conditions, namely (*i*) prescribed values of the velocity field $\boldsymbol{v}$, (*ii*) prescribed values of the traction field $\boldsymbol{t}$

and (*iii*) absorbing boundary conditions, which are typical for elastodynamics problems of scientific and engineering interest including material characterization and damage detection. Prescribing the value of the velocity field $\boldsymbol{v}$ is equivalent to prescribing the momentum field $\boldsymbol{m}$ as

$$\boldsymbol{m}_\alpha = \rho_\alpha \overline{\boldsymbol{v}} \quad \text{for } (t, \boldsymbol{x}) \in \mathscr{T} \times \partial \mathscr{D}_{\alpha v}, \tag{6}$$

where $\overline{\boldsymbol{v}}$ is the prescribed value of the velocity field, which includes $\overline{\boldsymbol{v}} = \boldsymbol{0}$ in case of a fixed boundary, and $\partial \mathscr{D}_{\alpha v}$ is the portion of the outer boundary of $\mathscr{D}_\alpha$ where $\overline{\boldsymbol{v}}$ is prescribed. Prescribing the value of the traction field $\boldsymbol{t}$ is expressed in terms of the strain $\boldsymbol{\gamma}_\alpha$ as

$$\boldsymbol{I}_{n_\alpha}^\intercal \boldsymbol{c}_\alpha \boldsymbol{\gamma}_\alpha = \overline{\boldsymbol{t}} \quad \text{for } (t, \boldsymbol{x}) \in \mathscr{T} \times \partial \mathscr{D}_{\alpha t}, \tag{7}$$

where $\overline{\boldsymbol{t}}$ is the prescribed value of the traction field, which includes $\overline{\boldsymbol{t}} = \boldsymbol{0}$ in case of a free boundary, $\partial \mathscr{D}_{\alpha t}$ is the portion of the outer boundary of $\mathscr{D}_\alpha$ where $\overline{\boldsymbol{t}}$ is prescribed and $\boldsymbol{I}_{n_\alpha} \equiv n_{\alpha i} \boldsymbol{I}_i$, being $n_{\alpha i}$ the ith component of the outer unit normal of $\partial \mathscr{D}_{\alpha t}$. Absorbing boundary conditions refer to a boundary that does not reflect the incoming waves.

Finally, for bi-phase solids only, interface conditions are prescribed at the interface $\mathscr{L}_{\alpha,\beta}$. Here, we consider perfect interface conditions, i.e. continuity of the velocity field and equilibrium of the traction field, which are given in terms of momentum and strain as follows

$$\begin{cases} \rho_\alpha^{-1} \boldsymbol{m}_\alpha = \rho_\beta^{-1} \boldsymbol{m}_\beta \\ \boldsymbol{I}_{n_\alpha}^\intercal \boldsymbol{c}_\alpha \boldsymbol{\gamma}_\alpha + \boldsymbol{I}_{n_\beta}^\intercal \boldsymbol{c}_\beta \boldsymbol{\gamma}_\beta = \boldsymbol{0} \end{cases} \quad \text{for } (t, \boldsymbol{x}) \in \mathscr{T} \times \mathscr{L}_{\alpha,\beta}. \tag{8}$$

Within the present DG framework, the three types of boundary conditions (including the absorbing boundary conditions) and the interface conditions are enforced via suitable definitions of the numerical flux as proposed by Zhan et al. [40]. The numerical flux is a key ingredient of DG formulations and will be introduced in Section 3.

3. Implicit-mesh discontinuous Galerkin methods

3.1. Implicitly-defined meshes

The discontinuous Galerkin method typically requires a suitable mesh of the domain under analysis. Here, we use the implicitly-defined mesh technique [30–32], whereby the domain discretization is obtained by intersecting the implicitly-defined phases and a structured background grid that is easily generated for the rectangle $\mathscr{R}$ containing the solid.

Consider a structured grid $\mathscr{G} \equiv \bigcup_{\boldsymbol{j}} \mathscr{C}^{\boldsymbol{j}}$, where $\mathscr{C}^{\boldsymbol{j}} \equiv [x_1^{\boldsymbol{j}}, x_1^{\boldsymbol{j}} + h_1] \times \cdots \times [x_d^{\boldsymbol{j}}, x_d^{\boldsymbol{j}} + h_d] \subset \mathscr{R}$ is a d-dimensional rectangular cell, $\boldsymbol{j}$ is the d-tuple identifying the location of the cell within the grid and $x_i^{\boldsymbol{j}}$ and h_i are the cell's lower end and the cell's size in the ith direction, respectively. Each cell $\mathscr{C}$ is intersected with the implicitly-defined phases of the solid and is classified on the basis of its volume fraction f_α given by

$$f_\alpha \equiv \frac{1}{V_{\mathscr{C}}} \int_{\mathscr{C} \cap \mathscr{D}_\alpha} 1 \, \mathrm{d}V, \tag{9}$$

where $V_{\mathscr{C}}$ is the volume of the cell, that is $V_{\mathscr{C}} = h_1 h_2$ in 2D and $V_{\mathscr{C}} = h_1 h_2 h_3$ in 3D. Referring to the phase α, *entire* cells are those cells falling entirely inside $\mathscr{D}_\alpha$ and having volume fraction $f_\alpha = 1$; *empty* cells are those cells falling entirely outside $\mathscr{D}_\alpha$ and having volume fraction $f_\alpha = 0$; *large* cells are those cells cut by $\mathscr{L}$ and having volume fraction $\overline{f} < f_\alpha < 1$; *small* cells are those cells cut by $\mathscr{L}$ and having volume fraction $0 < f_\alpha \leq \overline{f}$. The same classification is performed by intersecting the cells with the phase β in bi-phase solids. Henceforth, entire and large cells are collectively referred to as *primary* cells.

In the classification above, the parameter $\overline{f}$ denotes a user-defined volume fraction threshold introduced to identify the small cells, i.e. those cells whose presence would lead to overly-small time-step restrictions and ill-conditioned discrete operators. Here, to address the small-cell problem, we employ a cell-merging strategy whereby small cells are merged with their neighbors. In particular, each small cell is merged with one primary cell among the neighboring cells of its 3×3 neighborhood, in 2D, or $3 \times 3 \times 3$ neighborhood, in 3D. To select the neighbor for merging, the neighboring cells are grouped in the following order: in 3D, we consider first the cells sharing a face with the small cell, second the cells sharing an edge with the small cell and third the cells sharing a corner with

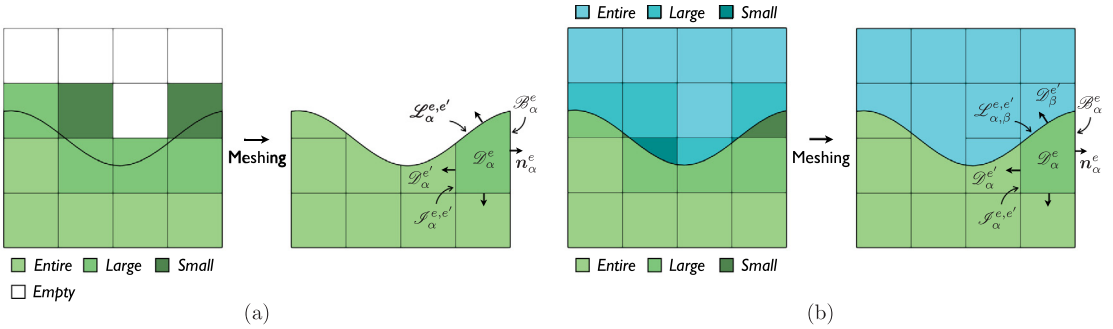

Fig. 2. Cell classification and implicitly-defined meshes for (a) the single-phase solid and (b) the bi-phase solid shown in Fig. 1 intersected with a 4×4 background grid.

the small cell; within each group the neighboring cells are ordered according to their volume fraction. Then, the neighbor targeted for merging is the first cell in the first non-empty group. In 2D, the search for the target neighbor starts from the neighboring cells that share an edge with the small cell. A few comments are in order:

- Although each small cell is merged with one primary cell only, multiple small cells are allowed to target the same primary cell, possibly leading to a cluster of multiple cells merged together.
- The cell-merging procedure is simple and cheap to implement, in both 2D and 3D. In essence, once the volume fractions are precomputed (using the quadrature schemes discussed in Section 3.3), this algorithm entails a simple sequence of logical boolean checks. As such, the merging procedure in 3D presents no unexpected additional costs over the 2D case.
- The volume fraction threshold is chosen based on previous studies regarding implicit-mesh DG methods, see, e.g., Refs. [30–32], which empirically show that values of threshold between 0.2 and 0.5 perform well in a wide variety of circumstances including moving adaptively-refined meshes. For further theoretical and experimental analyses on the effect of the volume fraction threshold in the context of embedded-boundary DG methods for elliptic PDEs, the interested reader is also referred to Ref. [29].
- In general, it is possible that a small cell does not find a suitable neighboring cell for merging. This typically happens when the interfacial/boundary geometry is unresolved, i.e., when it is so complex that the background grid cannot properly resolve the geometric features. Unresolved geometries are not considered in the present implementation. Instead, in all the presented numerical tests, the background grid is always chosen fine enough that the proposed cell-merging strategy never fails.

Once the cell-merging procedure is completed, the cells of the grid consist of *non-merged* cells, i.e. the primary cells that have not been targeted during the merging process, *merged* cells, i.e. the union of small cells and their merging neighbors, and empty cells. Then, the implicitly-defined mesh $\mathcal{M}_\alpha$ of the phase α is written $\mathcal{M}_\alpha \equiv \bigcup_{e=1}^{N_\alpha^e} \mathscr{D}_\alpha^e$, where $\mathscr{D}_\alpha^e$ is the eth implicitly-defined mesh element of the phase α and N_α^e is the number of mesh elements. Finally, the implicitly-defined mesh $\mathcal{M}$ of a single-phase solid coincides with the implicitly-defined mesh of the phase α, i.e. $\mathcal{M} \equiv \mathcal{M}_\alpha$, whereas, the implicitly-defined mesh $\mathcal{M}$ of a bi-phase solid is the collection of mesh elements of the phase α and the phase β, i.e. $\mathcal{M} \equiv \mathcal{M}_\alpha \bigcup \mathcal{M}_\beta$. Ultimately, these implicitly-defined meshes consist of a collection of standard d-dimensional rectangular elements and a relatively smaller number of curved elements that conform with the curvature of the zero-contour of the level set function.

Figs. 2 and 3 show a few implicitly-defined meshes obtained with the procedure described above. Fig. 2a shows the classification of the cells of a 4×4 grid when intersected with the single-phase solid of Fig. 1b and the corresponding implicitly-defined mesh; Fig. 2a also highlights an implicitly-defined element $\mathscr{D}_\alpha^e$ and its outer boundary $\mathscr{B}_\alpha^e \cup \mathscr{L}_\alpha^e$, intraphase boundary $\mathscr{I}_\alpha^{e,e'}$ shared with the neighboring element $\mathscr{D}_\alpha^{e'}$, and outer unit normal $\boldsymbol{n}_\alpha^e$. Similarly, Fig. 2b shows the classification of the cells of a 4×4 grid when intersected with the bi-phase solid of Fig. 1c and the corresponding implicitly-defined mesh; Fig. 2b also highlights an implicitly-defined element $\mathscr{D}_\alpha^e$ and its outer boundary $\mathscr{B}_\alpha^e$, intraphase boundary $\mathscr{I}_\alpha^{e,e'}$ shared with the neighboring element $\mathscr{D}_\alpha^{e'}$ of the same phase, interface boundary $\mathscr{L}_{\alpha,\beta}^{e,e'}$ shared with the neighboring element $\mathscr{D}_\beta^{e'}$ of the phase β, and outer unit normal $\boldsymbol{n}_\alpha^e$. Finally,

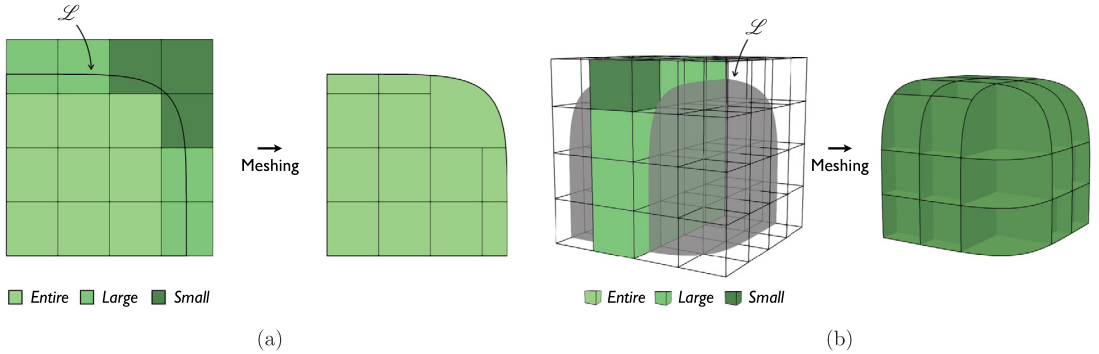

Fig. 3. Examples of (a) a two-dimensional implicitly-defined mesh and (b) a three-dimensional implicitly-defined mesh where multiple small cells are merged with the same cell.

Fig. 3 shows a 2D example and a 3D example of implicitly-defined meshes where multiple small cells have targeted the same nearby cell for merging.

3.2. Discontinuous Galerkin formulation

Once the mesh of the domain is generated, discontinuous Galerkin formulations are derived by introducing of a suitable space of discontinuous basis functions and stating the governing equations in weak form. Here, given that the discretization is constructed using Cartesian grids and the majority of the mesh elements are standard d-dimensional rectangles, it is natural to define the local basis functions as tensor-product polynomials.

Let $\mathscr{D}^e$ be a mesh element of $\mathcal{M}$ associated with the primary grid cell $\mathscr{C}^j$ (and all the small cells that are merged with $\mathscr{C}^j$) and let $\mathcal{P}^e_{hp}$ be the space of tensor-product polynomials of degree p in the (hyper) rectangular volume occupied by $\mathscr{C}^j$. Then, the space $\mathcal{V}_{hp}$ of discontinuous basis functions for the mesh $\mathcal{M}$ based on the grid $\mathscr{G}$ is

$$\mathcal{V}_{hp} \equiv \left\{ v : \mathscr{G} \to \mathbb{R} \mid v|_{\mathscr{D}^e} \in \mathcal{P}^e_{hp},\ \forall \mathscr{D}^e \in \mathcal{M} \right\}, \tag{10}$$

while the related space $\mathcal{V}^N_{hp}$ of discontinuous polynomials vector fields is $\mathcal{V}^N_{hp} \equiv (\mathcal{V}_{hp})^N$.

For a single-phase solid, the weak form of the governing equations is obtained by multiplying Eq. (1) by the test functions $\boldsymbol{V} \in \mathcal{V}^{N_U}_{hp}$, integrating over a generic mesh element $\mathscr{D}^e_\alpha$ and performing integration by parts in space, which yield

$$\int_{\mathscr{D}^e_\alpha} \boldsymbol{V}^\intercal \frac{\partial \boldsymbol{U}_\alpha}{\partial t} \mathrm{d}V = \int_{\mathscr{D}^e_\alpha} \boldsymbol{V}^\intercal \boldsymbol{S}_\alpha \mathrm{d}V + \int_{\mathscr{D}^e_\alpha} \frac{\partial \boldsymbol{V}^\intercal}{\partial x_i} \boldsymbol{F}_{\alpha i} \mathrm{d}V - \int_{\mathscr{B}^e_\alpha \cup \mathscr{L}^e_\alpha} \boldsymbol{V}^\intercal \widehat{\boldsymbol{F}}_n \mathrm{d}S - \sum_{e' \in \mathcal{N}^e_\alpha} \int_{\mathscr{I}^{e,e'}_\alpha} \boldsymbol{V}^\intercal \widehat{\boldsymbol{F}}_n \mathrm{d}S. \tag{11}$$

Similarly, for a bi-phase solid, one obtains

$$\begin{aligned} \int_{\mathscr{D}^e_\alpha} \boldsymbol{V}^\intercal \frac{\partial \boldsymbol{U}_\alpha}{\partial t} \mathrm{d}V = & \int_{\mathscr{D}^e_\alpha} \boldsymbol{V}^\intercal \boldsymbol{S}_\alpha \mathrm{d}V + \int_{\mathscr{D}^e_\alpha} \frac{\partial \boldsymbol{V}^\intercal}{\partial x_i} \boldsymbol{F}_{\alpha i} \mathrm{d}V + \\ & - \int_{\mathscr{B}^e_\alpha} \boldsymbol{V}^\intercal \widehat{\boldsymbol{F}}_n \mathrm{d}S - \sum_{e' \in \mathcal{N}^e_\alpha} \int_{\mathscr{I}^{e,e'}_\alpha} \boldsymbol{V}^\intercal \widehat{\boldsymbol{F}}_n \mathrm{d}S - \sum_{e' \in \mathcal{N}^e_{\alpha,\beta}} \int_{\mathscr{L}^{e,e'}_{\alpha,\beta}} \boldsymbol{V}^\intercal \widehat{\boldsymbol{F}}_n \mathrm{d}S. \end{aligned} \tag{12}$$

Note that Eq. (12) is valid for a mesh element $\mathscr{D}^e_\beta$ of $\mathcal{M}_\beta$ if one switches the subscripts α and β and considers that $\mathscr{L}^{e,e'}_{\beta,\alpha}$ coincides with $\mathscr{L}^{e,e'}_{\alpha,\beta}$ but has opposite unit normal. In Eqs. (11) and (12), $\mathcal{N}^e_\alpha$ denotes the set of mesh elements of $\mathcal{M}_\alpha$ that are neighbors of $\mathscr{D}^e_\alpha$, $\mathcal{N}^e_{\alpha,\beta}$ denotes the set of mesh elements of $\mathcal{M}_\beta$ that are neighbors of $\mathscr{D}^e_\alpha$, and $\widehat{\boldsymbol{F}}_n$ is the so-called *numerical flux*. The expression of $\widehat{\boldsymbol{F}}_n$ depends on: the solution state $\boldsymbol{U}^e_\alpha$ and the boundary conditions at $\mathscr{B}^e_\alpha \cup \mathscr{L}^e_\alpha$, the adjacent solution states $\boldsymbol{U}^e_\alpha$ and $\boldsymbol{U}^{e'}_\alpha$ of neighboring mesh elements of the same phase at $\mathscr{I}^{e,e'}_\alpha$, and the adjacent solution states $\boldsymbol{U}^e_\alpha$ and $\boldsymbol{U}^{e'}_\beta$ of neighboring mesh elements of different phases at $\mathscr{L}^{e,e'}_{\alpha,\beta}$. In all cases mentioned above, the expression of the numerical flux used in this work is based on the exact Riemann

solver developed by Zhan et al. [40] for anisotropic elastodynamics wherein adjacent elements are allowed to have different values of density and/or elastic stiffness constants.

The final semidiscrete evolution equation is obtained by expressing $\boldsymbol{U}$ over each mesh element as a linear combination of the spatial basis functions with time-dependent coefficients. Using a compact notation, this is written as

$$\boldsymbol{U}(t,\boldsymbol{x}) = \mathbb{B}^e_\alpha(\boldsymbol{x})\mathbb{X}^e_\alpha(t), \quad \text{for } \boldsymbol{x} \in \mathscr{D}^e_\alpha, \tag{13}$$

where $\mathbb{B}^e_\alpha$ is a $N_U \times N_U N_p$ matrix containing the basis functions, $\mathbb{X}^e_\alpha$ is the vector of coefficients of length $N_U N_p$ and $N_p \equiv (1+p)^d$ is the number of basis functions. For a bi-phase solid, substituting Eq. (13) into Eq. (12) and letting $\boldsymbol{V}$ range over the basis functions, one obtains

$$\mathbb{M}^e_\alpha \dot{\mathbb{X}}^e_\alpha = \int_{\mathscr{D}^e_\alpha} \mathbb{B}^{e\intercal}_\alpha \boldsymbol{S}_\alpha \mathrm{d}V + \int_{\mathscr{D}^e_\alpha} \frac{\partial \mathbb{B}^{e\intercal}_\alpha}{\partial x_i} \boldsymbol{F}_{\alpha i} \mathrm{d}V - \int_{\mathscr{B}^e_\alpha} \mathbb{B}^{e\intercal}_\alpha \widehat{\boldsymbol{F}}_n \mathrm{d}S - \sum_{e' \in \mathcal{N}^e_\alpha} \int_{\mathscr{I}^{e,e'}_\alpha} \mathbb{B}^{e\intercal}_\alpha \widehat{\boldsymbol{F}}_n \mathrm{d}S - \sum_{e' \in \mathcal{N}^e_{\alpha,\beta}} \int_{\mathscr{L}^{e,e'}_{\alpha,\beta}} \mathbb{B}^{e\intercal}_\alpha \widehat{\boldsymbol{F}}_n \mathrm{d}S, \tag{14}$$

where the superimposed dot denotes the time derivative and $\mathbb{M}^e_\alpha$ is the mass matrix of the element $\mathscr{D}^e_\alpha$ given by

$$\mathbb{M}^e_\alpha \equiv \int_{\mathscr{D}^e_\alpha} \mathbb{B}^{e\intercal}_\alpha \mathbb{B}^e_\alpha \mathrm{d}V. \tag{15}$$

An expression similar to the one given in Eq. (14) is obtained for single-phase solids if one substitutes Eq. (13) into Eq. (11).

3.3. High-order quadrature rules for implicitly-defined elements

In Eqs. (14) and (15), several volumetric and boundary integrals need to be evaluated. For the entire elements and those cell boundaries that are not cut by the embedded boundaries, see e.g. Fig. 2, these integrals are evaluated with high-order accuracy using tensor-product Gauss–Legendre quadrature rules. Meanwhile, to retain the high-order accuracy of the method in proximity of the embedded boundaries, suitable high-order integration schemes should be employed to evaluate the domains and boundary integrals of the cells cut by the zero-contour of the level set function. Here, we make use of the high-order accurate quadrature algorithms developed in [49]; an open-source implementation of these algorithms is also available [51]. A few examples of the kinds of quadrature schemes produced by these algorithms are shown in Fig. 4; it is worth stressing that the quadrature points are always inside the domain of integration and the quadrature weights are always strictly positive. The interested reader is referred to Ref. [49] for a detailed description of the algorithms generating the quadrature rules.

As a last comment, we note that some of the terms (e.g. the mass matrix) appearing in Eq. (14) involve the evaluation of the integral of polynomial functions. By exploiting the properties of homogeneous functions along with additional assumptions on the implicitly-defined geometry, these integrals may be transformed into lower-dimensional ones, thus reducing the computational cost associated with the numerical quadrature, see e.g. Refs. [52,53]. However, this is in general not possible and full quadrature schemes are required to evaluate, for example, the contribution of a non-polynomial source term or the integrals involving the numerical flux, which is obtained using the Riemann solver. Another possibility is to use quadrature compression techniques, see e.g., Ref. [54]; roughly speaking, these methods discard as many quadrature nodes as possible (modifying the weights of the remaining nodes) while preserving quadrature accuracy on a predetermined class of integrand functions. These methods could be useful in cases of highly complex interfacial/boundary geometry, e.g., in handling sub-grid feature. We note, however, in the present work, the quadrature algorithms of Ref. [49] always yielded efficient quadrature schemes with small-to-moderate node counts. In particular, we have not faced any problems concerning large quadrature node counts for the kinds of problems we have applied these DG methods to.

3.4. hp Adaptive mesh refinement

Owing to the discontinuous nature of DG methods, the present implicit-mesh DG framework can naturally be coupled to an *hp* adaptive mesh refinement strategy. To enable AMR capabilities, the framework is integrated

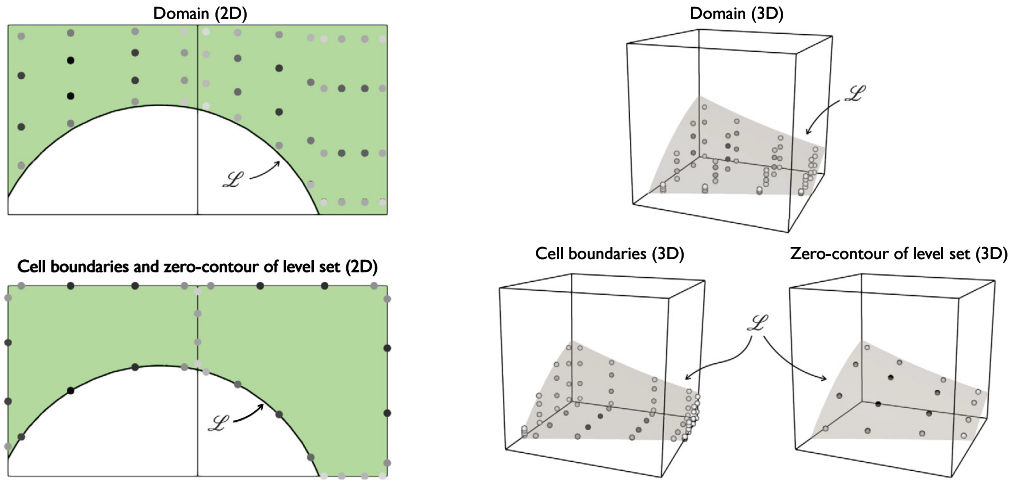

Fig. 4. Examples of the location of the quadrature points obtained using the algorithm of Ref. [49]; in the images, the quadrature points are colored according to their weight: light grays denote small weights while dark grays denote large weights.

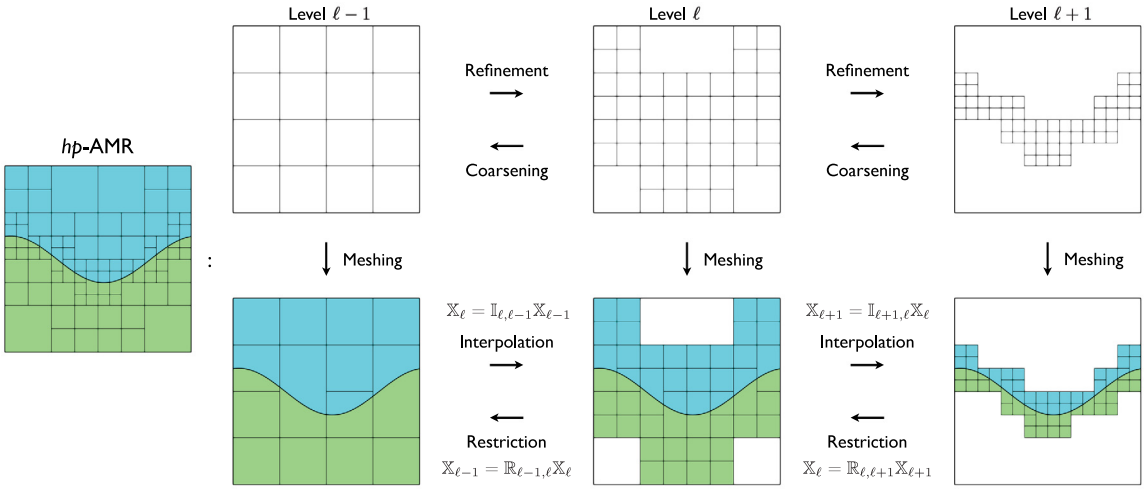

Fig. 5. Illustration of an *hp*-AMR configuration where the implicitly-defined meshes are generated from a three-level hierarchy of structured grids. The figure also sketches the operations performed by the interpolation operator $\mathbb{I}_{\ell+1,\ell}$ and the restriction operator $\mathbb{R}_{\ell,\ell+1}$.

into `AMReX` [55], an open-source software library (https://amrex-codes.github.io) with functionalities for writing massively parallel applications based on adaptive structured Cartesian grids. In `AMReX`, the AMR structure is represented as a hierarchy of overlapping levels of refinement ranging from the level identified by $\ell = 0$, which contains the coarsest grid, to the level identified by $\ell = \ell_{\text{finest}}$, which contains the finest grid. The grid at the level $\ell = 0$ is created at the beginning of the simulations and statically covers the entire domain of analysis; the grids at the levels $\ell > 0$ are created and destroyed dynamically based on user-defined refinement and coarsening criteria. `AMReX` has been mainly employed in combination with finite volume schemes, see Ref. [55] and the references therein, but recently has been extended to include the implicit-mesh DG framework, see Ref. [32]. Fig. 5 illustrates a typical multi-level mesh supported by the present implementation, where the final mesh shown on the leftmost end of the figure is obtained by overlapping three implicitly-defined meshes (the bottom row of images), which in turn are generated from a three-level hierarchy of structured grids (the top row of images) according to the procedure presented in Section 3.1. In the figure and in the remainder of the paper, the present AMR scheme is referred to as *hp*-AMR since it allows the use of different polynomial orders at different levels.

It is worth recalling some details about the operations associated with the use of adaptive mesh refinement, even though a thorough discussion regarding adaptive implicitly-defined meshes can be found in Refs. [30,56] as well as Ref. [32].

The evolution of the *hp*-AMR levels is governed by tagging and un-tagging operations, whereby a two-value variable, or tag, is assigned to each cell at each level. The tag determines whether the cell should be replaced by a set of finer cells or whether the existing finer resolution is not further required at that location. Various criteria might be considered to assign and evolve the tags of the cells, with examples ranging from static manual tagging to dynamic solution-dependent tagging. In this work, we evolve the tag of a generic cell on the basis of the DG solution at that cell. More specifically, with reference to the cell classification introduced in Section 3.1, a primary cell is tagged for refinement if the DG solution of the associated implicitly-defined element $\mathscr{D}^e$, which is identified by the coefficients of the element's basis functions contained in $\mathbb{X}^e$, satisfies a condition of the form

$$f_{\text{tag}}(\mathbb{X}^e) > 0, \tag{16}$$

where f_{tag} is a user-defined function. Small cells do not have a DG solution directly associated with them but inherit the DG solution of the primary cells with which they are merged; therefore, if a primary cell is tagged for refinement so are all small cells that are merged with it. Empty cells are never tagged for refinement. In case of a two-phase solid, oftentimes (typically in proximity of the embedded boundary) an element $\mathscr{D}^e_\alpha$ of the phase α with DG solution $\mathbb{X}^e_\alpha$ and an element $\mathscr{D}^{e'}_\beta$ of the phase β with DG solution $\mathbb{X}^{e'}_\beta$ are associated with the same cell $\mathscr{C}$; in such a case, Eq. (16) will involve both $\mathbb{X}^e_\alpha$ and $\mathbb{X}^{e'}_\beta$. We note that Eq. (16) is used not only for refinement but also for coarsening. In fact, during the course of a numerical simulation, Eq. (16) is evaluated also for the cells covered by a finer grid and, if the DG solution of an element associated with a primary cell $\mathscr{C}$ ceases to fulfill Eq. (16), then all the finer cells covering the cell $\mathscr{C}$ and the merged small cells are removed.

When cells are refined or coarsened, the DG solution of the associated implicitly-defined elements must be suitably interpolated at the finer level or restricted at the coarser level, respectively. These operations are linear, local to the elements and can be implemented using matrix–vector products involving block-sparse operators. In particular, consider a vector $\mathbb{X}_\ell$ containing the DG solution of all the implicitly-defined elements at level ℓ. Then, as sketched in Fig. 5, the interpolation operator $\mathbb{I}_{\ell+1,\ell}$ transfers the DG solution $\mathbb{X}_\ell$ of the coarse level ℓ to the DG solution $\mathbb{X}_{\ell+1}$ of the fine level $\ell+1$. Formally, the interpolation operation can be written as

$$\mathbb{X}_{\ell+1} = \mathbb{I}_{\ell+1,\ell}\mathbb{X}_\ell, \tag{17}$$

where $\mathbb{I}_{\ell+1,\ell}$ is a block-structured matrix computed via the Galerkin projection, so that, given a fine element $\mathscr{D}^{e'}_{\ell+1}$ of level $\ell+1$ and a coarse element $\mathscr{D}^e_\ell$ of level ℓ, the block $\mathbb{I}^{e',e}_{\ell+1,\ell}$ is given by

$$\mathbb{I}^{e',e}_{\ell+1,\ell} \equiv (\mathbb{M}^{e'}_{\ell+1})^{-1} \int_{\mathscr{D}^{e'}_{\ell+1}\cap\mathscr{D}^e_\ell} \mathbb{B}^{e'\intercal}_{\ell+1}\mathbb{B}^e_\ell \, \mathrm{d}V. \tag{18}$$

To illustrate how $\mathbb{I}_{\ell+1,\ell}$ acts on $\mathbb{X}_\ell$, consider Fig. 6, which shows a two-element coarse mesh $\mathcal{M}_\ell \equiv \{\mathscr{D}^1_\ell, \mathscr{D}^2_\ell\}$ generated from a 1×2 coarse grid and a five-element fine mesh $\mathcal{M}_{\ell+1} \equiv \{\mathscr{D}^1_{\ell+1}, \mathscr{D}^2_{\ell+1}, \ldots, \mathscr{D}^5_{\ell+1}\}$ generated from a 2×4 grid that is obtained by refining the coarse grid with a refinement ratio of 2. Using Eq. (18) for the AMR configuration of Fig. 6, Eq. (17) becomes

$$\begin{pmatrix} \mathbb{X}^1_{\ell+1} \\ \mathbb{X}^2_{\ell+1} \\ \mathbb{X}^3_{\ell+1} \\ \mathbb{X}^4_{\ell+1} \\ \mathbb{X}^5_{\ell+1} \end{pmatrix} = \begin{bmatrix} \mathbb{I}^{1,1}_{\ell+1,\ell} & \mathbf{0} \\ \mathbb{I}^{2,1}_{\ell+1,\ell} & \mathbf{0} \\ \mathbb{I}^{3,1}_{\ell+1,\ell} & \mathbf{0} \\ \mathbb{I}^{4,1}_{\ell+1,\ell} & \mathbb{I}^{4,2}_{\ell+1,\ell} \\ \mathbf{0} & \mathbb{I}^{5,2}_{\ell+1,\ell} \end{bmatrix} \begin{pmatrix} \mathbb{X}^1_\ell \\ \mathbb{X}^2_\ell \end{pmatrix}. \tag{19}$$

It is interesting to note that Eq. (19) reflects the configurations of the meshes of Fig. 6, including the case of the fine element $\mathscr{D}^4_{\ell+1}$ that partially covers the two distinct coarse elements $\mathscr{D}^1_\ell$ and $\mathscr{D}^2_\ell$.

The counterpart of the interpolation operation is the restriction operation, whereby the DG solution $\mathbb{X}_{\ell+1}$ of the fine level $\ell+1$ is transferred to the DG solution $\mathbb{X}_\ell$ of the coarse level ℓ. Similar to Eq. (17), this operation can formally be written as

$$\mathbb{X}_\ell = \mathbb{R}_{\ell,\ell+1}\mathbb{X}_{\ell+1}, \tag{20}$$

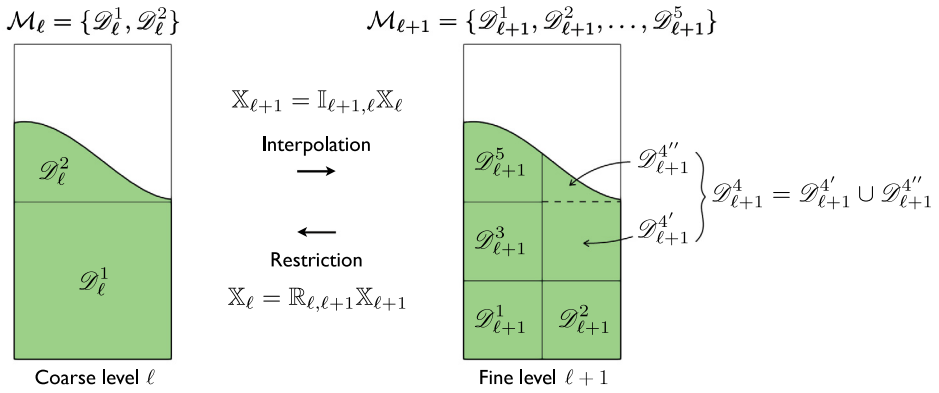

Fig. 6. Example of an *hp*-AMR configuration where the fine element $\mathscr{D}^4_{\ell+1}$ partially covers the two coarse elements $\mathscr{D}^1_\ell$ and $\mathscr{D}^2_\ell$.

where, using the Galerkin projection, the restriction operator $\mathbb{R}_{\ell,\ell+1}$ is related to the interpolation operator such that

$$\mathbb{R}_{\ell,\ell+1} = \mathbb{M}_\ell^{-1}\mathbb{I}^{\mathsf{T}}_{\ell+1,\ell}\mathbb{M}_{\ell+1}, \tag{21}$$

where $\mathbb{M}_\ell$ is the block-diagonal mass matrix of the implicitly-defined elements of the level ℓ.

It is worth noting that the interpolation and restriction operators defined via Eqs. (18) and (20), respectively, are valid regardless of the choice of basis functions and, therefore, naturally enable the use of different polynomial orders at different AMR levels. From an implementation viewpoint, the present DG scheme requires the evaluation of the interpolation operators and the Cholesky decomposition of the mass matrices, while the restriction operator can be applied on-the-fly using Eq. (21). Moreover, all standard (hyper) rectangular elements (which represent the majority of the mesh elements) share the same mass matrix, which can be precomputed and stored at the beginning of the simulations; the same applies to the interpolation operator between two standard elements of two different AMR levels. Conversely, the mass matrices and the interpolation operators of the cut elements are in general unique and are computed via Eqs. (15) and (18), respectively, using high-order quadrature rules.

3.5. Time-stepping

The last aspect of the numerical framework regards the time-stepping, i.e. the update in time of the coefficients of the spatial basis functions. Whether a single-level or an *hp*-AMR scheme is considered, the time-evolution equation for the coefficients $\mathbb{X}^e_\alpha$ of a generic eth element $\mathscr{D}^e_\alpha$ belonging to the phase α can be written as

$$\mathbb{M}^e_\alpha \dot{\mathbb{X}}^e_\alpha = \mathbb{A}^e_\alpha(t, \mathbb{X}), \tag{22}$$

where $\mathbb{A}^e_\alpha(t, \mathbb{X})$ stems from the evaluation of the right-hand side of Eq. (14) and $\mathbb{X}$ formally contains the coefficients of all the mesh elements; note however that only the DG solution from the neighboring elements of $\mathscr{D}^e_\alpha$ is required to compute $\mathbb{A}^e_\alpha(t, \mathbb{X})$. Integration in time of Eq. (22) is performed via an explicit high-order Runge–Kutta algorithm [23] matching the order of the highest spatial discretization among the mesh levels. As explicit time-integration schemes are conditionally stable, at a generic level ℓ with mesh size h_ℓ and using a DG scheme with polynomial degree p_ℓ, the maximum time step τ_ℓ is subject to the following CFL condition

$$\frac{\tau_\ell}{h_\ell} < \frac{C_\ell \overline{f}_\ell}{c(1+2p_\ell)}, \tag{23}$$

where $\overline{f}_\ell$ is the volume fraction threshold triggering the cell-merging at the level ℓ and C_ℓ is a constant smaller than 1 that does not depend on h_ℓ or p_ℓ; in all simulations presented in Section 4, $\overline{f}_0 = \overline{f}_1 = 0.3$ and $C_0 = C_1 = 0.833$. Moreover, in Eq. (23), $c \equiv c_\alpha$ for single-phase solids or $c \equiv \max\{c_\alpha, c_\beta\}$ for bi-phase solids, where c_α and c_β are the maximal speeds of the elastic waves in the phases α and β, respectively. For isotropic solids, c_α coincides with

the speed of the P-waves in the phase α. For general anisotropic solids, the wave speed depends on the direction of propagation and, therefore, the maximum wave speed is evaluated as [50]

$$c_\alpha^2 \equiv \max_{\boldsymbol{n}}\{\text{eig}(\boldsymbol{\Gamma}_\alpha(\boldsymbol{n}))\}, \tag{24}$$

where $\boldsymbol{\Gamma}_\alpha(\boldsymbol{n}) \equiv \rho_\alpha^{-1}\boldsymbol{I}_n^\intercal \boldsymbol{c}_\alpha \boldsymbol{I}_n$, $\boldsymbol{I}_n \equiv n_i \boldsymbol{I}_i$ and eig($\bullet$) returns the eigenvalues of $\bullet$.

Note that Eq. (23) assumes $\tau_\ell \sim h_\ell/(1+2p_\ell)$; this is typical of explicit Runge–Kutta discontinuous Galerkin schemes for first-order hyperbolic PDEs, where the order of the temporal discretization matches the order of the spatial discretization, see e.g. [57]. Finally, the time step τ of the Runge–Kutta algorithm is $\tau \equiv \min_\ell \tau_\ell$.

4. Results

In this section, the capabilities of the presented implicit-mesh DG framework are assessed for two- and three-dimensional test cases involving wave propagation in single- and bi-phase elastic solids.

The numerical simulations use implicitly-defined meshes generated either from uniform grids with mesh size h or from a two-level hp-AMR, where level 0 and level 1 have mesh size h_0 and h_1, respectively. For the simulations using the hp-AMR, the coarse level 0 is generated at the beginning of the simulation and is kept fixed, while the fine level 1 is dynamically updated during the time evolution by refining the coarse cells with a refinement ratio r such that $h_1 = h_0/r$ and the number of fine cells replacing a coarse cell is r^d. Tensor-product Legendre polynomials of degree p are employed to define the space $\mathcal{P}^e_{hp}$, and thus the space $\mathcal{V}_{hp}$ introduced in Eq. (10); the corresponding DG scheme is denoted by DG$_p$. We recall that the present hp-AMR strategy allows the use of different polynomial orders for different AMR levels.

4.1. Convergence analysis

Reported here are the results of several convergence tests on single- and bi-phase solids with isotropic, orthotropic and anisotropic constitutive behavior, in two and three-dimensions. For both the single-phase solid simulations and the bi-phase solid simulations, we assume that the phases have density $\rho_\alpha = \rho_\beta = \rho$ and stiffness $\boldsymbol{c}_\alpha = \boldsymbol{c}_\beta = \boldsymbol{c}$. In this section, all quantities are assumed non-dimensional. The considered isotropic solid has density $\rho = 1$, Young's modulus $Y = 1$ and Poisson's ratio $\nu = 0.3$. The considered orthotropic solid is a FCC Copper crystal [58] with density $\rho = 8.92$ and non-zero elastic constants $c_{11} = 168$, $c_{12} = 121$, $c_{44} = 75$. In 2D, the constitutive behavior of the considered anisotropic solid represents the in-plane behavior of a multilayered composite material [59] with density $\rho = 1.6$ and stiffness matrix

$$\boldsymbol{c} = \begin{bmatrix} 0.5637 & 0.2963 & 0.3158 \\ & 0.5637 & 0.3158 \\ \text{Sym.} & & 0.3111 \end{bmatrix}. \tag{25}$$

In 3D, the considered anisotropic solid is an Olivine crystal [38,60] with density $\rho = 1.0$ and whose orthorhombic axes are tilted and aligned with the directions $[1,1,1]$, $[-1,1,0]$ and $[-1,-1,2]$ such that the stiffness matrix in the global reference system is

$$\boldsymbol{c} = \begin{bmatrix} 185.8 & 67.3 & 76.2 & 2.69 & 17.6 & -5.44 \\ & 170.0 & 62.67 & 4.62 & -6.60 & -6.53 \\ & & 219.8 & 3.08 & 30.48 & -2.72 \\ & & & 59.0 & -1.905 & 2.83 \\ & \text{Sym.} & & & 79.2 & 2.12 \\ & & & & & 57.0 \end{bmatrix}. \tag{26}$$

We start by constructing an exact solution of Eq. (1) with zero source term. Consider a plane-wave vector field of the form

$$\boldsymbol{U}(t,\boldsymbol{x}) = \widetilde{\boldsymbol{U}}\sin(\omega t - \kappa_i x_i), \tag{27}$$

where κ_i is the ith component of the wave vector $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_d)$, ω is the angular frequency and $\widetilde{\boldsymbol{U}}$ is a constant vector. For a given choice of $\boldsymbol{\kappa}$, by plugging Eq. (27) into Eq. (1) with $\boldsymbol{S}_\alpha = \boldsymbol{0}$, ω and $\widetilde{\boldsymbol{U}}$ are obtained as the eigenvalue and the eigenvector, respectively, of the eigenvalue problem

$$\boldsymbol{A}_\kappa \widetilde{\boldsymbol{U}} = \omega \widetilde{\boldsymbol{U}}, \tag{28}$$

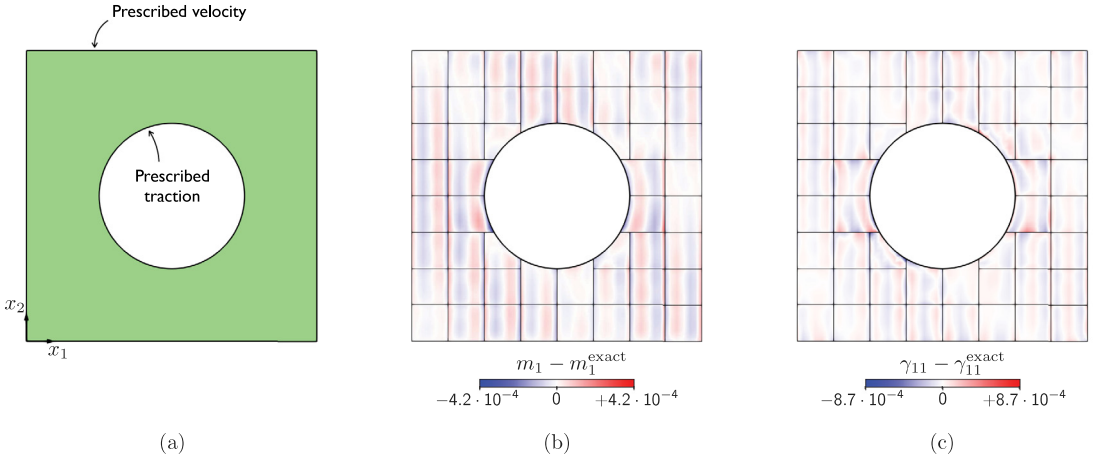

Fig. 7. (a) Geometry and boundary conditions for the single-phase solid employed in the 2D hp-convergence analysis. Error in (b) the momentum component m_1 and (c) the strain component γ_{11} obtained with an implicitly-defined mesh generated from an 8^2 background grid and a DG_3 scheme for the anisotropic material response.

where $\boldsymbol{A}_\kappa$ is $N_U \times N_U$ matrix given by

$$\boldsymbol{A}_\kappa \equiv \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_\kappa^{\intercal}\boldsymbol{c} \\ \rho^{-1}\boldsymbol{I}_\kappa & \boldsymbol{0} \end{bmatrix} \quad \text{and} \quad \boldsymbol{I}_\kappa \equiv \kappa_i \boldsymbol{I}_i. \tag{29}$$

There are in general N_U couples $\{\omega_v, \widetilde{\boldsymbol{U}}_v\}$, $v = 1, \ldots, N_U$, that are solution of Eq. (28); see Ref. [40] for more details about the eigenvalue properties of the matrix $\boldsymbol{A}_\kappa$. Then, an exact solution Eq. (1) can be written as the following linear superposition of plane waves

$$\boldsymbol{U}^{\text{exact}}(t, \boldsymbol{x}) = \sum_{v=1}^{N_U} \widetilde{\boldsymbol{U}}_v \sin(\omega_v t - \kappa_i x_i), \tag{30}$$

where the vectors $\widetilde{\boldsymbol{U}}_v$, $v = 1, \ldots, N_U$, are normalized to have unit amplitude. In all simulations, the exact solution $\boldsymbol{U}^{\text{exact}}$ given in Eq. (30) is employed to set the initial conditions as $\boldsymbol{U}_{\alpha 0}(\boldsymbol{x}) = \boldsymbol{U}_{\beta 0}(\boldsymbol{x}) = \boldsymbol{U}^{\text{exact}}(0, \boldsymbol{x})$, while the maximum eigenvalue $\omega_{\max} \equiv \max_v \omega_v$ determines the final time T of evolution as $T = 2\pi/\omega_{\max}$.

As the last ingredient of this convergence analysis, we introduce two error measures between the solution $\boldsymbol{U}$ computed via the present DG scheme and the exact solution $\boldsymbol{U}^{\text{exact}}$ given in Eq. (30). The error measures are

$$e_{L_\infty}(\boldsymbol{U}, \boldsymbol{U}^{\text{exact}}) \equiv \frac{\|\boldsymbol{U} - \boldsymbol{U}^{\text{exact}}\|_{L_\infty}}{\|\boldsymbol{U}^{\text{exact}}\|_{L_\infty}} \quad \text{and} \quad e_{L_2}(\boldsymbol{U}, \boldsymbol{U}^{\text{exact}}) \equiv \left[\frac{E(\boldsymbol{U} - \boldsymbol{U}^{\text{exact}})}{E(\boldsymbol{U}^{\text{exact}})}\right]^{1/2}, \tag{31}$$

where the L_∞ norm is evaluated by computing the maximum error at the quadrature points among all the components of $\boldsymbol{U}$ and the e_{L_2} error is evaluated by introducing an energy norm. In particular, let $E_\alpha(\boldsymbol{U})$ denote the energy associated with the solution $\boldsymbol{U}$ for the phase α given by

$$E_\alpha(\boldsymbol{U}) = \frac{1}{2}\int_{\mathscr{D}_\alpha} \left(\rho_\alpha^{-1}\boldsymbol{m}_\alpha^{\intercal}\boldsymbol{m}_\alpha + \boldsymbol{\gamma}_\alpha^{\intercal}\boldsymbol{c}_\alpha\boldsymbol{\gamma}_\alpha\right) \mathrm{d}V; \tag{32}$$

then, in Eq. (31), $E(\boldsymbol{U}) = E_\alpha(\boldsymbol{U})$ and $E(\boldsymbol{U}) = E_\alpha(\boldsymbol{U}) + E_\beta(\boldsymbol{U})$ for single-phase and bi-phase solids, respectively.

Fig. 7a shows the geometry and the boundary conditions for the 2D single-phase solid case. The geometry consists of a square with a circular cavity and is defined in the background unit square $[0, 1]^2$ by the level set function

$$\varphi(\boldsymbol{x}) = R^2 - (x_1 - o_1)^2 - (x_2 - o_2)^2, \tag{33}$$

where $R = 0.25$ and $o_1 = o_2 = 0.5$. A velocity field $\overline{\boldsymbol{v}}$ is prescribed on the outer boundary $\mathscr{B}_\alpha$ of the background square, whereas a traction field $\overline{\boldsymbol{t}}$ is prescribed on the zero contour $\mathscr{L}_\alpha$ of the level set function. The exact solution $\boldsymbol{U}^{\text{exact}}$ is specified by $\kappa_1 = 2\pi\cos(\pi/6)$ and $\kappa_2 = 2\pi\sin(\pi/6)$ and is employed to evaluate the fields $\overline{\boldsymbol{v}}$ and $\overline{\boldsymbol{t}}$ at

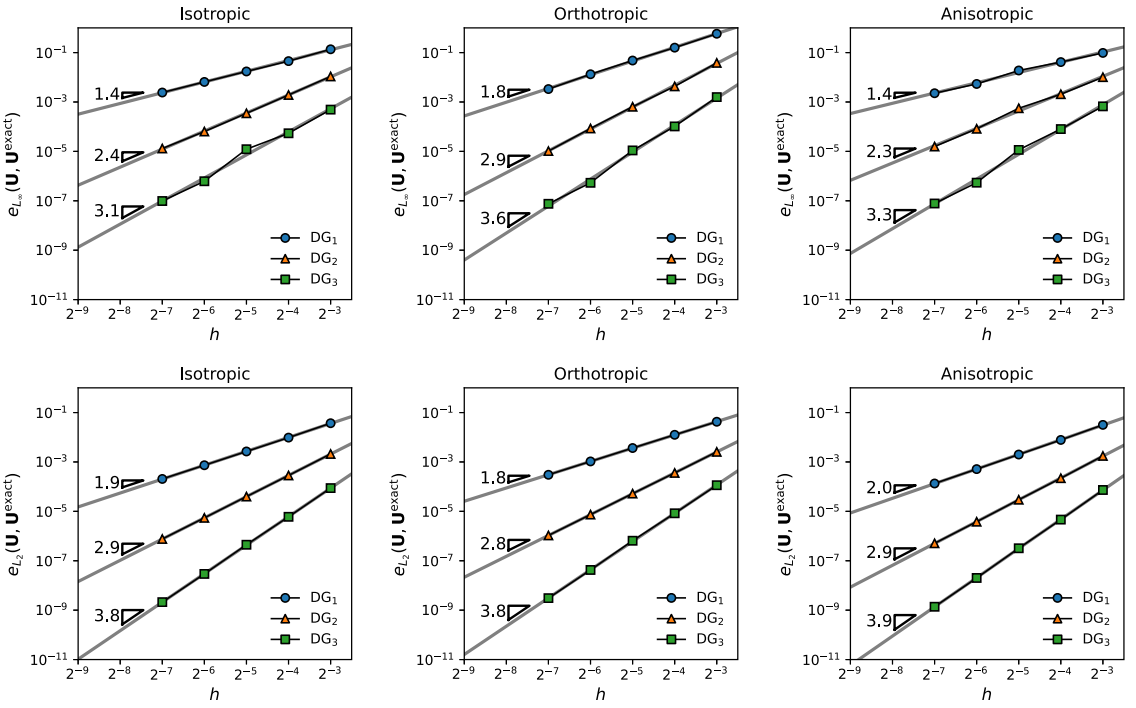

Fig. 8. (top row) e_{L_∞} error and (bottom row) e_{L_2} error for the 2D single-phase solid of Fig. 7 for different constitutive behaviors.

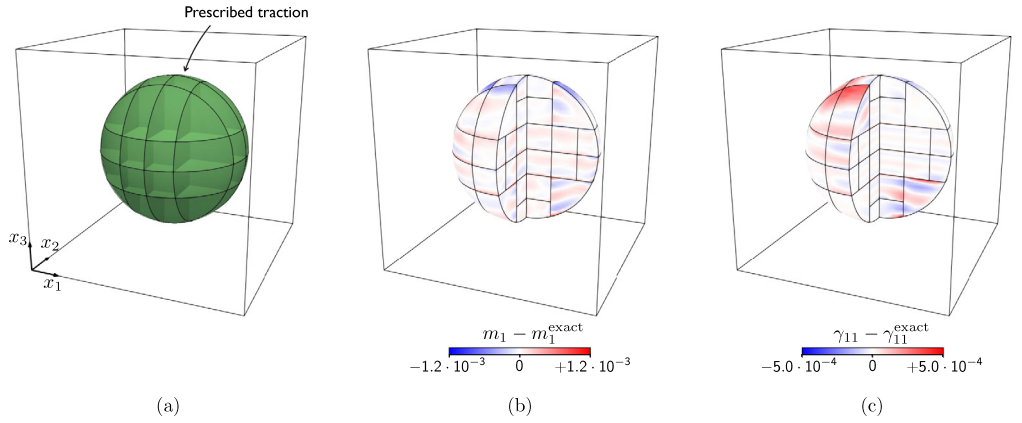

Fig. 9. (a) Geometry, boundary conditions and implicitly-defined mesh generated from an 8^3 background grid for the single-phase solid employed in the 3D *hp*-convergence analysis. Error in (b) the momentum component m_1 and (c) the strain component γ_{11} obtained with the implicitly-defined mesh of figure (a) and a DG_3 scheme for the anisotropic material response.

any $(t, \boldsymbol{x})$ on the geometry's boundaries. Figs. 7b and 7c show the error in the momentum component m_1 and the strain component γ_{11}, respectively, when an 8^2 background grid and a DG_3 scheme are employed for the anisotropic material response case. The figures also display the implicitly-defined mesh generated from the background grid. *hp*-convergence plots of the two error measures given in Eq. (31) for the 2D single-phase solid with isotropic, orthotropic and anisotropic behavior are shown in Fig. 8.

Fig. 9a shows the geometry and the boundary conditions for the 3D single-phase solid case. The geometry consists of a sphere implicitly-defined in the background unit cube $[0, 1]^3$ by the level set function

$$\varphi(\boldsymbol{x}) = (x_1 - o_1)^2 + (x_2 - o_2)^2 + (x_3 - o_3)^2 - R^2, \tag{34}$$

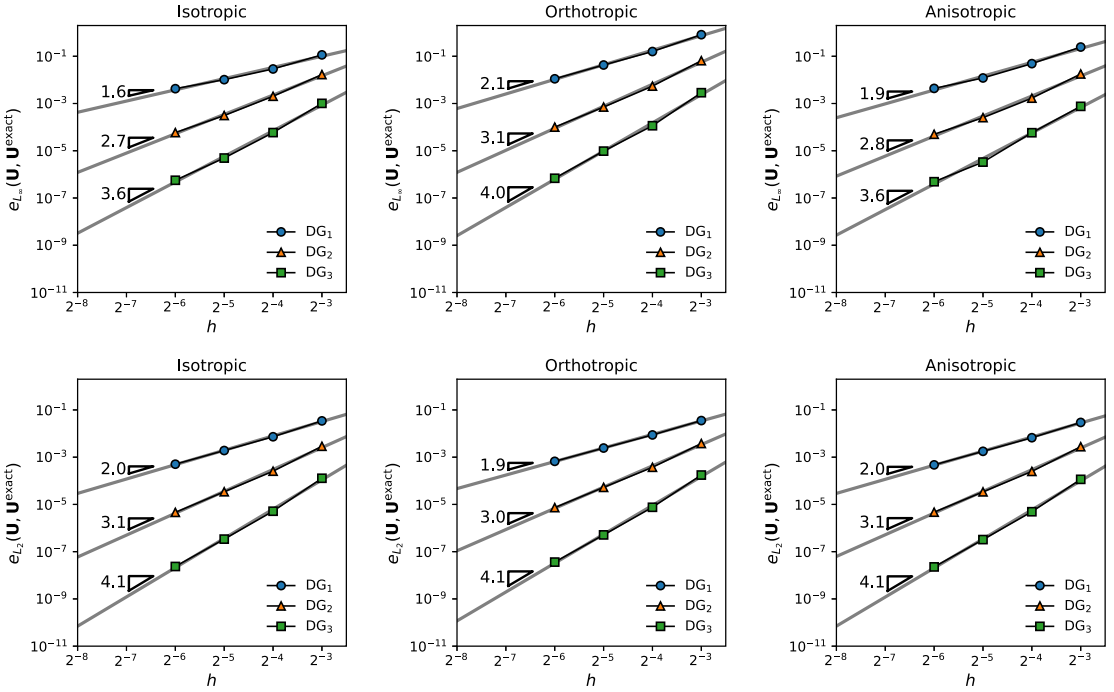

Fig. 10. (top row) e_{L_∞} error and (bottom row) e_{L_2} error for the 3D single-phase solid of Fig. 9 for different constitutive behaviors.

where $R = 0.35$ and $o_1 = o_2 = o_3 = 0.5$. A traction field $\bar{\boldsymbol{t}}$ is prescribed on sphere's outer boundary using the exact solution $\boldsymbol{U}^{\text{exact}}$ with $\kappa_1 = 2\pi\cos(\pi/3)\sin(\pi/6)$, $\kappa_2 = 2\pi\sin(\pi/3)\sin(\pi/6)$ and $\kappa_3 = 2\pi\cos(\pi/6)$. Fig. 9a also shows the implicitly-defined generated from an 8^3 background grid and some of the implicitly-defined elements in proximity of the embedded boundary. Figs. 9b and 9c show the error in the momentum component m_1 and the strain component γ_{11}, respectively, when the implicit-mesh of Fig. 9a and a DG_3 scheme are employed for the anisotropic material response case. The hp-convergence plots of the two error measures given in Eq. (31) for the 3D single-phase solid with isotropic, orthotropic and anisotropic behavior are then reported in Fig. 10.

Fig. 11a shows the geometry and the boundary conditions for the 2D two-phase solid case. The geometry is periodic and is defined in the background unit square $[0, 1]^2$ by the level set function

$$\varphi(\boldsymbol{x}) = \cos(2\pi x_1)\cos(2\pi x_2) - 1/8. \tag{35}$$

Periodic boundary conditions are prescribed on the outer boundaries $\mathscr{B}_\alpha$ and $\mathscr{B}_\beta$ of the background square, whereas perfect interface conditions as given in Eq. (8) are prescribed on $\mathscr{L}_{\alpha,\beta}$. The exact solution $\boldsymbol{U}^{\text{exact}}$ is specified by $\kappa_1 = 2\pi$ and $\kappa_2 = 0$. Figs. 11b and 11c show the error in the momentum component m_1 and the strain component γ_{11}, respectively, when an 8^2 background grid and a DG_3 scheme are employed for the anisotropic material response case. The obtained hp-convergence plots for the isotropic, orthotropic and anisotropic behavior are reported in Fig. 12.

Fig. 13a shows the geometry and the boundary conditions for the 3D two-phase solid case. The geometry is periodic and is defined in the background unit cube $[0, 1]^3$ by the level set function

$$\varphi(\boldsymbol{x}) = -\cos(2\pi x_1)\cos(2\pi x_2)\cos(2\pi x_3) - 1/8. \tag{36}$$

Periodic boundary conditions are prescribed on the outer boundaries $\mathscr{B}_\alpha$ and $\mathscr{B}_\beta$ of the background square, whereas perfect interface conditions are prescribed on $\mathscr{L}_{\alpha,\beta}$. The exact solution $\boldsymbol{U}^{\text{exact}}$ is specified by $\kappa_1 = 2\pi$ and $\kappa_2 = \kappa_3 = 0$. Figs. 13b and 13c display the implicitly-defined mesh of the phase α and the phase β, respectively, generated from an 8^3 background grid. For this mesh, Figs. 14a and 14b show the error in the momentum component m_1 and the strain component γ_{11}, respectively, when a DG_3 scheme is employed for the anisotropic material response case. Finally, the hp-convergence plots for the isotropic, orthotropic and anisotropic behavior are reported in Fig. 15.

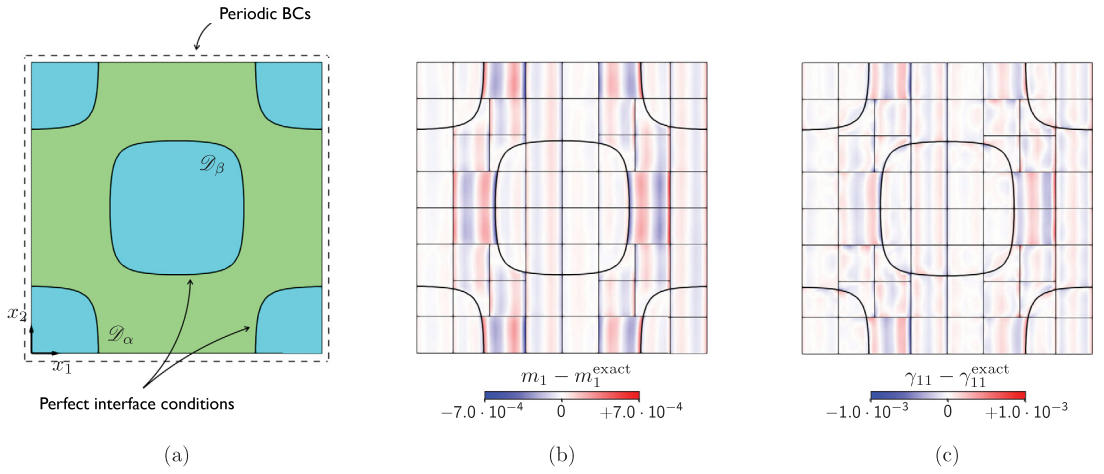

Fig. 11. (a) Geometry and boundary conditions for the bi-phase solid employed in the 2D hp-convergence analysis. Error in (b) the momentum component m_1 and (c) the strain component γ_{11} obtained with an implicitly-defined mesh generated from an 8×8 background grid and a DG_3 scheme for the anisotropic material response.

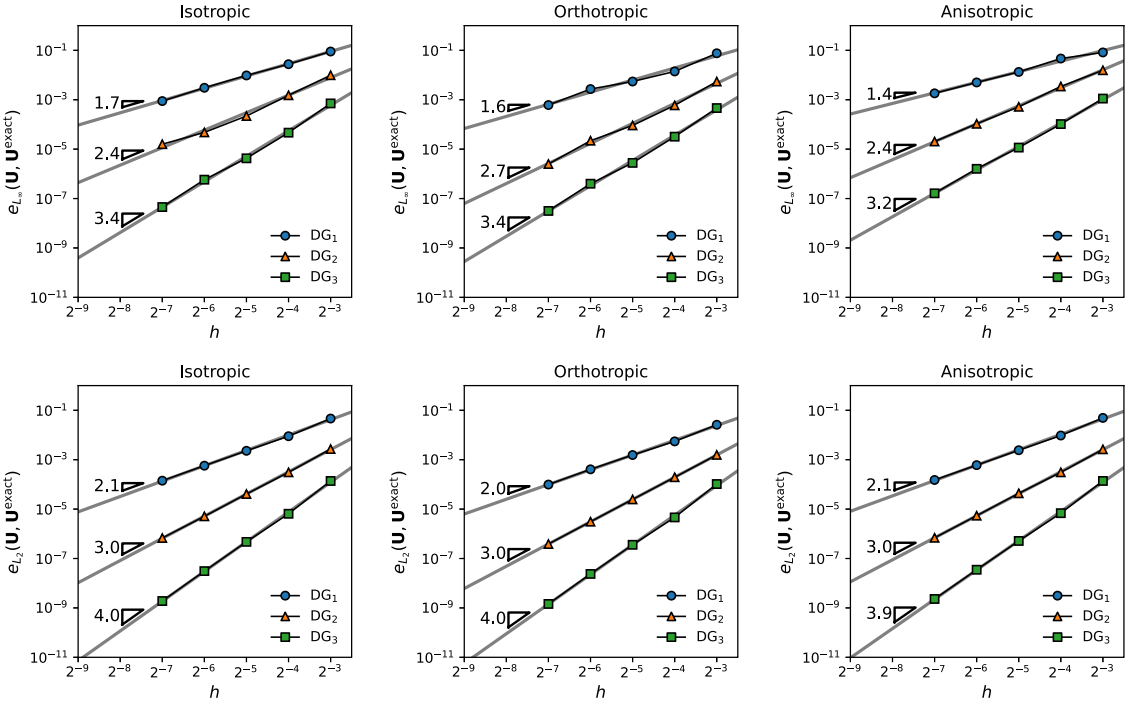

Fig. 12. (top row) e_{L_∞} error and (bottom row) e_{L_2} error for the 2D single-phase solid of Fig. 11 for different constitutive behaviors.

To conclude this part of the numerical results, we observe that in all simulations the error between the exact solution and the numerical solution in the extended elements is mildly larger than the error in the regular (hyper) rectangular elements, see for example Fig. 9c or Fig. 14c. This has also been observed for other applications of the present implicit-mesh DG approaches [30–32] and is an expected behavior if one considers that the extended elements are in general larger and geometrically less regular than the (hyper) rectangular elements. However, as shown by all the hp-convergence plots, and consistently with previous observations [30–32], the present implicit-mesh DG method provides a high-order accurate solution of the elastodynamics problem demonstrated by a $\mathcal{O}(h^{p+a})$

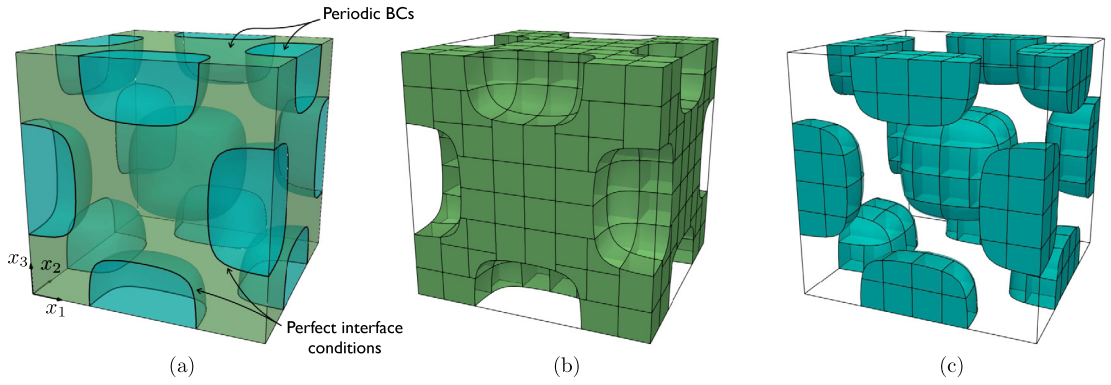

Fig. 13. (a) Geometry and boundary conditions for the bi-phase solid employed in the 3D *hp*-convergence analysis. Implicitly-defined mesh of (b) the phase α and (c) the phase β generated from an $8 \times 8 \times 8$ background grid for the solid of figure (a).

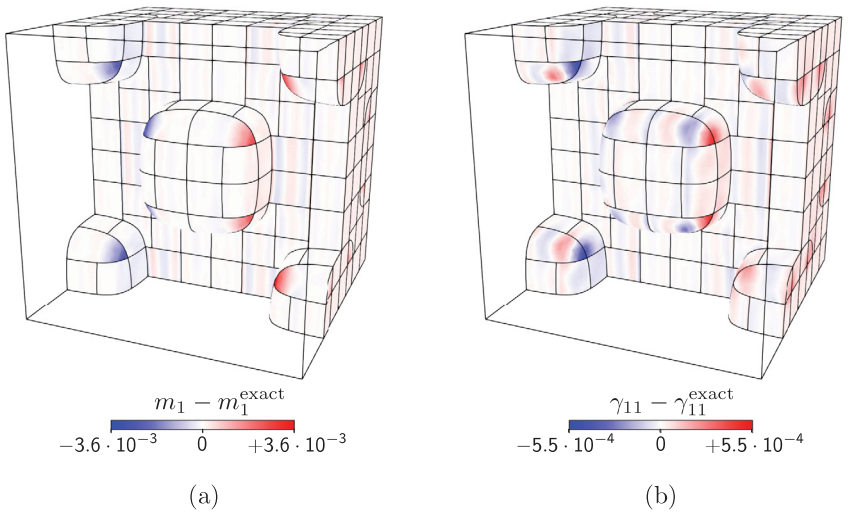

Fig. 14. Error in (a) the momentum component m_1 and (b) the strain component γ_{11} obtained with the implicitly-defined mesh of Fig. 13 and a DG_3 scheme for the anisotropic material response.

convergence rate in the energy norm, with a ranging from 0.8 to 1.2, and by a $\mathcal{O}(h^{p+a})$ convergence rate in the L_∞ norm, with a ranging from 0.1 to 1.1. We note that these results, showing high-order accuracy in both the maximum and energy norms, demonstrates the method yields low numerical dispersion and dissipation errors. The dispersion analysis of Ref. [61] offers another approach for examining these effects.

4.2. Lamb's problem

In this section, we consider a classical problem in elastodynamics, namely the Lamb's problem, which admits an exact solution [62,63] and has been used to assess various numerical models, see e.g. Refs. [42,64,65]. The problem consists of evaluating the distribution of the mechanical fields due to a concentrated force that is applied perpendicular to an infinite free surface, i.e. with zero-traction boundary conditions. The well-known distinctive feature of this problem is the appearance of the Rayleigh waves, which travel along the free surface of the domain but not in its depth.

Following the problem setup of Refs. [42,64,65], the 2D geometry is implicitly-defined in the background rectangle $\mathscr{R} = [0, 4000\,\text{m}] \times [0, 3000\,\text{m}]$ by the level set function

$$\varphi(\boldsymbol{x}) = x_2 - H - \tan(\vartheta)x_1, \tag{37}$$

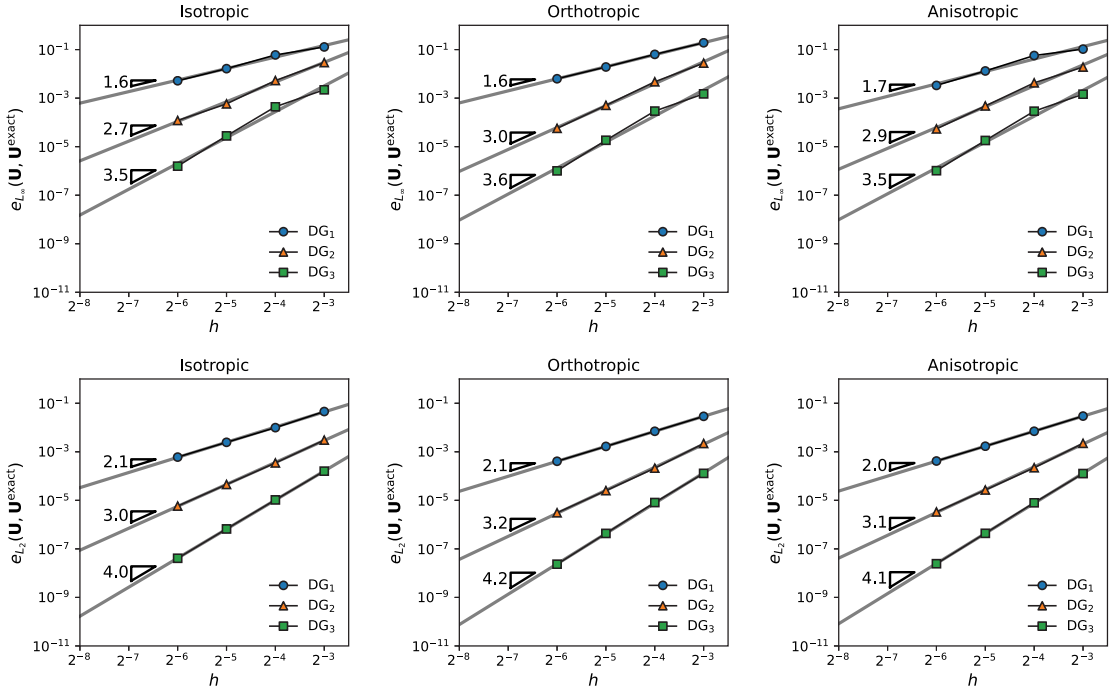

Fig. 15. (top row) e_{L_∞} error and (bottom row) e_{L_2} error for the 3D single-phase solid of Fig. 13 for different constitutive behaviors.

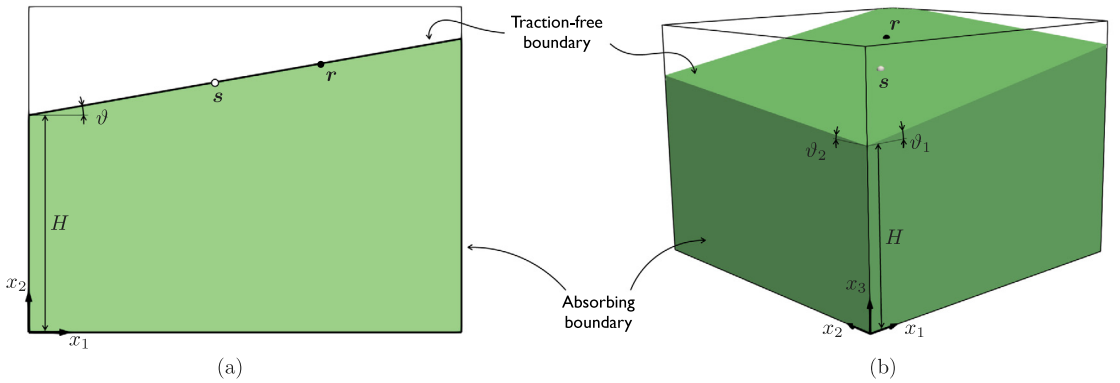

Fig. 16. Geometry and boundary conditions for (a) the 2D and (b) the 3D Lamb's problem. In the figures, $\boldsymbol{s}$ denotes the location of the source point where the concentrated force is applied whereas $\boldsymbol{r}$ denotes the location of the receiver where the elastodynamic response is measured.

where $H = 2000\,\mathrm{m}$ and $\vartheta = 10°$. We note that the problem may be stated in a reference system that is aligned with the free surface and the elastic domain may be straightforwardly meshed with a simple structured grid without involving implicitly-defined elements; nevertheless, this setup is a common benchmark problem, including to assess EB approaches, see, e.g., Ref. [42]. In 3D, we consider a simple extension of the 2D case, whereby the geometry is implicitly-defined in the background prism $\mathscr{R} = [0, 4000\,\mathrm{m}] \times [0, 4000\,\mathrm{m}] \times [0, 3000\,\mathrm{m}]$ by the level set function

$$\varphi(\boldsymbol{x}) = x_3 - H - \tan(\vartheta_1)x_1 - \tan(\vartheta_2)x_2, \tag{38}$$

where $H = 2000\,\mathrm{m}$, $\vartheta_1 = 10°$ and $\vartheta_1 = 5°$. The 2D geometry and the 3D geometry are displayed in 16a and Fig. 16b, respectively. In both figures, the point $\boldsymbol{s}$ denotes the location of the force and the point $\boldsymbol{r}$ denotes the location of a receiver where the resulting elastodynamic response is measured; in 2D, the source point and the

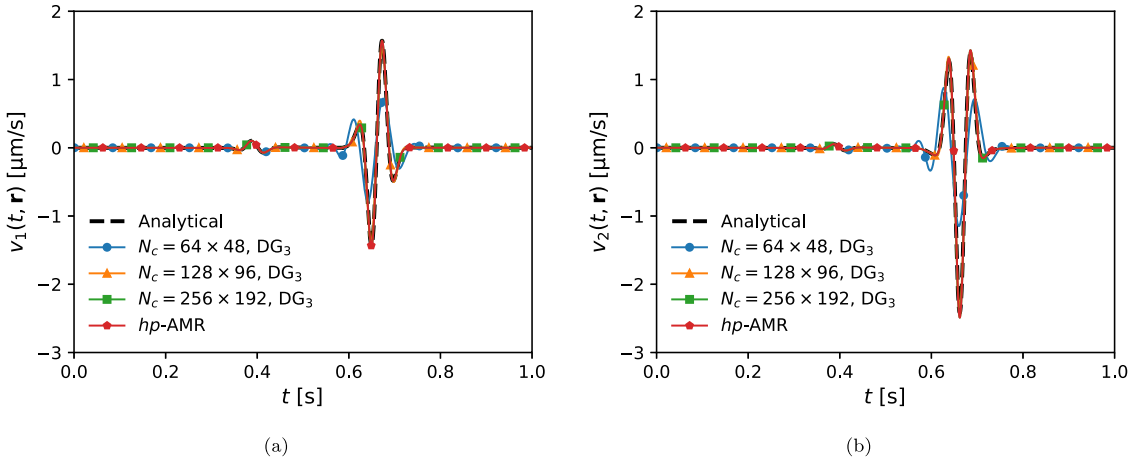

Fig. 17. Comparison between the analytical solution [62] and the DG solution for the 2D Lamb's problem in terms of (a) horizontal velocity component v_1 and (b) vertical velocity component v_2 evaluated at the surface receiver $\boldsymbol{r}$ denoted by the black dot in Fig. 16a.

receiver are located at $\boldsymbol{s} = (s_1, s_2)$ and $\boldsymbol{r} = (r_1, r_2)$, where $s_1 = 1720\,\text{m}$, $r_1 = 2694.96\,\text{m}$ and s_2 and r_2 are obtained via Eq. (37); in 3D, the source point and the receiver are located at $\boldsymbol{s} = (s_1, s_2, s_3)$ and $\boldsymbol{r} = (r_1, r_2, r_4)$, where $s_1 = s_2 = 1720\,\text{m}$, $r_1 = r_2 = 2694.96\,\text{m}$ and s_3 and r_3 are obtained via Eq. (38). The elastic domain is an isotropic solid with density $\rho_\alpha = 2200.0\,\text{kg/m}^3$ and elastic properties determined by the velocity of the P-waves $c_P = 3200.0\,\text{m/s}$ and the velocity of the S-waves $c_S = 1847.5\,\text{m/s}$. Traction-free boundary conditions, i.e. $\overline{\boldsymbol{t}} = \boldsymbol{0}$, are prescribed on $\mathscr{L}_\alpha$ whereas absorbing boundary conditions are prescribed on $\mathscr{B}_\alpha$. The final time of the simulation is $T = 1\,\text{s}$. Finally, the concentrated force is modeled by setting $\rho_\alpha \boldsymbol{b}_\alpha$ in the source term $\boldsymbol{S}_\alpha$, see Eq. (2), as

$$\rho_\alpha \boldsymbol{b}_\alpha = R_w(t)\delta(\boldsymbol{x} - \boldsymbol{s})\hat{\boldsymbol{u}}, \tag{39}$$

where $\hat{\boldsymbol{u}}$ is the unit vector perpendicular to $\mathscr{L}_\alpha$, $\delta(\boldsymbol{x})$ is the Dirac delta function, and $R_w(t)$ is the Ricker wavelet defined as

$$R_w(t) \equiv a_1 \left(\frac{1}{2} + a_2 (t - t_0)^2 \right) e^{a_2 (t - t_0)^2} \tag{40}$$

being $a_1 = -2000\,\text{kg/(m}^2\text{s}^2)$, $a_2 = -\pi^2 f_c^2$, $f_c = 14.5\,\text{Hz}$, $t_0 = 0.08\,\text{s}$.

In 2D, we consider three implicitly-defined meshes generated from uniform grids and an implicitly-defined mesh associated with a two-level hp-AMR scheme. The three uniform meshes are generated from a 64×48 grid, a 128×96 grid and a 256×192 grid, and use a DG_3 scheme. For the AMR test problem, level $\ell = 0$ uses a 64×48 grid and a DG_1 scheme, whereas level $\ell = 1$ uses a DG_3 scheme and is dynamically generated from level $\ell = 0$ via a refinement ratio $r = 4$; it follows that the hp-AMR has the same effective resolution of the finest uniform mesh. To evolve the cell tagging, the function f_{tag} introduced in Eq. (16) implements the following energy-based threshold as

$$f_{\text{tag}} \equiv E_\alpha^e - E_0, \tag{41}$$

where E_α^e denotes the energy associated with the solution $\boldsymbol{U}_\alpha^e$ for the element $\mathscr{D}_\alpha^e$, i.e. it is computed via Eq. (32) where $\mathscr{D}_\alpha$ is replaced by $\mathscr{D}_\alpha^e$, and E_0 is a threshold value chosen to be $E_0 = 10^{-12}\,\text{J}$. We remark that much more sophisticated refinement/coarsening criteria for DG methods exist in the literature, see for example Ref. [66] where the authors use a criterion involving the DG solution and its derivatives in space; in this work we use a simpler criterion tuned such that the propagating waves are resolved by the DG_3 scheme while the remaining parts of the domain are resolved by the DG_1 scheme.

For the considered meshes, the values of the velocity components v_1 and v_2 measured at the receiver $\boldsymbol{r}$ of Fig. 16a are reported as functions of time in Figs. 17a and 17b, respectively. The plots show the expected convergence of the DG solution with respect to the number of mesh elements, and thus the mesh size, and the comparison between the numerical solution and the analytical solution [62], which is well recovered by the present scheme.

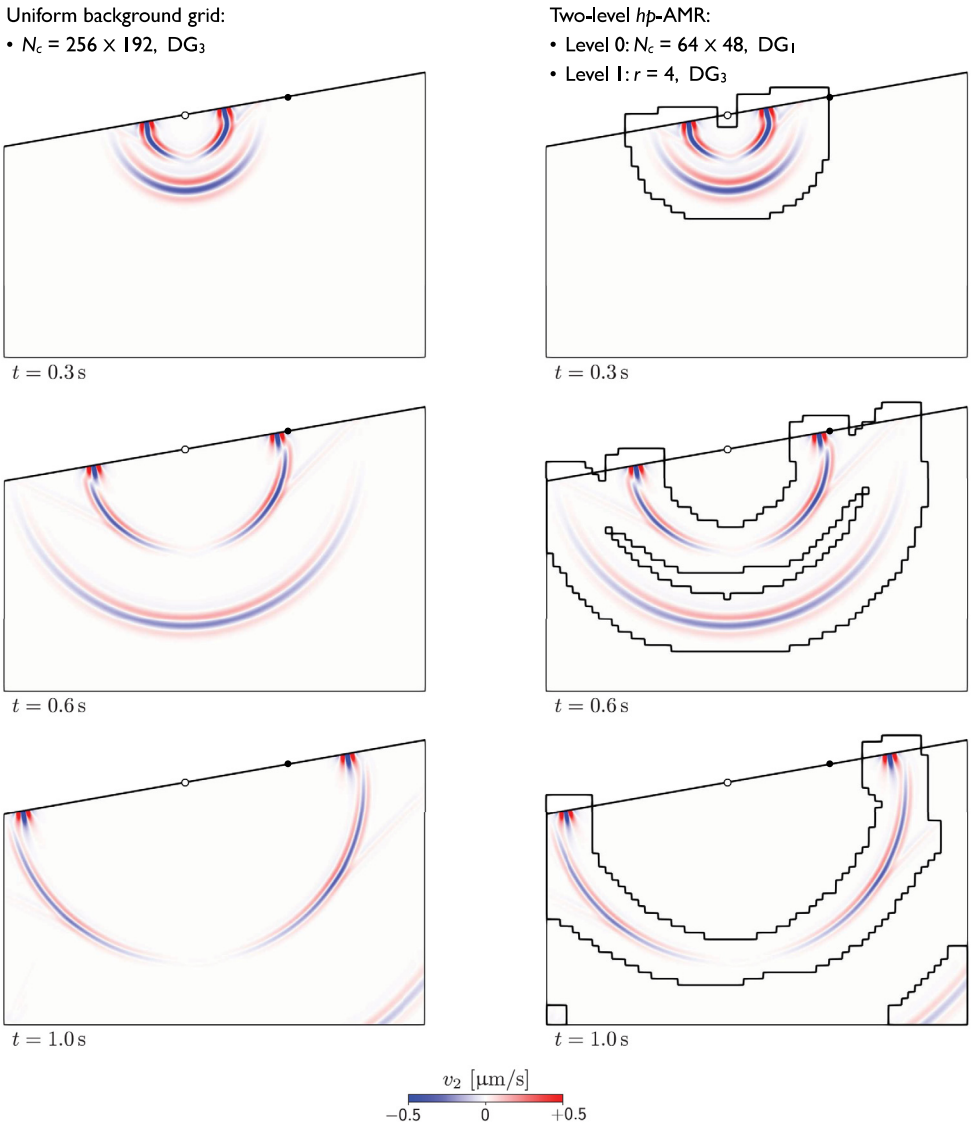

Fig. 18. Snapshots of the vertical velocity component v_2 at the time instants $t = 0.3\,\mathrm{s}$, $0.6\,\mathrm{s}$ and $1.0\,\mathrm{s}$ for the 2D Lamb's problem. (left column) Uniform background grid. (right column) Two-level hp-AMR. In the right column, the stepped lines denote the boundary of the AMR level $\ell = 1$.

A clearer view of the wave structure generated by the concentrated force is displayed in Fig. 18, where the distribution of the velocity component v_2 is displayed at the time instants $t = 0.3\,\mathrm{s}$, $0.6\,\mathrm{s}$ and $1.0\,\mathrm{s}$. The left column of the figure shows the results computed with the finest uniform grid, whereas the right column shows the results computed with the hp-AMR strategy; the same results are obtained with the two numerical setups. Moreover, in either case, it is possible to distinguish the larger semicircle of the P-waves, which at $t = 1.0\,\mathrm{s}$ have almost left the domain of analysis, the smaller semicircle of the S-waves, which are traveling slower than the P-waves but have a similar spatial distribution, and the Rayleigh waves, which are traveling attached to the free surface at a speed that is slightly slower than that of the S-waves.

In 3D, guided by the 2D results, we consider only an implicitly-defined mesh associated with a two-level hp-AMR scheme. The level $\ell = 0$ uses a $64 \times 64 \times 48$ grid and a DG_1 scheme, whereas level $\ell = 1$ uses a DG_3 scheme and is dynamically generated from level $\ell = 0$ via a refinement ratio $r = 4$. Cell tagging is performed using Eq. (41) where $E_0 = 10^{-18}\,\mathrm{J}$.

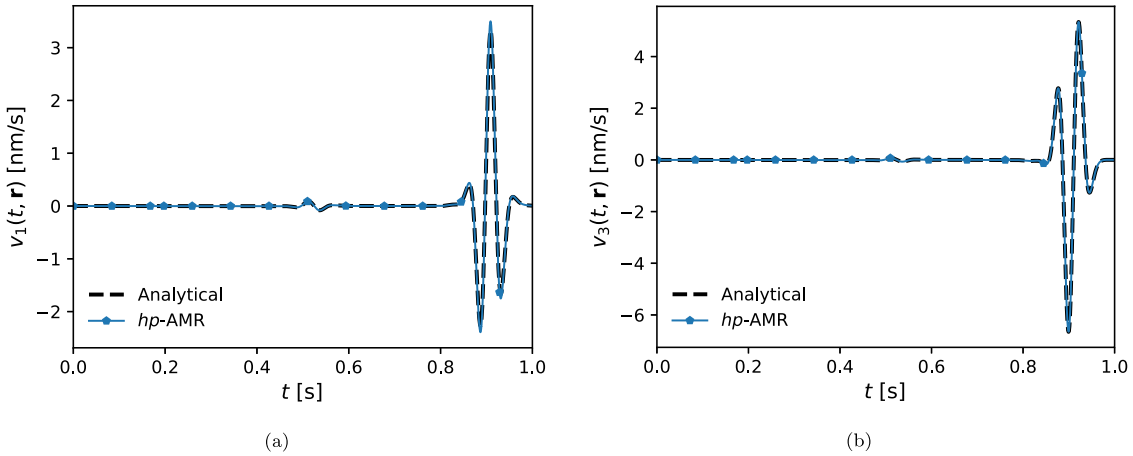

Fig. 19. Comparison between the analytical solution [63] and the DG solution for the 3D Lamb's problem in terms of (a) horizontal velocity component v_1 and (b) vertical velocity component v_3 evaluated at the surface receiver $\boldsymbol{r}$ denoted by the black dot in Fig. 16b.

The values of the velocity components v_1 and v_3 measured at the receiver $\boldsymbol{r}$ of Fig. 16b are reported as functions of time in Figs. 19a and 19b, respectively. As in prior tests, the DG solution matches well with the exact solution. Meanwhile, the left column of Fig. 20 shows the arrangement of the AMR level $\ell = 1$ at the time instants $t = 0.3$ s, 0.6 s and 1.0 s, while the right column of Fig. 20 shows the distribution of the velocity component v_3 and the location of the P-, S- and Rayleigh waves at the same time instants.

4.3. Single interface problem

Another classical problem in elastodynamics that has been modeled using different numerical methods, see e.g. Refs. [8,38,67], regards the evaluation of the wave structure caused by a concentrated force acting in proximity of the interface between an isotropic solid and an orthotropic solid. In the literature, the interface is typically aligned with the global reference system, i.e., the interface is perfectly horizontal or vertical; here, we instead place the interface on an angle in order to yield non-trivial implicitly-defined mesh geometry, similar to the case of the Lamb's problem discussed in Section 4.2. The geometry is depicted in Fig. 21 and is defined in the background square $\mathscr{R} = [-0.33\,\text{m}, 0.33\,\text{m}]^2$ by the level set function

$$\varphi(\boldsymbol{x}) = \cos(\vartheta)x_1 + \sin(\vartheta)x_2 \tag{42}$$

where $\vartheta = 10°$. Fig. 21 also shows the location of the source point $\boldsymbol{s} = (s_1, s_2)$ where the concentrated force is applied, the location of four receiver points $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, $\boldsymbol{r}_3$ and $\boldsymbol{r}_4$, where the mechanical signals are evaluated, and a local reference system that is aligned with the interface between the two phases. In this local reference system, the coordinates of the source and receiver points are $s_\eta = -0.02\,\text{m}$, $s_\xi = 0$, $r_{1\eta} = -0.105\,\text{m}$, $r_{2\eta} = -0.035\,\text{m}$, $r_{3\eta} = -0.01\,\text{m}$, $r_{4\eta} = 0.105\,\text{m}$, and $r_{1\xi} = r_{2\xi} = r_{3\xi} = r_{4\xi} = -0.08\,\text{m}$. The phase α and the phase β are an orthorhombic solid and an isotropic solid, respectively, whose properties are $\rho_\alpha = \rho_\beta = 7100\,\text{kg/m}^3$ and

$$\boldsymbol{c}_\alpha = \begin{bmatrix} 165.0 & 50.0 & 0 \\ & 62.0 & 0 \\ \text{Sym.} & & 39.6 \end{bmatrix} \text{GPa} \quad \text{and} \quad \boldsymbol{c}_\beta = \begin{bmatrix} 165.0 & 85.8 & 0 \\ & 165.0 & 0 \\ \text{Sym.} & & 39.6 \end{bmatrix} \text{GPa}, \tag{43}$$

where the elastic components are referred to the local reference system. Perfect-interface conditions are prescribed on $\mathscr{L}_{\alpha,\beta}$ whereas absorbing boundary conditions are prescribed on $\mathscr{B}_\alpha$ and $\mathscr{B}_\beta$. The final time of the simulation is $T = 100$ μs. Finally, the concentrated force is modeled by setting $\rho_\alpha \boldsymbol{b}_\alpha$ according to Eq. (39), where $\hat{\boldsymbol{u}}$ here is the unit vector parallel to $\mathscr{L}_{\alpha,\beta}$ and the parameters of the Ricker wavelet $R_w(t)$ are $a_1 = 10^{12}\,\text{kg/(m}^2\text{s}^2)$, $a_2 = -\pi^2 f_c^2$, $f_c = 170.0\,\text{kHz}$, $t_0 = 6\,\mu\text{s}$.

We consider an implicitly-defined mesh generated from a 256^2 uniform grid as well as one associated with a two-level hp-AMR scheme. In the AMR case, level $\ell = 0$ uses a 64^2 grid and a DG_1 scheme, whereas level $\ell = 1$

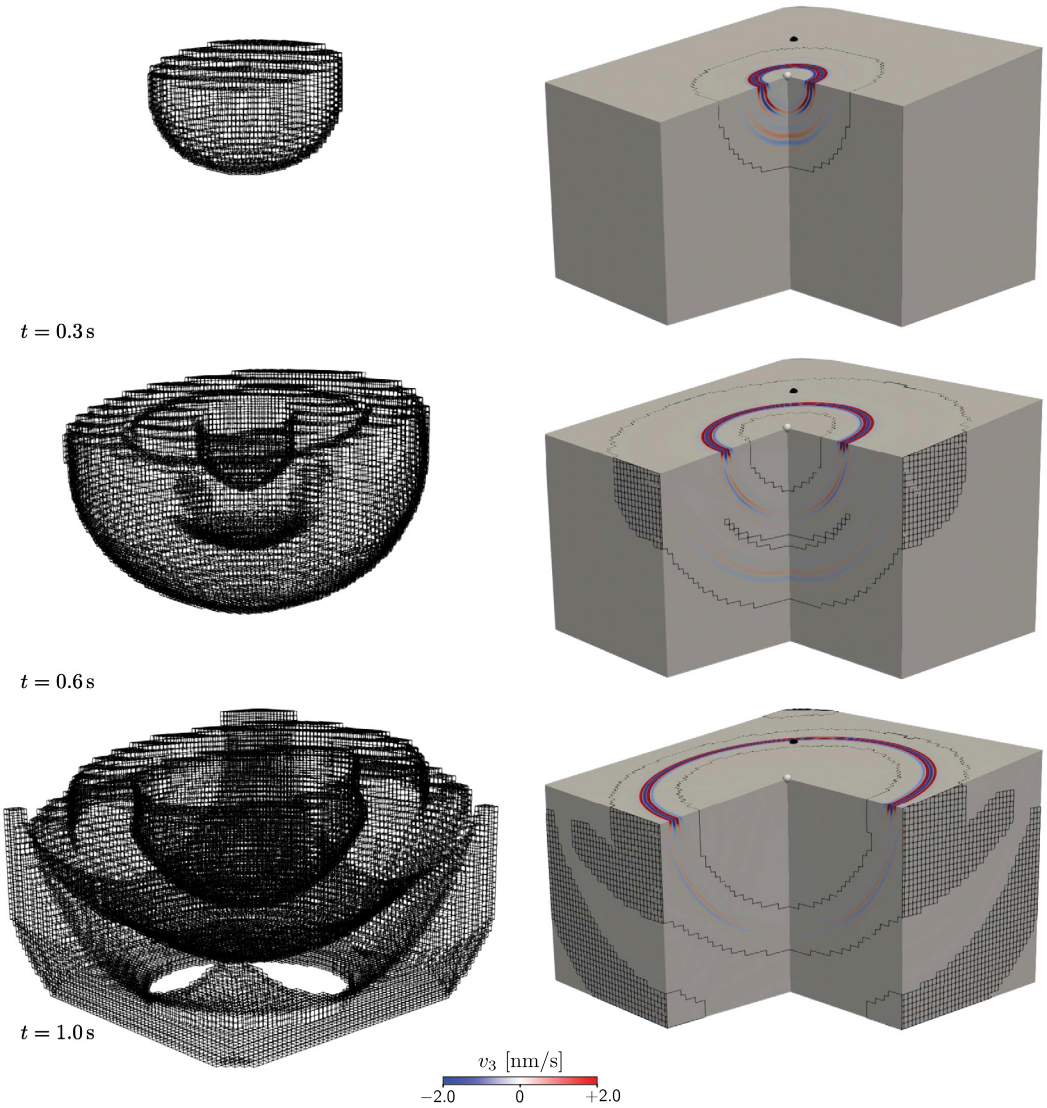

Fig. 20. Snapshots of (left column) the boundary of the AMR level $\ell = 1$ and (right column) the vertical velocity component v_3 at the time instants $t = 0.3\,\mathrm{s}$, $0.6\,\mathrm{s}$ and $1.0\,\mathrm{s}$ for the 3D Lamb's problem. In the right column, the stepped lines denote the trace of AMR level $\ell = 1$ on the domain boundaries.

uses a DG_3 scheme and is dynamically generated from level $\ell = 0$ via a refinement ratio $r = 4$. Similar to the Lamb's problem in 2D, the hp-AMR has the same effective resolution of the uniform mesh. To evolve the cell tagging, the function f_{tag} introduced in Eq. (16) implements the following energy-based threshold

$$f_{\mathrm{tag}} \equiv \max\{E^e_\alpha, E^e_\beta\} - E_0, \tag{44}$$

where $E_0 = 10^4\,\mathrm{J}$ and it is clear that $E^e_\alpha = 0$ if the element $\mathscr{D}^e_\alpha$ is empty.

Figs. 22a to 22d report the velocity component v_η at the receiver locations $\boldsymbol{r}_1$ to $\boldsymbol{r}_4$, respectively, and show that the results obtained with the uniform mesh and the results obtained with the hp-AMR are overlapping and match very well with the reference solution [8]. Finally, the wave structure generated by the concentrated force is displayed in Fig. 23 at the time instants $t = 30\,\mu\mathrm{s}$, $60\,\mu\mathrm{s}$ and $100\,\mu\mathrm{s}$ in terms of the velocity component v_η. From the figures,

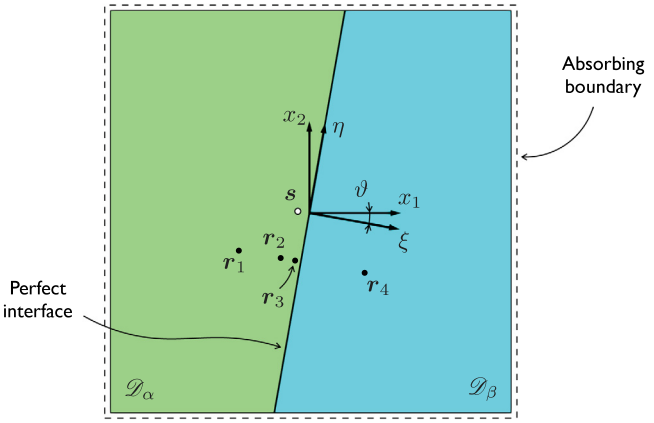

Fig. 21. Geometry and boundary conditions for the single interface problem.

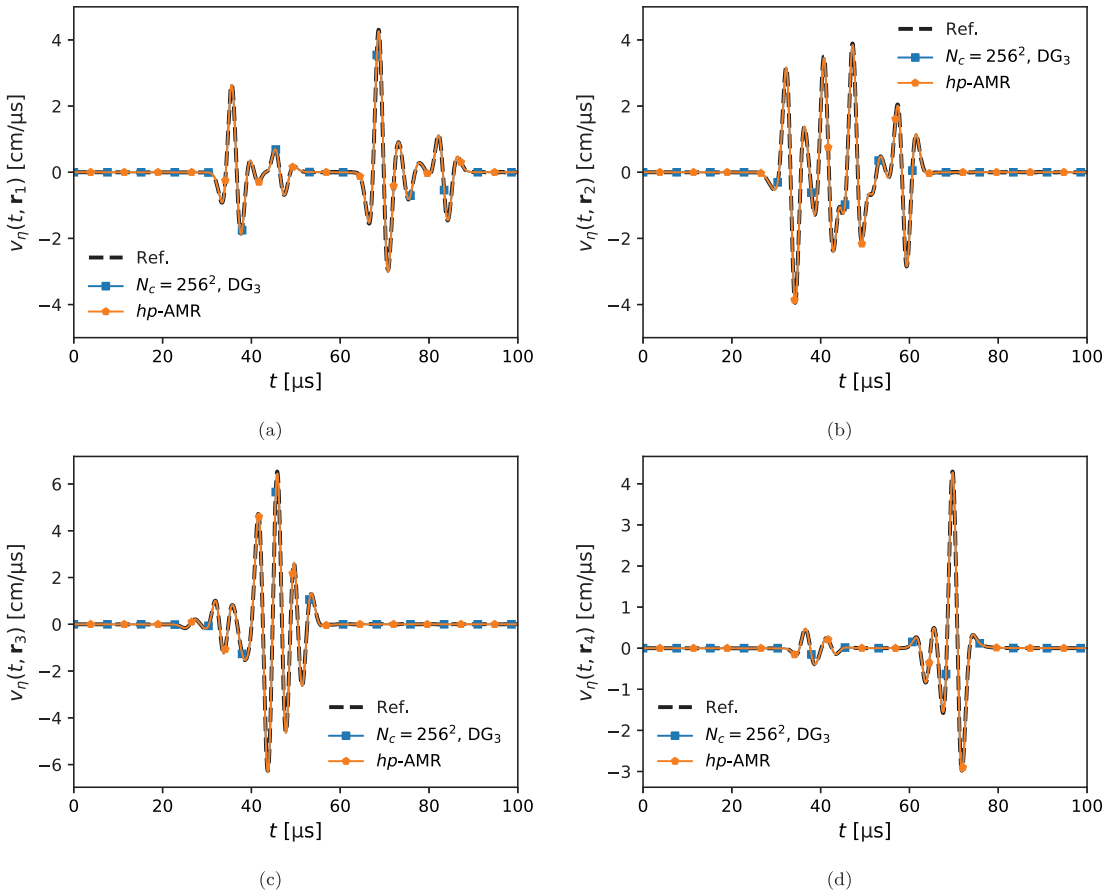

Fig. 22. Comparison between the reference solution [8] and the present formulation for the single interface problem in terms of the velocity component v_η evaluated at the receivers (a) $\boldsymbol{r}_1$, (b) $\boldsymbol{r}_2$, (c) $\boldsymbol{r}_3$ and (d) $\boldsymbol{r}_4$ denoted by the black dots in Fig. 21.

one can clearly observe the structure of the isotropic waves (characterized by semicircles) and the structure of the orthotropic waves, which propagate faster along the direction perpendicular to the interface and slower along the direction parallel to the interface. Fig. 23 also shows that the selected tagging criterion allows the hp-AMR scheme to reproduce the solution obtained with the uniform mesh.

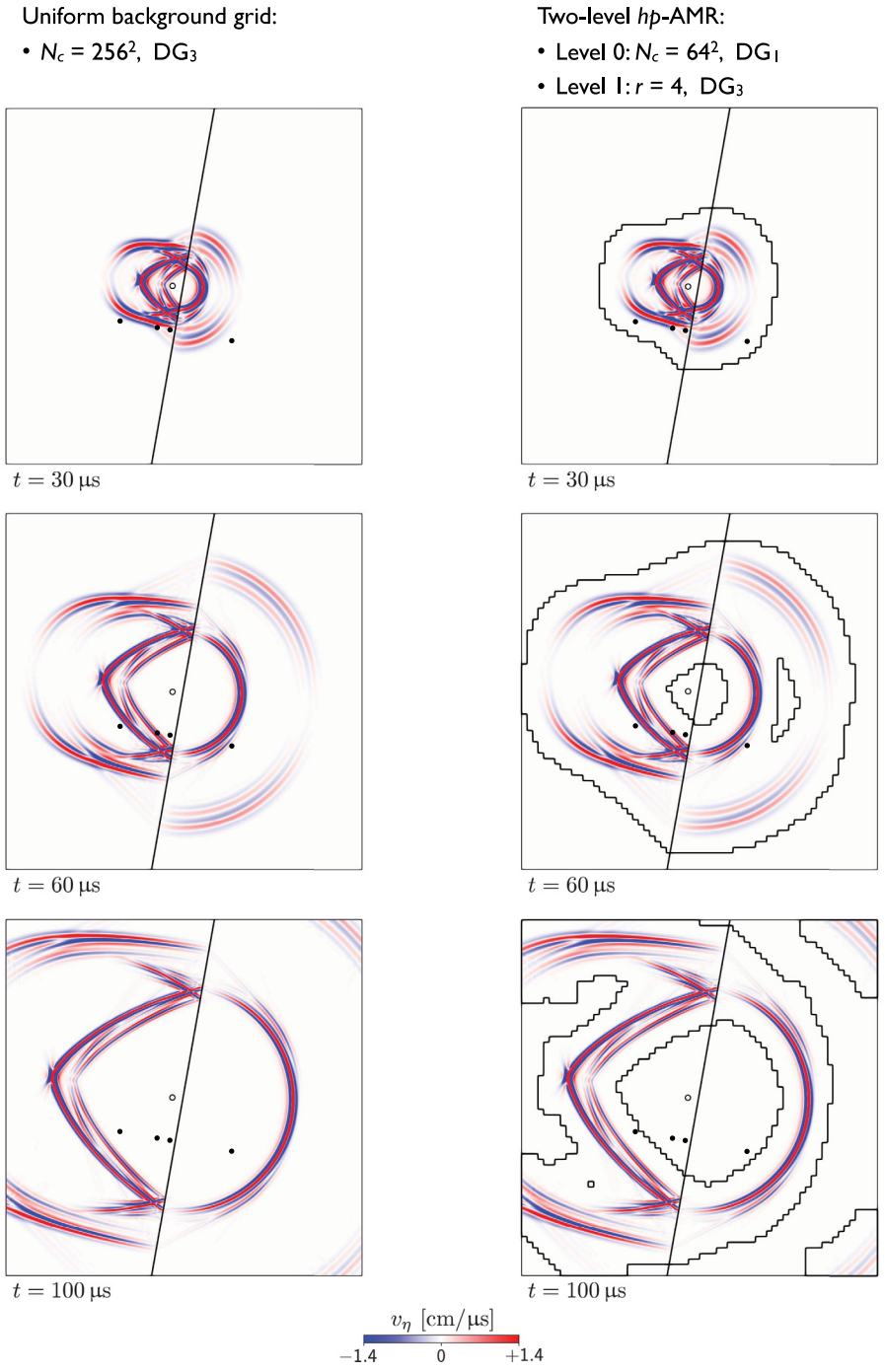

Fig. 23. Snapshots of the velocity component v_η at the time instants $t = 30\,\mu s$, $60\,\mu s$ and $100\,\mu s$ for the single-interface interface problem. (left column) Uniform background grid. (right column) Two-level hp-AMR. In the right column, the stepped lines denote the boundaries between the hp-AMR levels.

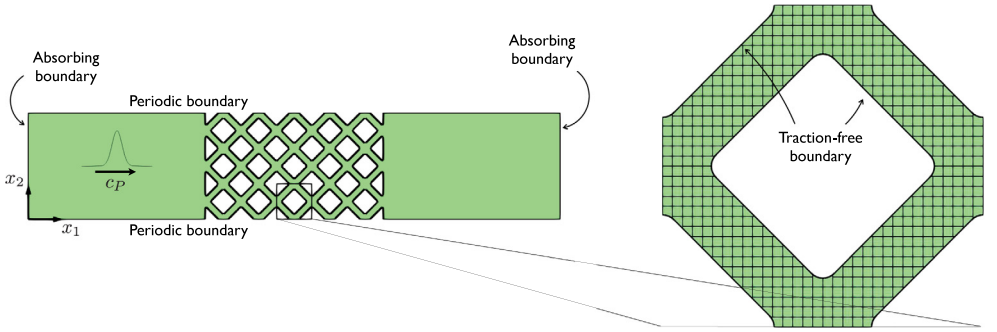

Fig. 24. (Left) Geometry, initial conditions and boundary conditions for the 2D structured solid problem. (Right) Closeup on the geometry and the implicitly-defined mesh of the unit-cell.

4.4. Structured solids

We conclude the numerical results by discussing the application of the present implicit-mesh DG framework to the analysis of an elastic wave propagating through a structured solid. The solid is referred to as *structured* because it is made of a periodically repeating structure, or unit cell, whose characteristic dimensions are smaller than the dimensions of the solid itself. This scenario is common in the area of metamaterial design, where one is interested in obtaining non-conventional macroscopic elastodynamic properties for the solid, such as negative refractive index or negative effective density [68,69], by changing the geometric features and/or by coupling dissimilar materials at the scale the unit cell. Note that the aim of this section is not to provide an investigation of a structured solid in the context of metamaterials but to use the structured solid as an example of a (mildly) complex geometry where the elastodynamic problem can be solved with high-order accuracy in time and space by means of the present framework. For this last set of tests, we also consider non-dimensional units.

The geometry for the considered 2D case is displayed in Fig. 24. The solid consists of two homogeneous ends and a central structured region, whose unit cell is displayed in the right-hand side of Fig. 24. To construct the geometry, we consider three level set functions:

$$\varphi_1(\boldsymbol{x}) = |\sin(\pi(x_1 - x_2 - 1/2))| - W, \quad \varphi_2(\boldsymbol{x}) = |\sin(\pi(x_1 + x_2 - 1/2))| - W \tag{45a}$$

and

$$\varphi_3(\boldsymbol{x}) = (L - x_1)(x_1 - 2L), \tag{45b}$$

which are combined together to introduce a unique level set function as

$$\varphi(\boldsymbol{x}) = 1 - \left[\max\left(0, 1 - \frac{\varphi_1(\boldsymbol{x})}{\delta_1}\right)\right]^{\delta_2} - \left[\max\left(0, 1 - \frac{\varphi_2(\boldsymbol{x})}{\delta_1}\right)\right]^{\delta_2} - \left[\max\left(0, 1 - \frac{\varphi_3(\boldsymbol{x})}{\delta_1}\right)\right]^{\delta_2}. \tag{46}$$

In Eqs. (45), the functions φ_1 and φ_2 define the lattice structure of the unit cell, with W controlling the width of the unit cell's struts, and the function φ_3 controls the location of the transition between the homogeneous ends and the structured region. Meanwhile, Eq. (46) provides a way to blend multiple level set functions where the positive parameters δ_1 and δ_2 control the sharpness of the transition among the level set functions; in particular, a low value of δ_1 and a high value of δ_2 make the transition sharper whereas a high value of δ_1 and a low value of δ_2 make the transition smoother. Here, we use $L = 5$, $W = 0.4$, $\delta_1 = 0.2$ and $\delta_2 = 2$.

The geometry is periodic along the x_2 direction and, owing to its periodicity, the numerical problem is setup in the background rectangle $\mathscr{R} = [0, 3L] \times [0, 1]$ where the implicitly-defined mesh is generated from a background grid consisting of 480×32 cells. The implicitly-defined mesh is partially shown in the right-hand side of Fig. 24 for the unit cell. Absorbing boundary conditions are prescribed at $x_1 = 0$ and $x_1 = 3L$, periodic boundary conditions are prescribed at $x_2 = 0$ and $x_2 = 1$, and zero-traction boundary conditions are prescribed on the zero-contour $\mathscr{L}_\alpha$ of the level set function φ. Absorbing and periodic boundary conditions are chosen to minimize the reflection of the elastic waves from the background rectangle's boundaries while zero-traction boundary conditions are typical of

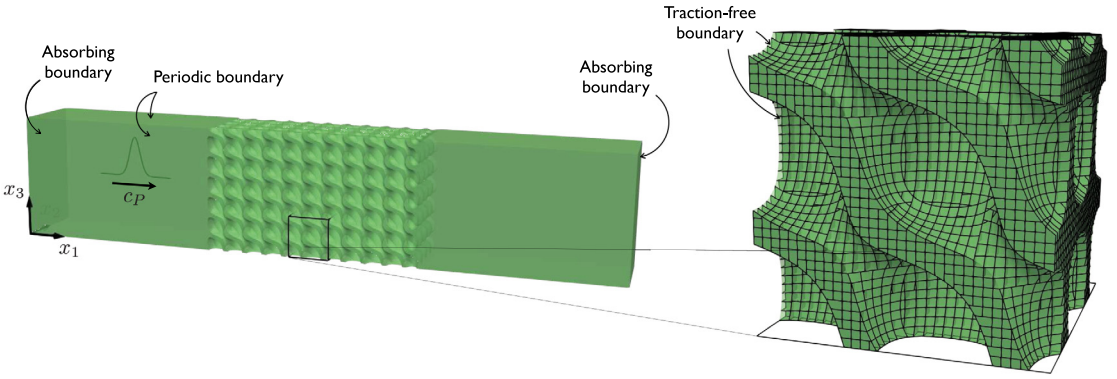

Fig. 25. (Left) Geometry, initial conditions and boundary conditions for the 3D structured solid problem. (Right) Closeup on the geometry and the implicitly-defined mesh of the unit-cell.

single-phase metamaterials; different types of boundary condition could also be considered, especially if finite-size specimens are to be modeled. The solid is assumed isotropic with density $\rho = 1$ and elastic constants defined by the velocity of the P-waves $c_P = 1$ and the velocity of the S-waves $c_S = 0.56$. Initial conditions are prescribed as

$$\boldsymbol{U}_{\alpha 0} = \widetilde{\boldsymbol{U}} e^{-25(x_1-2.5)^2}, \tag{47}$$

where $\widetilde{\boldsymbol{U}}$ is the eigenvector solution of Eq. (28) with $\boldsymbol{\kappa} = (1, 0)$ and $\omega = c_P$; this initiates a wave that propagates from the homogeneous end of the solid to the structured region along the direction of the positive x_1 axis.

Similar to the 2D case, the considered 3D solid consists of two homogeneous ends and a central structured region as shown in Fig. 25. The unit cell is a Schwarz diamond [70] and is displayed in the right-hand side of Fig. 24. Consider the following functions:

$$\varphi_1(\boldsymbol{x}) = \sin(\xi_1)\sin(\xi_2)\sin(\xi_3) + \sin(\xi_1)\cos(\xi_2)\cos(\xi_3) + \cos(\xi_1)\cos(\xi_2)\sin(\xi_3) + \cos(\xi_1)\cos(\xi_2)\cos(\xi_3), \tag{48a}$$

with $\xi_1 \equiv 2\pi x_1$, $\xi_2 \equiv 2\pi x_2$ and $\xi_3 \equiv 2\pi x_3$, and

$$\varphi_2(\boldsymbol{x}) = \begin{cases} x_1/L - 1 & \text{if } x_1 < 3L/2 \\ 2 - x_1/L & \text{if } x_1 \geq 3L/2, \end{cases} \tag{48b}$$

where $L = 5$. Then, to construct the whole geometry, we define the following level set function

$$\varphi(\boldsymbol{x}) = w(\boldsymbol{x})\big(1/2 - \varphi_1(\boldsymbol{x})\big)\big(1/2 + \varphi_1(\boldsymbol{x})\big) + \big(1 - w(\boldsymbol{x})\big)\varphi_2(\boldsymbol{x}), \tag{49}$$

where $w(\boldsymbol{x})$ is a function controlling the transition between the homogeneous region and the structured region and is given by

$$w(\boldsymbol{x}) = \frac{1}{\pi}\left[\operatorname{atan}(50(x_1 - L)) - \operatorname{atan}(50(x_1 - 2L))\right]. \tag{50}$$

The numerical setup of the 3D problem is similar to the numerical setup of the 2D problem: the 3D geometry is periodic along the x_2 and the x_3 directions, and the implicitly-defined mesh is generated from a background grid consisting of $480 \times 32 \times 32$ cells in the background prism $\mathscr{R} = [0, 3L] \times [0, 1] \times [0, 1]$. The right-hand side of Fig. 25 shows the resulting implicitly-defined mesh corresponding to the unit cell. Absorbing boundary conditions are prescribed at $x_1 = 0$ and $x_1 = 3L$, periodic boundary conditions are prescribed at $x_2 = 0$ and $x_2 = 1$ and at $x_3 = 0$ and $x_3 = 1$, and zero-traction boundary conditions are prescribed on the zero-contour $\mathscr{L}_\alpha$ of the level set function φ. The same elastic properties for the 2D case are employed, while initial conditions are given as in Eq. (47) where $\widetilde{\boldsymbol{U}}$ is the eigenvector solution of the 3D version of Eq. (28) with $\boldsymbol{\kappa} = (1, 0, 0)$ and $\omega = c_P$.

Figs. 26 and 27 show a few snapshots of the velocity component v_1 at the time instants $t = 2.5$, 7.5, 12.5, 17.5 and 22.5 for the 2D setup and the 3D setup, respectively. In both cases, it is possible to observe that part of the wave gets reflected by the structured region and part of it gets transmitted, whilst a complex distribution of the mechanical field is induced by the geometry of the structured solid. Moreover, upon recalling that the length of the structured

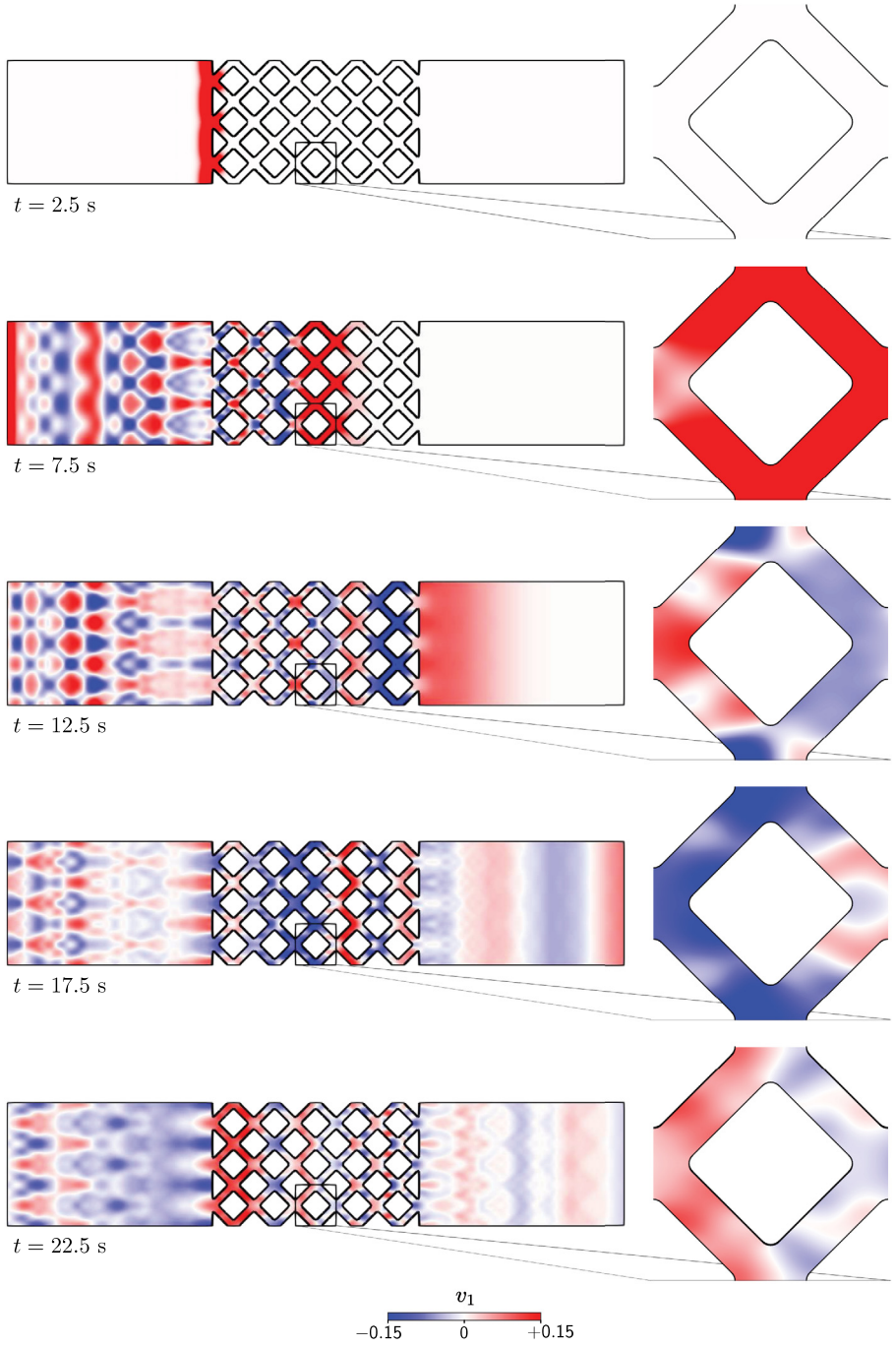

Fig. 26. Snapshots of the velocity component v_1 at the time instants $t = 2.5$, 7.5, 12.5, 17.5 and 22.5 for the 2D structured solid problem.

region is $L = 5$ and the wave travels at $c_P = 1$, it is interesting to notice that, between $t = 2.5$ and $t = 7.5$, the wave is not able to propagate from the beginning to the structured region to its end. This means that, from a macroscopic viewpoint, the geometry of the structured region is responsible for slowing down the wave speed. This is one example of several well-known features of metamaterials. In the context of metamaterials, it would be also possible to analyze the frequency content of the reflected signal and of the transmitted signal and to investigate the

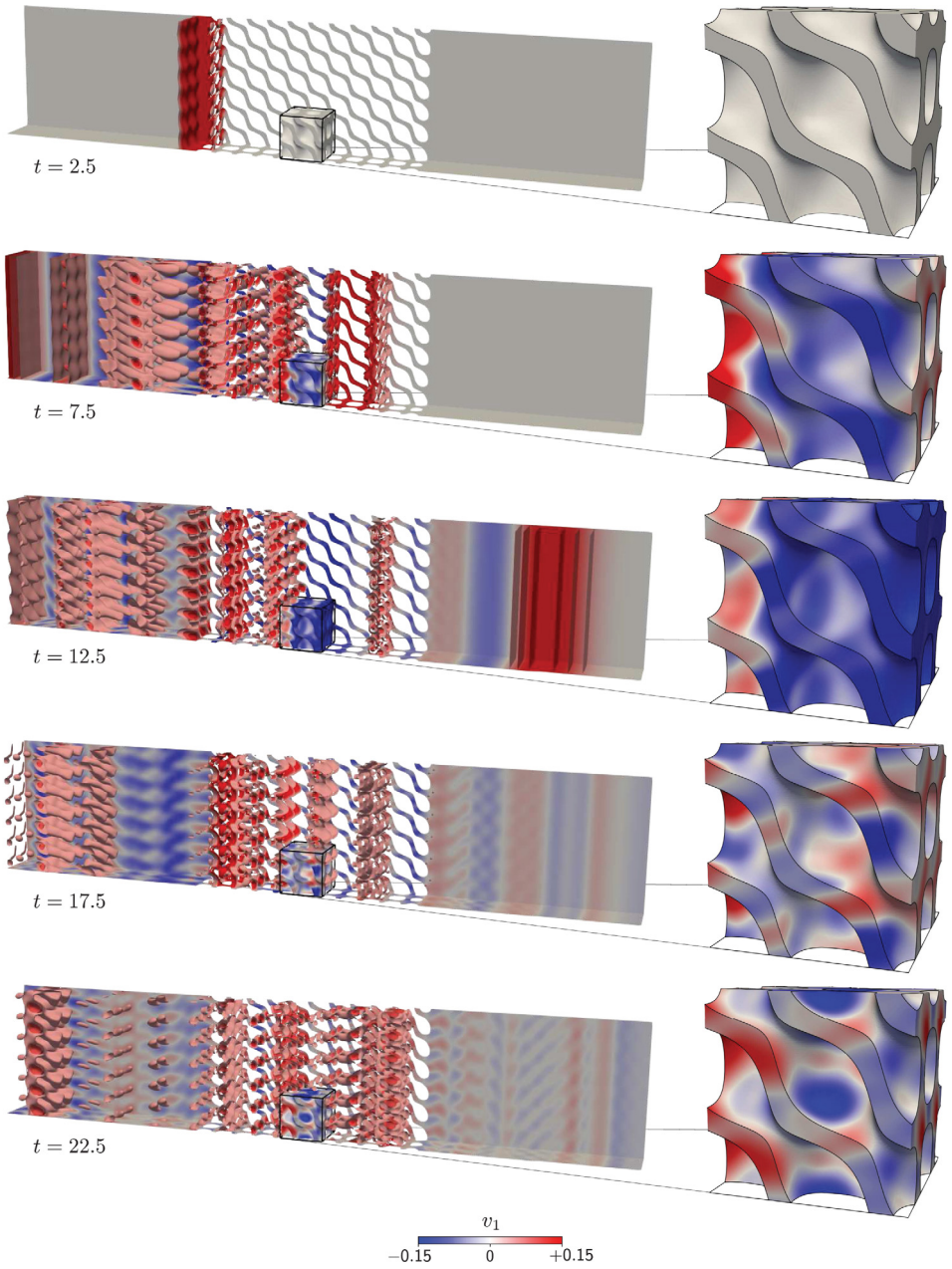

Fig. 27. Snapshots of the velocity component v_1 at the time instants $t = 2.5$, 7.5, 12.5, 17.5 and 22.5 for the 3D structured solid problem.

stop-band properties of the structured solid, which might be considered as a filter for elastic waves. However, these aspects are outside the scope of this paper and the application of the present framework to the analysis and design of metamaterials will be discussed elsewhere.

5. Conclusions

We have presented a discontinuous Galerkin framework for modeling wave propagation in single-phase and bi-phase elastic solid with complex geometries and general anisotropic constitutive behavior. The framework belongs to embedded-boundary methods and is referred to as implicit-mesh DG method because it is based on the use of structured grids where the curved geometries are represented implicitly via level set functions and the domain

discretization is generated by intersecting the level set functions with the grid cells, while a suitable cell-merging technique avoids the presence of overly small cut cells. The novelty of the method regards the space discretization and, in particular, the use of high-order accurate quadrature rules for implicitly-defined domains and boundaries, which allow resolving the presence of the embedded geometries as well as enforcing boundary and interface conditions with high-order accuracy.

Various numerical tests have been considered and discussed, including several hp-convergence analyses in 2D and 3D and for single- and bi-phase solids, a few case studies involving 2D and 3D hp-AMR, as well as an application of the present method to the analysis of waves propagating in 2D and 3D structured solids. The results demonstrate that the method achieves high-order accuracy in the maximum norm and is capable of dealing with implicitly-defined curved geometries, whilst taking advantage of the ease of generation and manipulation of structured grids.

The approach also offers several avenues of further research in the area of elastodynamics. First, we recall that the present DG method has been employed to model waves in linear elastic solids with spatially constant elastic properties. Therefore, a natural extension of the method would be to consider space-varying material properties so as to model functionally-graded materials with complex geometries; similarly, the method could be extended to account for viscoelastic attenuation [71], non-linear elastic waves [72], or additional kinds of material properties and interface conditions, such as those typical of elastic–acoustic coupling, see e.g. [39,40]. Second, it is worth noting that the implicitly-defined technique has been successfully employed for the solution of elliptic PDEs in combination with Local DG methods [73] or Interior Penalty DG methods [33]; therefore, the proposed approach may naturally be extended to the solution of elastodynamics in its primal form, see e.g. [41]. Third, we remark that the numerical tests feature a smooth geometry implicitly defined by a smooth level set function, including for the case of the structured solids wherein multiple level set functions were blended together to form a unique level set function; nevertheless, this does not represent a requirement (or limitation) for the present implicit-mesh DG framework, which can be used in combination with more complex geometry definitions, provided that the corresponding quadrature rules be available. To this end, one possibility is to leverage the high-order accurate quadrature algorithms recently developed in Ref. [74], which can handle various kinds of complex geometry, such as intersecting/overlapping domains containing corners, junctions, tunnels, and multiple components, among other kinds of interfacial features. This would allow extending the present approach to the modeling of engineering components, which are typically characterized by more complex geometrical features than the ones considered in this work. Finally, we note that we have not thoroughly discussed here the performance of the implementation; nevertheless, upon recalling that the proposed approach is a fully explicit scheme, a few comments regarding the computational cost are worth mentioning:

- When a static mesh is employed, all quantities related to the implicitly-defined mesh may be precomputed and stored at the beginning of the simulations; these include the quadrature nodes and weights of the elements and element boundaries, the mass matrix given in Eq. (15) and, for static hp-AMR schemes, the interpolation operator given in Eq. (18). Then, the computational cost of the proposed scheme is in general linear in the number of mesh elements and linear in the number of time steps.
- In case of dynamic meshes, even though the aforementioned quantities are computed at each regridding, i.e. every time the mesh is updated, profiling experiment reveals that the cost of constructing the implicitly-defined mesh is still negligible compared to the evaluation of the right-hand side of Eq. (22). When combined with the smaller number of mesh elements required by adaptively refined meshes, this confirms the benefits of hp-AMR schemes.
- In addition to the points above, another important contributing factor to the computational cost of the method is the inter-processor communication time for parallel simulations and/or the host-device communication time for simulations run using modern accelerators, such as general-purpose graphical processing units. In such cases, the computational cost of data movement is mainly related to the distribution of the computational load and the memory layout among the processors, whose design naturally impacts the efficiency of the numerical scheme.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported in part by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration, through U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under contract DE-AC02-05CH11231. It was also supported by the Applied Mathematics Program of the U.S. Department of Energy Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231. Some computations used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

[1] C. Chapman, Fundamentals of Seismic Wave Propagation, Cambridge University Press, 2004.
[2] M. Mitra, S. Gopalakrishnan, Guided wave based structural health monitoring: A review, Smart Mater. Struct. 25 (5) (2016) 053001.
[3] A. Srivastava, Elastic metamaterials and dynamic homogenization: A review, Int. J. Smart Nano Mater. 6 (1) (2015) 41–60.
[4] L. Wu, Y. Wang, K. Chuang, F. Wu, Q. Wang, W. Lin, H. Jiang, A brief review of dynamic mechanical metamaterials for mechanical energy manipulation, Mater. Today (2020).
[5] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009.
[6] M. Aftosmis, M. Berger, G. Adomavicius, A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries, in: 38th Aerospace Sciences Meeting and Exhibit, 2000, p. 808.
[7] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, Internat. J. Numer. Methods Engrg. 104 (7) (2015) 472–501.
[8] D. Komatitsch, C. Barnes, J. Tromp, Simulation of anisotropic wave propagation based upon a spectral element method, Geophysics 65 (4) (2000) 1251–1260.
[9] K.-J. Bathe, Finite Element Procedures, Klaus-Jurgen Bathe, 2006.
[10] S. Ham, K.-J. Bathe, A finite element method enriched for wave propagation problems, Comput. Struct. 94 (2012) 1–12.
[11] J. Parvizian, A. Düster, E. Rank, Finite cell method, Comput. Mech. 41 (1) (2007) 121–133.
[12] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Comput. Methods Appl. Mech. Engrg. 197 (45–48) (2008) 3768–3782.
[13] M. Joulaian, S. Duczek, U. Gabbert, A. Düster, Finite and spectral cell method for wave propagation in heterogeneous materials, Comput. Mech. 54 (3) (2014) 661–675.
[14] M. Elhaddad, N. Zander, S. Kollmannsberger, A. Shadavakhsh, V. Nübel, E. Rank, Finite cell method: High-order structural dynamics for complex geometries, Int. J. Struct. Stab. Dyn. 15 (07) (2015) 1540018.
[15] A. Abedian, J. Parvizian, A. Düster, H. Khademyzadeh, E. Rank, Performance of different integration schemes in facing discontinuities in the finite cell method, Int. J. Comput. Methods 10 (03) (2013) 1350002.
[16] D. Motamedi, S. Mohammadi, Dynamic crack propagation analysis of orthotropic media by the extended finite element method, Int. J. Fract. 161 (1) (2010) 21–39.
[17] E.B. Chin, A.A. Mokhtari, A. Srivastava, N. Sukumar, Spectral extended finite element method for band structure calculations in phononic crystals, J. Comput. Phys. 427 (2021) 110066.
[18] I. Benedetti, V. Gulizzi, A. Milazzo, X-Ritz solution for nonlinear free vibrations of plates with embedded cracks, Aerotec. Missili Spaz. 98 (1) (2019) 75–83.
[19] K. Park, H. Chi, G.H. Paulino, On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration, Comput. Methods Appl. Mech. Engrg. 356 (2019) 669–684.
[20] G.D. Manolis, A comparative study on three boundary element method approaches to problems in elastodynamics, Internat. J. Numer. Methods Engrg. 19 (1) (1983) 73–91.
[21] I. Benedetti, M.H. Aliabadi, A fast hierarchical dual boundary element method for three-dimensional elastodynamic crack problems, Internat. J. Numer. Methods Engrg. 84 (9) (2010) 1038–1067.
[22] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2002) 1749–1779.
[23] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems, J. Comput. Phys. 141 (2) (1998) 199–224.
[24] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, J. Comput. Phys. 183 (2) (2002) 508–532.
[25] O. Zanotti, F. Fambri, M. Dumbser, Solving the relativistic magnetohydrodynamics equations with ADER discontinuous Galerkin methods, a posteriori subcell limiting and adaptive mesh refinement, Mon. Not. R. Astron. Soc. 452 (3) (2015) 3010–3029.
[26] A. Cangiani, Z. Dong, E.H. Georgoulis, Hp-version space-time discontinuous Galerkin methods for parabolic problems on prismatic meshes, SIAM J. Sci. Comput. 39 (4) (2017) A1251–A1279.
[27] A. Cangiani, Z. Dong, E.H. Georgoulis, P. Houston, hp-Version Discontinuous Galerkin Methods on Polygonal and Polyhedral Meshes, Springer, 2017.

[28] P.F. Antonietti, G. Pennesi, V-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes, J. Sci. Comput. 78 (1) (2019) 625–652.
[29] A. Johansson, M.G. Larson, A high order discontinuous Galerkin nitsche method for elliptic problems with fictitious boundary, Numer. Math. 123 (4) (2013) 607–628.
[30] R. Saye, Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: Part I, J. Comput. Phys. 344 (2017) 647–682.
[31] R. Saye, Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: Part II, J. Comput. Phys. 344 (2017) 683–723.
[32] V. Gulizzi, A.S. Almgren, J.B. Bell, A coupled discontinuous Galerkin-finite volume framework for solving gas dynamics over embedded geometries, J. Comput. Phys. 450 (2022) 110861.
[33] V. Gulizzi, I. Benedetti, A. Milazzo, An implicit mesh discontinuous Galerkin formulation for higher-order plate theories, Mech. Adv. Mater. Struct. 27 (17) (2020) 1494–1508.
[34] P.F. Antonietti, C. Marcati, I. Mazzieri, A. Quarteroni, High order discontinuous Galerkin methods on simplicial elements for the elastodynamics equation, Numer. Algorithms 71 (1) (2016) 181–206.
[35] D. Appelö, T. Hagstrom, An energy-based discontinuous Galerkin discretization of the elastic wave equation in second order form, Comput. Methods Appl. Mech. Engrg. 338 (2018) 362–391.
[36] P.F. Antonietti, I. Mazzieri, High-order discontinuous Galerkin methods for the elastodynamics equation on polygonal and polyhedral meshes, Comput. Methods Appl. Mech. Engrg. 342 (2018) 414–437.
[37] M. Dumbser, M. Käser, An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—II. The three-dimensional isotropic case, Geophys. J. Int. 167 (1) (2006) 319–336.
[38] J. de la Puente, M. Käser, M. Dumbser, H. Igel, An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes-IV. Anisotropy, Geophys. J. Int. 169 (3) (2007) 1210–1228.
[39] L.C. Wilcox, G. Stadler, C. Burstedde, O. Ghattas, A high-order discontinuous Galerkin method for wave propagation through coupled elastic–acoustic media, J. Comput. Phys. 229 (24) (2010) 9373–9396.
[40] Q. Zhan, Q. Ren, M. Zhuang, Q. Sun, Q.H. Liu, An exact Riemann solver for wave propagation in arbitrary anisotropic elastic media with fluid coupling, Comput. Methods Appl. Mech. Engrg. 329 (2018) 24–39.
[41] P.F. Antonietti, B.A. De Dios, I. Mazzieri, A. Quarteroni, Stability analysis of discontinuous Galerkin approximations to the elastodynamics problem, J. Sci. Comput. 68 (1) (2016) 143–170.
[42] M. Tavelli, M. Dumbser, D.E. Charrier, L. Rannabauer, T. Weinzierl, M. Bader, A simple diffuse interface approach on adaptive cartesian grids for the linear elastic wave equations with complex topography, J. Comput. Phys. 386 (2019) 158–189.
[43] S. Sticko, G. Kreiss, Higher order cut finite elements for the wave equation, J. Sci. Comput. 80 (3) (2019) 1867–1887.
[44] S. Adjerid, K. Moon, An immersed discontinuous Galerkin method for acoustic wave propagation in inhomogeneous media, SIAM J. Sci. Comput. 41 (1) (2019) A139–A162.
[45] R. Saye, Interfacial gauge methods for incompressible fluid dynamics, Sci. Adv. 2 (6) (2016) e1501869.
[46] R. Saye, Fast multigrid solution of high-order accurate multiphase Stokes problems, Commun. Appl. Math. Comput. Sci. 15 (2) (2020) 147–196.
[47] V. Gulizzi, I. Benedetti, A. Milazzo, A high-resolution layer-wise discontinuous Galerkin formulation for multilayered composite plates, Compos. Struct. 242 (2020) 112137.
[48] G. Guarino, V. Gulizzi, A. Milazzo, High-fidelity analysis of multilayered shells with cut-outs via the discontinuous Galerkin method, Compos. Struct. (2021) 114499.
[49] R. Saye, High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles, SIAM J. Sci. Comput. 37 (2) (2015) A993–A1019.
[50] J.M. Carcione, Wave Fields in Real Media: Wave Propagation in Anisotropic, Anelastic, Porous and Electromagnetic Media, Elsevier, 2007.
[51] R. Saye, Algoim–algorithms for implicitly defined geometry, level set methods, and Voronoi implicit interface methods, 2019, https://algoim.github.io/.
[52] E.B. Chin, J.B. Lasserre, N. Sukumar, Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra, Comput. Mech. 56 (6) (2015) 967–981.
[53] P.F. Antonietti, P. Houston, G. Pennesi, Fast numerical integration on polytopic meshes with applications to discontinuous Galerkin finite element methods, J. Sci. Comput. 77 (3) (2018) 1339–1370.
[54] Y. Sudhakar, A. Sommariva, M. Vianello, W.A. Wall, On the use of compressed polyhedral quadrature formulas in embedded interface methods, SIAM J. Sci. Comput. 39 (3) (2017) B571–B587.
[55] W. Zhang, A. Almgren, V. Beckner, J. Bell, J. Blaschke, C. Chan, M. Day, B. Friesen, K. Gott, D. Graves, et al., AMReX: A framework for block-structured adaptive mesh refinement, J. Open Source Softw. 4 (37) (2019) 1370.
[56] D. Fortunato, C.H. Rycroft, R. Saye, Efficient operator-coarsening multigrid schemes for local discontinuous Galerkin methods, SIAM J. Sci. Comput. 41 (6) (2019) A3913–A3937.
[57] B. Cockburn, C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.
[58] I. Benedetti, M.H. Aliabadi, A three-dimensional grain boundary formulation for microstructural modeling of polycrystalline materials, Comput. Mater. Sci. 67 (2013) 249–260.

[59] V. Gulizzi, I. Benedetti, A. Milazzo, A novel boundary element formulation for anisotropic fracture mechanics, Theor. Appl. Fract. Mech. 104 (2019) 102329.
[60] J.T. Browaeys, S. Chevrot, Decomposition of the elastic tensor and geophysical applications, Geophys. J. Int. 159 (2) (2004) 667–678.
[61] W. FG201, L.-Y. Fu, Numerical dispersion analysis of discontinuous Galerkin method with different basis functions for acoustic and elastic wave equations, Geophysics 83 (3) (2018) T87–T101.
[62] P. Berg, F. If, P. Nielsen, O. Skovgaard, K. Helbig, Analytical reference solutions, Model. Earth Oil Explor. 77 (1994) 421–427.
[63] E. Kausel, Lamb's problem at its simplest, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 469 (2149) (2013) 20120462.
[64] D. Komatitsch, J.-P. Vilotte, The spectral element method: An efficient tool to simulate the seismic response of 2D and 3D geological structures, Bull. Seismol. Soc. Am. 88 (2) (1998) 368–392.
[65] M. Käser, M. Dumbser, An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—I. The two-dimensional isotropic case with external source terms, Geophys. J. Int. 166 (2) (2006) 855–877.
[66] O. Zanotti, F. Fambri, M. Dumbser, A. Hidalgo, Space–time adaptive ADER discontinuous Galerkin finite element schemes with a posteriori sub-cell finite volume limiting, Comput. & Fluids 118 (2015) 204–224.
[67] J.M. Carcione, D. Kosloff, R. Kosloff, Wave-propagation simulation in an elastic anisotropic (transversely isotropic) solid, Quart. J. Mech. Appl. Math. 41 (3) (1988) 319–346.
[68] R. Zhu, X.N. Liu, G.K. Hu, C.T. Sun, G.L. Huang, Negative refraction of elastic waves at the deep-subwavelength scale in a single-phase metamaterial, Nature Commun. 5 (1) (2014) 1–8.
[69] A.A. Mokhtari, Y. Lu, A. Srivastava, On the emergence of negative effective density and modulus in 2-phase phononic crystals, J. Mech. Phys. Solids 126 (2019) 256–271.
[70] T. Maconachie, M. Leary, B. Lozanovski, X. Zhang, M. Qian, O. Faruque, M. Brandt, SLM lattice structures: Properties, performance, applications and challenges, Mater. Des. 183 (2019) 108137.
[71] M. Käser, M. Dumbser, J. De La Puente, H. Igel, An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—III. Viscoelastic attenuation, Geophys. J. Int. 168 (1) (2007) 224–242.
[72] O. Bou Matar, P.-Y. Guerder, Y. Li, B. Vandewoestyne, K. Van Den Abeele, A nodal discontinuous Galerkin finite element method for nonlinear elastic wave propagation, J. Acoust. Soc. Am. 131 (5) (2012) 3650–3663.
[73] R.I. Saye, Efficient multigrid solution of elliptic interface problems using viscosity-upwinded local discontinuous Galerkin methods, Commun. Appl. Math. Comput. Sci. 14 (2) (2019) 247–283.
[74] R.I. Saye, High-order quadrature on multi-component domains implicitly defined by multivariate polynomials, J. Comput. Phys. 448 (2022) 110720.