Journal of Computational Physics 233 (2013) 430–448

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A second-order sharp numerical method for solving the linear elasticity equations on irregular domains and adaptive grids – Application to shape optimization

Maxime Theillard [a,*], Landry Fokoua Djodom [b], Jean-Léopold Vié [d], Frédéric Gibou [a,c]

[a] Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106, United States
[b] Graduate Aeronautical Laboratories, California Institute of Technology, Pasadena, CA 91125, United States
[c] Department of Computer Science, University of California, Santa Barbara, CA 93106, United States
[d] Ecole Polytechnique, 91210 Palaiseau, France

ARTICLE INFO

Article history:
Received 23 September 2011
Received in revised form 22 May 2012
Accepted 4 September 2012
Available online 17 September 2012

Keywords:
Second-order discretization
Linear elasticity
Hybrid finite volume/finite difference
Level-set
Irregular domains
Octree data structure
Quadtree/octree data structure
Non-graded adaptive grid

ABSTRACT

We present a numerical method for solving the equations of linear elasticity on irregular domains in two and three spatial dimensions. We combine a finite volume and a finite difference approaches to derive discretizations that produce second-order accurate solutions in the L^∞-norm. Our discretization is 'sharp' in the sense that the physical boundary conditions (mixed Dirichlet/Neumann-type) are imposed *at* the interface and the solution is computed inside the irregular domain only, without the need of smearing the solution across the interface. The irregular domain is represented implicitly using a level-set function so that this approach is applicable to free moving boundary problems; we provide a simple example of shape optimization to illustrate this capability. In addition, we provide an extension of our method to the case of adaptive meshes in both two and three spatial dimensions: we use non-graded quadtree (2D) and octree (3D) data structures to represent the grid that is automatically refined near the irregular domain's boundary. This extension to quadtree/octree grids produces second-order accurate solutions albeit non-symmetric linear systems, due to the node-based sampling nature of the approach. However, the linear system can be solved with simple linear solvers; in this work we use the BICGSTAB algorithm.

Published by Elsevier Inc.

1. Introduction

The equations of linear elasticity are used extensively in structural analysis and engineering design to understand and simulate how solid objects deform and respond to internal stress under prescribed loading conditions. Finite element methods are one of the most widely used techniques to solve the equations of linear elasticity (see e.g. [8]). One of the strong features of the FEM approach is that it provides a theoretical framework that allows the definition of *a priori* error estimates that can be used to design automatic refinement criteria. In addition, the theoretical framework allows to perform convergence analysis using the norm induced by the solution space. However, a computational difficulty with this approach is that the mesh needs to conform to the irregular domain's boundary and meshing procedures that guarantee the quality of the mesh can be computationally expensive. The case where the domain's boundary evolves in time, as it is the case in shape optimization problems for example, exacerbates this difficulty as the domain needs to be frequently remeshed. The extended finite element approach (XFEM) avoids the difficulty associated with frequent remeshing since the mesh need not conform to

* Corresponding author.
E-mail address: maxime_theillard@umail.ucsb.edu (M. Theillard).

0021-9991/$ - see front matter Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.jcp.2012.09.002

the domain's boundaries. Remarkable numerical results are achieved even though the theoretical framework of XFEM is still under development [6,4].

Numerical methods that implicitly describe the irregular domain's boundary originated from the pioneering work of Peskin (the immersed boundary method [IBM]) to model blood flows in the heart [30,31]. Since then, implicit representations have been the focus of modern numerical methods and the main quest has been on how to impose the different boundary conditions at the irregular domain's boundary. In the IBM framework, boundary conditions are modeled as a source term that is localized on the interface through the use of a discretized δ-function. This method therefore smears out the solution across the interface. The immersed interface method (IIM), introduced by Li and LeVeque modifies the discretization of differential operators near the irregular boundary using Taylor-type expansions to impose boundary conditions in a sharp (as opposed to smeared out) fashion [17]. In this case the overall discretization is second-order accurate in the L^∞-norm. The IIM framework is therefore accurate but can be difficult to generalize to three spatial dimensions, especially when cross-derivative terms are present.

The level-set method, originally developed to study curvature driven flows [28] is another implicit method that has been very successful in simulating free moving boundary problems [27,35]. Allaire et al. [11] used the level-set method in the case of shape optimization, generalizing the two previous level-set-based works of Wiegmann and Sethian [16] and that of Santosa and Osher [36]. This method considers 'ersatz materials', where the solution inside the irregular domain is padded with a ghost material with small characteristic parameters. An advantage of this approach is its simplicity; a potential drawback is that the extension by a weak material alters the original elasticity problem in the sense that the boundary conditions are not imposed in a 'sharp' manner, which in turn may affect the accuracy.

Imposing boundary conditions in a 'sharp' fashion in the level-set framework is often performed using the ghost-fluid method. Originally developed for compressible flows where jump conditions must be imposed in order to avoid the $O(1)$ numerical errors near shocks and contact discontinuities [10], the ghost-fluid method has been applied to a wide range of problems: for example, Liu et al. derived a first-order ghost-fluid approach for the Poisson equation on irregular domains where jump conditions on the solution and its gradient are to be imposed [18]. Nguyen et al. applied this method to the simulation of flame propagation [26]. Gibou et al. developed second-order [12] and fourth-order [13] accurate ghost-fluid methods for the Poisson and the heat equation where Dirichlet boundary conditions are imposed. Enright et al. used this approach in the context of free surface flows [9], Gibou et al. used it into simulate the Stefan problem [14] and Mirzadeh et al. applied it to the case of the nonlinear Poisson–Boltzmann equations [24].

Although very successful for the problems mentioned above, the ghost-fluid method is a less natural approach for imposing boundary conditions involving fluxes, i.e. Neumann or Robin type boundary conditions. Finite volume methods or variational approaches are in those cases better suited because the use of the divergence theorem naturally reveals explicitly the normal derivative of the solution (i.e. fluxes). Almgren et al. introduced a cut-cell variational approach [2]. Papac et al. developed a second-order accurate symmetric discretization of the Poisson and the heat equations with Robin boundary conditions, as well as to the Stefan problem [29]. Helgadottir proposed a numerical method for the nonlinear Poisson–Boltzmann equations [15]. A second-order accurate finite volume approach for incompressible flows in arbitrary geometries has been developed and analyzed in Purvis and Burkhalter [32] and Ng et al. [25]. In this case, the standard projection method [7] requires the solution of a Poisson equation with Neumann boundary conditions imposed on irregular domains.

In this paper, we propose a simple sharp numerical approach for the elasticity problem that avoids the need of the ersatz material. We consider the case where the domain's boundary is subject to a combination of Neumann-type and Dirichlet boundary conditions. Dirichlet boundary conditions are imposed on the interface using an approach similar to the work of Gibou [12] and Chen et al. [5]. We then derive a finite volume approach that easily handles the Neumann-type boundary condition at the domain's boundary, which is difficult to discretize on arbitrary domains with standard finite differences. The linear system is symmetric and the solution is second-order accurate in the L^∞-norm.

The method is also extended to the case of nonuniform grids, where the finest cells are automatically located near the interface. Given the Elliptic nature of the equations of linear elasticity, the solutions are expected to be smooth except near the interface, where kinks may occur. It follows that an adaptive scheme that imposes the finest cells near the interface will be able to accurately resolve the solution while significantly reducing the overall computational complexity. We make use of the framework introduced by Min et al. [22] for solving partial differential equations on quadtree/octree data structures. This approach allows the design of second-order finite difference schemes on non-graded adaptive grids, i.e. on grids for which the ratio between adjacent cells is not bounded. This, in turn, gives a straightforward and efficient mesh generation procedure. This framework has been applied to the Navier–Stokes equations, the Poisson and the heat equations, Stefan-type problems and the Poisson–Boltzmann equation [20,15,5], but has never been developed for the linear elasticity equations. In the case of non-graded grids, the linear system produced from discretizing the elasticity equations is non-symmetric (see [23]), but it is still represented by an M-matrix guaranteeing convergence of iterative solvers.

2. Governing equations

Let Ω be a computational domain and Ω^- and Ω^+ two complementary subsets of Ω separated by an interface Γ. We use a level-set function ϕ to define the irregular domain $\Omega^- = \{\mathbf{x} : \phi(\mathbf{x}) \leqslant 0\}$ and $\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$. At equilibrium the governing equation of linear elasticity is:

$$-\nabla \cdot \underline{\underline{\sigma}} = \mathbf{f} \quad \text{on } \Omega^-, \tag{1}$$

where the stress tensor $\underline{\underline{\sigma}}$ is related to the displacement field through Hooke's law:

$$\underline{\underline{\sigma}} = 2\mu \underline{\underline{e}}(\mathbf{u}) + \lambda \mathrm{Tr}(\underline{\underline{e}}(\mathbf{u}))\mathrm{Id}, \tag{2}$$

where

$$\underline{\underline{e}}(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^\top\right).$$

Id is the identity matrix, Tr refers to the trace operator, λ and μ are the Lamé coefficients of the material and f is the given body forces. The displacement $\mathbf{u}$ is the unknown of the problem. We assume that Γ is split into two different parts Γ_D and Γ_N where Dirichlet and Neumann-type (traction) boundary conditions, respectively, are imposed:

$$\mathbf{u} = \mathbf{u}_D \quad \text{on } \Gamma_D, \tag{3}$$

$$\underline{\underline{\sigma}} \cdot \underline{n} = \mathbf{g} \quad \text{on } \Gamma_N, \tag{4}$$

where $\mathbf{u}_D$ and $\mathbf{g}$ are given.

3. Numerical method

In this section, we present our method for solving the equations of linear elasticity in two and three spatial dimensions. The approach is presented in the case of uniform grids in Section 3.1 before being extended to the case of quadtree/octree adaptive grids in Section 3.2.

3.1. Discretizations on uniform grids

The computational domain is geometrically discretized by a uniform mesh. The solution is stored at the vertices (i,j) of a uniform grid in two spatial dimensions (respectively (i,j,k) in three spatial dimensions). In two spatial dimensions, for each node (i,j) we consider the cell $C_{ij} = \left[i-\frac{1}{2}, i+\frac{1}{2}\right] \times \left[j-\frac{1}{2}, j+\frac{1}{2}\right]$, so that the given node is located at the center of the cell, as depicted in 1. Likewise in three spatial dimensions, the node (i,j,k) is surrounded by the cubic cell $C_{ijk} = \left[i-\frac{1}{2}, i+\frac{1}{2}\right] \times \left[j-\frac{1}{2}, j+\frac{1}{2}\right] \times \left[k-\frac{1}{2}, k+\frac{1}{2}\right]$. We describe the discretization in the case where the boundary is subject to a Neumann-type boundary condition. Dirichlet boundary conditions are handled using an approach similar to the work of Gibou et al. [12] and Chen et al. [5].

Integrating the equilibrium Eq. (1) over $C^-_{ijk} \triangleq C_{ijk} \cap \Omega^-$ and applying the divergence theorem leads to:

$$\int_{\partial C^-_{ijk}} -\underline{\underline{\sigma}} \cdot \mathbf{n}\, ds = \int_{C^-_{ijk}} \mathbf{f}\, dx.$$

Since the boundary ∂C^-_{ijk} has two components, the faces of grid cell $\partial C^-_{ijk} \backslash \Gamma$ and the portion of the interface crossing the cell $C^-_{ijk} \cap \Gamma$, we consider separately the contribution of the two components. Applying the traction boundary condition (4), we have:

$$\int_{\partial C^-_{ijk} \backslash \Gamma} -\underline{\underline{\sigma}} \cdot \mathbf{n}\, ds = \int_{C^-_{ijk} \cap \Gamma} \mathbf{g}\, ds + \int_{C^-_{ijk}} \mathbf{f}\, dx. \tag{5}$$

The approximation of the right-hand-sides requires the integration of a known quantity over the sets $\partial C^-_{ijk} \backslash \Gamma$ and C^-_{ijk}. In this work, we use the geometric integration procedures introduced in [21]: this approach is based on a decomposition of the grid into simplices, i.e. in triangles in two spatial dimensions and tetrahedra in three spatial dimensions. Then line, surface or volume integrals are readily obtained using standard second-order accurate quadrature rules. This approach has been shown to be robust to the perturbation of the location of an irregular domain with respect the grid. In Sections 3.1.1 and 3.1.2, we focus on the approximation of the left-hand-side of Eq. (5).

3.1.1. Finite volume approach in two spatial dimensions

Eq. (5) leads to two equations:

$$-\int_{\partial C^-_{ij} \backslash \Gamma} \left[\left((\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} \right) N_1 + \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) N_2 \right] ds = \int_{C^-_{ij}} f_1 dx + \int_{C^-_{ij} \bigcap \Gamma} g_1 \tag{6}$$

and

$$-\int_{\partial C^-_{ij} \backslash \Gamma} \left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) N_1 + \left((\lambda + 2\mu) \frac{\partial v}{\partial y} + \lambda \frac{\partial u}{\partial x} \right) N_2 \right] ds = \int_{C^-_{ij}} f_2 dx + \int_{C^-_{ij} \bigcap \Gamma} g_2. \tag{7}$$

As depicted in 1, consider the vertical right face F_1 represented by the portion of the segment $(i+\frac{1}{2}) \times (j-\frac{1}{2}, j+\frac{1}{2})$ covered by the irregular domain. For this segment, the normal is given by $N_1 = 1$ and $N_2 = 0$. The integral $\int_{F_1} \frac{\partial u}{\partial x} ds$ is approximated using the standard central differencing formulas:

$$\int_{F_1} \frac{\partial u}{\partial x} ds = L_1 \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x^3), \tag{8}$$

where Δx is the step size in the x-direction and L_1 is the length of F_1 (see Fig. 1). The discretization of the integral $\int_{F_1} \frac{\partial u}{\partial y} ds$ needs to ensure that it only makes use of well-defined nodes (meaning that their surrounding cell C_{ij} intersects the inside domain Ω^-). This is achieved by using the following first-order discretization:

$$\int_{F_1} \frac{\partial u}{\partial y} ds = \frac{L_T}{2} \left(\frac{u_{ij+1} - u_{ij}}{\Delta y} + \frac{u_{i+1,j+1} - u_{i+1,j}}{\Delta y} \right) + \frac{L_B}{2} \left(\frac{u_{ij} - u_{ij-1}}{\Delta y} + \frac{u_{i+1,j} - u_{i+1,j-1}}{\Delta y} \right) + O(\Delta y^2), \tag{9}$$

where the coefficients L_T and L_B are defined as:

$$\begin{cases} L_T = \Delta y \frac{\phi_T}{\phi_T - \phi_B} & L_B = 0 \quad \text{if} \quad \phi_T < 0 \quad \text{and} \quad \phi_B \geqslant 0, \\ L_B = \Delta y \frac{\phi_B}{\phi_T - \phi_B} & L_T = 0 \quad \text{if} \quad \phi_B < 0 \quad \text{and} \quad \phi_T \geqslant 0, \\ L_T = L_B = \frac{\Delta y}{2} & \quad \text{if} \quad \phi_T < 0 \quad \text{and} \quad \phi_B < 0, \\ L_B = L_T = 0 & \quad \text{if} \quad \phi_B \geqslant 0 \quad \text{and} \quad \phi_T \geqslant 0, \end{cases}$$

where $\phi_T = \phi_{i+\frac{1}{2}j+\frac{1}{2}}$ and $\phi_B = \phi_{i+\frac{1}{2}j-\frac{1}{2}}$. The same integral approximations hold for the displacement component v. In the case scenario represented in Fig. 1, $L_T = 0$ and $L_B = L_1$. It bears emphasis that these coefficients, which represent length portions covered by the irregular domain, have been chosen to ensure that Eq. (9) only involves well-defined nodes. For example, the left hand side of Eq. (6), on the face F_1, gives:

$$\begin{aligned} -\int_{F_1} \left((\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} \right) ds = &-(\lambda + 2\mu)(L_B + L_T) \frac{u_{i+1,j} - u_{i,j}}{\Delta x} - \lambda L_B \left(\frac{v_{ij} - v_{ij-1}}{2\Delta y} + \frac{v_{i+1,j} - v_{i+1,j-1}}{2\Delta y} \right) \\ &- \lambda L_T \left(\frac{v_{ij+1} - v_{ij}}{2\Delta y} + \frac{v_{i+1,j+1} - v_{i+1,j}}{2\Delta y} \right). \end{aligned} \tag{10}$$

The same methodology is applied to the other faces. This finite volume approach is performed for the cells of all the nodes in the computational box Ω which intersect with the domain Ω^-. This includes then nodes outside of the region Ω^- for which the corresponding cells still intersect Ω^-. The final discretizations for (6) and (7) are obtained by summing up all the contributions on the four faces, which form two different rows in the resulting linear system. If N is the number of grid points involved in the schemes, the resulting linear system has $2N$ rows.

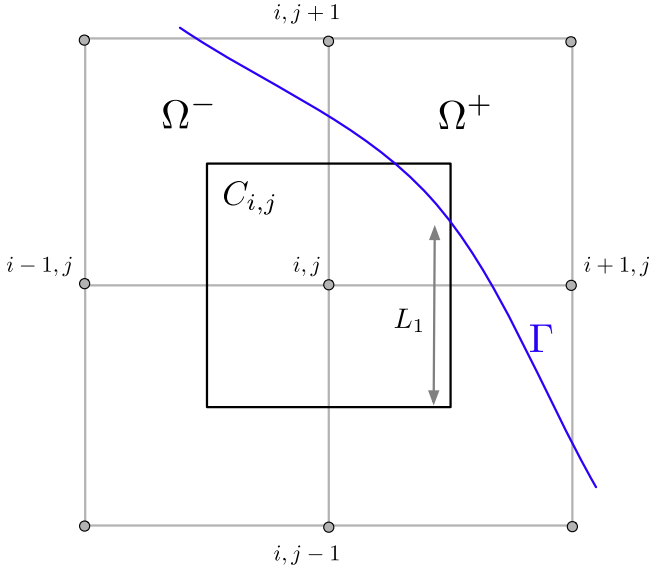

Fig. 1. Discretization near the interface in the two dimensional case.

3.1.2. Finite volume approach in three spatial dimensions

The three dimensional case is the extension of the idea presented above in two spatial dimensions. Eq. (5) is written component-wise as:

$$-\int_{\partial C_{ijk}\cap\Omega^-}\left[\left((\lambda+2\mu)\frac{\partial u}{\partial x}+\lambda\frac{\partial v}{\partial y}+\lambda\frac{\partial w}{\partial z}\right)N_1+\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)N_2+\mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)N_3\right]ds=\int_{C_{ijk}\cap\Omega^-}f_1\,dx+\int_{C_{ijk}\cap\partial\Omega^-}g_1,$$

$$-\int_{\partial C_{ijk}\cap\Omega^-}\left[\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)N_1+\left((\lambda+2\mu)\frac{\partial v}{\partial y}+\lambda\frac{\partial u}{\partial x}+\lambda\frac{\partial w}{\partial z}\right)N_2+\mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)N_3\right]ds=\int_{C_{ijk}\cap\Omega^-}f_2\,dx+\int_{C_{ijk}\cap\partial\Omega^-}g_2$$

and

$$-\int_{\partial C_{ijk}\cap\Omega^-}\left[\mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)N_1+\mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)N_2+\left((\lambda+2\mu)\frac{\partial w}{\partial z}+\lambda\frac{\partial u}{\partial x}+\lambda\frac{\partial v}{\partial y}\right)N_3\right]ds=\int_{C_{ijk}\cap\Omega^-}f_3\,dx+\int_{C_{ijk}\cap\partial\Omega^-}g_3.$$

Referring to Fig. 2, F_1 is the face at $i+\frac{1}{2}$ (corresponding to $N_1=1, N_2=0$ and $N_3=0$). The boundary integrals on that face is computed using the following approximations:

$$\int_{F_1}\frac{\partial u}{\partial x}ds=S\left(\frac{u_{i+1,j,k}-u_{i,j,k}}{h}\right),$$

$$\int_{F_1}\frac{\partial u}{\partial y}ds=\frac{S_R}{4}\left(\frac{u_{i,j+1,k}-u_{i,j,k}}{h}+\frac{u_{i+1,j+1,k}-u_{i+1,j,k}}{h}\right)+\frac{S_L}{4}\left(\frac{u_{i,j,k}-u_{i,j-1,k}}{h}+\frac{u_{i+1,j,k}-u_{i+1,j-1,k}}{h}\right)$$

and

$$\int_{F_1}\frac{\partial u}{\partial z}ds=\frac{S_T}{4}\left(\frac{u_{i,j,k+1}-u_{i,j,k}}{h}+\frac{u_{i+1,j,k}-u_{i+1,j,k+1}}{h}\right)+\frac{S_B}{4}\left(\frac{u_{i,j,k}-u_{i,j,k-1}}{h}+\frac{u_{i+1,j,k}-u_{i+1,j,k-1}}{h}\right),$$

where S is the area of F_1 inside Ω. The areas S_R, S_L, S_T and S_B are defined in Fig. 2. In the case where either L_L or L_R is equal to zero then S_R and S_L are defined by:

$$\begin{cases} S_L = S & S_R = 0 \quad \text{if} \quad L_R = 0, \\ S_R = S & S_L = 0 \quad \text{if} \quad L_L = 0. \end{cases}$$

Likewise, if either L_T or L_B is equal to zero then S_T, S_B are defined by:

$$\begin{cases} S_T = S & S_B = 0 \quad \text{if} \quad L_B = 0, \\ S_B = S & S_T = 0 \quad \text{if} \quad L_T = 0. \end{cases}$$

Similarly to the two dimensional case, the final discretizations are obtained by summing the contributions on all the faces.

Remarks:

- Given the finite volume approximation, the resulting linear system is symmetric.
- The discretizations of the cross-derivatives terms lead to some potential positive extra-diagonal elements, and therefore the linear system cannot be proved to be positive definite.

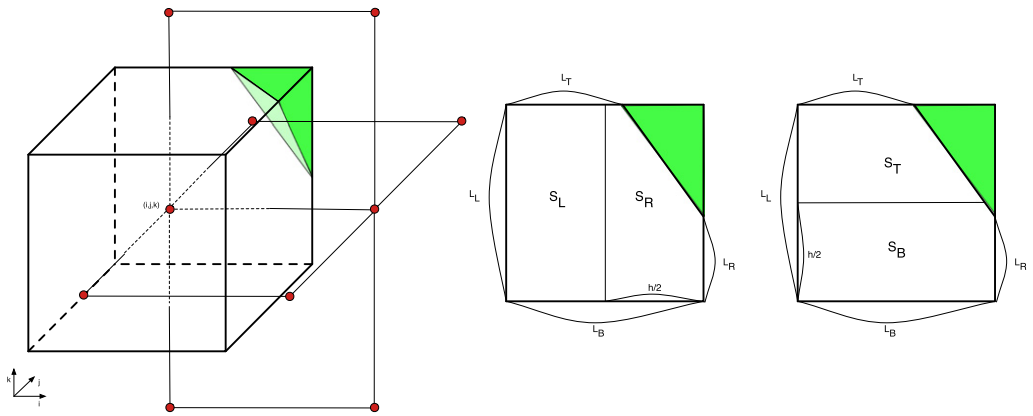

Fig. 2. Left: grid configuration near the interface for a cell C_{ijk}. The outside domain Ω^+ corresponding to $\phi>0$ is depicted in green. Right: right face of cell C_{ijk}. The lengths use in the discretizations are denoted by L_T, L_R, L_B, L_L and the areas denoted by S_R, S_L, S_T, S_B. The length of the cell face is denoted by h. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- The discretizations do not rely on the ersatz material assumption and thus qualifies as a 'sharp' method.

3.2. Extension to quadtree/octree adaptive grids

The differential form of the equations in three spatial dimensions is given by:

$$-(2\mu+\lambda)\frac{\partial^2 u}{\partial x^2}-\mu\frac{\partial^2 u}{\partial y^2}-\mu\frac{\partial^2 u}{\partial z^2}-(\lambda+\mu)\left[\frac{\partial^2 v}{\partial x\partial y}+\frac{\partial^2 w}{\partial x\partial z}\right]=f_1, \tag{11}$$

$$-(2\mu+\lambda)\frac{\partial^2 v}{\partial y^2}-\mu\frac{\partial^2 v}{\partial x^2}-\mu\frac{\partial^2 v}{\partial z^2}-(\lambda+\mu)\left[\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 w}{\partial y\partial z}\right]=f_2 \tag{12}$$

and

$$-(2\mu+\lambda)\frac{\partial^2 w}{\partial z^2}-\mu\frac{\partial^2 w}{\partial x^2}-\mu\frac{\partial^2 w}{\partial y^2}-(\lambda+\mu)\left[\frac{\partial^2 u}{\partial x\partial z}+\frac{\partial^2 v}{\partial y\partial z}\right]=f_3. \tag{13}$$

These equations can be solved by using standard finite difference schemes on quadtree/octree data structures using the discretization operators introduced in [22,5]. The major drawback is that typical finite differences schemes are intricate in handling Neumann-type boundary conditions on irregular domains, especially in the case of non-uniform grids. However, the meshing strategy of an elliptic problem suggests that the smallest cells should be placed on (and near) the interface. In turn, we can easily apply the finite volume approach near the interface since the cells are locally uniform and resort to the finite difference approximations elsewhere. This hybrid approach combines the effectiveness in solving the elasticity equations on quadtree/octree grids and the natural framework offered by finite volume discretization to impose Neumann-type boundary conditions. The solution strategy developed to solve the equations of linear elasticity on adaptive grids is presented as follows:

1. Generate an adaptive mesh that is refined with a desired band of uniform cells around the interface. In the case of an adaptive mesh, it is trivial to impose a small band of uniform cells around the interface, and we found that for our simulations it is enough to impose a band of four cells around the interface.
2. In the uniform mesh region, i.e. near the interface, use the finite volume approach as described in Section 3.1.
3. For the points that are inside the domain but far from the interface, use the standard finite difference approximations in quadtree/octree. We make use of the framework introduced by Min and Gibou [22,5] to discretize the system of equations. This framework and the discretizations are recalled in the following section.

We note that the resulting linear system is not symmetric in the case of quadtree/octree discretizations because of the approximation of the different operators.

3.2.1. Quadtree and octree grid structures

Quadtree and octree data structures, see e.g. [33,34], are efficient in storing adaptive grids in two and three spatial dimensions, respectively. Their construction starts from the root cell that represents the computational domain. Then recursively, the tree is constructed by adding children of the cell, corresponding to a split of the parent cell in equal parts (see Fig. 3). In the case of quadtree, four children are added, whereas in three spatial dimensions, eight children are added. The level of each cell is defined as the depth of the tree characterizing that cell. In the case of non-graded grids, the difference of level between two adjacent cells can be arbitrary, whereas graded grids limit the difference of levels between adjacent cell to one. Considering non-graded grids leads to versatile grid generation procedures, is applicable to more general refinement criteria and reduces the total number of nodes.

3.2.2. Domain description and grid generation

Although several refinement criteria could be defined, we are principally interested in automatically refining the mesh where the interface lies. Based on the work of Strain [37] and Min [19], Min and Gibou [22] proposed the following simple refinement criteria for generating such grids: starting from a root cell split any cell C if:

$$\min_{v\in\text{vertices}(C)}|\phi(v)|\leqslant \text{Lip}(\phi)\cdot\text{diag-size}(C), \tag{14}$$

where diag-size (C) refers to the length of the diagonal of the current cell C and v refers to a vertex (node) of the current cell. Typical results are depicted in Fig. 4. We also note that it is straightforward to generate a grid with a uniform band around the interface.

3.2.3. Discretization of standard operators

The main difficulty in discretizing general differential operators on non-graded grids is due to the so-called T-junctions, i.e. nodes for which at least one of its neighbors in the Cartesian directions is missing. In this case, one must appropriately

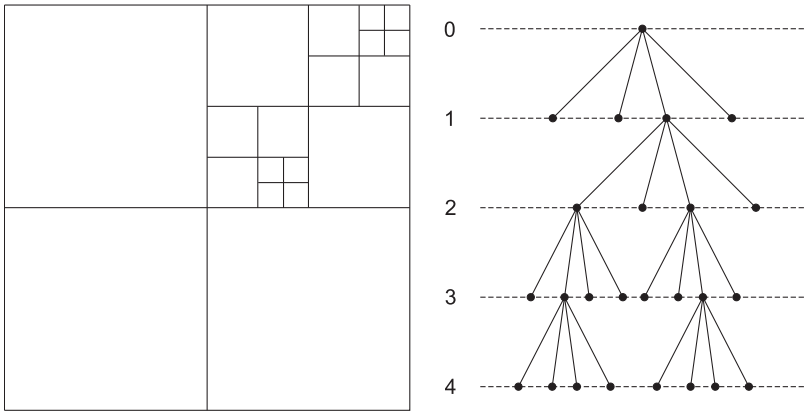

Fig. 3. Non-graded adaptive discretization of a two-dimensional domain (left) and its corresponding quadtree (right).

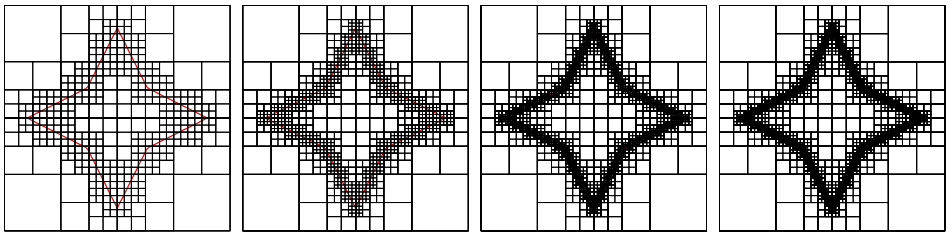

Fig. 4. Example of non-graded grids generated by using (14). The maximum level of refinement is increased from left to right.

define a ghost node that will be used in the discretizations. Min and Gibou [22] observed that for node-based quadtree/octree grids, it is always possible to define a third-order accurate ghost node in arbitrary Cartesian grid configuration. This definition then allows to literally port standard discretizations as if the grid was uniform. We also note that these definitions only uses the node values of the cells adjacent to the node at which one seeks to write the discretizations, which is an advantage of this approach. In what follows, we summarize the formulas from [22]:

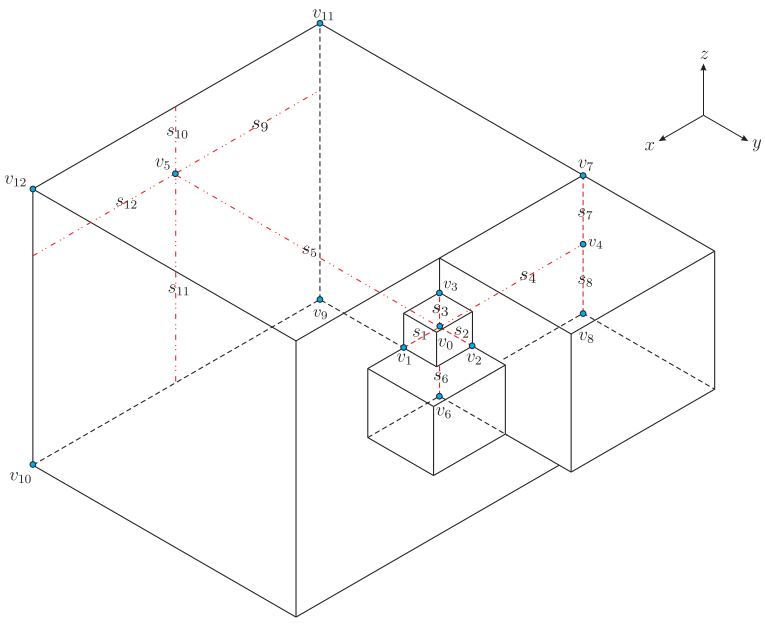

Fig. 5. General octree neighborhood configuration around a grid node v_0.

Referring to Fig. 5, which represents the general configuration of neighboring nodes in the case of an octree, a T-junction node v_0 has four regular neighboring nodes v_1, v_2, v_3 and v_6 and two ghost nodes v_4 and v_5. The third-order accurate interpolations of a node-sampled function $\phi : \{v_i\} \to \mathbb{R}$ at the ghost nodes v_4 and v_5 are defined as:

$$\phi_4^G = \frac{s_7\phi_8 + s_8\phi_7}{s_7 + s_8} - \frac{s_7 s_8}{s_3 + s_6}\left(\frac{\phi_3 - \phi_0}{s_3} + \frac{\phi_6 - \phi_0}{s_6}\right)$$

and

$$\phi_5^G = \frac{s_{11}s_{12}\phi_{11} + s_{11}s_9\phi_{12} + s_{10}s_{12}\phi_9 + s_{10}s_9\phi_{10}}{(s_{10} + s_{11})(s_9 + s_{12})} - \frac{s_{10}s_{11}}{s_3 + s_6}\left(\frac{\phi_3 - \phi_0}{s_3} + \frac{\phi_6 - \phi_0}{s_6}\right) - \frac{s_9 s_{12}}{s_1 + s_4}\left(\frac{\phi_1 - \phi_0}{s_1} + \frac{\phi_4^G - \phi_0}{s_4}\right), \tag{15}$$

where s_i refers to the distance between v_0 and v_i.

The definition of the ghost nodes allows to define finite differences for $\phi_x, \phi_y, \phi_z, \phi_{xx}, \phi_{yy}$ and ϕ_{zz} at every nodes using standard finite difference formulas in a dimension by dimension framework. For example, referring to Fig. 5, the following formulas define the standard discretization for the first and second-order derivatives:

$$\begin{aligned}
D_x^0\phi_0 &= \frac{\phi_1 - \phi_0}{s_1}\cdot\frac{s_4}{s_1 + s_4} + \frac{\phi_0 - \phi_4^G}{s_4}\cdot\frac{s_1}{s_1 + s_4},\\
D_y^0\phi_0 &= \frac{\phi_2 - \phi_0}{s_2}\cdot\frac{2}{s_2 + s_5} + \frac{\phi_0 - \phi_5^G}{s_5}\cdot\frac{s_2}{s_5 + s_2},\\
D_z^0\phi_0 &= \frac{\phi_6 - \phi_0}{s_6}\cdot\frac{2}{s_6 + s_3} + \frac{\phi_0 - \phi_3}{s_3}\cdot\frac{s_6}{s_6 + s_3},
\end{aligned} \tag{16}$$

$$\begin{aligned}
D_{xx}^0\phi_0 &= \frac{\phi_1 - \phi_0}{s_1}\cdot\frac{2}{s_1 + s_4} - \frac{\phi_0 - \phi_4^G}{s_4}\cdot\frac{2}{s_1 + s_4},\\
D_{yy}^0\phi_0 &= \frac{\phi_2 - \phi_0}{s_2}\cdot\frac{2}{s_2 + s_5} - \frac{\phi_0 - \phi_5^G}{s_5}\cdot\frac{2}{s_5 + s_2},\\
D_{zz}^0\phi_0 &= \frac{\phi_6 - \phi_0}{s_6}\cdot\frac{2}{s_6 + s_3} - \frac{\phi_0 - \phi_3}{s_3}\cdot\frac{2}{s_6 + s_3}.
\end{aligned}$$

Cross derivatives can be obtained using these formulas. For example to calculate $D_{xy}^0\phi_0$, we first compute $\psi = D_y^0\phi$ at every node surrounding node v_0 using Eq. (16) before using once again Eq. (16) to compute $D_x^0\psi_0 = D_{xy}^0\phi_0$.

For nodes where the finite difference approximations are used, each row of the linear system is filled using the coefficients of the approximations of the different operators in Eqs. (11)–(13).

4. Numerical examples

In this section, we provide numerical examples in two and three spatial dimensions that indicate that our approach is second-order accurate in both the L^1 and L^∞ norms. In order to compute the order of accuracy of the method, we first consider a non-graded adaptive quadtree grid, on which we calculate the numerical solution and its error to the analytical solution. Then we recursively split every cell of the grid, increasing each time res_{min} and res_{max} by a factor of two, and computing the new solution. The parameters res_{min} and res_{max} refer to the minimum and maximum resolutions (minimum and maximum number of nodes in one direction), respectively. We use a BiCGSTAB solver (with an error tolerance of 10^{-12}) to invert the linear system.

4.1. Numerical examples in two spatial dimensions

4.1.1. Mixed boundary conditions on a star geometry

We considered Ω^- to be defined by the zero contour of the level-set function $\Phi(r,\theta) = \max(r - 0.75(1 - 0.2\sin(9\theta)), -r + 0.25(1 - 0.1\sin(13\theta)))$, where r and θ are the polar coordinates. This domain is represented in Fig. 6. Neumann-type boundary conditions are assumed on the outer contour, while Dirichlet boundary conditions are assumed on the inner contour so that the problem is numerically well-posed. We take the exact solution $u = \cos(2\pi x)\cos(2\pi y)$, $v = \sin(2\pi x)\sin(2\pi y)$, and the Lamé parameters to be $\lambda = \mu = 1$. The convergence results are shown in Table 1. The numerical error on the finest grid is depicted in Fig. 6: as expected the error is maximum at the interface where Neumann-type boundary conditions are imposed, while it is minimum on the Dirichlet boundary.

4.1.2. Influence of the Lamé parameters

Here we consider a domain Ω^- to be an ellipse given by the equation $4x^2 + y^2 = 0.7^2$ with an elliptic hole given par the equation $x^2 + 4y^2 = 0.2^2$. We take for the exact solution $u(x,y) = e^{-xy}$ and $v(x,y) = e^{-2xy}$. We impose a Dirichlet boundary condition on the half domain $x > 0$ and Neumann-type on the other half. In order to study the influence of the Lamé coefficients on the convergence of our method, we consider two different cases where the ratio between those two parameters is

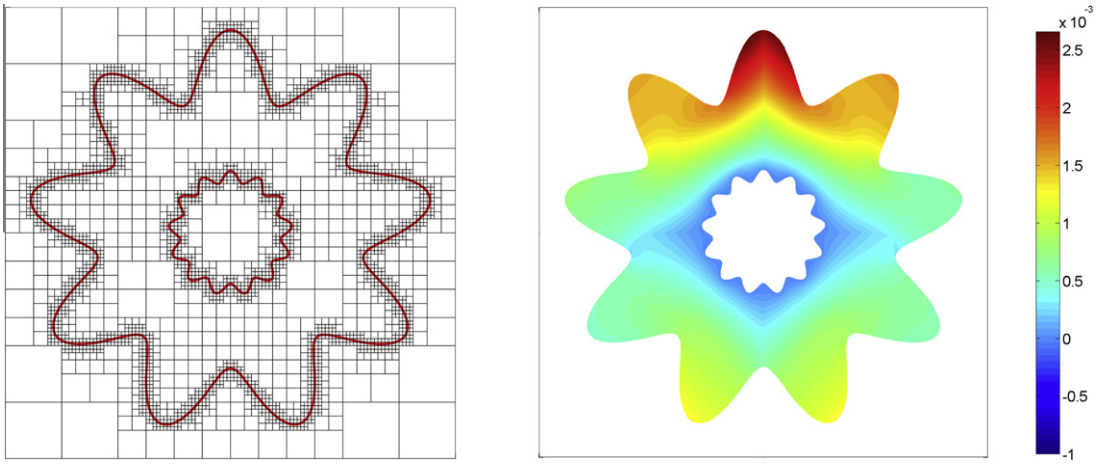

Fig. 6. Left: domain considered in Example 4.1.1. The depicted quadtree grid is the coarsest grid used in the convergence analysis. Right: numerical error on the finest grid.

large: $\lambda = 100,\ \mu = 1$ and $\lambda = 1,\ \mu = 100$. Convergence results are provided in Tables 2 and 3. Note that the numerical error is smaller in the second case but that both set of parameters lead to second-order accurate results.

4.1.3. Punctured infinite plate

This is a classical two dimensional Lamé problem that has many applications in fracture mechanics. In linear elasticity, plane problems are those in which one dimension is either very small or very large compared to the other two dimensions. If the thickness is very small, the two dimensional problem corresponds to a plane stress problem, i.e. a problem where there is no stress along the third direction in the stress tensor; if the thickness is very large, it is a plane strain problem. The example

Table 1
Convergence analysis for Example 4.1.1.

res_{min}/res_{max}	Grid points	L^1 Error	Order	L^∞ Error	Order
8/64	1942	8.776×10^{-2}	-	1.303×10^{0}	-
16/128	7239	2.173×10^{-2}	2.014	2.682×10^{-1}	2.281
32/256	27,901	6.154×10^{-3}	1.820	7.465×10^{-2}	1.845
64/512	109,497	1.627×10^{-3}	1.919	1.949×10^{-2}	1.937
128/1024	433,777	4.179×10^{-4}	1.961	4.975×10^{-3}	1.970

Table 2
Accuracy for Example 4.1.2 with $\lambda = 100$ and $\mu = 1$.

res_{min}/res_{max}	Grid points	L^1 Error	Order	L^∞ Error	Order
8/64	945	1.060×10^{-1}	-	1.533×10^{0}	-
16/128	3505	1.397×10^{-3}	6.245	1.513×10^{-2}	6.663
32/256	13,473	3.509×10^{-4}	1.993	4.366×10^{-3}	1.793
64/512	52,801	9.828×10^{-5}	1.836	1.148×10^{-3}	1.927
128/1024	209,025	2.567×10^{-5}	1.937	2.936×10^{-4}	1.967

Table 3
Accuracy for Example 4.1.2 with $\lambda = 1$ and $\mu = 100$.

res_{min}/res_{max}	Grid points	L^1 Error	Order	L^∞ Error	Order
8/64	945	3.901×10^{-4}	-	6.277×10^{-3}	-
16/128	3505	4.701×10^{-5}	3.053	8.134×10^{-4}	2.948
32/256	13,473	1.234×10^{-5}	1.929	2.379×10^{-4}	1.774
64/512	52,801	3.298×10^{-6}	1.903	6.228×10^{-5}	1.933
128/1024	209,025	7.641×10^{-7}	2.109	1.564×10^{-5}	1.993

considered in this section is a plate containing a circular hole in its center. The dimensions of the plate are supposed to be considerably larger than the radius of the hole, so that the plate can be assumed infinite. The thickness is taken very small in order to satisfy the plane stress assumption. The plate is subjected to a uniform traction and the stress distribution around the hole is investigated. This situation describes a typical mode I crack initiation and the hoop stress distribution at the hole allows to determine the stress intensity factor of the crack, which reflects the fracture toughness of a given material.

We consider a domain $\Omega = [-1,1]^2$ and a subdomain Ω^- which defines a square with side L with a center hole of radius R. The geometry and the notations are depicted in Fig. 7. We apply a uniform stress on a part of the boundary $\Gamma_N = \Gamma_1 \cup \Gamma_2$ and a stress free condition on the other parts of the boundary, i.e. $\Gamma = \Gamma_3 \cup \Gamma_4 \cup \Gamma_5$ is stress free. We assume that $L \gg R$ so that we can consider that the plate is infinite. Then the boundary conditions are as follows:

$$\sigma_{rr} = \sigma_{r\theta} = 0 \quad \text{at} \quad \mathrm{r} = \mathrm{R}, \tag{17}$$

$$\sigma_{rr} = \frac{\sigma_0}{2}(1+\cos(2\theta)), \quad \sigma_{\theta\theta} = \frac{\sigma_0}{2}(1-\cos(2\theta)), \quad \text{and} \quad \sigma_{r\theta} = -\frac{\sigma_0}{2}\sin(2\theta) \quad \text{at} \quad r = \infty. \tag{18}$$

The geometry and the loading condition being axisymmetric, this case depicts a perfect Lamé problem for which the solution is obtained via the Airy Stress potential approach. It is well known that the appropriate Airy stress function has the following form in polar coordinates:

$$\phi(r,\theta) = \left(Ar^2 + Br^4 + \frac{C}{r^2} + D\right)\cos(2\theta).$$

The stress components are given by:

$$\sigma_{rr} = \frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2} + \frac{1}{r}\frac{\partial\phi}{\partial r}, \quad \sigma_{\theta\theta} = \frac{\partial^2\phi}{\partial r^2}, \quad \sigma_{r\theta} = -\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\phi}{\partial\theta}\right),$$

which combined with the boundary conditions (17 and 18) allow the constants A, B, C and D to be identified. The stress field solution is given by:

$$\begin{aligned}
\sigma_{rr} &= \frac{\sigma_0}{2}\left(1-\frac{R^2}{r^2}\right) + \frac{\sigma_0}{2}\left(1+3\frac{R^4}{r^4}-4\frac{R^2}{r^2}\right)\cos(2\theta),\\
\sigma_{\theta\theta} &= \frac{\sigma_0}{2}\left(1+\frac{R^2}{r^2}\right) - \frac{\sigma_0}{2}\left(1+3\frac{R^4}{r^4}\right)\cos(2\theta),\\
\sigma_{r\theta} &= -\frac{\sigma_0}{2}\left(1-3\frac{R^4}{r^4}+2\frac{R^2}{r^2}\right)\sin(2\theta).
\end{aligned} \tag{19}$$

Note that when $r = R$, $\sigma_{rr} = \sigma_{r\theta} = 0$ and $\sigma_{\theta\theta} = \sigma_0(1-2\cos(2\theta))$. Then the stress is maximum at $\theta = \frac{\pi}{2}$ and is equal to $\sigma_{\theta\theta} = 3\sigma_0$. The minimum is achieved at $\theta = 0$, corresponding to a value $\sigma_{\theta\theta} = -\sigma_0$. As r$\to\infty$, i.e. far from the hole, the stress field is an uniaxial tension where in Cartesian coordinates, $\sigma_{xx} = \sigma_0$, $\sigma_{xy} = \sigma_{yy} = 0$.

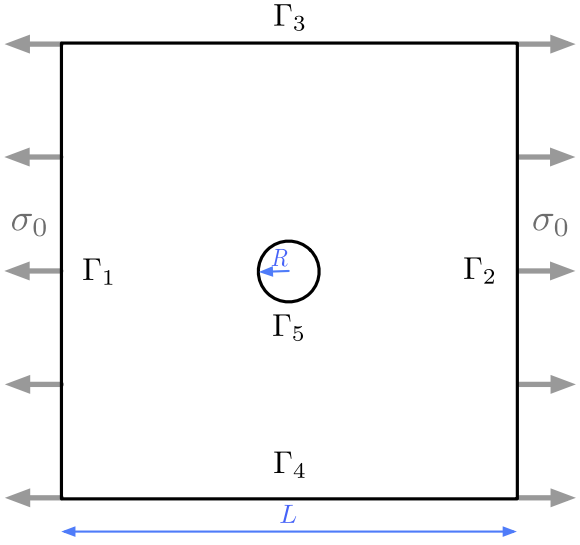

Fig. 7. Geometry of the problem considered in Example 4.1.3.

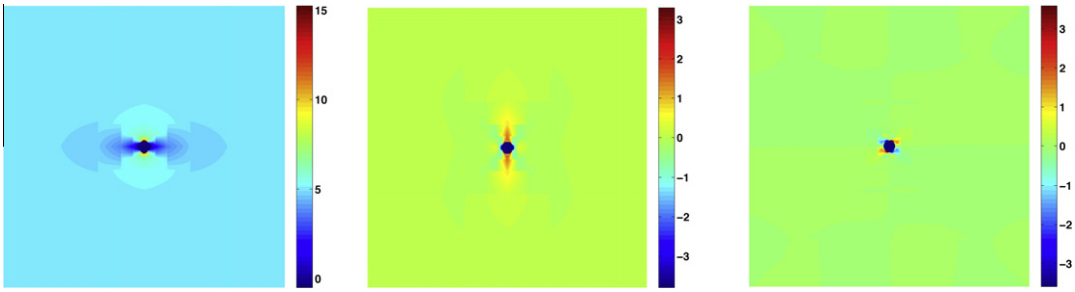

Fig. 8. Numerical stress tensor. σ_{xx} (left), σ_{yy} (center) and σ_{xy} (right).

Since only Neumann-type boundary conditions are imposed, there is no uniqueness of solution for the displacement $\mathbf{u} = (u, v)$. In order to obtain well-posedness, the symmetry of the loading and the geometry is taken into account, which translates into a symmetry of the response. In particular, it is assumed that nodes close to the interface that are on the center vertical line can only be subjected to a vertical motion, i.e. $u(x = 0, y) = 0$. Similarly, the nodes close to the interface that are on the center horizontal line are assumed to move only horizontally, i.e. we set $v(x, y = 0) = 0$.

The study of the stress field inside the plate is performed in three steps: first, the solution $\mathbf{u}$ is obtained by solving the linear elasticity equations using the quadtree framework described above in this paper. The resulting solution is valid inside the domain but in order to compute the stress the solution gradient is needed, including near the interface. Since the use of finite differences to calculate the gradient requires a proper extrapolation of the solution outside the domain, we use the extrapolation procedure of [3], extended to the case of quadtree/octree in [22]. Once the solution is extended accurately, the gradient can be calculated, from which the stress tensor is found using Hooke's law. We set the different parameters as: $R = 0.03125,\ L = 0.8,\ \sigma_0 = 5,\ \lambda = 5,\ \mu = 10$ and use the quadtree grid depicted in Fig. 9. Our approach is validated by comparing the numerical results to the analytical solution.

It is clear that given the finiteness of the domain, the exact solution is altered from the ideal case. Therefore, it is important to note that in this example, we are interested in validating the model of infinite plate from an engineering point of view, rather that performing a convergence analysis of the numerical schemes. In what follows, we show that our approach allows to recover the fundamental results related to an infinite plate with a hole in the case where the ratio R/L is sufficiently small. We start by seeking numerically a hole radius for which the infinite plate approximation is valid. Once such a value had been identified, we begin our analysis by examining the components of the stress tensor in Cartesian coordinates to illustrate the uniaxial stress behavior far from the hole. Then we analyze the behavior of the hoop stress on the hole and look for the maximum and minimum values along with their directions, in order to get the stress intensity factor. We then consider two specific directions, $\theta = 0$ and $\theta = \pi/2$, and compare the evolution of the stress components σ_{rr} and $\sigma_{\theta\theta}$ as function of the distance r to the analytical solution.

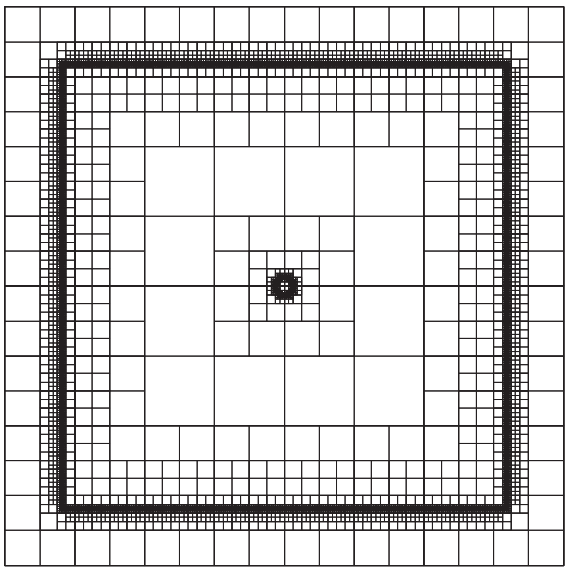

Fig. 9. Level 9 quadtree grid used in Example 4.1.3.

As the ration R/L decreases, it is important to verify that we approach the infinite plate model. To do so, we study the error between the maximum and minimum hoop stresses at the hole. As these errors decrease, the stress intensity factor gets closer to 3, which validates the model: starting with a plate having a hole of radius $R = 0.125$, on a level 7 quadtree grid ($res_{max} = 128$), we recursively reproduce the calculations for a hole twice as small. In order to keep the resolution of the grid consistent with the size of the hole we also double the maximum resolution of the grid. For each radius, the maximum and minimum values of the hoop stress at the hole are computed and so are the errors. The results obtained are illustrated in Fig. 11. It clearly appears that as the radius diminishes, these errors are considerably damped as expected, up to a certain radius of typically $R = 1/64$ from which they stay constant. From this stage on, the infinite approximation is valid and refining the grid is the only way to damp the numerical error. In what follows the radius of the hole is set to $1/32$.

In Fig. 8, the components of the stress tensor in Cartesian coordinates are represented. As expected all the components of the stress tensor are homogeneous far from the hole, but only σ_{xx} is not zero, corresponding to a state of uniaxial traction in the x-direction. The numerical value of the 'infinite' stress is set to σ_0 by the Dirichlet boundary condition.

Of particular interest is the hoop stress distribution at the surface of the hole, depicted in Fig. 10. The maximum and minimum hoop stresses are obtained at $\theta = \frac{\pi}{2}$ and $\theta = 0$, respectively, and the corresponding numerical values are 14.86 and -4.92. These agree well with the theoretical values of 15 and -5, respectively. The stress concentration factor is given by $K = \max(\sigma_{\theta\theta}(r = R))/\sigma_0$, and is found to be K = 2.972, with a relative error of 9.33×10^{-3}. Again, this value is close to the theoretical one under the 'infinite' plate assumption.

The stress tensor components σ_{rr} and $\sigma_{r\theta}$ in the directions $\theta = 0$ and $\theta = \pi/2$ are compared to the analytical ones in Fig. 12. The numerical solution is close to the analytical one and, as expected, the error is bigger close to the hole.

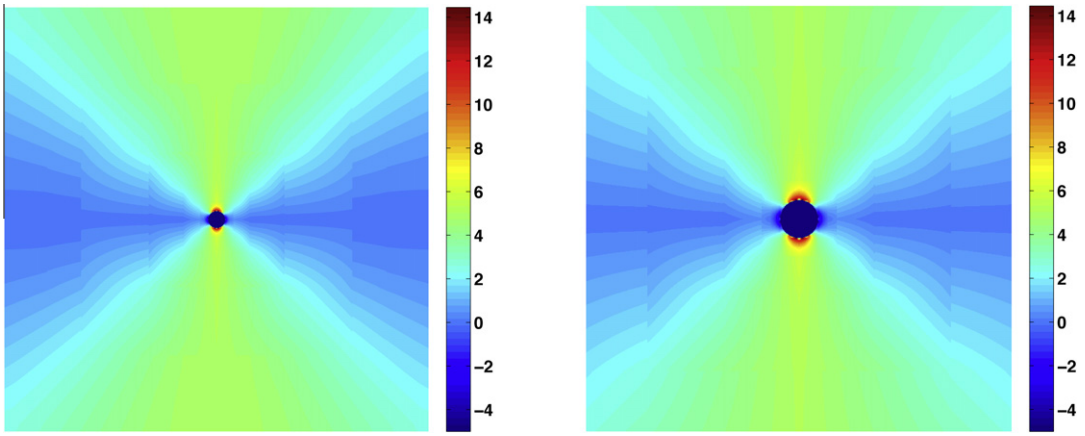

Fig. 10. The $\sigma_{\theta\theta}$-component of the stress tensor of the numerical solution (left). Zoom on the area near the hole (right).

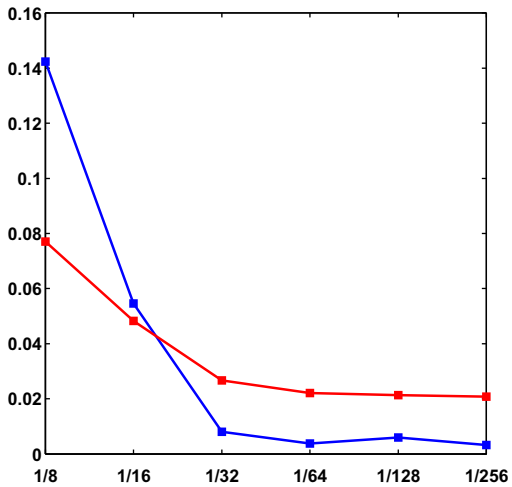

Fig. 11. Relative difference between the numerical maximum (red) and minimum (blue) values in $\sigma_{\theta\theta}$ and the maximum/minimum values predicted by the infinite plate model versus the size of the hole. To obtain those result we start with a plate of length $L = 0.8$ and a initial hole of radius $R = 1/8$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

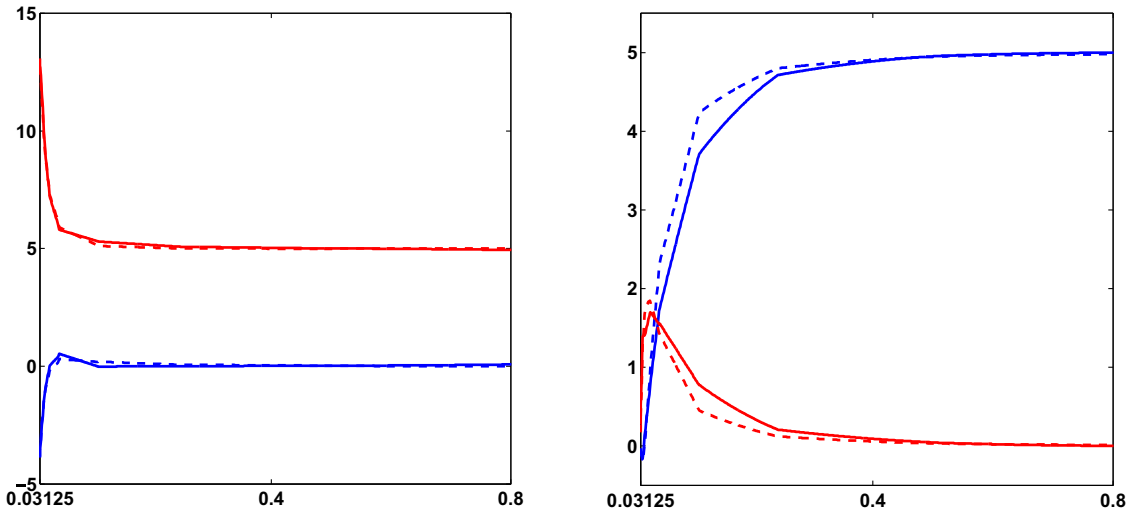

Fig. 12. Stress tensor components in polar coordinates (from left to right $\sigma_{\theta\theta}$ and σ_{rr}, in the x-(blue) and y-directions (red). The solid lines represent the numerical solution while the dash line represent the exact solution for an infinite plate. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1.4. Shape optimization

One advantage in considering an embedded boundary approach for solving the linear elasticity equations is that the meshing procedure is straightforward and robust. In fact, meshing the domain for a given geometry is typically about 200 times faster than a linear system solve. To exemplify this capability, we apply our algorithm to the case of shape optimization. In particular, we follow the approach of Allaire et al. to shape optimization [1,11] except that we use our framework to solve the elasticity equation instead of considering the erzatz materials and the finite element method. Here we first briefly introduce the shape optimization framework and we refer the interested reader to [1,11] for more details. This method is then illustrated with a numerical example.

Given the governing equations of Section 2, the aim of the shape optimization in elasticity is to minimize the compliance, defined as the work done by external forces on Ω^- under a fixed volume constraint. Mathematically, given a fixed volume we seek to maximize the stiffness/rigidity of the object under consideration:

$$J(\Omega^-) = \int_{\Omega^-} \underline{\underline{\sigma}} : \underline{\underline{e}}(\mathbf{u}) = \int_{\Gamma_N} \underline{g} \cdot \mathbf{u}\, dx + \int_{\Omega} \underline{f} \cdot \mathbf{u}.$$

The shape derivative of this functional is given by:

$$J'(\Omega^-)(\boldsymbol{\theta}) = -\int_{\Gamma} \boldsymbol{\theta} \cdot \mathbf{n} \left(2\mu \left| \underline{\underline{e}}(\mathbf{u}) \right|^2 + \lambda \left(Tr\, \underline{\underline{e}}(\mathbf{u}) \right)^2 \right) dx,$$

while the volume constraint is expressed as:

$$V'(\Omega^-)(\boldsymbol{\theta}) = \int_{\Gamma} \boldsymbol{\theta} \cdot \mathbf{n}\, ds.$$

The optimality condition for the domain Ω^- is that there exists a Lagrange multiplier $l \in \mathbb{R}$ such that:

$$J'(\Omega^-)(\boldsymbol{\theta}) + lV'(\Omega^-)(\boldsymbol{\theta}) = 0,$$

hence the equation:

$$\int_{\Gamma} \boldsymbol{\theta} \cdot \mathbf{n} \left(l - 2\mu |\underline{\underline{e}}(\mathbf{u})|^2 - \lambda \left(Tr(\underline{\underline{e}}(\mathbf{u})) \right)^2 \right) ds = 0.$$

To choose a convenient descent direction we use a gradient algorithm with projection [1,11]. Then we take the following expression for $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \left(2\mu |\underline{\underline{e}}(\mathbf{u})|^2 + \lambda \left(Tr(\underline{\underline{e}}(\mathbf{u})) \right)^2 - l \right) \mathbf{n}. \tag{20}$$

In addition, since we need to impose that the volume be constant, the condition $V'(\Omega^-)(\boldsymbol{\theta}) = 0$ must be satisfied, and therefore:

$$\int_\Gamma \left(2\mu|\underline{\underline{e}}(\mathbf{u})|^2 + \lambda\Big(Tr(\underline{\underline{e}}(\mathbf{u}))\Big)^2\right) ds = \int_\Gamma l\, ds = l|\Gamma|',$$

thus:

$$l = \frac{\int_\Gamma \left(2\mu|\underline{\underline{e}}(\mathbf{u})|^2 + \lambda\Big(Tr(\underline{\underline{e}}(\mathbf{u}))\Big)^2\right) ds}{|\Gamma|}.$$

The descent direction $\boldsymbol{\theta}$ corresponds to the normal velocity according to which the domain Ω^- must be advected so that the compliance decreases and the volume remains constant. Since the domain Ω^- is represented by a level-set function ϕ, evolving Ω^- according to the descent direction velocity is equivalent to solving the following Hamilton–Jacobi equation for ϕ:

$$\frac{\partial\phi}{\partial t} + \boldsymbol{\theta}|\nabla\phi| = 0. \tag{21}$$

Even though the Lagrange multiplier is chosen such that the volume remains constant, numerical artifacts can generate small variations in the volume. In order to ensure that the volume is conserved through the optimization process, we compare for each iteration the volume of the updated shape to the volume of the initial shape. If this difference is larger than a given tolerance we evolve the current contour in the normal direction with the following velocity:

$$\boldsymbol{\theta}'\Delta t = \frac{V_{\text{current}} - V_{\text{initial}}}{|\Gamma|}, \tag{22}$$

where V_{initial} is the volume of the initial shape, and V_{current} is the volume of the evolving shape. This process is repeated until the difference between the two volumes is less than the given tolerance (in this work the tolerance is set to be 10^{-8}). The iterative algorithm we use is structured as follows:

1. Initialization of all the quantities: Ω^0, u^0, μ and λ.
2. Iterate until convergence:
   (a) Compute $\mathbf{u}^k$ the solution of the elasticity problem on Ω^k,
   (b) Compute $\boldsymbol{\theta}^k$ from $\mathbf{u}^k$,
   (c) Evolve ϕ^k to ϕ^{k+1} according to the velocity $\boldsymbol{\theta}^k$ using (21),
   (d) Ensure that the volume is conserved using (22),
   (e) Reinitialize ϕ^{k+1},
   (f) Adapt the grid to the new zero level-set ϕ^{k+1}.

Note: The level-set function is evolved and reinitialized using the technology presented in [22].

The numerical example we present here is that first proposed in [1]. The initial geometric configuration Ω^- is represented in Fig. 13. Its boundary $\partial\Omega^-$ is split into three different parts:

$$\partial\Omega^- = \Gamma \cup \Gamma_D \cup \Gamma_N, \tag{23}$$

where $\Gamma_D = \Gamma_4 \cup \Gamma_5$, $\Gamma_N = \Gamma_6$ and $\Gamma = \Gamma_1 \cup \Gamma_3$ (cf Fig. 13). $\Gamma \neq \varnothing$ is the part of the boundary that we want to optimize, and thus that will be modified through the optimization process, while Γ_D and Γ_N are fixed. On Γ_D we apply Dirichlet boundary conditions by setting the displacement field to be null, and on Γ_N we apply a stress $\mathbf{g} = -p\mathbf{e}_y$, where $\mathbf{e}_y$ is the unit vector in the y-direction. The boundary Γ is stress-free. We take the following parameters: $a = 0.45$, $b = 0.40$, $c = 0.10$, $d = 0.20$, $\mu = 5.56$, $\lambda = 12.96$ and $p = 1$. We take for the time step:

$$\Delta t = 0.4\frac{\Delta x}{\theta_{\max}},$$

where $\theta_{\max}$ is the maximum interface's velocity.

The results are presented in Figs. 14 and 15. The compliance and the volume of the geometry through the shape optimization process are represented in Fig. 14. Clearly the compliance decreases with the number of iterations. The volume does not remain constant, but its variations are relatively small compared to its initial value. In Fig. 15, we illustrate the evolution of the shape at different iterations. Our results agree with the results of previous works [1,11].

Note: We have not optimized the gradient descent algorithm and have focused on only giving an illustrative example of the applicability of our linear elasticity solver.

4.2. Numerical examples in three spatial dimensions

4.2.1. Mixed boundary conditions

In this example we consider the domain Ω^- to be contained between two spheres of radii $R = 0.7$ and $R = 0.2$, as depicted in Fig. 16. The level-set function associated to this geometry is $\phi(r,\theta,\phi) = \max(r - 0.7, 0.2 - r)$, where r is the first polar

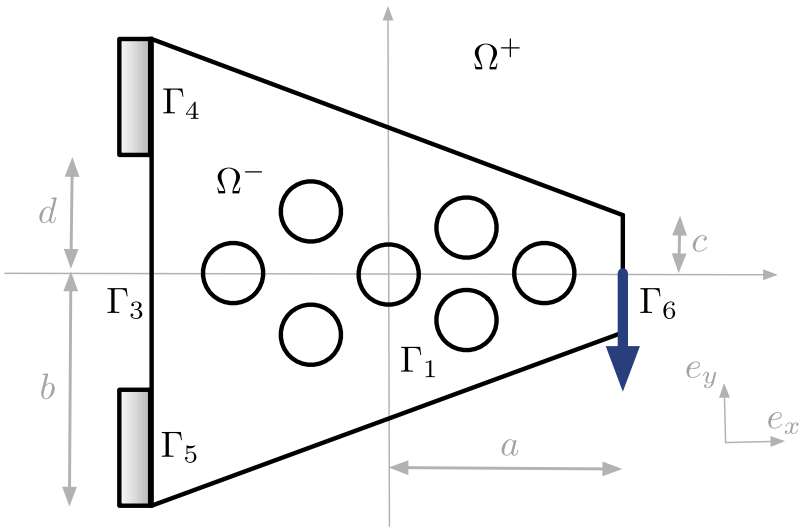

Fig. 13. Initial optimization geometry for the shape optimization Example 4.1.4.

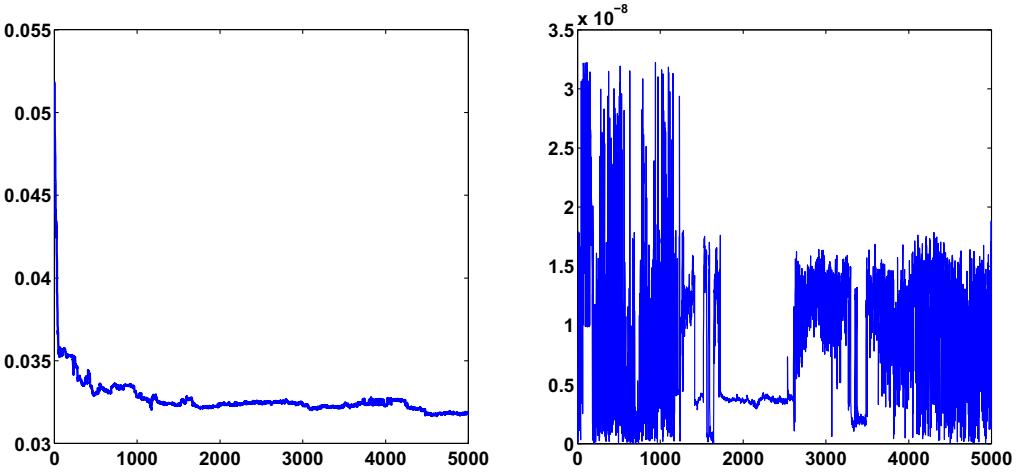

Fig. 14. Shape optimization: compliance (left) and relative variations of the volume (right) versus the number of iterations.

coordinate. The whole domain is $\Omega = [-1,1]^3$. We take for the exact solution $u(x,y,z) = v(x,y,z) = w(x,y,z) = \cos(x)\cos(y)\cos(z)$. Dirichlet boundary conditions are imposed on the inner sphere while Neumann-type boundary conditions are imposed on the outer sphere and the Lamé parameters are chosen to be $\lambda = 2$ and $\mu = 7$. Convergence results are given in Table 4 and suggest that the methods is second-order accurate in both L^1 and L^∞ norms.

4.2.2. Cylinder tank under pressure

This example is of practical importance in engineering, as it has applications in the design of pipes. We consider a cylinder tank subject to an internal pressure P, while its external, top and bottom faces are stress free. We recall R_i and R_e the internal and external radii, and H the height of the cylinder. We neglect body forces so that the stress tensor is solution of the equilibrium equation $\nabla \cdot \underline{\underline{\sigma}} = 0$, and the boundary conditions are:

$$\underline{\underline{\sigma}} \cdot \boldsymbol{e}_r = -P\boldsymbol{e}_r \quad \text{at} \quad r = R_i,$$
$$\underline{\underline{\sigma}} \cdot \boldsymbol{e}_r = 0 \quad \text{at} \quad r = R_e,$$
$$\underline{\underline{\sigma}} \cdot \boldsymbol{e}_z = 0 \quad \text{at} \quad z = \pm H.$$

The geometry of the problem is given by Fig. 17. Seeking a solution with a cylindrical symmetry, the analytical expression for the stress tensor is given in cylindrical coordinates:

$$\underline{\underline{\sigma}}(r,\theta,z) = \left(A - \frac{B}{r^2}\right)\boldsymbol{e}_r \otimes \boldsymbol{e}_r + \left(A + \frac{B}{r^2}\right)\boldsymbol{e}_\theta \otimes \boldsymbol{e}_\theta,$$

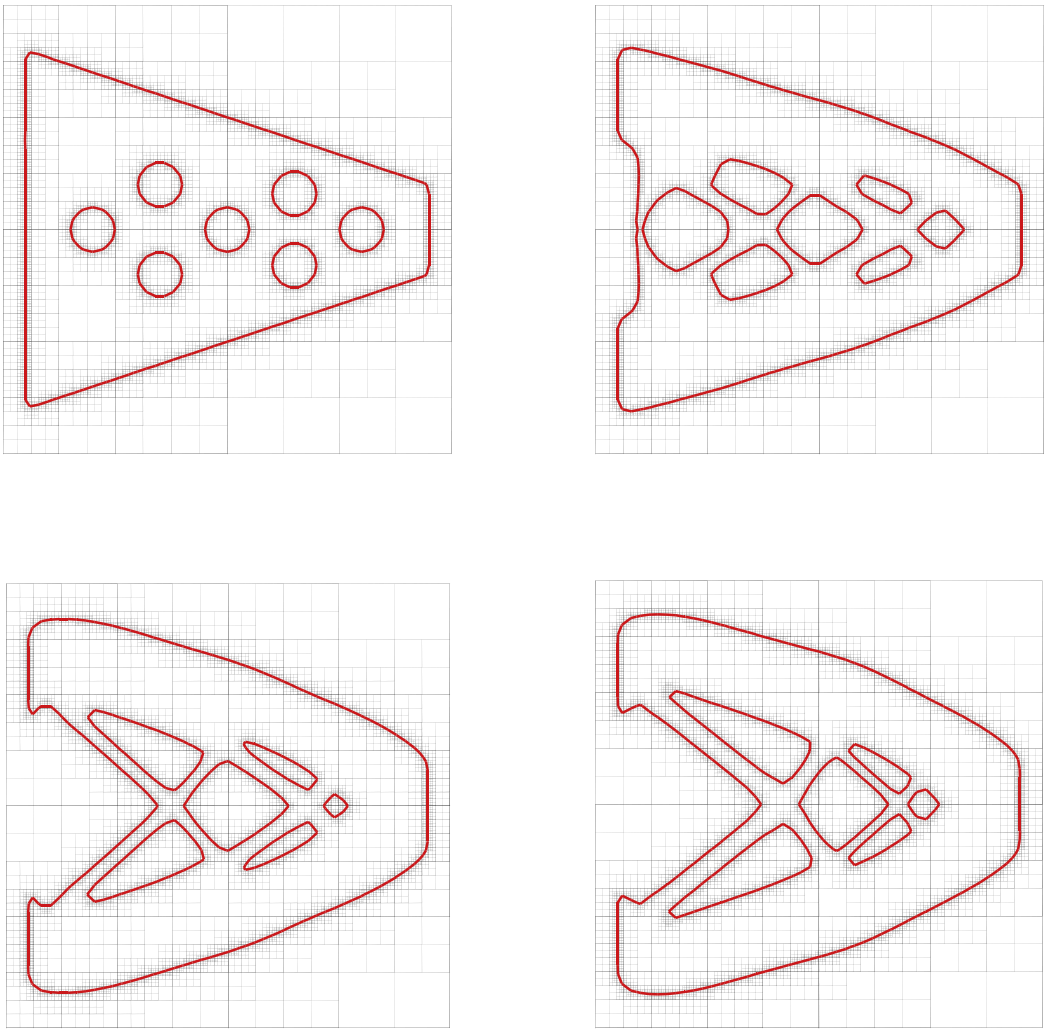

Fig. 15. Shape evolution for Example 4.1.4.

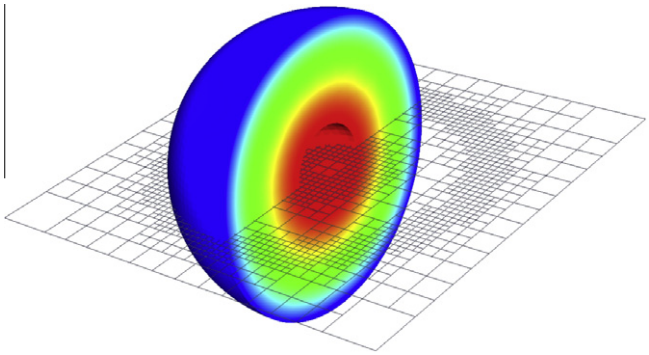

Fig. 16. Exact solution considered in Example 4.2.1. A slice of the coarsest octree used in the convergence analysis is also displayed. Neumann-type boundary conditions are imposed on the outer sphere while Dirichlet boundary conditions are imposed on the inner one.

where the coefficients A and B are defined by:

$$A = \frac{PR_i^2}{R_e^2 - R_i^2}, \quad B = \frac{PR_i^2 R_e^2}{R_e^2 - R_i^2}.$$

Table 4
Accuracy for Example 4.2.1.

res_{min}/res_{max}	Grid points	L^1 Error	Order	L^∞ Error	Order
16/64	62,203	2.175×10^{-3}	-	4.011×10^{-3}	-
32/128	454,189	6.769×10^{-4}	1.684	1.182×10^{-3}	1.760
64/256	3,460,153	1.918×10^{-4}	1.819	3.485×10^{-4}	1.766
128/512	26,988,529	3.643×10^{-5}	2.396	8.708×10^{-5}	2.002

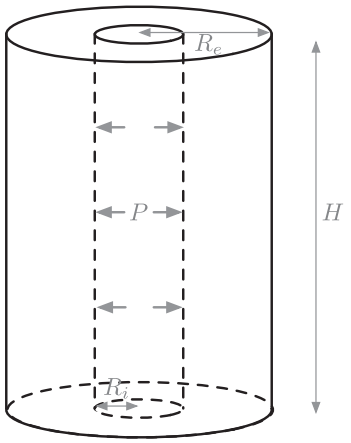

Fig. 17. Problem considered in 4.2.2.

Integrating the strain tensor obtained from the constitutive law, the displacement field is purely radial:

$$\mathbf{u}(r,\theta,z) = \left(\frac{A}{\lambda+\mu}r + \frac{B}{2\mu r}\right)\boldsymbol{e}_r.$$

This problem is numerically ill-posed, in the sense there is no uniqueness for the displacement field $\mathbf{u} = (u, v, w)$. As for Example 4.1.3, the well posedness is obtained by taking into account the symmetry of the problem: in particular it is assumed that at nodes close to the interface $u(x = 0, y, z) = 0$, $v(x, y = 0, z) = 0$, $w(x, y, z = 0) = 0$. In our simulations we consider that the cylinder is made out of steal ($\lambda = 115$ GPa, $\mu = 77$ GPa) and we take for the internal pressure $P = 100$ GPa. For geometric parameters we use the following values:

$$R_e = 0.9 \quad R_i = 0.3 \quad H = 3.9.$$

The calculations are performed on level 8 octree ($res_{max} = 256$). Fig. 18 shows the radial displacement and the components $\sigma_{\theta\theta}$ and σ_{rr} of the stress tensor. The effect of the grid refinement procedure on the relative error for the stress tensor is depicted in Fig. 19. As we can see the error is efficiently damped as the grid is refined close to the interface.

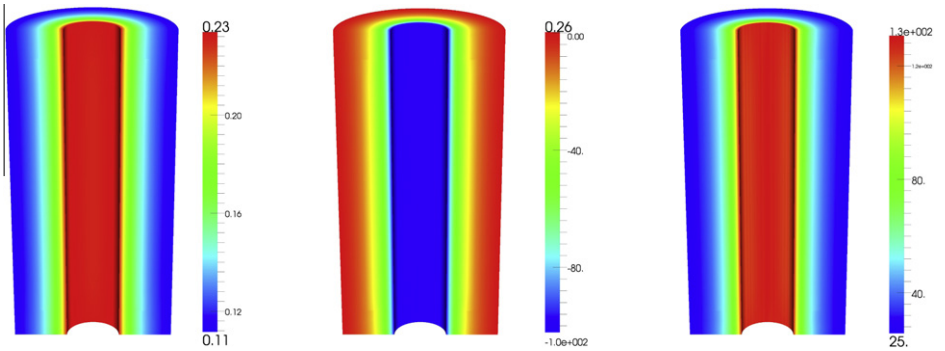

Fig. 18. Numerical results: radial displacement (left), σ_{rr} (center) and $\sigma_{\theta\theta}$ (right).

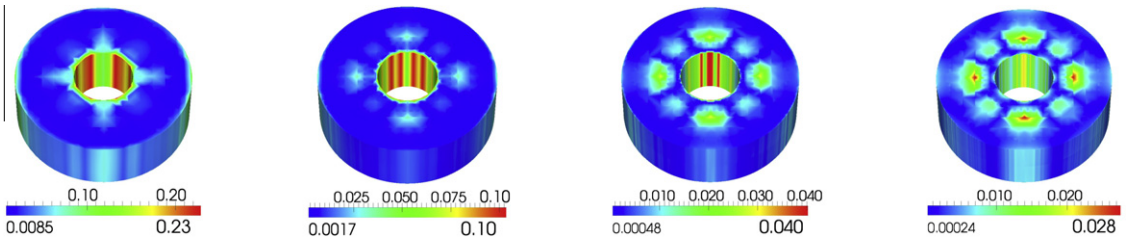

Fig. 19. Evolution of the L^1 relative error on the stress tensor as the grid is refined. The error on the coarsest grid (level 5 octree, $res_{max} = 32$ is represented on the left, the one on the finest grid (level 8, $res_{max} = 256$ is represented on the right

5. Concluding remarks

We have presented a novel second-order accurate model for solving the equations of linear elasticity in two and three spatial dimensions. Our method has been implemented on non-graded adaptive grids, i.e. grids for which the ratio between adjacent cells is not constrained. This allows to use a simple algorithm to generate a mesh with the finest resolution at the interface. The method makes use of a hybrid discretization near the interface and a finite difference scheme on quadtree/octree grids that lead to a straightforward approach. The linear systems are symmetric positive definite in the case of uniform grids and non-symmetric, but still diagonally dominant, in the case of non-graded grids. The model has been tested for both two and three spatial dimensional complex geometries and the second-order accuracy have been found for the L^1 and L^∞ norms. This work has been motivated by the desire to solve for the elasticity equations in complex domains that can undergo changes of geometries, as it occurs in shape optimization problems.

Acknowledgment

This research was supported in part by ONR under grant agreement N00014-11-1-0027, by the National Science Foundation under grant agreement CHE 1027817, by the Department of Energy under grant agreement DE-FG02-08ER15991, by the Institute for Collaborative Biotechnologies through contract No. W911NF-09-D-0001 from the U.S. Army Research Office and by the W.M. Keck Foundation.

References

[1] Grégoire Allaire, Conception Optimale de Structures, Springer, 2006.
[2] A. Almgren, J. Bell, P. Colella, L. Howell, M. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1–46.
[3] T. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349–355.
[4] I. Babuska, U. Banerjee, J.E. Osborn, Generalized finite element methods: main ideas results and perspective, Int. J. Comput. Methods 1 (2004) 67–103.
[5] Han Chen, Chohong Min, Frédéric Gibou, A supra-convergent finite difference scheme for the poisson and heat equations on irregular domains and non-graded adaptive Cartesian grids, J. Sci. Comput. 31 (1–2) (2007) 19–60.
[6] J. Chessa, T. Belytschko, An extended finite element method for two-phase fluids, ASME J. Appl. Mech. 70 (2003) 10–17.
[7] A. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[8] P. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland Publishing Company, 1979.
[9] D. Enright, D. Nguyen, F. Gibou, R. Fedkiw, Using the particle level set method and a second order accurate pressure boundary condition for free surface flows, in: Proceedings of 4th ASME-JSME Joint Fluids Engineering Conference, Number FEDSM2003–45144, ASME, 2003.
[10] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[11] A.-M. Toader, G. Allaire, F. Jouve, Structural optimization using sensitivity analysis and a level-set method, J. Comput. Phys. 194 (2004) 363–393.
[12] F. Gibou, A second-order-accurate symmetric discretization of the poisson equation on irregular domains, J. Comput. Phys. 176 (1) (2002) 205–227.
[13] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.
[14] F. Gibou, R. Fedkiw, R. Caflisch, S. Osher, A level set approach for the numerical simulation of dendritic growth, J. Sci. Comput. 19 (2003) 183–199.
[15] A. Helgadottir, F. Gibou, A poisson–boltzmann solver on irregular domains with Neumann or robin boundary conditions on non-graded adaptive grid, J. Comput. Phys. 230 (10) (2011) 3830–3848.
[16] A. Wiegmann, J.A. Sethian, Structural boundary design via level-set and immerse interface methods, J. Comput. Phys. 163 (2000) 489–528.
[17] R. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[18] X.D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 154 (2000) 151.
[19] C. Min, Local level set method in high dimension and codimension, J. Comput. Phys. 200 (2004) 368–382.
[20] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, J. Comput. Phys. 219 (2) (2006) 912–929.
[21] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, J. Comput. Phys. 226 (2) (2007) 1432–1443.
[22] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, J. Comput. Phys. 225 (2007) 300–321.
[23] C. Min, F. Gibou, H. Ceniceros, A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids, J. Comput. Phys. 218 (1) (2006) 123–140.
[24] Mohammad Mirzadeh, Maxime Theillard, Frédéric Gibou, A second-order discretization of the nonlinear poisson-boltzmann equation over irregular geometries using non-graded adaptive Cartesian grids, J. Comput. Phys. 230 (5) (2010) 2125–2140.

[25] Yen Ting Ng, Chohong Min, Frédéric Gibou, An efficient fluid–solid coupling algorithm for single-phase flows, J. Comput. Phys. 228 (23) (2009) 8807–8829.
[26] D. Nguyen, R. Fedkiw, H. Jensen, Physically based modeling and animation of fire, in: ACM Transactions on Graphics, (SIGGRAPH Proc.), vol. 29, 2002, pp. 721–728.
[27] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, 2002.
[28] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[29] Joseph Papac, Frédéric Gibou, Christian Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, J. Comput. Phys. 229 (3) (2010) 875–889.
[30] C. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[31] C. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[32] J.W. Purvis, J.E. Burkhalter, Prediction of critical mach number for store configurations, AIAA J. 17 (1979) 1170–1177.
[33] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley, New York, 1989.
[34] H. Samet, Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS, Addison-Wesley, New York, 1990.
[35] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.
[36] J.F. Santosa, S. Osher, Level-set methods for optimization problems involving geometry and constraint:frequencies of a two-density inhomogeneous drum, J. Comput. Phys. 171 (2001) 272–288.
[37] J. Strain, Tree methods for moving interfaces, J. Comput. Phys. 151 (1999) 616–648.