A second-order virtual node algorithm for nearly incompressible linear elasticity in irregular domains

Yongning Zhu[a], Yuting Wang[a], Jeffrey Hellrung[a], Alejandro Cantarero[a], Eftychios Sifakis[a], Joseph M. Teran[a]

[a] *University of California – Los Angeles, Department of Mathematics*
520 Portola Plaza, Math Sciences Building 6363, Los Angeles, CA 90095

Abstract

We present a cut cell method in $\mathbf{R}^2$ for enforcing Dirichlet and Neumann boundary conditions with nearly incompressible linear elastic materials in irregular domains. Virtual nodes on cut uniform grid cells are used to provide geometric flexibility in the domain boundary shape without sacrificing accuracy. We use a mixed formulation utilizing a MAC-type staggered grid with piecewise bilinear displacements centered at cell faces and piecewise constant pressures at cell centers. These discretization choices provide the necessary stability in the incompressible limit and the necessary accuracy in cut cells. Numerical experiments suggest second order accuracy in L^∞. We target high-resolution problems and present a class of geometric multigrid methods for solving the discrete equations for displacements and pressures that achieves nearly optimal convergence rates independent of grid resolution.

Keywords: Linear elasticity, Virtual node methods, Variational methods, Mixed finite element method

1. Introduction

We focus on the equilibrium equations of linear elasticity in an arbitrary domain Ω

$$-\nabla \cdot \boldsymbol{\sigma}(u) = f \qquad \text{in } \Omega \tag{1}$$
$$u = u_0 \qquad \text{on } \Gamma_d \tag{2}$$
$$(\boldsymbol{\sigma}(u) \cdot n) = \tau \qquad \text{on } \Gamma_n. \tag{3}$$

We use u to denote the material displacement map, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, f is the external force per unit area, u_0 is the prescribed Dirichlet boundary displacements (over the Dirichlet portion of $\partial\Omega$: Γ_d) and τ is the prescribed external surface traction (over the Neumann portion of $\partial\Omega$: Γ_n). In linear elasticity, the stress $\boldsymbol{\sigma}(u)$ is linearly dependent on the Cauchy strain $\boldsymbol{\epsilon}(u)$:

$$\boldsymbol{\epsilon}(u) = \frac{\nabla u + (\nabla u)^T}{2} \tag{4}$$
$$\boldsymbol{\sigma}(u) = 2\mu\,\boldsymbol{\epsilon}(u) + \lambda \operatorname{tr} \boldsymbol{\epsilon}(u)\,\mathbf{I} \tag{5}$$
$$= \mu\,(\nabla u + (\nabla u)^T) + \lambda\,(\nabla \cdot u)\,\mathbf{I}. \tag{6}$$

Therefore, the equations of linear elastic equilibrium can be equivalently written as

$$-(\mu\Delta\mathbf{I} + (\lambda+\mu)\nabla\nabla^T)u = f \qquad \text{in } \Omega \tag{7}$$
$$u = u_0 \qquad \text{on } \Gamma_d \tag{8}$$
$$\mu(u_n + \nabla(u \cdot n)) + \lambda(\nabla \cdot u)n = \tau \qquad \text{on } \Gamma_n. \tag{9}$$

*Correspondence to Alejandro Cantarero,

Email addresses: yzhu@math.ucla.edu (Yongning Zhu), yuting@math.ucla.edu (Yuting Wang), jhellrung@ucla.edu (Jeffrey Hellrung), cantarer@math.ucla.edu (Alejandro Cantarero), sifakis@math.ucla.edu (Eftychios Sifakis), jteran@math.ucla.edu (Joseph M. Teran)

When supporting irregular domain geometries, a natural choice for the numerical approximation of these equations is the finite element method (FEM) with unstructured meshes that conform to the geometry of $\partial\Omega$. However, meshing complex geometries can prove difficult and time-consuming when the boundary frequently changes shape. This task is even more difficult when using the more elaborate element types seen in mixed FEM formulations. Mixed formulations are typically necessary for stability with the nearly incompressible materials we consider here. In many applications, such as shape optimization for elastic materials or crack progagation, it is necessary to change the geometry of the domain at each iteration of a simulation. In such cases, frequent unstructured remeshing can be prohibitively costly (especially in 3D). Also, many numerical methods, such as finite differences, do not naturally apply to unstructured meshes. These concerns motivated the development of "embedded" (or "immersed") methods that approximate solutions to (7) on Cartesian grids or structured meshes that do not conform to the boundary. Retention of higher order accuracy in L^∞ with such embedding strategies is an ongoing area of research. Typically, the difficulty is to determine numerical stencils near the domain boundary that retain the accuracy achieved in the interior, away from the irregular features. While the accuracy of the discretization is an issue, the ability to use a structured and regular grid greatly facilitates the implementation of efficient solution methods. Furthermore, for high resolution discretizations, particularly in the case of nearly incompressible materials, efficient solution of the discrete systems can be challenging. In these cases, direct methods become too slow and memory intensive. Geometric multigrid methods and domain decomposition approaches have been shown to provide very favorable performance in this setting, however their application to embedded discretizations of irregular domains is not straightforward. Ultimately, special attention must be paid near the boundary for both discretization accuracy and efficient numerical linear algebra.

With these concerns in mind, we introduce a second order virtual node method for approximating the equations of linear elastic equilibrium with irregular embedded Neumann and Dirichlet boundaries on a uniform Cartesian grid. We use a regular grid because it simplifies the implementation, permits straightforward Lagrange multiplier spaces for Dirichlet constraints and naturally allows for higher order accuracy in L^∞. Furthermore, the approach has excellent potential for efficient parallel implementation as indicated by the similarities to the first order method investigated in [1]. The method is most suited for problems like level set-based shape optimization where the geometry of the domain is frequently changing and constant remeshing is a clearly inferior alternative to embedding (see e.g [2–7]). Also, we allow for nearly incompressible materials by introducing pressure as an additional unknown in a mixed variational formulation. Our discretization of this variational formulation is then based on a MAC-type staggering of x and y-component displacements with pressures at cell centers. Our approach combines piecewise bilinear interpolation of displacement components with the addition of "virtual" nodes on cut cells that accurately account for the irregular shape of the geometry boundary. The variational nature of the method naturally enables symmetric numerical stencils at the boundary. We use Lagrange multipliers to enforce embedded Dirichlet conditions weakly. In the general case, our choice of Lagrange multiplier space admits an efficient means for smoothing boundary equations in our geometric multigrid method for solving the discretized equations. Numerical experiments indicate second order accuracy in L^∞ independent of Poisson's ratio, domain geometry or boundary condition type.

The remainder of the paper proceeds as follows: in section 2 we discuss results from existing methods; in section 3 we discuss the mixed variational formulation we use to accurately handle the nearly incompressible case; in section 3.1 we detail our use of embedding over MAC-type staggered grids to discretize the mixed formulation; in section 4 we develop a novel approach to enforcing Dirichlet boundary conditions as a constraint on the Neumann problem and lastly we discuss a novel geometric multigrid method for the solution of our discretized equations in section 5.

2. Existing methods

Our approach is second-order accurate in L^∞ with both embedded Neumann and embedded Dirichlet boundary conditions over irregular domains and our discrete systems are developed to facilitate a class of mulitgrid methods that achieve nearly optimal convergence rates independent of grid resolution. Furthermore, our method retains these properties with nearly incompressible materials. While there are many existing approaches for linear elasticity, particularly in the case of unstructured mesh based approaches, ours is the first embedded method to support this feature set. In our discussion of existing approaches, we will focus only on methods that avoid unstructured meshing when addressing irregular boundaries for incompressible materials. Also, we will discuss methods addressing the difficulties that arise when ensuring scalability to high resolution problems, particularly in the case of nearly incompressible materials.

Embedded techniques use a computational domain that simply encompasses rather than geometrically adheres to the irregular domain (see Figure 1). This avoids the difficulties associated with unstructured mesh generation, but can complicate the enforcement of boundary conditions. A good review of these issues is given by Lew et al. in [8]. They point out that these techniques actually originated with the papers of Harlow and Welch [9] and Charles Peskin [10], at least in the context of incompressible materials. Some of the first embedded methods were fictitious domain methods by Hyman [11] and Saul'ev [12]. The fictitious domain approach has been used with incompressible materials in a number of works [13–21]. These approaches embed the irregular geometry in a more simplistic domain for which fast solvers exist (e.g. Fast Fourier Transforms). The calculations include fictitious material in the complement of the domain of interest. A forcing term (often from a Lagrange multiplier) is used to maintain boundary conditions at the irregular geometry. Although these techniques naturally allow for efficient solution procedures, they depend on a smooth solution across the embedded domain geometry for optimal accuracy, which is not typically possible.

The extended finite element method (XFEM) and related approaches in the finite element literature also make use of geometry embedded in regular elements. Although originally developed for crack-based field discontinuities in elasticity problems, these techniques are also used with embedded problems in irregular domains. Daux et al. first showed that these techniques can naturally capture embedded Neumann boundary conditions [22, 23]. These approaches are equivalent to the variational cut cell method of Almgren et al. in [24]. Enforcement of Dirichlet constraints is more difficult with variational cut cell approaches [8, 25] and typically involves a Lagrange multiplier or stabilization. Dolbow and Devan recently investigated the convergence of such approaches with incompressible materials and point out that much analysis in this context remains to be completed [26]. Despite the lack of thorough analysis, such XFEM approaches appear to be very accurate and have been used in many applications involving incompressible materials in irregular domains [27–31].

There are also many finite difference (FDM) and finite volume methods (FVM) that utilize cut uniform grid cells. Many of these methods have been developed in the context of incompressible flow. For example, Almgren et al. use cut uniform bilinear cells to solve the Poisson equation for pressures in incompressible flow calculations [24]. Marelle et al. use collocated grids and define define sub cell interface and boundary geometry in cut cells via level sets [32]. Ng et al. also use level set descriptions of the irregular domain and achieve second order accuracy in L^∞ for incompressible flows [33]. The approach of Batty and Bridson is similar, but not as accurate [34]. Although not technically a cut cell approach, the immersed interface method has been used to improve accuracy for incompressible flow calculations in irregular domains [21, 35–39]. Cut cell FDM and FVM have also been developed for incompressible and nearly incompressible elastic materials. Bijelonja et al. use cut cell FVM to enforce incompressibility more accurately than is typically seen with FEM [40]. Beirão da Veiga et al. use polygonal FVM cells to avoid remeshing with irregular domains [41]. Barton et al. [42] and Hill et al. [43] use cut cells with Eulerian elastic/plastic flows.

Many approaches have been proposed to solve elasticity equations in a scalable way at high resoultions. For this class of problems, iterative methods are usually employed rather than direct methods due to the amount memory needed to use such methods. For iterative methods to be scalable, we mean that the method requires only a constant (and small) number of iterations to obtain a solution and that that constant is independent of the grid resolution. While many methods of this type have proven quite effective, accommodating mixed boundary conditions on an embedded interface is highly nontrivial, especially when efficiency of implementation is a priority. Most methods have also been created specifically to work with either pure Dirichlet or pure traction boundary conditions, but have not been demonstrated to be effective in both cases. Constructing preconditioners for solving the KKT systems that result from discretizing the equations in a mixed formulation have been studied by Klawonn [44, 45] and Bramble and Pasciak [46]. Work has also been done on using domain decomposition methods with PCG [47] and GMRES [48] to solve Stokes and elasticity problems. Balancing domain decomposition by constraints (BDDC) has also been used to build preconditioners for solving these problems [49, 50]. Many authors have also looked at applying multigrid methods to problems in solid mechanics (e.g. [1, 51–62]), including handling issues arising from nearly incompressible materials. Mixed FEM formulations are one example that maintain good multigrid convergence properties for nearly incompressible materials demonstrated on the pure Dirichlet boundary case [54, 58, 61]. FOSLS methods have been demonstrated to produce systems that can be effectively solved using algebraic multigrid methods by rewriting the elasticity equations as a first order system using least squares [55, 60]. Multigrid applied to FEM discretized equations using a smoother based on a Schur complement has been studied by different authors [57, 59]. While demonstrating the ability to solve large problems, the Schur complement approach requires the action of the inverse of the displacements matrix in the smoothing process which is a more expensive smoothing operation than that offered by

other methods. Distributive smoothers offer a different option for the smoothing process that has proved effective on elasticity equations discretized with FEM [56] and on staggered grids [62]. In our approach, we will look at using distributive smoothing similar to those described in [62].

3. Mixed finite element formulation

In order to accurately handle linear elastic materials near the incompressible limit, we use an augmented form of the equilibrium equations. By introducing a pressure variable as an unknown, we can achieve a stable numerical discretization independent of the degree of incompressibility. We will use the weak form of this augmented system to derive a mixed finite element formulation [63]. The augmented form of our equations arises by introducing $p = -\lambda/\mu \nabla \cdot u$. With this definition, $\boldsymbol{\sigma}(u) = \mu\,(\nabla u + (\nabla u)^T) - \mu\, p\, \mathbf{I}$ and the derived PDE

$$-\mu(\Delta \mathbf{I} + \nabla\nabla^T)u + \mu\nabla p = \quad f \quad \text{in } \Omega \tag{10}$$

$$-\mu\nabla \cdot u - \frac{\mu^2}{\lambda} p = \quad 0 \quad \text{in } \Omega \tag{11}$$

$$u = \quad u_0 \quad \text{on } \Gamma_d \tag{12}$$

$$\mu(u_n + \nabla(u \cdot n)) - \mu p\, n = \quad g \quad \text{on } \Gamma_n \tag{13}$$

is then equivalent to the original PDE.

We use this augmented form of the equations to derive an equivalent variational form of the linear elastic equilibrium equations. A weak form can be derived by taking the inner product of the strong form with an arbitrary vector valued function $v \in V_0 = (H^1_{0,\Gamma_d}(\Omega))^d$ and by enforcing the equation $p = -\lambda/\mu \nabla \cdot u$ weakly:

$$\text{Find } (\boldsymbol{u}, p) \in H^1(\Omega) \times H^1(\Omega) \times L^2(\Omega), \boldsymbol{u}|_{\Gamma_d} = \boldsymbol{u}_0, \text{ such that} \tag{14}$$

$$\int_\Omega 2\mu \left(\frac{\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T}{2} \right) : \left(\frac{\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T}{2} \right) - \mu p (\nabla \cdot \boldsymbol{v})\, d\boldsymbol{x} \tag{15}$$

$$= -\int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\boldsymbol{x} + \int_{\partial\Omega} \boldsymbol{g} \cdot \boldsymbol{v}\, ds \quad \forall \boldsymbol{v} \in H^1_{0,\Gamma_d}(\Omega) \times H^1_{0,\Gamma_d}(\Omega) \tag{16}$$

$$\int_\Omega \left(-\mu q \nabla \cdot \boldsymbol{u} - \frac{\mu^2}{\lambda} pq \right) d\boldsymbol{x} = 0 \quad \forall q \in L^2(\Omega). \tag{17}$$

3.1. Discretization

We discretize this variational formulation using a mixed finite element method defined on a MAC-type staggered grid. Han et al. demonstrated the stability and optimal convergence of this formulation applied to the Stokes equations on a square domain [64]. We generalize this approach to the case of nearly incompressible linear elasticity in embedded domains. We approximate the Sobolev space $V = H^1(\Omega) \times H^1(\Omega)$ with a finite element subspace V^h, where each displacement component of a function in V^h is represented as a piecewise bilinear scalar function defined on a staggered quadrilateral grid (see Figure 1). To be more specific, consider the staggered grids:

$$\mathcal{G}^x_h = \{(ih, (j - 1/2)h) : (i, j) \in I_x \subset Z^2\},$$

$$\mathcal{G}^y_h = \{((i - 1/2)h, jh) : (i, j) \in I_y \subset Z^2\}.$$

Here, h is the discrete spacing between grid points. Furthermore, we use the following notation to denote quadrilaterals defined by these grids:

$$T^x_{ij} = \{(x, y) : ih < x < (i + 1)h, (j - 1/2)h < y < (j + 1/2)h\},$$

$$T^y_{ij} = \{(x, y) : (i - 1/2)h < x < (i + 1/2)h, jh < y < (j + 1)h\}.$$

The sets I_x and I_y used in the definition of grids $\mathcal{G}^x_h$ and $\mathcal{G}^y_h$ are defined as the collection of vertices incident on some quadrilateral T^x_{ij} or T^y_{ij}, respectively, whose intersection with the domain Ω is non-empty. In other words, I_x and I_y are the sets of vertices in the staggered lattices that are at most a distance of h away from Ω. Henceforth, we will use

$$\mathcal{T}^x_h = \{T^x_{ij} : \ T^x_{ij} \cap \Omega \neq \emptyset\}$$

and

$$\mathcal{T}_h^y = \{T_{ij}^y : \ T_{ij}^y \cap \Omega \neq \emptyset\}$$

to denote the collection of x and y grid quadrilaterals that intersect (or embed) the domain Ω. We construct two subspaces of $H^1(\Omega)$ based on these quadrangulations respectively:

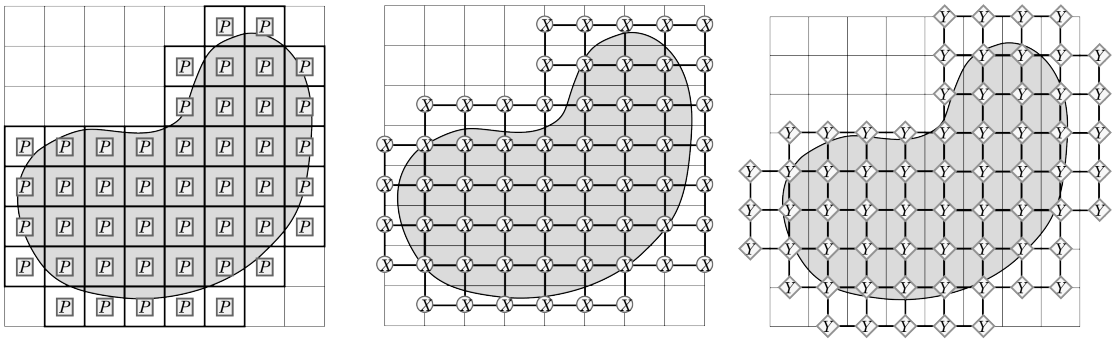
$$V_x^h = \{v_h \in C^{(0)}(\Omega) : v_h|_{T_{ij}^x} \in Q_1(T_{ij}^x) \ \forall \ T_{ij}^x \in \mathcal{T}_h^x\},$$

$$V_y^h = \{v_h \in C^{(0)}(\Omega) : v_h|_{T_{ij}^y} \in Q_1(T_{ij}^y) \ \forall \ T_{ij}^y \in \mathcal{T}_h^y\}$$

where $Q_1(T_{ij}^k)$ is the space of bilinear functions on the quadrilateral T_{ij}^k. For simplicity of notation in subsequent equations we will also use mappings $\eta_1 : I_1 = \{1, 2, ..., N_x\} \rightarrow I_x$ and $\eta_2 : I_2 = \{1, 2, ..., N_y\} \rightarrow I_y$ to associate each x and y grid vertex with a unique integer between 1 and $N_x = |I_x|$ and 1 and $N_y = |I_y|$ respectively. With this convention, any approximated solution $\boldsymbol{u} \in V_x^h \times V_y^h$ can be expressed as

$$\boldsymbol{u}(\boldsymbol{x}) \quad = \quad \begin{pmatrix} \sum\limits_{k_1 \in I_1} u_{1k_1} N_{1k_1}(\boldsymbol{x}) \\ \sum\limits_{k_2 \in I_2} u_{2k_2} N_{2k_2}(\boldsymbol{x}) \end{pmatrix} \tag{18}$$

where N_{1k_1} and N_{2k_2} are the commonly used piecewise bilinear interpolating functions associated with nodes k_1 and k_2 respectively in $\mathcal{T}_h^x$ and $\mathcal{T}_h^y$. Our discrete equations for the approximate solution $\boldsymbol{u}$ can thus be seen to be over $N_x + N_y$ unknowns.



(a) $\mathcal{T}_h^p$ (b) $\mathcal{T}_h^x$ (c) $\mathcal{T}_h^y$

Figure 1: Staggered grid finite element quadrangulation and embedded domain boundary.

We additionally approximate the space for pressure $V_p = L^2(\Omega)$ with a piecewise constant finite element space V_p^h defined on a quadrangulation $\mathcal{T}_h^p$ over the primary grid (or henceforth, the pressure grid) $\mathcal{G}_h^p$:

$$\mathcal{G}_h^p = \{((i + 1/2)h, (j + 1/2)h) : (i, j) \in I_p \subset Z^2\},$$

$$T_{ij}^p = \{(x, y) : ih < x < (i + 1)h, \ jh < y < (j + 1)h\},$$

$$\mathcal{T}_h^p = \{T_{ij}^p : T_{ij}^p \cap \Omega \neq \emptyset\},$$

$$V_p^h = \{p_h \in L^2(\Omega) : p_h|_{T_{ij}^p} \in P_0(T_{ij}^p) \ \forall \ T_{ij}^p \in \mathcal{T}_h^p\}$$

where $P_0(T_{ij}^p)$ is the space of constant functions on the quadrilateral T_{ij}^p. The grid $\mathcal{G}_h^p$ is a cell-centered grid (as opposed to a node-centered grid, such as $\mathcal{G}_h^x$ or $\mathcal{G}_h^y$). That is, we assume that pressure variables live at the cell centers of this grid. In other words, there is one pressure variable located in each $T_{ij}^p \in \mathcal{T}_h^p$. The set I_p is defined similarly to I_x and I_y, however here it refers to the collection of cell centered indices in the grid $\mathcal{G}_h^p$ whose associated quadrilaterals

T^p_{ij} have a non-zero intersection with Ω. For the sake of simplicity in subsequent equations, we again use a mapping $\eta_3 : I_3 = \{1, 2, ..., N_p\} \to I_p$ to associate each cell in the pressure grid with a unique integer between 1 and $N_p = |I_p|$. Thus, any approximated solution of the pressure has the following representation:

$$p(\boldsymbol{x}) = \sum_{k_3 \in I_3} p_{k_3} \chi_{T^p_{k_3}}(\boldsymbol{x}) \tag{19}$$

where $\chi_{T^p_k}(\boldsymbol{x})$ is the characteristic function for the quadrilateral T^p_k. That is,

$$\chi_{T^p_k}(x) = \begin{cases} 1, & \boldsymbol{x} \in T^p_k \\ 0, & \boldsymbol{x} \notin T^p_k. \end{cases}$$

We choose test functions $\boldsymbol{v}(\boldsymbol{x}) = N_{mk_m}(\boldsymbol{x})\boldsymbol{e}_m,\ m = 1, 2$ and substitute the finite element discretization (18), (19) into each term in the mixed variational form (14),

$$\begin{aligned}
& 2\mu \int_\Omega \frac{\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T}{2} : \frac{\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T}{2}\, d\boldsymbol{x} \\
= \ & \mu \int_\Omega \nabla \boldsymbol{u} : (\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T)\, d\boldsymbol{x} \\
= \ & \mu \sum_{i,j \in \{1,2\}} \int_\Omega u_{i,j}(v_{i,j} + v_{j,i})\, d\boldsymbol{x} = \mu \sum_{i,j \in \{1,2\}} \int_\Omega u_{i,j}(N_{mk_m,j}(\boldsymbol{x})\delta_{mi} + N_{mk_m,i}(\boldsymbol{x})\delta_{mj})\, d\boldsymbol{x} \\
= \ & \mu \sum_{i \in \{1,2\}} \int_\Omega (u_{i,m} + u_{m,i}) N_{mk_m,i}(\boldsymbol{x}) = \mu \sum_{i \in \{1,2\}} \int_\Omega \left(\sum_{k_i \in I_i} u_{ik_i} N_{ik_i,m}(\boldsymbol{x}) + \sum_{k_m \in I_m} u_{mk_m} N_{mk_m,i}(\boldsymbol{x}) \right) N_{mk_m,i}(\boldsymbol{x}) \\
= \ & \mu \sum_{i \in \{1,2\}} \sum_{k_i \in I_i} u_{ik_i} \int_\Omega N_{ik_i,m}(\boldsymbol{x}) N_{mk_m,i}(\boldsymbol{x}) + \mu \sum_{k_m \in I_m} u_{mk_m} \sum_{i \in \{1,2\}} \int_\Omega N^2_{mk_m,i}(\boldsymbol{x})
\end{aligned}$$

$$\begin{aligned}
-\mu \int_\Omega p\, \nabla \cdot \boldsymbol{v} d\boldsymbol{x} &= -\mu \sum_{k_p \in I_p} p_{k_p} \int_{T^p_{k_p} \cap \Omega} N_{mk_m,m}(\boldsymbol{x}) d\boldsymbol{x} \\
\int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\boldsymbol{x} &= \int_\Omega f_m N_{mk_m}(\boldsymbol{x})\, d\boldsymbol{x} \\
\int_{\Gamma_n} \boldsymbol{g} \cdot \boldsymbol{v}\, ds &= \int_{\Gamma_n} g_m N_{mk_m}(\boldsymbol{x})\, ds.
\end{aligned}$$

We can also choose $\boldsymbol{v} = 0$ and $q(\boldsymbol{x}) = \chi_{T^p_{k_p}}(\boldsymbol{x})$ to derive the pressure equations:

$$-\mu \sum_{i \in \{1,2\}} \sum_{k_i \in I_i} u_{ik_i} \int_{T^p_{k_p} \cap \Omega} N_{ik_i,i}(\boldsymbol{x})\, d\boldsymbol{x} - \frac{\mu^2}{\lambda} \sum_{k_p \in I_p} p_{k_p} \int_{T^p_{k_p} \cap \Omega} 1\, d\boldsymbol{x} = 0.$$

Since the variational form is derived from an energy minimization problem, the discretized linear system can trivially be seen to be symmetric. Specifically, if we take the convention that $\boldsymbol{u}^h \in \mathbb{R}^{N_x + N_y}$ is our vector of displacement unknowns (where we assume that x degrees of freedom are ordered first and y second) and $\boldsymbol{p}^h \in \mathbb{R}^{N_p}$ is the vector of pressure unknowns, then our system over the vector $\hat{\boldsymbol{u}}^h$ of $N = N_x + N_y + N_p$ degrees of freedom is of the form:

$$\begin{pmatrix} \mathbf{L}^h_u & \mathbf{G}^{hT} \\ \mathbf{G}^h & \mathbf{D}^h_p \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^h \\ \boldsymbol{p}^h \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^h \\ \mathbf{0} \end{pmatrix} \qquad \text{or} \qquad \hat{\mathbf{L}}^h \hat{\boldsymbol{u}}^h = \hat{\boldsymbol{f}}^h \tag{20}$$

where $\hat{\boldsymbol{u}}^h = (\boldsymbol{u}^h, p^h)$ and $\hat{\boldsymbol{f}}^h = (\boldsymbol{f}^h, 0)$. Furthermore, our use of regular grids gives the discrete equations a finite difference interpretation. If we scale the system by $\frac{1}{h^2}$, each block in the discrete system approximates the corresponding differential operator in (10), i.e. (20) discretizes the following equation:

$$h^2 \begin{pmatrix} -\mu(\Delta + \nabla\nabla^T) & \mu\nabla \\ -\mu\nabla^T & -\frac{\mu^2}{\lambda} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} h^2 \\ 0 \end{pmatrix}. \tag{21}$$

The linear system is the Hessian matrix of a saddle point problem, therefore the discretized system is symmetric but indefinite. In fact, the first diagonal block $\mathbf{L}_u^h$ is positive definite, and the other diagonal block is negative definite.

3.2. Implementation details

For ease of implementation, we perform the integrations involved in the discrete equations in an element-by-element fashion. Each area integral is represented as a sum of integrals over spatially disjoint elements whose union is the embedded domain. Specifically, we individually address the integration over the intersection of each quadrilateral of the pressure grid with the embedded domain $T_{k_p}^p \cap \Omega$:

$$\int_\Omega N_{mk_m,i}^2(\boldsymbol{x})\,d\boldsymbol{x} = \sum_{k_p \in I_p} \int_{T_{k_p}^p \cap \Omega} N_{mk_m,i}^2(\boldsymbol{x})\,d\boldsymbol{x}$$

$$\int_\Omega N_{ik_i,m}(\boldsymbol{x}) N_{mk_m,i}(\boldsymbol{x})\,d\boldsymbol{x} = \sum_{k_p \in I_p} \int_{T_{k_p}^p \cap \Omega} N_{ik_i,m}(\boldsymbol{x}) N_{mk_m,i}(\boldsymbol{x})\,d\boldsymbol{x}$$

$$\int_\Omega N_{mk_m}(\boldsymbol{x})\,d\boldsymbol{x} = \sum_{k_p \in I_p} \int_{T_{k_p}^p \cap \Omega} N_{mk_m}(\boldsymbol{x})\,d\boldsymbol{x}$$

$$\int_{\Gamma_n} N_{mk_m}(\boldsymbol{x})\,ds = \sum_{k_p \in I_p} \int_{T_{k_p}^p \cap \Gamma_n} N_{mk_m}(\boldsymbol{x})\,ds.$$

In the interior, this simply amounts to evaluating the same integrals over each full quadrilateral $T_{k_p}^p$. However, at the boundary, care must be taken to respect the material region alone when the intersection between the pressure cells and the embedded domain is non-trivial. In both the boundary and interior cases there will be 13 degrees of freedom involved in the integration over such a pressure cell. This is because the staggering of variables leads to 13 interpolating functions supported over a given pressure cell (6 x-components, 6 y-components and one pressure). In other words, we express the matrix (or stiffness matrix) in our discrete system as a sum of 13×13 element stiffness matrices $\mathbf{A}^{k_p}$. Furthermore, we break the integrals involved in a given element $T_{k_p}^p$ up into four subintegrals over the subquadrants $(\omega_1, \omega_2, \omega_3, \omega_4)$ of $T_{k_p}^p$ (see Figure 3). This is because the integrands are all smooth over these regions. Notably, they are quadratic and we simply preform these integrations analytically. The elemental stiffness matrix is accumulated from the following sub-stiffness-matrices $\mathbf{A}^{k_p} = \mathbf{A}_{\omega_1}^{k_p} + \mathbf{A}_{\omega_2}^{k_p} + \mathbf{A}_{\omega_3}^{k_p} + \mathbf{A}_{\omega_4}^{k_p}$. For example, $\mathbf{A}_{\omega_1}^{k_p}$ involves x_1, x_2, x_3, x_4, y_7, y_8, y_{10} and y_{11} as demonstrated in Figure 3 center, therefore, it only has non-zero values on rows and columns involving these degrees of freedom. The resulting equations based on those degrees of freedom are shown in Figure 2. If we order the 13 nodes with indices shown in Figure 3 left, then on the interior of the domain, where $T_{k_p}^p \cap \Omega = T_{k_p}^p$, the sum of these four subintegrals is always the same:

$$\mathbf{A}^{k_p} = \begin{pmatrix} \mu \begin{pmatrix} 1/4 & 0 & 0 & -1/4 & 0 & 0 & 9/64-3/32-3/64 & 3/64-1/32-1/64 \\ 0 & 1/4 & -1/4 & 0 & 0 & 0 & 3/64 & 3/32-9/64 & 1/64 & 1/32-3/64 \\ 0 & -1/4 & 3/2 & -1 & 0 & -1/4-3/32 & 1/16 & 1/32 & 3/32-1/16-1/32 \\ -1/4 & 0 & -1 & 3/2 & -1/4 & 0-1/32-1/16 & 3/32 & 1/32 & 1/16-3/32 \\ 0 & 0 & 0 & -1/4 & 1/4 & 0-3/64 & 1/32 & 1/64-9/64 & 3/32 & 3/64 \\ 0 & 0 & -1/4 & 0 & 0 & 1/4-1/64-1/32 & 3/64-3/64-3/32 & 9/64 \\ 9/64 & 3/64-3/32-1/32-3/64-1/64 & 1/4 & 0 & 0 & 0 & -1/4 & 0 \\ -3/32 & 3/32 & 1/16-1/16 & 1/32-1/32 & 0 & 3/2 & 0 & -1/4 & -1 & -1/4 \\ -3/64-9/64 & 1/32 & 3/32 & 1/64 & 3/64 & 0 & 0 & 1/4 & 0 & -1/4 & 0 \\ 3/64 & 1/64 & 3/32 & 1/32-9/64-3/64 & 0 & -1/4 & 0 & 1/4 & 0 & 0 \\ -1/32 & 1/32-1/16 & 1/16 & 3/32-3/32 & -1/4 & -64 & -1/4 & 0 & 3/2 & 0 \end{pmatrix} & -\mu h \begin{pmatrix} -1/8 \\ 1/8 \\ -3/4 \\ 3/4 \\ -1/8 \\ 1/8 \\ -1/8 \\ -3/4 \\ -1/8 \\ 1/8 \\ 3/4 \\ 1/8 \end{pmatrix} \\ -\mu h \begin{pmatrix} -1/8 & 1/8 & -3/4 & 3/4 & -1/8 & 1/8 & -1/8 & -3/4 & -1/8 & 1/8 & 3/4 & 1/8 \end{pmatrix} & -\dfrac{\mu^2}{\lambda} h^2 \end{pmatrix}. \tag{22}$$

The global stiffness matrix generated from $\mathbf{A}^{k_p}$ has a stencil shown in Figure 4.

However for boundary cells where $T_{k_p}^p \cap \Omega \neq T_{k_p}^p$, we have to perform the integrations involved in each of $\mathbf{A}_{\omega_i}^{k_p}$ carefully, taking into account the boundary geometry. We discuss this in the next section. The process of constructing the global stiffness matrix $\mathbf{A}$ from each of the 13×13 element stiffness matrices $\mathbf{A}^{k_p}$ is explained in Algorithm 1.

$\mathbf{A}_{\omega_1}^{k_p} =$

i \ j	X_1	X_2	X_3	X_4	Y_7	Y_8	Y_{10}	Y_{11}	P_{13}
X_1	$\mu\int_{\omega_1} \nabla N_{xk_i}\cdot\nabla N_{xk_j} + N_{xk_i,x}N_{xk_j,x}\,dx$				$\mu\int_{\omega_1} N_{xk_i,y}N_{yk_j,x}dx$				$-\mu\int_{\omega_1} N_{xk_i,x}dx$
X_2									
X_3									
X_4									
Y_7	$\mu\int_{\omega_1} N_{yk_i,x}N_{xk_j,y}dx$				$\mu\int_{\omega_1} \nabla N_{yk_i}\cdot\nabla N_{yk_j} + N_{yk_i,y}N_{yk_j,y}\,dx$				$-\mu\int_{\omega_1} N_{yk_i,y}dx$
Y_8									
Y_{10}									
Y_{11}									
P_{13}	$-\mu\int_{\omega_1} N_{xk_j,x}dx$				$-\mu\int_{\omega_1} N_{yk_j,y}dx$				$-\mu^2/\lambda\int_{\omega_1} 1dx$

Figure 2: Equations used to build the element stiffness matrix $\mathbf{A}_{\omega_1}^{k_p}$.

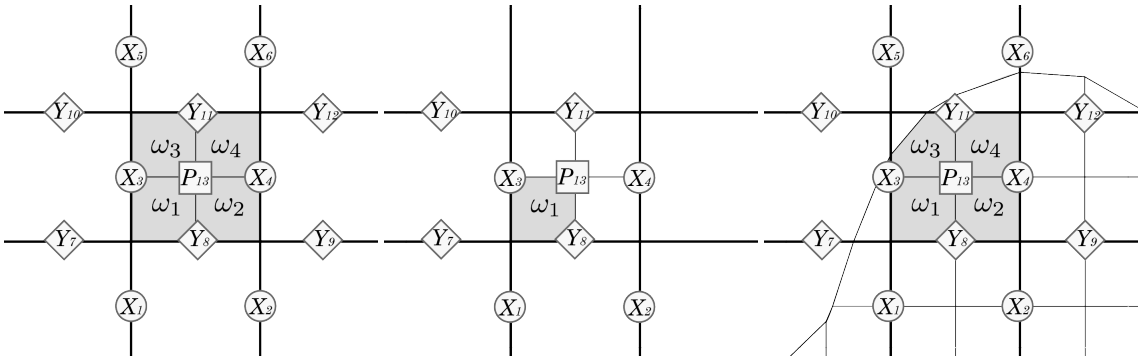


Figure 3: Left: one interior pressure cell and the variables corresponding to the 13 degrees of freedom of the elemental stiffness matrix by taking integral over the pressure cell. Right: the four integral subcells of the pressure cell. Center: the variables that an integral over subcell ω_1 contributes to.

3.3. Discrete geometric representation and cut cell integration

We discretize the domain Ω by embedding it in a regular grid. Specifically, we use a signed distance level set function defined over a doubly refined subgrid:

$$\mathcal{G}^{\phi} = \{(ih/2, jh/2)\}.$$

This doubly refined subgrid is thus a superset of all grid nodes in the x, y and p grids. The signed distance values at the nodes of the doubly refined grid $\mathcal{G}^{\phi}$ are used to determine the points of intersection between the zero isocontour and the coordinate axes aligned edges of $\mathcal{G}^{\phi}$. The boundary of Ω is then approximated by a segmented curve $\partial\Omega^h$ connecting these intersection points. The geometric domain is approximated within the region enclosed by $\partial\Omega^h$ (see Figure 5). Near the boundary, the domain within each subgrid cell is approximated by a polygon determined from the boundary edges of the subgrid cell and by straight lines that connect boundary intersection points as demonstrated in Figure 5. Thus we can think of our discrete domain as a union of doubly refined uncut quadrilaterals on the interior and cut polygonal regions contained in doubly refined quadrilaterals on the boundary.

This partitioning of the domain into doubly refined quadrilaterals naturally supports our integration conventions needed for the matrices $\mathbf{A}_{\omega_i}^{k_p}$ discussed in the previous section. The integrals needed for these matrices are evaluated trivially when ω_i is not cut. However when ω_i is cut by the boundary, we can still perform the integrations analytically following ideas from the recent cut cell approach in [65]. The integrands of each term in the matrices $\mathbf{A}_{\omega_i}^{k_p}$ are

Algorithm 1 Construction of global stiffness matrix $\mathbf{A}$ from elemental $\mathbf{A}^{k_p}$

$\mathbf{A} \leftarrow \mathbf{0}$
for $k_p = 1$ to N_p **do**
  if $T^p_{k_p} \cap \Omega = T^p_{k_p}$ **then** ▷ Interior cell: use precomputed $\mathbf{A}^{k_p}$
    Use $\mathbf{A}^{k_p}$ from equation (22)
  else ▷ Boundary cell: compute $\mathbf{A}^{k_p}$ from $\mathbf{A}^{k_p}_{\omega_i}$
    Perform integration over each subquadrant ω_i to compute $\mathbf{A}^{k_p}_{\omega_i}$
    $\mathbf{A}^{k_p} = \mathbf{A}^{k_p}_{\omega_1} + \mathbf{A}^{k_p}_{\omega_2} + \mathbf{A}^{k_p}_{\omega_3} + \mathbf{A}^{k_p}_{\omega_4}$
  end if
  for $i^p = 1$ to 13 **do**
    $i = \text{mesh}(k_p, i^p)$ ▷ The mesh maps the 13 degrees of freedom involved in $\mathbf{A}^{k_p}$ to their position in a global array
    for $j^p = 1$ to 13 **do**
      $j = \text{mesh}(k_p, j^p)$
      $A_{ij} += A^{k_p}_{i^p j^p}$
    end for
  end for
end for

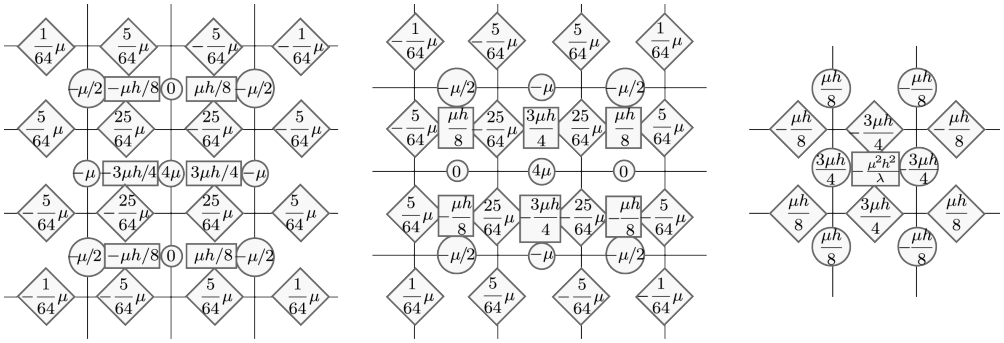
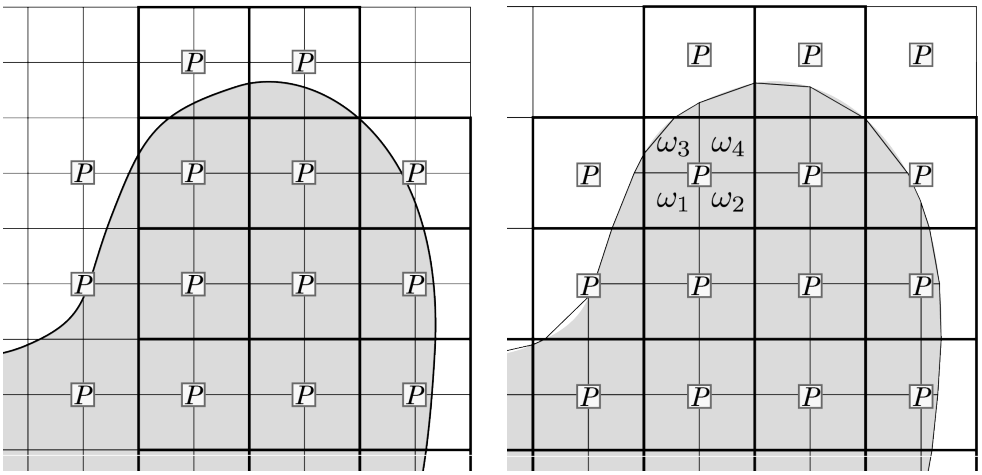


Figure 4: Global stiffness matrix stencils centered at an interior x variable (left), y variable (middle) and p variable (right).



(a) (b)

Figure 5: A zoom-in view of Figure 1(a). A levelset function is sampled on a doubly refined grid (left); a segmented curve $\partial\Omega_h$ is generated to approximate the boundary of the geometric domain (right).

polynomials in x and y of degree 2. That is, each integral is of the form:

$$\int_{\omega_i\cap\Omega} ax^2 + bxy + cy^2 + dx + ey + f d\boldsymbol{x}$$

Our level set based representation of the geometry means that the domain of integration $\omega_i \cap \Omega$ is polygonal. In other words, we need to evaluate a second order polynomial over a polygonal domain. This task can be done trivially by noting that

$$\int_{\omega_i\cap\Omega} ax^2 + bxy + cy^2 + dx + ey + f d\boldsymbol{x} = \int_{\omega_i\cap\Omega} \nabla\cdot\begin{pmatrix} \frac{ax^3}{3} + \frac{bx^2y}{2} + cxy^2 + \frac{dx^2}{2} + exy + fx \\ 0 \end{pmatrix} d\boldsymbol{x}.$$

That is, because $\omega_i \cap \Omega$ is polygonal and our integrand can be expressed in terms of the divergence of a (non-unique) cubic function, application of the divergence theorem yields the easily evaluated forumula:

$$\int_{\omega_i\cap\Omega} \nabla\cdot\begin{pmatrix} \frac{ax^3}{3} + \frac{bx^2y}{2} + cxy^2 + \frac{dx^2}{2} + exy + fx \\ 0 \end{pmatrix} d\boldsymbol{x} = \sum_{s=1}^{N_{\partial(\omega_i\cap\Omega)}} n_{1s} \int_{\partial(\omega_i\cap\Omega)_s} \left(\hat{a}(s)t^3 + \hat{b}(s)t^2 + \hat{c}(s)t + \hat{d}(s)\right) dt.$$

Here, the $N_{\partial(\omega_i\cap\Omega)}$ is the number of line segments in the boundary of the polygonal domain, $\partial(\omega_i \cap \Omega)_s$ is the s-th segment in the polygonal boundary, parameterized by the arc-length variable t, n_{1s} is the x component of the outward normal to the s-th segment and $\hat{a}(s), \hat{b}(s), \hat{c}(s), \hat{d}(s)$ are the cubic coefficients arising in the boundary integrals over each segment. Again, each term in the sum can be evaluated analytically. This careful treatment of the integrals arising in each $\mathbf{A}_{\omega_i}^{k_p}$ is the key to obtaining second order accuracy in L^∞.

4. Dirichlet boundary conditions

We have thus far assumed that our solution satisfies the Dirichlet boundary conditions and that our test functions vanish on the Dirichlet boundary. However, because we use a regular grid that does not conform to the actual domain, it is not convenient to directly define a finite element space with a specific value at the irregular boundary. Instead, we

can enforce these conditions weakly using the following variational problem:

$$\text{find } (\boldsymbol{u}, p) \in H^1(\Omega) \times H^1(\Omega) \times L^2(\Omega), \text{ such that}$$

$$\int_\Omega 2\mu \left(\frac{\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T}{2} \right) : \left(\frac{\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T}{2} \right) + \mu p (\nabla \cdot \boldsymbol{v}) \, d\boldsymbol{x}$$

$$= -\int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x} + \int_{\partial\Omega} \boldsymbol{g} \cdot \boldsymbol{v} \, ds \quad \forall \boldsymbol{v} \in H^1_{0,\Gamma_d}(\Omega) \times H^1_{0,\Gamma_d}(\Omega) \tag{23}$$

$$\int_\Omega \left(-\mu q \nabla \cdot \boldsymbol{u} - \frac{\mu^2}{\lambda} pq \right) d\boldsymbol{x} = 0 \quad \forall q \in L^2(\Omega) \tag{24}$$

$$\int_{\Gamma_d} \boldsymbol{u} \cdot \boldsymbol{w} \, ds = \int_{\Gamma_d} \boldsymbol{u}_0 \cdot \boldsymbol{w} \, ds \quad \forall \boldsymbol{w} \in (H^{-1/2}(\Gamma_d))^2. \tag{25}$$

Here, we introduce the Dirichlet condition as a constraint. Specifically, we require that the L^2 inner product of the solution and an arbitrary function $\boldsymbol{w} \in (H^{-1/2}(\Gamma_d))^2$ is the same as the inner product of the Dirichlet data $\boldsymbol{u}_0$ with $\boldsymbol{w}$. This makes the problem a constrained minimization.

4.1. Discretizing the Dirichlet problem

In order to discretize the Dirichlet condition in the weak formulation, we approximate $(H^{-1/2}(\Gamma_d))^2$ using a subspace $\Lambda^h_x \times \Lambda^h_y = P_0(\mathcal{T}^x \cap \Gamma^h_d) \times P_0(\mathcal{T}^y \cap \Gamma^h_d)$, which is composed of piecewise constant functions over x and y component grid cells that intersect the Dirichlet boundary. Here we use Γ^h_d to denote the portion of $\partial\Omega^h$ over which the Dirichlet constraint is being enforced. We call any x or y cell T^i with $T^i \cap \partial\Omega^h \neq \emptyset$ a boundary cell. The superscript i is used to denote whether the cell is in the x or y grids with $i = 1$ signifying an x cell and $i = 2$ signifying a y cell. We use $\boldsymbol{w}_{T^i} = \chi_{T^i}(\boldsymbol{x})\boldsymbol{e}_i$ as the basis functions for $\Lambda^h_x \times \Lambda^h_y = P_0(\mathcal{T}^x \cap \Gamma^h_d) \times P_0(\mathcal{T}^y \cap \Gamma^h_d)$. Here, $\chi_{T^i}(\boldsymbol{x})$ is the characteristic function of the cell T^i:

$$\chi_{T^i}(x) = \begin{cases} 1, & \boldsymbol{x} \in T^i \\ 0, & \boldsymbol{x} \notin T^i. \end{cases}$$

Note that we have one basis function per boundary x or y cell. If we use N_{xd} and N_{yd} to denote the number of x and y boundary cells respectively, we can see that the dimension of the space $\Lambda^h_x \times \Lambda^h_y$ is $N_{xd} + N_{yd}$.

With this approximation, the Dirichlet boundary condition constraint can be expressed as a linear system $\mathbf{B}^h \boldsymbol{u}^h = \boldsymbol{u}^h_0$ ($\mathbf{B}^h \in \mathbb{R}^{(N_{xd}+N_{yd}) \times (N_x+N_y)}$, $\boldsymbol{u}^h_0 \in \mathbb{R}^{(N_{xd}+N_{yd})}$) where each equation enforces an integral constraint over the intersection of the discrete boundary Γ^h_d with some x or y boundary cell T^i:

$$\sum_{k_i \in I_i} u_{ik_i} \int_{T^i \cap \Gamma^h_d} N_{ik_i}(\boldsymbol{x}) \, ds = \int_{T^i \cap \Gamma^h_d} u_{0i}(\boldsymbol{x}) \, ds$$

where $\boldsymbol{u}_0 = (u_{01}, u_{02})$. In practice, we evaluate the integral for a given boundary cell T^i over the portion of the Dirichlet boundary curve $T^i \cap \Gamma^h_d$ from the four subquadrilaterals of T^i arising from the doubly refined grid (as discussed in section 3.3). This is simple because in each of these subquadrilaterals Γ^h_d is just a single line segment. We use the following approximation for the right hand side terms in the constraint system:

$$\int_{T^i \cap \Gamma^h_d} u_{0i}(\boldsymbol{x}) \, ds = \sum_{j=1}^4 \int_{\omega^i_j \cap \Gamma^h_d} u_{0i}(\boldsymbol{x}) \, ds \approx \sum_{j=1}^4 u_{0i}(c(\omega^i_j \cap \Gamma^h_d)) \int_{\omega^i_j \cap \Gamma^h_d} 1 \, ds$$

where ω^i_j is one of the four subquadrilaterals of the cell T^i and $c(\omega^i_j \cap \Gamma^h_d)$ is the midpoint of the segment $\omega^i_j \cap \Gamma^h_d$. We use the same treatment for the entries in the matrix on the left hand side:

$$\int_{T^i \cap \Gamma^h_d} N_{ik_i}(\boldsymbol{x}) \, ds = \sum_{j=1}^4 \int_{\omega^i_j \cap \Gamma^h_d} N_{ik_i}(\boldsymbol{x}) \, ds.$$

We note that the integrand here is simply an x or y bilinear interpolating function. Therefore, the four terms in the sum can be evaluated analytically because they are simply quadratics in any linear parameterization of a given boundary segment $\omega^i_j \cap \Gamma^h_d$.

The discrete constrained minimization problem can be solved using a Lagrange multiplier method resulting in the following KKT system:

$$\begin{pmatrix} \mathbf{L}_u^h & \mathbf{G}^{h^T} & \mathbf{B}^{h^T} \\ \mathbf{G}^h & \mathbf{D}_p^h & \mathbf{0} \\ \mathbf{B}^h & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^h \\ \boldsymbol{p}^h \\ \boldsymbol{\lambda}^h \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^h \\ \mathbf{0} \\ \boldsymbol{u}_0^h \end{pmatrix}. \tag{26}$$

There is one Lagrange multiplier degree of freedom per Dirichlet constraint. In other words, $\boldsymbol{\lambda}^h$ is in $\mathbb{R}^{(N_{xd}+N_{yd})}$. When we consider boundary equations in the sections that follow, we temporarily eliminate pressure variables $\boldsymbol{p}^h$ with the following substitution $\mathbf{L}^h = \mathbf{L}_u^h - \mathbf{G}^{h^T}(\mathbf{D}_p^h)^{-1}\mathbf{G}^h$:

$$\begin{pmatrix} \mathbf{L}^h & \mathbf{B}^{h^T} \\ \mathbf{B}^h & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^h \\ \boldsymbol{\lambda}^h \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^h \\ \boldsymbol{u}_0^h \end{pmatrix}. \tag{27}$$

This system is extremely ill-conditioned for nearly incompressible materials, however it will simplify the exposition of the forthcoming discussion of Dirichlet boundary condition treatment. Furthermore, when performing equation relaxation in our multigrid solver, we temporarily perform this elimination when treating equations near the boundary of the domain. Before discussing our geometric multigrid solution approach for these systems of equations, we would like to first discuss some important aspects of the Dirichlet system. Our treatment of the Dirichlet condition is somewhat nonstandard and we list here a few important details related to the constraint matrix $\mathbf{B}^h$.

1. $\mathbf{B}^h$ consists of two decoupled blocks. There is one block for the x boundary equations and one for the y equations. The only non-zero columns of $\mathbf{B}^h$ are associated with nodes that are incident on an x or y boundary cell. Therefore, for sufficiently interior degrees of freedom, the KKT system is exactly the same as (21) or (37). That is, although the constraint matrix is in $\mathbb{R}^{(N_{xd}+N_{yd})\times(N_x+N_y)}$, it really only acts on a small subset of the $N_x + N_y$ displacement degrees of freedom.
2. $\mathbf{B}^h \in \mathbb{R}^{(N_{xd}+N_{yd})\times(N_x+N_y)}$, where $(N_{xd} + N_{yd}) < (N_x + N_y)$. In fact,

$$\mathbf{B}^h = \begin{pmatrix} \mathbf{B}^x & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^y \end{pmatrix},$$

   where $\mathbf{B}^x \in \mathbb{R}^{N_{xd}\times N_x}$ and $\mathbf{B}^y \in \mathbb{R}^{N_{yd}\times N_y}$. However, it can be shown that $\mathbf{B}^h$ has full row rank. We refer the reader to the work [65] for a more detailed discussion of why this is so.
3. Our numerical linear algebra approach to the problem is based on the construction of a (full column rank) matrix $\mathbf{Z}^h \in \mathbb{R}^{(N_x+N_y)\times((N_x+N_y)-(N_{xd}+N_{yd}))}$ whose columns span the kernel space of $\mathbf{B}^h$. Since $\mathbf{B}^h$ has full row rank, there exists a column permutation $\mathcal{P}$, such that $\mathbf{B}^h\mathcal{P} = [\mathbf{B}_m|\mathbf{B}_{n-m}]$, where $\mathbf{B}_m \in \mathbb{R}^{(N_{xd}+N_{yd})\times(N_{xd}+N_{yd})}$ is non-singular and $\mathbf{B}_{m-n} \in \mathbb{R}^{(N_{xd}+N_{yd})\times((N_x+N_y)-(N_{xd}+N_{yd}))}$. With this permutation, we can construct a so called fundamental basis for the null-space of $\mathbf{B}^h\mathcal{P}$:

$$\mathbf{Z}^h = \begin{bmatrix} -\mathbf{B}_m^{-1}\mathbf{B}_{n-m} \\ \mathbf{I} \end{bmatrix}. \tag{28}$$

4. The vector

$$\boldsymbol{c}^h = \begin{bmatrix} \mathbf{B}_m^{-1}\boldsymbol{u}_0^h \\ 0 \end{bmatrix} \tag{29}$$

   satisfies $\mathbf{B}^h\mathcal{P}\boldsymbol{c}^h = \boldsymbol{u}_0^h$. Therefore, all solutions can be expressed as $\boldsymbol{u}^h = \mathcal{P}(\boldsymbol{c}^h + \mathbf{Z}^h\boldsymbol{v}^h)$ with $\mathbf{v}^h \in \mathbb{R}^{((N_x+N_y)-(N_{xd}+N_{yd}))}$.
5. Our construction of a null-space for the Dirichlet constraint allows us to eliminate the Lagrange multipliers. Substituting $\boldsymbol{u}^h = \mathcal{P}(\boldsymbol{c}^h + \mathbf{Z}^h\boldsymbol{v}^h)$ in (27), we have

$$\mathbf{L}^h\mathcal{P}(\boldsymbol{c}^h + \mathbf{Z}^h\boldsymbol{v}^h) + \mathbf{B}^{h^T}\boldsymbol{\lambda}^h = \boldsymbol{f}^h$$

   Left multiplying this equation with $(\mathcal{P}\mathbf{Z}^h)^T$, and applying the property $\mathbf{B}^h\mathcal{P}\mathbf{Z}^h = 0$, we have

$$(\mathcal{P}\mathbf{Z}^h)^T\mathbf{L}^h\mathcal{P}\mathbf{Z}^h\boldsymbol{v}^h + (\mathbf{B}^h\mathcal{P}\mathbf{Z}^h)^T\boldsymbol{\lambda}^h = (\mathcal{P}\mathbf{Z}^h)^T(\boldsymbol{f}^h - \mathbf{L}^h\mathcal{P}\boldsymbol{c}^h)$$

$$(\mathbf{Z}^{h^T}\mathcal{P}^T\mathbf{L}^h\mathcal{P}\mathbf{Z}^h)\boldsymbol{v}^h = (\mathcal{P}\mathbf{Z}^h)^T(\boldsymbol{f}^h - \mathbf{L}^h\mathcal{P}\boldsymbol{c}^h). \tag{30}$$

That is, if we can solve $\boldsymbol{v}^h$ from the reduced system (30), then the KKT system solution can be reconstructed with $\boldsymbol{u}^h = \mathcal{P}(\boldsymbol{c}^h + \mathbf{Z}^h\boldsymbol{v}^h)$. Without loss of generality, we assume the variables to be reordered such that $\mathcal{P} = \mathcal{I}$. We will make use of this property in the smoother for our geometric multigrid method.

4.2. Constructing the null-space for the Dirichlet constraints

Our treatment of the Dirichlet conditions is based on our ability to construct a null-space $\mathbf{Z}^h$ satisfying $\mathbf{B}^h\mathbf{Z}^h = 0$ and a special solution $\boldsymbol{c}^h$ satisfying $\mathbf{B}^h\boldsymbol{c}^h = \boldsymbol{u}_0^h$. The main issue we will discuss now is how to find a fundamental basis $\mathbf{Z}^h$ that produces a numerically well-conditioned system of equations. This topic was originally discussed in the work of Bedrossian et al., [65]. However, we found that in our case of nearly incompressible materials, an even more aggressive approach is needed. Bedrossian et al. first suggested an ordering for the boundary integral equations and incident boundary nodes that led to a readily inverted upper triangular $\mathbf{B}_m$. However, they showed that although this construction was straightforward, the conditioning of this $\mathbf{B}_m$ deteriorated exponentially in the discretization resolution and was thus not practical. They then showed that it is possible to derive a diagonal $\mathbf{B}_m$ with a slight modification to the definition of the constraints. Specifically, they showed that an aggregation scheme where cells of a "double-wide" grid were used to define the extent of the line integral constraints led to a diagonal $\mathbf{B}_m$. This was done by choosing the center node of the 9 nodes incident on the four original cells in the "double-wide" cell as the representative node in the permutation of columns in $\mathbf{B}^h$. In other words, the center node of the four aggregated cells was given the same index as the row associated with the constraint over those cells. This aggregation of cells is equivalent to replacing a collection of rows in the original "single-wide" $\mathbf{B}^h$ with the sum of the collection of rows. The choice of which rows to sum together is determined by the "double-wide" cell they belong to. Since no aggregated rows have a non-zero entry associated with the center node of any other aggregate, the $\mathbf{B}_m$ is diagonal and thus $\mathbf{Z}^h$ is trivially constructed. This aggregation can equivalently be seen as using a subspace $\Lambda_x^{2h} \times \Lambda_y^{2h} \cap H^{-1/2}(\Gamma_d)$ that is based on a coarsened grid. Remarkably, this process does not affect the L^∞ convergence behavior of the scheme. Unfortunately, while the reduced system in [65] had a satisfactory condition number for the Poisson equation with an incomplete Choleksy preconditioned conjugate gradient solver, we found that this was not the case for nearly incompressible linear elasticity and geometric multigrid. Specifically, the conditioning of the equations at the boundary was poor enough to make the treatment of boundary regions prohibitively costly when performing the smoothing operations used in geometric multigrid.

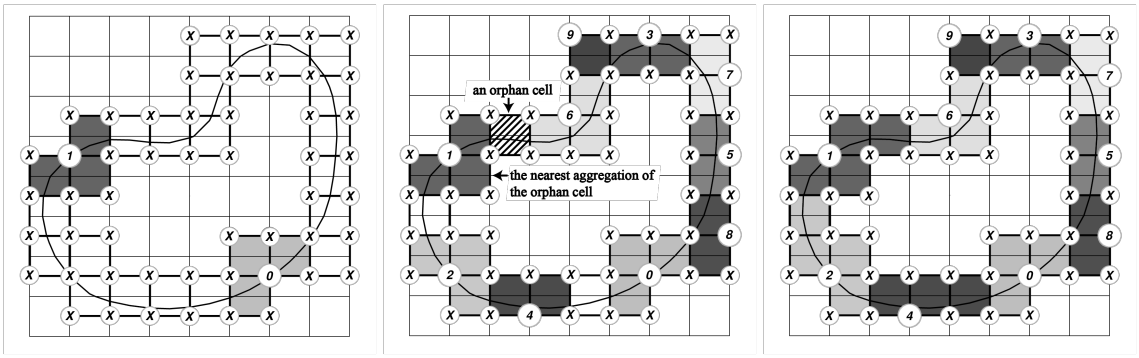
We propose a new boundary integral constraint aggregation that allows us to generate a diagonal $\mathbf{B}_m$ and a better-conditioned reduced system. We note that the elements of the original $\mathbf{B}^h$ are scaled by a segment length; therefore, a very small segment (associated with a node that is far from the boundary) will generate very small diagonal elements for a $\mathbf{B}_m$ arising from aggregation. We found this to be a source of the suboptimal conditioning. We propose a modification to the aggregation process that is designed to provide larger diagonal entries in subsequent $\mathbf{B}_m$. We first define the weight of each node to be the sum of the column in the original "single-wide" $\mathbf{B}^h$. This weight is the diagonal entry in $\mathbf{B}_m$ that would arise from an aggregation of the four cells centered around the node. To increase the diagonal entries in the final $\mathbf{B}_m$, we simply select four cell aggregates such that the associated representative nodes will be chosen in descending order of the total weight of each node. After we choose a node to define a four cell aggregate, the nine nodes incident on the four cells are eliminated from consideration. This process of prioritizing the aggregation rather than inheriting it from the coarse grid has a drawback. There may be some rows that are never added to an aggregate region. For these rows, it may be impossible to choose a representative node. We resolve this by aggregating such a row into a spatially adjacent aggregate. In practice, we also found that limiting representative nodes in the aggregation to ghost nodes (i.e. those geometrically outside the domain Ω) gave even better conditioning. We briefly summarize this process in Algorithm 2.

5. Multigrid

We develop an efficient multigrid solver for the discrete systems produced by our method. Our method is purely geometric, and based on the Multigrid Correction Scheme (see Algorithm 3). The framework admits a simple implementation, however special care is needed to retain near-textbook multigrid convergence rates, especially in the

Algorithm 2 Aggregation Selection

procedure AGGREGATIONSELECTION($\mathbf{B}^x$, $\mathbf{B}^y$)
  for v **in** $\{1, 2\}$ **do**
    aggregation cells list $\mathcal{A}ggr^v = \{\}$
    aggregation representative list $\mathcal{R}ep^v = \{\}$
    set all active node^v variables and all integral cells^v active
    row weight sum $w_i \leftarrow \sum_j \mathbf{B}^v_{ij}$
    Sort w in a decreasing order
    for w_j **in** w **do**
      if node^v_j is active **then**
        $\mathcal{R}ep^v \leftarrow \mathcal{R}ep^v \cup \{j\}$
        $\mathcal{A}ggr^v \leftarrow \mathcal{A}ggr^v \cup \{c$ for all cell$_c$ adjacent to node$^v_j\}$
        deactivate all cells adjacent to node^v_j
        deactivate all nodes adjacent to node^v_j
      end if
    end for ▷ Now every ghost node belongs to at least one aggregation
    for all boundary cell^v_i **do**
      if cell$_i$ is inactive **then**
        Find the closest aggregation of cell^v_i $\mathcal{A}ggr^v_k$
        $\mathcal{A}ggr^v_k \leftarrow \mathcal{A}ggr^v_k \cup \{i\}$
      end if
    end for ▷ Pickup orphans
  end for
end procedure



(a) (b) (c)

Figure 6: Cell aggregations for x nodes with representative nodes denoted by their number. a) x component boundary cells and the first two representative nodes and the incident $x-$ cells of each representative node; b) all representative nodes for x component cells and their incident cells, orphan cells will be attached to their nearest neighbor cells; c) final cell aggregations together with their representative nodes for the x grid.

presence of highly irregular domains or nearly incompressible materials. The sections that follow will detail the key components of our multigrid solver: a hierarchy of discretizations, a smoothing procedure, and appropriate transfer operators (i.e. restriction and prolongation) between levels of the hierarchy. Although our design decisions include certain common practices, these components have been significantly customized to fit the needs of the specific discretization being followed, and facilitate both convergence and computational efficiency even near the incompressible limit.

Algorithm 3 Multigrid defect correction

procedure V-CYCLE($\hat{\mathbf{L}}^h$,$\hat{\boldsymbol{u}}^h$,$\hat{\boldsymbol{f}}^h$)
  if problem at low resolution and easy to solve **then**
    $\hat{\boldsymbol{u}}^h \leftarrow (\hat{\mathbf{L}}^h)^{-1}\hat{\boldsymbol{f}}^h$ and return;
  end if
  PreRelaxation($\hat{\mathbf{L}}^h$,$\hat{\boldsymbol{u}}^h$,$\hat{\boldsymbol{f}}^h$)
  Restriction: $\hat{\boldsymbol{f}}^{2h} \leftarrow \mathbf{R}(\hat{\boldsymbol{f}}^h - \hat{\mathbf{L}}^h\hat{\boldsymbol{u}}^h)$
  V-Cycle($\hat{\mathbf{L}}^{2h}$,$\hat{\boldsymbol{u}}^{2h}$,$\hat{\boldsymbol{f}}^{2h}$)
  Prolongation: $\hat{\boldsymbol{u}}^h \leftarrow \hat{\boldsymbol{u}}^h + \mathbf{P}\hat{\boldsymbol{u}}^{2h}$
  PostRelaxation($\hat{\mathbf{L}}^h$,$\hat{\boldsymbol{u}}^h$,$\hat{\boldsymbol{f}}^h$)
end procedure

5.1. Discretization hierarchy

We consider a hierarchy of resolutions, each corresponding to a discretization on a progressively larger grid size. In particular, we employ a grid step of h on the finest level of the hierarchy (numbered as level zero), followed by discretizations with grid step sizes of $2h, 4h, \ldots, 2^L h$, for a total of $L+1$ hierarchy levels. In detail, the hierarchy is constructed as follows:

- At every level of the hierarchy, say the l-th one, we define the background grids $\mathcal{G}^x_{2^l h}, \mathcal{G}^y_{2^l h}, \mathcal{G}^p_{2^l h}$ corresponding to the x-, y-, and p-variables respectively.
- A level set function is computed over the respective doubly-refined subgrids $\mathcal{G}^\phi_h, \mathcal{G}^\phi_{2h}, \mathcal{G}^\phi_{4h}, \ldots$ for each level. Obviously, coarser levels may fail to resolve certain high-frequency features of the domain geometry, leading to possible discrepancies between the discrete systems at various levels, which will be further addressed in our discussion of the smoother and transfer operators.
- Using the level set values associated with a given grid, we generate the discrete domains $\mathcal{T}^x_{2^l h}, \mathcal{T}^y_{2^l h}, \mathcal{T}^p_{2^l h}$, and allocate the unknown arrays $\boldsymbol{u}^{2^l h}$ and $\boldsymbol{p}^{2^l h}$ as well as the right-hand sides $\boldsymbol{f}^{2^l h}$ and $\boldsymbol{f}_p^{2^l h}$ of the respective equations. The discrete operators $\mathbf{L}_u^{2^l h}$, $\mathbf{G}^{2^l h}$ and $\mathbf{D}_p^{2^l h}$ of the system (20) are likewise defined on the discrete domain associated with the l-th level of the hierarchy, following the same process detailed in section 3.1. Note that, although at the finest level of the hierarchy we have used $\boldsymbol{f}_p^h = 0$ by virtue of our discretization, the right hand side $\boldsymbol{f}_p^{2^l h}$ for coarser levels ($l \geq 1$) will generally be nonzero in the Multigrid Correction Scheme (see Algorithm 3).

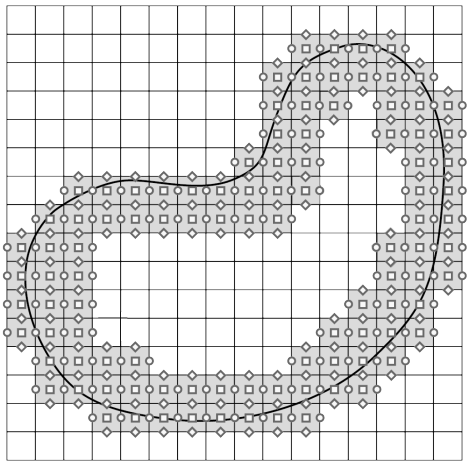
From this point on, we will simply use h instead of $2^l h$ to denote the grid spacing at any specific level of the multigrid hierarchy, whenever this does not incur any ambiguity. When there is a Dirichlet boundary condition, a constraint matrix $\mathbf{B}^h$ is defined for each level, enforcing the integral of x or y displacement components along the discretized domain boundary $\partial\Omega^h$ of the current level and within each cell aggregation (each cell group with the same color in Figure 6) precomputed on the same level, to be the same as that of the Dirichlet values for the corresponding displacement component. Each of these constraints (or cell aggregations) corresponds to one Lagrange multiplier. In the fundamental basis method, we eliminate these multipliers by solving for the fundamental basis coefficients $\boldsymbol{v}^h$ in $\boldsymbol{c}^h + \mathbf{Z}^h\boldsymbol{v}^h$. By definition of $\boldsymbol{c}^h$ (29) and $\mathbf{Z}^h$ (28), there is a one-to-one mapping between $\boldsymbol{v}^h$ components and active x and y degrees of freedom that are not aggregation representatives. Thus, the reduced system (30) is defined on these degrees of freedom only. Due to the fact that the reconstructed $\boldsymbol{u}^h = \boldsymbol{c}^h + \mathbf{Z}^h\boldsymbol{v}^h$ satisfies the constraints automatically, we do not need to restrict any residual for the constraint system, i.e. on coarser levels, the Dirichlet boundary constraint values are always zero. Also, $\boldsymbol{\lambda}^h$ do not need to be solved, therefore, we do not need to record their values, nor prolongate their corrections. When we restrict the residuals of the governing equation, i.e. $\boldsymbol{r}^h = \boldsymbol{f}^h - \mathbf{L}^h\boldsymbol{u}^h - \mathbf{B}^{\mathbf{h}^T}\boldsymbol{\lambda}^h$, we restrict zero for all equations that will need a $\boldsymbol{\lambda}^h$ value. In other words, we restrict zero residuals from equations involving boundary nodes, i.e. the nodes in Figure 6 cells. Although omitting these equations from the inter-grid

transfers is a deviation from conventional practice, we compensate by moderately increasing the smoothing effort in the boundary band, effectively driving the residuals closer to zero (which is the value that is actually restricted). This approach avoids the use of specialized, elaborate transfer operators between the $\boldsymbol{\lambda}$ variables, which are not in perfect correspondence across levels due to the potentially different aggregations employed at each level. Thus, a single level discretization is defined for all levels with powers of 2 resolutions.

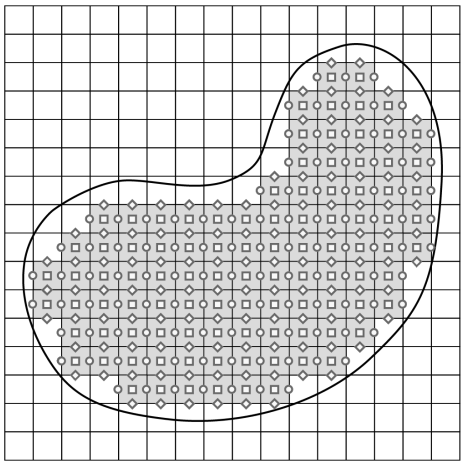
5.2. Relaxation

The interior equations are uniform and have the same properties, while near the boundary, the equations have very different stencils. In order to design a stable and efficient relaxation while keeping the computational cost low, we define two (overlapping) sets of equations, and apply an appropriate relaxation scheme to each one. The two sets correspond to equations in the interior of the discrete domain, and equations near the boundary, respectively. We define the extent of the interior region by excluding a 5×5 block of cells, centered around any cell that is either entirely exterior to the domain, or intersects a Dirichlet boundary. See Figure 7, right, for an example using a single cell block. The interior region is relaxed with the distributive process detailed in section 5.2.1.

We then define the boundary band to be the union of all 7×7 blocks of cells centered at each cell that intersects the Dirichlet boundary and then removing all the non-active cells (i.e. cells that are completely exterior to the domain). This defines the set of equations to which we will apply a boundary relaxation. In each single level relaxation, we first sweep over the boundary band, and apply a few iterations of boundary relaxations, then apply one iteration of interior relaxation followed by another few iterations of boundary relaxations. In Figure 7, left, we show an example of the cells and equations that end up in a boundary region calculated with the method described above, but using a 3×3 box rather than the 7×7 box used in our simulations.



(a) An example of boundary band pressure cells and boundary variables using a 3×3 box centered at each cell that contains the boundary.



(b) An example of distributive pressure cells and variables relaxed using distributive relaxation. The region is defined by excluding a single cell box centered at each cell containing the boundary.

Figure 7: Boundary band and distributive region.

The efficiency of a multigrid method is closely related to the smoothing efficiency of a single level relaxation. With Poisson's equation, simple Jacobi or Gauss-Seidel will typically suffice as an efficient smoother. These techniques efficiently reduce the high-frequency component of the error and make it possible for a coarse grid to provide a meaningful correction to a finer grid. This property is fundamentally important for the efficiency of the geometrically hierarchical approach to solving the equations. Unfortunately, the equations of nearly incompressible linear elasticity with augmented pressure require more care than the comparably simplistic discrete Poisson equation. Although our system is not symmetric positive definite, we can modify the equations to a more convenient form as in [1] to

design a proper geometric multigrid smoother. We confirm that a change of variables leads to an approximate block triangularization of the discrete system with each diagonal block being a symmetric semi-definite discretization of the Laplacian. Our smoother is then constructed to be an emulation of the Gauss-Seidel relaxation applied on each block.

5.2.1. Approximated distributive relaxation

We follow the idea in [1] and develop a distributive relaxation. At the continuous PDE level, we apply a change of variable:

$$\begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{I} & -\nabla \\ \nabla^T & -2\Delta \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ q \end{pmatrix} \qquad \text{or} \qquad \hat{\boldsymbol{u}} = \hat{\mathbf{M}}\hat{\boldsymbol{v}} \tag{31}$$

and substitute into (7) to achieve a new system

$$\begin{pmatrix} \mu\Delta\mathbf{I} & 0 \\ \mu(1+\frac{\mu}{\lambda})\nabla^T & -\mu(1+\frac{2\mu}{\lambda})\Delta \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ q \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} \qquad \text{or} \qquad \hat{\mathbf{L}}\hat{\mathbf{M}}\hat{\boldsymbol{v}} = \hat{\boldsymbol{f}} \tag{32}$$

for some auxiliary variable $\hat{\boldsymbol{v}} = (\boldsymbol{v}, q)$. The derived PDE system is a block lower triangular system, and can be solved in a forward substitution process, i.e. first solve the $\boldsymbol{v}$ equations, and then freeze the $\boldsymbol{v}$ variables in the second equation and solve the q equation. Moreover, with a certain choice of discretizations, the same triangulation can be realized on the discretized system, i.e. $\hat{\mathbf{L}}^h\hat{\mathbf{M}}^h$ is also a block lower triangular linear system [1].

Due to the fact that each of the diagonal blocks of the discrete auxiliary system is a discretization of the Laplacian operator, we can relax the whole system using a Gauss-Seidel relaxation on each component of the $\boldsymbol{v}$ variables followed by another Gauss-Seidel relaxation on the q variables and achieve the same smoothing efficiency as that of the Gauss-Seidel relaxation applied on Poisson's equation. At any approximation of $\boldsymbol{v}$ and q, the approximate solutions to the augmented system can be reconstructed using (31).

In practice, we do not need to explicitly construct $\hat{\boldsymbol{v}}$. In a Gauss-Seidel relaxation applied on $\hat{\boldsymbol{v}}$, we iteratively solve for local corrections $\hat{v}_i \leftarrow \hat{v}_i + \delta e_i$, such that the local residual $(\hat{\boldsymbol{f}} - \hat{\mathbf{L}}\hat{\mathbf{M}}\hat{\boldsymbol{v}})_i$ is zeroed out. Therefore, $\delta = (\hat{\mathbf{L}}\hat{\mathbf{M}})_{ii}^{-1}\hat{r}_i$. Such corrections invoke local corrections to $\hat{\boldsymbol{u}}$ in a distributive pattern, i.e. $\hat{u}_i \leftarrow \hat{u}_i + \delta\hat{\mathbf{M}}e_i$, thus defines the distributive relaxation scheme in Algorithm 4.

Algorithm 4 Distributive Smoothing

```
1: procedure DistributiveSmoothing(L̂^h, M̂^h, û^h, f̂^h)
2:     for v in {u_1, u_2, p} do                    ▷ Must iterate on u_1 and u_2 before p
3:         for i in Lattice[v] do                   ▷ i is an equation index
4:             r ← f̂^h_i − L̂^h_i · û^h              ▷ L̂^h_i is the i-th row of L̂^h
5:             δ ← r/(L̂^h M̂^h)_ii                  ▷ (L̂^h M̂^h)_ii is a precomputed constant for each component
6:             û^h += δ m_i^T                       ▷ m_i is the i-th row of M̂^h
7:         end for
8:     end for
9: end procedure
```

For a staggered finite difference discretization, the triangularization of the discretized system can be achieved by discretizing the change of variable operator using centered differences for the gradient and divergence operators and a five point stencil for the Laplacian operator as shown in [1]. However, when we use a finite element discretization, there is no discrete change of variables with same sparsity that leads to an exact triangularization. Instead, we discretize the gradient operator in (31) using the stencils derived in a finite element method, i.e. $\nabla^h = \frac{1}{\mu h^2}\mathbf{G}^{hT} = \begin{pmatrix} D_x^h \\ D_y^h \end{pmatrix}$ mapping from p variables to x and y variables with the locations illustrated in Figure 4-right and the stencils being:

$$D_x^h = \frac{1}{h}\begin{bmatrix} -1/8 & 1/8 \\ -3/4 & 3/4 \\ -1/8 & 1/8 \end{bmatrix} \qquad D_y^h = \frac{1}{h}\begin{bmatrix} 1/8 & 3/4 & 1/8 \\ -1/8 & -3/4 & -1/8 \end{bmatrix}. \tag{33}$$

Similarly, the Laplacian operator in (31) is discretized from a standard piecewise bi-linear finite element discretization.

$$M_p^h = \frac{1}{h^2} \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & -8/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \tag{34}$$

Although, the linear system $\hat{\mathbf{L}}^h\hat{\mathbf{M}}^h$ is not block triangular, our numerical results show that the derived distributive relaxation is able to reduce the high-frequency error components efficiently. Results of using this relaxation scheme will be shown in section 6.

5.2.2. Higher order defect correction

We also adopt the idea of higher order defect correction, which will allow us to develop a less expensive distributive relaxation. In a defect correction scheme for an arbitrary linear system $\mathbf{L}\boldsymbol{u} = \boldsymbol{f}$, we solve for the correction $\delta\boldsymbol{u} = \boldsymbol{u}^{\text{exact}} - \boldsymbol{u}$, which satisfies an equation $\mathbf{L}\delta\boldsymbol{u} = \boldsymbol{f} - \mathbf{L}\boldsymbol{u}$. In practice, we approximate the equation with another system $\mathbf{L}^{\text{approx}}\delta\boldsymbol{u} = \boldsymbol{f} - \mathbf{L}\boldsymbol{u}$ that is easier to solve. For example, in a multigrid correction scheme, a coarse grid system is used as the approximated equation for solving the correction at the fine grid resolution, i.e. $\mathbf{L} = \mathbf{L}^{\text{fine}}$, $\mathbf{L}^{\text{approx}} = \mathbf{L}^{\text{coarse}}$. In the high order defect correction scheme, a lower order discretization is employed as the approximated system for solving a higher order discretization correction. In our case, the finite difference discretization is a lower order system, and the finite element discretization is a high order system, i.e. $\mathbf{L} = \mathbf{L}^{\text{fem}}$, $\mathbf{L}^{\text{approx}} = \mathbf{L}^{\text{fdm}}$. In other words, we consider solving the following correction equation:

$$\hat{\mathbf{L}}^{fd,h}\delta\hat{\boldsymbol{u}} = \hat{\boldsymbol{f}} - \hat{\mathbf{L}}^{fe,h}\hat{\boldsymbol{u}}.$$

In our case, the lower order operator $\hat{\mathbf{L}}^{fd,h}$ is the staggered finite difference scheme detailed in [1]. Note that this operator is only lower order near the boundary, yet second order in the interior; however, we still use this finite difference scheme as the lower order operator in the defect correction process, even when we only focus on the domain interior. One of the benefits we obtain from such an approximation is that we can use the existing distributive relaxation with the exact triangulation of the discretized system (32). To be specific, let us rewrite the finite difference system as

$$\hat{\mathbf{L}}^{fd,h}\hat{\boldsymbol{u}} = \begin{pmatrix} \mathbf{L}_u^{fd,h} & \mathbf{G}^{fd,h^T} \\ \mathbf{G}^{fd,h} & \mathbf{D}_p^{fd,h} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{fd,h} \\ \boldsymbol{p}^{fd,h} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^{fd,h} \\ \mathbf{0} \end{pmatrix} \tag{35}$$

and rewrite the finite element system scaled by $1/h^2$ to match the scaling of the differential equation

$$\frac{1}{h^2}\hat{\mathbf{L}}^{fe,h}\hat{\boldsymbol{u}} = \frac{1}{h^2}\begin{pmatrix} \mathbf{L}_u^{fe,h} & \mathbf{G}^{fe,h^T} \\ \mathbf{G}^{fe,h} & \mathbf{D}_p^{fe,h} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{fe,h} \\ \boldsymbol{p}^{fe,h} \end{pmatrix} = \frac{1}{h^2}\begin{pmatrix} \boldsymbol{f}^{fe,h} \\ \mathbf{0} \end{pmatrix} \tag{36}$$

In a high order defect correction scheme employing a finite difference discretization, we solve for a correction $\delta\hat{\boldsymbol{u}} = \hat{\mathbf{M}}^{fd,h}\delta\hat{\boldsymbol{v}}$ to locally satisfy

$$\hat{\mathbf{L}}^{fd,h}\hat{\mathbf{M}}^{fd,h}\delta\hat{\boldsymbol{v}} = \frac{1}{h^2}(\hat{\boldsymbol{f}}^{fe,h} - \hat{\mathbf{L}}^{fe,h}\hat{\boldsymbol{u}}^{\text{current}})$$

This derives a sparser distributive relaxation, shown in Algorithm 5.

The previous two types of distributive relaxation are not applicable for variables near the domain boundary. In fact, near the boundary, some of the variables in the distribution stencil may not exist. We follow the idea in [1], and temporarily build an unaugmented system in the boundary band (see Figure 7, left).

5.3. Boundary relaxation

Near the domain boundary, the previous distributive relaxation is not well defined; a special relaxation is required. In the Neumann boundary condition case, we eliminate p from the augmented system (20) by left multiplying the equation with

$$\hat{\mathbf{U}} = \begin{pmatrix} \mathbf{I} & -\mathbf{G}^T\mathbf{D}_p^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}.$$

Algorithm 5 High Order Defect Correction Distributive Smoothing

procedure HIGHORDERDEFECTCORRECTIONDISTRIBUTIVESMOOTHING($\hat{\mathbf{L}}^{fd}$,$\hat{\mathbf{M}}^{fd}$,$\hat{\mathbf{L}}^{fe}$,$\hat{\mathbf{M}}^{fe}$,$\hat{\boldsymbol{u}}$,$\hat{\boldsymbol{f}}^{fe,h}$)
  for v **in** $\{u_1, u_2, p\}$ **do** ▷ Must iterate on u_1 and u_2 before p
    for i **in** Lattice[v] **do** ▷ i is an equation index
      $r \leftarrow \hat{\boldsymbol{f}}_i^{fe,h} - \hat{\mathbf{L}}_i^{fe} \cdot \hat{\boldsymbol{u}}$ ▷ $\hat{\mathbf{L}}_i^{fe}$ is the i-th row of $\hat{\mathbf{L}}^{fe}$
      $\delta \leftarrow r/(\hat{\mathbf{L}}^{fd}\hat{\mathbf{M}}^{fd})_{ii}$ ▷ $(\hat{\mathbf{L}}^{fd}\hat{\mathbf{M}}^{fd})_{ii}$ is a precomputed constant for each component
      $\hat{\boldsymbol{u}}$ += δm_i^T ▷ m_i is the i-th row of $\hat{\mathbf{M}}^{fd}$
    end for
  end for
end procedure

Therefore,

$$\hat{\mathbf{U}}\hat{\mathbf{L}}\hat{\boldsymbol{u}} = \begin{pmatrix} \mathbf{L}_u - \mathbf{G}^T\mathbf{D}_p^{-1}\mathbf{G} & \mathbf{0} \\ \mathbf{G} & \mathbf{D}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \hat{\mathbf{U}}\hat{\boldsymbol{f}}. \tag{37}$$

In the first equation for $\boldsymbol{u}$, the equation is symmetric and positive definite, and hence can be solved again using Gauss-Seidel relaxation. This unaugmented system is a consistent discretization to the original PDE (7). Although Gauss-Seidel relaxation is not an efficient smoother for the unaugmented system if defined everywhere, for the purpose of boundary relaxation, we only build the temporary unaugmented system and relax it within a very narrow boundary band as demonstrated in Figure 7, and temporarily freeze the interior variables. The solution is strongly restricted by nearby interior values, therefore the Gauss-Seidel relaxation is still efficient and stable. Typically, with about 5 to 10 sweeps of boundary relaxation before and after each interior relaxation sweep, the boundary residual is reduced to as small as the interior residual. Once we relaxed $\boldsymbol{u}$ well enough, we freeze $\boldsymbol{u}$ and substitute into the second equation in (37) to resolve pressure variables.

5.4. Boundary relaxation for the reduced system in Dirichlet boundary condition case

In the Dirichlet boundary condition case, the boundary system (27) is a KKT system, which is indefinite, and cannot be resolved using Gauss-Seidel relaxations. Alternative approaches such as Kaczmarz relaxation or box relaxation may be efficient smoothers, however, their computational cost is much more expensive. Instead, we follow the fundamental basis method, to solve $\boldsymbol{v}^h$ from the reduced system (30), and reconstruct the solution of the KKT system (27) using $\boldsymbol{u}^h = \boldsymbol{c}^h + \mathbf{Z}^h\boldsymbol{v}^h$. Since $\mathbf{L}^h$ is symmetric positive definite, $\mathbf{Z}^{h^T}\mathbf{L}^h\mathbf{Z}^h$ is also symmetric positive definite, and hence can be solved using Gauss-Seidel relaxation (see Algorithm 6).

In practice, a Gauss-Seidel iteration on (30) iteratively solves for a correction on each single degree of freedom by solving the following scalar equation

$$\boldsymbol{e}_i^T\mathbf{L}_r^h(\boldsymbol{v}^h + \delta\boldsymbol{e}_i) = \boldsymbol{e}_i^T\mathbf{Z}^{h^T}(\boldsymbol{f}^h - \mathbf{L}^h\boldsymbol{c}^h), \tag{38}$$

where $\mathbf{L}_r^h = \mathbf{Z}^{h^T}\mathbf{L}^h\mathbf{Z}^h$ i.e.

$$(\mathbf{L}_r^h)_{ii}\delta = \boldsymbol{e}_i^T\mathbf{Z}^{h^T}(\boldsymbol{f}^h - \mathbf{L}^h\boldsymbol{c}^h - \mathbf{L}^h\mathbf{Z}^h\boldsymbol{v}^h) = \boldsymbol{e}_i^T\mathbf{Z}^{h^T}(\boldsymbol{f}^h - \mathbf{L}^h\boldsymbol{u}^h) \tag{39}$$

and then applies the correction: $\boldsymbol{v}^h \leftarrow \boldsymbol{v}^h + \delta\boldsymbol{e}_i$. Equivalently, $\boldsymbol{u}^h$ is updated as $\boldsymbol{u}^h \leftarrow \boldsymbol{u}^h + \delta\mathbf{Z}^h\boldsymbol{e}_i$. Therefore we can equivalently solve for a correction on $\boldsymbol{u}$ to emulate the Gauss-Seidel iteration on $\boldsymbol{v}^h$ (see Algorithm 7).

5.5. Coarsening

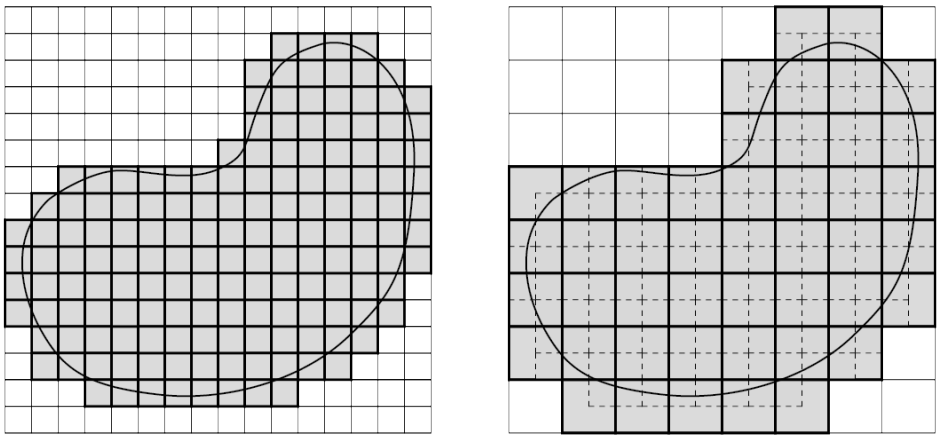
In a geometric multigrid method, we define a discretization on each level. On the interior of the region, we restrict residuals from the fine grid to coarse grid by applying a restriction operator $\mathbf{R}$ for each component defined on the staggered grids with stencils illustrated in Figure 9 . We consider two types of prolongation stencils. First, we consider prolongation $\mathbf{P}_{lo} = 4\mathbf{R}^T$. Second, we consider piecewise bi-linear interpolation for $\boldsymbol{u}$ in combination with the same pressure prolongation as in $\mathbf{P}_{lo}$, which we donate as $\mathbf{P}_{hi}$.

Algorithm 6 Dirichlet boundary relaxation - $\boldsymbol{v}^h$

$\boldsymbol{v}^h \leftarrow \mathbf{0}$
for $i = 1$ to m **do**
    $\delta \leftarrow \boldsymbol{e}_i^T \mathbf{Z}^{h^T}(\boldsymbol{f}^h - \mathbf{L}^h \boldsymbol{c}^h - \mathbf{L}^h \mathbf{Z}^h \boldsymbol{v}^h)/(\mathbf{L}^h)_{ii}$
    $\boldsymbol{v}^h$ += $\delta \boldsymbol{e}_i$
end for

Algorithm 7 Dirichlet boundary relaxation - $\boldsymbol{u}^h$

$\boldsymbol{u}^h \leftarrow \boldsymbol{c}^h$
for $i = 1$ to m **do**
    $\delta \leftarrow \boldsymbol{e}_i^T \mathbf{Z}^{h^T}(\boldsymbol{f}^h - \mathbf{L}^h \boldsymbol{u}^h)/(\mathbf{L}^h)_{ii}$
    $\boldsymbol{u}^h$ += $\delta \mathbf{Z}^h e_i$
end for



(a) Fine grid active cells

(b) Coarse grid active cells, overlaid with the fine active cells demonstrated using dashed lines

Figure 8: Coarsening of grid variables

However, on the boundary, there is no guarantee that all dependencies of the coarse grid variable restriction stencils are active fine grid degrees of freedom. Therefore, we truncate our restriction stencils to the active degrees of freedom, which is equivalent to restricting zero residuals from inactive regions. Also, when a Dirichlet boundary condition presents, we cannot compute the residuals $\boldsymbol{r}^h = \boldsymbol{f}^h - \mathbf{L}^h \boldsymbol{u}^h - \mathbf{B}^{\mathbf{h}^T} \boldsymbol{\lambda}^h$ when a $\boldsymbol{\lambda}$ value is involved. In this case, we apply a boundary relaxation strong enough such that the boundary residual is smaller than interior relaxations, and restrict zero boundary residual for these equations.

The coarse grid constraint system right hand side should have been computed from the restriction of the fine grid constraint system residual. However, due to the fact that our solutions $\boldsymbol{u}_p^h = \boldsymbol{c}^h + \mathbf{Z}^h \boldsymbol{v}^h$ always satisfy the boundary constraints exactly, the coarse grid Dirichlet boundary condition is always zero.

Also, prolongation is implemented in a distributive way, i.e. we iterate over the active coarse grid corrections, and distribute their values to all active fine level degrees of freedom. Near the domain boundary this is equivalent to prolongating a zero correction from exterior coarse grid locations, which is reasonable. We notice that this prolongation may lead to a solution away from the fundamental basis solution. Therefore, a projection onto the solution space needs to be applied after the prolongation step. The projected solution is $\boldsymbol{u}_p^h = \boldsymbol{c}^h + \mathbf{Z}^h \boldsymbol{v}^h$, for some $\boldsymbol{v}^h$, and according to the definition of $\mathbf{Z}^h$ and $\boldsymbol{c}^h$ in (28) and (29), we have $\boldsymbol{v}^h = \boldsymbol{Q}\boldsymbol{u}^h$, where $\boldsymbol{Q}$ projects a solution vector to a sub-vector

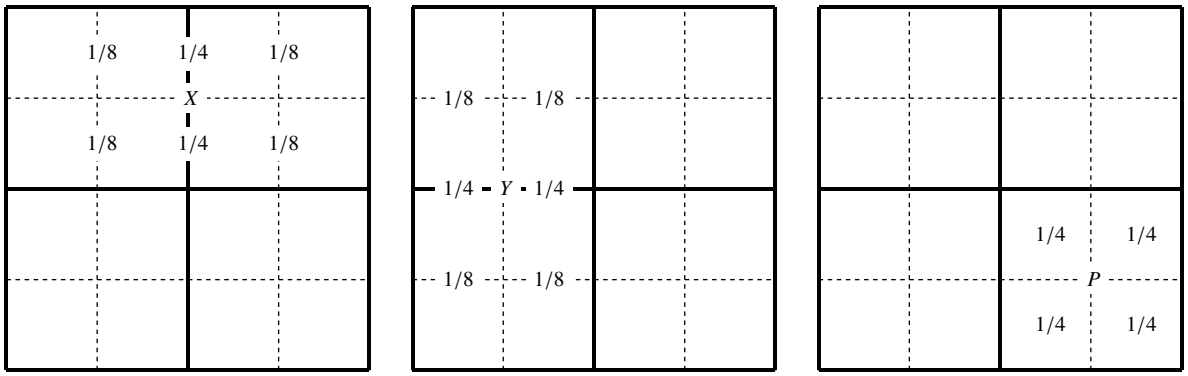


Figure 9: Restriction operator stencils.

by eliminating the degrees of freedom that correspond to aggregation representative nodes. Therefore, the projected solution is $\boldsymbol{u}_p^h = \boldsymbol{c}^h + \mathbf{Z}^h \boldsymbol{Q} \boldsymbol{u}^h$.

Now that we have covered all the components of a multigrid method, we will next look at the results we are able to obtain using these methods.

6. Numerical examples

We investigate two aspects of our algorithm: discretization error and multigrid efficiency. In this section, we apply our method on various domains with Neumann or Dirichlet boundary conditions and with a wide range of Poisson's ratios. We considered three deformations defined on three geometric domains:

1. **Keyhole domain** A Keyhole domain is enclosed by a smooth curve connecting 8 tangential circles with centers

$$\begin{aligned} \boldsymbol{c}_1 &= (0.25, 0.25); \quad \boldsymbol{c}_2 = (0.75, 0.25); \\ \boldsymbol{c}_3 &= (0.25, 0.75); \quad \boldsymbol{c}_4 = (0.75, 0.75); \\ \boldsymbol{s}_1 &= (0.5, 0.6875); \quad \boldsymbol{s}_2 = (0.5, 0.3125); \\ \boldsymbol{s}_3 &= (0.3125, 0.5); \quad \boldsymbol{s}_4 = (0.6875, 0.5); \end{aligned}$$

and radius 0.2 for the first 4 circles and $r_s = \dfrac{\sqrt{17}}{4} - 0.2$ for the last 4 circles. The radius r_s is chosen such that the circle curves are tangential and hence generate a smooth boundary. The keyhole domain can also be represented by the zero levelset of the following function:

$$\varphi(\boldsymbol{x}) = \max\left\{ \min\left\{\text{dist}(\boldsymbol{x}, \mathbf{0}, r_0), \min_i\{\text{dist}(\boldsymbol{x}, \boldsymbol{c}_i, 0.2)\}\right\}, -\min_i\{\text{dist}(\boldsymbol{x}, \boldsymbol{s}_i, r_s)\}\right\}$$

where $\text{dist}(\mathbf{x}, \mathbf{x}_0, r) = |\mathbf{x} - \mathbf{x_0}| - r$, and $r_0 = \left|\frac{0.2}{\sqrt{17}}(4, 1) - (0.25, 0.25)\right|$.
A constant divergence deformation is considered, giving the exact boundary conditions and the exact solution for the purpose of error computation.

$$\phi_1(x, y) = 2x + \frac{1}{2}\cos\pi x \sin\pi y \tag{40}$$

$$\phi_2(x, y) = 2y - \frac{1}{2}\sin\pi x \cos\pi y \tag{41}$$

2. **Flower domain** A flower-shaped domain with inner radius 0.2, outer radius 0.4 is considered with a levelset function:

$$\varphi(\boldsymbol{x}) = \text{dist}(\boldsymbol{x}, \mathbf{0}.5, 0.3 + 0.1\cos 5\theta)$$

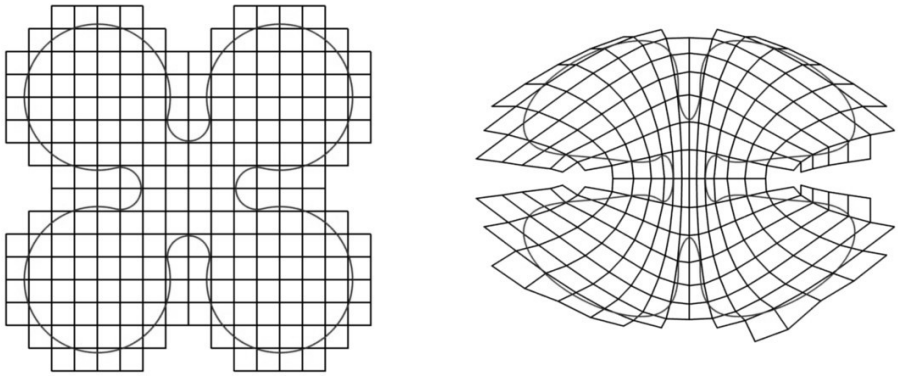


Figure 10: Left: undeformed keyhole domain; right: deformed keyhole domain

where θ is the argument of (x, y). A deformation with spatially varying divergence is considered as an exact solution.

$$\phi_1(x, y) = \frac{2x}{\sqrt{\pi}} \cos \frac{\pi}{2} y \tag{42}$$

$$\phi_2(x, y) = \frac{2x}{\sqrt{\pi}} \sin \frac{\pi}{2} y \tag{43}$$

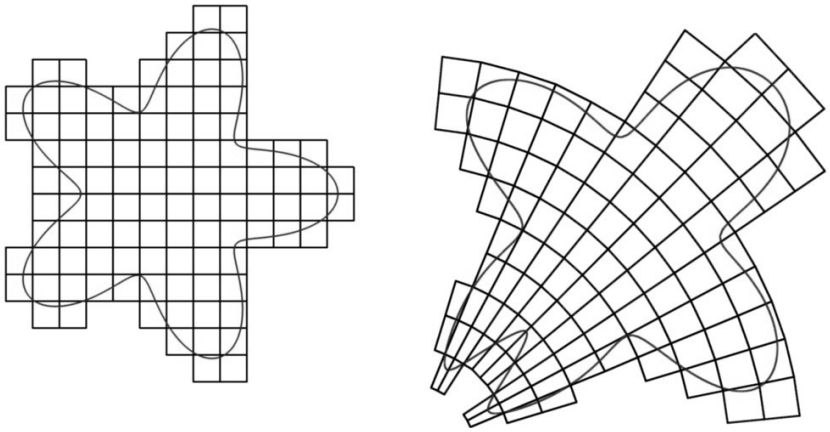


Figure 11: Left: undeformed flower domain; right: deformed flower domain

3. **Spiral domain** We consider a spiral shaped domain, defined by the zero levelset of

$$\varphi(\boldsymbol{x}) = r(\boldsymbol{y}) - (0.33 + 0.08 \cos 5\theta(\boldsymbol{y})^{\frac{1}{3}})$$

where $\boldsymbol{y}$ is $\boldsymbol{x} - (0.5, 0.5)$ rotated around $(0.5, 0.5)$ by $\theta = 14(2r(\boldsymbol{x}))^{\frac{1}{6}}$ and the deformation is given by:

$$\phi_1(x, y) = \left(\frac{1}{2}x + \frac{1}{2}\right) \cos\left(\frac{\pi}{6} + \frac{2}{3}\pi y\right) \tag{44}$$

$$\phi_2(x, y) = \left(\frac{1}{2}x + \frac{1}{2}\right) \sin\left(\frac{\pi}{6} + \frac{2}{3}\pi y\right) \tag{45}$$

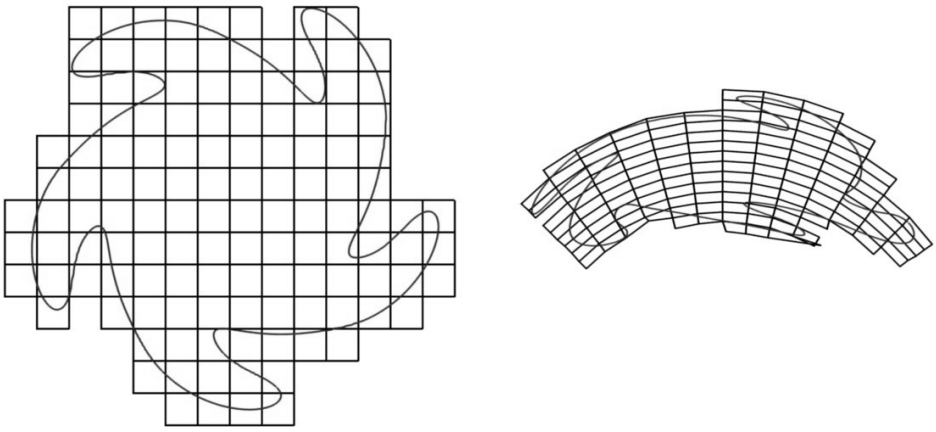


Figure 12: Left: undeformed spiral domain, right: deformed spiral domain

6.1. Discretization error

All our testing domains are embedded in a $[0, 1]^2$ domain, and we discretize this square domain with a regular grid of different resolutions ranging from 32 to 1024 in each direction. We plotted $\log_2 |\boldsymbol{u}^{\text{exact}} - \boldsymbol{u}|_\infty$ versus $\log_2$ resolution and estimated the solution accuracy order by fitting the data with a linear function. We remove the Neumann boundary condition null space by enforcing a non-embedded Dirichlet condition on all degrees of freedom within the domain $[7/16, 9/16]^2$. From the plotted error convergence behavior, we observe a second-order convergence for all three types of domains (see Figures 13, 14, and 15) for both Neumann and Dirichlet boundary conditions and for a wide range of material parameters including near-incompressible materials. We notice that the order of accuracy is slightly smaller for domains with complicated boundaries. An important source of the inaccuracy is introduced by the inconsistent domain discretization at different resolutions.

6.2. Multigrid efficiency

We also investigated the efficiency of the multigrid methods. First, we consider a periodic boundary condition problem defined on $[0, 1]^2$ and with the exact solution given by

$$\phi_1(x, y) = \sin 2\pi x + \cos 2\pi y$$
$$\phi_2(x, y) = \cos 2\pi x + \sin 2\pi y.$$

Although periodic boundary conditions will not appear in practical elasticity problems, we consider the periodic boundary condition problem to evaluate the multigrid solver while avoiding issues that may arise with boundary relaxation. We first fix the problem resolution to 128×128 and apply finite element distributive relaxation and the distributive relaxation for the finite difference defect correction problem as the interior relaxations. We also apply the bilinear prolongation and a prolongation with $\mathbf{P} = 4\mathbf{R}^T$. While low incompressibility problems generate convergence rates no larger than 0.3 for a multigrid V-(1,1) cycle with all different prolongation and distribution options, we focus on the the harder high-incompressible case with Poisson's ratio being 0.49, and investigate both V-(1,1) cycle and W-(1,1) cycle convergence. As shown in Table 1, both finite element distributive relaxation and the finite difference defect correction scheme generate convergence rates less than 0.5 with a multigrid V-(1,1) cycle. Although finite difference defect correction distributive relaxation generates slower convergence than finite element distributive relaxation for the V-(1,1) cycle, with the help of a bilinear interpolation or W-(1,1) cycle, we are able to generate a better convergence rate of 0.23.

We further investigate the multigrid convergence rate for various resolutions by sticking to one scheme which uses finite element distribution, low order prolongation and a V-(1,1) cycle, and plot the convergence rate for problems discretized with resolutions from 32 to 1024 (see Figure 16). A consistent multigrid convergence rate is observed under refinement.

boundary condition	distribution	multigrid cycle	$\mathbf{P}_{hi}$	$\mathbf{P}_{lo}$
Periodic	FD	V-(1,1)	0.24	0.42
	FD	W-(1,1)	0.23	0.25
	FE	V-(1,1)	0.13	0.24
	FE	W-(1,1)	0.13	0.30
Dirichlet	FD	V-(1,1)	0.72	0.72
	FD	W-(1,1)	0.72	0.72
	FE	V-(1,1)	0.37	0.36
	FE	W-(1,1)	0.42	0.42
Neumann	FD	V-(1,1)	0.70	0.70
	FD	W-(1,1)	0.68	0.68
	FE	V-(1,1)	0.50	0.50
	FE	W-(1,1)	0.35	0.35

Table 1: Multigrid asymptotic convergence rates for different combinations of boundary conditions, interpolation and distributive relaxation on the flower domain (Poisson's ratio=0.49, resolution=128×128). For optional prolongations, $\mathbf{P}_{hi}$ is bilinear interpolation and $\mathbf{P}_{lo}$ is for $\mathbf{P} = 4\mathbf{R}^T$. For the optional distributions, FE is the distribution matrix discretized with the bilinear finite element method, and FD is using defect correction by employing finite difference distributive relaxation.

While all schemes give a nice convergence rate in periodic cases, the convergence rate with Neumann boundary conditions and Dirichlet boundary conditions varies. In non-trivial boundary condition cases, the convergence rate is mainly restricted by the efficiency of the boundary relaxation. Therefore, the convergence rate of a V-(1,1) cycle and a W-(1,1) cycle are very similar. Also different prolongation schemes generate very similar convergence rates (see Table 1 for the convergence rate of all algorithm options for the flower domain problem at a fixed resolution of 128×128). The difference between using a finite element distributive relaxation and a finite difference defect correction distributive relaxation reflects the efficiency of the whole smoother in combination with the boundary relaxations.

We further investigate the convergence rate under refinement, due to the fact that different prolongation schemes and multigrid cycles generate similar convergence rates. We only plot the asymptotic convergence rate of a V-(1,1) cycle with $\mathbf{P} = 4\mathbf{R}^T$ and using finite element distributive relaxation in the interior. For both the Dirichlet and Neumann boundary condition problem, we plot the asymptotic convergence rate for resolutions from 32 to 1024 and the residual reduction at each iteration for representative resolution numbers that are powers of 2. We observed consistent convergence rate at all resolutions (see Figure 17 and Figure 18).

7. Conclusion and future work

We developed a second-order mixed finite element discretization for linear elasticity of all material parameters from compressible to highly incompressible. We developed a multigrid method for the linear system induced by the discretization. By applying approximated distributive relaxation, we can achieve a fast and parameter-independent convergence rate when no boundary conditions are present. With specified boundary conditions defined on a variety of domains, we also demonstrated that the multigrid method can maintain a good convergence rate with only a small number of boundary relaxations. However, the optimum convergence demonstrated in the periodic boundary condition case was not achieved. In the future, we are interested in investigating a more efficient boundary smoother that avoids unaugmentation as well as a continuous extension to Stokes equations.

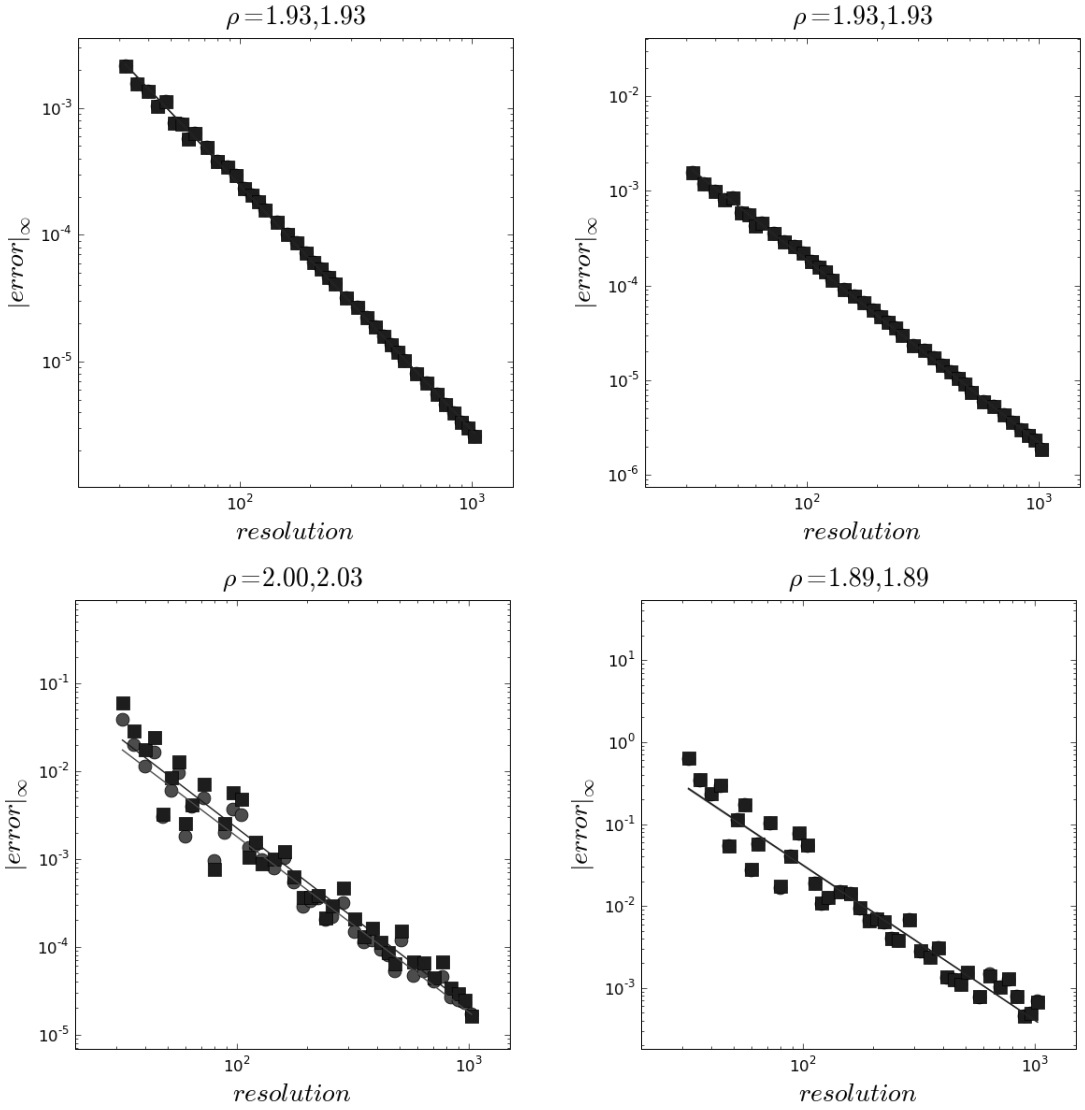


Figure 13: Order of accuracy, ρ, for keyhole domain; top: Neumann boundary condition; bottom: Dirichlet boundary condition; left: Poisson's ratio=0.3; right: Poisson's ratio=0.49; square marker: x component; circle marker: y component.

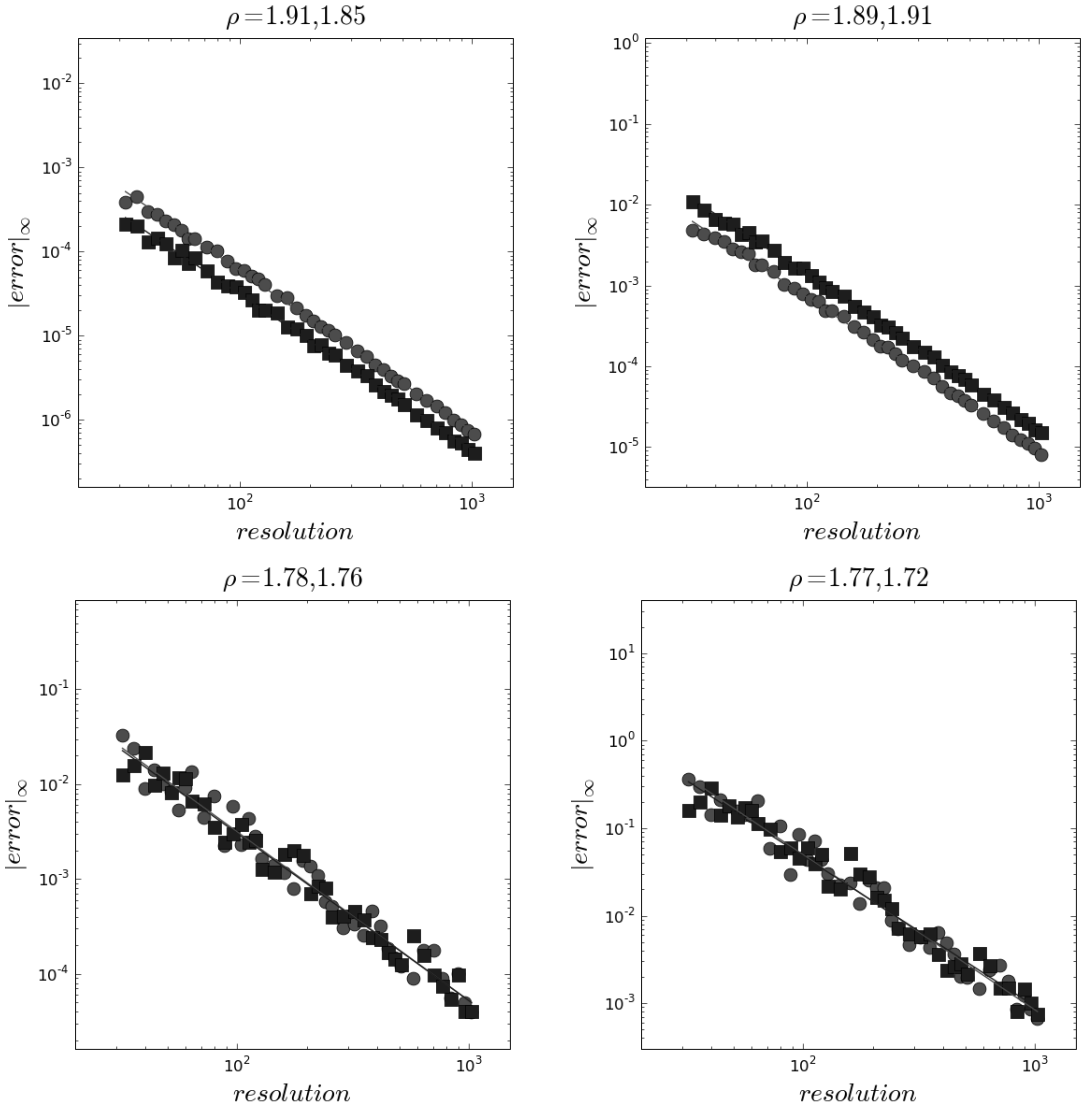


Figure 14: Order of accuracy, ρ, for flower domain. Top: Neumann boundary condition, bottom: Dirichlet boundary condition; left: Poisson's ratio=0.3; right: Poisson's ratio=0.49; square marker: x component; circle marker: y component.

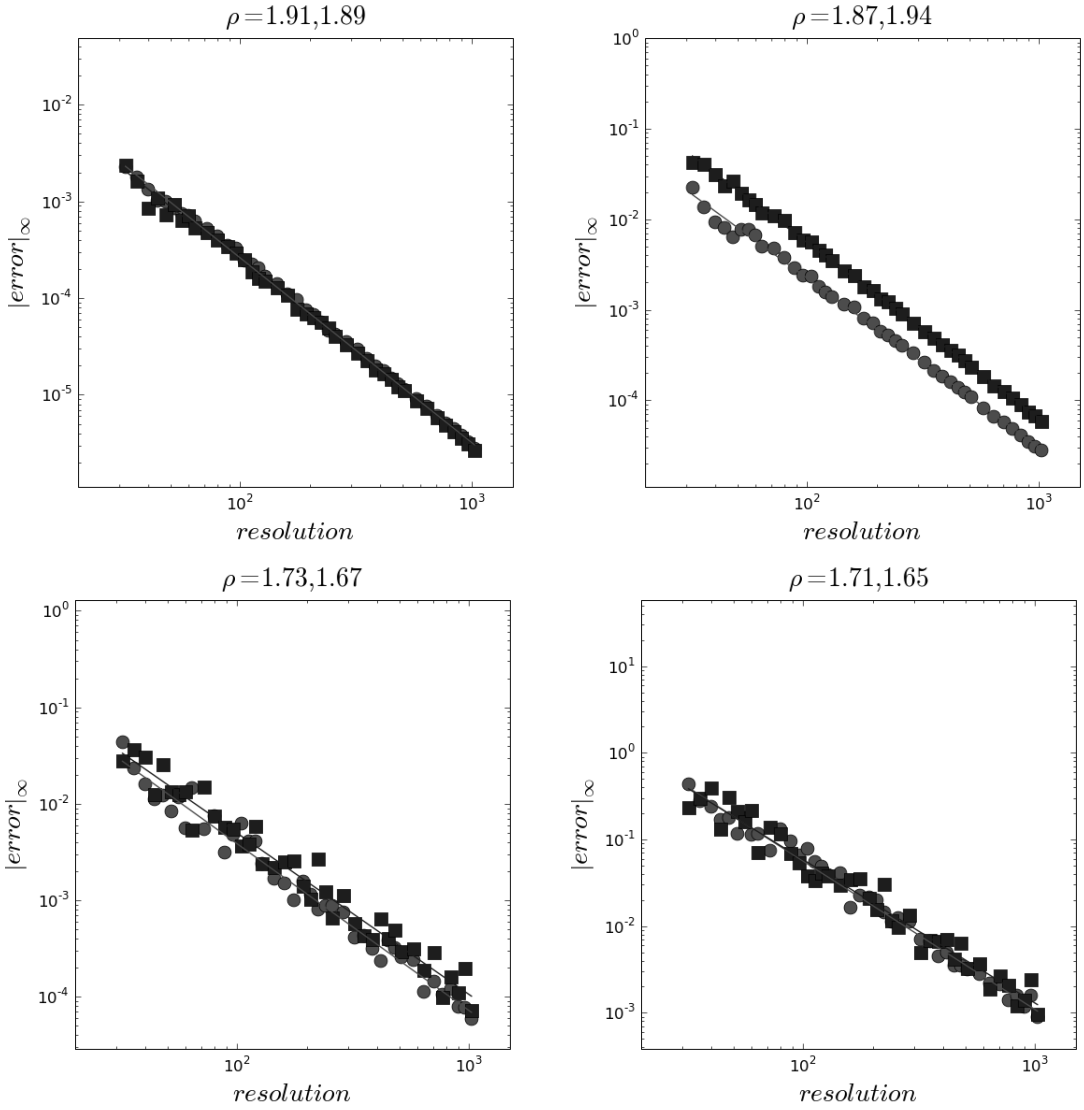


Figure 15: Order of accuracy, ρ, for spiral domain. Top: Neumann boundary condition; bottom: Dirichlet boundary condition; left: Poisson's ratio=0.3; right: Poisson's ratio=0.49; square marker: x component; circle marker: y component.

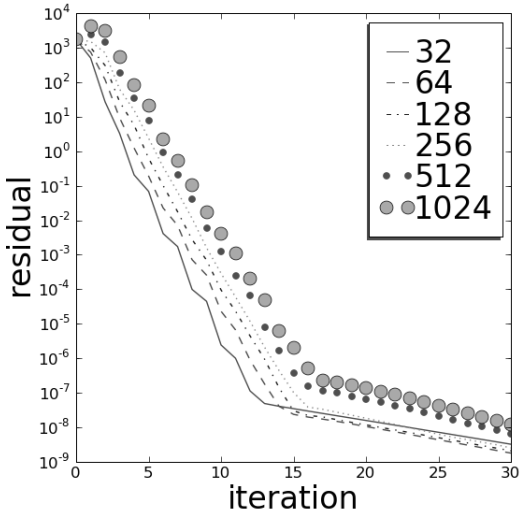


Figure 16: Multigrid V-(1,1) cycle convergence under refinement (Poisson's ratio = 0.49, periodic boundary conditions, finite element distribution).

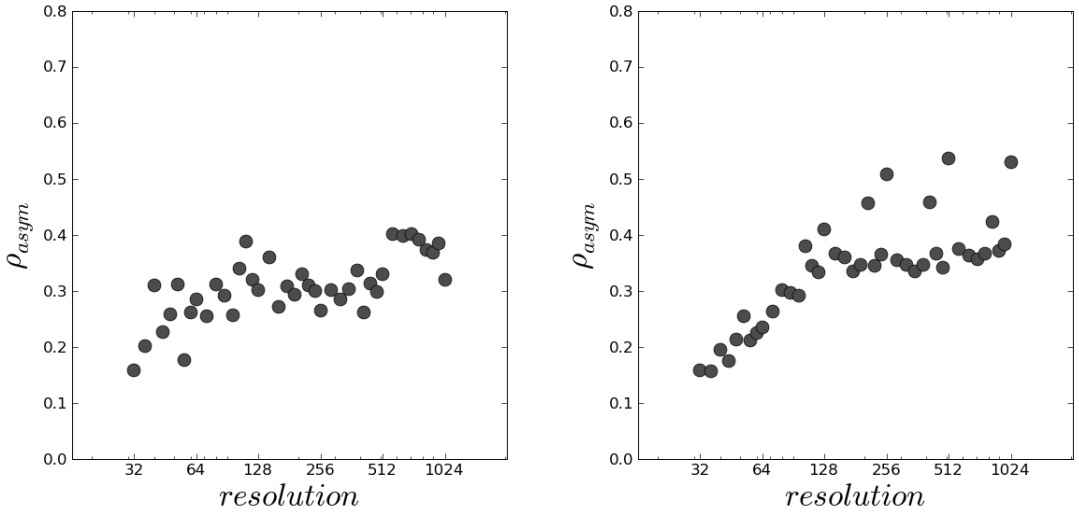


Figure 17: Multigrid V-(1,1) cycle convergence rate at resolutions from 32 to 1024 (Poisson's ratio = 0.49). Left: Dirichlet boundary condition; right: Neumann boundary condition.

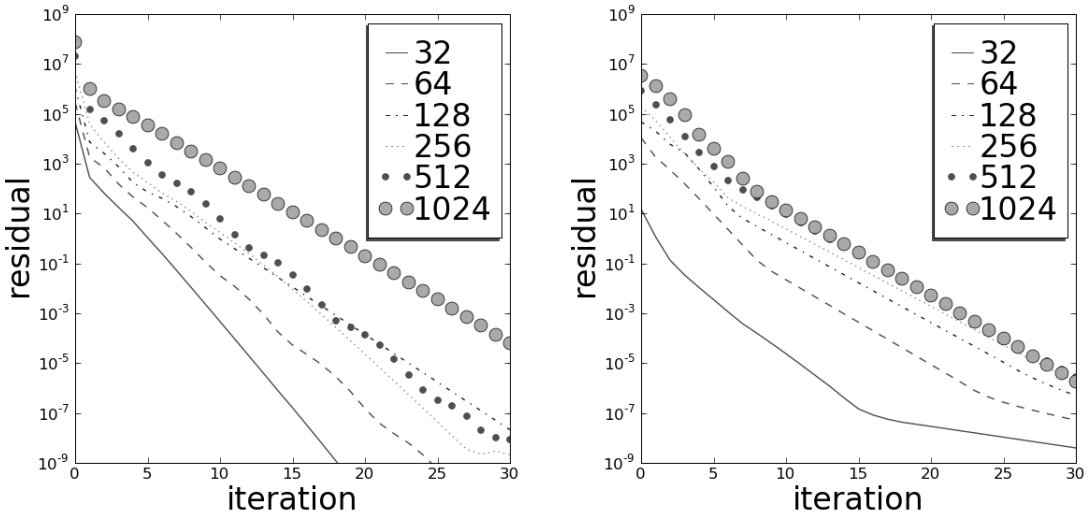


Figure 18: Multigrid V-(1,1) cycle convergence under refinement (Poisson's ratio = 0.49). Left: Dirichlet boundary condition; right: Neumann boundary condition.

References

[1] Y. Zhu, E. Sifakis, J. Teran, A. Brandt, An efficient multigrid method for the simulation of high-resolution elastic solids, ACM Transactions on Graphics 29 (2010) 1–18.

[2] G. Allaire, F. Jouve, A. Toader, Structural optimization using sensitivity analysis and a level-set method, Journal of Computational Physics 194 (2004) 363–393.

[3] V. Challis, A. Roberts, A. Wilkins, Fracture resistance via topology optimization, Structural and Multidisciplinary Optimization 36 (2008) 263–271.

[4] P. Duysinx, L. Miegroet, T. Jacobs, C. Fleury, Generalized shape optimization using x-fem and level set methods, in: M. P. Bendsøe, N. Olhoff, O. Sigmund (Eds.), IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials, volume 137 of *Solid Mechanics and Its Applications*, Springer Netherlands, 2006, pp. 23–32.

[5] S. Osher, F. Santosa, Level set methods for optimization problems involving geometry and constraints: I. frequencies of a two-density inhomogeneous drum, Journal of Computational Physics 171 (2001) 272–288.

[6] J. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, Journal of Computational Physics 163 (2000) 489–528.

[7] P. Wei, M. Wang, A structural optimization method with xfem and level set model, in: I. Horváth, Z. Rusák (Eds.), Proceedings of the TMCE 2008.

[8] A. Lew, G. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, International Journal for Numerical Methods in Engineering 76 (2008) 427–454.

[9] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Physics of fluids 8 (1965) 2182–2189.

[10] C. Peskin, Flow patterns around heart valves: a numerical method, Journal of Computational Physics 10 (1972) 252–271.

[11] M. Hyman, Non-iterative numerical solution of boundary-value problems, Applied Scientific Research, Section B 2 (1952) 325–351.

[12] V. Sau'lev, On solving boundary value problems on high-performance computers by fictitious domain methods, Siberian Mathematical Journal 4 (1963) 912–925.

[13] R. Glowinski, T. Pan, T. Hesla, D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, International Journal of Multiphase Flow 25 (1999) 755–794.

[14] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 112 (1994) 133–148.

[15] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flows, Journal of Computational Physics 169 (2001) 363–426.

[16] L. Parussini, V. Pediroda, Fictitious domain approach with hp-finite element approximation for incompressible fluid flow, Journal of Computational Physics 228 (2009) 3891–3910.

[17] L. Parussini, Fictitious domain approach via lagrange multipliers with least squares spectral element method, Journal of Scientific Computing 37 (2008) 316–335.

[18] F. Bertrand, P. Tanguy, F. Thibault, A three-dimensional fictitious domain method for incompressible fluid flow problems, International Journal for Numerical Methods in Fluids 25 (1997) 719–736.

[19] J. Teran, C. Peskin, Tether force constraints in stokes flow by the immersed boundary method on a periodic domain, SIAM Journal on Scientific Computing 31 (2009) 3404–3416.

[20] G. Biros, L. Ying, D. Zorin, A fast solver for the stokes equations with distributed forces in complex geometries, Journal of Computational Physics 193 (2004) 317–348.

[21] V. Rutka, A staggered grid-based explicit jump immersed interface method for two-dimensional stokes flows, International Journal for Numerical Methods in Fluids 57 (2008) 1527–1543.

[22] C. Daux, N. Moës, J. Dolbow, N. Sukumar, T. Belytschko, Arbitrary branched and intersecting cracks with the extended finite element method, International Journal for Numerical Methods in Engineering 48 (2000) 1741–1760.

[23] N. Sukumar, D. Chopp, N. Moës, T. Belytschko, Modeling holes and inclusions by level sets in the extended finite-element method, Computer methods in applied mechanics and engineering 190 (2001) 6183–6200.

[24] A. Almgren, J. Bell, P. Colella, T. Marthaler, A cartesian grid projection method for the incompressible euler equations in complex geometries, SIAM Journal of Scientific Computing 18 (1997) 1289–1309.

[25] N. Moës, E. Béchet, M. Tourbier, Imposing dirichlet boundary conditions in the extended finite element method, International Journal for Numerical Methods in Engineering 67 (2006) 1641–1669.

[26] J. Dolbow, A. Devan, Enrichment of enhanced assumed strain approximations for representing strong discontinuities: addressing volumetric incompressibility and the discontinuous patch test, International journal for numerical methods in engineering 59 (2004) 47–67.

[27] G. Wagner, N. Moës, W. Liu, T. Belytschko, The extended finite element method for rigid particles in stokes flow, International Journal for Numerical Methods in Engineering 51 (2001) 293–313.

[28] R. Becker, E. Burman, P. Hansbo, A nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity, Computer Methods in Applied Mechanics and Engineering 198 (2009) 3352–3360.

[29] A. Coppola-Owen, R. Codina, Improving eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions, International Journal for Numerical Methods in Fluids 49 (2005) 1287–1304.

[30] J. Chessa, T. Belytschko, An extended finite element method for two-phase fluids, Journal of Applied Mechanics 70 (2003) 10–17.

[31] A. Gerstenberger, W. Wall, An extended finite element method/lagrange multiplier based approach for fluid-structure interaction, Computer Methods in Applied Mechanics and Engineering 197 (2008) 1699–1714.

[32] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface cartesian grid method i: an easily implemented technique for 3d moving boundary computations, Journal of Computational Physics 210 (2005) 1–31.

[33] Y. Ng, C. Min, F. Gibou, An efficient fluid-solid coupling algorithm for single-phase flows, Journal of Computational Physics 228 (2009) 8807–8829.
[34] C. Batty, F. Bertails, R. Bridson, A fast variational framework for accurate solid-fluid coupling, ACM Transactions on Graphics 26 (2007).
[35] A. Wiegmann, K. Bube, The explicit-jump immersed interface method: finite difference methods for pdes with piecewise smooth solutions, SIAM Journal on Numerical Analysis (2000) 827–862.
[36] Z. Li, X. Wan, K. Ito, S. Lubkin, An augmented approach for the pressure boundary condition in a stokes flow, Communications in Computational Physics 1 (2006) 874–885.
[37] G. Chen, Z. Li, P. Lin, A fast finite difference method for biharmonic equations on irregular domains and its application to an incompressible stokes flow, Advances in Computational Mathematics 29 (2008) 113–133.
[38] S. Xu, The immersed interface method for simulating prescribed motion of rigid objects in an incompressible viscous flow, Journal of Computational Physics 227 (2008) 5045–5071.
[39] V. Rutka, A. Wiegmann, A fast finite difference method for elliptic pdes in domains with non-grid aligned boundaries with application to 3d linear elasticity, Progress in industrial mathematics at ECMI 2002 (2004) 363.
[40] I. Bijelonja, I. Demirdžić, S. Muzaferija, A finite volume method for incompressible linear elasticity, Computer Methods in Applied Mechanics and Engineering 195 (2006) 6378–6390.
[41] L. Beirão da Veiga, V. Gyrya, K. Lipnikov, G. Manzini, Mimetic finite difference method for the stokes problem on polygonal meshes, Journal of Computational Physics 228 (2009) 7215–7232.
[42] P. Barton, D. Drikakis, An Eulerian method for multi-component problems in non-linear elasticity with sliding interfaces, Journal of Computational Physics 229 (2010) 5518–5540.
[43] D. Hill, D. Pullin, M. Ortiz, D. Meiron, An eulerian hybrid weno centered-difference solver for elastic-plastic solids, Journal of Computational Physics 229 (2010) 9053–9072.
[44] A. Klawonn, Block-triangular preconditioners for saddle point problems with a penalty term, SIAM Journal on Scientific Computing 19 (1998) 172–184.
[45] A. Klawonn, An optimal preconditioner for a class of saddle point problems with a penalty term, SIAM Journal on Scientific Computing 19 (1998) 540–552.
[46] J. Bramble, J. Pasciak, A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems, Mathematics of Computation 50 (1998) 1–17.
[47] C. Farhat, M. Lesoinne, K. Pierson, A scalable dual-primal domain decomposition method, Numerical linear algebra with applications 7 (2000) 687–714.
[48] A. Klawonn, L. Pavarino, Overlapping Schwarz methods for mixed linear elasticity and Stokes problems, Computer Methods in Applied Mechanics and Engineering 165 (1998) 233–245.
[49] C. Dohrmann, A preconditioner for substructuring based on constrained energy minimization, SIAM Journal on Scientific Computing 25 (2003) 246–258.
[50] L. Pavarino, O. Widlund, S. Zampini, BDDC preconditioners for spectral element discretizations of almost incompressible elasticity in three dimensions, SIAM Journal on Scientific Computing 32 (2010) 3604–3626.
[51] R. Verfürth, A multilevel algorithm for mixed problems, SIAM Journal on Numerical Analysis 21 (1984) 264–271.
[52] M. Kočvara, J. Mandel, A multigrid method for three-dimensional elasticity and algebraic convergence estimates, Applied Mathematics and Computation 23 (1987) 121 – 135.
[53] Z. Huang, A multi-grid algorithm for mixed problems with penalty, Numerische Mathematik 57 (1990) 227–247.
[54] S. Brenner, A nonconforming mixed multigrid method for the pure displacement problem in planar linear elasticity, SIAM Journal on Numerical Analysis 30 (1993) 116–135.
[55] Z. Cai, T. Manteuffel, S. McCormick, S. Parter, First-order system least squares (FOSLS) for planar linear elasticity: pure traction problem, SIAM Journal on Numerical Analysis 35 (1998) 320–335.
[56] R. Hiptmair, R. H. W. Hoppe, Multilevel methods for mixed finite elements in three dimensions, Numerische Mathematik 82 (1999) 253–279.
[57] O. Axelsson, A. Padiy, On a robust and scalable linear elasticity solver based on a saddle point formulation, International Journal for Numerical Methods in Engineering 44 (1999) 801–818.
[58] J. Schöberl, Multigrid methods for a parameter dependent problem in primal variables, Numerische Mathematik 84 (1999) 97–119.
[59] C. Wieners, Robust multigrid methods for nearly incompressible elasticity, Computing 64 (2000) 289–306.
[60] J. Heys, T. Manteuffel, S. McCormick, J. Ruge, First-order system least squares (FOSLS) for coupled fluid-elastic problems, Journal of Computational Physics 195 (2004) 560–575.
[61] Y.-J. Lee, J. Wu, J. Chen, Robust multigrid method for the planar linear elasticity problems, Numerische Mathematik 113 (2009) 473–496.
[62] F. Gaspar, J. Gracia, F. Lisbona, C. Oosterlee, Distributive smoothers in multigrid for problems with dominating grad-div operators, Numerical linear algebra with applications 15 (2008) 661–683.
[63] F. Brezzi, M. Fortin, Mixed and hybrid finite element methods, Springer-Verlag: New York, 1991.
[64] H. Han, X. Wu, A new mixed finite element formulation and the MAC method for the Stokes equations, SIAM Journal on Numerical Analysis 35 (1998) 560–571.
[65] J. Bedrossian, J. von Brecht, S. Zhu, E. Sifakis, J. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains, Journal of Computational Physics 229 (2010) 6405–6426.