ORIGINAL PAPER

Multigrid solvers for immersed finite element methods and immersed isogeometric analysis

F. de Prenter[1,3] · C. V. Verhoosel[1] · E. H. van Brummelen[1] · J. A. Evans[2] · C. Messe[2,4] · J. Benzaken[2,5] · K. Maute[2]

Abstract

Ill-conditioning of the system matrix is a well-known complication in immersed finite element methods and trimmed isogeometric analysis. Elements with small intersections with the physical domain yield problematic eigenvalues in the system matrix, which generally degrades efficiency and robustness of iterative solvers. In this contribution we investigate the spectral properties of immersed finite element systems treated by Schwarz-type methods, to establish the suitability of these as smoothers in a multigrid method. Based on this investigation we develop a geometric multigrid preconditioner for immersed finite element methods, which provides mesh-independent and cut-element-independent convergence rates. This preconditioning technique is applicable to higher-order discretizations, and enables solving large-scale immersed systems at a computational cost that scales linearly with the number of degrees of freedom. The performance of the preconditioner is demonstrated for conventional Lagrange basis functions and for isogeometric discretizations with both uniform B-splines and locally refined approximations based on truncated hierarchical B-splines.

Keywords Immersed finite element method · Fictitious domain method · Iterative solver · Preconditioner · Multigrid

1 Introduction

Immersed methods are useful tools to avoid laborious and computationally expensive procedures for the generation of body-fitted finite element discretizations or analysis-suitable NURBS geometries in isogeometric analysis, specifically for problems on complex, moving, or implicitly defined geometries. Immersed finite element techniques, such as the finite cell method [1–3], CutFEM [4,5], and immersogeometric analysis [6,7], have been successfully applied to a broad range of problems. Noteworthy applications include isogeometric analysis on trimmed CAD objects, e.g., [8–13], fluid–structure interaction with large displacements, e.g.,

[14–21], scan based analysis [22–27] and topology optimization, e.g., [28–34].

An essential aspect of finite element methods and isogeometric analysis is the computation of the solution to a system of equations. This is specifically challenging for systems derived from immersed methods, since such methods generally yield severely ill-conditioned system matrices [35]. For this reason, many researchers resort to direct solvers, e.g., [2,3,9–12,24], the efficiency of which is not affected by the conditioning of the system matrix. Nevertheless, the computational cost of direct solvers, both in terms of memory and floating point operations, scales poorly with the size of the system. With iterative solvers, the scaling between the computational cost and the size of the system is generally better, making these more suitable for large systems of equations [36]. However, the efficiency and reliability of iterative solution methods depends on the conditioning of the system. Without dedicated treatments, the severe ill-conditioning of linear systems derived from immersed finite element methods generally forestalls convergence of iterative solution procedures. Multiple resolutions for these conditioning problems have been proposed, the most prominent of which are the ghost penalty, e.g., [4,5,37], constraining, extending, or aggregation of basis functions,

✉ F. de Prenter
    fritsdeprenter@gmail.com

1   Department of Mechanical Engineering, Eindhoven
    University of Technology, Eindhoven, The Netherlands

2   Ann & H.J. Smead Department of Aerospace Engineering
    Sciences, University of Colorado Boulder, Boulder, USA

3   Reden B.V., Hengelo (Ov.), The Netherlands

4   German Aerospace Center (DLR), Bremen, Germany

5   Walt Disney Animation Studios, Burbank, USA

e.g., [13,38–46], and preconditioning, which is discussed in detail below.

Several dedicated preconditioners have been developed for immersed finite element methods. It is demonstrated in [47] that a diagonal preconditioner in combination with the constraining of very small basis functions results in an effective treatment for systems with linear bases. With certain restrictions to the cut-element-geometry, [48] derives that a scalable preconditioner for linear bases is obtained by combining diagonal scaling of basis functions on cut elements with standard multigrid techniques for the remaining basis functions. In [49] the scaling of a Balancing Domain Decomposition by Constraints (BDDC) is tailored to cut elements, and this is demonstrated to be effective with linear basis functions. An algebraic preconditioning technique is presented in [35], which results in an effective treatment for smooth function spaces. References [50] and [51] establish that additive Schwarz preconditioners can effectively resolve the conditioning problems of immersed finite element methods with higher-order discretizations, for both isogeometric and hp-finite element function spaces and for both symmetric positive definite (SPD) and non-SPD problems. Furthermore, the numerical investigation in [50] conveys that the conditioning of immersed systems treated by an additive Schwarz preconditioner is very similar to that of mesh-fitting systems. In particular, the condition number of an additive Schwarz preconditioned immersed system exhibits the same mesh-size dependence as mesh-fitting approaches [52,53], which opens the doors to the application of established concepts of multigrid preconditioning. It should be mentioned that similar conditioning problems as in immersed methods occur in XFEM and GFEM. Dedicated preconditioners have been developed for these problems as well, a survey of which can be found in [50].

Multigrid methods effectively resolve the mesh-size dependence of the conditioning of linear systems and its effect on the convergence of iterative solution methods. In particular, the use of overlapping Schwarz smoothers can lead to multigrid methods with provably mesh-independent convergence rates [54–56]. There exists a rich literature on multigrid techniques, and interested readers are directed to the reference works [57–60]. Multigrid methods have not been studied extensively in the context of immersed finite element methods, but detailed studies regarding closely related aspects are available. In isogeometric analysis, multigrid is an established concept, e.g., [61–68]. In regard of the present manuscript [69–72] are particularly noteworthy, as these all employ smoothers that are based on Schwarz-type techniques. Another interesting contribution is [73], which presents a multigrid technique for locally refined function spaces with truncated hierarchical B-splines, that are also employed in this manuscript. Further noteworthy references are multigrid preconditioners for XFEM [74,75], unfitted dis-

continuous Galerkin (UDG) and CutFEM with ghost penalty stabilization [76], unfitted interface problems [77], and an algebraic multigrid (AMG) preconditioner that is applied to immersed systems which have been treated by an aggregation procedure [78].

The main objective of this contribution is to develop a geometric multigrid preconditioning technique that is applicable to higher-order immersed finite element methods with conventional, isogeometric, and locally refined basis functions. This preconditioner enables iterative solution methods with a convergence rate that is unaffected by either the cut elements or the grid size, such that the solution is obtained at a computational cost that scales linearly with the number of degrees of freedom (DOFs). The intrinsic dependence on mesh regularity in geometric multigrid approaches is non-restrictive for immersed finite element methods, as these methods generally employ structured grids. Based on the observations regarding the mesh-size dependence of the additive Schwarz preconditioner developed in [50], this contribution further investigates the spectral properties of immersed systems treated with Schwarz-type preconditioners, in order to establish the suitability of these as smoothers in a multigrid method. This results in a preconditioning technique with the desired properties, the performance of which is demonstrated on a range of test cases with multi-million DOFs. This numerical investigation includes the discretization order, the application to locally refined bases with truncated hierarchical B-splines, and a detailed study of aspects that are specific for immersed finite elements.

This contribution only considers SPD problems. Multigrid methods are an established concept in fluid mechanics as well [60], however, and similar Schwarz-type methods have successfully been applied to flow problems with both immersed finite element methods [50] and mesh-fitting multigrid solvers [71], i.e., Vanka-smoothers [79]. Therefore, it is anticipated that the presented preconditioning technique extends matatis mutandis to non-SPD and mixed formulations.

In Sect. 2 of this contribution the employed immersed finite element method is presented, including the quadrature on cut elements and the applied discretization spaces. Section 3 investigates spectral properties of immersed finite element methods and presents the developed geometric multigrid preconditioner. In Sect. 4 this preconditioning technique is assessed on a range of test cases, and conclusions are drawn in Sect. 5.

2 Immersed finite element formulation

We consider problems on a two-dimensional or three-dimensional domain $\Omega \subset \mathbb{R}^d$ ($d \in \{2, 3\}$), that is referred to as the physical domain. The physical domain is encap-
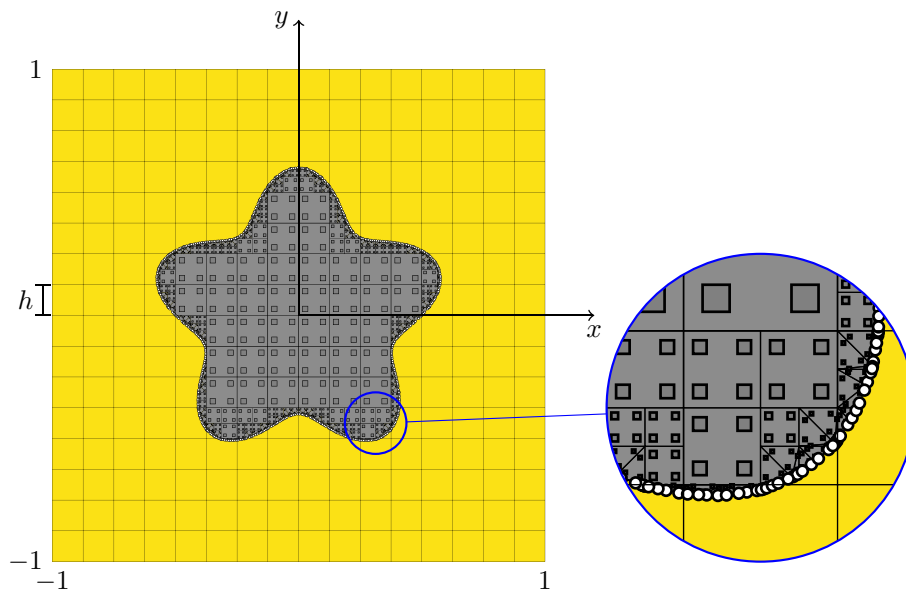
Fig. 1 The physical domain Ω defined by the level set function $\psi(x, y) = 0.5 + 0.1\sin(5\theta) - r(x, y)$ that is inspired by [80], with $\theta = \arctan2(y, x)$ and $r(x, y)^2 = x^2 + y^2$. The physical domain is encapsulated by the embedding domain $(-1, 1)^2$ on which the grid with grid size $h = \frac{1}{8}$ is posed. Quadrature is performed by the integration procedure that is outlined in [23]. This procedure recursively bisects cut elements, until a maximum integration depth is reached. The bisected elements are then triangulated, to obtain integration subcells. Standard Gaussian integration points are applied on these subcells to evaluate volumetric integrals (gray squares). Boundary integrals are computed by Gaussian quadrature on the approximate boundary, formed by the edges of the integration subcells (white circles)

sulated by a fictitious extension, to obtain an embedding domain of simple shape, as illustrated in Fig. 1. Because of the simple shape of the embedding domain, it is trivial to generate a tensor product mesh. These structured meshes render immersed finite elements ideally suitable for geometric multigrid approaches. The set of elements that intersects the physical domain is referred to as the background grid, and can serve as a substructure to construct different types of basis functions. We denote this mesh by $\mathcal{T}_h$, where h refers to the grid size, and the basis functions that are supported on the physical domain span the approximation space $\mathcal{V}_h$. A non-trivial aspect of immersed finite element methods is the integration over cut elements. Herein we employ the procedure as outlined in [23], which is illustrated in Fig. 1.

The numerical examples in this contribution consider problems in linear elasticity:

$$\begin{cases} \mathrm{div}\,(\boldsymbol{\sigma}) + \boldsymbol{f} = \boldsymbol{0} & \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{g}^D & \text{on } \Gamma^D, \\ \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{g}^N & \text{on } \Gamma^N, \end{cases} \quad (1)$$

with Cauchy stress tensor $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{u}) = \lambda\,\mathrm{div}\,(\boldsymbol{u})\,\mathbf{I} + 2\mu\nabla^s\boldsymbol{u}$, Lamé parameters λ and μ, ∇^s denoting the symmetric gradient operator, $\mathbf{n}$ the exterior unit normal vector and $\Gamma^D \cup \Gamma^N = \partial\Omega$ complementary parts of the boundary on which Dirichlet and Neumann conditions are prescribed. The formulations

in this section are restricted to pure Dirichlet or Neumann boundary conditions, but can easily be modified to mixed boundary conditions with a prescribed normal displacement and tangential traction or Robin-type conditions. We consider approximations of (1) based on a symmetric and coercive variational form:

$$\begin{cases} \text{Find } \boldsymbol{u}_h \in \mathcal{V}_h \text{ such that for all } \boldsymbol{v}_h \in \mathcal{V}_h: \\ \quad a_h\,(\boldsymbol{v}_h, \boldsymbol{u}_h) = b_h\,(\boldsymbol{v}_h)\,, \end{cases} \quad (2)$$

with the bilinear and linear operators defined as:

$$a_h\,(\boldsymbol{v}_h, \boldsymbol{u}_h) = \int_\Omega \nabla^s \boldsymbol{v}_h : \boldsymbol{\sigma}\,(\boldsymbol{u}_h)\,\mathrm{d}V$$

$$+ \int_{\Gamma^D} \left(\lambda\beta_h^\lambda(\boldsymbol{v}_h \cdot \boldsymbol{n})(\boldsymbol{u}_h \cdot \boldsymbol{n}) + 2\mu\beta_h^\mu \boldsymbol{v}_h \cdot \boldsymbol{u}_h\right)\,\mathrm{d}S,$$

$$b_h\,(\boldsymbol{v}_h) = \int_\Omega \boldsymbol{v}_h \cdot \boldsymbol{f}\,\mathrm{d}V \quad (3)$$

$$+ \int_{\Gamma^D} \left(\lambda\beta_h^\lambda(\boldsymbol{v}_h \cdot \boldsymbol{n})(\boldsymbol{g}^D \cdot \boldsymbol{n}) + 2\mu\beta_h^\mu \boldsymbol{v}_h \cdot \boldsymbol{g}^D\right)\,\mathrm{d}S$$

$$+ \int_{\Gamma^N} \boldsymbol{v}_h \cdot \boldsymbol{g}^N\,\mathrm{d}S.$$

The weak formulation in (2) and (3) imposes the Dirichlet conditions by the penalty method, which bypasses the computation of local stabilization parameters. The analysis of the conditioning of immersed finite elements in [35]
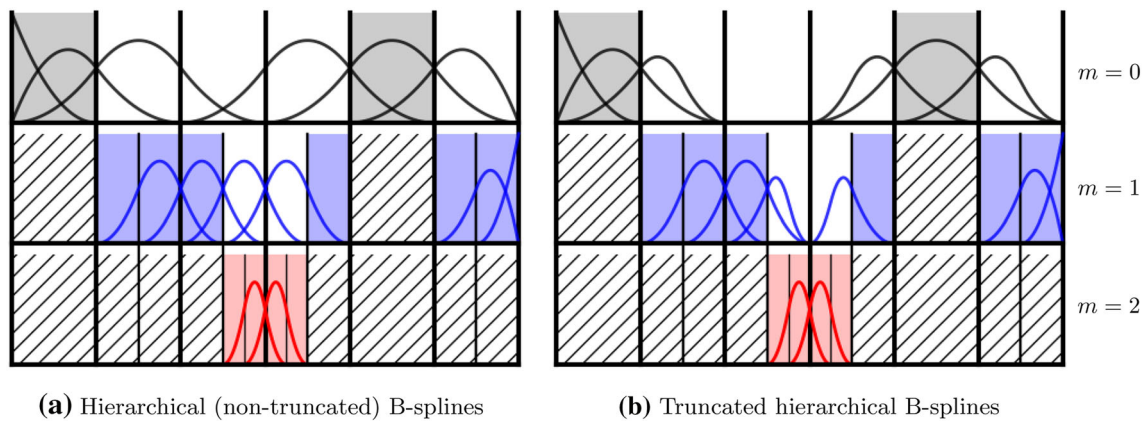
(a) Hierarchical (non-truncated) B-splines

(b) Truncated hierarchical B-splines

Fig. 2 Illustration of an open, hierarchically refined, quadratic B-spline basis. The local refinement level is denoted by $m \leq M$, with M the number of local refinement levels. Active elements K_i^m of level $m \in \{0, 1, 2\}$ are indicated in gray, blue, and red, respectively. The set of all active elements of level m is denoted by $\mathcal{T}_h^m$, and the union of these sets forms the hierarchically refined mesh $\mathcal{T}_h = \cup_{m=0}^{m=M} \mathcal{T}_h^m$. Refined elements, i.e., that have active elements at a finer level underneath, are shown blank

and inactive elements, i.e., that have an active element at a coarser level above, are hatched. **a** Shows the active standard (non-truncated) hierarchical B-spline basis functions. A basis function of level m is active under the condition that: (i) it is supported on at least one active element of level m, and (ii) it is not supported on inactive elements of level m, i.e., the entire support is of local refinement level $\geq m$. **b** Shows the truncated basis, which is obtained by truncating the basis functions with respect to active basis functions of finer levels. (Color figure online)

conveys that cut-element-specific conditioning problems are not essentially affected by the type of weak enforcement of the boundary conditions, provided that with a Nitsche-type enforcement of Dirichlet conditions coercivity of the variational form is retained [81,82]. The parameters β_h^λ and β_h^μ are chosen inversely proportional to the grid size. This is relevant in regard of multigrid techniques, since due to these parameters the operators $a_h (\cdot, \cdot)$ and $b_h (\cdot)$ are mesh-size dependent. The coarse problems in this contribution inherit the parameters in the weak formulation from the finest grid, see Remark 3.1 for details.

An advantageous property of immersed methods is that different approximation spaces can relatively easily be employed, by virtue of the regularity of the underlying mesh. We herein consider uniform discretizations with both traditional Lagrange basis functions and B-splines [83], see e.g., [84] for a detailed introduction into B-spline based finite element methods. In immersed finite element methods, it is generally not known a priori which regions of the grid require elevated approximation properties. Therefore it is important that immersed grids facilitate local refinement strategies, to provide sufficient accuracy in essential regions of the problem domain. For this reason [51] develops a preconditioning strategy that is tailored to immersed hp-adaptive C^0-bases, and in this contribution we consider the preconditioning of locally refined B-spline bases, by means of truncated hierarchical B-splines (THB-splines) [85]. THB-splines are a popular refinement strategy for B-splines in isogeometric analysis, and fit naturally into the framework of immersed methods. The construction of THB-splines is illustrated in

Fig. 2. THB-splines are posed on a hierarchy of meshes, each of which has a certain number of active B-splines. The truncated basis is then obtained by truncating active B-splines with respect to the active B-splines on the finer grids in which they are nested. We refer the reader to [85,86] for details regarding the construction of THB-splines. Truncating the basis functions reduces the computational cost, because the truncated basis functions have smaller supports than the standard (non-truncated) hierarchical basis functions. This results in a sparser system matrix. In the context of the multigrid preconditioner developed herein, THB-splines have a particularly important advantage concerning the computational cost. As will be elaborated in Sect. 3.4, the block selection for the Schwarz-type smoother in the developed preconditioner yields smaller blocks with THB-splines than it would with a non-truncated basis, and results in a nearly diagonal treatment of untrimmed basis functions. Furthermore, it is demonstrated in [73] that mesh-fitting systems with truncated bases are generally better conditioned than systems with non-truncated bases, and that multigrid methods with a diagonal smoother are effective for systems with THB-splines.

3 Multigrid methods for immersed finite element methods

This section presents the developed geometric multigrid preconditioner for immersed finite element methods. Section 3.1 discusses the conditioning effects of both cut elements

and the mesh size of the background grid. In Sect. 3.2 the employed preconditioning algorithm is introduced. Section 3.3 considers the prolongation and restriction operators, specifically in the context of locally refined grids, and effective smoothers for immersed methods are discussed in Sect. 3.4.

3.1 Conditioning aspects of immersed finite element methods

Introducing the basis $\{\boldsymbol{\phi}_i\}_{i=1}^n$ for the approximation space $\mathcal{V}_h$, the variational formulation in (2) leads to the linear system:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4}$$

with symmetric positive definite (SPD) matrix $\mathbf{A}$ with $A_{ij} = a_h\left(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j\right)$, solution vector $\mathbf{x}$ such that $\mathbf{u}_h = \sum_{i=1}^n x_i \boldsymbol{\phi}_i$, and right hand side vector $\mathbf{b}$, with $b_i = b_h\left(\boldsymbol{\phi}_i\right)$. Immersed finite element methods generally lead to systems of equations that are fundamentally difficult to solve. Without dedicated treatment of the cut-element-specific ill-conditioning of systems derived from immersed finite element methods, in general the convergence of iterative solvers is severely retarded [35,50,51]. An important indicator of the feasibility of iterative solution procedures for a linear system as in (4) is the condition number, $\kappa\left(\mathbf{A}\right)$. For the symmetric positive definite (SPD) systems considered in this contribution, the condition number is equal to the quotient of the largest to the smallest eigenvalue. It should be noted that the convergence of iterative solution methods is not merely dependent on the condition number, but rather depends on the entire spectrum of the system matrix, i.e., on the distribution of the complete set of eigenvalues. Systems with well-clustered spectra generally lead to faster convergence than systems with eigenvalues that are spread out. For a detailed discussion about the aspects affecting the performance of iterative solvers, the reader is directed to [52,87].

To elucidate the general characteristics of spectra emanating from immersed finite element methods, Fig. 3 displays the Jacobi-preconditioned spectrum of an immersed approximation of the Laplace operator on the star-shaped domain from Fig. 1, as well as 4 characteristic eigenmodes. The immersed approximation space is composed of quadratic Lagrange basis functions on a mesh with $h = \frac{1}{8}$. This domain and type of basis functions are used for all examples in this section, but similar results can be obtained with B-spline bases. Dirichlet boundary conditions are imposed by means of a penalty method with parameter $\frac{2}{h}$. Because symmetric positive definite (SPD) systems only posses real and positive eigenvalues, their spectra can be conveniently represented by plotting the eigenvalues $\lambda_i \in \mathbb{R}^+$ versus their index $i \in \mathbb{N}$, as in Fig. 3a. Figure 3b displays a very small eigenmode that is only supported on a small cut element, which is exemplary

for eigenmodes that are common in immersed finite element methods. As described in detail in [35], basis functions on small cut elements can become almost linearly dependent, which yields very small eigenvalues and is not repaired by Jacobi preconditioning. It can be observed that this eigenmode is very similar to the largest eigenmode of the system, plotted in Fig. 3c, as both eigenmodes consist of essentially the same basis functions. However, while the almost linearly dependent basis functions are subtracted from each other such that these cancel out in Fig. 3b, the opposite happens in Fig. 3c. Note that with the quadratic Lagrange basis, there are 4 almost linearly dependent basis functions that are only supported on the small cut elements, such that the largest eigenvalue is bounded from below by approximately 4. Based on an analysis of the typical small eigenmodes in immersed finite element methods, an estimate of the condition number of SPD systems for second-order PDEs is derived in [35]:

$$\kappa\left(\mathbf{A}\right) = \mathcal{O}\left(\eta^{-(2p+1-2/d)}\right), \tag{5}$$

with p the polynomial degree of the approximation space, d the number of dimensions and η the smallest volume fraction, defined as the smallest relative intersection of an element with the physical domain:

$$\eta = \min_{K_i \in \mathcal{T}_h | K_i \cap \Omega \neq \emptyset} \frac{|K_i \cap \Omega|}{|K_i|}. \tag{6}$$

Since cut elements can be arbitrarily small, systems derived from immersed finite element formulations can be arbitrarily ill-conditioned. As a result, iterative solvers are generally ineffective in case that no dedicated treatment for the cut-element-induced conditioning problem is applied, see e.g., the introduction in Sect. 1. Figure 3d and e portray characteristic eigenfunctions that are not exclusive to immersed finite element methods. The smooth eigenmode in Fig. 3d is very similar to the usual smallest eigenmode in mesh-fitting finite elements, and is in close correspondence with the smallest analytical eigenmode of the considered PDE. Figure 3e plots the oscillatory eigenfunction with the highest frequency that can be captured by the grid. This eigenmode is similar to the usual largest eigenmode in mesh-fitting systems. The ratio between the largest and smallest eigenvalue in mesh-fitting systems, and with that the condition number, is therefore mesh-size dependent. It can be shown that for second-order PDEs it holds that [53]:

$$\kappa\left(\mathbf{A}\right) = \mathcal{O}\left(h^{-2}\right). \tag{7}$$

This deteriorates the conditioning for very fine meshes, which also retards the convergence of iterative solvers. In particular, smooth eigenmodes as in Fig. 3d—which yield
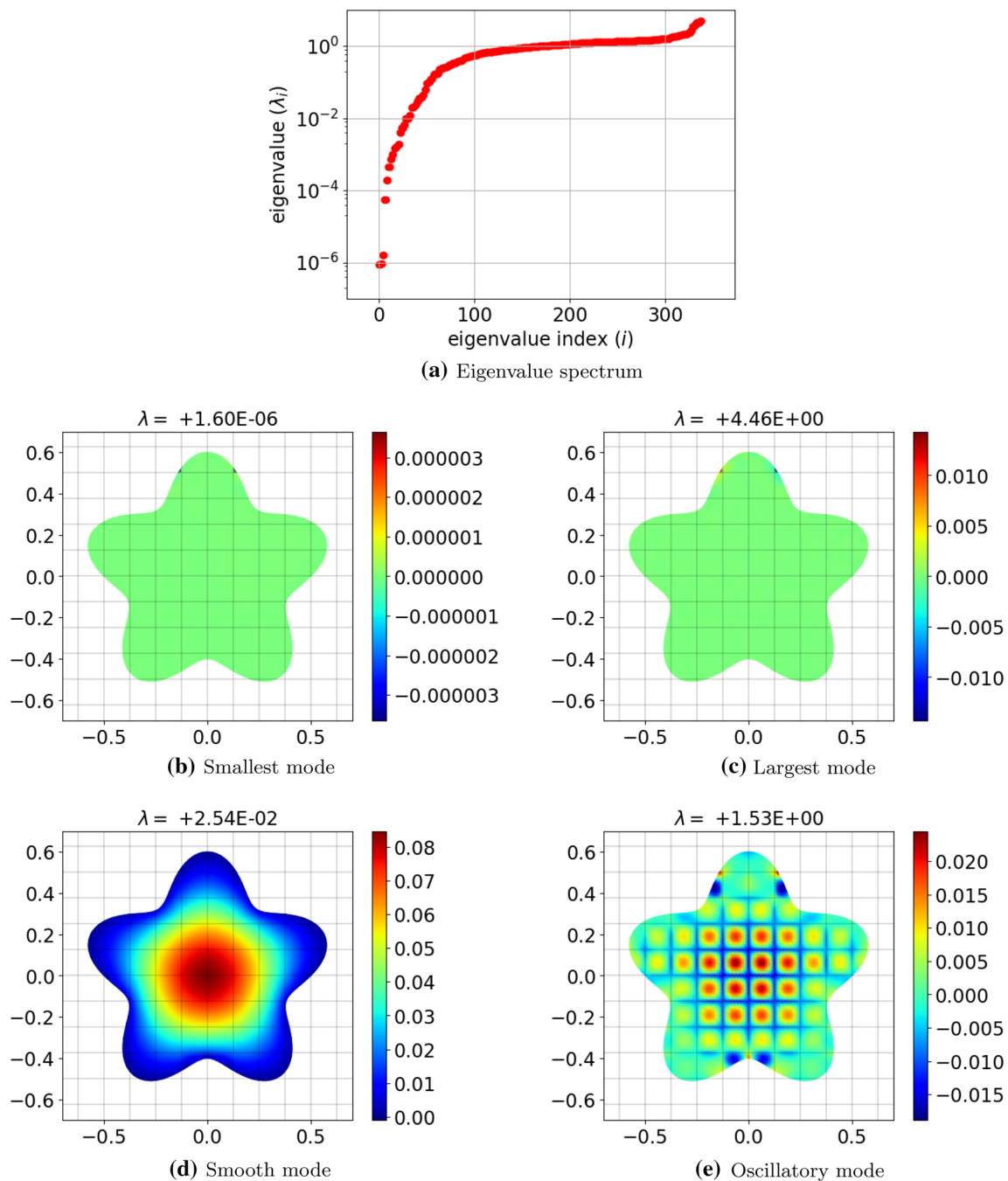
(a) Eigenvalue spectrum

(b) Smallest mode

(c) Largest mode

(d) Smooth mode

(e) Oscillatory mode

Fig. 3 Typical spectrum and characteristic eigenmodes of an immersed system that is preconditioned with Jacobi

small eigenvalues of $\mathcal{O}(h^2)$ with Jacobi preconditioning—converge slowly. For fixed point iteration methods a convergence rate of $1 - \mathcal{O}(h^2)$ can be derived [59], and also plain Krylov subspace methods generally require $\mathcal{O}\left(h^{-1}\right)$ iterations [52].

Multigrid methods are often applied to resolve the mesh-dependence of the condition number according to (7) and the corresponding slow convergence, and provide a conditioning and convergence rate that is independent of the mesh size. The reader is directed to [57–60] for reference works on multigrid techniques. In [50] an additive Schwarz preconditioner is presented that is tailored to immersed finite element methods, and resolves the conditioning problems related to cut elements. Furthermore, it is observed that the systems treated with this preconditioner behave similarly to mesh-fitting systems with respect to the grid size, both in terms of the condition number and in terms of the number of iterations with Krylov subspace methods. The computational cost of
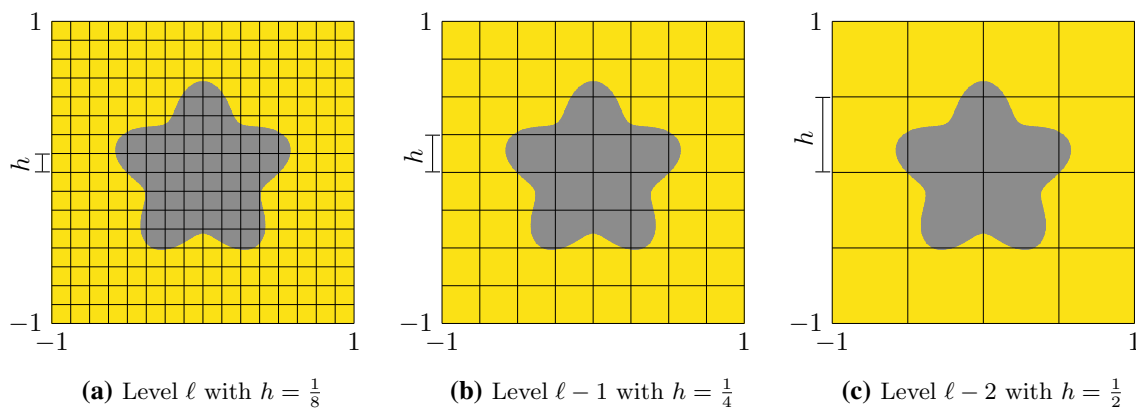
(a) Level ℓ with $h = \frac{1}{8}$ **(b)** Level $\ell - 1$ with $h = \frac{1}{4}$ **(c)** Level $\ell - 2$ with $h = \frac{1}{2}$

Fig. 4 A hierarchy of nested discretizations on which the problem in (2) can be solved

Schwarz-preconditioned iterative solvers scales better with the size of the system than direct solvers, but is suboptimal in the sense that it is not yet linear with the number of degrees of freedom. In the following sections we therefore incorporate aspects of this preconditioner in a multigrid framework, to obtain a solution method that is robust to cut elements and independent of the size of the system.

3.2 Multigrid V-cycle algorithm

This section introduces the elemental principles of multi-grid techniques, and presents the V-cycle algorithm that is employed in this contribution. We discuss the geometric multigrid method in the context of the correction scheme (CS), but the analyses, algorithms and results extend mutatis mutandis to the full approximation scheme (FAS).

We consider an algebraic system of the form (4). Let $\mathbf{x}'$ denote an approximation to the solution of (4) and let $\mathbf{r}$ denote the corresponding residual according to:

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}'. \tag{8}$$

To obtain the solution to (4), the approximation $\mathbf{x}'$ must be corrected as $\mathbf{x}' + \bar{\mathbf{x}}$ with:

$$\bar{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{r}. \tag{9}$$

Multigrid methods are based on the notion that if $\mathbf{A}$ derives from an elliptic operator, then $\mathbf{A}^{-1}$ can be approximated by a combination of fixed point iterations and a coarse grid correction. The fixed point iterations efficiently approximate oscillatory components of the solution, as in Fig. 3e. Since this reduces the oscillatory components of the error, these are commonly referred to as smoothing [59]. The coarse grid correction amends the approximation by the solution on a coarser grid, which enhances the approximation of smooth components of the solution, as in Fig. 3d. This again involves

the (approximate) inverse of a matrix, analogous to (9), but corresponding to a coarser grid. This inverse can likewise be approximated by smoothing operations and a coarse grid correction on an even coarser grid, see Fig. 4. In multigrid methods, the smoothing operations and coarse grid correction are therefore applied recursively, until a grid is reached that is sufficiently coarse to enable direct inversion at a negligible computational expense. The result of the multigrid cycle is denoted by $\tilde{\mathbf{x}}$ and approximates the vector $\bar{\mathbf{x}}$ in (9).

The multigrid V-cycle that is considered in this contribution is outlined in Algorithm 1. The input parameters of the algorithm are the number of (remaining) recursive multigrid levels ℓ, see Fig. 4, and the residual of the linear system $\mathbf{r}_\ell$ at the current level. It is emphasized that the number of multigrid levels follows a different convention than the number of hierarchical refinement levels in locally refined discretizations. The coarsest multigrid level in the V-cycle is $\ell = 1$, while the coarsest hierarchical refinement level is $m = 0$, see Fig. 2. The algorithm initializes the vector $\tilde{\mathbf{x}}_\ell$ in line 2, and in line 4 conducts a pre-smoothing step,[1] i.e., fixed point iteration, with $\gamma > 0$ denoting a relaxation parameter for stability and $\mathbf{M}_\ell^{-1}$ an approximate factorization or inverse of $\mathbf{A}_\ell$, e.g., Jacobi or Gauss–Seidel. Subsequently, smooth components of the error—which were not effectively reduced by the aforementioned smoothing operation—are treated by applying a coarse grid correction with coarser level $\ell - 1$ in lines 7–9. The recursive nature is implemented by applying the V-cycle also to obtain an approximate solution to the coarse grid correction problem. The direct solver in line 14 is only applied at the coarsest level $\ell = 1$. Note that the residual in the input of the algorithm should be interpreted as in (8) only on the finest level, and is a restriction of a finer residual in the recursive applications of the algorithm in line 8. To

---

[1] Note that in this contribution we only apply a single pre-smoothing and post-smoothing operation in each cycle, and do not consider the possibility of multiple smoothing operations.

enforce symmetry of the linear operator induced by the algorithm, the post-smoothing operation in line 12 is performed with the adjoint of $\mathbf{M}_\ell^{-1}$. This is relevant for Gauss–Seidel-type smoothers as these are generally nonsymmetric, and is realized by a reverse sweep. The approximate solution $\tilde{\mathbf{x}}_\ell$ to $\mathbf{A}_\ell^{-1}\mathbf{r}_\ell$ is returned in line 16. The coarse grid correction is treated in more detail in Sect. 3.3, and the smoothing operations are discussed in detail in Sect. 3.4.

Algorithm 1: V-cycle(ℓ, $\mathbf{r}_\ell$)

---

1 **if** $\ell > 1$ **then**

2     $\tilde{\mathbf{x}}_\ell = \mathbf{0}$ # initialize approximation of $\mathbf{A}_\ell^{-1}\mathbf{r}_\ell$

3   # pre-smooth

4     $\tilde{\mathbf{x}}_\ell = \tilde{\mathbf{x}}_\ell + \gamma\mathbf{M}_\ell^{-1}\mathbf{r}_\ell$ # smooth

5     $\mathbf{r}_\ell = \mathbf{r}_\ell - \mathbf{A}_\ell\tilde{\mathbf{x}}_\ell$ # update residual

6   # coarse grid correction

7     $\mathbf{r}_{\ell-1} = \mathbf{R}_\ell\mathbf{r}_\ell$ # restrict current residual to coarser grid

8     $\tilde{\mathbf{x}}_{\ell-1} = $ V-cycle($\ell - 1$, $\mathbf{r}_{\ell-1}$)   # compute correction

9     $\tilde{\mathbf{x}}_\ell = \tilde{\mathbf{x}}_\ell + \mathbf{R}_\ell^T\tilde{\mathbf{x}}_{\ell-1}$ # prolongate coarse grid correction

10     $\mathbf{r}_\ell = \mathbf{r}_\ell - \mathbf{A}_\ell\mathbf{R}_\ell^T\tilde{\mathbf{x}}_{\ell-1}$ # update residual

11   # post-smooth

12     $\tilde{\mathbf{x}}_\ell = \tilde{\mathbf{x}}_\ell + \gamma\mathbf{M}_\ell^{-T}\mathbf{r}_\ell$ # smooth

13 **else if** $\ell == 1$ **then**

14     $\tilde{\mathbf{x}}_\ell = \mathbf{A}_\ell^{-1}\mathbf{r}_\ell$ # direct solve at coarsest level

15 **end**

16 **return** $\tilde{\mathbf{x}}_\ell$

The multigrid V-cycle in Algorithm 1 can itself be applied as a solver, by iteratively performing the cycle to reduce the residual in every step, i.e., the approximation $\mathbf{x}'$ is simply updated by directly adding $\tilde{\mathbf{x}}$. This is generally true for multigrid cycles, e.g., also the W-cycle or FMG-cycle [59]. We consider the V-cycle, as this simple setup is already suitable to demonstrate the effectivity of the multigrid concept in immersed FEM. Additionally, when the pre-smoothing and post-smoothing operations are chosen such that these are adjoint, the V-cycle algorithm yields a symmetric positive definite (SPD) linear operator. This has the advantage that, instead of direct application of the V-cycle as a solver, it can also be employed as a preconditioner in a conjugate gradient (CG) algorithm. In the results presented in Sect. 4, this multigrid-preconditioned CG-solver is applied. The advantage of Krylov subspace solvers compared to multigrid as a standalone solver is that the convergence of Krylov methods is not purely governed by the smallest eigenmode in the system. Therefore, these are more robust to artifacts in the spectrum resulting from e.g., geometrical complexities such as the artificial coupling that is observed in Sect. 4.1 [88].

3.3 Restriction, prolongation, and coarse grid correction

The restriction and prolongation operations to communicate between different grid sizes, such as those in Fig. 4, are essential aspects of the coarse grid correction in lines 7–10 of Algorithm 1. Under the usual assumption that the grid size is doubled in the mesh coarsening, the grid lines of the coarser level $\ell - 1$ in Fig. 4b coincide with grid lines at the finer level ℓ in Fig. 4a. Therefore, the level $\ell - 1$ space is nested in the level ℓ space and the basis functions on level $\ell - 1$ can be represented identically by linear combinations of the basis functions on level ℓ. Denoting by $\boldsymbol{\Phi}_\ell$ and $\boldsymbol{\Phi}_{\ell-1}$ vectors of basis functions on level ℓ and level $\ell - 1$, respectively, there exists a matrix of coefficients $\mathbf{R}_\ell$ such that:

$$\boldsymbol{\Phi}_{\ell-1} = \mathbf{R}_\ell\boldsymbol{\Phi}_\ell. \tag{10}$$

The matrix $\mathbf{R}_\ell$ defines the restriction operator. This restriction operator is employed in line 7 to restrict the residual at level ℓ to the level $\ell - 1$ coarse mesh. In line 8 of Algorithm 1, the solution to the level $\ell - 1$ problem is approximated by recursive application of the V-cycle algorithm, except at the coarsest level $\ell = 1$ where a direct solution is carried out. The (approximate) solution to the level $\ell - 1$ problem constitutes the coarse grid correction, which is prolongated and added to $\tilde{\mathbf{x}}_\ell$ at level ℓ in line 9. Let us note, that by the nesting of the approximation spaces at the different levels, the prolongation from level $\ell - 1$ to level ℓ corresponds to injection. By virtue of relation (10) between the basis functions, we have the identities:

$$\tilde{\mathbf{x}}_{\ell-1}^T\boldsymbol{\Phi}_{\ell-1} = \tilde{\mathbf{x}}_{\ell-1}^T\left(\mathbf{R}_\ell\boldsymbol{\Phi}_\ell\right) = \left(\mathbf{R}_\ell^T\tilde{\mathbf{x}}_{\ell-1}\right)^T\boldsymbol{\Phi}_\ell \tag{11}$$

In terms of the coefficients, the prolongation thus corresponds to the adjoint of the restriction operator. In line 10 the level ℓ residual is updated after the coarse grid correction.

Remark 3.1 The coarse grid correction in Algorithm 1 is performed purely algebraically. Therefore, the coarse problem inherits the weak formulation from the fine problem, without adapting the mesh-dependent parameters in the operators in (3). In essence, the finer level basis functions are replaced by the coarser level basis functions, such that the system matrix and residual at the coarser level can simply be obtained as $\mathbf{A}_{\ell-1} = \mathbf{R}_\ell\mathbf{A}_\ell\mathbf{R}_\ell^T$ and $\mathbf{r}_{\ell-1} = \mathbf{R}_\ell\mathbf{r}_\ell$. An alternative approach is to adapt the operators in weak form (2) to the coarser grid, as in e.g., [89,90]. The implementation of the algebraic approach in Algorithm 1 is considerably simpler, however, and has been observed to adequately resolve the smooth eigenmodes in all considered cases. It should be noted that the extension of this simple algebraic coarsening approach
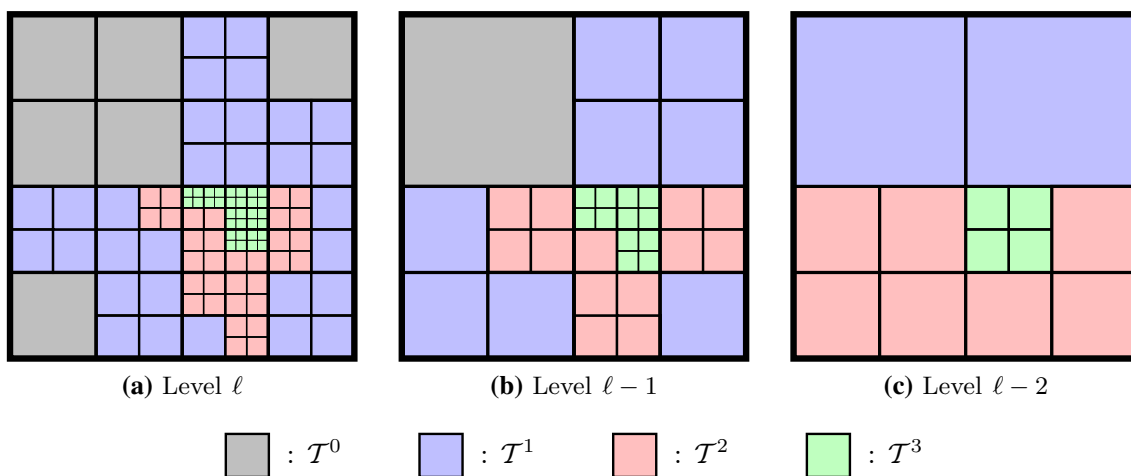
(a) Level ℓ **(b)** Level $\ell - 1$ **(c)** Level $\ell - 2$

$\blacksquare : \mathcal{T}^0$ $\blacksquare : \mathcal{T}^1$ $\blacksquare : \mathcal{T}^2$ $\blacksquare : \mathcal{T}^3$

Fig. 5 Hierarchy of nested, non-uniform meshes with $M = 3$ levels of local refinements. The different local refinement levels are indicated by the colors illustrated at the bottom. Note that the gray elements in **a** are of the same size as blue elements in **b** and the red elements in **c**, and that the green elements of the highest local refinement level have different sizes on the different grids. The grids contain, respectively, 100, 34, and 13 elements and the unrefined elements have a size of,

respectively, $\frac{1}{4}$, $\frac{1}{2}$ and 1 times the length of the domain. The coarser grid of level $\ell - 1$ in **b** is obtained by applying Algorithm 2 to the finer grid of level ℓ in **a**. Note that elements of level m in **b** only intersect elements of level m or lower in **a**. Recursively applying the algorithm to the grid of level $\ell - 1$ results in the coarsest grid of level $\ell - 2$ in **c**. Note that this coarsest mesh does not contain active unrefined gray elements of refinement level $m = 0$. (Color figure online)

to other grid-size dependencies is not verified in this contribution. Examples of other grid-size dependencies in the weak formulation are the stabilization parameter in Nitsche's method, see [81,82] and for spectral effects specifically [91], and the ghost penalty, see [4,37].

We have so far restricted ourselves to uniform meshes. For hierarchically refined meshes, as illustrated in Fig. 5a, the coarsening procedure in the multigrid method requires reconsideration. To enable an adequate approximation of functions that are smooth with respect to the local mesh width on level ℓ, we apply a type of hierarchical derefinement to obtain a nested coarse space on level $\ell - 1$. In practice, the mesh on level $\ell - 1$ is the coarsest mesh for which the mesh of level ℓ can be obtained by a single level of hierarchical refinements, i.e., one uniform subdivision of a certain set of elements. The construction of such non-uniform coarser spaces is summarized in Algorithm 2, and illustrated in Fig. 5. The algorithm denotes components of the mesh at level ℓ by: $K_i^m \in \mathcal{T}_h^m \subset \mathcal{T}_h$. As introduced in Fig. 2, K_i^m denotes an active element at local refinement level m with index i. The set of all active elements of local refinement level m is denoted by $\mathcal{T}_h^m = \{K_i^m\}$, and $\mathcal{T}_h = \cup_{m=0}^{m=M} \mathcal{T}_h^m$ denotes the full mesh, with M denoting the number of local refinement levels. Elements of the coarser mesh at level $\ell - 1$ are denoted as $k_i^m \in \mathcal{T}_{2h}^m \subset \mathcal{T}_{2h}$. The subscript $2h$ for the coarse mesh is to be conceived of as a symbolic notation. Note that the elements in $\mathcal{T}_{2h}^{m+1}$ are of the same size as the elements in $\mathcal{T}_h^m$. In line 1 of the algorithm, the coarser mesh is initialized as a uniform mesh with only unrefined elements of local refinement level $m = 0$, i.e., $\mathcal{T}_{2h} = \mathcal{T}_{2h}^0$. Line 3 initiates a

loop over the refinement levels $0 \leq m \leq M - 1$. Within this loop, the algorithm loops over all the elements $k_i^m \in \mathcal{T}_{2h}^m$, that are currently m times refined. If k_i^m intersects an element of $\mathcal{T}_h$ that is refined more than m times, element k_i^m is refined in lines 7–9. Note that line 7 abuses notation to simplify the expression, and that the refining of element k_i^m implies removing k_i^m from $\mathcal{T}_{2h}^m$ and adding the refinements to $\mathcal{T}_{2h}^{m+1}$. Note that this hierarchical derefinement procedure differs from the approach in [73], which employs an existing hierarchy of refined grids for the coarse grid corrections.

Algorithm 2: Coarsen($\mathcal{T}_h$)

---

1   initialize $\mathcal{T}_{2h}$ # initialize coarse mesh $\mathcal{T}_{2h} = \mathcal{T}_{2h}^0$ with uniform unrefined elements

2   # loop over local refinement levels
3   **for** $m \in \{0, ..., M - 1\}$ **do**

4   # loop over coarse mesh elements of level m
5   **for** $k_i^m \in \mathcal{T}_{2h}^m$ **do**

6   # check if k_i^m intersects elements of fine mesh $\mathcal{T}_h$ of local refinement level $> m$
7   **if** $\left(\cup_{\tilde{m}=m+1}^{\tilde{m}=M} \mathcal{T}_h^{\tilde{m}} \right) \cap k_i^m \neq \emptyset$ **then**

8   refine k_i^m

9   **end**

10   **end**

11   **end**

12   **return** $\mathcal{T}_{2h}$

3.4 Smoothers for immersed finite element methods

In lines 4 and 12 of Algorithm 1 smoothing operations, i.e., fixed point iterations, are performed to resolve the components of the error that cannot be adequately captured by the coarse grid. Effective application of the multigrid algorithm requires that the eigenvalues of $\gamma \mathbf{M}^{-1}\mathbf{A}$ that correspond to non-smooth eigenfunctions are close to 1. Note that in the formulations from here on, the subscripts indicating the level in the multigrid solver are omitted to simplify the notation. All these formulations are independent of the level ℓ, however. Stability of fixed point iterations requires:

$$0 \le \lambda_{\min}\left(\gamma \mathbf{M}^{-1}\mathbf{A}\right) \le \lambda_{\max}\left(\gamma \mathbf{M}^{-1}\mathbf{A}\right) \le 2, \qquad (12)$$

with $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denoting the smallest and largest eigenvalues, such that the spectral radius of the fixed point iteration is bounded:

$$\rho\left(\mathbf{I} - \gamma \mathbf{M}^{-1}\mathbf{A}\right) < 1, \qquad (13)$$

with $\mathbf{I}$ denoting the identity matrix with the same size as system matrix $\mathbf{A}$. In the case that $\lambda_{\max}\left(\gamma \mathbf{M}^{-1}\mathbf{A}\right) > 2$, the error component in the direction of the eigenvector corresponding to the largest eigenvalue will increase with smoothing, which may result in divergence. For this reason, smoothers such as Jacobi and additive Schwarz require a sufficiently small relaxation parameter γ. Smoothers such as Gauss–Seidel and multiplicative Schwarz are unconditionally stable and do not require relaxation, see e.g., [52, Theorems 4.10 and 14.9]. As already mentioned in the description of Algorithm 1 in Sect. 3.2, it should be noted that symmetry of the linear operator that is induced by the V-cycle insists that the post-smoothing operation is the adjoint of the pre-smoothing operation. Furthermore, it should be mentioned that the computational efficiency can potentially be improved by performing multiple smoothing operations in each cycle. Such enhancements are, however, not considered in this contribution.

Jacobi iterations are not suitable as a smoother for immersed finite elements, which is illustrated by the example in Fig. 3. The smallest eigenmode in Fig. 3b with a very small eigenvalue is barely affected by the smoothing, and cannot be captured on a coarser grid. Furthermore, the relatively large eigenmodes caused by almost linear dependencies as in Fig. 3c impose a small relaxation parameter, which further impairs the conditioning. The following subsections examine the suitability of a Gauss–Seidel smoother and Schwarz-type smoothers based on the preconditioner for immersed finite elements developed in [50].

Gauss–Seidel

A typical spectrum and characteristic eigenmodes with standard Gauss–Seidel preconditioning are shown in Fig. 6. Similar to Fig. 3 for Jacobi preconditioning, these figures correspond to the Laplace operator on the geometry in Fig. 1 with quadratic Lagrange basis functions and boundary conditions imposed by the penalty method. To obtain a symmetric preconditioner, a double fixed point iteration with adjoint Gauss–Seidel operations is applied in these figures:

$$\left(\mathbf{I} - \mathbf{M}^{-\mathrm{T}}\mathbf{A}\right)\left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)\mathbf{y}_{\lambda_i} = (1 - \lambda_i)\,\mathbf{y}_{\lambda_i}, \qquad (14)$$

with $\left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\right)$ corresponding to the initial Gauss–Seidel sweep, $\left(\mathbf{I} - \mathbf{M}^{-\mathrm{T}}\mathbf{A}\right)$ corresponding to the reverse sweep, and $\mathbf{y}_{\lambda_i}$ denoting the eigenvector corresponding to the ith eigenvalue λ_i. Hence, Fig. 6 presents the eigenmodes of the system $\left(\mathbf{M}^{-1} + \mathbf{M}^{-\mathrm{T}} - \mathbf{M}^{-\mathrm{T}}\mathbf{A}\mathbf{M}^{-1}\right)\mathbf{A}$. As will follow in (15), this is actually very similar to the V-cycle, except for the omission of the coarse grid correction. To reduce the computational cost, the Gauss–Seidel routine is implemented with a graph coloring algorithm[2] [92], which has a negligible effect on the spectrum. In Fig. 6b a very small eigenvalue is plotted, which is similar to the smallest eigenmode with Jacobi preconditioning in Fig. 3b. Note that the eigenvalues of these eigenfunctions differ by a factor of (approximately) 2, which is an expected consequence of the double iteration with Gauss–Seidel versus the single iteration with Jacobi. In Fig. 6c it is shown that also the spectrum with Gauss–Seidel preconditioning contains an eigenmode that is in close correspondence with the smallest analytical eigenmode of the PDE, and is similar to the usual smallest eigenmode with mesh-fitting techniques. Since the colors are selected such that basis functions of the same color do not intersect, the unit vectors corresponding to basis functions with the last color of the graph coloring algorithm yield an eigenspace with an eigenvalue of exactly 1. This can be observed in the spectrum in Fig. 6a and is illustrated in Fig. 6d.

Gauss–Seidel is not suitable as a smoother for immersed finite element methods, despite the unconditional stability by which—in contrast to Jacobi—it does not require more

---

[2] This algorithm divides the basis functions in sets, or colors, such that the supports of basis functions with the same color do not intersect.
Footnote 2 continued
The union over the sets of all colors constitutes the full approximation space. Note that, for Lagrange basis functions on uniform grids, this requires $(p + 1)^d$ different colors. As basis functions of the same color do not intersect, updating the approximation of the solution, $\tilde{\mathbf{x}}$, at an index corresponding to a certain color, does not affect the residual, $\mathbf{r}$, at the other indices corresponding to that color. This enables a Gauss–Seidel routine that sweeps over all indices of a certain color at once, instead of sequentially updating each index of the approximation of the solution and updating residual accordingly.
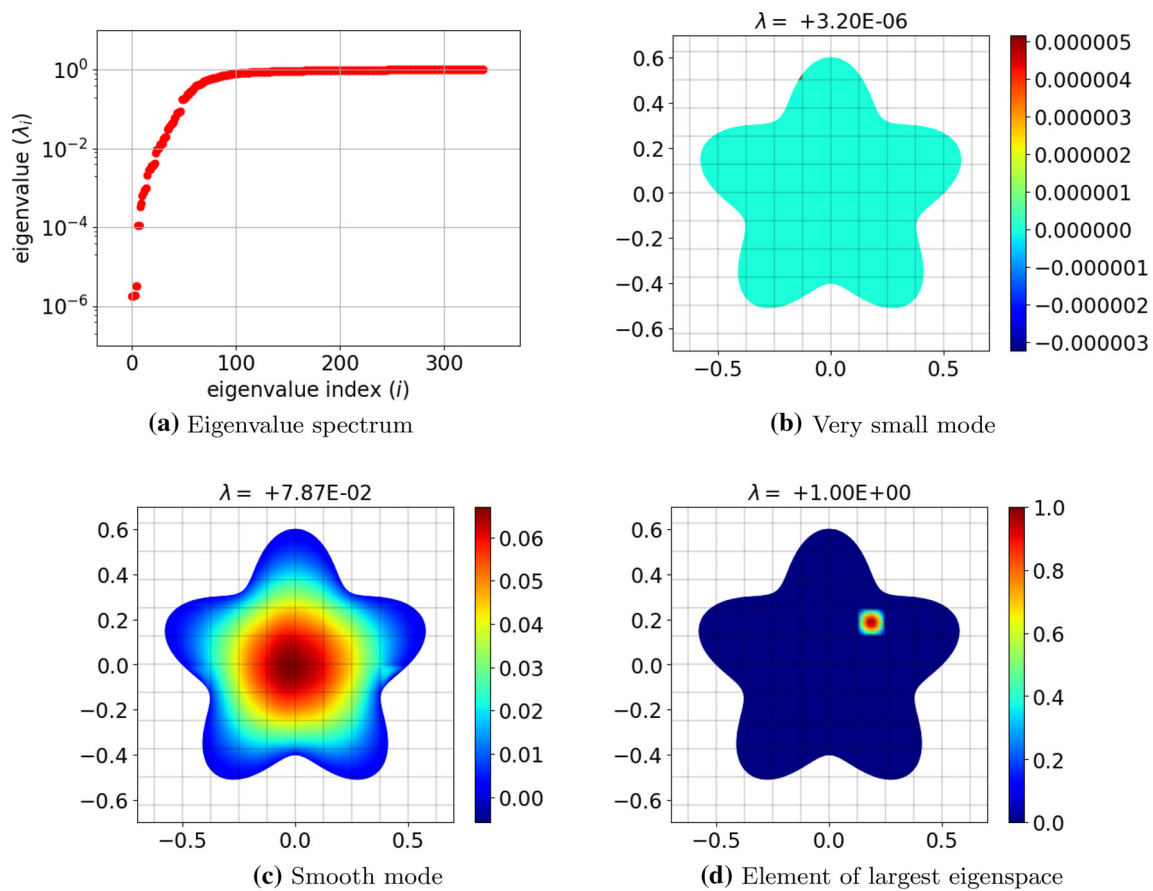
(a) Eigenvalue spectrum

(b) Very small mode

(c) Smooth mode

(d) Element of largest eigenspace

Fig. 6 Typical spectrum and characteristic eigenmodes of an immersed system that is preconditioned by a double fixed point iteration with Gauss–Seidel

relaxation in an immersed setting than in a mesh-fitting setting. The problem that renders Gauss–Seidel ineffective for immersed finite element methods is the small eigenmode plotted in Fig. 6b. Similar to Jacobi, the eigenvalue of this mode is too small to be adequately treated by the smoothing operations, and it is also not resolved by the coarse grid correction. This is shown in Fig. 7, which displays the spectrum of the same problem preconditioned by the V-cycle in Algorithm 1 with $\ell = 2$ levels and a single Gauss–Seidel sweep as smoother. This eigenvalue problem can be formulated as:

$$\left(\mathbf{I} - \mathbf{M}_\ell^{-\mathrm{T}}\mathbf{A}_\ell\right)\left(\mathbf{I} - \mathbf{R}_\ell^{\mathrm{T}}\mathbf{A}_{\ell-1}^{-1}\mathbf{R}_\ell\mathbf{A}_\ell\right)\left(\mathbf{I} - \mathbf{M}_\ell^{-1}\mathbf{A}_\ell\right)\mathbf{y}_{\lambda_i}$$
$$= (1 - \lambda_i)\,\mathbf{y}_{\lambda_i}, \tag{15}$$

or as:

$$\text{V-cycle}\!\left(\ell = 2, \mathbf{r}_\ell = \mathbf{A}\mathbf{y}_{\lambda_i}\right) = \lambda_i\mathbf{y}_{\lambda_i}. \tag{16}$$

While a comparison of the spectra in Figs. 6a and 7a, i.e., without and with the coarse grid correction, reveals that several small eigenmodes are resolved by the coarse grid correction, Figs. 6b and 7b demonstrate that both spectra contain

approximately the same small eigenmode. The eigenvalues of this mode are barely affected by the coarse grid correction, with eigenvalue 3.20×10^{-6} for the double fixed point iteration and eigenvalue 3.78×10^{-6} for the full V-cycle.

Additive Schwarz

The spectra and eigenfunctions with Jacobi and Gauss–Seidel in Figs. 3 and 6 clearly demonstrate that these are not robust to cut elements. This is consistent with the analysis of the conditioning problems in [35], which points out that diagonal preconditioners do not adequately mitigate the almost linear dependencies that occur in immersed finite element methods. In [50] it is derived that almost linearly dependent basis functions can be effectively treated collectively, when these are inverted in a block manner by a Schwarz-type method. Based on the additive Schwarz lemma, [50] shows that additive Schwarz preconditioning is actually a very natural approach to resolve the small eigenmodes caused by almost linear dependencies. Furthermore, it is demonstrated that in terms of the condition number and the number of iterations in an iterative solution method, immersed meth-
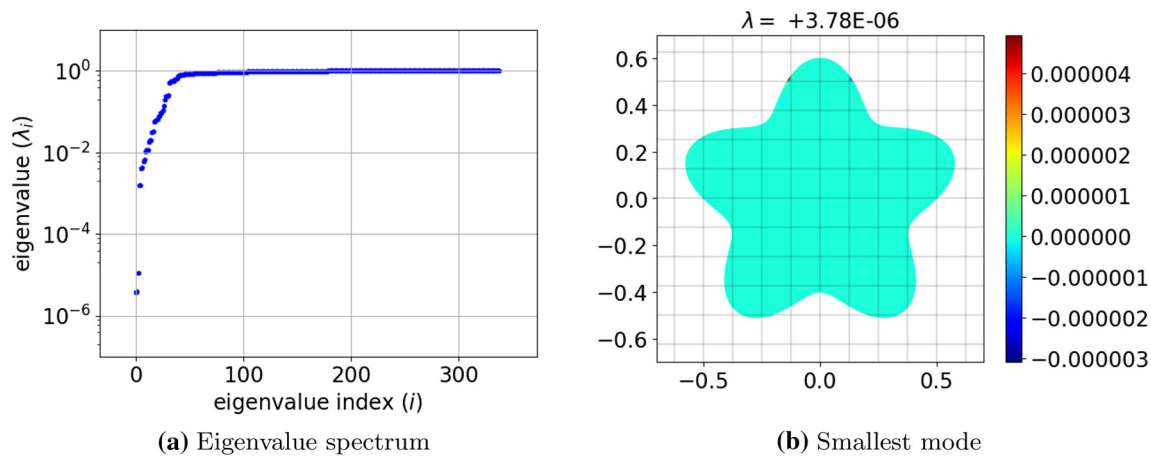
(a) Eigenvalue spectrum

(b) Smallest mode

Fig. 7 Spectrum and a characteristic very small eigenmode of an immersed system that is preconditioned by a two-level V-cycle with Gauss–Seidel as smoother

ods with additive Schwarz preconditioning behave similar to mesh-fitting techniques. This section therefore examines the spectrum of immersed methods with additive Schwarz preconditioning, to establish the suitability as a smoother in a multigrid method.

In additive Schwarz preconditioning, a set of N index blocks is selected, which correspond to sets of basis functions. For each index block $j \leq N$, the system matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is restricted to the indices in the block, denoted by $\mathbf{A}_j \in \mathbb{R}^{n_j \times n_j}$. The block matrices are then inverted and prolongated to a matrix of size $n \times n$. The additive Schwarz preconditioner is obtained by summing these matrices:

$$\mathbf{M}^{-1} = \sum_{j=1}^{N} \mathbf{P}_j \underbrace{\left(\mathbf{P}_j^{\mathrm{T}} \mathbf{A} \mathbf{P}_j \right)^{-1}}_{\mathbf{A}_j^{-1}} \mathbf{P}_j^{\mathrm{T}}, \tag{17}$$

with $\mathbf{P}_j \in \mathbb{R}^{n \times n_j}$ a matrix that prolongates a block vector $\mathbf{y}_j \in \mathbb{R}^{n_j}$ to a vector $\mathbf{P}_j \mathbf{y}_j = \mathbf{y} \in \mathbb{R}^n$ – with nonzero entries only at the indices in block j—and the transpose of $\mathbf{P}_j$ a restriction operator that restricts a vector $\mathbf{z} \in \mathbb{R}^n$ to a block vector $\mathbf{P}_j^{\mathrm{T}} \mathbf{z} = \mathbf{z}_j \in \mathbb{R}^{n_j}$—containing only the indices in block j.

An essential aspect of additive Schwarz preconditioners is the choice of the index blocks. It is pointed out in [50] that almost linearly dependent basis functions are required to be in an index block together. Furthermore, it is demonstrated that devising a block for each cut element with all basis functions supported on it is an effective strategy to satisfy this requirement for uniform grids. As demonstrated in [51], however, it is not trivial to generalize this concept to locally refined meshes. Therefore this contribution applies an alternative strategy to select the Schwarz blocks based on so-called encapsulating supports, which is inspired by the Schwarz-type smoother developed for divergence-

conforming discretizations in [71]. Accordingly, for every basis function a block is devised, containing all the basis functions whose support completely lies inside the support of the basis function associated to the block, see Fig. 8. Note that in this contribution the support of a basis function refers to the support within the physical domain. The block that is associated to function ϕ_j is defined as:

$$\left\{ \phi_k \ : \ \mathrm{supp}_{\Omega} \left(\phi_k \right) \subseteq \mathrm{supp}_{\Omega} \left(\phi_j \right) \right\}, \tag{18}$$

with $\mathrm{supp}_{\Omega}(\phi_k)$ denoting the support of basis function ϕ_k within physical domain Ω. For vector-valued problems, separate blocks are devised for basis functions describing different vectorial components of the solution, similar to the blocks in [50]. By construction, each block therefore contains the basis function associated to it, and for untrimmed (truncated hierarchical) B-splines this approach yields an approximately diagonal preconditioner. Let us note here the importance of the truncation of the basis functions in the locally refined approximations. As non-truncated hierarchical bases yield a very large number of basis functions with overlapping supports, this correspondingly results in very large blocks in the additive Schwarz preconditioner, which leads to significant computational costs. Trimmed basis functions on small cut elements—which can be almost linearly dependent—are also assigned to blocks associated to other functions. This satisfies the requirement formulated in [50] that almost linearly dependent basis functions need to be in a block together, and therefore resolves the small eigenmodes that are characteristic for immersed finite element methods. This strategy to select the Schwarz blocks is directly applicable to both B-splines on uniform grids and to truncated hierarchical B-splines on non-uniform grids, in contrast to the element-wise strategy in [50]. It is, however, not natural to directly apply this strategy to Lagrange basis functions,

Fig. 8 Greville abscissae and supports of quadratic B-splines assigned to a block. **a** Displays the support within the physical domain of the function associated to the block in orange. **b** Displays the supports of the other functions assigned to the block in purple. The fictitious support of the functions is not considered, but is hatched to increase the clarity of the figures. Note that the supports of the functions in **b** completely lie inside the support of the function in **a**. (Color figure online)

(a) Function associated to the block

(b) Other functions assigned to the block
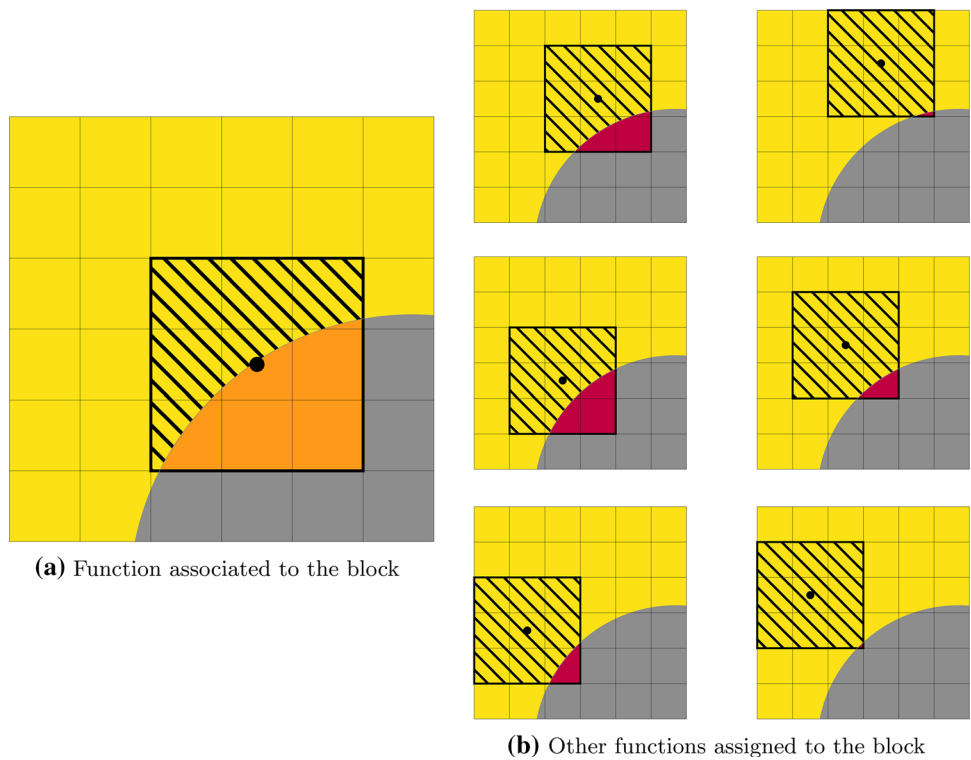
as these are not all supported on the same number of elements. For the Lagrange bases, blocks are therefore only devised for the nodal basis functions.[3] This implies that with Lagrange bases a block is devised for every cluster of 2^d elements. Note that this yields approximately the same number of blocks for uniform Lagrange bases as for uniform B-spline bases and even an identical treatment in case of linear basis functions, but, in contrast, does not reduce to a purely diagonal treatment of untrimmed basis functions for higher-order Lagrange bases. It should be mentioned that the block selection described here is not the only possible and effective method to select blocks for immersed finite elements, and interested readers are directed to [69,70,72] for a study of suitable block selections in isogeometric analysis and to the reference works [56,93] for considerations regarding the block selections with traditional finite element bases.

Efficiency and stability of additive Schwarz as a smoother requires adequate selection of the relaxation parameter γ. For the smoothing operations to efficiently reduce the error components in the directions of modes with eigenfunctions that can not be adequately captured on coarser grids, it is

required that the relaxed eigenvalues $\gamma \lambda_i \left(\mathbf{M}^{-1} \mathbf{A} \right)$ corresponding to such eigenmodes are close to 1. The requirement with regard to stability is formulated in (12), and states that $\lambda_{\min} \left(\mathbf{M}^{-1} \mathbf{A} \right) \geq 0$ and that $\gamma \lambda_{\max} \left(\mathbf{M}^{-1} \mathbf{A} \right) \leq 2$. The positivity of the eigenmodes follows from the symmetric positive definiteness of both $\mathbf{M}^{-1}$ and $\mathbf{A}$. Since the eigenvalues of $\mathbf{M}^{-1} \mathbf{A}$ coincide with the eigenvalues of $\mathbf{M}^{-\frac{1}{2}} \mathbf{A} \mathbf{M}^{-\frac{1}{2}}$, the eigenmodes can be bounded from above by:

$$\lambda_{\max} \left(\mathbf{M}^{-1} \mathbf{A} \right) = \lambda_{\max} \left(\mathbf{M}^{-\frac{1}{2}} \mathbf{A} \mathbf{M}^{-\frac{1}{2}} \right)$$

$$= \max_{\mathbf{y}} \frac{\mathbf{y}^{\mathrm{T}} \mathbf{M}^{-\frac{1}{2}} \mathbf{A} \mathbf{M}^{-\frac{1}{2}} \mathbf{y}}{\mathbf{y}^{\mathrm{T}} \mathbf{y}} = \max_{\mathbf{z}} \frac{\mathbf{z}^{\mathrm{T}} \mathbf{A} \mathbf{z}}{\mathbf{z}^{\mathrm{T}} \mathbf{M} \mathbf{z}} \tag{19a}$$

$$= \max_{\mathbf{z}} \frac{\mathbf{z}^{\mathrm{T}} \mathbf{A} \mathbf{z}}{\min_{\sum_{j=1}^{N} \mathbf{P}_j \mathbf{z}_j = \mathbf{z}} \sum_{j=1}^{N} \mathbf{z}_j^{\mathrm{T}} \mathbf{P}_j^{\mathrm{T}} \mathbf{A} \mathbf{P}_j \mathbf{z}_j} \tag{19b}$$

$$= \max_{\{\mathbf{z}_j\}_{j=1}^{N}} \frac{\left(\sum_{j=1}^{N} \mathbf{P}_j^{\mathrm{T}} \mathbf{z}_j^{\mathrm{T}} \right) \mathbf{A} \left(\sum_{j=1}^{N} \mathbf{P}_j \mathbf{z}_j \right)}{\sum_{j=1}^{N} \mathbf{z}_j^{\mathrm{T}} \mathbf{P}_j^{\mathrm{T}} \mathbf{A} \mathbf{P}_j \mathbf{z}_j} \tag{19c}$$

$$\leq \max_{K_i} \max_{\{\mathbf{z}_j\}_{j=1}^{N_{K_i}}} \frac{\left(\sum_{j=1}^{N_{K_i}} \mathbf{P}_j^{\mathrm{T}} \mathbf{z}_j^{\mathrm{T}} \right) \mathbf{A}^{K_i} \left(\sum_{j=1}^{N_{K_i}} \mathbf{P}_j \mathbf{z}_j \right)}{\sum_{j=1}^{N_{K_i}} \mathbf{z}_j^{\mathrm{T}} \mathbf{P}_j^{\mathrm{T}} \mathbf{A}^{K_i} \mathbf{P}_j \mathbf{z}_j} \tag{19d}$$

$$\leq \max_{K_i} \max_{\{\mathbf{z}_j\}_{j=1}^{N_{K_i}}} \frac{N_{K_i} \sum_{j=1}^{N_{K_i}} \mathbf{z}_j^{\mathrm{T}} \mathbf{P}_j^{\mathrm{T}} \mathbf{A}^{K_i} \mathbf{P}_j \mathbf{z}_j}{\sum_{j=1}^{N_{K_i}} \mathbf{z}_j^{\mathrm{T}} \mathbf{P}_j^{\mathrm{T}} \mathbf{A}^{K_i} \mathbf{P}_j \mathbf{z}_j} = \max_{K_i} N_{K_i}, \tag{19e}$$

---

[3] In this contribution different Lagrange basis functions are indicated as nodal, edge, face and volume functions. With d denoting the number of dimensions: *nodal* functions attain the value 1 at a vertex of the grid and span 2^d elements, *edge* functions attain the value 1 on an edge and span 2^{d-1} elements, *face* functions attain the value 1 on a face and span 2^{d-2} elements, and *volume* or *element internal* functions attain the value 1 inside an element and span 1 element. Note that face functions are not considered in the two-dimensional example.

(a) Eigenvalue spectrum

(b) Smooth mode (smallest)

(c) Nodal mode

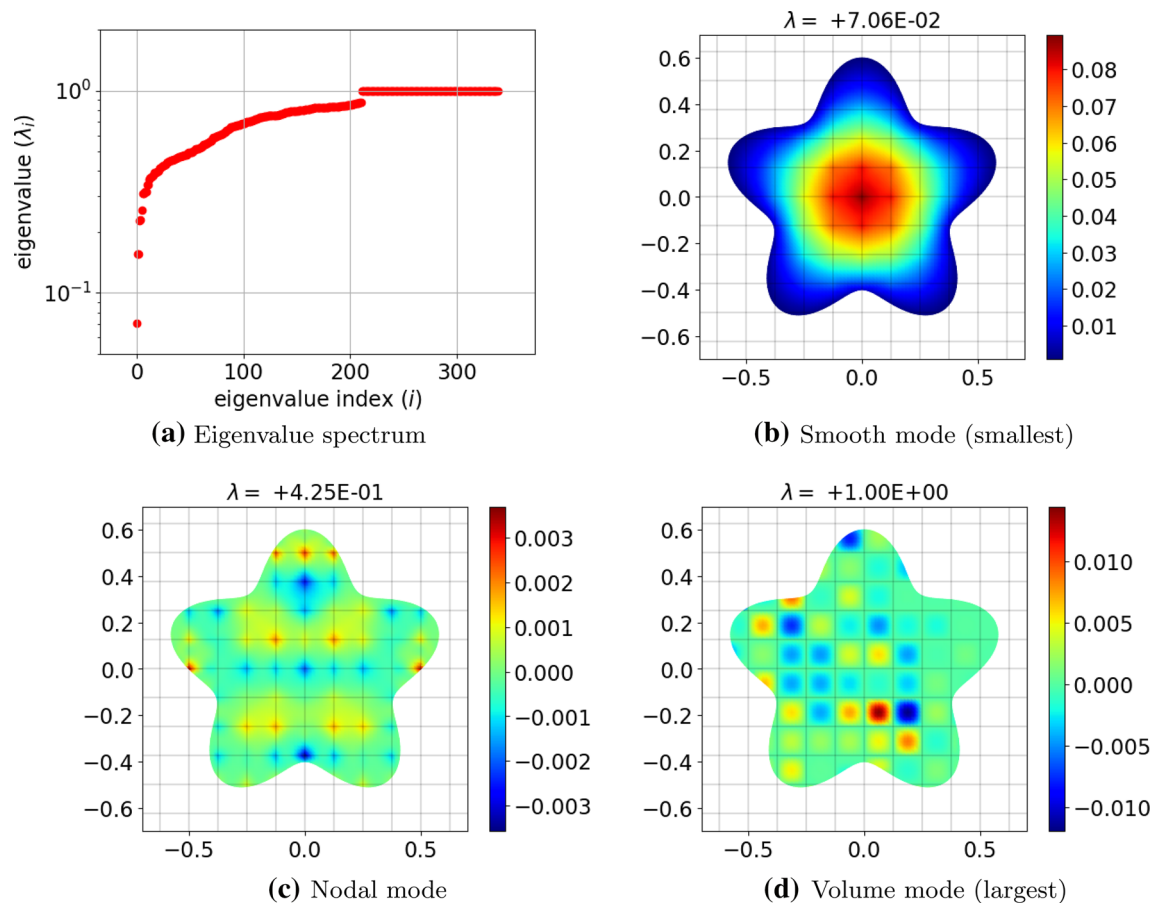
(d) Volume mode (largest)

Fig. 9 Typical spectrum and characteristic eigenmodes of an immersed system that is preconditioned by a double fixed point iteration with additive Schwarz

with $\mathbf{A}^{K_i} \in \mathbb{R}^{n \times n}$ the part of system matrix $\mathbf{A}$ that results from the integration over element K_i, and N_{K_i} denoting the number of blocks containing basis functions that are supported on this element. Note that in (19b) the additive Schwarz lemma is applied, see e.g., [56,93]. In (19d) the maximal quotient with system matrix $\mathbf{A}$ is replaced by the maximum over the quotients with the element contributions $\mathbf{A}^{K_i}$, and in (19e) the Cauchy-Schwarz inequality is applied. For Lagrange basis functions this bound is observed to be considerably sharp, as volume basis functions are contained in N_{K_i} blocks and form eigenfunctions similar to the one in Fig. 9d. These functions are not captured in the coarse grid correction, such that efficient smoothing requires $\gamma = N_{K_i}^{-1} = 4^{-1}$ for a two-dimensional Lagrange basis. This relaxation parameter yields $\gamma \lambda_{\max} \left(\mathbf{M}^{-1}\mathbf{A} \right) \leq 1 < 2$, such that also the stability condition is satisfied.

Figure 9 presents the spectrum and characteristic eigenmodes of the Laplace operator on the geometry in Fig. 1 with a quadratic Lagrange basis that is preconditioned by a double fixed point iteration with additive Schwarz and the relaxation parameter $\gamma = \frac{1}{4}$. The double fixed point iteration is applied such that later on these results can eas-

ily be related to the results with the V-cycle with additive Schwarz smoothing in Fig. 10. Figure 9b shows that the smallest eigenvalue in the system with additive Schwarz does not correspond to an eigenfunction on a small cut element. Instead, the smallest eigenmode is similar to the usual smallest eigenmode with mesh-fitting methods, which can be considered as the smoothest possible mode satisfying the boundary conditions. Figure 9c plots an example of an eigenfunction that almost entirely consists of nodal basis functions. The spectrum contains multiple of such modes, and these are important because these are the smallest eigenmodes that cannot be adequately captured by the coarse grid correction. Therefore, these nodal modes form the bottleneck for the condition number of immersed systems preconditioned by the V-cycle with additive Schwarz smoothing, as will follow in Fig. 10. In fact, the eigenvalue of such modes can be clarified. Since nodal basis functions are in only 1 index block, a fixed point iteration reduces the contribution of such functions by approximately a factor $1 - \gamma = \frac{3}{4}$. The double fixed point iteration therefore results in an eigenvalue of approximately $1 - (1 - \gamma)^2 = \frac{7}{16} \approx 0.438$. The largest eigenmodes in the spectrum have an eigenvalue of approxi-
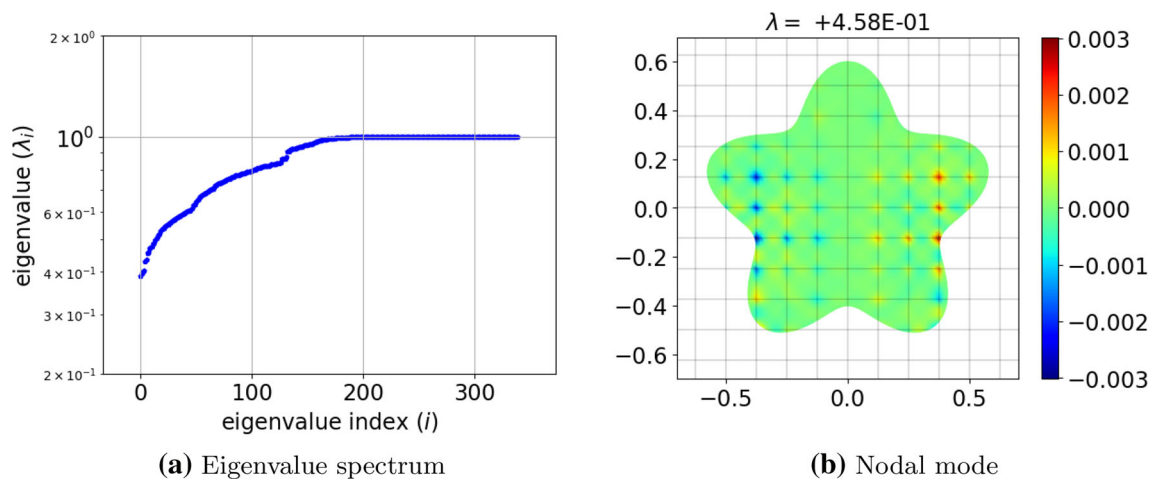
(a) Eigenvalue spectrum

(b) Nodal mode

Fig. 10 Spectrum and eigenmode of an immersed system that is preconditioned by a two-level V-cycle with additive Schwarz as smoother

(a) 32×32 elements
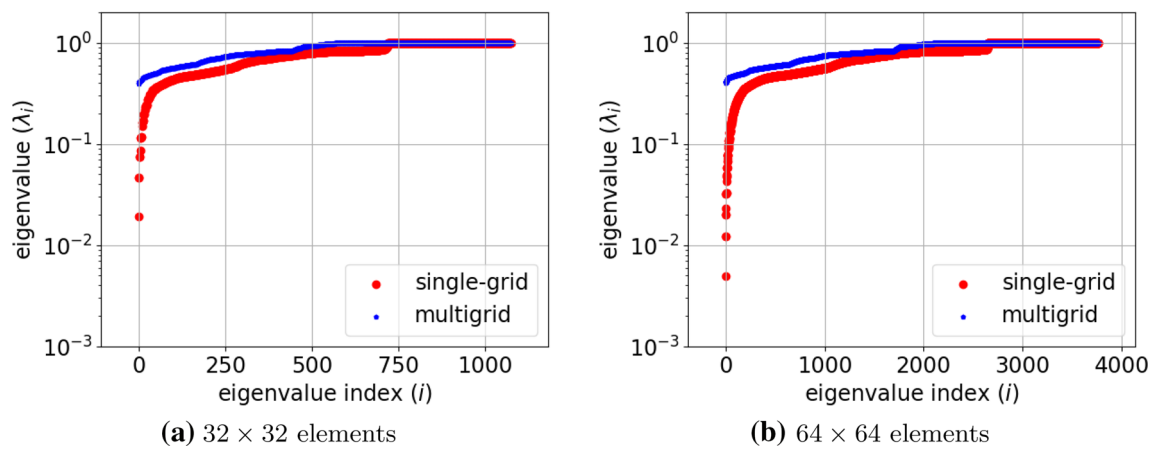
(b) 64×64 elements

Fig. 11 Spectra of immersed systems that are preconditioned by a double fixed point iteration with additive Schwarz (single-grid) and by a two-level V-cycle with additive Schwarz as smoother (multigrid)

mately 1, and correspond to eigenfunctions consisting almost entirely out of basis functions that are only supported on 1 element and therefore are contained in 4 index blocks. On the interior this only involves volume basis functions, resulting in the volume mode in Fig. 9d. By virtue of the eigenvalue of approximately 1, error components in the direction of the eigenvectors of these modes are effectively eliminated by the additive Schwarz smoother.

Based on the smallest eigenmodes in the spectrum with additive Schwarz, this technique is suitable as a smoother in a multigrid method for immersed finite elements. The results of this smoother in a two-level V-cycle are presented in Fig. 10. It can be observed from the spectra in Figs. 9a and 10a— i.e., respectively without the coarse grid correction and with the coarse grid correction—that the smooth eigenmodes are effectively resolved, such that a method is obtained that is robust to both cut elements and the grid size. It is noteworthy that the smallest eigenvalues with the multigrid method correspond to nodal modes, see Fig. 10b. The limited effec-

tiveness of the multigrid procedure for these modes derives from the fact that these modes are relatively insensitive to the smoothing operations, see Fig. 9c, and that these modes cannot be adequately captured on a coarser grid. With multigrid as a standalone solver, these modes would yield a convergence rate between 0.5 and 0.6. This rate can be improved by applying the multigrid cycle as a preconditioner in a Krylov subspace solver.

Figure 11 presents the spectra of the same problem with finer grids, preconditioned by a double fixed point iteration with additive Schwarz and preconditioned by the full V-cycle with additive Schwarz smoothing. Comparing these, and also the spectra with a grid of 16×16 elements in Figs. 9a and 10a, conveys that systems without the coarse grid correction closely follow the grid-size dependence of the conditioning formulated in (7), and that with the full V-cycle a conditioning is obtained that is independent of the grid size, with for this problem eigenvalues $0.4 < \lambda_{min} < 0.5$ and $\lambda_{max} = 1$.

While smoothing with additive Schwarz results in a conditioning that is independent of the grid size, the method is affected by the small relaxation parameter. This will be even more severe in three dimensions and for B-spline bases, which are supported on more elements and therefore require smaller and degree-dependent relaxation parameters, i.e., for B-splines $N^{K_i} = (p+1)^d$ with p the spline degree and d the number of dimensions. Therefore, the next section considers multiplicative Schwarz as a smoother, which is unconditionally stable [52] and thereby circumvents the small relaxation parameter required for stability. It should be mentioned that, alternatively, other smoothing schemes such as restricted additive Schwarz [94,95] can also be considered. The stability of such schemes with immersed finite elements does require verification, however.

Multiplicative Schwarz

Multiplicative Schwarz can be considered as the block equivalent of Gauss–Seidel. While with additive Schwarz all the locally inverted block matrices are applied to the same residual—similar to the diagonal elements in Jacobi—with multiplicative Schwarz the residual is updated after each block—similar to the update of the residual after each diagonal element in Gauss–Seidel. Multiplicative Schwarz can be formulated by initializing the zero vector $\tilde{\mathbf{y}}^0 = \mathbf{0}$, defining the initial residual $\tilde{\mathbf{r}}^0 = \mathbf{r}$, and looping over the blocks $j \leq N$:

$$
\begin{aligned}
\delta\tilde{\mathbf{y}}^j &= \mathbf{P}_j \mathbf{A}_j^{-1} \mathbf{P}_j^{\mathrm{T}} \tilde{\mathbf{r}}^{j-1}, \\
\tilde{\mathbf{y}}^j &= \tilde{\mathbf{y}}^{j-1} + \delta\tilde{\mathbf{y}}^j, \\
\tilde{\mathbf{r}}^j &= \tilde{\mathbf{r}}^{j-1} - \mathbf{A}\delta\tilde{\mathbf{y}}^j.
\end{aligned}
\tag{20}
$$

The linear operator of multiplicative Schwarz is then defined as $\mathbf{M}^{-1}\mathbf{r} = \tilde{\mathbf{y}}^N$. Similar to Gauss–Seidel, multiplicative Schwarz does not require stabilization [52], which is the most important motivation to examine multiplicative Schwarz as a smoother in multigrid methods for immersed finite elements.

The linear operator induced by a fixed point iteration with multiplicative Schwarz is not symmetric. Therefore, it is important that in the post-smoothing the direction in the loop over the index blocks is reversed, to restore symmetry of the linear operator induced by the V-cycle. As can be observed in (20), the application of multiplicative Schwarz requires updating the residual at every step, which is computationally expensive and impedes parallelization. Therefore, similar to standard Gauss–Seidel, the index blocks are ordered by a graph coloring algorithm [92], which has a negligible effect on the spectrum. To this end, the blocks are divided in C colors indicated by $c \leq C$. The basis functions in a block of a certain color do not intersect functions in a different block with the same color. Therefore, blocks of the same color

are not affected by each other's update of the residual. As a result, the residual only needs to be updated after executing all blocks of a certain color. This reduces the computational cost and enables parallelization of the routine. The multiplicative Schwarz procedure with graph coloring can be formulated by initializing the zero vector $\tilde{\mathbf{z}}^0 = \mathbf{0}$, defining the initial residual again as $\tilde{\mathbf{r}}^0 = \mathbf{r}$, and looping over the colors $c \leq C$:

$$
\begin{aligned}
\delta\tilde{\mathbf{z}}^c &= \sum_{j \in \mathcal{J}_c} \mathbf{P}_j \mathbf{A}_j^{-1} \mathbf{P}_j^{\mathrm{T}} \tilde{\mathbf{r}}^{c-1}, \\
\tilde{\mathbf{y}}^c &= \tilde{\mathbf{y}}^{c-1} + \delta\tilde{\mathbf{y}}^c, \\
\tilde{\mathbf{r}}^c &= \tilde{\mathbf{r}}^{c-1} - \mathbf{A}\delta\tilde{\mathbf{y}}^c,
\end{aligned}
\tag{21}
$$

with $\mathcal{J}_c$ denoting the set of blocks with color c. The linear operator of multiplicative Schwarz with graph coloring is defined as $\mathbf{M}^{-1}\mathbf{r} = \tilde{\mathbf{z}}^C$. The required minimal number of colors can be deduced by considering the overlap between the supports of basis functions. Because the supports of identically-colored basis functions are not allowed to intersect, for scalar problems and uniform grids the required number of blocks equals the number of elements on which basis functions are supported. For Lagrange bases, uniform B-spline bases, and truncated hierarchical B-spline bases, the employed number of colors therefore amounts to 2^d, $(p+1)^d$, and $M(p+1)^d$, respectively. For hierarchical bases the number of colors required for a uniform grid is multiplied by the number of hierarchical levels and for vector-valued problems the number of colors is multiplied by the number of components of the vector.

Figure 12 presents typical spectra of immersed systems, preconditioned by a symmetric double fixed point iteration with multiplicative Schwarz and preconditioned by a two-level V-cycle with multiplicative Schwarz smoothing. These results again pertain to the Laplace operator on the domain in Fig. 1 with quadratic Lagrange basis functions. The single-grid results without the coarse grid correction show a largest eigenmode of 1 and a mesh-dependent smallest eigenmode of order $\mathcal{O}(h^2)$, such that the condition number follows the relation for mesh-fitting systems in (7). Similar to the results with additive Schwarz in Fig. 11, the results with the two-level V-cycle show a spectrum that is robust to both cut elements and the mesh size of the background grid. The conditioning with multiplicative Schwarz is much better and even nearly optimal, however, because it is not compromised by the small relaxation parameter that was required for the stability of additive Schwarz. Therefore, we conclude that multiplicative Schwarz is the most suitable smoothing procedure for multigrid methods for immersed systems. In view of its superior smoothing properties over additive Schwarz, in the numerical examples in Sect. 4 we restrict our considerations to multiplicative Schwarz.
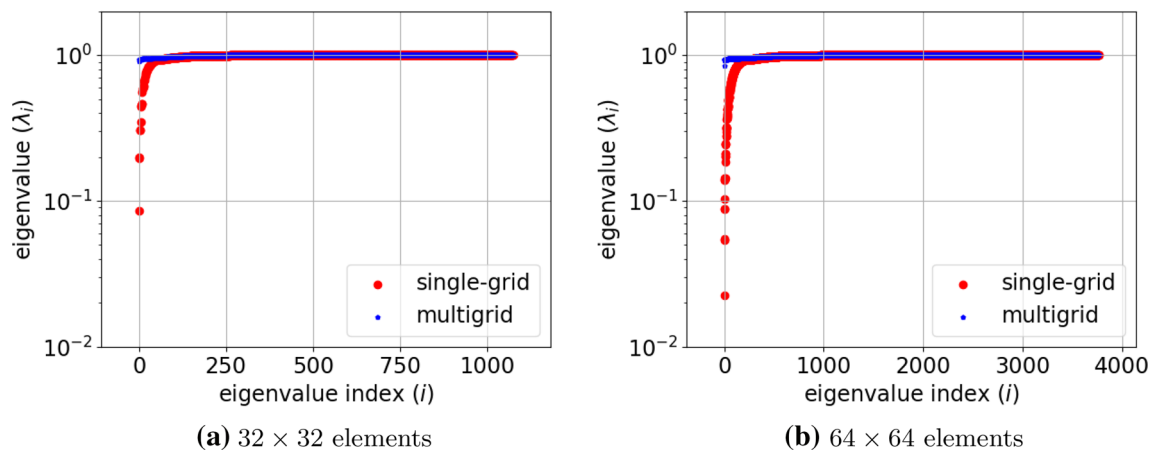
(a) 32×32 elements   **(b)** 64×64 elements

Fig. 12 Spectra of immersed systems that are preconditioned by a double fixed point iteration with multiplicative Schwarz (single-grid) and by a two-level V-cycle with multiplicative Schwarz as smoother (multigrid)

Remark 3.2 The resulting preconditioning techniques for Lagrange bases and B-splines have much in common, but also small differences are to be noted. As mentioned previously, Lagrange bases require a considerably smaller number of colors in the implementation. Also, the solver for the Lagrange bases requires fewer iterations, as can be observed in the results in Sect. 4. While both bases require approximately the same number of Schwarz blocks on the same grid, an advantage of the B-splines is that untrimmed basis functions result in a diagonal treatment, while untrimmed Lagrange basis functions still result in blocks with size $(2p - 1)^d$. Also the considerably smaller number of degrees of freedom with B-splines is an aspect to consider. The methods for both bases, however, can be further optimized in regard of the computational efficiency. Therefore, we would like to emphasize that this contribution is intended to demonstrate the feasibility of multigrid methods for immersed finite elements and immersed isogeometric analysis, and not as a qualitative comparison between the different bases. Furthermore, while the computation time is linear with the number of degrees of freedom for both bases, the absolute CPU times reported in Fig. 15 are highly dependent on the implementation.

Remark 3.3 Similar to the Schwarz-type methods for immersed finite elements presented in [50] and [51], it is possible that block matrices contain eigenvalues of the order of the machine precision. Directly inverting such block matrices is unstable, as due to round-off errors these inverses can be inaccurate and can even contain negative eigenvalues for positive definite systems. As described in detail in [50, Remark 3.3], in case a block matrix contains an eigenvalue that is a factor 10^{16} smaller than the largest diagonal entry of the matrix, the basis function that is dominant in the corresponding eigenvector is removed from the block. Because this only pertains to basis functions with extremely small contribu-

tions, this does not affect the convergence of the iterative solver or the accuracy of the solution.

Remark 3.4 In principle, the ill-conditioning effects of small cut elements only need to be resolved on the finest grid, as the coarser levels are only required to resolve smooth components of the error. Diagonal smoothers therefore suffice on the coarser levels, and Schwarz-type smoothing is only required on the finest level. Furthermore, this opens the possibility to apply black-box coarsening algorithms as used in algebraic multigrid techniques (AMG). Applying Schwarz-type smoothing on all levels, however, retains the natural recursive character of the multigrid algorithm. For conciseness, we have therefore opted not to include results with diagonal smoothing on the coarser levels in the numerical results in Sect. 4.

4 Numerical examples

In this section we assess the developed geometric multigrid preconditioning technique for immersed finite element methods on a range of numerical examples. First, Sect. 4.1 considers three-dimensional elasticity problems on uniform grids with both B-splines and Lagrange basis functions. The goal of these simulations is to demonstrate the performance of the preconditioner on increasingly complex geometries. An aspect that is not covered in this section is the treatment of local mesh refinements. Therefore the preconditioner is applied to a level set based topology optimization problem with truncated hierarchical B-splines in Sect. 4.2. Our implementation is based on the open source software package Nutils ([96], www.nutils.org). The code to construct the multigrid preconditioner and data to reproduce the most prominent results can be downloaded from https://gitlab.com/fritsdeprenter/multigrid-immersed-fem.

In all examples, the systems are solved by a conjugate gradient solver that is preconditioned with the multigrid V-cycle in which the pre-smoothing and post-smoothing operations consist of a single multiplicative Schwarz sweep, as described in Sect. 3.4. Furthermore, all simulations are performed with quadratic discretizations, except for the investigation into the effects of the discretization order for the first example in Sect. 4.1. This is sufficient to demonstrate the difference between Lagrange basis functions and B-splines, and quadratic function spaces already result in severely ill-conditioned systems in case a dedicated treatment is not applied. The assembly of quadratic systems is considerably less expensive than systems of third degree or higher, however, enabling finer grids to be assessed.

4.1 Linear elasticity problems

We first consider the deformation of a tooth-shaped domain subject to a distributed boundary traction. Next, an apple-shaped geometry containing two geometrically singular points is presented, which is deformed by a gravitational load. The third example considers a complex geometry of a μCT-scanned trabecular bone specimen, that is compressed by a prescribed displacement at the boundary. In this third example an effect regarding the number of levels in the preconditioner is observed. This effect is further investigated in the specifically designed test case in the last example, which is posed on a geometry in the shape of a triple helix.

Tooth-shaped geometry subject to a distributed traction

This first example considers a dimensionless problem posed on an immersed geometry with the shape of a tooth, see Fig. 13. The embedding domain is the cube $(-2, 2)^3$, and the tooth-shaped geometry is obtained by trimming with the level set function:

$$\psi_1(x, y, z) = 16 \left(1 - \sum_{i=0}^{6} e^{-\rho_i(x,y,z)}\right) - x^4 - y^4 - z^4,$$

$$(22)$$

with:

$$
\begin{aligned}
\rho_0(x, y, z) &= x^2 + y^2 + (z - 2)^2, \\
\rho_1(x, y, z) &= (x - 2)^2 + (y - 2)^2 + \left(\tfrac{z+2}{2}\right)^2, \\
\rho_2(x, y, z) &= (x - 2)^2 + (y + 2)^2 + \left(\tfrac{z+2}{2}\right)^2, \\
\rho_3(x, y, z) &= (x + 2)^2 + (y + 2)^2 + \left(\tfrac{z+2}{2}\right)^2, \\
\rho_4(x, y, z) &= (x + 2)^2 + (y - 2)^2 + \left(\tfrac{z+2}{2}\right)^2, \\
\rho_5(x, y, z) &= x^2 + \left(\tfrac{y}{2}\right)^2 + \left(\tfrac{z+2}{2}\right)^2, \\
\rho_6(x, y, z) &= \left(\tfrac{x}{2}\right)^2 + y^2 + \left(\tfrac{z+2}{2}\right)^2.
\end{aligned}
$$

In this level set $\rho_0(x, y, z)$ creates the dent in the surface of the tooth, $\rho_1(x, y, z)$, $\rho_2(x, y, z)$, $\rho_3(x, y, z)$, and $\rho_4(x, y, z)$ create the dents in the sides of the tooth and the roots are obtained by $\rho_5(x, y, z)$ and $\rho_6(x, y, z)$. The tips of the roots below $z = -1$ are trimmed by a second trimming operation with the level set function:

$$\psi_2(x, y, z) = z + 1. \tag{23}$$

This creates four surfaces on which homogeneous Dirichlet conditions are applied. On the rest of the boundary a normal traction is applied with the magnitude:

$$\mathbf{g}^N(x, y, z) = -\mathbf{n}e^{-\frac{1}{4}\left((x-2)^2+(y-2)^2+(z-2)^2\right)}, \tag{24}$$

which concentrates around the corner of the tooth. The Lamé parameters are set to $\lambda = \mu = 10^3$, and the Dirichlet conditions are enforced by the penalty method with penalty parameters $\beta_h^\lambda = \beta_h^\mu = \frac{2}{h}$. The resulting displacements and stresses are shown in Fig. 13.

To demonstrate the robustness of the preconditioning technique to the number of elements, we discretize the embedding domain with grids of 20^3, 40^3 and 80^3 elements. This results in, respectively, 129×10^3, 874×10^3 and 6.43×10^6 degrees of freedom (DOFs) with quadratic Lagrange basis functions and 21.6×10^3, 129×10^3 and 878×10^3 DOFs with quadratic B-splines. The integration depth as defined in Sect. 2 is set to 2 for the grid with 20^3 elements, 1 for the grid with 40^3 elements, and 0 for the grid with 80^3 elements. This implies that the grid with 20^3 elements is first partitioned by 2 consecutive bisectioning operations before it is triangulated, the grid with 40^3 elements is first partitioned by 1 bisectioning operation before it is triangulated, and cut elements in the grid with 80^3 elements are directly triangulated. Note that this results in identical integrated geometries for all grid sizes. The multigrid preconditioner is applied with 2 and 3 levels as defined in Sect. 3 for the grid with 20^3 elements, 2, 3 and 4 levels for the grid with 40^3 elements, and 2, 3, 4 and 5 levels for the grid with 80^3 elements. These numbers of levels are chosen such that for all grid sizes, the largest number of levels results in a direct solution in the preconditioner for a system derived from a discretization with 5^3 elements.

The convergence of the multigrid-preconditioned conjugate gradient solver is plotted in Fig. 14. The results demonstrate a convergence behavior that is virtually independent of the number of elements. Also, the number of iterations is nearly independent of the number of levels in the preconditioner. This can be attributed to the simple compact shape of the geometry, which is resolved well even on the coarsest grid of 5^3 elements, such that with all numbers of levels the coarse grid corrections provide an effective approximation of the smooth eigenmodes. Finally, it can be observed that

Fig. 13 Displacements and stresses of the tooth-shaped geometry. The displayed results are computed on a discretization with 80^3 elements and quadratic B-spline basis functions

(a) Displacement magnitude

(b) Frobenius norm of the stress tensor
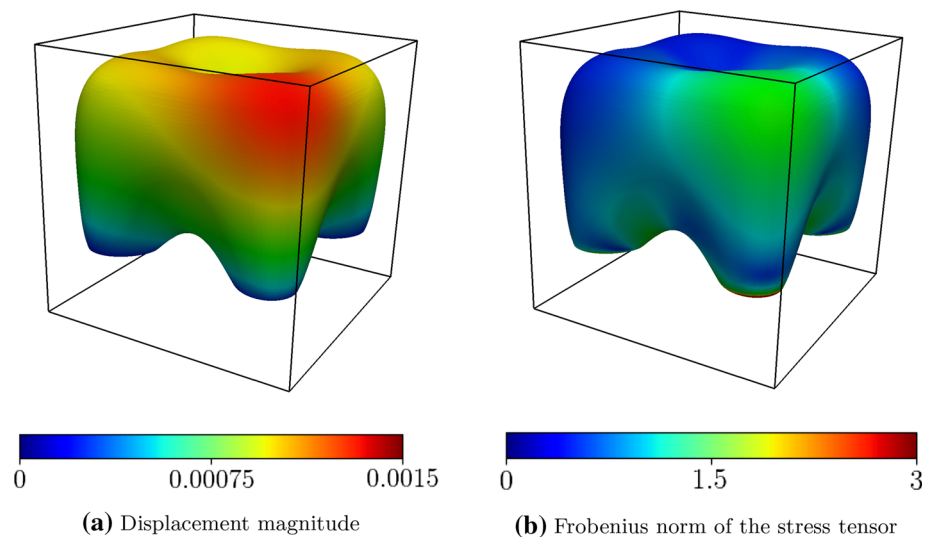
the systems with B-splines require more iterations than the systems with Lagrange basis functions.

Figure 15 plots the measured CPU times to set up the multigrid preconditioners and solve the linear systems versus the grid size. These results pertain to the systems in Fig. 14 with the largest number of multigrid levels and a direct solve on the grid of 5^3 elements. The CPU times correspond to computations on a single core, but it should be noted that the method is suitable for parallelization as demonstrated in [51]. It is clearly visible that the observed CPU times scale linearly with the number of DOFs. An interesting observation is that the setup time for B-spline bases is significantly lower than that for Lagrange systems. This is explained by the computationally inexpensive diagonal treatment of untrimmed B-splines, which contrasts the block treatment of untrimmed Lagrange basis functions. Asymptotically, the majority of the computational workload for B-spline systems consists of the diagonal scaling of these untrimmed basis functions. On coarse grids, the relative number of cut elements is much larger however, such that the CPU time to invert the blocks associated to cut B-splines dominates. As the number of cut elements does not scale linearly with the number of DOFs, the scaling rate deviates in the preasymptotic regime. Note that with Lagrange basis functions the construction of the preconditioner for cut basis functions is very similar to that of untrimmed basis functions, such that this deviation is not observed. It is interesting to observe that the total CPU time to obtain the solution with either Lagrange basis functions or B-splines is comparable, with the Lagrange systems being solved moderately faster. We attribute this to the number of iterations and the number of colors in the coloring algorithm, *cf.* the discussion on the computational cost of Lagrange and B-spline bases in Remark 3.2.

The effect of the discretization order is investigated in Fig. 16, which plots the convergence of systems with both

Lagrange bases and B-splines for different orders and different numbers of levels in the multigrid preconditioner. The systems correspond to grids with 40^3 elements and a maximum integration depth of 1. The Lagrange bases contain, respectively, 116×10^3, 874×10^3 and 2.89×10^6 DOFs and the B-spline bases 116×10^3, 129×10^3 and 143×10^3 DOFs. Note that the results with quadratic bases in Fig. 16c and d are the same as those in Fig. 14c and d, but with the horizontal axis rescaled to facilitate a comparison with the linear and cubic systems. Additionally, both the bases and the treatment of the first order systems in Fig. 16a and b are identical. It is observed that, similar to the quadratic results in Fig. 14, the number of levels in the multigrid preconditioner does not significantly affect the convergence. Furthermore, the convergence of the systems with Lagrange basis functions is virtually unaffected by the discretization order, in contrast to the convergence rate of the B-spline systems, which reduces with an increase in the polynomial degree. This increase in the number of iterations with the B-spline degree has been previously reported in the literature, see e.g. [61], and can be associated to oscillatory eigenmodes with small eigenvalues that occur in high-order isogeometric discretizations, e.g., [64]. This effect can be mitigated by *p*-multigrid methods, see e.g., [97,98], or by dedicated smoothers for isogeometric analysis such as the multi-iterative method [63,64], smoothers based on spline-space splittings [65–68], or multiplicative Schwarz smoothers with block sizes that increase with the discretization order [72]. In this regard it should be noted that, although for both bases the number of blocks is essentially independent of the discretization order, the size of the Schwarz blocks in the Lagrange systems increases with the discretization order, while the treatment of untrimmed B-splines is still diagonal. We have experienced that the CPU times for Lagrange and B-spline systems of the same degree are similar, which is in agreement with the observations on
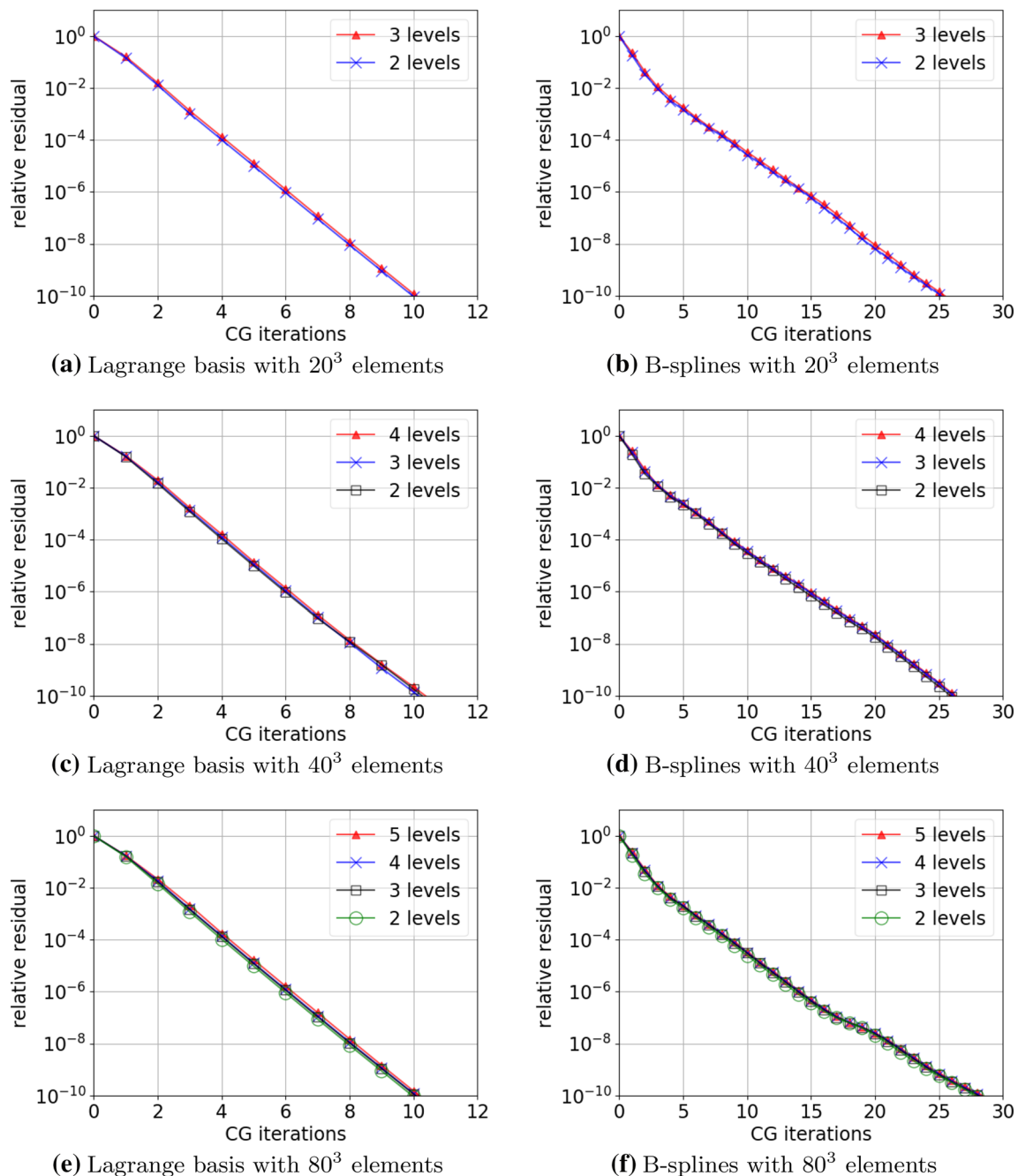
Fig. 14 Convergence plots of the problems on the tooth-shaped geometry with quadratic discretizations

quadratic systems in Fig. 15. Additionally, similar to the results with quadratic bases on different grid sizes in Fig. 14, for both types of basis functions the convergence rate with linear and cubic bases has been found to be virtually independent of the grid size and the number of DOFs.

Apple-shaped geometry subject to a gravitational load

This second example is designed to establish the suitability of the preconditioner for non-convex geometries with sharp reentrant corners where stress singularities are to be expected, such that the effectivity of multigrid methods is not generally evident [99]. A dimensionless problem is posed on the geometry with the shape of an apple in Fig. 17. The embedding domain is again the cube $(-2, 2)^3$, and the shape of the apple is derived through a trimming operation with the level set function:

$$\psi_1(x, y, z) = 1 - \left(\frac{z}{1.7}\right)^2 - \left(\frac{r(x, y)}{1 + z/17} - 0.7\right)^2, \quad (25)$$

Fig. 15 CPU times for the setup of the multigrid preconditioner and for the iterative solving of the systems with quadratic discretizations and different grid sizes. The dotted lines indicate the linear scaling with the number of DOFs

(a) Lagrange basis   **(b)** B-splines
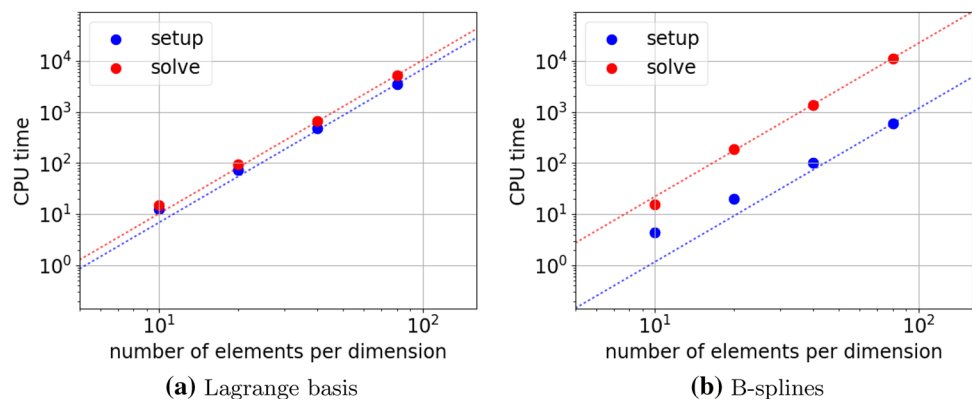
with $r(x, y)^2 = x^2 + y^2$. A second level set function models a bite being taken out of the apple:

$$\psi_2(x, y, z) = (x - 2)^2 + y^2 + z^2 - 1. \qquad (26)$$

Homogeneous Dirichlet conditions are imposed on the surface of the bite, and a volumetric load $\mathbf{f} = (0, 0, -1)$ is applied to model gravity acting on the apple. The Lamé parameters are again set to $\lambda = \mu = 10^3$ and the Dirichlet conditions are again enforced by the penalty method with parameters $\beta_h^\lambda = \beta_h^\mu = \frac{2}{h}$. Figure 17 displays the resulting displacements and stresses.

The problem is discretized with quadratic Lagrange basis functions and quadratic B-splines on a background grid with 80^3 elements. This yields 4.87×10^6 DOFs supported in the physical domain with the Lagrange basis and 667×10^3 DOFs with the B-splines. The integration depth on the cut elements is set to 0, implying that the the cut elements are directly triangulated and not first partitioned with a bisectioning operation. The convergence of the multigrid-preconditioned conjugate gradient solver with 2, 3, 4 and 5 levels for both bases is shown in Fig. 18. The obtained numbers of iterations do not show an effect of the cusps in the geometry, and are similar to those for the tooth-shaped geometry. Also, the convergence is again virtually independent of the number of levels in the multigrid cycle.

Trabecular bone specimen loaded in compression

This third three-dimensional test case considers the challenging geometry of a μCT-scanned trabecular bone specimen. This geometry was first presented in [23], and is displayed in Fig. 19, together with the embedding domain of dimension $(0 \text{ mm}, 1.28 \text{ mm})^3$. A linear elastic material model is employed with Young's modulus $E = 10$ GPa and Poisson's ratio $\nu = 0.3$. The specimen is compressed with an average uniaxial compressive strain of 1%, by imposing a homogeneous Dirichlet condition at the top boundary, and at the bottom boundary prescribing a normal displacement

of 0.0128 mm while constraining the tangential displacement. These boundary conditions are weakly enforced by the penalty method with penalty parameters $\beta_h^\lambda = \beta_h^\mu = \frac{2}{h}$.

We consider different grids on the embedding domain with 32^3, 64^3 and 128^3 elements. The linear systems derived from these grids contain, respectively, 182×10^3, 1.03×10^6 and 6.65×10^6 DOFs with the quadratic Lagrange bases and 39.9×10^3, 189×10^3 and 1.05×10^6 DOFs with the quadratic B-splines. The integration depth is set to 2 for the grid with 32^3 elements, 1 for the grid with 64^3 elements, and 0 for the grid with 128^3 elements. The multigrid cycle by which the systems are preconditioned applies 2 and 3 levels for the discretizations with 32^3 elements, 2, 3 and 4 levels for the discretizations with 64^3 elements, and 2, 3, 4 and 5 levels for the discretizations with 128^3 elements. With these numbers of levels, the preconditioner with the largest number of levels applies a direct solver to a system derived from 8^3 elements for all three grid sizes, similar to the first example. The convergence of the systems with the different numbers of levels in the preconditioner is shown in Fig. 20. It can be observed that the convergence of the systems with 32^3 elements and 2 levels, 64^3 elements with 2 and 3 levels, and 128^3 elements with 2, 3 and 4 levels is very similar to that in the previous examples. For this test case, slightly more iterations are required. This is conjecturally connected with the complicated multiscale geometry, by which smaller geometric features are less adequately represented on coarse meshes. The convergence with the largest number of levels in the preconditioner, i.e., the preconditioners in which a direct solver is applied to a system with 8^3 elements, is significantly slower. This is caused by underresolution and corresponding nonphysical behavior in the systems with 8^3 elements, which was also observed in [23]. Both with the Lagrange and the B-spline bases, these coarse systems contain basis functions with a disjoint support in the physical domain, i.e., basis functions that cover the gap between disconnected parts of the geometry as e.g., in the circle in Fig. 19a. Physically, disconnected parts of the geometry should be able to move freely with respect to each other. When a basis function is
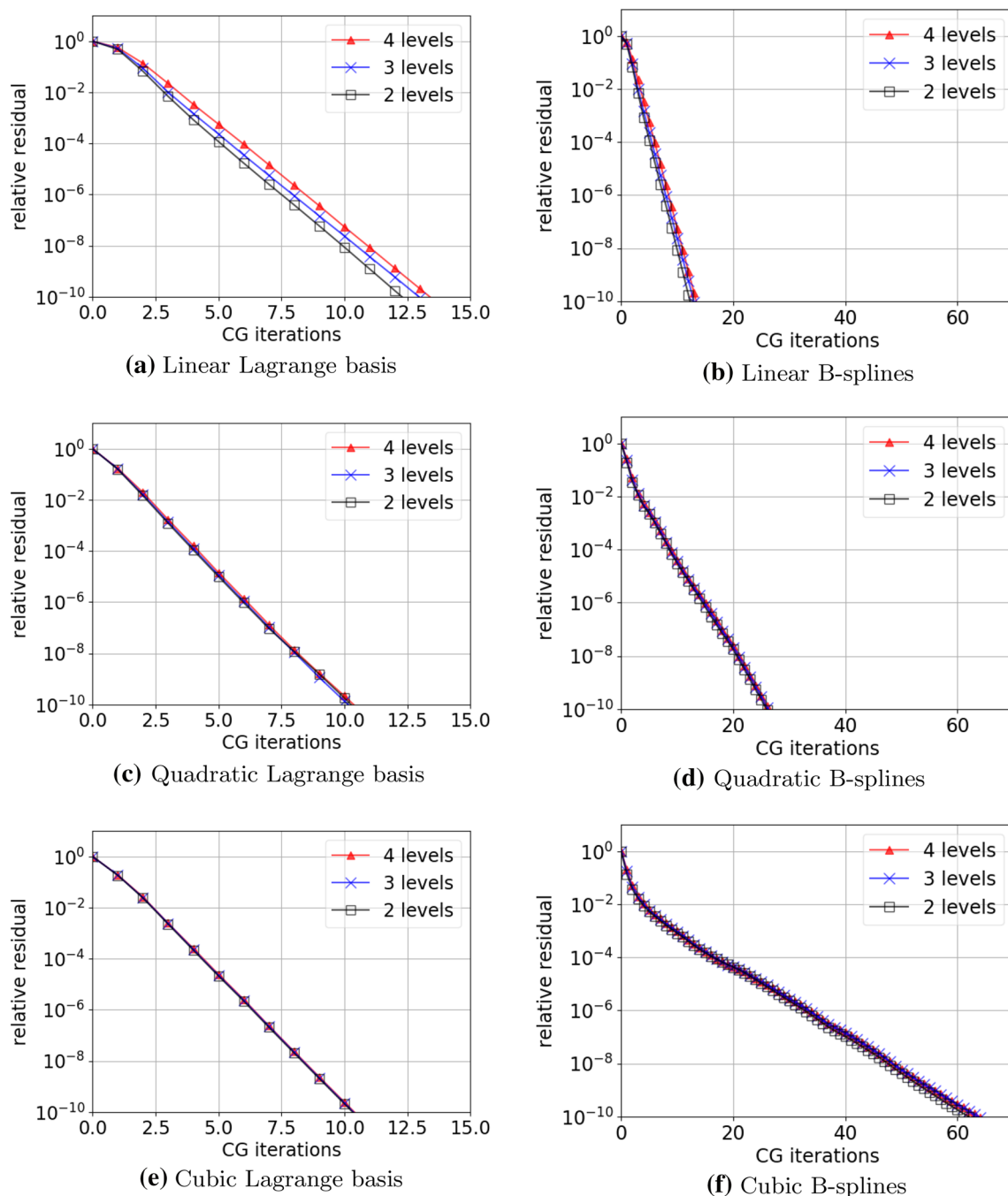
(a) Linear Lagrange basis

(b) Linear B-splines

(c) Quadratic Lagrange basis

(d) Quadratic B-splines

(e) Cubic Lagrange basis

(f) Cubic B-splines

Fig. 16 Convergence plots of the problems on the tooth-shaped geometry with 40^3 elements and different discretization orders

supported on both parts, however, these disconnected parts are artificially coupled in the numerical model. This results in very different stiffness properties between the systems with a mesh of 8^3 elements and systems with meshes of 16^3 elements and finer. Therefore, the coarse grid corrections with 8^3 elements do not approximate the smooth eigenmodes of the finer meshes adequately, which retards the convergence of the multigrid-preconditioned iterative solver. A similar effect can be observed in nearly incompressible elasticity,

where the convergence deteriorates when the coarse grids experience volumetric locking, as in e.g., [100].

To illustrate the artificial coupling, the eigenfunction with the smallest eigenvalue in the three-level preconditioned system with B-splines on a grid of 32^3 elements is shown in Fig. 21. This function was obtained by 100 iterations in a power algorithm, and it is clearly observable that this smallest preconditioned eigenmode contains peaks in the displacement and stress fields at points where basis functions of the

Fig. 17 Displacements and stresses in the geometry with the shape of an apple. The results are obtained with a B-spline basis on a grid of 80^3 elements
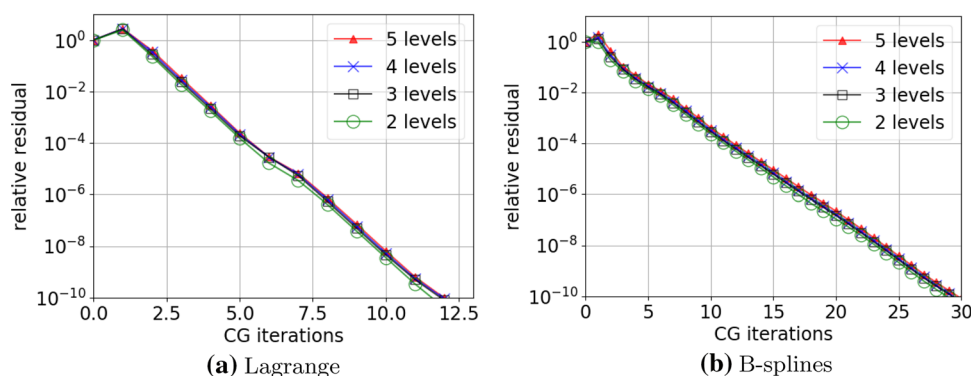

0   0.015   0.030
(a) Displacement magnitude

0   3   6
(b) Frobenius norm of the stress tensor
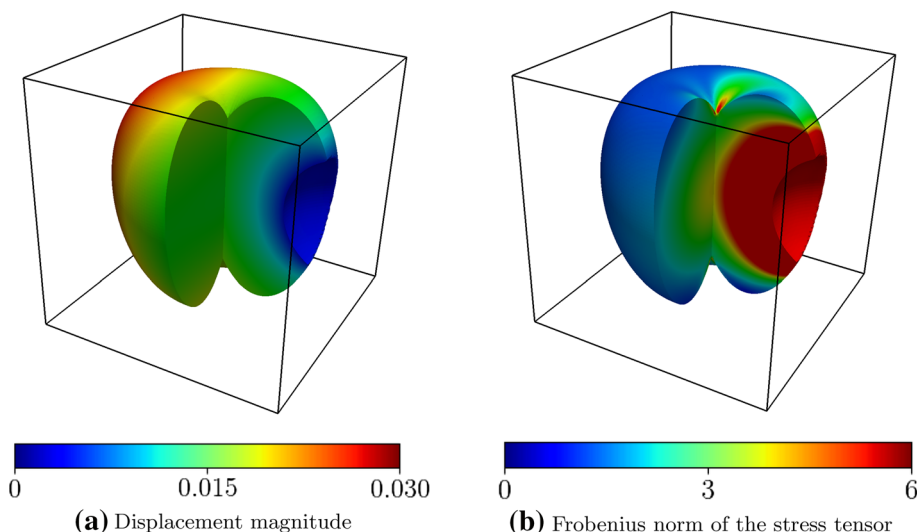
Fig. 18 Convergence of the problem on the apple-shaped geometry with a grid of 80^3 elements and quadratic bases

(a) Lagrange

(b) B-splines

coarse grid are supported on disconnected parts of the geometry. Such an artificial-coupling mode does not occur if the system is preconditioned with two levels, i.e., with a direct solver applied to the system with 16^3 elements. While the artificial-coupling effect only moderately increases the number of iterations in this test case, and the iterative solver still converges in an acceptable number of iterations, it can not be ruled out that it can potentially hinder the convergence in other test cases more severely. Therefore, this effect is investigated in more detail in the next test case, which presents a geometry that is specifically designed to study this effect.

Triple helix loaded in compression

This final three-dimensional example is specifically designed to investigate the effect observed in the trabecular bone specimen, by which disconnected parts of the domain are artificially coupled on coarse grids. We consider a dimensionless problem on the geometry in Fig. 22, which consists of a triple helix that is connected by a half ring at the top and bottom boundary. The embedding domain again consists of the cube $(-2, 2)^3$. The helixes are obtained by rotations of the level set function:

$$\psi_1(x, y, z) = r_{\text{inner}}^2 - (r(x, y) - R_{\text{outer}})^2$$
$$- \left(z - 2 \frac{\arctan2(y, x)}{\pi} \right)^2, \tag{27}$$

with $r_{\text{inner}} = 0.25$, $R_{\text{outer}} = 1.5$, and $r(x, y)^2 = x^2 + y^2$. For the half ring at the top and bottom boundary the level set function:

$$\psi_2^\pm(x, y, z) = r_{\text{inner}}^2 - (r - R_{\text{outer}})^2 - (z \pm 2)^2, \tag{28}$$

is applied. The Lamé parameters are again set to $\lambda = \mu = 10^3$. At the top boundary homogeneous Dirichlet conditions are imposed, and at the bottom boundary a normal displacement of 0.04 prescribed, resulting in an average compression of 1%. It should be noted that, as opposed to the bottom boundary condition on the trabecular bone specimen, the tangential displacement at the bottom boundary is not constrained. The boundary conditions are again weakly imposed by the penalty method with parameters $\beta_h^\lambda = \beta_h^\mu = \frac{2}{h}$. The solution with B-splines on a grid with 128^3 elements is shown in Fig. 22.

The embedding domain is discretized with 128^3 elements, yielding 6.96×10^6 and 1.11×10^6 DOFs in the quadratic

Fig. 19 Solution of the elasticity problem on the trabecular bone geometry

(a) Displacement magnitude [mm]

(b) Frobenius norm of the stress tensor [MPa]
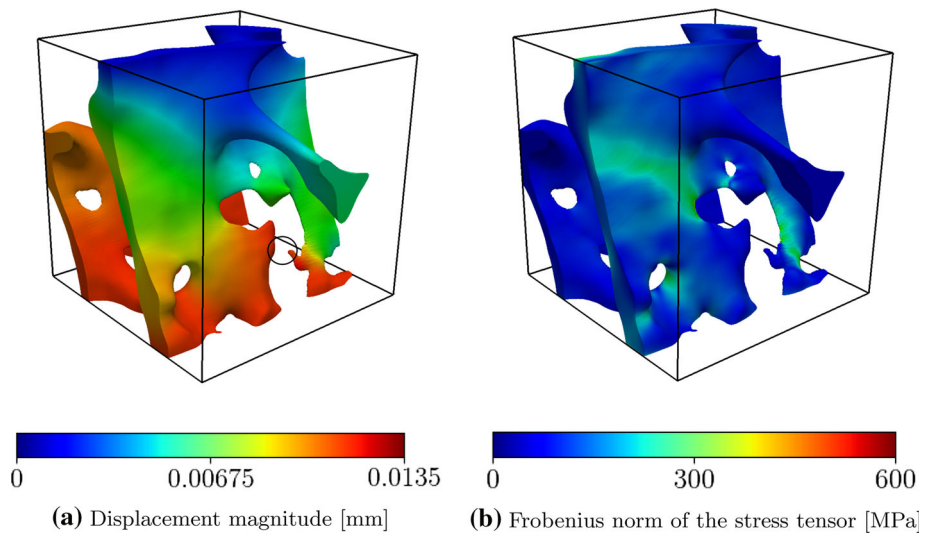
Fig. 20 Convergence plots of the test case on the trabecular bone specimen

(a) Lagrange basis with 32^3 elements

(b) B-splines with 32^3 elements

(c) Lagrange basis with 64^3 elements

(d) B-splines with 64^3 elements

(e) Lagrange basis with 128^3 elements

(f) B-splines with 128^3 elements
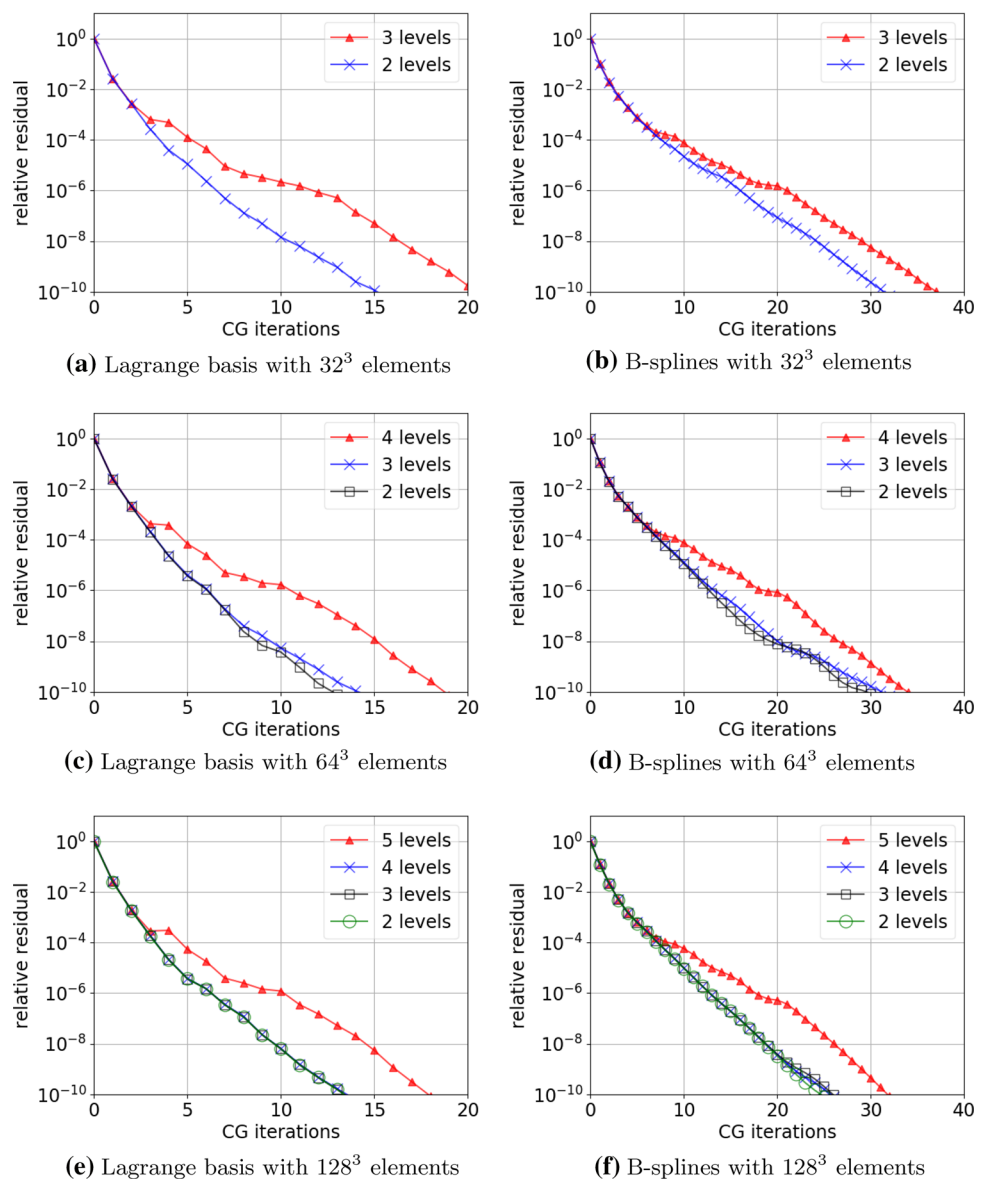
Fig. 21 Displacement and stress of the smallest eigenmode of the three-level preconditioned system with quadratic B-splines, formed on a grid of 32^3 elements. The eigenmode shows an artificial stress state caused by a nonphysical coupling between disconnected parts of the geometry and the resulting deformation

(a) Displacement magnitude [mm]   **(b)** Frobenius norm of the stress tensor [MPa]
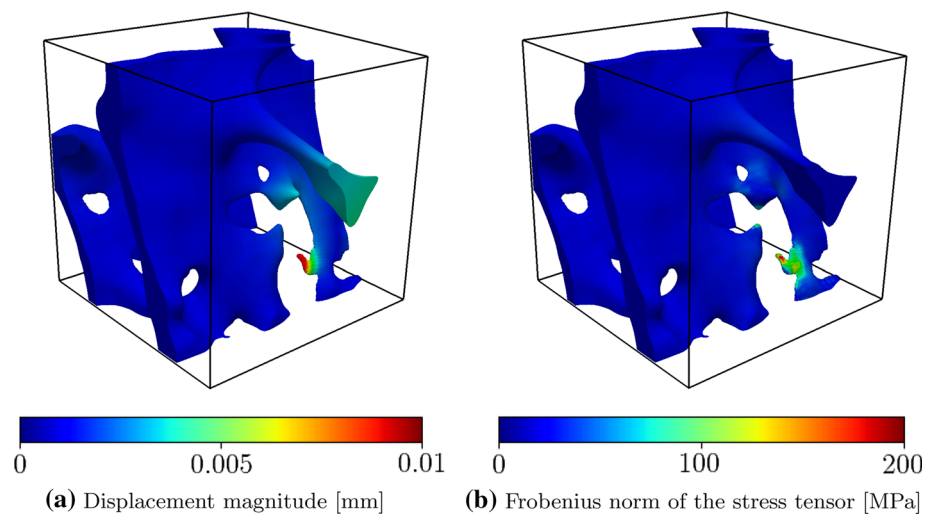
Fig. 22 Solution of the elasticity problem on the triple-helix-geometry with B-spline basis functions on a grid of 128^3 elements

(a) Displacement magnitude   **(b)** Frobenius norm of the stress tensor
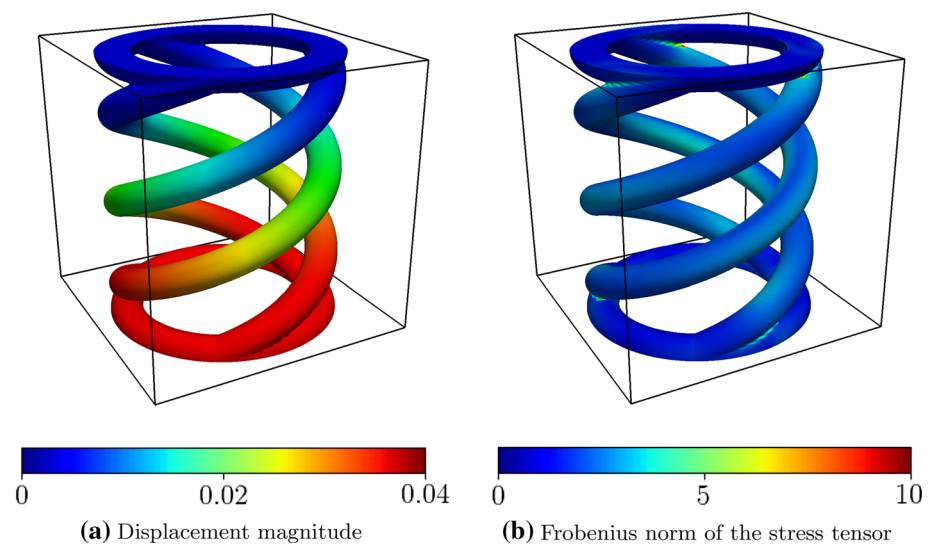
Fig. 23 Convergence of the triple helix geometry with 128^3 elements

(a) Lagrange   **(b)** B-splines
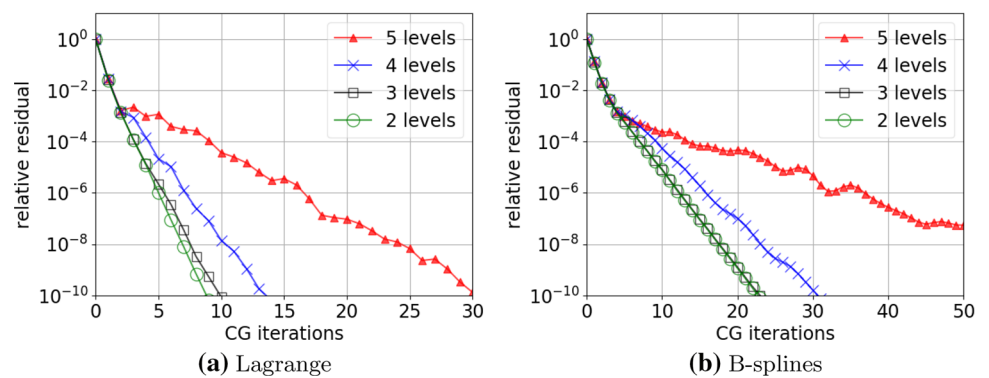
Lagrange and B-spline bases, respectively. The integration depth is set to 0, and the preconditioner applies 2, 3, 4 and 5 levels in the V-cycle. The convergence plots are shown in Fig. 23. It is clearly observable that the convergence is severely retarded with 5 levels in the preconditioner. This could be anticipated, because the separation of the helixes is approximately equal to the mesh size on the coarsest level of

8^3 elements. Hence, artificial coupling between the helixes will occur on the coarsest mesh, reducing the effectiveness of the coarse grid correction. To further illustrate this effect, Fig. 24 displays the smallest eigenmode in a system with B-splines on 32^2 elements that is preconditioned with 3 levels in the V-cycle, which behaves similarly as it also contains 8^3 elements at the coarsest level. This mode is the equivalent of the

Fig. 24 Smallest eigenmode of a system with quadratic B-splines on a grid of 32^3 elements that is preconditioned by a V-cycle with 3 levels. The stress clearly indicates the artificial coupling that was also observed in the test case on the trabecular bone geometry. Furthermore, it is visible that the mismatch between the stiffness properties of the coarsest and the finer grids yields a smooth eigenmode that is not captured by the coarse grid correction

(a) Displacement magnitude   **(b)** Frobenius norm of the stress tensor
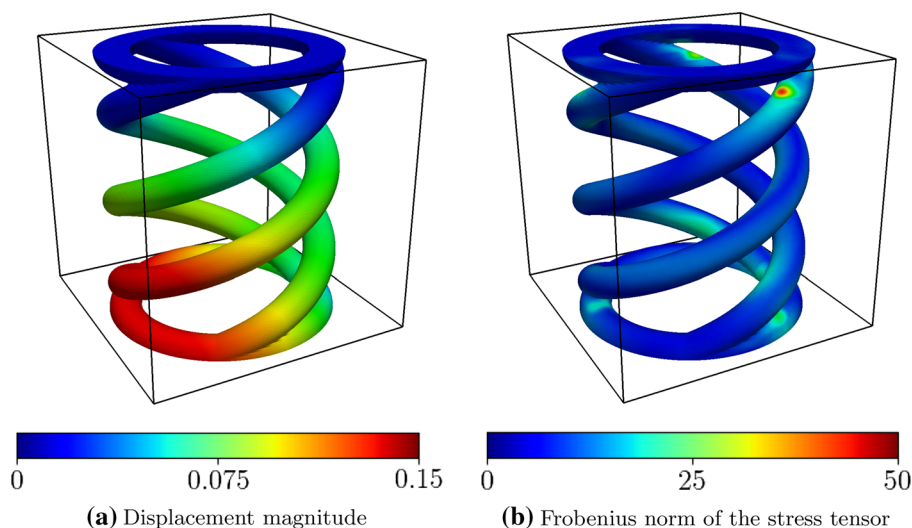
smallest mode with the trabecular bone geometry in Fig. 21, as both these smallest eigenmodes clearly demonstrate the aforementioned artificial coupling effect. The preconditioner with 4 levels—for which approximately 2 elements of the coarsest level fit between separate helixes—also converges significantly slower. For the quadratic B-splines that span 3 elements this can obviously be attributed to the artificial coupling effect. For the Lagrange basis functions that only span 2 elements, it should be noted that basis functions covering the gap between a helix and a half ring can still artificially increase the stiffness of the connection between these. In the systems preconditioned by a V-cycle with 2 and 3 levels, the coarsest level contains 64^3 and 32^3 elements, respectively, such that the nonphysical coupling effect is not observed and the convergence is similar to that of the previous examples. We expect that the observed deterioration of the convergence behavior can be mitigated by a dedicated coarsening algorithm, which prevents disjoint supports in coarse basis functions by applying local refinements or XFEM-type enrichments on the coarse grid. In this regard the work presented in [75] is noteworthy, since it considers a multigrid approach in which a similar form of artificial coupling through a crack on the coarse level is precluded. Furthermore, it should be noted that this effect will not occur with an algebraic multigrid technique (AMG), as a black-box coarsening algorithm will not pick up a connectivity between nonintersecting basis functions on different sides of the gap.

4.2 A level set based topology optimization problem with truncated hierarchical B-splines

In this example we apply the developed multigrid preconditioning technique to a dimensionless level set based topology optimization problem, which is inspired by the classical MBB beam [101]. This test case is of particular interest,

Fig. 25 Design space of 3×1 and boundary conditions of the topology optimization problem. The vertical displacement is constrained at the right bottom, and a vertical load is applied at the left top. The horizontal displacement is constrained at the left boundary, which imposes a symmetry condition, such that the problem setup resembles a simply supported beam

because it demonstrates the robustness to evolving geometries, and it establishes the suitability to locally refined grids with truncated hierarchical B-splines. Because this involves a large number of computations on an evolving geometry, this numerical experiment is performed in two dimensions to reduce the computational cost. The design space and boundary conditions of the optimization problem are shown in Fig. 25, and samples of the grids and geometries during the procedure are presented in Fig. 26. The objective of the design problem is to minimize the strain energy of the structure, subject to a volume constraint that restricts the volume to 30% of the design domain. The level set function is discretized by linear basis functions, which are smoothed by a linear filter [102]. To mitigate the dependence of the optimization results on the initial level set function, a hole seeding method is used that considers the co-evolution of a density field [103]. The parameters of the discretized level set and density fields are treated as optimization variables, and are updated in the optimization process by the globally convergent method of moving asymptotes (GCMMA) [104]. This test case is intended to establish that elasticity problems on these geometries and grids can be robustly solved in an

Fig. 26 Samples of grids, geometries and solutions during the optimization procedure

(a) Step 0

(b) Step 30

(c) Step 45

(d) Step 70

(e) Step 120

(f) Step 150

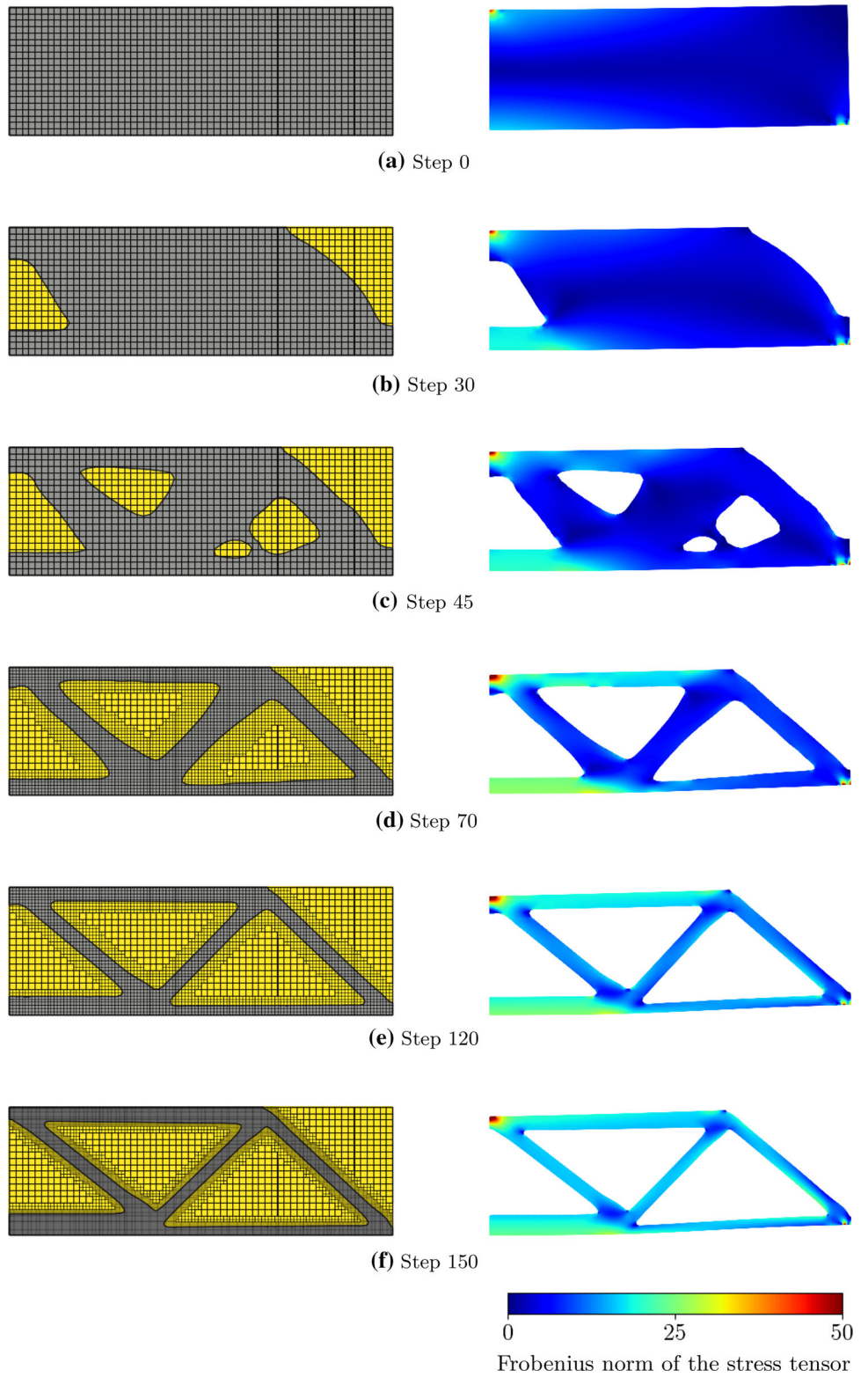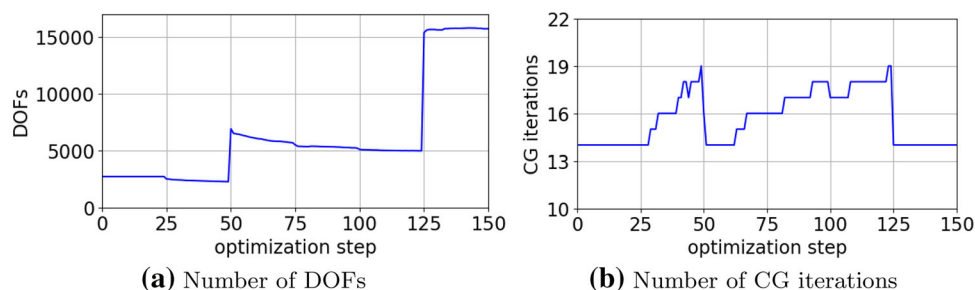0   25   50

Frobenius norm of the stress tensor

Fig. 27 Number of DOFs and required number of CG iterations to reduce the residual by 10^{-10} during the different steps of the topology optimization procedure

(a) Number of DOFs

(b) Number of CG iterations

iterative manner by means of the developed multigrid pre-conditioner. This opens the doors to level set based topology optimization problems beyond the reach of direct solvers.

On the locally refined grids supplied by the topology optimization procedure, quadratic truncated hierarchical B-splines are constructed, which are trimmed with the provided level set functions. A uniform grid of 60×20 elements is applied in steps 0–49 of the routine, as visible in Fig. 26a, b and c. In steps 50–124, grids with a single level of hierarchical refinements are used, see Fig. 26d and e. The refinement patterns on these grids are determined based on the designs at iterations 49 (i.e., the final design on the uniform grid), 74 and 99. In steps 125–150, grids with two levels of hierarchical refinements are employed. The two-level refinement in steps 125–149 is based on the final design with a single level of refinements in step 124, and is updated in step 150 based on the design in step 149. After this mesh update, the topology optimization algorithm reaches convergence and terminates. The final design is displayed in Fig. 26f.

For each level set and each mesh occurring in the topology-optimization procedure, we consider a linear elasticity problem with Lamé parameters $\lambda = \mu = 10^3$. The Dirichlet conditions are weakly imposed by the penalty method, using the penalty parameters $\beta_h^\lambda = \beta_h^\mu = 4\frac{2}{h}$, with $h = \frac{1}{20}$ denoting the unrefined element size and the factor 4 to cover for the local refinements. The vertical support is applied over the width of one unrefined element, and the load at the left top is applied over the same width and has a normal traction with a magnitude of 20. We apply the conjugate gradient solver, preconditioned by the multigrid cycle with the tailored multiplicative Schwarz smoother. Three levels are applied in the preconditioner, such that the direct solver is applied to a coarsest grid of 15×5 elements, with a similar pattern of 0, 1, or 2 levels of local refinements as supplied by the optimization procedure for that step.

Figure 27 plots the number of DOFs and the number of CG iterations during the procedure. The number of DOFs clearly shows a sharp increase after 50 and 125 iterations, when the number of local refinement levels is increased. The number of iterations in Fig. 27b demonstrates that the preconditioning technique is robust to cut elements, and is not sensitive to changes in the geometry and topology. It can be observed
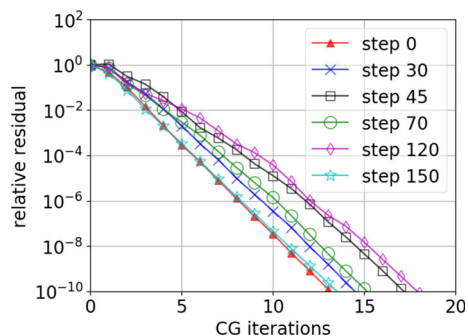

Fig. 28 Convergence plot of the samples of the level set based topology optimization problem with truncated hierarchical B-splines in Fig. 26

that the number of iterations moderately increases from steps 25 and 50 and between steps 50 and 125, as the complexity of the evolving physical domain increases. Furthermore, the number of iterations is reduced when more levels of local refinements are applied. These effects are similar to those observed in the three-dimensional test cases; since the extra levels of local refinements are also applied to the coarser grids, the coarse grid correction terms resolve the smooth eigenmodes of the finest grid more accurately, which clarifies the reduction in the number of iterations. Figure 28 shows the convergence of the conjugate gradient solver for the same steps as in Fig. 26. It is visible that these systems show slightly different initial behavior in roughly the first 10 iterations, and converge with approximately the same convergence rate after that, indicating that the convergence is only very mildly affected by the local refinements.

5 Conclusion

This contribution develops a geometric multigrid preconditioner that enables iterative solutions for higher-order immersed finite element methods at a computational cost that is linear with the number of degrees of freedom. This preconditioning technique is robust to the cut elements in immersed methods, and is applicable to traditional, isogeometric, and locally refined discretizations. This is an improvement with respect to state-of-the-art preconditioning techniques for immersed finite element methods and

immersed isogeometric analysis, as these are either restricted to linear discretizations [47–49] or provide convergence rates that are dependent on the grid size [35,50,51].

A spectral analysis of immersed finite element methods reveals a spectrum that consists of a combination of (*i*) generic modes that can also be observed in mesh-fitting methods and (*ii*) modes that are characteristic to immersed finite elements and are only supported on small cut elements. Furthermore, immersed systems that are treated by a Schwarz-type preconditioner—with blocks selected such that the linear dependencies on small cut elements are resolved—possess essentially identical spectral properties as mesh-fitting methods. This confirms the observations with additive Schwarz preconditioners in [50], and opens the door to the application of the established and highly efficient framework of multigrid techniques in immersed formulations.

The presented examples convey that the developed multigrid preconditioner results in a number of iterations that is (*i*) independent of the mesh size, (*ii*) only very slightly affected by the number of levels in the multigrid cycle, and (*iii*) only marginally affected by the geometry of the problem. The computational effort to solve the systems does depend on the discretization order through the size of the Schwarz blocks (Lagrange) or the number of colors in the graph coloring algorithm and the number of iterations (Splines), which is in agreement with observations regarding mesh-fitting approaches in the literature. These observations indicate that, while the examples in this contribution already contain multi-million degrees of freedom, the preconditioned iterative solution method enables large-scale computations with immersed finite element methods. While the presented results were obtained with a sequential implementation, further upscaling of the number of degrees of freedom requires an efficient parallel implementation, for which the procedure is suitable.

The numerical results are obtained with a baseline multigrid algorithm and relatively straightforward selection of the Schwarz blocks. While the scaling of the computational cost is already optimal with respect to the system size, the framework of multigrid preconditioners with Schwarz-type smoothers allows for adjustments that can enhance the efficiency even further. First of all, as observed in [69,70,72], the efficiency of Schwarz-type smoothers in isogeometric methods is sensitive to the size and the overlap in the selection of the blocks, and different block selections have not been investigated in this contribution. Second, smoothing with multiplicative Schwarz is computationally more expensive than smoothing with additive Schwarz. While multiplicative Schwarz generally results in superior spectral properties and fewer iterations, a detailed comparison between these in the context of the overall computational cost has not been performed. Additionally, alternative smoothing

schemes such as restricted additive Schwarz can be considered [94,95]. The verification of the stability of such alternatives in immersed finite element methods does, however, require further research. Finally, the V-cycle with a single pre-smoothing and post-smoothing operation is the simplest symmetric multigrid cycle, and it is anticipated that improvements of the cycle design are possible.

Another recommendation pertains to the robustness with respect to the geometrical complexity. Different grid sizes can yield very different stiffness properties when coarse grid basis functions cover the gap between disconnected parts of the geometry, a feature that is a consequence of the geometric coarsening and would not be present in algebraic multigrid methods (AMG). This reduces the effectivity of the proposed geometric multigrid method. This effect can be precluded by a dedicated coarsening algorithm that identifies coarse grid basis functions with disjoint supports, and resolves this with a local refinement or an XFEM-type enrichment.

This contribution only considers symmetric positive definite problems. Based on the investigation of the conditioning problems for different partial differential equations in [50], this is representative for the specific cut-element-related conditioning problems in immersed finite element methods and immersed isogeometric analysis. While symmetric positive definite problems cover a large variety of problems in computational mechanics, the developed preconditioning technique is not immediately applicable to nonsymmetric and mixed formulations in, in particular, flow problems. However, multigrid methods are commonly applied in mesh-fitting flow problems, see e.g., [60], and have been observed to be effective with very similar Schwarz (or Vanka [79]) blocks in [71]. Furthermore, it is demonstrated in [50] that Schwarz-type methods can effectively resolve the cut-element-specific conditioning problems in immersed flow problems. Therefore, it is anticipated that the developed multigrid preconditioner extends mutatis mutandis to problems that are not symmetric positive definite.

References

1. Parvizian J, Düster A, Rank E (2007) Finite cell method. Comput Mech 41(1):121–133

2. Düster A, Parvizian J, Yang Z, Rank E (2008) The finite cell method for three-dimensional problems of solid mechanics. Comput Methods Appl Mech Eng 197(45):3768–3782

3. Schillinger D, Ruess M (2015) The finite cell method: a review in the context of higher-order structural analysis of CAD and image-based geometric models. Arch Comput Methods Eng 22(3):391–455

4. Burman E, Hansbo P (2012) Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. Appl Numer Math 62:328–341

5. Burman E, Claus S, Hansbo P, Larson M, Massing A (2015) Cut-FEM: discretizing geometry and partial differential equations. Int J Numer Methods Eng 104(7):472–501

6. Kamensky D, Hsu M-C, Schillinger D, Evans J, Aggarwal A, Bazilevs Y, Sacks M, Hughes T (2015) An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. Comput Methods Appl Mech Eng 284:1005–1053

7. Hsu M-C, Kamensky D, Xu F, Kiendl J, Wang C, Wu M, Mineroff J, Reali A, Bazilevs Y, Sacks M (2015) Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. Comput Mech 55(6):1211–1225

8. Schmidt R, Wüchner R, Bletzinger K-U (2012) Isogeometric analysis of trimmed NURBS geometries. Comput Methods Appl Mech Eng 241:93–111

9. Rank E, Ruess M, Kollmannsberger S, Schillinger D, Düster A (2012) Geometric modeling, isogeometric analysis and the finite cell method. Comput Methods Appl Mech Eng 249:104–115

10. Schillinger D, Dede L, Scott M, Evans J, Borden M, Rank E, Hughes T (2012) An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. Comput Methods Appl Mech Eng 249:116–150

11. Ruess M, Schillinger D, Bazilevs Y, Varduhn V, Rank E (2013) Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell methodod. Int J Numer Methods Eng 95(10):811–846

12. Ruess M, Schillinger D, Özcan A, Rank E (2014) Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. Comput Methods Appl Mech Eng 269:46–71

13. Marussig B, Hughes T (2017) A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects. Arch Comput Methods Eng 25(4):1059–1127

14. Bazilevs Y, Hsu M-C, Scott M (2012) Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. Comput Methods Appl Mech Eng 249:28–41

15. Hsu M-C, Kamensky D, Bazilevs Y, Sacks M, Hughes T (2014) Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. Comput Mech 54(4):1055–1071

16. Burman E, Fernández M (2014) An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes. Comput Methods Appl Mech Eng 279:497–514

17. Massing A, Larson M, Logg A, Rognes M (2016) A Nitsche-based cut finite element method for a fluid–structure interaction problem. Commun Appl Math Comput Sci 10(2):97–120

18. Kadapa C, Dettmer W, Perić D (2016) A fictitious domain/distributed Lagrange multiplier based fluid–structure interaction scheme with hierarchical B-spline grids. Comput Methods Appl Mech Eng 301:1–27

19. Kadapa C, Dettmer W, Perić D (2017) A stabilised immersed boundary method on hierarchical B-spline grids for fluid–rigid body interaction with solid–solid contact. Comput Methods Appl Mech Eng 318:242–269

20. Wu M, Kamensky D, Wang C, Herrema A, Xu F, Pigazzini M, Verma A, Marsden A, Bazilevs Y, Hsu M-C (2017) Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: application to a hydraulic arresting gear. Comput Methods Appl Mech Eng 316:668–693

21. Kadapa C, Dettmer W, Perić D (2018) A stabilised immersed framework on hierarchical B-spline grids for fluid–flexible structure interaction with solid–solid contact. Comput Methods Appl Mech Eng 335:472–489

22. Ruess M, Tal D, Trabelsi N, Yosibash Z, Rank E (2012) The finite cell method for bone simulations: verification and validation. Biomech Model Mechanobiol 11(3):425–437

23. Verhoosel C, Van Zwieten G, Van Rietbergen B, De Borst R (2015) Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. Comput Methods Appl Mech Eng 284:138–164

24. Elhaddad M, Zander N, Bog T, Kudela L, Kollmannsberger S, Kirschke J, Baum T, Ruess M, Rank E (2017) Multi-level hp-finite cell method for embedded interface problems with application in biomechanics. Int J Numer Methods Biomed Eng 34(4):e2951

25. Duczek S, Berger H, Gabbert U (2015) The finite pore method: a new approach to evaluate gas pores in cast parts by combining computed tomography and the finite cell method. Int J Cast Met Res 28(4):221–228

26. Würkner M, Duczek S, Berger H, Köppe H, Gabbert U (2018) A software platform for the analysis of porous die-cast parts using the finite cell method. Springer, Berlin, pp 327–341

27. Hoang T, Verhoosel CCV, Qin C-Z, Auricchio F, Reali A, van Brummelen E (2019) Skeleton-stabilized immersogeometric analysis for incompressible viscous flow problems. Comput Methods Appl Mech Eng 344:421–450

28. Parvizian J, Düster A, Rank E (2012) Topology optimization using the finite cell method. Optim Eng 13(1):57–78

29. van Dijk N, Maute K, Langelaar M, van Keulen F (2013) Level-set methods for structural topology optimization: a review. Struct Multidiscip Optim 48(3):437–472

30. Nadal E, Ródenas JJ, Albelda J, Tur M, Tarancón JE, Fuenmayor FJ (2013) Efficient finite element methodology based on cartesian grids: application to structural shape optimization. Abstr Appl Anal 2013:953786. https://doi.org/10.1155/2013/953786

31. Bandara K, Rüberg T, Cirak F (2016) Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. Comput Methods Appl Mech Eng 300:510–539

32. Groen J, Langelaar M, Sigmund O, Ruess M (2017) Higher-order multi-resolution topology optimization using the finite cell method. Int J Numer Methods Eng 110(10):903–920

33. Villanueva C, Maute K (2017) CutFEM topology optimization of 3D laminar incompressible flow problems. Comput Methods Appl Mech Eng 320:444–473

34. Burman E, Elfverson D, Hansbo P, Larson M, Larsson K (2018) Shape optimization using the cut finite element method. Comput Methods Appl Mech Eng 328:242–261

35. de Prenter F, Verhoosel C, van Zwieten G, van Brummelen E (2017) Condition number analysis and preconditioning for the finite cell method. Comput Methods Appl Mech Eng 316:297–327

36. Barrett R, Berry M, Chan T, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, van der Vorst H (1994) Templates for the solution of linear systems: building blocks for iterative methods. SIAM, Philadelphia

37. Burman E (2010) Ghost penalty. CR Math 348(21):1217–1220
38. Höllig K, Reif U, Wipper J (2001) Weighted extended B-spline approximation of Dirichlet problems. SIAM J Numer Anal 39(2):442–462
39. Höllig K, Apprich C, Streit A (2005) Introduction to the WEB-method and its applications. Adv Comput Math 23(1):215–237
40. Rüberg T, Cirak F (2012) Subdivision-stabilised immersed B-spline finite elements for moving boundary flows. Comput Methods Appl Mech Eng 209:266–283
41. Rüberg T, Cirak F (2014) A fixed-grid B-spline finite element technique for fluid–structure interaction. Int J Numer Methods Fluids 74(9):623–660
42. Rüberg T, Cirak F, García-Aznar J (2016) An unstructured immersed finite element method for nonlinear solid mechanics. Adv Model Simul Eng Sci 3(1):623–660
43. Marussig B, Zechner J, Beer G, Fries T-P (2017) Stable isogeometric analysis of trimmed geometries. Comput Methods Appl Mech Eng 316:497–521
44. Badia S, Verdugo F, Martín A (2018) The aggregated unfitted finite element method for elliptic problems. Comput Methods Appl Mech Eng 336:533–553
45. Badia S, Martín A, Verdugo F (2018) Mixed aggregated finite element methods for the unfitted discretization of the stokes problem. SIAM J Sci Comput 40(6):B1541–B1576
46. Marussig B, Hiemstra R, Hughes T (2018) Improved conditioning of isogeometric analysis matrices for trimmed geometries. Comput Methods Appl Mech Eng 334:79–110
47. Lang C, Makhija D, Doostan K, Maute A (2014) A simple and efficient preconditioning scheme for heaviside enriched XFEM. Comput Mech 54(5):1357–1374
48. Lehrenfeld C, Reusken A (2017) Optimal preconditioners for Nitsche-XFEM discretizations of interface problems. Numer Math 135(2):313–332
49. Badia S, Verdugo F (2018) Robust and scalable domain decomposition solvers for unfitted finite element methods. J Comput Appl Math 344:740–759
50. de Prenter F, Verhoosel C, van Brummelen E (2019) Preconditioning immersed isogeometric finite element methods with application to flow problems. Comput Methods Appl Mech Eng 348:604–631
51. Jomo J, de Prenter F, Elhaddad M, D'Angella D, Verhoosel C, Kollmannsberger S, Kirschke J, Nübel V, van Brummelen E, Rank E (2019) Robust and parallel scalable iterative solutions for large-scale finite cell analyses. Finite Elem Anal Des 163:14–30
52. Saad Y (2003) Iterative methods for sparse linear systems. SIAM, Philadelphia
53. Johnson C (1987) Numerical solution of partial differential equations by the finite element method. Cambridge University Press, Cambridge
54. Arnold D, Falk R, Winther R (2000) Multigrid in H(div) and H(curl). Numer Math 85(2):197–217
55. Schöberl J, Zulehner W (2003) On Schwarz-type smoothers for saddle point problems. Numer Math 95(2):377–399
56. Smith B, Bjørstad P, Gropp W (1996) Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge University Press, Cambridge
57. Hackbusch W (1985) Multi-grid methods and applications. Springer, Berlin
58. Wesseling P (1992) An introduction to multigrid methods. Wiley, Hoboken
59. Briggs W, Henson V, McCormick S (2000) A multigrid tutorial. SIAM, Philadelphia
60. Brandt A, Livne O, Techniques Multigrid (1984) Guide with applications to fluid dynamics, Revised edn. SIAM, Philadelphia, p 2011
61. Gahalaut K, Kraus J, Tomar S (2013) Multigrid methods for isogeometric discretization. Comput Methods Appl Mech Eng 253:413–425
62. Buffa A, Harbrecht H, Kunoth A, Sangalli G (2013) BPX-preconditioning for isogeometric analysis. Comput Methods Appl Mech Eng 265:63–70
63. Donatelli M, Garoni C, Manni C, Serra-Capizzano S, Speleers H (2015) Robust and optimal multi-iterative techniques for IgA Galerkin linear systems. Comput Methods Appl Mech Eng 284:230–264
64. Donatelli M, Garoni C, Manni C, Serra-Capizzano S, Speleers H (2017) Symbol-based multigrid methods for Galerkin B-spline isogeometric analysis. SIAM J Numer Anal 55(1):31–62
65. Hofreither C, Takacs S (2017) Robust multigrid for isogeometric analysis based on stable splittings of spline spaces. SIAM J Numer Anal 55(4):2004–2024
66. Hofreither C, Takacs S, Zulehner W (2017) A robust multigrid method for Isogeometric analysis in two dimensions using boundary correction. Comput Methods Appl Mech Eng 316:22–42
67. Takacs S (2018) Robust approximation error estimates and multigrid solvers for isogeometric multi-patch discretizations. Math Models Methods Appl Sci 28(10):1899–1928
68. Sogn J, Takacs S (2019) Robust multigrid solvers for the biharmonic problem in isogeometric analysis. Comput Math Appl 77(1):105–124
69. Beirão Da Veiga L, Cho D, Pavarino L, Scacchi S (2012) Overlapping Schwarz methods for isogeometric analysis. SIAM J Numer Anal 50(3):1394–1416
70. Beirão Da Veiga L, Cho D, Pavarino L, Scacchi S (2013) Isogeometric Schwarz preconditioners for linear elasticity systems. Comput Methods Appl Mech Eng 253:439–454
71. Coley C, Benzaken J, Evans J (2018) A geometric multigrid method for isogeometric compatible discretizations of the generalized Stokes and Oseen problems. Numer Linear Algebra Appl 25(3):e2145
72. de la Riva A, Rodrigo C, Gaspar F (2018) An efficient multigrid solver for isogeometric analysis. arXiv:1806.05848 [math.NA]
73. Hofreither C, Jüttler B, Kiss G, Zulehner W (2016) Multigrid methods for isogeometric analysis with THB-splines. Comput Methods Appl Mech Eng 308:96–112
74. Berger-Vergiat L, Waisman H, Hiriyur B, Tuminaro R, Keyes D (2012) Inexact Schwarz-algebraic multigrid preconditioners for crack problems modeled by extended finite element methods. Int J Numer Methods Eng 90(3):311–328
75. Hiriyur B, Tuminaro R, Waisman H, Boman E, Keyes D (2012) A quasi-algebraic multigrid approach to fracture problems based on extended finite elements. SIAM J Sci Comput 34(2):A603–A626
76. Nüssing A (2018) Fitted and unitted finite element methods for solving the EEG forward problem. Ph.D. thesis, University of Münster
77. Ludescher T, Gross S, Reusken A (2018) A multigrid method for unfitted finite element discretizations of elliptic interface problems. arXiv:1807.10196 [math.NA]
78. Verdugo F, Martín A, Badia S (2019) Distributed-memory parallelization of the aggregated unfitted finite element method. Comput Methods Appl Mech Eng 357:112583
79. Vanka S (1986) Block-implicit multigrid solution of Navier–Stokes equations in primitive variables. J Comput Phys 65(1):138–158
80. Lehrenfeld C (2016) High order unfitted finite element methods on level set domains using isoparametric mappings. Comput Methods Appl Mech Eng 300:716–733
81. Nitsche J (1971) Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen die keinen Randbedingungen unterworfen sind. In: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, pp 9–15

82. Embar A, Dolbow J, Harari I (2010) Imposing Dirichlet boundary conditions with Nitsche's method and spline based finite elements. Int J Numer Methods Eng 83(7):877–898

83. Prenter P (1975) Splines and variational methods. Wiley, Hoboken

84. Cottrell J, Hughes T, Bazilevs Y (2009) Isogeometric analysis: toward integration of CAD and FEA. Wiley, Hoboken

85. Giannelli C, Jüttler B, Speleers H (2012) THB-splines: the truncated basis for hierarchical splines. Comput Aided Geom Des 29(7):485–498

86. Bracco C, Buffa A, Giannelli C, Vazquez R (2019) Adaptive isogeometric methods with hierarchical splines: an overview. Discrete Contin Dyn Syst 39(1):241–261

87. Greenbaum A (1997) Iterative methods for solving linear systems. SIAM, Philadelphia

88. Trottenberg U, Oosterlee C, Schüller A (2000) Multigrid. Elsevier, Amsterdam

89. Bramble J, Pasciak J, Xu J (1991) The analysis of multigrid algorithms with nonnested spaces or noninherited quadratic forms. Math Comput 56(193):1–34

90. Gopalakrishnan J, Kanschat G (2003) A multilevel discontinuous Galerkin method. Numer Math 95(3):527–550

91. Harari I, Albocher U (2018) Spectral investigations of Nitsche's method. Finite Elem Anal Des 145:20–31

92. Adams M, Brezina M, Hu J, Tuminaro R (2003) Parallel multigrid smoothing: polynomial versus Gauss–Seidel. J Comput Phys 188(2):593–610

93. Toselli A, Widlund O (2005) Domain decomposition methods: algorithms and theory. Springer, Berlin

94. Cai X, Sarkis M (1999) A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J Sci Comput 21(2):792–797

95. Frommer A, Szyld D (2001) An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. SIAM J Numer Anal 39(2):463–479

96. van Zwieten G, van Zwieten J, Verhoosel C, Fonn E, Hoitinga W (2018) Nutils v3.0. https://doi.org/10.5281/zenodo.1405137

97. Tielen R, Möller M, Vuik C (2018) Efficient multigrid based solvers for isogeometric analysis. In: Proceedings of the 6th European conference on computational mechanics and the 7th European conference on computational fluid dynamics

98. Tielen R, Möller M, Göddeke D, Vuik C (2019) Efficient p-multigrid methods for isogeometric analysis. arXiv:1901.01685 [math.NA]

99. Yserentant H (1986) The convergence of multilevel methods for solving finite-element equations in the presence of singularities. Math Comput 47(176):399–409

100. Wieners C (2000) Robust multigrid methods for nearly incompressible elasticity. Computing 64(4):289–306

101. Olhoff N, Bendsøe M, Rasmussen J (1991) On CAD-integrated structural topology and design optimization. Comput Methods Appl Mech Eng 89(1):259–279

102. Kreissl S, Maute K (2012) Levelset based fluid topology optimization using the extended finite element method. Struct Multidiscip Optim 46(3):311–326

103. De S, Hampton J, Maute K, Doostan A (2019) Topology optimization under uncertainty using a stochastic gradient-based approach. arXiv:1902.04562 [math.OC]

104. Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J Optim 12(2):555–573