Condition number analysis and preconditioning of the finite cell method

Condition number analysis and preconditioning of the finite cell method

F. de Prenter[a,*], C.V. Verhoosel[a], G.J. van Zwieten[a], E.H. van Brummelen[a]

[a] *Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands*

Abstract

The (Isogeometric) Finite Cell Method – in which a domain is immersed in a structured background mesh – suffers from conditioning problems when cells with small volume fractions occur. In this contribution, we establish a rigorous scaling relation between the condition number of (I)FCM system matrices and the smallest cell volume fraction. Ill-conditioning stems either from basis functions being small on cells with small volume fractions, or from basis functions being nearly linearly dependent on such cells. Based on these two sources of ill-conditioning, an algebraic preconditioning technique is developed, which is referred to as *Symmetric Incomplete Permuted Inverse Cholesky* (SIPIC). A detailed numerical investigation of the effectivity of the SIPIC preconditioner in improving (I)FCM condition numbers and in improving the convergence speed and accuracy of iterative solvers is presented for the Poisson problem and for two- and three-dimensional problems in linear elasticity, in which Nitche's method is applied in either the normal or tangential direction. The accuracy of the preconditioned iterative solver enables mesh convergence studies of the finite cell method.

Keywords: Finite Cell Method, Isogeometric Analysis, Condition number, Preconditioning, Immersed/fictitious domain methods, Iterative solvers

1. Introduction

The Isogeometric Finite Cell Method (IFCM, *e.g.,* [1–3]) combines Isogeometric Analysis (IGA, [4, 5]) with the Finite Cell Method (FCM, [6, 7]). This combination enables application of IGA to trimmed, coupled and overlapping domains in Computer Aided Design (CAD) [8–10], topologically complex structures such as porous materials, composites and scanned data [11, 12], and problems with moving boundaries such as Fluid Structure Interactions [13–15], without laborious (re-)meshing procedures.

While FCM was initially introduced as a p-FEM method [6], it is nowadays commonly used in combination with IGA. The main advantage of FCM over standard finite element methods is that the mesh does not need to match the boundaries of the physical domain. Instead, the physical domain is immersed in a topologically and geometrically simpler encapsulating mesh, on which a myriad of approximation spaces – including Non-Uniform Rational B-splines (NURBS) – can be formed. In (I)FCM, the complexity of the physical domain is captured by an advanced integration procedure for the cells that intersect the domain boundary. Dirichlet boundary conditions on the immersed boundaries are weakly imposed using Nitsche's method [16].

When the encapsulating mesh contains cells that only intersect the physical domain on a small fraction of their volume, FCM has been found to be prone to conditioning problems. These conditioning problems impede solving the resulting system of equations. Due to this ill-conditioning, in various contributions use was made of direct solvers, *e.g.,* [1–3, 7, 9, 10]. The required memory

[*] Corresponding author. Tel.: +31 61 516 2599
Email addresses: f.d.prenter@tue.nl (F. de Prenter), c.v.verhoosel@tue.nl (C.V. Verhoosel), g.j.v.zwieten@tue.nl (G.J. van Zwieten), e.h.v.brummelen@tue.nl (E.H. van Brummelen)

and number of floating point operations for direct solvers scales different with the system size than for iterative solvers, making direct solvers unsuitable for large problems [17]. To phrase the recent review article by Schillinger and Ruess [3]: "... the development of suitable preconditioning techniques that open the door for efficient iterative solution methods in the finite cell method seems very desirable ...". As a matter of fact, various conditioning strategies have already been proposed. The most prominent of these are:

- *Fictitious domain stiffness:* In order to increase the contribution of basis functions of which only a small fraction of their support intersects the physical domain, many authors have used a virtual stiffness, *e.g.,* [3]. This implies that volumetric operators are not only integrated over the physical domain but also – multiplied by a small parameter – over the fictitious domain. See [18] for a mathematical analysis of this approach.

- *Basis function manipulation:* The adverse effect of basis functions with small supports within the physical domain can be ameliorated by eliminating them from the system. This can either be done by constraining them to geometrically nearby functions [19, 20] (as is also done in Web-splines [21, 22]) or by simply excluding them from the approximation space, *e.g.,* [12, 23, 24].

- *Ghost penalty:* The contribution of basis function with small supports within the physical domain can also be increased by adding an extra term to the variational formulation. An example of such a modification of the formulation is the addition of the Ghost penalty [25–27].

- *Basis function scaling:* Diagonal scaling of the system matrix has been demonstrated to resolve conditioning problems in specific cases in the context of linear basis functions and the extended finite element method (XFEM) [28], and in the context of weak coupling strategies in nonconforming finite element methods for diffusion-reaction problems [29].

Although these methods (and combinations thereof) can be effective in improving the conditioning of finite cell systems, they generally manipulate the weak formulation and/or approximation space, which inevitably affects the solution and may compromise stability properties of the employed approximation spaces. The exception to this is the basis function scaling approach applied in the context of XFEM, which, in fact, can be interpreted as a diagonal preconditioning technique. It is the primary objective of this work to extend this approach to be effective in the context of the (I)FCM, yielding a fully automated and robust algebraic preconditioning technique (*i.e.,* a preconditioner constructed exclusively with information from the system matrix).

In this work, we perform a detailed analysis of the conditioning problems associated with (I)FCM. An important novel contribution of this work is the derivation of an explicit scaling relation between the condition number and the smallest cell volume fraction for the discretization of elliptic second order partial differential equations using polynomial bases on uniform meshes. This relation reveals a strong dependence of the condition number on the order of the employed discretization, which corroborates the need to develop a strategy to improve the conditioning of (I)FCM. Motivated by the aforementioned relation, in this work an algebraic preconditioning technique called SIPIC (*Symmetric Incomplete Permuted Inverse Cholesky*) is developed, based on basis function scaling in combination with local orthonormalization on cells with very small volume fractions. This local orthonormalization approach is similar to the manipulation of basis functions in the Stable Generalized Finite Element Method [30]. We present the algorithm for the construction of an algebraic preconditioner, which generally preserves the sparsity pattern of the system or yields negligible fill-in. The construction of this preconditioner does not add any significant computational cost. The analysis and application in this work mainly focus on (I)FCM. We expect the SIPIC preconditioning technique to be applicable to a much broader range of numerical methods however, as the indicated sources of ill-conditioning in (I)FCM can also occur in other immersed, fictitious domain or enriched finite element methods such as XFEM and weak coupling strategies, *e.g.,* [29, 31, 32].

Section 2 of this paper presents the variational formulation of the finite cell method. In Section 3, the explicit relation between the condition number and the smallest cell volume fraction for uniform meshes is derived and numerically verified. The construction of the algebraic preconditioner is described in Section 4. In Section 5, the numerical implementation is demonstrated and the effect of the proposed preconditioner is shown for numerical examples. A novel contribution in this section is the weak imposition of Dirichlet boundary conditions in either only the normal or only the tangential direction. Furthermore, it is demonstrated that preconditioning enables detailed mesh convergence analysis of FCM. Conclusions are finally drawn in Section 6. A special conditioning technique for the local eigenvalue problem is described in Appendix A.

2. The finite cell method

Consider an elliptic second order partial differential equation over an open bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$ ($d \in \{2, 3\}$), supplemented with Dirichlet and Neumann conditions on complementary parts of the boundary $\partial\Omega$:

$$\begin{cases} -\mathrm{div}(\mathcal{D}\nabla u) = f & \text{in } \Omega, \\ n \cdot \mathcal{D}\nabla u = g^N & \text{on } \Gamma^N, \\ u = g^D & \text{on } \Gamma^D. \end{cases} \tag{2.1}$$

In this expression, n denotes the outward pointing normal vector and $\mathcal{D}$ is a bounded, self-adjoint and elliptic linear operator, *i.e.,*

$$\varepsilon\|\xi\|_e^2 \leq \xi \odot \mathcal{D}\xi \leq C_{\mathcal{D}}\|\xi\|_e^2 \quad \text{and} \quad \zeta \odot \mathcal{D}\xi = \xi \odot \mathcal{D}\zeta \quad \forall \xi, \zeta, \tag{2.2}$$

for some $0 < \varepsilon \leq C_{\mathcal{D}}$, with $\|\cdot\|_e$ denoting the Euclidean norm and $\odot$ denoting a general contraction[1]. For the Dirichlet (Γ^D) and Neumann (Γ^N) boundaries it holds that $\Gamma^D \neq \emptyset$, $\Gamma^D \cap \Gamma^N = \emptyset$ and $\overline{\Gamma^N \cup \Gamma^D} = \Gamma = \partial\Omega$. Equation (2.1) can *e.g.,* represent Poisson's problem with a scalar field u and $\mathcal{D}$ equal to identity. When u in (2.1) is vector valued, the equation can *e.g.,* be interpreted as a problem of linear elasticity.

Classical Galerkin methods for solving problems as (2.1) (such as the Finite Element Method (FEM), *e.g.,* [33, 34]) employ boundary fitted meshes on which essential boundary conditions can be imposed in a strong manner – *i.e.,* encoded into the approximation space. The Finite Cell Method (FCM) is an unfitted Galerkin method, and uses a geometrically simple mesh that encapsulates Ω such as a rectilinear discretization of $\Omega \cup \Omega_{\text{fict}}$ as shown in Figure 2.1. The variational problem equivalent to (2.1) considered in the finite cell method is of the form [16]

$$\begin{cases} \text{find } u_h \in \mathcal{V}_h(\Omega) \text{ such that:} \\ \mathcal{F}(v_h, u_h) = \ell(v_h) \quad \forall v_h \in \mathcal{V}_h(\Omega) \end{cases} \tag{2.3}$$

with

$$\mathcal{F}(v_h, u_h) = \mathcal{F}^1(v_h, u_h) + \mathcal{F}^2(v_h, u_h) + \mathcal{F}^3(v_h, u_h), \tag{2.4a}$$

$$\ell(v_h) = \int_\Omega v_h \odot f \mathrm{d}V + \int_{\Gamma^N} v_h \odot g^N \mathrm{d}S + \int_{\Gamma^D} -g^D n \odot \mathcal{D}\nabla v_h + \beta v_h \odot g^D \mathrm{d}S, \tag{2.4b}$$

and

$$\mathcal{F}^1(v_h, u_h) = \int_\Omega \nabla v_h \odot \mathcal{D}\nabla u_h \mathrm{d}V, \tag{2.5a}$$

$$\mathcal{F}^2(v_h, u_h) = \int_{\Gamma^D} -\left(v_h n \odot \mathcal{D}\nabla u_h + u_h n \odot \mathcal{D}\nabla v_h\right) \mathrm{d}S, \tag{2.5b}$$

$$\mathcal{F}^3(v_h, u_h) = \int_{\Gamma^D} \beta v_h \odot u_h \mathrm{d}S. \tag{2.5c}$$

[1] By general contraction we indicate a contraction over all tensor dimensions, *i.e.,* a multiplication of scalar values, an inner product between vectors, a double inner product for second order tensors, *etc.*

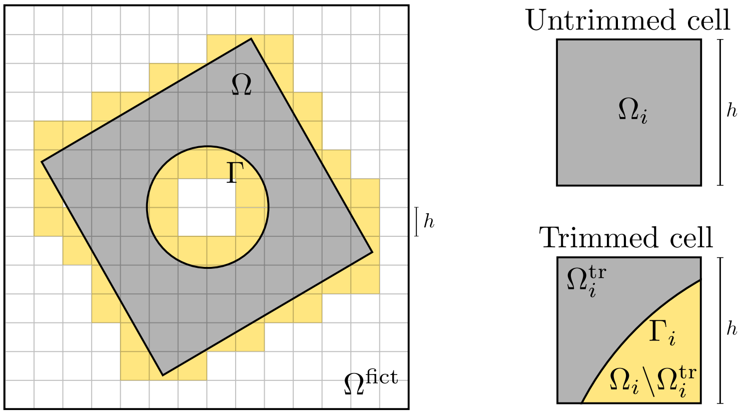


Figure 2.1: A geometrically complex domain Ω that is encapsulated by the geometrically simple, rectilinear domain $\Omega \cup \Omega_{\text{fict}}$.

In these formulations $\mathcal{V}_h(\Omega) \subset H^1(\Omega)$ denotes the finite dimensional approximation space and $\beta > 0$ is a mesh-dependent (or $\mathcal{V}_h(\Omega)$-dependent) parameter to ensure coercivity of $\mathcal{F}(\cdot, \cdot)$, as is shown in Appendix A. Because $\mathcal{F}(\cdot, \cdot)$ is coercive on finite dimensional function spaces with respect to the $H^1(\Omega)$ norm, it defines the equivalent norm

$$\|v_h\|_{\mathcal{F}}^2 = \mathcal{F}(v_h, v_h), \tag{2.6}$$

referred to as the finite cell norm (or FCM norm).

The value of β can be chosen globally or locally (*i.e.,* one constant for the whole boundary Γ^D or a separate constant for every cell that is intersected by Γ^D). The global approach is presented in [23], and is identical to the local approach when the whole domain is treated as one cell. A standard discrete trace inequality (*e.g.,* [35]) conveys that with the local approach, the lower bound for β_i on cell Ω_i is of the order of magnitude of $1/\widetilde{h}_i$, with $\widetilde{h}_i$ the typical length scale of $\Omega_i^{\text{tr}} = \Omega_i \cap \Omega$ (Figure 2.1). With the global approach, the lower bound for β is of the same order of magnitude as $\max_i \beta_i$. Consequently, one cell that is intersected by Γ^D such that its Ω_i^{tr} is small, will cause a large value of β on the entire boundary Γ^D. As a result, global stabilization negatively affects the conditioning of FCM (Appendix B). Moreover, it can degenerate FCM to a penalty method [9, 10]. Local stabilization – as employed herein – is not prone to these deficiencies. For $\mathcal{F}(\cdot, \cdot)$ to be coercive, on every cell it must hold that $\beta_i > C_i$, with [23]

$$C_i = \max_{v_h \in \mathcal{V}_h(\Omega_i)} \frac{\|n \cdot \mathcal{D}\nabla v_h\|_{L^2(\Gamma_i^D)}^2}{\mathcal{F}_i^1(v_h, v_h)}, \tag{2.7}$$

where $\mathcal{V}_h(\Omega_i)$ denotes the restriction of the approximation space $\mathcal{V}_h((\Omega)$ to Ω_i, $\Gamma_i^D = \Omega_i \cap \Gamma^D$ is the part of the boundary Γ^D contained in cell Ω_i (Figure 2.1), and

$$\mathcal{F}_i^1(v_h, u_h) = \int_{\Omega_i^{\text{tr}}} \nabla v_h \odot \mathcal{D}\nabla u_h \mathrm{d}V. \tag{2.8}$$

A robust and well-conditioned method to compute C_i is described in Appendix A. In this work, $\beta_i = 2C_i$ is applied for all examples and test cases. This choice is essentially arbitrary and does not affect the strategy to construct a preconditioner.

3. Condition number analysis

The condition number of an invertible matrix $\mathbf{A}$ is defined as

$$\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 \geq 1, \tag{3.1}$$

with $\|\cdot\|_2$ the (induced) Euclidean norm:

$$\|\mathbf{A}\|_2 = \max_{\mathbf{x}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}. \tag{3.2}$$

When solving a linear system of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, the condition number $\kappa_2(\mathbf{A})$ is of major importance. Well-conditioned systems (condition numbers close to 1) have a smaller propagation of errors in $\mathbf{A}$ or $\mathbf{b}$, converge faster when solved iteratively, and have a smaller uncertainty when the error is estimated using the residual. For example, when $\mathbf{A}$ is symmetric positive definite – such as system matrices originating from most FCM approximations – the Conjugate Gradient method has a convergence bound that depends on the condition number as

$$\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}} \leq 2 \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^i \|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{A}}, \tag{3.3}$$

where $\mathbf{x}_i$ is the approximation after i iterations and $\|\cdot\|_{\mathbf{A}}$ denotes the energy norm $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^T\mathbf{A}\mathbf{x}$ [17]. Hence, the guaranteed reduction in the error per iteration is much larger for $\kappa_2(\mathbf{A}) \sim 1$ than for $\kappa_2(\mathbf{A}) \gg 1$. The estimate of $\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}}$ in terms of the residual $\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|_2$ also depends on the condition number:

$$\frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|_2^2}{\|\mathbf{A}\|_2} \leq \|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}}^2 \leq \kappa_2(\mathbf{A}) \frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|_2^2}{\|\mathbf{A}\|_2}. \tag{3.4}$$

Equation (3.4) conveys that, in the case of ill-conditioning, reduction of the residual to a value below a specified tolerance does not guarantee convergence by the same amount of the (unknown) error in the solution (*e.g.,* the error in the energy norm).

Because the bilinear operator in (2.4) is symmetric and coercive, system matrices originating from this FCM formulation are Symmetric Positive Definite (SPD). In Section 3.1 we derive an equivalent definition of the norm for SPD matrices, which enables the application to FCM system matrices in Section 3.2. In Section 3.3 we verify the obtained theoretical estimate of the condition number with an example.

3.1. Equivalent norms for SPD matrices

Because symmetric matrices have orthogonal eigenvectors, definition (3.2) implies

$$\|\mathbf{A}\|_2 = |\lambda|_{\max} \qquad \text{and} \qquad \|\mathbf{A}^{-1}\|_2 = \frac{1}{|\lambda|_{\min}}, \tag{3.5}$$

in which $|\lambda|_{\max}$ and $|\lambda|_{\min}$ denote the absolute maximal and minimal eigenvalues of $\mathbf{A}$, respectively. Furthermore, for symmetric matrices the largest and smallest eigenvalues correspond to the maximal and minimal Rayleigh quotients

$$\lambda_{\max} = \max_{\mathbf{y}} \frac{\mathbf{y}^T\mathbf{A}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \qquad \text{and} \qquad \lambda_{\min} = \min_{\mathbf{y}} \frac{\mathbf{y}^T\mathbf{A}\mathbf{y}}{\mathbf{y}^T\mathbf{y}}. \tag{3.6}$$

If $\mathbf{A}$ is positive definite as well (*i.e.,* $\mathbf{A}$ is SPD), all eigenvalues are positive such that $\lambda_{\max} = |\lambda|_{\max}$ and $\lambda_{\min} = |\lambda|_{\min}$. Combination of (3.5) and (3.6) yields

$$\|\mathbf{A}\|_2 = \max_{\mathbf{y}} \frac{\mathbf{y}^T\mathbf{A}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \qquad \text{and} \qquad \|\mathbf{A}^{-1}\|_2 = \max_{\mathbf{y}} \frac{\mathbf{y}^T\mathbf{y}}{\mathbf{y}^T\mathbf{A}\mathbf{y}}, \tag{3.7}$$

which shows that the norm and inverse norm of SPD matrices are determined by the maximal and minimal quotient of the energy norm and the Euclidean norm of a vector.

3.2. Condition numbers in FCM

To apply (3.7) to system matrices originating from FCM formulations similar to (2.4), we consider a function $v_h = \mathbf{\Phi}^T \mathbf{y} \in \mathcal{V}_h(\Omega)$ for some unique $\mathbf{y} \in \mathbb{R}^n$, with $\mathbf{\Phi}$ denoting the vector containing all basis functions. The FCM system matrix is defined as

$$\mathbf{A} = \mathcal{F}(\mathbf{\Phi}, \mathbf{\Phi}^T), \tag{3.8}$$

and the following is noted:

$$\|\mathbf{y}\|_{\mathbf{A}}^2 = \mathbf{y}^T \mathbf{A} \mathbf{y} = \mathcal{F}(\mathbf{y}^T \mathbf{\Phi}, \mathbf{\Phi}^T \mathbf{y}) = \mathcal{F}(v_h, v_h) = \|v_h\|_{\mathcal{F}}^2, \tag{3.9}$$

showing that the FCM norm of a function v_h is equal to the energy norm of the corresponding vector $\mathbf{y}$ with system matrix $\mathbf{A}$. Therefore the $\| \cdot \|_2$ norm of $\mathbf{A}$ and its inverse according to (3.7) can be interpreted as the quotient of the FCM norm of a function and the Euclidean norm of the corresponding vector:

$$\|\mathbf{A}\|_2 = \max_{v_h, \mathbf{y}} \frac{\|v_h\|_{\mathcal{F}}^2}{\|\mathbf{y}\|_2^2}, \quad \|\mathbf{A}^{-1}\|_2 = \max_{v_h, \mathbf{y}} \frac{\|\mathbf{y}\|_2^2}{\|v_h\|_{\mathcal{F}}^2}. \tag{3.10}$$

When a function v_h exists for which the FCM norm $\|v_h\|_{\mathcal{F}}$ is very small in relation to the corresponding norm $\|\mathbf{y}\|_2$ of the coefficients vector, ill-conditioning can occur on account of the norm $\|\mathbf{A}^{-1}\|_2$ being very large. This is a situation typically encountered in FCM. Since in FCM the relative position of the mesh to the physical domain Ω is arbitrary, the volume of a trimmed cell $\Omega_i^{\mathrm{tr}} = \Omega_i \cap \Omega$ (Figure 2.1) can be arbitrarily small. When a function is only supported on Ω_i^{tr}, its support[2]– and as a result also its FCM norm – can therefore be arbitrarily small as well. The vector that corresponds to this function does not depend on the volume of Ω_i^{tr}, however. As a result, the inverse norm $\|\mathbf{A}^{-1}\|_2$ in (3.10) can become arbitrarily large. In contrast, the norm $\|\mathbf{A}\|_2$ in itself is not sensitive to the relative position of the mesh. Consequently, FCM can yield arbitrarily large condition numbers, in case the mesh is positioned such that a, or multiple, small trimmed cells occur.

To provide a quantitative estimate of the condition number, we restrict ourselves to situations where:

- The number of dimensions d, is larger than one.
- A local stabilization parameter β_i, is used, as described in Section 2.
- A piecewise polynomial basis is applied (*e.g.,* B-splines, Lagrange, or spectral bases).
- A uniform mesh with mesh size h is used.

The restriction to uniform meshes is made to simplify the derivation, but is not necessary for the final estimate. The volume fraction η_i is defined as the fraction of cell Ω_i that intersects physical domain Ω,

$$\eta_i = \frac{|\Omega_i \cap \Omega|}{|\Omega_i|} = \frac{|\Omega_i^{\mathrm{tr}}|}{|\Omega_i|} = \frac{|\Omega_i^{\mathrm{tr}}|}{h^d}. \tag{3.11}$$

The index of the smallest volume fraction is denoted by

$$\imath = \arg\min_i \eta_i, \tag{3.12}$$

such that the smallest volume fraction is $\eta = \eta_\imath$. Furthermore, we make three assumptions on the shape of the trimmed cells:

[2] In the remainder of this manuscript, the term support refers to the support in the physical domain.

1. $\partial\Omega_\imath \cap \Omega$, the intersection between the (untrimmed) boundary of the cell with the smallest volume fraction and the physical domain Ω, contains parts of at most one of two opposing cell faces. For example, if $\Omega_\imath = (0,1)^d$, then
$$\big\{x \in \Omega_\imath^{\mathrm{tr}} | x_j = 0\big\} \neq \emptyset \quad \Rightarrow \quad \big\{x \in \Omega_\imath^{\mathrm{tr}} | x_j = 1\big\} = \emptyset, \tag{3.13}$$
for all indices $j \leq d$.
2. There exists a positive constant C_R such that for every cell Ω_i, the radius R_i of the smallest ball enclosing Ω_i^{tr} is bounded by
$$R_i \leq C_R |\Omega_i^{\mathrm{tr}}|^{\frac{1}{d}}. \tag{3.14}$$
It follows from (3.11) that $R_i \leq C_R h \eta_i^{\frac{1}{d}}$.
3. There exists a positive constant C_Γ such that for every cell Ω_i that is trimmed by Γ, the surface measure of $\Gamma_i = \Gamma \cap \Omega_i$ (Figure 2.1) is bounded by
$$|\Gamma_i| \leq C_\Gamma |\Omega_i^{\mathrm{tr}}|^{\frac{d-1}{d}} = C_\Gamma h^{d-1} \eta_i^{\frac{d-1}{d}}. \tag{3.15}$$

The first assumption implies that there exists a function in the approximation space that is only supported on $\Omega_\imath$. For η small enough, this first assumption is automatically satisfied if assumption 2 or 3 holds. The second assumption guarantees shape regularity of trimmed cells, *e.g.,* [13, 36]. This assumption underlies the FCM method, and serves to keep the penalty parameter bounded. This excludes pathological cases, such as trimmed cells with highly distorted aspect ratios. The third assumption bounds the size of the intersection between Γ and a cell by the size of the trimmed cell. Under assumptions 2 and 3 and using the bound on $\mathcal{D}$ from (2.2), it can be shown that $\exists C_\beta > 0$ such that the local stabilization parameter satisfies
$$\beta_i \leq C_\beta C_{\mathcal{D}} |\Omega_i^{\mathrm{tr}}|^{-1/d} = C_\beta C_{\mathcal{D}} h^{-1} \eta_i^{-1/d}, \tag{3.16}$$
see *e.g.,* [35]. The restrictions and assumptions mentioned above, are only required to prove an estimate of the condition number of the FCM system matrix. The strategy to construct a preconditioner (proposed in Section 4) does not depend on these restrictions and assumptions.

From the literature (*e.g.,* [37]) we know that the condition number of system matrices originating from classical FEM approximations of second order elliptic differential equations with quasi-uniform meshes, scales with h^{-2}. The norm and largest eigenvalue of such matrices scale with h^{d-2} and the corresponding eigenvector represents the function with the highest frequency that can be represented on the mesh. The smallest eigenvalue of such a matrices scales with h^d and the corresponding eigenvector represents the function with the lowest frequency. Note that this function approximates the analytical lowest eigenmode of the operator, which is independent of the mesh size h. However, the norm of the corresponding coefficient vector does depend on h by virtue of the fact that the dimension of the approximation space is altered under mesh refinement. Because $\mathcal{F}^1(\cdot,\cdot)$ coincides with the standard FEM bilinear form, the lower bound of the norm $\|\mathbf{A}\|_2$ scales with at least h^{d-2}. Under assumptions 2 and 3, it can be verified that for a local stabilization parameter (see Appendix B for details)[3]
$$\frac{|\mathcal{F}^2(v_h, v_h)|}{\mathbf{y}^T\mathbf{y}} \leq C h^{d-2} \qquad \text{and} \qquad \frac{\mathcal{F}^3(v_h, v_h)}{\mathbf{y}^T\mathbf{y}} \leq C h^{d-2}. \tag{3.17}$$
Therefore the steepest lower bound for the largest eigenvalue of $\mathbf{A}$ indeed scales with
$$\|\mathbf{A}\|_2 \geq c_{\mathcal{F}} h^{d-2}, \tag{3.18}$$
for some $c_{\mathcal{F}} > 0$. We emphasize that (3.17) only holds for locally stabilized systems. A derivation of (3.17) and an elaboration of the effect of global stabilization is given in Appendix B.

[3] In the remainder of this manuscript, the variables c and C, without subscript or superscript, denote general positive constants that may attain different values in different statements or equations.

To estimate $\|\mathbf{A}^{-1}\|$ using (3.10), we need to evaluate the smallest eigenvalue. As mentioned before, for matrices originating from classical FEM formulations, the eigenvector that corresponds to the smallest eigenvalue represents the function with the lowest frequency. This is generally not the case for matrices originating from FCM formulations. In most FCM cases, the smallest eigenvalue corresponds to a function which is only supported on a cell with a very small volume fraction. Under assumption 1, there exists a function v_h with a corresponding coefficient vector of order one, $\|\mathbf{y}\|_2 \sim 1$, that is only supported on $\Omega_\imath^{\mathrm{tr}}$:

$$v_h|_{\Omega_\imath^{\mathrm{tr}}} = \xi \prod_{j=1}^{d} \left(\frac{x_j - \hat{x}_j}{h} \right)^p , \tag{3.19}$$

In (3.19), p is the order of the discretization and j is the index of the dimension. In case v_h is vector valued ξ is some vector with Euclidean norm $\|\xi\|_e = 1$, if v_h is a scalar then $\xi = 1$. Furthermore $\hat{x} \in \partial\Omega_\imath \cap B_{R_\imath}(\Omega_\imath^{\mathrm{tr}})$ – *i.e.*, $\hat{x}$ lies in the intersection between the (untrimmed) boundary of $\Omega_\imath$ and the smallest ball enclosing $\Omega_\imath^{\mathrm{tr}}$. For example, if there is a vertex in $\partial\Omega_\imath \cap B_{R_\imath}(\Omega_\imath^{\mathrm{tr}})$ (under assumption 1 there can be at most one), then $\hat{x}$ coincides with this vertex. Under assumption 2, we can show that for the function v_h according to (3.19), it holds that

$$\|v_h\|_{L^\infty(\Omega_\imath^{\mathrm{tr}})} \leq \left(\frac{2R_\imath}{\sqrt{d}h} \right)^{pd} \leq \left(\frac{2C_R}{\sqrt{d}} \right)^{pd} \eta^p, \tag{3.20}$$

and

$$\|\nabla v_h\|_{L^\infty(\Omega_\imath^{\mathrm{tr}})} \leq \frac{p\sqrt{d}}{h} \left(\frac{2R_\imath}{\sqrt{d}h} \right)^{pd-1} \leq p\sqrt{d} \left(\frac{2C_R}{\sqrt{d}} \right)^{pd-1} \frac{\eta^{p-1/d}}{h}, \tag{3.21}$$

where $\|\cdot\|_{L^\infty}$ denotes the supremum of the Euclidean norm. Recalling assumption 3 and the upper bound for $\beta_\imath$ given in (3.16), the following bounds hold

$$\begin{aligned}
\mathcal{F}^1(v_h, v_h) &= \int_{\Omega_\imath^{\mathrm{tr}}} \nabla v_h \odot \mathcal{D} \nabla v_h \mathrm{d}V \leq |\Omega_\imath^{\mathrm{tr}}| C_{\mathcal{D}} \|\nabla v_h\|^2_{L^\infty(\Omega_\imath^{\mathrm{tr}})} && \text{(3.22a)}\\
&\leq C_{\mathcal{D}} p^2 d \left(\frac{2C_R}{\sqrt{d}} \right)^{2pd-2} h^{d-2} \eta^{2p+1-2/d}, \\
\mathcal{F}^2(v_h, v_h) &\leq \int_{\Gamma_\imath^D} 2 \|v_h\|_e C_{\mathcal{D}} \|\nabla v_h\|_e \mathrm{d}S && \text{(3.22b)}\\
&\leq 2|\Gamma_\imath^D| \|v_h\|_{L^\infty(\Omega_\imath^{\mathrm{tr}})} C_{\mathcal{D}} \|\nabla v_h\|_{L^\infty(\Omega_\imath^{\mathrm{tr}})} \\
&\leq 2C_\Gamma C_{\mathcal{D}} p\sqrt{d} \left(\frac{2C_R}{\sqrt{d}} \right)^{2pd-1} h^{d-2} \eta^{2p+1-2/d}, \\
\mathcal{F}^3(v_h, v_h) &= \int_{\Gamma_\imath^D} \beta_\imath v_h \odot v_h \mathrm{d}S \leq |\Gamma_\imath^D| \beta_\imath \|v_h\|^2_{L^\infty(\Omega_\imath^{\mathrm{tr}})} && \text{(3.22c)}\\
&\leq C_\Gamma C_\beta C_{\mathcal{D}} \left(\frac{2C_R}{\sqrt{d}} \right)^{2pd} h^{d-2} \eta^{2p+1-2/d},
\end{aligned}$$

which enables us to give an upper bound for the complete FCM norm of the function specified in (3.19):

$$\|v_h\|^2_{\mathcal{F}} \leq C_{\mathcal{F}} h^{d-2} \eta^{2p+1-2/d}, \tag{3.23}$$

with $C_{\mathcal{F}}$ according to

$$C_{\mathcal{F}} = C_{\mathcal{D}} p^2 d \left(\frac{2C_R}{\sqrt{d}} \right)^{2pd-2} + 2C_\Gamma C_{\mathcal{D}} p\sqrt{d} \left(\frac{2C_R}{\sqrt{d}} \right)^{2pd-1} + C_\Gamma C_\beta C_{\mathcal{D}} \left(\frac{2C_R}{\sqrt{d}} \right)^{2pd} . \tag{3.24}$$

Because $\|\mathbf{y}\|_2 \sim 1$ for the function specified in (3.19), there exists a constant $c_{\mathcal{F}^{-1}} \sim 1/C_{\mathcal{F}}$ such that

$$\|\mathbf{A}^{-1}\| \geq c_{\mathcal{F}^{-1}} \frac{\eta^{-(2p+1-2/d)}}{h^{d-2}}. \tag{3.25}$$

Combining (3.18) and (3.25), we obtain the following *lower bound* for the FCM condition number:

$$\kappa_2(\mathbf{A}) \geq c_{\mathcal{F}} c_{\mathcal{F}^{-1}} \eta^{-(2p+1-2/d)}. \tag{3.26}$$

The condition number of matrices originating from FCM formulations is therefore bounded from below by a constant scaling with $\eta^{-(2p+1-2/d)}$. The dependence of the scaling rate on the order of the approximation space conveys that the combination of FCM with higher-order methods, such as p-FEM and IGA, makes it particularly susceptible to conditioning problems.

3.3. Numerical test case: Poisson's problem on a unit square with circular exclusion

To verify the scaling relation (3.26), we consider the finite cell discretization of Poisson's problem with Dirichlet conditions on the entire boundary:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g^D & \text{on } \partial\Omega, \end{cases} \tag{3.27}$$

such that

$$\mathcal{F}(v_h, u_h) = \int_\Omega \nabla v_h \cdot \nabla u_h \mathrm{d}V + \int_{\Gamma^D} -(v_h n \cdot \nabla u_h + u_h n \cdot \nabla v_h) + \beta v_h u_h \mathrm{d}S. \tag{3.28}$$

Note that the data f and g_D in (3.27) do not appear in the system matrix and hence are not required for the computation of the condition number. The results presented in this section pertain to finite cell systems with a local stabilization parameter $\beta_i = 2C_i$ (see Section 2). Results for global stabilization – although not presented here – demonstrate the validity of scaling relation (B.8).

The domain Ω is the unit square ($L = 1$) with a centered circular exclusion of radius R, *i.e.,* $\Omega = \{x \in (-\frac{1}{2}, \frac{1}{2})^2 : \|x\|_e > R\}$. We partition the embedding domain ($\Omega \cup \Omega_{\text{fict}}$) with a Cartesian mesh with mesh size $h = \frac{1}{32}$, with the center of the exclusion positioned at a vertex of the mesh. This is illustrated in Figure 3.1, where the cells in white do not intersect the physical domain and hence do not contribute to the system matrix. Note that for graphical clarity the figure is not to scale. The radius R is taken equal to $R = \sqrt{1/8 - \sqrt{\eta_R/2}h} \approx \frac{1}{4}\sqrt{2}$, such that a volume fraction of approximately $\eta_R = 5 \cdot 10^{-3}$ occurs at the boundary of the circular exclusion. We consider uniform bivariate B-spline bases [4] of order $p \in \{1, 2, 3, 4\}$ and Lagrange bases of order $p \in \{1, 2\}$. The bisection-based tessellation scheme proposed in [12] with a maximal refinement depth of two is employed to accurately approximate the geometry of the domain. The number of Gauss points is selected such that exact integration of all operators is achieved over the tessellated domain. Figure 3.1 shows the interior (blue circles) and boundary (red squares) integration points for the linear bases.

To construct scenarios with different volume fractions, the domain is gradually rotated about the origin while the background mesh remains unmodified. Starting from a domain in which the outer boundaries are aligned with the mesh, it is rotated in 100 steps over an angle of 45°. Because elements with a volume fraction of $\eta_i \approx \eta_R = 5 \cdot 10^{-3}$ occur at the boundary of the circular exclusion regardless of the angle over which the domain is rotated, the minimal volume fraction η, is bounded from above by approximately η_R.

A power algorithm is used to compute the smallest and largest eigenvalues of the system matrix, and thereby the condition number for every configuration. The power algorithm is terminated when the Rayleigh quotient of two subsequent vectors has a relative difference of less than 10^{-6}. The inverse power method used to compute the smallest eigenvalue, relies on the inversion of the ill-conditioned finite cell system matrix. The preconditioner developed in this manuscript is used to reliably execute this inverse power iteration. As we will elaborate in the next section, the underlying mechanism leading to ill-conditioning is generally different for B-spline bases and Lagrange bases. As a result of this difference, the condition numbers for B-spline discretizations can be computed accurately over the full range of data shown in Figure 3.2, while machine precision

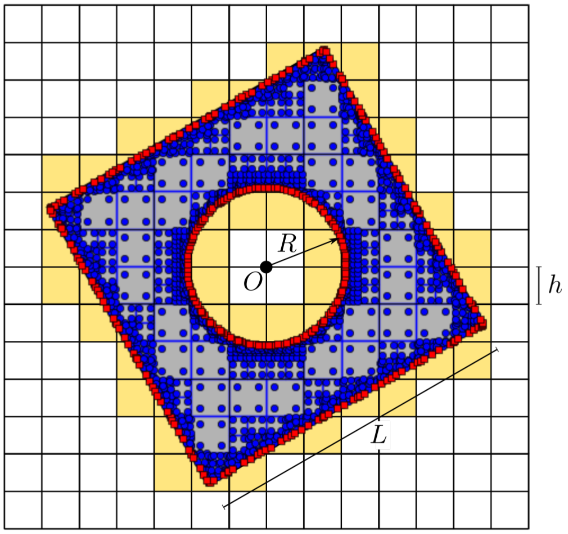


Figure 3.1: Schematic representation of a unit square domain with circular exclusion. Cells that intersect the physical domain are depicted in yellow. Volumetric (Gauss) integration points are indicated in blue and boundary integration points in red.

hinders the computation of condition numbers for Lagrange bases of third and fourth order (see Appendix C). Therefore, we do not present results for the Lagrange bases for $p \in \{3, 4\}$.

Figure 3.2 shows the resulting condition numbers plotted against the volume fractions. The upper bound of the volume fraction $\eta_R = 5 \cdot 10^{-3}$ is observed from these plots. The predicted scaling rates of the condition number according to (3.26) are depicted in black. Evidently, the numerical results for all considered basis functions closely resemble these scaling rates. Also in agreement with the above derivations is the observation that the condition numbers obtained for Lagrange bases are very similar to those obtained for B-splines of equal order.

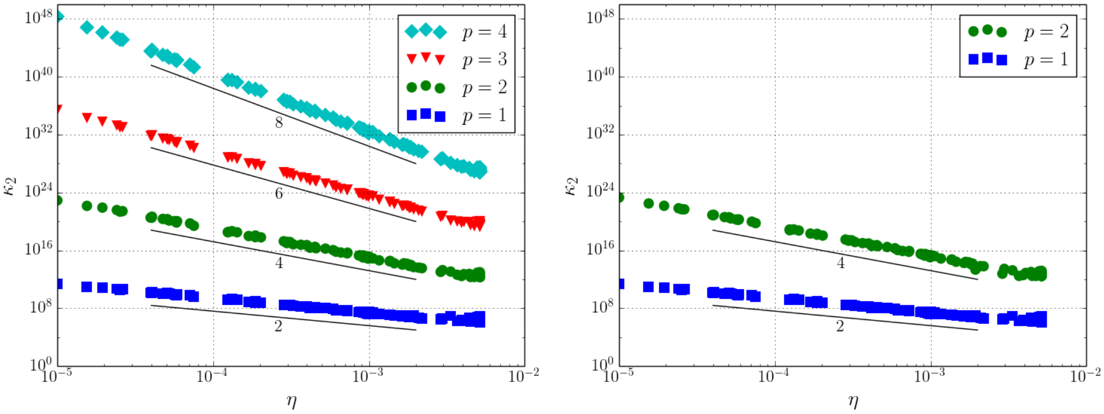
4. Algebraic preconditioning for FCM

Motivated by the condition number analysis presented above, in this section we develop an FCM preconditioner. The fundamental observation on which our developments are based, is that ill-conditioning of finite cell systems is caused by the occurrence of small eigenvalues in the system matrix $\mathbf{A}$ due to small trimmed cell volume fractions (see (3.25)). Recall that from equations (3.5) and (3.10), it follows that the smallest eigenvalue for finite cell systems can be expressed as

$$\lambda_{\min} = \frac{1}{\|\mathbf{A}^{-1}\|_2} = \min_{\mathbf{y}} \frac{\|\mathbf{y}\|^2_{\mathbf{A}}}{\|\mathbf{y}\|^2_2} = \min_{v_h, \mathbf{y}} \frac{\|v_h\|^2_{\mathcal{F}}}{\|\mathbf{y}\|^2_2}, \tag{4.1}$$

where the function $v_h \in \mathcal{V}_h(\Omega)$ corresponds to a vector $\mathbf{y} \in \mathbb{R}^n$ by the relation $v_h = \mathbf{\Phi}^T \mathbf{y}$, with $\mathbf{\Phi}$ denoting the vector containing all functions in the basis of $\mathcal{V}_h(\Omega)$. Hence, ill-conditioning occurs when the basis allows functions v_h and corresponding vectors $\mathbf{y}$ for which $\|v_h\|_{\mathcal{F}} \ll \|\mathbf{y}\|_2$. To improve the conditioning, we construct an alternative basis, $\overline{\mathbf{\Phi}} = \mathbf{S}\mathbf{\Phi}$, which precludes these large differences between the norm of functions and the norm of the coefficient vectors. For nonsingular preconditioning matrices $\mathbf{S}$, both $\mathbf{\Phi}$ and its preconditioned counterpart $\overline{\mathbf{\Phi}}$ span the same approximation space $\mathcal{V}_h(\Omega)$.

For every function $v_h \in \mathcal{V}_h(\Omega)$, there are unique coefficient vectors $\mathbf{y}$ and $\overline{\mathbf{y}}$ such that $v_h = \mathbf{\Phi}^T \mathbf{y} = \overline{\mathbf{\Phi}}^T \overline{\mathbf{y}} = \mathbf{\Phi}^T \mathbf{S}^T \overline{\mathbf{y}}$, from which it follows that $\mathbf{y} = \mathbf{S}^T \overline{\mathbf{y}}$. The preconditioned solution $\overline{\mathbf{x}}$, is obtained by solving the symmetrically preconditioned linear system $\overline{\mathbf{A}}\,\overline{\mathbf{x}} = \overline{\mathbf{b}}$, where the



(a) B-splines

(b) Lagrange

Figure 3.2: Condition number *vs.* the smallest volume fraction for the unit square with circular exclusion for (a) B-splines of order $p \in \{1, 2, 3, 4\}$, and (b) Lagrange polynomials of order $p \in \{1, 2\}$.

preconditioned system matrix $\overline{\mathbf{A}}$ and right hand side vector $\overline{\mathbf{b}}$ are given by

$$\begin{aligned} \overline{\mathbf{A}} = \mathcal{F}(\overline{\mathbf{\Phi}}, \overline{\mathbf{\Phi}}^T) &\quad = \mathcal{F}(\mathbf{S\Phi}, \mathbf{\Phi}^T\mathbf{S}^T) &&\quad = \mathbf{S}\mathcal{F}(\mathbf{\Phi}, \mathbf{\Phi}^T)\mathbf{S}^T &&\quad = \mathbf{SAS}^T, \\ \overline{\mathbf{b}} = \ell(\overline{\mathbf{\Phi}}) &\quad = \ell(\mathbf{S\Phi}) &&\quad = \mathbf{S}\ell(\mathbf{\Phi}) &&\quad = \mathbf{Sb}. \end{aligned} \tag{4.2}$$

In these expressions, $\mathbf{A}$ and $\mathbf{b}$ are the system matrix and right hand side vector corresponding to the original basis $\mathbf{\Phi}$. Given an original system $\mathbf{Ax} = \mathbf{b}$ and preconditioning matrix $\mathbf{S}$, the resulting symmetrically preconditioned system is

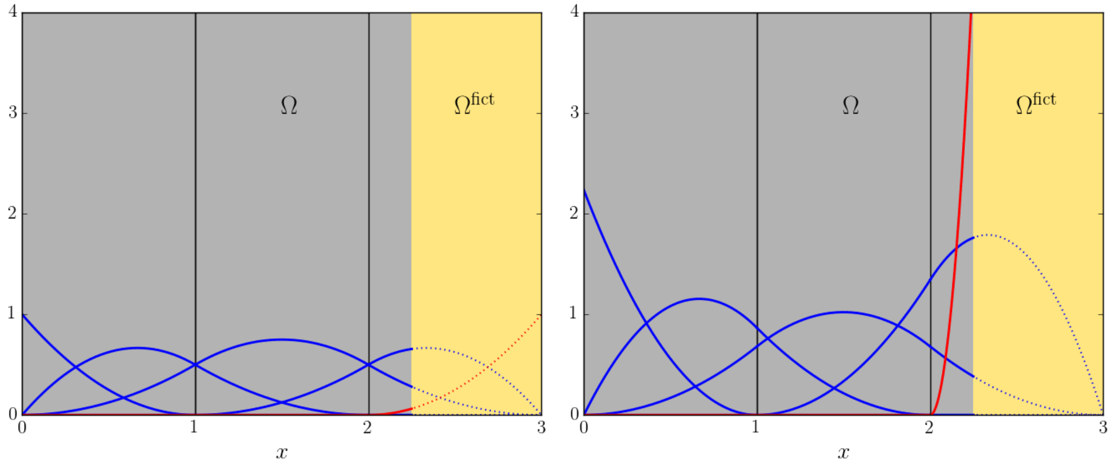
$$\begin{aligned} \mathbf{SAS}^T\overline{\mathbf{x}} &= \mathbf{Sb}, \\ \mathbf{x} &= \mathbf{S}^T\overline{\mathbf{x}}. \end{aligned} \tag{4.3}$$

In the remainder of this section, we will discuss the development of an effective FCM preconditioner. In Section 4.1, we first identify two sources for the occurrence of small eigenvalues. Each source of ill-conditioning can be remedied by a modification of the basis through a preconditioning matrix, which can be constructed algebraically (based solely on the original system matrix). Because this preconditioner is not interwoven with the rest of the method, it is robust and straightforward to automate and implement. Since this preconditioner does not manipulate the weak form problem or approximation space, the obtained solution is unaffected, and stability properties of the employed approximation space (*e.g.,* the inf-sup condition for mixed methods) are maintained. In Section 4.2, we will discuss how the developed preconditioner – which we refer to as *Symmetric Incomplete Permuted Inverse Cholesky* (SIPIC) – is related to well-established preconditioning strategies. Finally, we demonstrate the effectivity of the SIPIC preconditioner by means of numerical simulations in Section 4.3 and by a comparison to commonly used preconditioning techniques in Section 4.4. Various implementation aspects are discussed in Appendix C.

4.1. Small eigenvalues: sources and remedies

4.1.1. Diagonal scaling

A small eigenvalue can be caused by the occurrence of a (trimmed) basis function ϕ, with a small FCM norm relative to other (possibly untrimmed) basis functions. This can be inferred from equation (4.1) by considering the standard unit vector $\mathbf{y}$ ($\|\mathbf{y}\|_2 = 1$) corresponding to $v_h = \phi$. This situation is typical for B-spline basis functions whose support contains only a single trimmed cell with a small volume fraction, since all the derivatives up to order $p - 1$ of such basis functions vanish on the (original) boundaries of that cell. We illustrate this situation in Figure 4.1a, where the basis function with a small norm (depicted in red) is similar to the function specified in (3.19).



(a) Original

(b) Scaled

Figure 4.1: Second order B-spline basis functions over the domain $\Omega = (0, 2\frac{1}{4})$ created using the knot vector $[0, 0, 0, 1, 2, 3, 3, 3]$: (a) before scaling, and (b) after scaling. Note that the value and derivative of the rescaled function still vanish at $x = 2$, and that hence the continuity of the basis is unaffected.

To remedy this source of small eigenvalues, all basis functions ϕ are normalized with respect to the norm $\|\phi\|_{\mathcal{F}}$, which yields the modified basis functions $\phi^* = \phi/\|\phi\|_{\mathcal{F}}$. This diagonal scaling operation can be cast into the form of a preconditioning matrix $\mathbf{D}$ whose diagonal corresponds to the reciprocal of the square root of the main diagonal of system matrix $\mathbf{A}$:

$$\mathbf{D} = \begin{bmatrix} \frac{1}{\sqrt{A_{11}}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{A_{nn}}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\|\phi_1\|_{\mathcal{F}}} & & \\ & \ddots & \\ & & \frac{1}{\|\phi_n\|_{\mathcal{F}}} \end{bmatrix}. \tag{4.4}$$

In Figure 4.1b, we illustrate the effect of this diagonal scaling operation for the basis $\mathbf{\Phi}$ in Figure 4.1a, where the preconditioned basis follows from $\mathbf{\Phi}^* = \mathbf{D}\mathbf{\Phi}$. This operation does not preserve the partition of unity of the basis, but as the space itself is unaltered and remains polynomially complete, this does not affect the solution. This also follows from the fact that the preconditioning matrix $\mathbf{D}$ is nonsingular. We note that diagonal preconditioning has also been demonstrated to be effective in ameliorating similar ill-conditioning problems encountered in XFEM and weak coupling strategies [28, 29].

4.1.2. Local orthonormalization

Even if basis functions are properly scaled, small eigenvalues can occur when the global linear independence property of two (or more) basis functions is compromised by a trimming operation. This results in the near linear dependence of rows and columns of $\mathbf{A}$ associated with the trimmed cell basis functions and, accordingly, results in ill-conditioning of $\mathbf{A}$. For example, consider two (scaled) basis functions, ϕ^*_α and ϕ^*_β, that are very similar with respect to the finite cell norm, *i.e.,* $\|\phi^*_\alpha - \phi^*_\beta\|_{\mathcal{F}} \ll 1$ (or $\|\phi^*_\alpha + \phi^*_\beta\|_{\mathcal{F}} \ll 1$), where use is made of $\|\phi^*_\alpha\|_{\mathcal{F}} = \|\phi^*_\beta\|_{\mathcal{F}} = 1$. The difference (or sum) of these functions corresponds to a coefficient vector $\mathbf{y}$ with $\|\mathbf{y}\|_2 = \sqrt{2}$. From Equation (4.1) it then follows that the minimal eigenvalue is bounded from above by $\frac{1}{2}\|\phi^*_\alpha - \phi^*_\beta\|^2_{\mathcal{F}} \ll 1$ (or $\frac{1}{2}\|\phi^*_\alpha + \phi^*_\beta\|^2_{\mathcal{F}} \ll 1$). We refer to this source of ill-conditioning as *quasi linear dependence* of the trimmed basis.

In Figure 4.2, we illustrate the problem of quasi linear dependence for a one dimensional discretization of Poisson's problem with second order Lagrange basis functions. We consider the rightmost cell, $\Omega^{\text{tr}}_i = (\hat{x}, \hat{x} + \widetilde{h}) = (0, \frac{1}{16})$, of a discretization with mesh size $h = 1$ which is trimmed by a natural boundary (*i.e.,* no boundary terms). For functions that are only supported on this cell, the bilinear form is therefore simply $\mathcal{F}(v_h, u_h) = \int_\Omega \partial_x v_h \partial_x u_h \mathrm{d}V$. While the support of basis function ϕ_1 extends beyond the trimmed element, the basis functions ϕ_2 and ϕ_3,

$$\begin{aligned} \phi_2 &= 4\left(\frac{x-\hat{x}}{h}\right) - 4\left(\frac{x-\hat{x}}{h}\right)^2, \\ \phi_3 &= -\left(\frac{x-\hat{x}}{h}\right) + 2\left(\frac{x-\hat{x}}{h}\right)^2, \end{aligned} \tag{4.5}$$

are only supported on this cell. On the untrimmed cell, these basis functions are evidently linearly independent. Hence, for $\widetilde{h} = h$ such that $\eta = 1$, the $\mathcal{F}$-angle between ϕ_2 and ϕ_3 is

$$|\cos(\theta_{\mathcal{F}})| = \frac{|\mathcal{F}(\phi_2, \phi_3)|}{\|\phi_2\|_{\mathcal{F}} \|\phi_3\|_{\mathcal{F}}} = \sqrt{4/7}, \tag{4.6}$$

which is (in absolute sense) significantly smaller than unity. However, when $\eta = \widetilde{h}/h \ll 1$, the $\mathcal{F}$-angle is

$$|\cos(\theta_{\mathcal{F}})| = \frac{|-1 + 3\eta - \frac{8}{3}\eta^2|}{\sqrt{1 - 6\eta + \frac{44}{3}\eta^2 - 16\eta^3 + \frac{64}{9}\eta^4}} \approx 1 - \frac{1}{6}\eta^2, \tag{4.7}$$

which reveals a quasi linear dependence. Similar observations can be made when considering linear combinations of these functions. In the limit of η going to zero, the local coordinate $(x-\hat{x})/h \ll 1$, which reduces both ϕ_2 and ϕ_3 to linears on Ω_i^{tr}. Figure 4.2c shows the scaled basis functions for $\eta = \frac{1}{16}$,

$$\begin{aligned} \phi_2^* &= \frac{\phi_2}{\|\phi_2\|_{\mathcal{F}}} \approx \sqrt{\frac{h}{\eta}}\left(\frac{x-\hat{x}}{h}\right) - \sqrt{\frac{h}{\eta}}\left(\frac{x-\hat{x}}{h}\right)^2, \\ \phi_3^* &= \frac{\phi_3}{\|\phi_3\|_{\mathcal{F}}} \approx -\sqrt{\frac{h}{\eta}}\left(\frac{x-\hat{x}}{h}\right) + 2\sqrt{\frac{h}{\eta}}\left(\frac{x-\hat{x}}{h}\right)^2, \end{aligned} \tag{4.8}$$

where it is used that $\|\phi_2\|_{\mathcal{F}} \approx 4\sqrt{\eta/h}$ and $\|\phi_3\|_{\mathcal{F}} \approx \sqrt{\eta/h}$ if $\eta \ll 1$. When these functions are added, the linear term cancels and a function of the form (3.19) results:

$$\phi_2^* + \phi_3^* \approx \sqrt{\frac{h}{\eta}}\left(\frac{x-\hat{x}}{h}\right)^2, \tag{4.9}$$

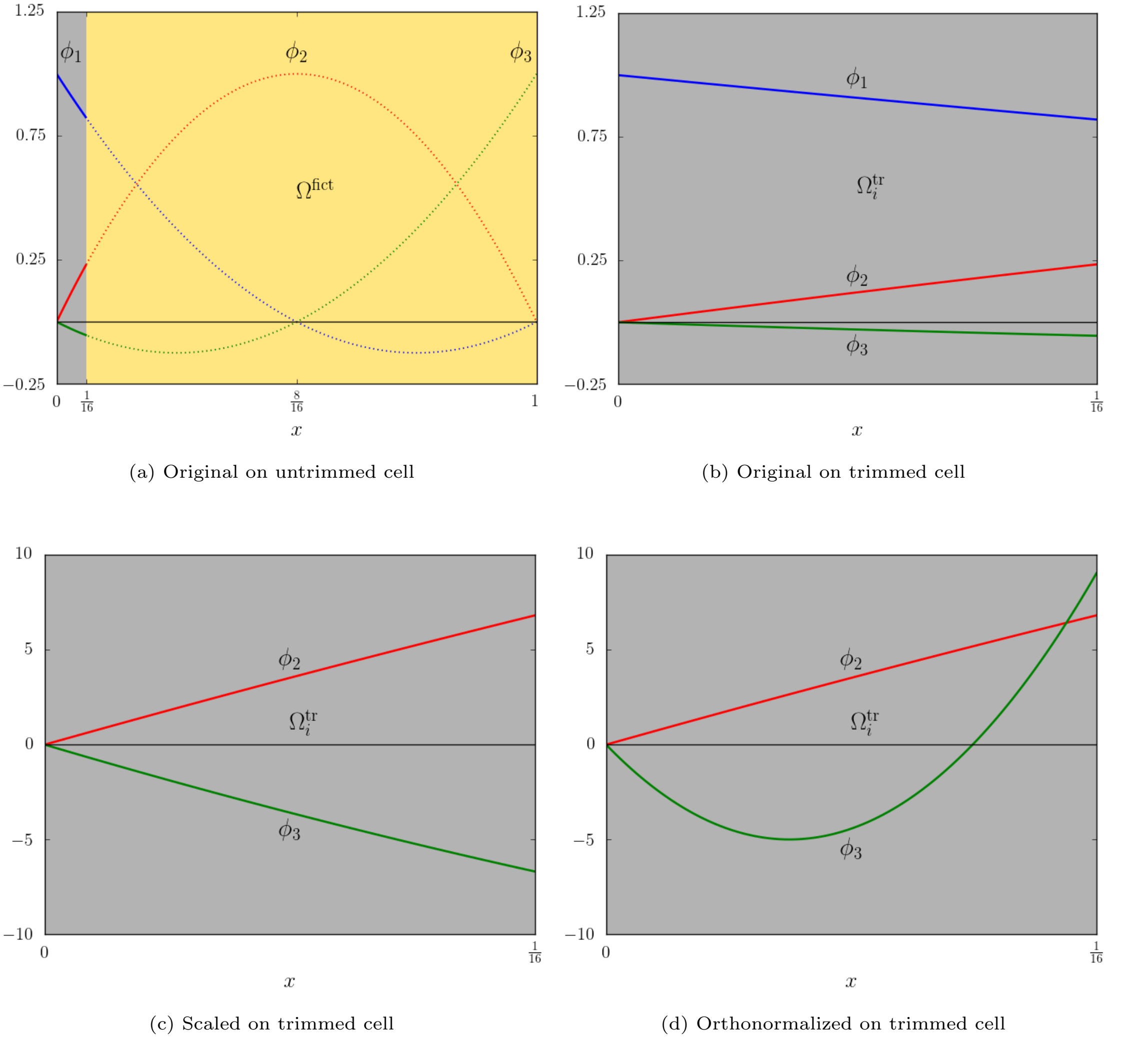
for which it can be shown that $\|\phi_2^* + \phi_3^*\|_{\mathcal{F}} \approx \frac{4}{3}\eta^2 \ll 1$. Hence, the scaled basis functions ϕ_2^* and ϕ_3^* are quasi linearly dependent for small volume fractions η. In this case, diagonal scaling is not effective in the sense that the upper bound of the minimal eigenvalue (4.1), and thereby the condition number, remains dependent on the volume fraction η, leading to ill-conditioning for small volume fractions. It is noted, however, that the scaling rate is improved with respect to that of the original system (see (3.26) and Section 4.3).

To remedy ill-conditioning due to quasi linear dependence, we propose to orthonormalize the quasi linearly dependent functions by the Gram-Schmidt procedure (see [38] for details). To illustrate how this Gram-Schmidt procedure can be used to construct a preconditioner, we again consider two quasi linearly dependent scaled functions, ϕ_α^* and ϕ_β^*. To improve conditioning, the function ϕ_β^* is made orthogonal to ϕ_α^* with respect to the finite cell inner product:

$$\phi_\beta^\perp = \phi_\beta^* - \frac{(\phi_\beta^*, \phi_\alpha^*)_{\mathcal{F}}}{\|\phi_\alpha^*\|_{\mathcal{F}}^2}\phi_\alpha^* = \phi_\beta^* - (\phi_\beta^*, \phi_\alpha^*)_{\mathcal{F}}\phi_\alpha^*. \tag{4.10}$$

In terms of the scaled basis $\mathbf{\Phi}^*$, the orthonormalization procedure can be expressed as a matrix-vector operation:

$$\mathbf{\Phi}^\perp = \mathbf{G}\mathbf{\Phi}^* = \mathbf{G}\mathbf{D}\mathbf{\Phi}, \tag{4.11}$$



(a) Original on untrimmed cell

(b) Original on trimmed cell

(c) Scaled on trimmed cell

(d) Orthonormalized on trimmed cell

Figure 4.2: Second order Lagrange basis functions over the right-most cell of the discretization with $h = 1$ of a trimmed one-dimensional domain. The volume fraction of this cell is equal to $\eta = \frac{1}{16}$.

where $\mathbf{G}$ equals the identity matrix except for the entry

$$G_{\beta\alpha} = -(\phi^*_\beta, \phi^*_\alpha)_{\mathcal{F}} = -[\mathbf{DAD}^T]_{\beta\alpha} = -\frac{A_{\beta\alpha}}{\sqrt{A_{\beta\beta}A_{\alpha\alpha}}}. \tag{4.12}$$

Since the norm of the basis functions is not preserved by the orthogonalization operation (4.11), in order to avoid improperly scaled basis functions leading to ill-conditioning, the orthogonalization procedure is followed by another normalization operation:

$$\overline{\mathbf{\Phi}} = \mathbf{D}^\perp \mathbf{G} \mathbf{\Phi}^* = \mathbf{D}^\perp \mathbf{G} \mathbf{D} \mathbf{\Phi}, \tag{4.13}$$

where $\mathbf{D}^\perp$ is the scaling matrix corresponding to $\mathbf{A}^\perp = \mathbf{G}\mathbf{A}^*\mathbf{G}^T = \mathbf{GDAD}^T\mathbf{G}^T$. The result of this orthonormalization procedure is the preconditioning matrix $\mathbf{S} = \mathbf{D}^\perp\mathbf{GD}$, which is algebraic by virtue of the fact that the three matrices $\mathbf{D}^\perp$, $\mathbf{G}$ and $\mathbf{D}$ can all be constructed on the basis of the original system matrix $\mathbf{A}$. We note that similar procedures have been applied in the context of SGFEM in order to orthonormalize the additional functions to the existing FEM approximation space [30].

With higher-order methods or in more than one dimension, it can occur that more than two functions are quasi linearly dependent, which requires multiple orthonormalization steps in the Gram-Schmidt procedure. The above concept can be applied without fundamental modifications in such cases, which is discussed in Appendix C. In the absence of boundary contributions to the system matrix, quasi linear dependence predominantly occurs between functions with the same support in the physical domain. The orthonormalization procedure outlined above does not change the support of basis functions in that case, and preserves the sparsity pattern of the system matrix. In the case of Dirichlet boundaries, it can incidentally occur that a cell is trimmed in such a way that the penalty term dominates the bilinear form for all functions supported on that cell, including functions that are also supported on other cells in the physical domain. In these cases, the Gram-Schmidt procedure may increase the support of basis functions, causing additional fill-in in the preconditioned system matrix. In our algorithm we limit this fill-in by appropriately ordering the quasi linearly dependent basis functions (see Appendix C). The effect of fill-in is studied numerically in Section 4.3.

To identify quasi linear dependencies, the $\mathcal{F}$-angle is used as a measure for the linear dependence between two basis functions and an orthonormalization threshold γ is introduced. Two functions with indices α and β are identified as being quasi linearly dependent if

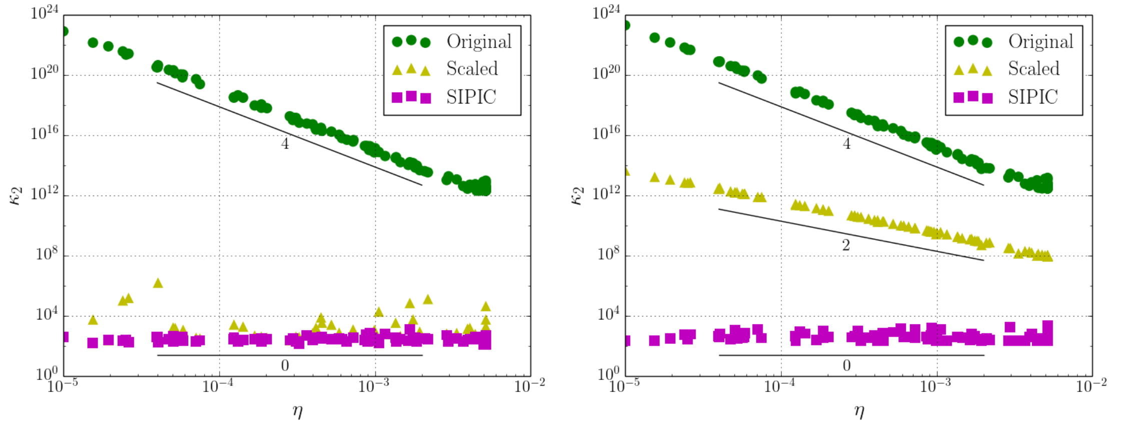
$$|\cos(\theta_{\mathcal{F}}(\phi_\alpha, \phi_\beta))| = \frac{|\mathcal{F}(\phi_\alpha, \phi_\beta)|}{\|\phi_\alpha\|_{\mathcal{F}}\|\phi_\beta\|_{\mathcal{F}}} = |\mathcal{F}(\phi^*_\alpha, \phi^*_\beta)| = |[\mathbf{DAD}^T]_{\alpha\beta}| > \gamma. \tag{4.14}$$

Therefore all quasi linear dependencies are identified by searching the off-diagonal terms of the scaled matrix $\mathbf{DAD}^T$ for entries larger (in absolute sense) than γ. A more detailed description of the identification of quasi linear dependencies is given in Appendix C. A value of $\gamma = 0.9$ has been found adequate for all our simulations. Larger values may omit some quasi linear dependencies, resulting in a less effective preconditioner. Smaller values can tag functions with different supports as being quasi linearly dependent, causing additional fill-in in the preconditioned system, as discussed in the previous paragraph and numerically studied in Section 4.3. The observed robustness and effectivity of $\gamma = 0.9$ does not necessarily extend to other (non-polynomial) bases or applications (such as XFEM or other immersed techniques) suffering from similar conditioning problems. Hence, careful selection of γ is required when applying the SIPIC preconditioner in these situations.

4.2. Preconditioner characterization and relation to alternative techniques

The preconditioner $\mathbf{S}$ developed herein approximates the inverse of the system matrix by:

$$\mathbf{SAS}^T \approx \mathbf{I} \qquad \text{or} \qquad \mathbf{S}^T\mathbf{S} \approx \mathbf{A}^{-1}. \tag{4.15}$$



(a) B-splines

(b) Lagrange

Figure 4.3: Original, scaled and SIPIC-preconditioned condition number *vs.* the smallest volume fraction for the unit square with circular exclusion with second order bases.

These identities would be exact if the Gram-Schmidt procedure would be applied to all basis functions. The developed preconditioner is incomplete in the sense that only quasi linearly dependent basis functions are orthonormalized. By construction, there exists a permutation matrix $\mathbf{P}$ which reorders the rows of $\mathbf{S}$ in such a way that a lower triangular matrix is obtained: $\mathbf{L} = \mathbf{P}\mathbf{S}\mathbf{P}^T$. Substitution in (4.15) yields

$$\mathbf{L}^T\mathbf{L} \approx \mathbf{P}\mathbf{A}^{-1}\mathbf{P}^T = [\mathbf{P}\mathbf{A}\mathbf{P}^T]^{-1} = \mathbf{C}^{-T}\mathbf{C}^{-1}, \tag{4.16}$$

where use has been made of the property $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ of permutation matrices and $\mathbf{C}$ corresponds to the Cholesky decomposition of the permuted matrix, *i.e.,* $\mathbf{C}\mathbf{C}^T = \mathbf{P}\mathbf{A}\mathbf{P}^T$. Equation (4.16) conveys that the lower triangular matrix $\mathbf{L}$ approximates the inverse Cholesky decomposition of the permuted matrix, $\mathbf{L} \approx \mathbf{C}^{-1}$, and that the preconditioner can be written as $\mathbf{S} = \mathbf{P}^T\mathbf{L}\mathbf{P} \approx \mathbf{P}^T\mathbf{C}^{-1}\mathbf{P}$. Hence this preconditioner can be interpreted as the conjugate permutation of an incomplete inverse Cholesky decomposition of a permutation of the system matrix. Note that this permutation is done to reduce fill-in as described in the last paragraph of Section 4.1.2. Since we apply $\mathbf{S}$ symmetrically (see Equation (4.2)) we refer to this technique as *Symmetric Incomplete Permuted Inverse Cholesky* (SIPIC) preconditioning.

4.3. Numerical test case continued: Poisson's problem on a unit square with circular exclusion

To test the effectivity of the SIPIC preconditioner, we again consider Poisson's problem on the rotating unit square with circular exclusion as introduced in Section 3.3. Besides the original and the SIPIC-preconditioned condition numbers, also the condition number after diagonal scaling is computed for second order ($p = 2$) B-spline and Lagrange bases. An orthonormalization threshold of $\gamma = 0.9$ is applied.

The results of all simulations are presented in Figure 4.3. It is observed that the SIPIC preconditioner drastically improves the condition number for all cases and yields condition numbers independent of the volume fraction (in the sense that there is no scaling relation). In the case of B-splines, diagonal scaling is observed to be effective for most configurations. This is explained by the fact that quasi linear dependencies in B-spline bases are uncommon, since on most of the trimmed cells there is only a single B-spline basis function whose support is restricted to that trimmed cell (see Section 4.1.1). This is in contrast to Lagrange bases, for which quasi linear dependencies are combinatorially more frequent (see Section 4.1.2). As derived for the one-dimensional example in Section 4.1.2, the condition number of the Lagrange discretizations after scaling still scales with η, but with a slope that is gentler than that of the original condition number. The difference in slope is explained by the fact that all Lagrange basis functions have non-zero first order derivatives on the boundary of their untrimmed support. The basis functions that are only supported on the cell with the smallest volume fraction therefore have a norm of order $\|\phi\|_{\mathcal{F}}^2 \sim \eta^{3-2/d} = \eta^2$. Diagonal scaling consequently increases the magnitude of these functions and the function corresponding to the smallest eigenvalue by a factor η^{-1}, which explains the observed slope difference of η^2. When a global stabilization parameter is applied, the preconditioner also drastically reduces the condition number. It is observed that – although not presented here – the preconditioned condition number still scales with $\eta^{-1/d}$ however, which is exactly the difference in slope between the original condition numbers of locally (see (3.26)) and globally (see (B.8)) stabilized systems. The origin of this effect and methods to resolve this are beyond the scope of this work.

Figure 4.4b presents the system matrix fill-in caused by large boundary contributions (see Section 4.1.2) for two orthonormalization thresholds: $\gamma_1 = 0.9$ and $\gamma_2 = 0.93$. Since no fill-in was observed for second order B-spline bases, we present results for cubic B-splines. The corresponding dependence of the condition number on the smallest volume fraction is shown in Figure 4.4a. The additional relative fill-in is defined here as the number of additional nonzero entries in the preconditioned system matrix ($\mathbf{SAS}^T$) divided by the total number of nonzero entries in the unpreconditioned matrix. For $\gamma_1 = 0.9$, it is observed that fill-in remains limited to approximately 1.5%, and that it does not scale with the smallest volume fraction η. The fill-in can be reduced further at the expense of performing fewer orthonormalization operations by increasing the orthogonalization threshold (see Section 4.1.2). This is indeed observed from the results for $\gamma_2 = 0.93$, where fill-in remains limited to approximately 0.5%. Although fewer orthonormalization operations are performed, it is observed that the effectivity of SIPIC preconditioning remains unaffected.

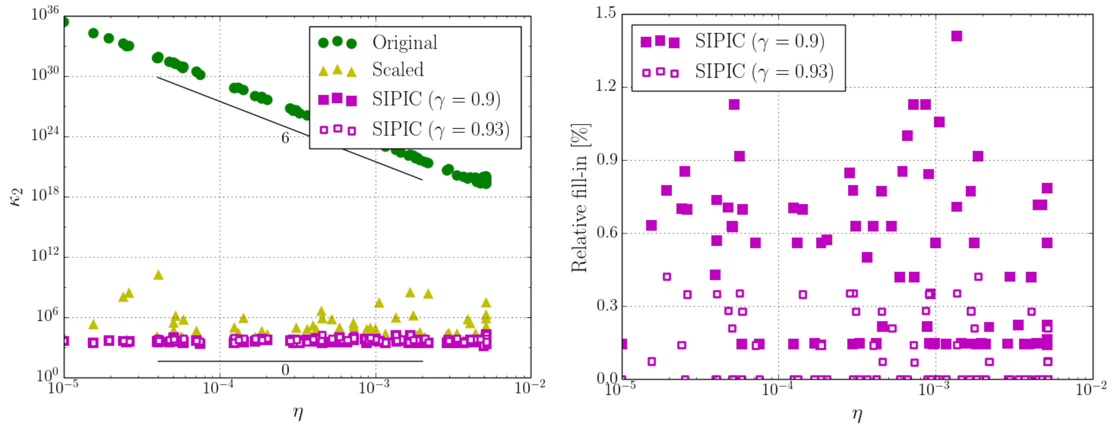
We note that the necessity to increase the threshold is debatable for two reasons:

- The fill-in is small and locally contained. An investigation of the functions whose support increases because of the preconditioner conveys that these are all functions for which the bilinear operator is governed by the penalty term on a small cell. Therefore, the effect is confined to functions that are originally supported on that small cell, and cannot spread through the system.
- The fill-in is expected to be a two-dimensional artifact. Under the assumption of mesh regularity, it can be shown that the penalty parameter β_i is of the order $\beta_i \sim \eta_i^{-1/d} h^{-1}$. Subject to the same assumption, the measure of the edges in a two-dimensional problem is of order $|\Gamma_i| \sim \eta_i^{1/d} h$, such that the total penalty term is of order $\beta_i |\Gamma_i| \sim 1$. Therefore a small cell (with a limited volumetric contribution) can still have a large penalty term. In the three-dimensional case, the size of the edge is of order $|\Gamma_i| \sim \eta_i^{2/d} h^2$, such that the total penalty term is of order $\beta_i |\Gamma_i| \sim \eta_i^{1/d} h$. Under the assumption of shape regularity, small cells will therefore have small penalty terms as well, if the number of dimensions is larger than two. As a result, it is unlikely that the penalty term on a cell with a small volume fraction dominates the bilinear form for functions that are also supported on other cells in more than two dimensions.

A detailed study of the sensitivity of the SIPIC preconditioning technique to the orthonormalization threshold is a topic of further study.

4.4. *Comparison with alternative preconditioning techniques*

To assess the performance of the SIPIC preconditioner for the finite cell method we have compared it with a selection of commonly used preconditioning techniques, *viz.* Jacobi preconditioning, incomplete Cholesky preconditioning, and LU-preconditioning. Note that the considered symmetrically-applied Jacobi preconditioner is equivalent to diagonal scaling, which is an intrinsic part of the SIPIC preconditioner (see Section 4.1.1). Our performance comparison on one hand focuses on the effectivity of the preconditioner, which we assess on the basis of the improvement of the condition number of the system (which is again computed using a power algorithm). On the other hand we assess the computational burden of the considered preconditioners based on



(a) Condition number

(b) Relative additional fill-in

Figure 4.4: Condition number and relative matrix fill-in *vs.* the smallest volume fraction for the unit square with circular exclusion with cubic B-splines.

the relative fill-in of the preconditioning matrix (or matrices in the case of LU). The relative fill-in of the preconditioning matrix is defined here as the number of off-diagonal entries in the preconditioner divided by the total number of nonzero entries in the system matrix. Note that this definition of the relative fill-in of the preconditioning matrix differs from the definition used for the relative additional fill-in in Figure 4.4b. For the SIPIC preconditioner a direct relation exists between the computational costs in terms of floating point operations and this fill-in definition, in the sense that the number of floating point operations needed in addition to setting up the Jacobi preconditioner scales with the fill-in. In practice this implies that – provided that the fill-in remains small – the computational overhead of the SIPIC preconditioner is comparable to that of a Jacobi preconditioner, which can be considered the most favorable from the computational complexity point of view.

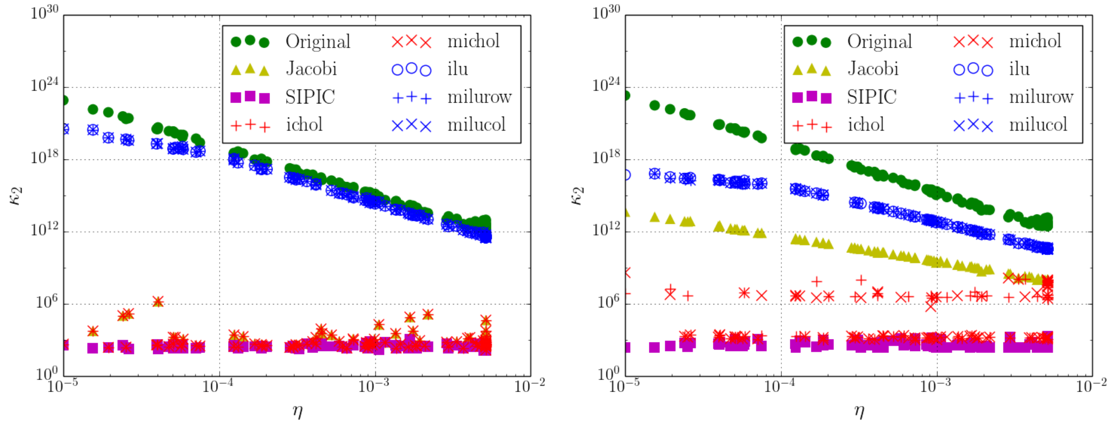
For our comparison we again consider the test case introduced in Section 3.3, for which the application of SIPIC was considered in Section 4.3. The results of this comparison are presented in the Figures 4.5 and 4.6. All presented results are based on the Matlab (2011b) implementation of the commonly used preconditioners. More specifically, we consider:

- An *incomplete Cholesky* (`ichol`) approximation with the same sparsity pattern as the system matrix (`type='nofill'`) and without additional row and column manipulation (`michol='off'`). Moreover, the diagonal of the system matrix that is fed to the algorithm is multiplied with the maximum of the row sum of the absolute values of the off-diagonal terms divided by the diagonal term to ensure diagonal dominance (`diagcomp=max(sum(abs(A),2)./diag(A))-2`). Without this optional argument, the incomplete Cholesky preconditioner fails due to the fact that the system matrix is only very slightly positive definite;

- A *Modified incomplete Cholesky* (`ichol` with the option `michol='on'`) approximation, similar to the aforementioned incomplete Cholesky preconditioner, but with the diagonal of the Cholesky factorization manipulated such that the row and column sum of the preconditioner are equal to the row and column sum of the system matrix;

- An *incomplete LU* (`ilu`) preconditioner with the same sparsity pattern as the system matrix (`type='nofill'`) and without additional row and column manipulation (`milu='off'`). Note that since this preconditioner is not symmetric, the power algorithm used to compute the condition number is applied to the matrix $\left(\mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\right)^T\left(\mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\right)$, the condition number of which is equal to the square of that of the preconditioned system matrix;

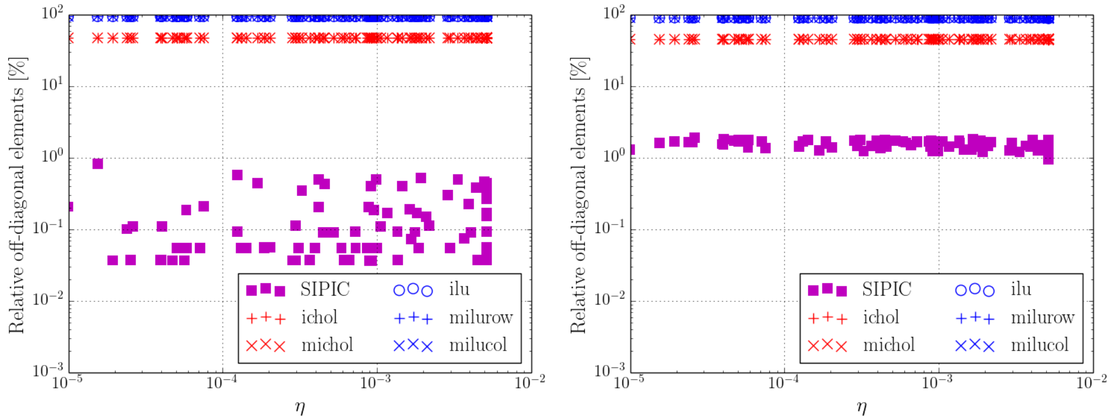
- A *row/column-modified incomplete LU* (`ilu` with the option `milu='row'` or `milu='column'`), which is similar to the standard incomplete LU-preconditioner, but with the diagonal of $\mathbf{U}$ manipulated such that the row/column sum of the preconditioner is equal to the row/column sum of the system matrix.



(a) B-splines

(b) Lagrange

Figure 4.5: Condition numbers *vs.* the smallest volume fraction for the unit square with circular exclusion with second order bases.



(a) B-splines

(b) Lagrange

Figure 4.6: Relative fill-in *vs.* the smallest volume fraction for the unit square with circular exclusion with second order bases.

It is noted that the employed Matlab preconditioner implementations also provide the possibility for setting a threshold drop tolerance. This setting is, however, not practical, since to avoid breakdown of the algorithm due to ill-conditioning of the system matrix, the threshold drop tolerance has to be set so low that the resulting system loses its sparsity. This problem could be ameliorated by restoring the diagonal dominance of the system matrix as discussed above for the incomplete Cholesky preconditioner. However, this approach is not effective, as it will neglect small off-diagonal elements, which are essential for improving the system conditioning since these (in combination with diagonal elements of the same magnitude) cause quasi linear dependencies.

From the condition numbers presented in Figure 4.5, it is observed that incomplete LU-hardly improves the condition number and that incomplete Cholesky performs better than Jacobi, but does not manage to resolve the quasi linear dependencies in all cases. Overall it is observed that SIPIC is the most effective preconditioner in all considered cases, where it should be noted that (modified) incomplete Cholesky preconditioning is also very competitive in terms of its effectivity for a significant number of test cases. The relative fill-in results are presented in Figure 4.6. By construction, the relative fill-in of the incomplete LU-factorizations is approximately one (this is the sum of the lower and upper diagonal matrices) and the relative fill-in of the incomplete Cholesky preconditioners is approximately a half. The SIPIC preconditioner is observed to be considerably sparser, with a relative fill-in of at most a few percent.

In summary, it is observed that the SIPIC preconditioner is considerably more effective in terms of improving the condition number than Jacobi and incomplete LU-preconditioning. While incomplete Cholesky preconditioning can compete with SIPIC in terms of effectivity, these preconditioners lead to considerably larger fill-in ratios. An additional advantage of SIPIC compared to the other preconditioners is that it is an approximation of the inverse of the system matrix instead of an approximation of the system matrix itself. As a result, with SIPIC preconditioning, the preconditioned system can be computed explicitly and has a minimal additional fill-in, such that it does not induce extra cost to an iterative step in the solver. In contrast, the other preconditioning techniques, besides Jacobi, approximate the system matrix itself with lower and upper diagonal matrices. Accordingly, these preconditioning techniques require the solution of a lower diagonal and an upper diagonal system in every iterative step.

5. Numerical simulations: SIPIC preconditioned finite cell analysis

The numerical simulations presented in the previous sections have demonstrated the effectivity of the SIPIC preconditioning technique for Poisson's problem as a model problem. To demonstrate that the condition number estimate and preconditioning solution are applicable to the broader class of symmetric and coercive unfitted Galerkin formulations, the test cases in this section involve problems of linear elasticity. While the simulations in Sections 3.3, 4.3 and 4.4 had a strong focus on studying the condition number, in this section we will demonstrate that SIPIC preconditioning also enables robust finite cell analyses using iterative solvers.

Since the considered finite cell systems are symmetric positive definite, a Conjugate Gradient (CG) solver is used. As discussed in Section 3, preconditioning of the finite cell system results in faster convergence of the solver (see (3.3)). Moreover, for well-conditioned systems the residual provides an adequate measure for the error in the energy norm (see (3.4)). As a result, for the same CG tolerance, the energy norm error of a well-conditioned system is generally expected to be smaller than that of an ill-conditioned system. These two beneficial effects of preconditioning enable the computation of high-accuracy finite cell approximations.

For the simulations presented in this section, the solver performance and accuracy aspects of finite cell preconditioning are discussed in the context of mesh convergence studies. Section 5.1 considers the elastic analysis of a square with a circular hole. Since condition numbers can be computed for this two-dimensional test case, it is also used to demonstrate that the results presented in the previous sections extend to linear elasticity. Furthermore, this section contains a further comparison of the SIPIC preconditioner to the other preconditioners introduced in Section 4.4, regarding both the effect on the condition number and the solver performance. Section 5.2 studies the performance of SIPIC preconditioning for a three-dimensional linear elasticity problem.

5.1. Square with a circular hole

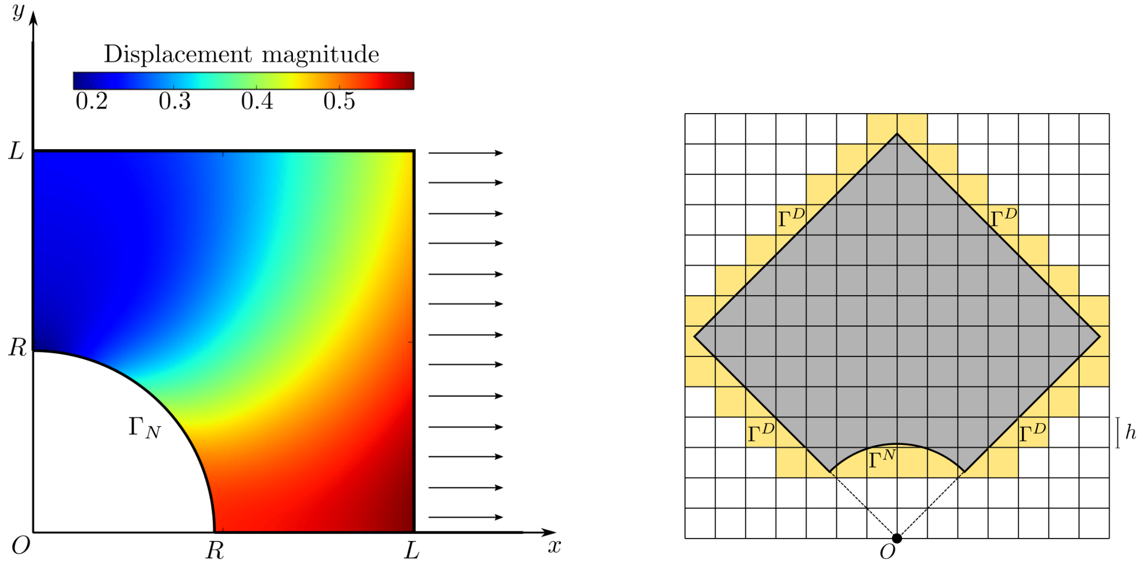
We consider the linear elastic analysis of a uniaxially-loaded infinite domain with a circular hole with radius $R = 3/(2\pi)$ (Figure 5.1a). Under plane strain conditions, the displacement field $u = (u_x, u_y)$ corresponding to a unit horizontal traction loading is given by

$$u_x = \frac{x}{\mu}\left(\frac{(2\mu+\lambda)r^2 + (\mu-\lambda)R^2}{4(\mu+\lambda)r^2} + \frac{\frac{3}{4}R^4 + x^2R^2}{r^4} - \frac{x^2R^4}{r^6}\right), \tag{5.1a}$$

$$u_y = \frac{y}{\mu}\left(\frac{-\lambda r^2 + (\mu+3\lambda)R^2}{4(\mu+\lambda)r^2} - \frac{\frac{3}{4}R^4 + y^2R^2}{r^4} + \frac{y^2R^4}{r^6}\right), \tag{5.1b}$$

with $r = \sqrt{x^2+y^2}$ and Lamé parameters $\lambda = \mu = 1$. This exact solution is reproduced on a truncated domain by considering the strong formulation

$$\begin{cases} -\mathrm{div}(\boldsymbol{\sigma}(u)) = 0 & \text{in } \Omega, \\ \boldsymbol{\sigma}(u)\cdot n = g^N & \text{on } \Gamma^N, \\ u = g^D & \text{on } \Gamma^D, \end{cases} \tag{5.2}$$



(a) Analytical solution

(b) Mesh orientation

Figure 5.1: A square with a circular hole is rotated over 45° with respect to a Cartesian mesh. The Neumann and Dirichlet data on Γ^N and Γ^D, respectively, match the exact solution of the infinite domain problem.

with boundary data g^D and g^N set in accordance with the analytical solution of the infinite domain problem (5.1). The Cauchy stress is related to the displacement field by Hooke's Law, $\boldsymbol{\sigma}(u) = \lambda \mathrm{div}(u)\mathbf{I} + 2\mu\nabla^s u$, with ∇^s the symmetric gradient operator.

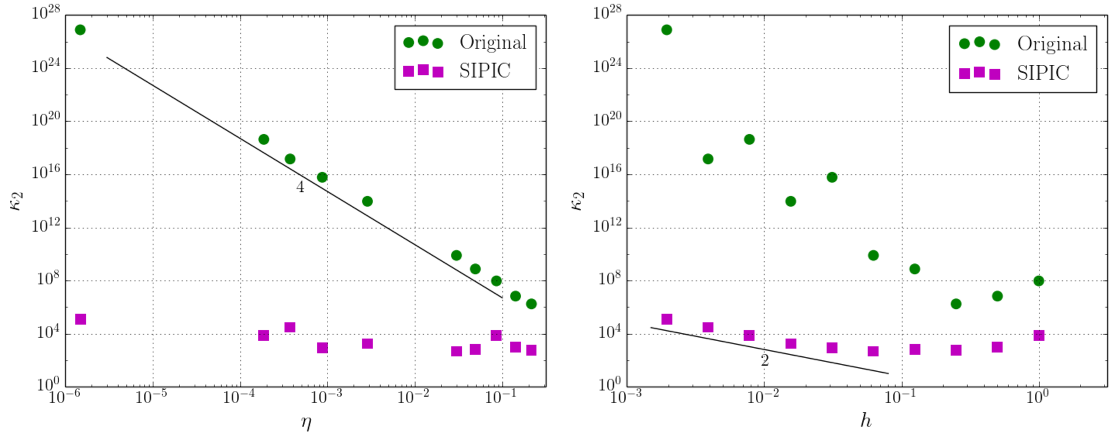
In view of symmetry, only a square of length $L = 1$ is considered (Figure 5.1a). This square is rotated over an angle of 45° with respect to the Cartesian mesh with size h in which it is immersed (Figure 5.1b). Quadratic B-spline spaces are constructed over a sequence of ten uniformly refined meshes, starting with cells of size $h = 1$. The number of elements that intersect the domain ranges from 4 on the coarsest mesh to 216,672 on the finest mesh. Similar to the test case in Section 3 and 4, the exact geometry of the domain is again approximated by a bisection-based tessellation scheme on which exact integration is performed (see [12] for details). To preserve the geometry parametrization provided by the bisection-based tessellation scheme, the maximal recursion depth is reduced with each mesh refinement. On the finest mesh (nine refinements) trimmed cells are directly triangulated for integration purposes. Note that exact geometric representation of the domain would yield homogeneous Neumann data: $g^N = 0$. However, since the employed tessellation is only an approximation of the exact geometry, a (small) traction is applied along the circular boundary in accordance with (5.1) and the approximate geometry.

In contrast to the weak form introduced in Section 2, the weak form for (5.2) contains two separate stabilization parameters following the approach in [9]:

$$\begin{aligned}\mathcal{F}(v_h, u_h) = &\int_\Omega \nabla^s v_h : \boldsymbol{\sigma}(u_h)\mathrm{d}V \\ &+ \int_{\Gamma^D} -\big(v_h \cdot \boldsymbol{\sigma}(u_h)\cdot n + u_h \cdot \boldsymbol{\sigma}(v_h)\cdot n\big)\mathrm{d}S \\ &+ \int_{\Gamma^D} \big(\beta^\lambda v_h \cdot (n \otimes n)\cdot u_h + \beta^\mu v_h \cdot u_h\big)\mathrm{d}S, \end{aligned} \tag{5.3a}$$

$$\begin{aligned}\ell(v_h) = &\int_{\Gamma^N} v_h \cdot g^N \mathrm{d}S + \int_{\Gamma^D} -g^D \cdot \boldsymbol{\sigma}(v_h)\cdot n\mathrm{d}S \\ &+ \int_{\Gamma^D} \big(\beta^\lambda v_h \cdot (n\otimes n)\cdot g^D + \beta^\mu v_h \cdot g^D\big)\mathrm{d}S, \end{aligned} \tag{5.3b}$$

with separate local stabilization parameters $\beta^\lambda = 2\lambda C^\lambda > \lambda C^\lambda$ and $\beta^\mu = 4\mu C^\mu > 2\mu C^\mu$ for C^λ



(a) *vs.* the smallest volume fraction

(b) *vs.* the mesh size

Figure 5.2: Original and SIPIC-preconditioned condition numbers for the square with a circular hole with second order B-spline bases.

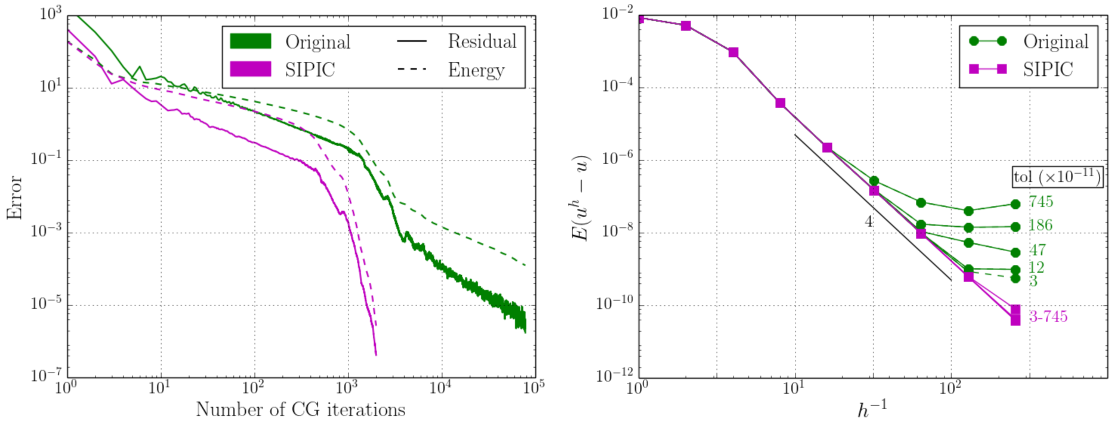
and C^μ according to

$$
\begin{aligned}
C_i^\lambda &= \max_{v_h \in \mathcal{V}_h(\Omega)} \frac{\int_{\Gamma_i^D} \operatorname{div}(v_h)^2 \mathrm{d}S}{\int_{\Omega_i^{\mathrm{tr}}} \operatorname{div}(v_h)^2 \mathrm{d}V}, \\
C_i^\mu &= \max_{v_h \in \mathcal{V}_h(\Omega)} \frac{\int_{\Gamma_i^D} \|\nabla^s v_h \cdot n\|_e^2 \mathrm{d}S}{\int_{\Omega_i^{\mathrm{tr}}} \|\nabla^s v_h\|_e^2 \mathrm{d}V}.
\end{aligned}
\tag{5.4}
$$

Figure 5.2 shows the original and SIPIC-preconditioned condition numbers versus the smallest volume fraction (Figure 5.2a) and the mesh size (Figure 5.2b). The scaling relation (3.26) for the original system is clearly observed from Figure 5.2a, despite the fact that the mesh size h is varied. One can observe that the SIPIC-preconditioned condition number is essentially independent of the minimal volume fraction. Both observations are in agreement with the numerical results in Section 4.3. Figure 5.2b shows the same data versus the mesh size h. The observed relation between the original condition number and the mesh size can be conceived of as an induced effect, due to a correlation between the mesh size h and smallest volume fraction η, *viz.,* in general smaller minimum volume fractions can be expected when the number of trimmed cells is increased (by h-refinement). An important observation from Figure 5.2b is that SIPIC preconditioning results in a scaling relation $\kappa_2(\mathbf{SAS}^T) \propto h^{-2}$, which resembles that of standard finite elements [37].

The performance of the SIPIC preconditioner is studied further in Figure 5.3. Figure 5.3a displays the convergence of the CG solver for the finest mesh (438,756 degrees of freedom) with and without SIPIC preconditioning. The original condition number $\kappa_2(\mathbf{A}) \approx 10^{27}$, whereas the SIPIC-preconditioned condition number equals $\kappa_2(\mathbf{SAS}^T s) \approx 10^5$. Both the residual ($\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|_2$, solid lines) and the error in the energy norm ($\|\mathbf{x} - \mathbf{x}_i\|_\mathbf{A}$, dashed lines) are shown. As expected from Equation (3.3), it is indeed observed that the SIPIC-preconditioned system converges significantly faster than the original system. For a CG solver tolerance of 10^{-6}, the preconditioned system requires approximately 50 times fewer iterations. Furthermore, following Equation (3.4), it is observed that the correlation between the residual and the energy norm error is stronger in the preconditioned system. For the same tolerance, the energy norm error in the original system is substantially larger than the energy norm error in the preconditioned system.

An important effect of the stricter correlation between the residual and the energy norm error for preconditioned systems is visible in Figure 5.3b. In this figure, we study the mesh convergence by plotting the strain energy of the error, $E(u - u_h) = \frac{1}{2}\nabla^s(u - u_h) : \boldsymbol{\sigma}(u - u_h)$, against the mesh size for various CG solver tolerances (ranging from 3×10^{-11} to 745×10^{-11}). The CG solver terminates when either the relative residual ($\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|_2 / \|\mathbf{b}\|_2$) or the absolute residual ($\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|_2$) reaches the specified tolerance, or when the number of CG iterations exceeds 100,000. One can



(a) CG iterations

(b) Mesh convergence

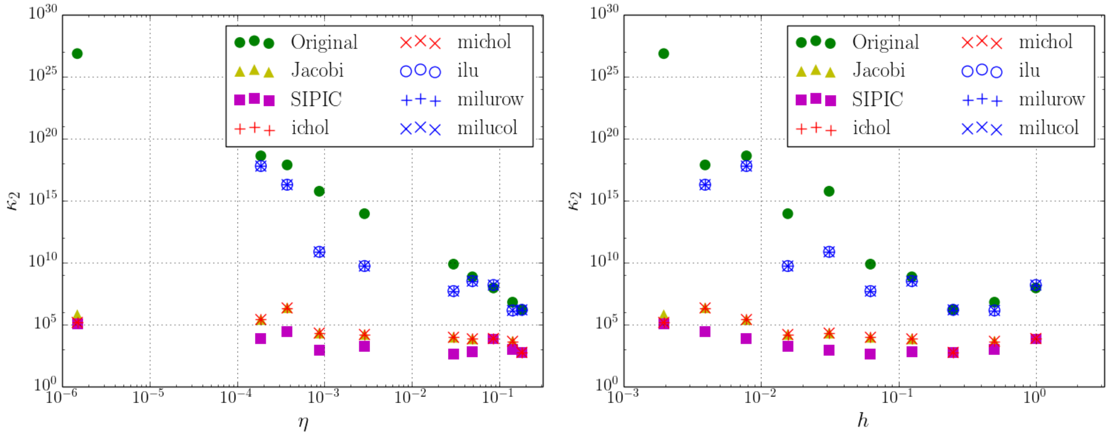
Figure 5.3: Original and SIPIC-preconditioned finite cell results for the square with a circular hole with second order B-spline bases: (a) Convergence of the residual and energy norm error *vs.* the CG iterations on the finest mesh; (b) Convergence of the strain energy error under mesh refinement for various CG solver tolerances.

observe that for the original system, the error due to not solving the linear system with sufficient accuracy becomes dominant when the mesh is refined. For these ill-conditioned systems, the precision with which the linear system can be solved hinders mesh convergence of the strain energy error under mesh refinement. Improving the accuracy of the original system's solution by lowering the CG tolerance is not a practical solution to this problem, since the number of iterations required to reach convergence of the CG solver (and thereby the computational effort) increases dramatically. For the the finest mesh with the smallest considered tolerance, a converged result was not obtained within 100,000 CG iterations (indicated by the dashed line in Figure 5.3b). The SIPIC preconditioner improves the quality of the solution measured in the strain energy of the error, and asymptotic convergence under mesh refinement is observed even for relatively large CG solver tolerances. The observed rate of mesh convergence of the strain energy error of h^4 resembles the optimal rate for standard finite elements of order $p = 2$.

In Figure 5.4 the effectivity of the SIPIC preconditioner is compared with the commonly used preconditioners discussed in Section 4.4. It is observed that the condition number with incomplete LU preconditioning remains strongly dependent on the volume fraction. The incomplete Cholesky and Jacobi preconditioning perform considerably better and yield virtually identical results. For all considered cases SIPIC yields the lowest condition number. On some of the considered meshes Jacobi and incomplete Cholesky preconditioning give similar results, by virtue of the fact that the condition number on these meshes is only weakly affected by quasi linear dependencies.

Figure 5.5 shows the relative number of off-diagonal elements in the preconditioner (a) and the required number of iterations to obtain a solution with an energy norm error smaller than 10^{-7} (b). Again, we define the relative fill-in as the number of off-diagonal entries in the preconditioner divided by the total number of nonzero entries in the system matrix. Similar to the results in Section 4.4 the relative fill-in of the incomplete LU factorizations (for both factors together) is approximately one and the relative fill-in of the incomplete Cholesky factorizations is approximately a half. The SIPIC preconditioner is considerably sparser and the relative number of off-diagonal elements is observed to be approximately proportional to h. This can be explained by the fact that quasi linear dependencies can only occur amongst trimmed functions, and the relative number of functions whose support is intersected by the boundary scales with h. Jacobi preconditioning is not displayed in this figure, as it does not have any off-diagonal elements.

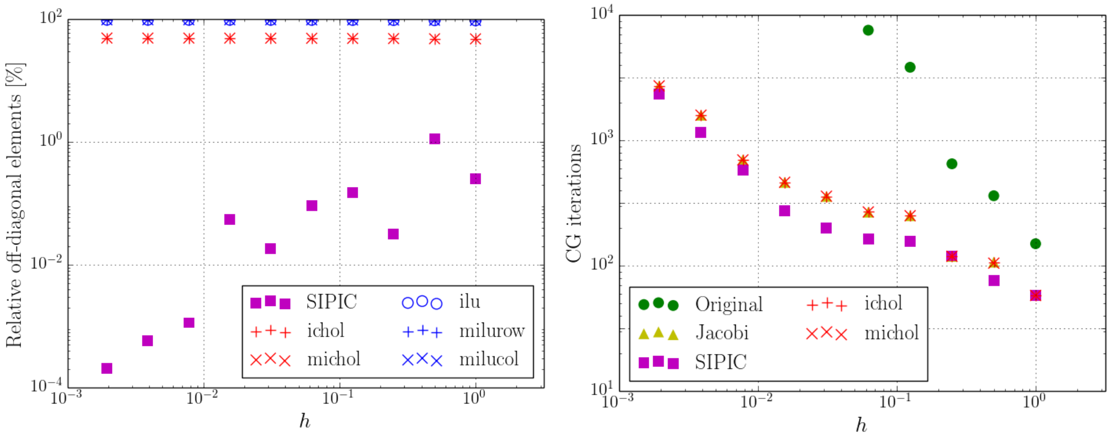
Figure 5.5b conveys that all considered preconditioning techniques effectively improve the convergence behavior, in the sense that the required number of iterations to reduce the energy norm error below 10^{-7} is significantly smaller than for the unpreconditioned system. Again the



(a) *vs.* the smallest volume fraction

(b) *vs.* the mesh size

Figure 5.4: Original and preconditioned condition numbers for various preconditioners for the square with a circular hole with second order B-spline bases.



(a) Relative number of off-diagonal elements.

(b) Required number of iterations

Figure 5.5: Required number of iterations and fill-in of various preconditioners against the mesh size for the square with circular exclusion with second order bases.

results for the incomplete Cholesky factorizations and the Jacobi preconditioner are virtually identical. The SIPIC preconditioner converges slightly faster, as is expected from the condition numbers in Figure 5.4. It should be noted that, even with the same number of required iterations, Jacobi preconditioning and SIPIC are computationally cheaper than incomplete Cholesky. This is because the latter approximates the system matrix itself, while Jacobi and SIPIC approximate the inverse (see Section 4.4). Note that incomplete LU preconditioning does not give a symmetric positive definite system and, accordingly, the conjugate gradient method cannot be applied directly.

5.2. Quarter solid torus

We consider the linear elastic analysis of a quarter of a three-dimensional solid torus (Figure 5.6). The radius of the torus, measured from its axis of revolution (z-axis) to the center of its circular cross-sectional area, is equal to $R = 2$. The radius of the cross-section is equal to $r = \frac{1}{2}\sqrt{2}$. On the bottom boundary (Γ_n^D, $y = 0$) only the normal (y-)component of the displacement is constrained. On the left boundary (Γ_t^D, $x = 0$) the downward (y-)displacement is prescribed, while the displacement in the z-direction is constrained. This boundary can move freely in the normal (x-)direction.

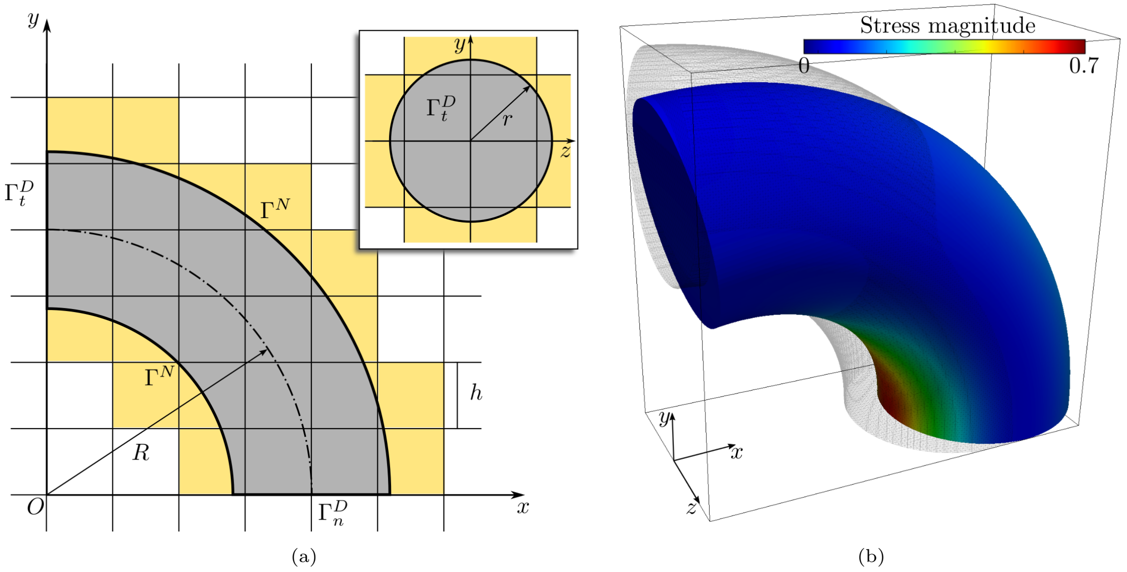


Figure 5.6: Problem setup for a quarter of a solid torus with radius of revolution R and cross-sectional radius r. The flat ends of the quarter torus coincide with the mesh. The stress magnitude is defined as the Euclidean norm of the Cauchy stress tensor, and the displacements in (b) are scaled.

The strong formulation is given by

$$\begin{cases} -\mathrm{div}(\boldsymbol{\sigma}) = 0 & \text{in } \Omega, \\ u \cdot n = 0 \quad \text{and} \quad \mathcal{P}\boldsymbol{\sigma} \cdot n = 0 & \text{on } \Gamma_n^D, \\ \mathcal{P}u = \mathcal{P}g^D \quad \text{and} \quad n \cdot \boldsymbol{\sigma} \cdot n = 0 & \text{on } \Gamma_t^D, \\ \boldsymbol{\sigma} \cdot n = 0 & \text{on } \partial\Omega \setminus (\Gamma_n^D \cup \Gamma_t^D), \end{cases} \tag{5.5}$$

with projection operator $\mathcal{P}$, $\mathcal{P}\xi = \xi - (n \otimes n) \cdot \xi$, and Dirichlet data $g^D = (0, 0, -1)$. Note that this condition is only applied on the tangential (left) boundary Γ_t^D, as on the normal (bottom) boundary Γ_n^D homogeneous Dirichlet conditions are applied. The Cauchy stress tensor is related to the displacement field by Hooke's law with Lamé parameters $\lambda = \mu = 1$. Note that this problem is underconstrained as it permits for a rigid body translation in the x-direction and an infinitesimal rigid body rotation around the y-axis. Herein, these rigid body modes are automatically accounted for by the employed CG solver. Also, the analysis only considers the strain energy, which is independent of the rigid body modes.

The solid torus is immersed in a Cartesian mesh with size h that is aligned with the coordinate system in Figure 5.6a. Quadratic B-spline spaces are considered over a sequence of six uniformly refined meshes, starting with a mesh of size $h = 1$. Again the domain is approximated and integration is performed according to a tessellation scheme [12]. The recursion depth for the tessellation scheme is equal to five on the coarsest mesh, and is decreased upon mesh refinement to preserve the geometry parametrization. The number of elements that intersect the computational domain ranges from 16 on the coarsest mesh to 168,928 on the finest mesh.

The operators for the finite dimensional weak formulation (2.3) corresponding to problem (5.5) – with the Dirichlet boundary conditions imposed by Nitsche's method – are given by

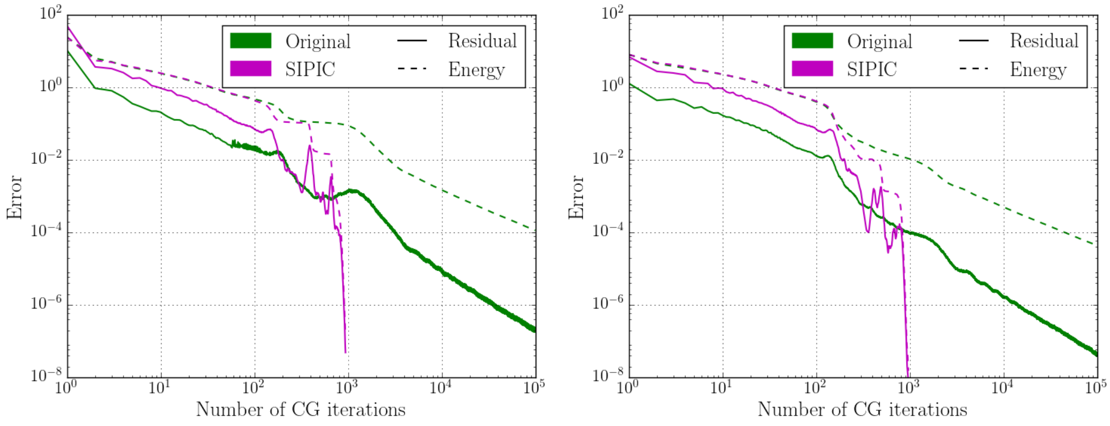
$$\begin{aligned}
\mathcal{F}(v_h, u_h) = &\int_{\Omega} \nabla^s v_h : \boldsymbol{\sigma}(u_h) \mathrm{d}V \\
&+ \int_{\Gamma_n^D} -\big(v_h \cdot (n \otimes n) \cdot \boldsymbol{\sigma}(u_h) \cdot n + u_h \cdot (n \otimes n) \cdot \boldsymbol{\sigma}(v_h) \cdot n\big) \mathrm{d}S \\
&+ \int_{\Gamma_n^D} (\beta^\lambda + \beta_n^\mu) v_h \cdot (n \otimes n) \cdot u_h \mathrm{d}S \\
&+ \int_{\Gamma_t^D} -\big(v_h \cdot \mathcal{P}\boldsymbol{\sigma}(u_h) \cdot n + u_h \cdot \mathcal{P}\boldsymbol{\sigma}(v_h) \cdot n\big) \mathrm{d}S \\
&+ \int_{\Gamma_t^D} \beta_t^\mu v_h \cdot \mathcal{P} u_h \mathrm{d}S, \qquad (5.6a) \\
\ell(v_h) = &\int_{\Gamma_t^D} -g^D \cdot \mathcal{P}\boldsymbol{\sigma}(v_h) \cdot n \mathrm{d}S \\
&+ \int_{\Gamma_t^D} \beta_t^\mu v_h \cdot \mathcal{P} g^D \mathrm{d}S. \qquad (5.6b)
\end{aligned}$$

The local stabilization parameters are taken as $\beta^\lambda = 2\lambda C^\lambda > \lambda C^\lambda$, $\beta_n^\mu = 4\mu C_n^\mu > 2\mu C_n^\mu$, and $\beta_t^\mu = 4\mu C_t^\mu > 2\mu C_t^\mu$. Provided that each trimmed cell intersects with at most one type of Dirichlet boundary, the constants C^λ, C_n^μ, and C_t^μ satisfy:

$$\begin{aligned}
C_i^\lambda &= \max_{v_h \in \mathcal{V}_h(\Omega)} \frac{\int_{\Gamma_i^D} \mathrm{div}(v_h)^2 \mathrm{d}S}{\int_{\Omega_i^{\mathrm{tr}}} \mathrm{div}(v_h)^2 \mathrm{d}V}, \\
C_{n,i}^\mu &= \max_{v_h \in \mathcal{V}_h(\Omega)} \frac{\int_{\Gamma_i^D} (n \cdot \nabla^s v_h \cdot n)^2 \mathrm{d}S}{\int_{\Omega_i^{\mathrm{tr}}} \|\nabla^s v_h\|_e^2 \mathrm{d}V}, \qquad (5.7) \\
C_{t,i}^\mu &= \max_{v_h \in \mathcal{V}_h(\Omega)} \frac{\int_{\Gamma_i^D} \|\mathcal{P}\nabla^s v_h \cdot n\|_e^2 \mathrm{d}S}{\int_{\Omega_i^{\mathrm{tr}}} \|\nabla^s v_h\|_e^2 \mathrm{d}V}.
\end{aligned}$$

The weak imposition of Dirichlet constraints in normal or tangential direction by means of the above operators has, to the best of our knowledge, not been reported in literature. Since the Dirichlet boundaries are aligned with the Cartesian mesh, the Dirichlet constraints can also be imposed strongly by using open knot vectors in the parametric directions corresponding to the x- and y-axes. This enables us to compare weak imposition of boundary conditions using Nitsche's method to (traditional) strong imposition of boundary conditions.

On the finest mesh, the number of degrees of freedom is equal to 592,680, which makes the computation of condition numbers impractical. Therefore, we will focus on the performance of the preconditioned iterative solver and the effect of SIPIC preconditioning on the mesh convergence behavior. Figure 5.7 displays the CG solver convergence on the finest mesh, for both impositions of boundary conditions. One can observe that the convergence of the CG method is very similar for the formulations with weakly and strongly imposed boundary conditions, for both the original and the SIPIC-preconditioned system. However, a significant difference between the convergence behavior of CG for the original and the preconditioned system is observed. SIPIC preconditioning reduces the number of CG iterations to reach a residual of 10^{-7} by more than a factor of 100 relative to the original system. Moreover, SIPIC preconditioning improves the correlation between the residual ($\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|_2$) and the energy norm error ($\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}}$). The effect of the stricter correlation between the residual and the energy norm error for preconditioned systems is visible in Figure 5.8, which studies mesh convergence of the error in the strain energy using a solver tolerance of $9.5 \cdot 10^{-7}$. The error in the strain energy is defined as the absolute difference in strain energy compared to an overkill solution computed on the finest mesh with cubic B-splines (*i.e.*, $|E(u_{\mathrm{overkill}}) - E(u_h)|$ with



(a) Nitsche BCs

(b) Strong BCs

Figure 5.7: Convergence of the residual and energy norm error *vs.* the solver iterations for the original and SIPIC-preconditioned finite cell results of the quarter solid torus problem with second order B-spline bases and a mesh size of $h = 0.03125$ (592,680 degrees of freedom). Boundary conditions are (a) imposed using Nitsche's method, and (b) strongly imposed through the approximation space.

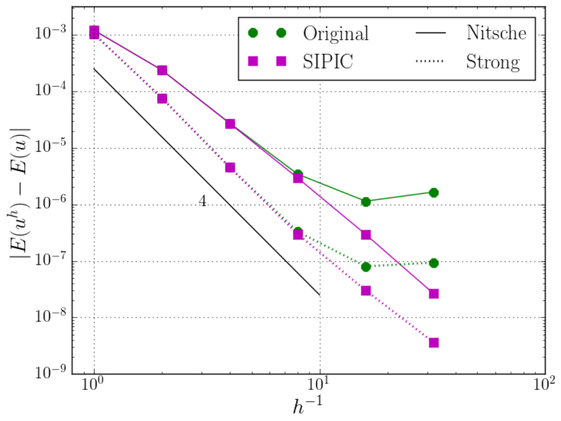


Figure 5.8: Convergence of the strain energy under mesh refinement for the original and SIPIC-preconditioned finite cell results for the quarter solid torus problem with second order B-spline bases. A CG solver tolerance of $9.5 \cdot 10^{-7}$ is used.

$E(u) = \frac{1}{2}\nabla^s u : \boldsymbol{\sigma}(u))$. Figure 5.8 clearly illustrates that ill-conditioning of the original system impedes asymptotic mesh convergence and hinders the computation of high-accuracy solutions. Using SIPIC preconditioning, optimal asymptotic mesh convergence behavior is observed. The rates of mesh convergence for both weakly and strongly imposed Dirichlet boundary conditions closely resemble the optimal rate of h^4 for standard finite elements, despite the fact that the mesh convergence rate can be affected by *e.g.,* the mesh dependent bilinear forms considered herein, or by a lack of accuracy in the overkill solution.

6. Concluding remarks

In this work, we have rigorously established a scaling relation for the condition number of second order elliptic finite cell systems. This scaling relation reveals that the condition number is inversely proportional to the smallest trimmed cell volume fraction, a result which confirms observations reported in the literature. Since this inverse proportionality scales with the power of $2p$, with p the order of the considered basis functions, ill-conditioning problems become particularly apparent when higher-order methods (*e.g.,* IGA or p-FEM) are considered. The obtained scaling relation is valid for locally stabilized systems, independent of the type of boundary conditions. For globally stabilized systems, an even stronger dependence of the condition number on the volume fraction is observed.

We have developed the SIPIC (*Symmetric Incomplete Permuted Inverse Cholesky*) preconditioning technique, which effectively improves the conditioning of finite cell systems. This improved conditioning has been observed directly in condition number computations and indirectly in the iterative solver performance. On one hand, the preconditioner applies diagonal scaling to the applied basis to avoid ill-conditioning due to small (in the finite cell norm) basis functions. On the other hand, ill-conditioning due to quasi linearly dependent basis functions is mitigated by the application of local Gram-Schmidt orthonormalization. The preconditioner is constructed at negligible computational cost and is solely based on information from the finite cell system matrix, which makes it algebraic. By virtue of this algebraic nature, it is straightforward to implement as it is non-intrusive in the rest of the numerical code. Applying the preconditioner does not yield significant fill-in to the finite cell system matrix. For locally stabilized systems, we found that the condition number of the SIPIC-preconditioned system matrix is independent of the smallest trimmed cell volume fraction, which is essential to enable robust solutions of finite cell systems. Although it is observed that for globally stabilized formulations a slight dependence of the condition number on the smallest volume fraction remains, the preconditioning technique also dramatically improves the conditioning of such systems.

The performance of the SIPIC preconditioner has been assessed in a comparison to a selection of commonly used preconditioners. We note that in particular incomplete Cholesky preconditioning is observed to be competitive with SIPIC for a large number of testcases, provided that diagonal dominance of the matrix fed to the construction algorithm is ensured by increasing the values on its main diagonal. However, we observe that the robustness of SIPIC is unmatched by incomplete Cholesky preconditioning (and all other considered preconditioners), in the sense that in a significant number of testcases the condition number improvement of the other preconditioners is sub-optimal compared to SIPIC. This sub-optimality is explained by the fact that incomplete Cholesky fails to detect quasi linear dependencies in some cases, and Jacobi preconditioning ignores quasi linear dependencies altogether. Moreover, SIPIC is observed to have a significantly lower relative matrix fill-in compared to incomplete Cholesky and incomplete LU preconditioning. Evidently, the observed robustness in condition number improvement and minimal fill-in of the SIPIC preconditioner stem from the fact that it is completely tailored to immersed systems, this in contrast to the other preconditioners considered in the comparison.

Herein we have restricted ourselves to finite cell systems. However, the sources of ill-conditioning indicate that similar condition number scaling relations hold for other immersed methods, enriched finite element methods such as XFEM and weak coupling strategies, *e.g.,* [29, 31, 32]. We expect the SIPIC preconditioning technique to also be effective for such problems. In this study, we considered piecewise polynomial bases, but the SIPIC preconditioning technique may also be effective with other types of bases. A particular point of attention for the application of SIPIC to different methods or approximation spaces is the selection of an appropriate orthonormalization threshold.

The current work has been restricted to symmetric positive definite systems, such that the bilinear form is an inner product with induced norm. Evidently, this covers a large variety of problems in computational mechanics. However, problems such as convection-dominated flows and mixed problems such as (Navier-)Stokes flows and incompressible elasticity are not within the range of applicability of this work. Generalization of the developments in this manuscript to such problems is a topic of further study.

Acknowledgement

The research of F. de Prenter was funded by the NWO under the Graduate Program Fluid & Solid Mechanics. The research of C.V. Verhoosel was funded by the NWO under the VENI scheme. All simulations in this work were performed using the open source software package Nutils (www.nutils.org).

References

[1] D. Schillinger, L. Dede, M. Scott, J. Evans, M. Borden, E. Rank, T. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces, Computer Methods in Applied Mechanics and Engineering 249 (2012) 116–150.

[2] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, A. Düster, Geometric modeling, isogeometric analysis and the finite cell method, Computer Methods in Applied Mechanics and Engineering 249 (2012) 104–115.

[3] D. Schillinger, M. Ruess, The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models, Archives of Computational Methods in Engineering 22 (3) (2015) 391–455.

[4] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (39) (2005) 4135–4195.

[5] J. Cottrell, T. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, Wiley, 2009.

[6] J. Parvizian, A. Düster, E. Rank, Finite cell method, Computational Mechanics 41 (1) (2007) 121–133.

[7] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Computer Methods in Applied Mechanics and Engineering 197 (45) (2008) 3768–3782.

[8] R. Schmidt, R. Wüchner, K.-U. Bletzinger, Isogeometric analysis of trimmed NURBS geometries, Computer Methods in Applied Mechanics and Engineering 241 (2012) 93–111.

[9] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, E. Rank, Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell methodod, International Journal for Numerical Methods in Engineering 95 (10) (2013) 811–846.

[10] M. Ruess, D. Schillinger, A. Özcan, E. Rank, Weak coupling for isogeometric analysis of nonmatching and trimmed multi-patch geometries, Computer Methods in Applied Mechanics and Engineering 269 (2014) 46–71.

[11] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, E. Rank, The finite cell method for bone simulations: verification and validation, Biomechanics and Modeling in Mechanobiology 11 (3) (2012) 425–437.

[12] C. Verhoosel, G. Van Zwieten, B. Van Rietbergen, R. De Borst, Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone, Computer Methods in Applied Mechanics and Engineering 284 (2015) 138–164.

[13] E. Burman, M. Fernández, An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes, Computer Methods in Applied Mechanics and Engineering 279 (2014) 497–514.

[14] D. Kamensky, M.-C. Hsu, D. Schillinger, J. Evans, A. Aggarwal, Y. Bazilevs, M. Sacks, T. Hughes, An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves, Computer Methods in Applied Mechanics and Engineering 284 (2015) 1005–1053.

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

[16] J. Nitsche, 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, 1971, pp. 9–15.

[17] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.

[18] M. Dauge, A. Düster, E. Rank, Theoretical and Numerical Investigation of the Finite Cell Method, Journal of Scientific Computing 65 (3) (2015) 1039–1064.

[19] T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows, Computer Methods in Applied Mechanics and Engineering 209 (2012) 266–283.

[20] T. Rüberg, F. Cirak, A fixed-grid b-spline finite element technique for fluid–structure interaction, International Journal for Numerical Methods in Fluids 74 (9) (2014) 623–660.

[21] K. Höllig, U. Reif, J. Wipper, Weighted extended B-spline approximation of Dirichlet problems, SIAM Journal on Numerical Analysis 39 (2) (2001) 442–462.

[22] K. Höllig, C. Apprich, A. Streit, Introduction to the WEB-method and its applications, Advances in Computational Mathematics 23 (1) (2005) 215–237.

[23] A. Embar, J. Dolbow, I. Harari, Imposing Dirichlet boundary conditions with Nitsche's method and spline based finite elements, International Journal for Numerical Methods in Engineering 83 (7) (2010) 877–898.

[24] R. Sanches, P. Bornemann, F. Cirak, Immersed b-spline (i-spline) finite element method for geometrically complex domains, Computer Methods in Applied Mechanics and Engineering 200 (13) (2011) 1432–1445.

[25] E. Burman, Ghost penalty, Comptes Rendus Mathematique 348 (21) (2010) 1217–1220.

[26] A. Massing, M. Larson, A. Logg, M. Rognes, A stabilized Nitsche fictitious domain method for the Stokes problem, Journal of Scientific Computing 61 (3) (2014) 604–628.

[27] E. Burman, S. Claus, P. Hansbo, M. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, International Journal for Numerical Methods in Engineering 104 (7) (2015) 472–501.

[28] C. Lehrenfeld, A. Reusken, Optimal preconditioners for Nitsche-XFEM discretizations of interface problems, Numerische Mathematik 135 (2) (2017) 313–332.

[29] P. Zunino, L. Cattaneo, C. Colciago, An unfitted interface penalty method for the numerical approximation of contrast problems, Applied Numerical Mathematics 61 (2011) 1059–1076.

[30] I. Babuška, U. Banerjee, Stable generalized finite element method (SGFEM), Computer Methods in Applied Mechanics and Engineering 201 (2012) 91–111.

[31] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, International Journal for Numerical Methods in Engineering 84 (3) (2010) 253–304.

[32] E. Wadbro, S. Zahedi, G. Kreiss, M. Berggren, A uniformly well-conditioned, unfitted Nitsche method for interface problems, BIT Numerical Mathematics 53 (2013) 791–820.

[33] T. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, 1987.

[34] O. Zienkiewics, R. Taylor, J. Zhu, The Finite Element Method: Its Basis and Fundamentals, Elsevier, 2005.

[35] J. Evans, T. Hughes, Explicit Trace Inequalities for Isogeometric Analysis and Parametric Hexahedral Finite Elements, Numerische Mathematik 123 (2) (2013) 259–290.

[36] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Computer Methods in Applied Mechanics and Engineering 191 (47–48) (2002) 5537–5552.

[37] C. Johnson, Numerical Solution of Partial Differential Equations by the Finite Element Method, Cambridge University Press, 1987.

[38] G. Golub, C. Van Loan, Matrix computations, John Hopkins University Press, 2012.

[39] A. Ern, J. Guermond, Theory and Practice of Finite Elements, Springer, 2004.

Appendix A. Local stabilization of finite cell systems

Coercivity of the finite cell bilinear form $\mathcal{F}(\cdot,\cdot)$, equation (2.4a), can be established by assuring the coercivity of the element-wise restrictions $\mathcal{F}_i(\cdot,\cdot)$. Coercivity of $\mathcal{F}_i(\cdot,\cdot)$ subject to the condition $\beta_i > C_i$ (see (2.7)), follows from

$$\begin{aligned}\mathcal{F}_i(v_h,v_h) \geq & \mathcal{F}_i^1(v_h,v_h) - |\mathcal{F}_i^2(v_h,v_h)| + \mathcal{F}_i^3(v_h,v_h)\\ \geq & \mathcal{F}_i^1(v_h,v_h) - 2\|n\cdot\mathcal{D}\nabla v_h\|_{L^2(\Gamma_i^D)}\|v_h\|_{L^2(\Gamma_i^D)} + \mathcal{F}_i^3(v_h,v_h)\\ \geq & \mathcal{F}_i^1(v_h,v_h) - \varepsilon_i\|n\cdot\mathcal{D}\nabla v_h\|^2_{L^2(\Gamma_i^D)} - \varepsilon_i^{-1}\|v_h\|^2_{L^2(\Gamma_i^D)} + \mathcal{F}_i^3(v_h,v_h)\\ \geq & \mathcal{F}_i^1(v_h,v_h) - C_i\varepsilon_i\mathcal{F}_i^1(v_h,v_h) - \varepsilon_i^{-1}\|v_h\|^2_{L^2(\Gamma_i^D)} + \mathcal{F}_i^3(v_h,v_h)\\ = & \left(1 - C_i\varepsilon_i\right)\mathcal{F}_i^1(v_h,v_h) + \int_{\Gamma_i^D}(\beta_i - \varepsilon_i^{-1})v_h \odot v_h \mathrm{d}S \geq \delta_i\|v_h\|^2_{H^1(\Omega_i^{\mathrm{tr}})},\end{aligned} \tag{A.1}$$

for some $\delta_i > 0$. In the second line in (A.1), the Cauchy-Schwarz inequality is applied. Line three follows from Young's inequality, while equation (2.7) is substituted in the fourth line. The last inequality is based on the fact that there exists a constant ε_i such that $C_i < \varepsilon_i^{-1} < \beta_i$ and applies a specific form of the Poincaré inequality (*e.g.*, Lemma B.63 in [39]). Global coercivity, $\mathcal{F}(v_h,v_h) \geq \delta\|v_h\|^2_{H^1(\Omega)}$ for some $\delta > 0$, follows by summation over all $\mathcal{F}_i$.

The values of C_i can be computed numerically. Because $\mathcal{V}_h(\Omega)$ is finite dimensional, (2.7) can be cast in matrix form as

$$C_i = \max_{\mathbf{x}} \frac{\mathbf{x}^T\mathbf{B}_i\mathbf{x}}{\mathbf{x}^T\mathbf{V}_i\mathbf{x}}, \tag{A.2}$$

with

$$\mathbf{B}_i = \int_{\Gamma_i^D}\Big(n\cdot\mathcal{D}\nabla\mathbf{\Phi}_i\Big)\Big(n\ \cdot\mathcal{D}\nabla\mathbf{\Phi}_i^T\Big)\mathrm{d}S \quad \text{and} \quad \mathbf{V}_i = \mathcal{F}_i^1(\mathbf{\Phi}_i,\mathbf{\Phi}_i^T), \tag{A.3}$$

where $\mathbf{\Phi}_i$ denotes the vector of all basis functions that are supported on Ω_i. The optimality condition associated with (A.2) implies

$$\left(\mathbf{B}_i - \frac{\mathbf{x}^T\mathbf{B}_i\mathbf{x}}{\mathbf{x}^T\mathbf{V}_i\mathbf{x}}\mathbf{V}_i\right)\mathbf{x} = \mathbf{0}, \tag{A.4}$$

because the matrices $\mathbf{B}_i$ and $\mathbf{V}_i$ are both symmetric. The solution to (2.7) therefore coincides with the largest eigenvalue of the generalized eigenvalue problem

$$\mathbf{B}_i\mathbf{x} = \lambda \mathbf{V}_i\mathbf{x}, \tag{A.5}$$

which enables computation of $C_i = \lambda_{\max}$. The numerical solution of this problem requires careful consideration of the following two aspects:

- The matrix $\mathbf{V}_i$, equation (A.3), is singular since it is based solely on the gradients of the shape functions in $\mathbf{\Phi}_i$. This singularity is not a fundamental problem for the computation of C_i in accordance with (2.7), by virtue of the fact that $\ker(\mathbf{V}_i) \subset \ker(\mathbf{B}_i)$. However, the kernel of $\mathbf{V}_i$ can negatively affect the robustness with which the largest eigenvalue can be computed.

- After removal of the kernel of $\mathbf{V}_i$, its conditioning can remain poor on cells with small volume fractions due to quasi linear dependence of basis functions (see Section 4.1.2).

The conditioning of $\mathbf{V}_i$ can be improved using the SIPIC preconditioner, but for these matrices a more direct solution exists. Since the eigenvalue problem is local to the cells, the numerical difficulties related to these conditioning aspects can be circumvented by applying a local change of basis. For scalar problems, quasi linear dependencies are avoided by employing the basis of monomials

$$\phi_{\boldsymbol{\rho}} = \prod_{j=1}^{d} (x_j - \hat{x}_j)^{\rho_j}\,, \tag{A.6}$$

with multi-index $\boldsymbol{\rho} = (\rho_1, \dots, \rho_d)$ for $0 \leq \rho_j \leq p$ and $\hat{x}$ the center of mass of the trimmed cell. With this basis, diagonal scaling suffices for the robust computation of C_i. Centering the monomials at the center of mass of the trimmed cell minimizes the quasi linear dependence problem because higher-order terms occur in separate functions. Therefore their contribution is not diminished by lower-order terms, that are larger in magnitude on small volume fractions of a cell. Furthermore, the kernel of $\mathbf{V}_i$ only consists of the function $\phi_{\mathbf{0}}$, such that removing the kernel is straightforward. For vector valued problems, one can use the functions $\phi_{\boldsymbol{\rho},k} = e_k\phi_{\boldsymbol{\rho}}$, with e_k the unit vector in the k^{th} direction. Besides removing rigid body translations, also rigid body rotations in the kernel of $\mathbf{V}_i$ can be systematically removed from the monomial basis.

Appendix B. Upper bounds for $\mathcal{F}^2(\cdot,\cdot)$ and $\mathcal{F}^3(\cdot,\cdot)$

Under the assumptions in Section 3, we can bound $\mathcal{F}^2(\cdot,\cdot)$ and $\mathcal{F}^3(\cdot,\cdot)$ from above. For the polynomial basis in Section 3.2, we additionally assume that $\exists C_{\mathcal{V}}$ (~ 1), independent of mesh size h, such that for every basis function ϕ in basis $\mathbf{\Phi}$ it holds that

$$\|\phi\|_{L^\infty} \leq C_{\mathcal{V}} \qquad \text{and} \qquad \|\nabla\phi\|_{L^\infty} \leq \frac{C_{\mathcal{V}}}{h}. \tag{B.1}$$

On cell Ω_i we can then bound $\mathcal{F}_i^2(\cdot,\cdot)$ and $\mathcal{F}_i^3(\cdot,\cdot)$:

$$\begin{aligned}
\mathcal{F}_i^2(\phi,\phi) &\leq \int_{\Gamma_i^D} 2\|\phi\|_e C_{\mathcal{D}}\|\nabla\phi\|_e \mathrm{d}S \leq 2|\Gamma_i^D|\|\phi\|_{L^\infty} C_{\mathcal{D}}\|\nabla\phi\|_{L^\infty} && \text{(B.2a)}\\
&\leq 2C_\Gamma C_{\mathcal{D}} C_{\mathcal{V}}^2 h^{d-2}\eta_i^{\frac{d-1}{d}}, \\
\mathcal{F}_i^3(\phi,\phi) &= \int_{\Gamma_i^D} \beta_i\phi^2 \mathrm{d}S \leq |\Gamma_i^D|\beta_i\|\phi\|_{L^\infty}^2 \leq C_\Gamma C_\beta C_{\mathcal{D}} C_{\mathcal{V}}^2 h^{d-2}\eta_i^{\frac{d-2}{d}}. && \text{(B.2b)}
\end{aligned}$$

For $d \geq 2$, the inequalities in (B.2) can be aggregated as

$$\mathcal{F}_i^2(\phi,\phi) + \mathcal{F}_i^3(\phi,\phi) \leq C_{\mathcal{F}_i^{23}} h^{d-2}\eta_i^{\frac{d-2}{d}} \leq C_{\mathcal{F}_i^{23}} h^{d-2}, \tag{B.3}$$

with $C_{\mathcal{F}_i^{23}} = (2 + C_\beta)\, C_\Gamma C_\mathcal{D} C_\mathcal{V}^2$. A similar derivation conveys that for the cross term between two different basis functions ϕ_α and ϕ_β

$$\mathcal{F}_i^2(\phi_\alpha, \phi_\beta) + \mathcal{F}_i^3(\phi_\alpha, \phi_\beta) \leq C_{\mathcal{F}_i^{23}} h^{d-2}. \tag{B.4}$$

The number of functions that is supported on a cell is $(p+1)^d$ for all piecewise polynomial spaces and the number of cells that support a function is is at most $(p+1)^d$ for B-splines and 2^d for most other bases. This bounds the sum of all terms per function by $C_{\mathcal{F}^{23}} h^{d-2}$, with $C_{\mathcal{F}^{23}}^{\text{spline}} = (p+1)^{2d} C_{\mathcal{F}_i^{23}}$ for B-splines and by $C_{\mathcal{F}^{23}}^{\text{other}} = (p+1)^d 2^d C_{\mathcal{F}_i^{23}}$ for most other bases. As a result:

$$\| \left[\mathcal{F}^2(\mathbf{\Phi}, \mathbf{\Phi}^T) + \mathcal{F}^3(\mathbf{\Phi}, \mathbf{\Phi}^T) \right] \mathbf{y} \|_2 \leq C_{\mathcal{F}^{23}} h^{d-2} \|\mathbf{y}\|_2, \tag{B.5}$$

such that

$$\max_{v_h, \mathbf{y}} \frac{\mathcal{F}^2(v_h, v_h) + \mathcal{F}^3(v_h, v_h)}{\mathbf{y}^T \mathbf{y}} \leq C_{\mathcal{F}^{23}} h^{d-2}. \tag{B.6}$$

The bounds in (3.17) follow directly from (B.6). Inequality (B.6) also follows directly from Hölders inequality. Because the sum of all terms per function is bounded by $C_{\mathcal{F}^{23}} h^{d-2}$, this value also bounds the $\|\cdot\|_\infty$ norm and the $\|\cdot\|_1$ norm (due to symmetry) of matrix $\left[\mathcal{F}^2(\mathbf{\Phi}, \mathbf{\Phi}^T) + \mathcal{F}^3(\mathbf{\Phi}, \mathbf{\Phi}^T) \right]$. By Hölders inequality this implies $\| \left[\mathcal{F}^2(\mathbf{\Phi}, \mathbf{\Phi}^T) + \mathcal{F}^3(\mathbf{\Phi}, \mathbf{\Phi}^T) \right] \|_2 \leq C_{\mathcal{F}^{23}} h^{d-2}$.

When a global stabilization parameter is applied (B.2b) no longer holds, because the bound for β does not depend on η_i but on η. Because β is proportional to $h^{-1}\eta^{-1/d}$, we can bound it from below by $\beta > c_\beta h^{-1} \eta^{-1/d}$ for some $0 < c_\beta$. For a coefficient vector $\mathbf{x}$ with the coefficients of basis functions supported on Γ^D set to 1 and all other coefficients set to 0, we have $u_h|_{\Gamma^D} = 1$ such that $\mathcal{F}^3(u_h, u_h) = \beta |\Gamma^D| > c_\beta h^{-1} \eta^{-1/d} |\Gamma^D|$ and $\|\mathbf{x}\|_2^2 < C_{\text{lift}} |\Gamma^D| h^{1-d}$ for some $0 < C_{\text{lift}}$. With this, we can form the following bound

$$\max_{v_h, \mathbf{y}} \frac{\mathcal{F}^3(v^h, v^h)}{\mathbf{y}^T \mathbf{y}} \geq \frac{\mathcal{F}^3(u^h, u^h)}{\mathbf{x}^T \mathbf{x}} > \frac{c_\beta}{C_{\text{lift}}} h^{d-2} \eta^{-1/d} \tag{B.7}$$

Therefore, the condition number of globally stabilized systems is estimated by

$$\kappa_2(\mathbf{A}) \geq c \eta^{-(2p+1-1/d)}, \tag{B.8}$$

which is different from (3.26) and explains the different slope that is observed when global stabilization is applied in Section 3.3 and 4.3.

Appendix C. Preconditioner-construction algorithm

Algorithm 1 outlines the construction of the SIPIC preconditioner $\mathbf{S}$. Since this preconditioner is algebraic, it is constructed solely based on the information contained in the system matrix $\mathbf{A}$. As the SIPIC preconditioner is incomplete in the sense that only functions that are quasi linearly dependent are orthonormalized, an orthonormalization threshold parameter $\gamma \in [0, 1]$ is to be provided as input to the algorithm.

The algorithm is initialized with the construction of the scaling matrix $\mathbf{D}$ and the sets $\mathcal{I}$ and $\mathcal{J}$ that respectively identify the current and total quasi linear dependencies. Subsequently, the linear dependencies are grouped in Σ, and for every sorted local group of indices $\sigma \in \Sigma$ the Gram-Schmidt orthonormalization procedure is applied. Finally, a check is performed to test if any new quasi linear dependencies have emerged after this orthonormalization procedure. If this is the case, these are added to the total set of quasi linear dependencies and the orthonormalization loop is restarted. Otherwise the construction of the SIPIC preconditioner is finished. In our simulations, we have observed convergence of this orthonormalization procedure in at most two iterations, resulting in negligible computational cost for the construction of the SIPIC preconditioner.

The subroutines in Algorithm 1 perform the following operations:

- `scale(`$\mathbf{A}$`)` returns the diagonal matrix $\mathbf{D}$ according to Equation (4.4).

- `identify(`$\mathbf{SAS}^T, \gamma$`)` locates quasi linear dependencies in the scaled system $\mathbf{SAS}^T$ (*i.e.,* the diagonal entries are equal to one). As mentioned in Section 4.1.2, a measure for the linear dependence of two basis functions with indices α and β is derived from the Cauchy-Schwarz inequality:

  $$\frac{|(\phi_\alpha, \phi_\beta)_\mathcal{F}|}{\|\phi_\alpha\|_\mathcal{F}\|\phi_\beta\|_\mathcal{F}} \leq 1, \tag{C.1}$$

  where equality to one indicates linear dependence. Two basis functions are identified as being quasi linearly dependent if the absolute value of the corresponding off-diagonal term in the scaled system matrix exceeds the orthogonalization threshold,

  $$\left|[\mathbf{SAS}^T]_{\alpha\beta}\right| > \gamma, \tag{C.2}$$

  which – exploiting the symmetry of the system matrix – results in the set of index pairs of quasi linearly dependent basis functions:

  $$\mathcal{I} = \left\{(\alpha, \beta) \mid \alpha > \beta, \ \left|[\mathbf{SAS}^T]_{\alpha\beta}\right| > \gamma\right\}. \tag{C.3}$$

- `group(`$\mathcal{I}, \mathbf{A}$`)` returns a set (Σ) of non-intersecting sets ($\sigma \in \Sigma$) in which intersecting sets in $\mathcal{I}$ are replaced by their union. That is, an element $\sigma \in \Sigma$ is a tuple containing the indices of all basis functions that are quasi linear dependent to each other. This tuple is ordered from the index whose corresponding row in the system matrix has the least nonzero entries to the index with the largest number of nonzero row entries in $\mathbf{A}$, in order to reduce fill-in as described in the last paragraph of Section 4.1.2.

- `orthonormalize(`$\mathbf{A}_\sigma$`)` applies the Gram-Schmidt orthonormalization to the functions indexed by σ, such that $\mathbf{S}_\sigma$ becomes the inverse Cholesky decomposition of $\mathbf{A}_\sigma$ and $\mathbf{S}_\sigma\mathbf{A}_\sigma\mathbf{S}_\sigma^T$ becomes the identity matrix.

  Instabilities of the construction algorithm can occur when – due to the presence of cells with extremely small volume fractions – some basis functions become linearly dependent in the sense that the absolute off-diagonal entries (C.2) are up to machine precision equal to one. When this occurs, $[\mathbf{SAS}^T]_{\alpha\alpha} = 0$ for some index α after orthogonalization, and rescaling will result in a division by zero. To stabilize the algorithm, a check is performed to test if $[\mathbf{SAS}^T]_{\alpha\alpha} > \varepsilon$ before rescaling. If this is not the case, the function is eliminated by simply deleting the row α from the preconditioning matrix. The preconditioner $\mathbf{S}$ is then no longer square, but $\mathbf{SAS}^T$ remains square and SPD and is merely reduced in size. This operation does not influence the quality of the obtained solution, because the remaining basis functions span the same approximation space up to machine precision. A value of $\varepsilon = 10^2 \cdot$ `eps` (with `eps` the machine precision) was found to be adequate for all simulations considered in this manuscript.

  As an example of a matrix that is singular up to machine precision we consider the 2×2 matrix

  $$\mathbf{A} = \begin{bmatrix} 1 & 1-\varepsilon^2 \\ 1-\varepsilon^2 & 1 \end{bmatrix}, \tag{C.4}$$

  with some parameter $\varepsilon \ll 1$ and condition number $\kappa_2 = 2/\varepsilon^2 - 1 \gg 1$. The SIPIC preconditioner for $\mathbf{A}$ can be computed analytically as

  $$\mathbf{S} = \begin{bmatrix} 1 & 0 \\ \frac{\varepsilon^2-1}{\varepsilon\sqrt{2-\varepsilon^2}} & \frac{1}{\varepsilon\sqrt{2-\varepsilon^2}} \end{bmatrix}. \tag{C.5}$$

The computation of this matrix by the SIPIC construction algorithm will be impeded by machine precision errors when $\varepsilon^2 \sim \texttt{eps}$. From (C.4), it is directly observed that in this case the rows of matrix $\mathbf{A}$ become linearly dependent up to machine precision. In the preconditioner construction algorithm, this results in the division by zero in the renormalization step after application of the Gram-Schmidt procedure. For this particular case, the Gram-Schmidt matrix equals

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ \varepsilon^2 - 1 & 1 \end{bmatrix}, \tag{C.6}$$

which results in the orthogonalized matrix $\mathbf{A}^\perp = \mathbf{GAG}^T$ (since $\mathbf{D} = \mathrm{diag}(\mathbf{A}) = \mathbf{I}$). Normalization of this matrix results in the division by the square root of

$$\begin{aligned} A^\perp_{22} &= \begin{bmatrix} \varepsilon^2 - 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1-\varepsilon^2 \\ 1-\varepsilon^2 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon^2 - 1 \\ 1 \end{bmatrix} \\ &= 1^2 - (1-\varepsilon^2)^2, \end{aligned} \tag{C.7}$$

which numerically becomes equal to zero when $\varepsilon^2 \sim \texttt{eps}$.

```
Input:  $\mathbf{A}, \gamma$                          # system matrix, orthonormalization threshold
Output: $\mathbf{S}$                                                  # SIPIC preconditioner

# Initialize
$\mathbf{S}$ = scale($\mathbf{A}$)                                            # diagonal scaling
$\mathcal{I} = \mathcal{J}$ = identify($\mathbf{SAS}^T, \gamma$)                          # initial dependencies

# Main loop
while $\mathcal{J} \neq \emptyset$:
    $\Sigma$ = group($\mathcal{I}, \mathbf{A}$)                         # group and order dependencies
    for $\sigma$ in $\Sigma$:
        $\mathbf{S}[\sigma,\sigma]$ = orthonormalize($\mathbf{A}[\sigma,\sigma]$)
    end
    $\mathcal{J}$ = identify($\mathbf{SAS}^T, \gamma$)
    $\mathcal{I} = \mathcal{I} \cup \mathcal{J}$                                  # append current dependencies
end

return $\mathbf{S}$


# Orthonormalization function
def orthonormalize($\mathbf{A}_\sigma$):
    $\mathbf{S}_\sigma$ = scale($\mathbf{A}_\sigma$)                                        # diagonal scaling
    for $i$ in range(len($\mathbf{S}_\sigma$)):                    # Gram-Schmidt orthonormalization
        for $j$ in range($i$):                                                   # $j < i$
            $\mathbf{S}_\sigma[i,:] = \mathbf{S}_\sigma[i,:] - \left(\mathbf{S}_\sigma \mathbf{A}_\sigma \mathbf{S}_\sigma^T\right)[i,j]\, \mathbf{S}_\sigma[j,:]$
        end
        $\mathbf{S}_\sigma[i,:] = \frac{\mathbf{S}_\sigma[i,:]}{\left(\sqrt{\mathbf{S}_\sigma \mathbf{A}_\sigma \mathbf{S}_\sigma^T}\right)[i,i]}$                                    # rescale
    end
    return $\mathbf{S}_\sigma$
end
```

Algorithm 1: Construction of the SIPIC preconditioner.