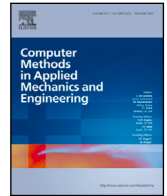


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

## Comput. Methods Appl. Mech. Engrg.

journal homepage: [www.elsevier.com/locate/cma](http://www.elsevier.com/locate/cma)

# An approximate block factorization preconditioner for mixed-dimensional beam-solid interaction

Max Firmbach<sup>a</sup>, Ivo Steinbrecher<sup>a</sup>, Alexander Popp<sup>a</sup>, Matthias Mayr<sup>a,b,\*</sup>

<sup>a</sup> Institute for Mathematics and Computer-Based Simulation, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, D-85577 Neubiberg, Germany

<sup>b</sup> Data Science & Computing Lab, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, D-85577 Neubiberg, Germany

## ARTICLE INFO

Dataset link: <https://github.com/trilinos/Trilinos>

### Keywords:

Approximate block factorization  
Physics-based block preconditioning  
Algebraic multigrid  
Sparse approximate inverse  
mixed-dimensional modeling  
beam-solid interaction

## ABSTRACT

This paper presents a scalable approximate block factorization preconditioner for mixed-dimensional models in beam-solid interaction and their application in engineering. In particular, it studies the linear systems arising from a regularized mortar-type approach for embedding geometrically exact beams into solid continua. Due to the lack of block diagonal dominance of the arising  $2 \times 2$  block system, an approximate Block-LU preconditioner is used. It exploits the sparsity structure of the beam sub-block to construct a sparse approximate inverse, which is then not only used to explicitly form an approximation of the Schur complement, but also acts as a smoother within the prediction and correction step of the arising Block-LU preconditioner. The Schur complement equation is tackled with an algebraic multigrid method. Although, for now, the beam sub-block is tackled by a one-level method only, the multi-level nature of the computationally demanding Schur equation delivers a scalable preconditioner in practice. In numerical test cases, the influence of different algorithmic parameters on the quality of the sparse approximate inverse is studied and the weak scaling behavior of the proposed preconditioner on up to 1000 MPI ranks is demonstrated. In addition, the robustness of the proposed method regarding material parameters and geometric properties is shown, before the preconditioner is finally applied for the analysis of steel-reinforced concrete structures in civil engineering.

## 1. Introduction

Originating from evolution in nature or human design processes, thin fibers embedded into solid continua can enhance the constitutive or functional properties of systems in science, engineering, and biomedicine. Applications can be found in different fields: In civil engineering for example, steel-reinforced concrete is used to amplify the load bearing capacity of concrete structures such as bridges. In aerospace engineering, fiber-reinforced composite materials are often used due to their unique combination of a high stiffness, but low specific weight. In biological tissues, collagen fibers are distributed throughout the arterial walls of the circulatory system. For all these application areas, finite element simulations can provide detailed insight in the system's behavior and potentially assist in improving or optimizing the system's design. While various mathematical models of fiber-enhanced continua are available in literature, their efficient solution on parallel computing clusters has not been studied in detail. To this

\* Corresponding author at: Institute for Mathematics and Computer-Based Simulation, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, D-85577 Neubiberg, Germany.

E-mail addresses: [max.firmbach@unibw.de](mailto:max.firmbach@unibw.de) (M. Firmbach), [ivo.steinbrecher@unibw.de](mailto:ivo.steinbrecher@unibw.de) (I. Steinbrecher), [alexander.popp@unibw.de](mailto:alexander.popp@unibw.de) (A. Popp), [matthias.mayr@unibw.de](mailto:matthias.mayr@unibw.de) (M. Mayr).

<https://doi.org/10.1016/j.cma.2024.117256>

Received 28 February 2024; Received in revised form 24 June 2024; Accepted 22 July 2024

Available online 30 July 2024

0045-7825/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

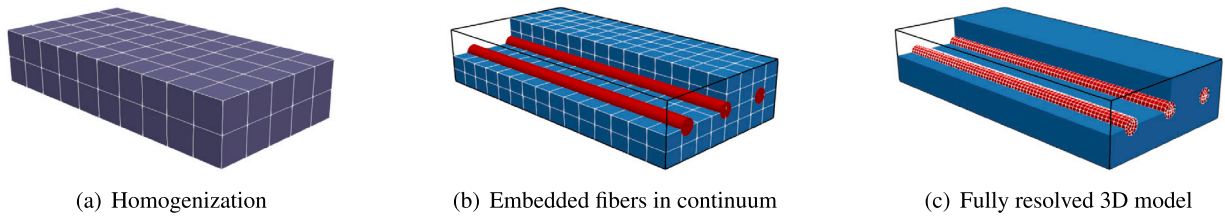


Fig. 1. Spectrum of modeling techniques for fibers embedded into three-dimensional solids [1].

end, this paper sets out to develop an efficient and scalable multi-level block preconditioning framework for a penalty-regularized mixed-dimensional approach recently proposed by Steinbrecher et al. [1,2].

Fig. 1 illustrates the range of modeling techniques for fully embedded fibers in solid bulk volumes. On the one hand, homogenized formulations as depicted in Fig. 1(a) incorporate all fiber information into the bulk constitutive law, usually leading to anisotropic formulations with preferential directions along the fiber orientation [3,4]. On the other end of the spectrum, both bulk field and fibers are resolved as three-dimensional (3D) continua, cf. Fig. 1(c), which allows the reuse of existing constitutive models and finite element technology from classical 3D computational solid dynamics. This approach enables the analysis of very detailed micro-mechanical features of individual fibers and the incorporation of advanced physical effects at the fiber-solid interface, though it comes at significant computational cost. Finally, to unify the high model quality of fully resolved models with the efficiency of homogenized models, fibers can be represented by dimensionally reduced structural models such as beams or trusses, which are embedded at arbitrary positions into a 3D solid domain (cf. Fig. 1(b)). In such mixed-dimensional 1D/3D models, the bulk field still portrays the same effects as in the fully resolved 3D model, but fibers are now reduced to a computationally efficient one-dimensional (1D) representation, making such models good candidates to study fiber-enhanced continua *at large scale*.

For the enforcement of coupling conditions in mixed-dimensional models, embedded mesh techniques are required. Although the imposition of constraints through Lagrange multipliers is well established in computational solid mechanics for both boundary-fitted meshes [5,6] and embedded meshes [7,8] as well as in computational contact mechanics [9–12], the construction of stable Lagrange multiplier spaces for mixed-dimensional beam-solid coupling still poses an open research question. Consequently, mixed-dimensional models for solid problems so far either use a penalty regularization to enforce the fiber-solid coupling constraints [1,2,13–15], employ a variationally consistent overlapping domain decomposition approach [16], or directly link embedded fibers to the surrounding volume discretization via the extended finite element method (XFEM) [17,18]. Similarly, mixed-dimensional 1D/3D models are also available for other types of physics, e.g., in fluid-beam interaction (FBI) [19–21] to counteract 3D/3D fluid-solid interaction (FSI) models such as [22,23].

Naturally, computational benefits of mixed-dimensional 1D/3D models are expected, since beam models require much fewer degrees of freedom (DOFs) than solid models to represent the embedded fibers. In [20], the mixed-dimensional approach reduces the number of DOFs for fiber modeling by 96% while keeping the  $L_2$  error of the bulk field below 1.5%. Yet, the solution process of mixed-dimensional models at large scale is not widely studied in literature, but will be tackled in this contribution. In particular, when using Krylov methods to iteratively solve the arising linear systems, suitable preconditioners are required to sufficiently improve the spectral properties of the linear system [24]. A few approaches can be found in literature: Block diagonal preconditioners for saddle point systems arising from 1D/2D coupling are investigated in [25]. A simplified model problem of 1D/3D coupling yielding a  $3 \times 3$  saddle point system is studied in [26], where a fractional Laplacian is used to approximate the Schur complement and a block diagonal preconditioner is employed for the coupled problem. An additive multigrid preconditioner for the arising fractional Laplacian is proposed in [27]. With mixed-dimensional 1D/3D couplings as one application area of interface-driven multi-physics problems, uniform convergence, parameter robustness and scalability have been achieved through a suitable subspace splitting and custom smoothers in [28]. Based on the framework of operator preconditioning [29], robust preconditioners have been proposed for 1D/3D domains coupled with Lagrange multipliers for applications in micro-circulation [30]. For  $2 \times 2$  systems, the use of Ruge–Stüben algebraic multigrid (AMG) methods to solve a Schur complement equation for the 3D bulk domain, while tackling the 1D domain by a direct solver, is briefly discussed in [31]. For penalty-based 1D/3D models and in particular for beam models serving as 1D models, preconditioners are yet to be developed.

In this paper, we will focus on our prior work on a regularized mortar-type embedding of geometrically exact nonlinear beams into 3D solids [1,2] and will devise scalable preconditioners for the arising systems of linear equations. We will study key properties of the linear systems, in particular their loss of both diagonal dominance and block diagonal dominance due to the penalty contributions, and design a preconditioner that is mostly agnostic to these challenges. To this end, we will interpret the system matrix containing solid and beam contributions as a  $2 \times 2$  block matrix and employ approximate block factorizations to arrive at a Block-LU preconditioner. For the approximate inversion of the beam sub-block of the coupled system matrix, we will construct a SPAI [32], which will not only allow us to explicitly form an approximate Schur complement, but will also serve as a smoother within the application of the Block-LU preconditioner. The original SPAI algorithm will be equipped with filtering and static enrichment algorithms to amplify its performance and robustness. To achieve scalability on parallel computing clusters, we will tackle the Schur complement equation itself with AMG methods from TRILINOS/MUELU [33]. We will then study the computational performance and demonstrate weak scalability, robustness under changes of physical parameters, and applicability to practical use cases in a series

of numerical experiments and investigate savings in wall clock time due to the reuse of the preconditioner throughout the entire load step. In sum, this contribution builds upon our prior work [1,2] and equips these models with scalable iterative solvers to facilitate their efficient application to large models and systems with thousands of embedded fibers. Furthermore, this contribution constitutes the first presentation of a scalable, preconditioned iterative solver for truly 1D/3D models applied to beam-solid coupling with weak scalability demonstrated on a distributed memory cluster using the message passing interface (MPI) on up to 1000 MPI ranks.

The remainder of this manuscript is organized as follows: Section 2 introduces the underlying mechanical problem of mixed-dimensional couplings of slender fibers embedded into solid continua and outlines the finite element discretization and the resulting linear systems. Relevant properties of these linear systems are then discussed in Section 3. The design of the preconditioner and its building blocks, in particular the SPAI algorithm, will be detailed in Section 4 along with a brief comparison to existing methods from literature. In Section 5, we will study the numerical properties of individual components of the preconditioner and assess its performance and scalability when applied to academic and engineering test cases. Section 6 will summarize our findings and hint at future research directions.

## 2. Mixed-dimensional modeling of fiber/solid systems

Since this manuscript concerns itself with preconditioner development for the mixed-dimensional modeling of the interaction of solid continua with slender fibers, we only give a brief introduction into the governing equations, discretization and coupling approach presented in [1,2]. For a broader overview of the fundamentals of beam-solid interaction, the interested reader is referred to [34].

### 2.1. Pure solid problem

The 3D solid bodies considered in this work are modeled as hyperelastic Boltzmann continua. The weak form of the equation of elastostatics describing the deformation of the solid body is given as

$$\delta W^S = \int_{\Omega^S} \mathbf{S} : \delta \mathbf{E} \, dV - \int_{\Omega^S} \mathbf{b} \cdot \delta \mathbf{u}^S \, dV - \int_{\Gamma^S} \mathbf{t} \cdot \delta \mathbf{u}^S \, dA = 0$$

with  $\mathbf{u}^S$  being the solid displacement field,  $\mathbf{S}$  and  $\mathbf{E}$  representing the second Piola–Kirchhoff stress tensor and the Green–Lagrange strain tensor,  $\mathbf{b}$  denoting the body forces,  $\mathbf{t}$  standing for the external traction field, and  $\delta$  indicating virtual, but kinematically admissible quantities in line with the concept of virtual work. Furthermore,  $\Omega^S$  is the solid domain and  $\Gamma^S$  is the Neumann boundary of the solid domain. In this work, all quantities denoted with an  $(\cdot)^S$  are associated with the solid continuum. For the spatial discretization of the solid domain, we employ displacement-based isoparametric finite elements interpolated by Lagrange polynomials resulting in the following discretized linear system

$$\mathbf{K}_{ss}^S \Delta \mathbf{d}^S = -\mathbf{f}_s^S$$

to be solved in every iteration of the Newton solver. Therein,  $\mathbf{f}_s^S$  represents the nonlinear residual vector and  $\mathbf{K}_{ss}^S$  denotes its linearization, i.e., the solid stiffness matrix. The discrete displacement vector and its increment are given by  $\mathbf{d}^S$  and  $\Delta \mathbf{d}^S$ , respectively.

### 2.2. Coupled beam-solid system with Simo–Reissner beam formulation

In the scope of this publication, Simo–Reissner (SR) and torsion-free Kirchhoff–Love (TF) beam theories are considered. With respect to the coupled beam-solid problem, the SR beam theory results in the most general coupling formulation. We first describe the coupled problem based on a SR beam formulation and defer the use of TF beam formulations to Section 2.3.

The general weak form for a 1D Cosserat continuum (applicable to SR and TF beam theory) is given by

$$\delta W^B = \delta \Pi_{int,(\cdot)}^B - \delta W_{ext}^B = 0,$$

where  $\delta \Pi_{int,(\cdot)}^B$  denotes the variation of the internal elastic energy function and  $\delta W_{ext}^B$  is the external virtual work acting on the beam. All quantities denoted with the superscript  $(\cdot)^B$  are associated with beam contributions. The internal elastic energy for a SR beam, i.e., a shear-deformable beam with six local modes of deformation, reads

$$\delta \Pi_{int,SR}^B = \frac{1}{2} \int_{\Omega^B} \Gamma^T \mathbf{C}_F \Gamma + \Omega^T \mathbf{C}_M \Omega \, ds.$$

Here,  $\Gamma$  represents the material deformation,  $\Omega$  the material curvature, and  $\mathbf{C}_F$  as well as  $\mathbf{C}_M$  are the respective constitutive matrices for the translational and rotational deformation modes. We refer the interested reader to [35,36] for more details on the SR beam theory and its finite element discretization.

The weak form of the beam is defined on the undeformed 1D centerline domain  $\Omega^B$ . The spatial interpolation of the beam finite elements is tailored to the particular beam model in order to ensure objectivity of the discrete formulation. The interpolation of the beam centerline position employs  $C^1$ -continuous third-order Hermite shape functions [36]. An objective interpolation of the finite cross section orientations is a non-trivial task and results in a nonlinear and deformation-dependent interpolation strategy. For a

more detailed description of this topic, the reader is referred to [36] and the references therein. The resulting linearized system of a pure SR beam problem reads

$$\begin{pmatrix} \mathbf{K}_{rr}^B & \mathbf{K}_{r\theta}^B \\ \mathbf{K}_{\theta r}^B & \mathbf{K}_{\theta\theta}^B \end{pmatrix} \begin{pmatrix} \Delta \mathbf{d}^B \\ \Delta \boldsymbol{\theta}^B \end{pmatrix} = - \begin{pmatrix} \mathbf{f}_r^B \\ \mathbf{f}_\theta^B \end{pmatrix}.$$

To clarify the following equations and simplifications in the TF case, the global discrete beam centerline degrees of freedom  $\mathbf{d}^B$  are gathered together as are the global beam orientational degrees of freedom  $\boldsymbol{\theta}^B$ . Therefore, the beam stiffness matrix is written as a  $2 \times 2$  block system with the individual blocks  $\mathbf{K}_{ab}^B$  with indices  $a, b \in \{r, \theta\}$ . Similarly, the residual is split into positional and rotational contributions  $\mathbf{f}_r^B$  and  $\mathbf{f}_\theta^B$ , respectively.

To fully embed the beam in the solid matrix, the following six local coupling constraints are defined:

$$\mathbf{0} = \mathbf{r}^B - \mathbf{x}^S \text{ on } \Omega^B, \quad (1)$$

$$\mathbf{0} = \boldsymbol{\psi}^{BS} \text{ on } \Omega^B. \quad (2)$$

Here,  $\mathbf{r}^B$  is the beam centerline position vector,  $\mathbf{x}^S$  is the solid position vector, and  $\boldsymbol{\psi}^{BS}$  denotes the (pseudo-)rotation vector describing the relative rotation between a suitable orthonormal triad field in the solid domain and the beam cross section orientation. The constraints given in (1) are referred to as the *positional* coupling constraints, since they enforce the position of the beam cross section centroid to be coupled to the underlying solid. In a similar manner, (2) is referred to as *rotational* coupling since a vanishing relative rotation enforces the beam cross section orientation to be coupled to the solid domain. For a more elaborate discussion on the coupling constraints, the interested reader is referred to [1,2]. The coupling constraints (1) and (2) are enforced via a Lagrange multiplier method. The total Lagrange multiplier potential reads

$$\Pi_{\lambda, \text{SR}} = \int_{\Omega^B} (\boldsymbol{\lambda}^V)^T (\mathbf{r}^B - \mathbf{x}^S) \, ds + \int_{\Omega^B} (\boldsymbol{\lambda}^R)^T \boldsymbol{\psi} \, ds, \quad (3)$$

where  $\boldsymbol{\lambda}^V$  and  $\boldsymbol{\lambda}^R$  are the Lagrange multiplier fields introduced to enforce the positional and rotational coupling, respectively. Accordingly, quantities denoted with  $(\cdot)^V$  and  $(\cdot)^R$  refer to positional and rotational coupling, respectively.

For the spatial interpolation of the Lagrange multiplier fields, we resort to the mortar-type approach and software implementation of [1,2]. There it is shown that a linear interpolation of the Lagrange multiplier field with a subsequent penalty regularization of the saddle point system results in a stable coupling formulation for our envisioned application range. The resulting global discrete Lagrange multiplier vectors for positional and rotational coupling are denoted with  $\boldsymbol{\lambda}^V$  and  $\boldsymbol{\lambda}^R$ , respectively. Variation of the coupling potential (3) and insertion of the spatially discretized quantities results in the weak form of the coupling formulation. With that we can now state the global discretized residual vector for the coupled problem:

$$\begin{pmatrix} \mathbf{f}_r^B + \mathbf{f}_{mt,r}^V \\ \mathbf{f}_\theta^B + \mathbf{f}_{mt,\theta}^R \\ \mathbf{f}_s^S + \mathbf{f}_{mt,s}^V + \mathbf{f}_{mt,s}^R \\ \mathbf{g}^V \\ \mathbf{g}^R \end{pmatrix} = \mathbf{0}.$$

Here,  $\mathbf{g}^V$  and  $\mathbf{g}^R$  are the constraint residual vectors for positional and rotational coupling, respectively. Moreover,  $\mathbf{f}_{mt,(c)}^{(c)}$  are the discrete coupling force contributions required to enforce the action of the Lagrange multipliers on the beam and solid degrees of freedom. We resort to a penalty regularization with the penalty parameters  $\epsilon^V > 0$  and  $\epsilon^R > 0$  to express the Lagrange multiplier unknowns  $\boldsymbol{\lambda}^V$  and  $\boldsymbol{\lambda}^R$  in terms of beam and solid unknowns, i.e.,

$$\boldsymbol{\lambda}^V \approx \epsilon^V (\mathbf{V}^V)^{-1} \mathbf{g}^V, \quad (4)$$

$$\boldsymbol{\lambda}^R \approx \epsilon^R (\mathbf{V}^R)^{-1} \mathbf{g}^R. \quad (5)$$

Therein, the diagonal matrices  $\mathbf{V}^V$  and  $\mathbf{V}^R$  are scaling matrices to scale the regularized equations in order to pass patch-test like problems, cf. [1]. With the penalty regularization, the Lagrange multipliers are no longer unknowns. Thus, we can state the final linearized system to be solved in every Newton iteration:

$$\begin{pmatrix} \mathbf{K}_{rr}^B + \epsilon^V \mathbf{D}^T (\mathbf{V}^V)^{-1} \mathbf{D} & \mathbf{K}_{r\theta}^B & -\epsilon^V \mathbf{D}^T (\mathbf{V}^V)^{-1} \mathbf{M} \\ \mathbf{K}_{\theta r}^B & \mathbf{K}_{\theta\theta}^B + \mathbf{Q}_{\theta\theta} + \epsilon^R \mathbf{Q}_{\theta\lambda} (\mathbf{V}^R)^{-1} \mathbf{Q}_{\lambda\theta} & \mathbf{Q}_{\theta s} + \epsilon^R \mathbf{Q}_{\theta\lambda} \mathbf{V}^{-1} \mathbf{Q}_{\lambda s} \\ -\epsilon^V \mathbf{M}^T (\mathbf{V}^V)^{-1} \mathbf{D} & \mathbf{Q}_{s\theta} + \epsilon^R \mathbf{Q}_{s\lambda} (\mathbf{V}^R)^{-1} \mathbf{Q}_{\lambda\theta} & \mathbf{K}_{ss}^S + \mathbf{Q}_{ss} + \epsilon^V \mathbf{M}^T (\mathbf{V}^V)^{-1} \mathbf{M} + \epsilon^R \mathbf{Q}_{s\lambda} (\mathbf{V}^R)^{-1} \mathbf{Q}_{\lambda s} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{d}^B \\ \Delta \boldsymbol{\theta}^B \\ \Delta \mathbf{d}^S \end{pmatrix} = - \begin{pmatrix} \mathbf{f}_r^B \\ \mathbf{f}_\theta^B \\ \mathbf{f}_s^S \end{pmatrix}. \quad (6)$$

Here,  $\mathbf{D}$  and  $\mathbf{M}$  are the so called mortar matrices for positional coupling, which only depend on the reference configuration, i.e., they are constant. The matrices  $\mathbf{Q}_{ab}$  with  $a, b \in \{s, \theta, \lambda\}$  are the coupling matrices for rotational coupling, which depend on the current configuration. We refer to the original publications [1,2] for details of the linearization procedure.

### 2.3. Coupled beam-solid system with torsion-free (TF) beam formulation

The torsion-free Kirchhoff–Love (TF) beam formulation (see [35,36]) represents a special case of the Simo–Reissner beam theory, where the assumptions of vanishing shear and torsion deformations are incorporated in the beam model and the resulting finite element formulation can be described solely by displacement degrees of freedom. The assumptions are valid for fibers with high slenderness ratios, a double symmetric cross section and a straight centerline in the reference configuration. For such fibers, the TF beam formulation results in an even more efficient numerical model than the SR formulation presented in the previous section. The internal energy of a TF beam reads

$$\delta \Pi_{int,TF}^B = \frac{1}{2} \int_{\Omega^B} EA \epsilon^2 + EI \kappa^2 ds$$

with  $E$  denoting the Young's modulus,  $A$  the cross section area,  $I$  the moment of inertia,  $\epsilon$  the axial tension and  $\kappa$  the scalar curvature, respectively. The TF beam formulation requires a  $C^1$ -continuous centerline interpolation of centerline positions  $\mathbf{r}$ , which is realized with third-order Hermite polynomials [35].

As the only field of unknowns along the TF beam centerline is a translation field, the total Lagrange multiplier potential of the coupling constraints in (3) simplifies to

$$\Pi_{\lambda,TF} = \int_{\Gamma} (\lambda^V)^T (\mathbf{r}^B - \mathbf{x}^S) ds$$

i.e., there is no rotational coupling for TF beams. The same penalty regularization as stated in (4) is employed, resulting in the following global linearized system :

$$\begin{pmatrix} \mathbf{K}_{rr}^B + \epsilon^V \mathbf{D}^T (\mathbf{V}^V)^{-1} \mathbf{D} & -\epsilon^V \mathbf{D}^T (\mathbf{V}^V)^{-1} \mathbf{M} \\ -\epsilon^V \mathbf{M}^T (\mathbf{V}^V)^{-1} \mathbf{D} & \mathbf{K}_{ss}^S + \epsilon^V \mathbf{M}^T (\mathbf{V}^V)^{-1} \mathbf{M} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{d}^B \\ \Delta \mathbf{d}^S \end{pmatrix} = - \begin{pmatrix} \mathbf{f}_r^B \\ \mathbf{f}_s^S \end{pmatrix}. \quad (7)$$

Again, for a more detailed information on the derivations, the interested reader is referred to [1]. It should be noted, that the coupling contributions in (7), i.e.,  $\mathbf{D}$  and  $\mathbf{M}$ , are constant due to the coupling taking place in the reference configuration.

### 3. Characteristics of linear systems arising in beam-solid coupling

To be able to construct efficient algebraic block preconditioning techniques to accelerate the convergence of the outer Krylov solver, certain matrix properties of the underlying linear systems  $\mathcal{A}\mathbf{x} = \mathbf{b}$  specified in (6) and (7) are of particular interest. A short explanation regarding conditioning, block diagonal dominance, symmetry and sparsity pattern is given below. For the sake of simplicity, the block systems in (6) and (7) are both abbreviated with the compact notation

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}^B \\ \mathbf{x}^S \end{pmatrix} = \begin{pmatrix} \mathbf{b}^B \\ \mathbf{b}^S \end{pmatrix} \quad \text{with } \mathbf{r} = \begin{pmatrix} \mathbf{r}^B \\ \mathbf{r}^S \end{pmatrix} = \begin{pmatrix} \mathbf{b}^B \\ \mathbf{b}^S \end{pmatrix} - \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}^B \\ \mathbf{x}^S \end{pmatrix} \quad (8)$$

throughout the remainder of this manuscript, where we have grouped the unknowns based on their physical meaning, i.e., being associated with the beams or the background solid. The concrete identification of individual matrix blocks in (8) with the linear systems from Sections 2.2 and 2.3 depends on the employed beam theory. In the case of a SR beam theory, the individual blocks are defined such that (8) represents (6) with  $\mathbf{b}^B = (\Delta(\mathbf{d}^B)^T, (\Delta\theta^B)^T)^T$  and  $\mathbf{b}^S = \Delta \mathbf{d}^S$ . For the TF case, (8) represents (7) with  $\mathbf{b}^B = \Delta \mathbf{d}^B$  and  $\mathbf{b}^S = \Delta \mathbf{d}^S$ . Generally speaking,  $\mathbf{A}$  denotes the matrix block containing the beam stiffness matrices  $\mathbf{K}_{ab}^B$  as well as stiffness and penalty contributions of the coupling constraints. In similar fashion, the sub-block  $\mathbf{C}$  refers to the sum of the solid stiffness matrix  $\mathbf{K}_{ab}^S$  and the respective interaction terms. The off-diagonal sub-blocks  $\mathbf{B}_1^T$  and  $\mathbf{B}_2$  represent the remaining coupling terms between both fields, respectively.

#### 3.1. Ill-conditioning due to penalty regularization

Due to the discretization and coupling approach introduced in Section 2, the linear system of equations suffers from ill-conditioning, which directly originates from the penalty parameters  $\epsilon^V$  and  $\epsilon^R$ , that are steering the strength of the interaction between solid and fibers. Naturally, larger values of  $\epsilon^V$  and  $\epsilon^R$  lead to a more accurate constraint enforcement, however enlarge the eigenvalue spectrum of the matrix and, thus, worsen the conditioning problems. To show this exemplarily, a small eigenvalue study is done for test cases I and IV introduced later in Section 5.1 with all parameters fixed despite  $\epsilon^V$  and  $\epsilon^R$  as shown in Table 1.

For both beam models, an increase of the penalty parameters results in an increasing maximum eigenvalue  $\lambda_{max}$  of the overall system, while the minimum eigenvalue  $\lambda_{min}$  remains constant. This results in growing condition number estimates given by the well-established definitions  $\lambda_{max}/\lambda_{min}$  and  $\|\mathcal{A}\|_1 \|\mathcal{A}^{-1}\|_1$ . Despite the bad conditioning, sub-block  $\mathbf{A}$  is still nonsingular, being an essential requirement for factorizations of the block matrix  $\mathcal{A}$ .

**Table 1**

Minimum and maximum eigenvalues  $\lambda_{min}$ ,  $\lambda_{max}$  of the given block system with condition number estimates for different beam models and varying positional and rotational penalty parameters  $\epsilon^V$ ,  $\epsilon^R$ .

Beam model	$\epsilon^V$	$\epsilon^R$	$\lambda_{min}$	$\lambda_{max}$	$\lambda_{max}/\lambda_{min}$	$\ \mathbf{A}\ _1 \ \mathbf{A}^{-1}\ _1$
TF	1	–	$0.3920 \times 10^{-7}$	1.0000	$2.5510 \times 10^7$	$4.0078 \times 10^7$
TF	10	–	$0.3920 \times 10^{-7}$	1.5042	$3.8372 \times 10^7$	$4.7060 \times 10^7$
TF	100	–	$0.3920 \times 10^{-7}$	14.915	$3.8048 \times 10^8$	$4.6951 \times 10^8$
TF	1000	–	$0.3920 \times 10^{-7}$	149.03	$3.8018 \times 10^9$	$4.6868 \times 10^9$
SR	1	0.1	$0.1047 \times 10^{-6}$	4.7644	$4.5505 \times 10^7$	$1.9389 \times 10^6$
SR	10	1	$0.1047 \times 10^{-6}$	47.199	$4.5081 \times 10^8$	$1.9089 \times 10^7$
SR	100	10	$0.1047 \times 10^{-6}$	471.55	$4.5039 \times 10^9$	$1.8604 \times 10^8$
SR	1000	100	$0.1047 \times 10^{-6}$	4715.1	$4.5034 \times 10^{10}$	$1.9031 \times 10^9$

### 3.2. Loss of block diagonal dominance

Independent of the actual choice of the beam model within beam-solid interaction, the matrices in the arising linear systems in (6) and (7) exhibit a  $2 \times 2$  block structure based on a physically motivated grouping of unknowns into beam unknowns  $\mathbf{x}^B$  and solid unknowns  $\mathbf{x}^S$ , respectively. This becomes particularly evident in the unified notation of (8). For a closer look at the arising matrices, we adopt the concept of *block diagonal dominance* for block matrices from [37]:

**Definition 3.1 (Block Diagonal Dominance).** Let  $\mathbf{H} \in \mathbb{C}^{N_R \times N_R}$  be a square block matrix with  $N_R$  block rows and block columns, respectively. With a given matrix norm  $\|\bullet\|$ , we assume  $\mathbf{H}$  to only contain nonsingular matrix blocks  $\mathbf{H}_{ii}$  on its main diagonal, i.e.,  $\det \mathbf{H}_{ii} \neq 0 \forall i \in 1, \dots, N_R$ . Then, a matrix  $\mathbf{H}$  is referred to as *block diagonally dominant*, if

$$\sum_{\substack{\zeta=1 \\ \zeta \neq i}}^{N_R} \|\mathbf{H}_{i\zeta}\| \leq \|\mathbf{H}_{ii}^{-1}\|^{-1} \quad \text{for } i = 1, \dots, N_R. \quad (9)$$

In general, the matrices in (6) and (7) do not satisfy the conditions for block diagonal dominance as outlined in Definition 3.1. For illustration purposes, we consider the mixed-dimensional modeling approach from Section 2 in the practical case of fiber-reinforced solids, where the stiffness of the fibers is much higher than the stiffness of the embedding solid, i.e.,  $E^B \gg E^S$ . To this end, we assume a fixed geometry and mesh, constant material parameters, and fiber and solid constitutive properties satisfying  $E^B \gg E^S$ . Since the projection operators  $\mathbf{D}$  and  $\mathbf{M}$  solely depend on the mesh, the only variable parameters left are  $\epsilon^V \gg 0$  and  $\epsilon^R \gg 0$ . With increasing penalty parameters  $\epsilon^V$  and  $\epsilon^R$ , the norm of the off-diagonal matrix blocks increases, too. In addition, the inversion of the diagonal matrix blocks results in denser matrices with a rapid growth of the norm and, thus, decreasing values on the right-hand side of inequality (9). For positional coupling, the block diagonal dominance property of the matrix becomes harder to achieve with an increasing penalty parameter  $\epsilon^V$ , especially for  $\epsilon^V \approx E^B$  as recommended for practical computations. The same holds true for the rotational coupling contributions for the recommended choice  $\epsilon^R \approx E^B R^2$  with  $R$  being the radius of the beam along the centerline (see [14]). To support this argument, we assess the property of block diagonal dominance for the first block row in (8), i.e., specifically the contribution of  $\mathbf{A}$ , by anticipating a small numerical example, in particular test case I introduced later in Section 5.1. Therein, the off-diagonal matrix block  $\mathbf{B}_1$  exhibits a norm  $\|\mathbf{B}_1\| = 1.0153$ , while the main diagonal block  $\mathbf{A}$ 's contribution evaluates to  $\|\mathbf{A}^{-1}\|^{-1} = 3.9202 \cdot 10^{-8}$ , hence violating the condition outlined in (9).

Due to this lack of block diagonal dominance, conventional block preconditioning methods based on block Jacobi or block Gauß–Seidel schemes are not applicable without major convergence problems (or even divergence) as already evidenced in [38,39]. Independently, the individual matrix blocks on the main diagonal lose their diagonal dominance as well due to the penalty contributions. Hence, conventional relaxation-based smoothers cannot be applied on individual blocks either.

### 3.3. Potential loss of symmetry

The symmetry of  $\mathbf{A}$  is governed by the beam formulation at hand: TF beam models always result in a symmetric  $\mathbf{A}$ , while all other beam models yield a non-symmetric beam sub-block  $\mathbf{A}$ . For pure positional coupling as applied for TF beam models, the off-diagonal matrix block are symmetric, i.e.,  $\mathbf{B}_1 = \mathbf{B}_2$ . In contrast, the additional coupling of rotational degrees of freedom for SR beams introduces non-symmetric off-diagonal terms.

### 3.4. Sparsity pattern

Of special interest is the matrix graph  $\mathcal{J}(\mathbf{A})$  of the sub-block  $\mathbf{A}$  related to the beam problem. Since we restrict ourselves so far to cases where the embedded fibers do not interact with each other but only with the surrounding solid, the matrix  $\mathbf{A}$  features a block diagonal sparsity structure, where the size of the small blocks depends on the number of beam finite elements used to discretize each individual fiber. We will illustrate and study the sparsity pattern  $\mathcal{J}(\mathbf{A})$  for different test cases in Section 5.1, in particular in Fig. 4.

#### 4. Block preconditioning for beam-solid interaction

The construction of a preconditioner, that captures the coupling interactions properly and is tailored to the specific matrix properties, is crucial for an efficient and scalable solution process. Preconditioners based on approximate block factorizations have been shown to be suited for similar problem types such as contact problems [12], incompressible flow [40,41], FSI [42,43], or magneto-hydro dynamics [38,44,45].

##### 4.1. Preconditioning based on a block factorization of the system matrix

Due to the lack of block diagonal dominance and the ill-conditioning of the matrix discussed in Section 3, we resort to a block factorization preconditioner. Specifically, we perform a block factorization into a lower triangular matrix  $\mathcal{L}$ , a diagonal matrix  $D$ , and an upper triangular matrix  $\mathcal{U}$ . The  $\mathcal{LDU}$  decomposition of the system matrix reads

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & \mathbf{C} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{B}_2\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B}_1^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

with the Schur complement  $\mathbf{S} := \mathbf{C} - \mathbf{B}_2\mathbf{A}^{-1}\mathbf{B}_1^T$ . To this end, the preconditioning matrix  $\mathbf{P}$  based on the above factorization is given by

$$\mathbf{P} := \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B}_2 & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B}_1^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}.$$

Expressing the application of the preconditioner as a fixed-point iteration over the index  $k$ , one application of the preconditioner yields

$$\begin{pmatrix} \mathbf{x}^{B,k+1} \\ \mathbf{x}^{S,k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{B,k} \\ \mathbf{x}^{S,k} \end{pmatrix} + \mathbf{P}^{-1} \begin{pmatrix} \mathbf{r}^{B,k} \\ \mathbf{r}^{S,k} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{B,k} \\ \mathbf{x}^{S,k} \end{pmatrix} + \begin{pmatrix} \mathbf{A} & \mathbf{AA}^{-1}\mathbf{B}_1^T \\ \mathbf{B}_2 & \mathbf{B}_2\mathbf{A}^{-1}\mathbf{B}_1^T + \mathbf{S} \end{pmatrix}^{-1} \left[ \begin{pmatrix} \mathbf{b}^B \\ \mathbf{b}^S \end{pmatrix} - \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}^{B,k} \\ \mathbf{x}^{S,k} \end{pmatrix} \right].$$

It is usually executed via a predictor–corrector scheme by first solving an equation related to the beam contribution, afterwards one related to the Schur complement, and lastly a correction step to the beam solution. To this end, after having to form an explicit representation of the Schur complement, a total of three linear systems have to be solved in every application of the preconditioner. The overall algorithm is given as Algorithm 1.

Since an exact version of a block factorization preconditioner is hard to achieve due to its immense computational cost, the following sections are devoted to the construction of an approximate block factorization preconditioner and its application to mixed-dimensional beam-solid interaction.

---

#### Algorithm 1: Full block factorization preconditioner for fiber-solid coupling

---

##### Procedure $\text{Preconditioner}(k_{max})$

// Form explicit Schur complement

$$\mathbf{S} = \mathbf{C} - \mathbf{B}_2\mathbf{A}^{-1}\mathbf{B}_1^T$$

for  $k = 1, \dots, k_{max}$  do

$$\begin{pmatrix} \mathbf{r}^{B,k} \\ \mathbf{r}^{S,k} \end{pmatrix} = \begin{pmatrix} \mathbf{b}^B \\ \mathbf{b}^S \end{pmatrix} - \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}^{B,k} \\ \mathbf{x}^{S,k} \end{pmatrix}$$

// Prediction step: solve for  $\mathbf{x}^{B,k+\frac{1}{2}}$

$$\mathbf{A}\mathbf{x}^{B,k+\frac{1}{2}} = \mathbf{r}^{B,k}$$

// Schur complement step: solve for  $\mathbf{x}^{S,k+1}$

$$\mathbf{S}\mathbf{x}^{S,k+1} = \mathbf{r}^{S,k} - \mathbf{B}_2\mathbf{x}^{B,k+\frac{1}{2}}$$

// Correction step: solve for  $\mathbf{x}^{B,k+1}$

$$\mathbf{A}\mathbf{x}^{B,k+1} = \mathbf{r}^{B,k} - \mathbf{B}_1^T\mathbf{x}^{S,k+1}$$

end

$$\text{return} \begin{pmatrix} \mathbf{x}^{B,k_{max}} \\ \mathbf{x}^{S,k_{max}} \end{pmatrix}$$


---

##### 4.2. Explicit sparse inverse approximation

The first major step of the preconditioner calculation consists of finding an explicit approximation of the Schur complement  $\hat{\mathbf{S}} := \mathbf{C} - \mathbf{B}_2\hat{\mathbf{A}}^{-1}\mathbf{B}_1^T$  with  $\hat{\mathbf{A}}$  denoting an easy-to-invert approximation of  $\mathbf{A}$ . Both quality and computational cost of the preconditioner are mainly governed by the choice  $\hat{\mathbf{A}}$  and  $\hat{\mathbf{S}}$  to approximate  $\mathbf{A}$  and  $\mathbf{S}$ , respectively. In traditional Schur complement based block preconditioners, the inverse  $\hat{\mathbf{A}}^{-1} \approx \mathbf{A}^{-1}$  appearing due to the block factorization and also in the Schur complement calculation itself is often approximated by some diagonal matrix, since the inversion of a diagonal matrix comes at a negligible computational

cost. The most simple approach is to base the inverse approximation on the diagonal part of the  $M \times M$  matrix  $\mathbf{A}$  resulting in  $\hat{\mathbf{A}} = \text{diag}(a_{ii}), i = 1, \dots, M$ . Another well-known approach takes the row sums of  $\mathbf{A}$  [46], reading

$$\hat{\mathbf{A}} = \text{diag} \left( \sum_{j=1}^M |a_{ij}| \right), \quad i = 1, \dots, M.$$

However, such simple diagonal approximations cannot be used in the present scenario since  $\mathbf{A}$  lacks diagonal dominance, cf. Section 3.2. Since  $\mathbf{A}$  resembles the sub-block related to the beam equations, which themselves yield a block diagonal structure of  $\mathbf{A}$ , a more sophisticated explicit approximation scheme of the inverse can be applied taking this particular sparsity structure into account.

Although in general the inverse of the sparse matrix  $\mathbf{A}$  cannot be expected to be sparse as well, explicit SPAIs aim at creating an explicit matrix representation  $\mathbf{A}^*$  of the approximation of the exact inverse  $\mathbf{A}^{-1}$ , that itself is still a sparse matrix. Ideally,  $\text{nnz}(\mathbf{A}^*)$ , the number of non-zeros of  $\mathbf{A}^*$ , does not exceed  $\text{nnz}(\mathbf{A})$ , since a matrix-vector product with  $\mathbf{A}^*$  must be performed at each Krylov iteration [32]. Although incomplete LU factorizations fall into that category, they require considerable effort to parallelize [47].

To take advantage of the block diagonal structure of  $\mathbf{A}$ , we pursue a fully parallelizable approach to construct an explicit sparse approximate inverse  $\mathbf{A}^*$  of the matrix  $\mathbf{A}$  based on the minimization of the Frobenius norm of the residual matrix  $\mathbf{A}\mathbf{A}^* - \mathbf{I}$ , see [32,48]. By choosing an appropriate sparsity pattern  $\mathcal{J}(\mathbf{A}^*)$  for the SPAI  $\mathbf{A}^*$  from the set  $\mathcal{J}$  describing all known patterns, the following least-squares problem needs to be solved:

$$\min_{\mathcal{J}(\mathbf{A}^*) \in \mathcal{J}} \|\mathbf{A}\mathbf{A}^* - \mathbf{I}\|_F \tag{10}$$

We represent the minimization procedure to compute  $\mathbf{A}^*$  by the operation  $\mathbf{A}^* \leftarrow \mathcal{M}(\mathbf{A}, \mathcal{J}(\mathbf{A}^*))$ . To this end, (10) requires to select an appropriate sparsity pattern  $\mathcal{J}(\mathbf{A}^*)$  and a practical approach to the minimization of the Frobenius norm in a distributed memory environment, which we will address in Sections 4.2.1 and 4.2.2, respectively.

#### 4.2.1. Selection of a sparsity pattern for the sparse approximate inverse calculation

The main challenge in (10) is the selection of a sparsity pattern  $\mathcal{J}(\mathbf{A}^*)$  to be used as input into the minimization procedure. An appropriate pattern needs to contain enough information of the inverse by retaining high values, but should also act as a filter to remove small entries in order to reduce fill-in. Straightforward approaches are based on a static sparsity pattern selection and include the choices  $\mathcal{J}(\mathbf{A}^*) := \mathcal{J}(\mathbf{A})$  and  $\mathcal{J}(\mathbf{A}^*) := \mathcal{J}(\mathbf{A}^T)$ , which are easy to obtain, but do not guarantee a good approximation quality. Especially in cases with a partially known sparsity structure of the inverse, e.g., for block-diagonal matrices such as  $\mathbf{A}$  in (8), more advanced static selections are able to deliver a satisfying approximation, yet not requiring dynamic pattern selection approaches as proposed in [32].

In this work, we follow the static pattern selection proposed in [49] and use powers of a sparsified version  $\underline{\mathcal{J}}(\mathbf{A})$  of the graph  $\mathcal{J}(\mathbf{A})$  of the original matrix  $\mathbf{A}$  to obtain an enriched sparsity pattern to be used for the minimization in (10). First,  $\underline{\mathcal{J}}(\mathbf{A})$  is obtained through a thresholding of  $\mathcal{J}(\mathbf{A})$  based on the entries in  $\mathbf{A}$  and using a drop-off tolerance  $\sigma$ . We represent this thresholding by the filter operation

$$\underline{\mathcal{J}}(\mathbf{A}) \leftarrow \mathcal{F}_{\mathcal{J}}(\mathbf{A}, \sigma)$$

delivering individual entries  $\underline{j}(\mathbf{A})_{ij}$  of the filtered graph via

$$\underline{j}(\mathbf{A})_{ij} := \begin{cases} 1 & \text{if } i = j \text{ or } \left| d_i^{-\frac{1}{2}} a_{ij} d_i^{-\frac{1}{2}} \right| > \sigma, \\ 0 & \text{otherwise,} \end{cases} \quad \text{where } d_i := \begin{cases} |a_{ii}| & \text{if } |a_{ii}| > \sigma, \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

The additional Jacobi scaling in (11) simplifies the thresholding and choosing of  $\sigma$  if  $\mathbf{A}$  is poorly scaled. In a second step, a refined sparsity pattern  $\mathcal{J}(\mathbf{A}^\ell)$  is calculated by taking powers  $\ell$  of the sparsified graph  $\underline{\mathcal{J}}(\mathbf{A})$ , reading

$$\mathcal{J}(\mathbf{A}^*) \leftarrow \mathcal{R}(\underline{\mathcal{J}}(\mathbf{A}), \ell).$$

The matrix  $\mathbf{A}^\ell$  is never calculated explicitly, but the powers are directly computed on its sparsified graph  $\underline{\mathcal{J}}(\mathbf{A})$ .

#### 4.2.2. Evaluation of the Frobenius norm on a parallel computer

Due to the 2-norm compatibility of the Frobenius norm, the problem can be decoupled into a sum of Euclidian norms, reading

$$\min_{\mathcal{J}(\mathbf{A}^*) \in \mathcal{J}} \|\mathbf{A}\mathbf{A}^* - \mathbf{I}\|_F = \sum_{i=1}^M \min_{\mathcal{J}(\mathbf{A}^*) \in \mathcal{J}} \left\| (\mathbf{A}\mathbf{A}^* - \mathbf{I}) \mathbf{e}_i \right\|_2^2 = \sum_{i=1}^M \min_{\mathcal{J}(\mathbf{A}^*) \in \mathcal{J}_i} \left\| \mathbf{A}\mathbf{A}_i^* - \mathbf{e}_i \right\|_2^2,$$

which can be solved independently for each row  $i = 1, \dots, M$  of  $\mathbf{A}^*$ . Since the matrix  $\mathbf{A}$  is usually stored in a row-wise distribution, where each parallel process stores a subset of all rows of  $\mathbf{A}$ , this decomposition renders the method, besides an initial communication step, inherently parallel. To this end, calculating a row of the SPAI means solving a small least-squares problem by applying a dense QR factorization.



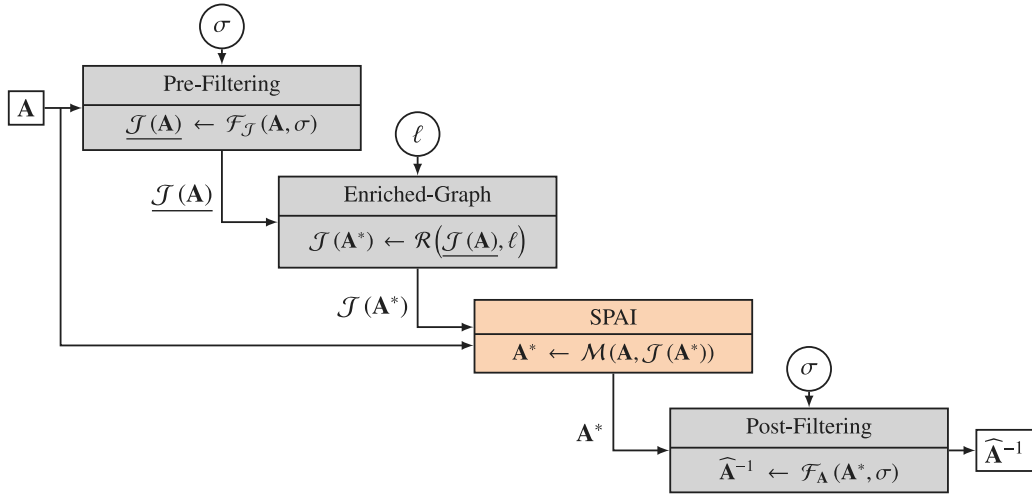


Fig. 2. Optional (gray) and mandatory (orange) steps of the SPAI computation and its flow of information with computed data (in boxes) and user parameters (in circles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 4.2.3. Practical algorithm

In practice, one usually does not solve (10) directly, but rather combines it with some pre- and post-operations, foremost the graph computation from Section 4.2.1 and the handling of the Frobenius norm outlined in Section 4.2.2. In addition, it appears beneficial to perform a post-filtering of  $A^*$  by dropping all entries with  $|a_{ij}^*| < \sigma \quad \forall i, j = 1, \dots, M$  to further reduce fill-in of  $\hat{A}^{-1}$  and, thus, the cost of applying the preconditioner [49]. The post-filtering will be denoted by the operator  $F_A(A^*, \sigma)$ .

We summarize all the steps to compute the sparse approximate inverse  $A^*$  in Fig. 2. Therein, optional steps are marked in gray, while the mandatory computation of  $A^*$  is highlighted in orange. User-given data to configure the individual steps is depicted in circles, while computed input and output data is put into rectangular boxes. The arrows indicate the flow of data between the individual steps.

Even though the method can be steered quite effectively by the refinement level  $\ell$  and threshold tolerance  $\sigma$ , there are still several problems to avoid. The method can still produce rather dense matrices with poor approximation quality. Depending on the threshold parameter  $\sigma$ , an aggressive dropping of values might result in a loss of information, which again results in a poor result and the approximation of the inverse might even be singular.

### 4.3. One-level approach for the predictor and corrector step

As outlined in Algorithm 1, the first and last linear equation to be solved during the application of the block preconditioner resemble the predictor step

$$A \mathbf{x}^{B,k+\frac{1}{2}} = \mathbf{r}^{B,k} \tag{12}$$

and the corrector step

$$A \mathbf{x}^{B,k+1} = \mathbf{r}^{B,k} - \mathbf{B}_1^T \mathbf{x}^{S,k+1}, \tag{13}$$

respectively. Both use the beam matrix  $A$  to update the beam unknowns. Following the assumptions made in Section 3.2, traditional smoothing approaches are not applicable to approximate the solutions of the linear systems in (12) and (13) without a major loss in convergence properties.

Based on the explicit SPAI calculation for the approximation of the Schur complement, a rather good approximation  $\hat{A}^{-1}$  is already available. Following [50], the sparse approximate inverse is reused as a smoother by applying the fixed-point iteration over index  $m$  to (12), reading

$$\mathbf{x}^{B,k+\frac{1}{2},m+1} = \mathbf{x}^{B,k+\frac{1}{2},m} + \hat{A}^{-1} \mathbf{r}^{B,k}, \tag{14}$$

and to (13), reading

$$\mathbf{x}^{B,k+1,m+1} = \mathbf{x}^{B,k+1,m} + \hat{A}^{-1} (\mathbf{r}^{B,k} - \mathbf{B}_1^T \mathbf{x}^{S,k+1}). \tag{15}$$

A comparison to more traditional smoothers, e.g., Jacobi and Gauß–Seidel methods, can also be found in the original publication [50].

**Remark 4.1** (*Challenges for Multilevel Methods*). Expanding the solution procedure of the predictor and corrector steps to an AMG approach to solve the beam-related equation is challenging, as there are still several open questions around the construction of AMG hierarchies for beam models: The beams are represented as 1D elements, for which a suitable coarsening scheme to form aggregates for coarser multigrid (MG) levels is still part of active research. Even with appropriate coarsening, fibers discretized with only a few elements would quickly form single-node aggregates being suboptimal for the restrictor and prolongator construction as well as effectively stalling the coarsening process. The addition of rotational components into the beam equation introduces another difficulty, as nodes with a different number of DOFs exist, hindering the application of conventional aggregation strategies. To project and, thus, treat important error modes correctly on coarser levels, a proper near nullspace has to be constructed, which strongly depends on the underlying beam formulation. For the use cases presented in this manuscript, which mostly feature a small number of beam elements used to discretize a fiber, the SPAI smoother from (14) and (15) proved to be very competitive in terms of approximation quality of the inverse, computational efficiency, robustness, and weak scaling behavior, see Section 5.

#### 4.4. Multilevel approach for the Schur complement step

The computationally most demanding part of the preconditioning algorithm involves the solution procedure for the Schur complement equation given as

$$\mathbf{S}\mathbf{x}^{S,k+1} = \mathbf{r}^{S,k} - \mathbf{B}_2\mathbf{x}^{B,k+\frac{1}{2}}.$$

Due to the explicit sparse approximation of the inverse of  $\mathbf{A}$  used to form the approximate Schur complement  $\hat{\mathbf{S}} := \mathbf{C} - \mathbf{B}_2\hat{\mathbf{A}}^{-1}\mathbf{B}_1^T$ , the resulting matrix  $\hat{\mathbf{S}}$  is rather dense compared to calculations using one of the diagonal approximation approaches for  $\mathbf{A}$  given in Section 4.1. The inverse of the Schur complement is approximated by a standard aggregation-based AMG method. As level smoother, a one-level domain decomposition with overlap  $\delta$  and an incomplete LU factorization with fill-in  $p$  and thresholding  $\tau$  of small entries is applied, often abbreviated as ILUT [51]. To accelerate convergence, a smoothing of the tentative prolongation operator basis functions can be done, also known as smoothed-aggregation algebraic multigrid (SA-AMG) [52]. This however leads to a higher fill-in of the coarse system matrix representations increasing the computational cost of the preconditioner setup. Especially the Galerkin product for the calculation of the coarse level operator and the incomplete LU factorization smoother are negatively influenced by the additional fill-in. Furthermore, the operator smoothing is classically based on a Jacobi method, which heavily relies on diagonal dominance. Thus, in some cases it makes sense to skip the prolongator smoothing and take advantage of the robustness of plain-aggregation algebraic multigrid (PA-AMG) by applying aggregate-wise constant basis functions [53]. Caused by the additional mixed-dimensional coupling terms in the bulk field's matrix, a similar increase in the fill-in of the coarse level operators of a Ruge–Stüben AMG method has been observed in [31].

#### 4.5. Approximate block factorization preconditioner for beam-solid interaction

The components described in Sections 4.1–4.4 are now put together to tailor the block preconditioner from Algorithm 1 to the specifics of mixed-dimensional beam-solid interaction, yielding the final preconditioner summarized in Algorithm 2.

In a pre-computation step, an explicit representation  $\hat{\mathbf{A}}^{-1}$  of the SPAI of the beam matrix  $\mathbf{A}$  is formed and also used to compute an approximation  $\hat{\mathbf{S}}$  to the Schur complement. Based on the sweep index  $k$  of the preconditioner, the main computation loop consists of the three steps: First, we predict the beam unknowns  $\mathbf{x}^{B,k+\frac{1}{2}}$  by using the SPAI as a smoother. Then, we solve the Schur complement equation for the solid unknowns  $\mathbf{x}^{S,k+1}$  using an AMG method. Finally, we again use the SPAI as a smoother to correct the beam unknowns to their final values  $\mathbf{x}^{B,k+1}$ .

In terms of computational effort, the computation of the SPAI as outlined in Section 4.2 comes at a certain cost, however is perfectly parallelizable and is used at three steps in Algorithm 2: once in the approximation of the Schur complement and twice to update the beam solution in the predictor and corrector step. A single application of SPAI as a smoother for the predictor or corrector step boils down to a sparse matrix–vector multiplication.

#### 4.6. Comparison to existing methods in literature

Having discussed all the details of our proposed preconditioner, we now want to outline its commonalities and differences compared to preconditioners available in the literature. In particular, we will discuss the work in [25–27,30,31].

In contrast to our work, the preconditioners from [25–27,30] are all tailored to saddle point systems. In the construction of a block diagonal preconditioner, they use only the diagonal part of an  $\mathcal{LDU}$  factorization of the original matrix. In [26,27,30], the arising Schur complement is approximated by a spectrally equivalent fractional Laplacian. AMG is used to tackle matrices arising from the 3D bulk field, while the embedded 1D domains are always handled by a direct solver.

The case of  $2 \times 2$  systems as they arise for example from a penalty regularization is covered in [31]. Due to the low cost of inverting the matrix of the 1D domain, they use an exact factorization to represent the Schur complement.

Similar to [31], we base the construction of the Schur complement on the original block matrix, however avoid the exact inversion of the beam sub-block  $\mathbf{A}$  and rather use its SPAI  $\hat{\mathbf{A}}^{-1}$ . In line with all available preconditioners, the matrix block associated with

**Algorithm 2:** Approximate block factorization preconditioner for fiber-solid coupling

---

```

Procedure Preconditioner( $k_{max}$ )
  // Pre-compute SPAI of  $\mathbf{A}$ 
   $\hat{\mathbf{A}}^{-1} \leftarrow \mathcal{F}_{\mathbf{A}}(\mathcal{M}(\mathbf{A}, \mathcal{R}(\mathcal{F}_{\mathcal{J}}(\mathbf{A}, \sigma), \ell)), \sigma)$ 
  // Form explicit, approximate Schur complement
   $\hat{\mathbf{S}} = \mathbf{C} - \mathbf{B}_2 \hat{\mathbf{A}}^{-1} \mathbf{B}_1^T$ 
  for  $k = 1, \dots, k_{max}$  do
     $\begin{pmatrix} \mathbf{r}^{B,k} \\ \mathbf{r}^{S,k} \end{pmatrix} = \begin{pmatrix} \mathbf{b}^B \\ \mathbf{b}^S \end{pmatrix} - \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}^{B,k} \\ \mathbf{x}^{S,k} \end{pmatrix}$ 
    // Prediction step: solve for  $\mathbf{x}^{B,k+\frac{1}{2}}$  with SPAI smoother
    for  $m = 1 \dots m_{max}$  do
       $\mathbf{x}^{B,k+\frac{1}{2},m+1} = \mathbf{x}^{B,k+\frac{1}{2},m} + \hat{\mathbf{A}}^{-1} \mathbf{r}^{B,k}$ 
    end
    // Schur complement step: solve for  $\mathbf{x}^{S,k+1}$  with AMG
     $\hat{\mathbf{S}} \mathbf{x}^{S,k+1} = \mathbf{r}^{S,k} - \mathbf{B}_2 \mathbf{x}^{B,k+\frac{1}{2}}$ 
    // Correction step: solve for  $\mathbf{x}^{B,k+1}$  with SPAI smoother
    for  $m = 1 \dots m_{max}$  do
       $\mathbf{x}^{B,k+1,m+1} = \mathbf{x}^{B,k+1,m} + \hat{\mathbf{A}}^{-1} (\mathbf{r}^{B,k} - \mathbf{B}_1^T \mathbf{x}^{S,k+1})$ 
    end
  end
  return  $\begin{pmatrix} \mathbf{x}^{B,k_{max}} \\ \mathbf{x}^{S,k_{max}} \end{pmatrix}$ 

```

---

the 3D bulk discretization is tackled by an AMG method. In our work though, the matrix arising from the 1D discretization is never explicitly inverted or factorized, but also treated in an approximate fashion (i.e., using the SPAI as a smoother, cf. Section 4.3) to facilitate large numbers of embedded fibers as well as finely resolved fiber discretizations. Regarding the underlying physical problems and applications, existing work is concerned with transport problems on the 1D domain, while our work is the first preconditioner tailored to multi-dimensional partial differential equations (PDEs) on the 1D domain, in particular geometrically exact beam models with up to nine DOFs per mesh node depending on the actual beam model at hand.

## 5. Numerical experiments

We present numerical examples to illustrate the influence of the different algorithmic parameters of the SPAI computation proposed in Section 4.2, to study and demonstrate the weak scaling behavior of the proposed preconditioner, to investigate the robustness of the proposed method regarding material parameters and geometric properties and to showcase its applicability to practical problems in civil engineering. All computations are done with our in-house multi-physics code 4C [54], which is built upon the TRILINOS project [55,56]. All preconditioning operations are done through the multigrid package MUELU [33] and its dependencies within the TRILINOS project. For the generation of the beam geometries, we rely on MESHpy [57].

### 5.1. Numerical study of the sparse approximate inverse calculation for the beam sub-block

A major component of a Schur complement based preconditioner is the approximation of the inverse appearing in the Schur complement calculation itself. In the presented approach, the combination of the drop-off tolerance  $\sigma$  of small values and the allowed fill-in (indirectly described by the refinement level  $\ell$ ) during the sparse approximate inverse calculation have a great influence on the quality of the sparse approximate inverse  $\mathbf{A}^*$  and, thus, on the convergence behavior of the block preconditioning method. In the following, we investigate test cases with different sparsity patterns of the sub-matrix  $\mathbf{A}$  and study the impact of different choices of both  $\sigma$  and  $\ell$  on the quality of the sparse approximate inverse as well as on the convergence behavior of the preconditioned linear solver.

To this end, we consider a simple 3D beam-solid interaction problem as shown in Fig. 3: A solid cube with edge length  $l^S = 1$  m is filled with randomly placed straight fibers, such that the beam-solid volume ratio is  $V^B/V^S \approx 0.2\%$  and that the fibers do not stick out of the solid volume. The cube is clamped at its bottom surface and loaded with a constant tensile load of  $q = 1$  N/m<sup>2</sup> at its top face. The solid is modeled by a St.-Venant–Kirchhoff material (Young’s modulus  $E^S = 1$  N/m<sup>2</sup>, Poisson’s ratio  $\nu^S = 0.3$ ) and discretized by first-order hexahedral finite elements. The fibers are represented by either torsion-free Kirchhoff–Love beams (TF) or Simo–Reissner beam elements (SR) using the following parameters: Young’s modulus  $E^B = 10$  N/m<sup>2</sup>, radius  $R = 0.005$  m and length  $l^B = 0.25$  m. In addition, a Poisson’s ratio of  $\nu^B = 0.0$  is used in the cases with SR beam models. The coupling conditions are enforced with penalty parameters  $\epsilon^V = 10$  N/m<sup>2</sup> and  $\epsilon^R = 10$  Nm/m, if applicable.

For the calculation of the sparse approximate inverse, the matrix block  $\mathbf{A}$  describing the contribution of the fibers is of particular interest. Therefore, six test cases with different beam formulations and varying number of beam finite elements per fiber are set up

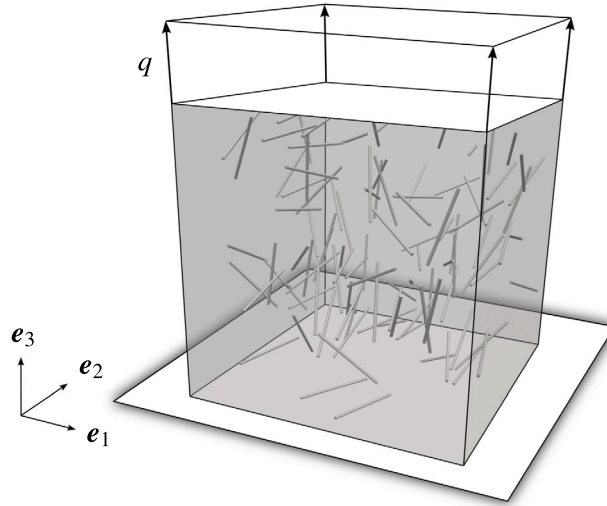


Fig. 3. Geometry and setup for the numerical study of the sparse approximate inverse calculation: a solid cube with edge length  $l^S = 1$  m is randomly filled with fibers of the same length  $l^b = 0.25$  m, clamped at its bottom and loaded with a distributed external load  $q = 1$  N/m<sup>2</sup>.

Table 2

Matrix size of  $A$  and number of non-zeros of the graph  $J(A)$  for the six different test cases.

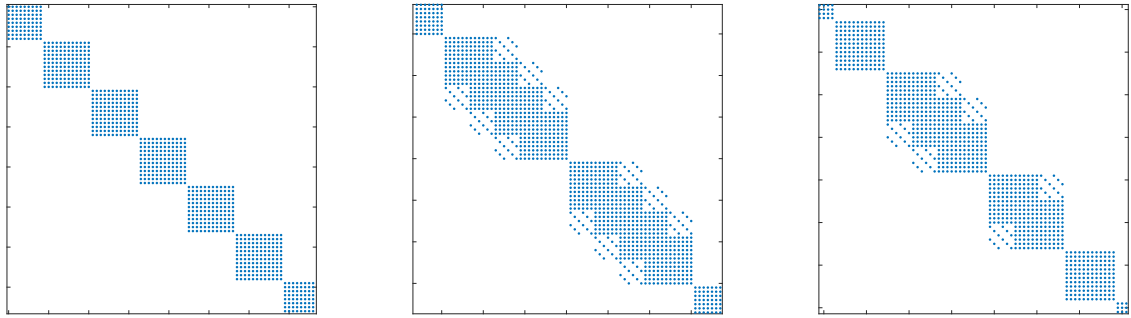
	Beam model	Size( $A$ )	nnz( $J(A)$ )
Test case I	TF	1224 × 1224	14 688
Test case II	TF	3060 × 3060	55 077
Test case III	TF	1854 × 1854	28 546
Test case IV	SR	2142 × 2142	44 982
Test case V	SR	5814 × 5814	184 518
Test case VI	SR	3402 × 3402	92 862

to trigger different sparsity patterns  $J(A)$ . In the test cases I and IV, the fibers are discretized by just one beam element, resulting in a block-diagonal matrix with fully populated sub-blocks. For test cases II and V, four beam elements are used per fiber, resulting in bigger and sparser sub-blocks. Test cases III and VI mix the other scenarios by randomly using between one and four beam elements per fiber. All test cases and their respective matrix sizes and number of non-zeros of  $J(A)$  are summarized in Table 2. The resulting sparsity patterns are illustrated exemplarily for the test cases I–III in Fig. 4.

The overall simulation is of quasi-static nature and imposes the total load over the course of two load steps, which is sufficient for investigating the key features of the linear solver like the iteration count and setup/solve timings. The nonlinear solver converges, if the nonlinear residual  $\|f\|_2$  drops below  $10^{-6}$  and if the full displacement increment  $\|\Delta d\|_2$  is smaller than  $10^{-8}$ . In each nonlinear iteration, a linear system is solved using a preconditioned GMRES method [58]. The preconditioner is configured as follows: the number of sweeps through the preconditioner is set to  $k = 1$ , and the number of iterations for the SPAI smoother is chosen as  $m = 1$ . The Schur complement equation is solved with a SA-AMG scheme with an ILUT smoother with overlap  $\delta = 1$ , fill-in level  $p = 1$  and drop-off tolerance  $\tau = 10^{-4}$ . The linear solver is assumed to be converged if the relative residual  $\|r^n\|_2 / \|r^0\|_2$  falls below  $10^{-8}$ . All simulations are done in serial on a single processor.

Using SPAIs, the convergence of the linear solver is tightly related to the parameters chosen for the SPAI calculation. In Table 3, different combinations of the drop-off tolerance  $\sigma$  and refinement level  $\ell$  are given for each of the six test cases. Each value pair represents the largest possible drop-off value  $\sigma$  possible for a fixed refinement level  $\ell$ , such that the linear solver still converges. For certain values of  $\ell$ , no convergence could be achieved at all for some test cases, even with a very small drop-off tolerance  $\sigma$ . In these situations, an appropriate value of  $\sigma$  is chosen, such that only explicit zero values are dropped from the matrix graph to still be able to make a fair comparison with the other test cases regarding the error norm and number of non-zero entries in the approximate inverse. In addition, the number of non-zero entries of the filtered graph  $\underline{J}(A)$  used as starting point for the sparsity pattern construction as well as the number of non-zero entries of the graph of the inverse approximation  $J(A^*)$  are given. The number of non-zeros for the unfiltered graph,  $\text{nnz}(J(A))$ , is illustrated in Table 2 for comparison. The behavior of the iterative linear solver is assessed by the averaged number of iterations per nonlinear solver step and three timings concerning the preconditioner setup time  $T_{\text{setup}}$ , the time spent for solving the linear system  $T_{\text{solve}}$ , and the total time  $T_{\text{total}} = T_{\text{setup}} + T_{\text{solve}}$ . The overall quality of the inverse approximation  $A^*$  is quantified by the error norm relative to the exact inverse  $A^{-1}$ .

In test case I, the dense sub-blocks of the block-diagonal sparsity pattern of  $A$  already lead to the exact graph of the inverse, which makes the approximation rather simple. For the given parameter combinations the upper bound of number of non-zeros to



(a) Test case I: one beam element per fiber (b) Test case II: four beam elements per fiber (c) Test case III: random number (1, 2, 3 or 4) of beam elements per fiber

Fig. 4. Partial visualization of the sparsity structure  $J(A)$  of matrix  $A$  for test cases I-III.

Table 3

Comparison of the influence of different values of  $\sigma$  and  $\ell$  on the number of linear solver iterations and timings.

	$\sigma$	$\ell$	$\text{nnz}(J(A))$	$\text{nnz}(J(A^*))$	#iter	CPU time (s)			rel. error norm $\frac{\ A^* - A^{-1}\ _F}{\ A^{-1}\ _F}$
						$T_{\text{setup}}$	$T_{\text{solve}}$	$T_{\text{total}}$	
Test case I	$10^{-10}$	1	14 572	14 572	10	3.390	0.755	4.145	$2.100 \cdot 10^{-3}$
	$10^{-8}$	2	11 576	14 688	10	3.338	0.754	4.092	$1.248 \cdot 10^{-12}$
	$10^{-6}$	3	9552	14 688	10	3.464	0.750	4.214	$1.248 \cdot 10^{-12}$
Test case II	$10^{-11}$	1	44 335	44 335	–	–	–	–	$7.653 \cdot 10^{-1}$
	$10^{-10}$	2	37 222	86 861	10	3.863	0.781	4.644	$6.000 \cdot 10^{-12}$
	$10^{-9}$	3	32 436	91 592	10	3.732	0.800	4.532	$1.981 \cdot 10^{-14}$
Test case III	$10^{-12}$	1	26 966	26 966	20	3.559	1.463	5.022	$1.415 \cdot 10^{-1}$
	$10^{-10}$	2	23 919	35 961	10	3.859	0.819	4.678	$3.433 \cdot 10^{-13}$
	$10^{-8}$	3	17 544	35 814	10	3.830	0.839	4.669	$3.433 \cdot 10^{-13}$
Test case IV	$10^{-12}$	1	26 662	26 662	–	–	–	–	$9.277 \cdot 10^{-1}$
	$10^{-11}$	2	26 614	44 978	9	3.901	0.719	4.620	$1.082 \cdot 10^{-12}$
	$10^{-7}$	3	16 748	42 641	9	4.100	0.761	4.861	$1.081 \cdot 10^{-12}$
Test case V	$10^{-12}$	1	92 731	92 731	–	–	–	–	$9.519 \cdot 10^{-1}$
	$10^{-12}$	2	92 731	249 034	–	–	–	–	$1.528 \cdot 10^{-1}$
	$10^{-11}$	3	87 453	331 152	9	5.172	0.834	6.006	$1.954 \cdot 10^{-7}$
Test case VI	$10^{-12}$	1	49 614	49 614	–	–	–	–	$9.530 \cdot 10^{-1}$
	$10^{-12}$	2	49 614	109 705	–	–	–	–	$5.970 \cdot 10^{-2}$
	$10^{-9}$	3	38 420	121 400	10	4.187	0.794	4.981	$3.829 \cdot 10^{-12}$

exactly compute  $A^{-1}$  is quickly reached resulting in low iteration counts and error norms. For the given problem, the setup timings are nearly identical, with only the solver timings for the first parameter combination taking a bit longer due to a not fully populated sparsity pattern.

The second test case is based on rather sparse sub-blocks of bigger size compared to test case I. Without a pattern refinement, the linear solver did not converge as the approximation quality of the inverse is not sufficient. For higher refinement levels, the number of non-zeros in  $J(A^*)$  quickly increases resulting in convergence of the method with still acceptable timings. Yet, there is not a lot of flexibility in choosing the drop-off tolerance  $\sigma$  to still retain convergence.

As test case III is a combination of the first two problems, the resulting behavior is a mix of these. Using the initial sparsity graph for the inverse approximation results in 20 linear solver iterations until convergence, which explains the high solving time  $T_{\text{solve}}$ . On the other hand, the setup time  $T_{\text{setup}}$  is comparable to the other tests. For increased refinement levels  $\ell$ , a similar behavior as in test cases I and II is observed.

For test case IV, the beam elements are switched to a Simo–Reissner formulation, which contributes additional rotational degrees of freedom into  $A$ . In contrast to test case I, this results in already sparse sub-blocks for using one beam element per fiber. Therefore, using just the graph  $J(A)$  results in a poor approximation of the sparsity pattern of the inverse and, thus, leads to no convergence. Using higher refinement levels  $\ell$  to enrich the input graph for the SPAI computation quickly heals this problem and even allows one to work with a more aggressive dropping scheme, i.e., using larger values for  $\sigma$ .

Test cases V and VI show a similar behavior and only converge with  $\ell = 3$ . The additional beam elements used per fiber increase the block size of each sub-block and thus leave more room for possible sparse approximations for the inverse. The static approach

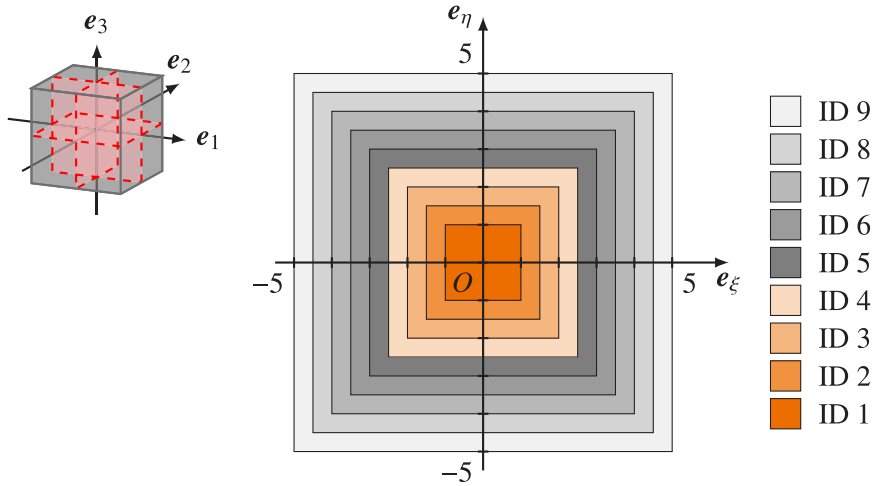


Fig. 5. Intersections of all cubes (IDs 1–9) of the weak scaling studies with planes spanned by basis vectors  $e_\xi$  and  $e_\eta$ ,  $\xi, \eta \in \{1, 2, 3\}$ ,  $\xi \neq \eta$  in the cartesian frame of reference. Orientation of the cutting planes is sketched in the top left.

**Table 4**  
Mesh refinement schedule for the weak scaling study.

ID	$n^{\text{proc}}$	$n_{\text{DOF}}^S$	$n_{\text{DOF}}^B$	$n_{\text{DOF}}^{\text{total}}$	$n_{\text{DOF}/\text{proc}}^{\text{total}}$
1	8	397 953	9132	407 085	50 885.6
2	27	1 316 928	31 824	1 348 752	49 953.8
3	64	3 090 903	74 916	3 165 819	49 465.9
4	125	6 001 128	146 724	6 147 852	49 182.8
5	216	10 328 853	257 448	10 586 301	49 010.7
6	343	16 355 328	413 976	16 769 304	48 890.1
7	512	24 361 803	621 876	24 983 679	48 796.2
8	729	34 629 528	892 932	35 522 460	48 727.7
9	1000	47 439 753	1 222 320	48 662 073	48 662.1

for choosing an appropriate sparsity pattern for the inverse presented in Section 4.2.3 is still able to produce good approximations, thus leading to convergence of the linear solver, but only for rather dense representations of  $\mathbf{A}^*$ .

In conclusion, a robust parameter combination is highly problem dependent and necessary to achieve convergence of the linear solver. In the presented cases, a refinement level of  $\ell = 3$  and a drop-off tolerance of  $\sigma = 10^{-9}$  to get rid of small values polluting the sparsity pattern showed to be sufficient to enable convergence even for challenging cases.

## 5.2. Weak scaling behavior

To study the performance of the proposed block preconditioner also for large-scale examples and on parallel computing clusters, we now conduct a weak scaling study. The problem setup is similar to test case I from Section 5.1. To guarantee that large problems exhibit the same fiber distribution at least in parts of the domain, the meshes are setup as follows: We first create the geometry and mesh for the largest problem by placing a cube with edge length 10 m inside a cartesian frame of reference, such that the cube's center of mass coincides with the origin  $O$  and its edges are oriented along the cartesian axes. Then, the cube is filled with randomly positioned and oriented straight fibers with length  $l^B = 0.25$  m and radius  $R = 0.005$  m. Only fibers, which are fully contained in the cube, are considered. This problem will be solved on 1000 MPI ranks. For smaller problems, the geometry is cut out of this initial cube. Also for the cut out problem, only fully contained fibers are considered. Fig. 5 shows the intersections of the series of cubes with each coordinate plane spanned by basis vectors  $e_\xi$  and  $e_\eta$ ,  $\xi, \eta \in \{1, 2, 3\}$ ,  $\xi \neq \eta$ . This process not only yields an almost constant beam-solid volume ratio for all problem sizes, but also guarantees that larger meshes are just extensions of the smaller meshes. The load per processor is kept constant at around 50k degrees of freedom. Meshing details are given in Table 4.

The boundary conditions are identical to test case I from Section 5.1, i.e., the bottom surface is clamped and the top surface loaded with a constant tensile load of  $q = 1 \text{ N/m}^2$ . The solid is again modeled by a St.-Venant-Kirchhoff material (Young's modulus  $E^S = 1 \text{ N/m}^2$ , Poisson's ratio  $\nu^S = 0.3$ ) and discretized with first-order hexahedral finite elements. The fibers are modeled using torsion-free Kirchhoff-Love beam elements with Young's modulus  $E^B = 10 \text{ N/m}^2$ . The positional coupling condition is enforced with a penalty parameter of  $e^\nu = 10 \text{ N/m}^2$ .

The overall simulation is again of quasi-static nature and imposes the total load over the course of two load steps. The nonlinear solver converges, if the nonlinear residual  $\|\mathbf{f}\|_2$  drops below  $10^{-6}$  and the full displacement increment  $\|\Delta \mathbf{d}\|_2$  is smaller than  $10^{-8}$ . For the solution of the linear system arising in each nonlinear iteration, a preconditioned GMRES method is applied with the

proposed preconditioner. Hereby, the parameters are set as follows: to increase the robustness of the smoothers in the parallel setting, the number of sweeps through the preconditioner is changed to  $k = 3$  as well as the number of iterations for the SPAI smoother to  $m = 3$  (in contrast to the parameter choice in Section 5.1). The SPAI computation for the beam sub-matrix  $A$  uses a drop tolerance  $\sigma = 10^{-8}$  and a refinement level  $\ell = 2$  to enrich the sparsity pattern. This choice is inspired by the results of the test case I in Section 5.1. Due to the inherently parallel nature of the SPAI computation (cf. Section 4.2.2), it has a marginal influence on the weak scalability compared to the other components of the overall preconditioner, especially the AMG components. Therefore this part of the algorithm is kept constant without parameter variation. The Schur complement equation is solved with an aggregation-based AMG method. Coarsening is performed until the number of unknowns on the coarsest level drops below 6500. The AMG hierarchy is traversed using a V-cycle with level transfer operators arising from either PA-AMG or SA-AMG. On all but the coarsest level, the level smoother is chosen as ILUT with overlap  $\delta = 1$ , fill-in level  $p = 1$  and drop-off tolerance  $\tau = 10^{-4}$ . The coarse level is solved with a direct solver using the distributed memory version of SUPERLU [59]. The outer GMRES solver is assumed to be converged if the relative residual  $\|r^n\|_2 / \|r^0\|_2$  falls below  $10^{-8}$ .

The scaling study is run on our in-house cluster (16 nodes with 2x Intel Xeon Cascade Lake CPUs with 26 cores, 20 nodes with 2x Intel Xeon Skylake CPUs with 24 cores, 1312 cores in total, Mellanox Infiniband Interconnect). The overall weak scaling performance is quantified by the averaged number of iterations per nonlinear Newton iteration and the timings for setting up the preconditioner  $T_{\text{setup}}$ , for solving the linear system  $T_{\text{solve}}$ , and the total solver time  $T_{\text{total}} = T_{\text{setup}} + T_{\text{solve}}$ . Since the setup of the preconditioner is expected to be expensive, we also examine the option of reusing the preconditioner throughout all Newton steps of a load step with the aim to reduce  $T_{\text{setup}}$  and, thus, also reduce  $T_{\text{total}}$ .

Fig. 6 summarizes the results of the weak scaling study. We first discuss the case where the preconditioner is built in every Newton step. Looking at the iteration counts in Fig. 6(a), the SA-AMG method delivers iteration counts independent of the problem size (with mostly 13 iterations per solve), while PA-AMG exhibits an increase in iterations by a factor of 2x (from 26 to 52 iterations) when increasing the problem size by 120x, i.e., from mesh ID 1 to mesh ID 9, cf. Table 4. Regarding the setup time  $T_{\text{setup}}$  required to build the preconditioner shown in Fig. 6(b), PA-AMG is more than twice as fast as SA-AMG due to the smaller support of PA-AMG interpolation functions and, thus, less fill-in in coarse level operators. In contrast, the time to solve the linear system is more than 2x smaller for SA-AMG due to the better approximation properties of smoothed interpolation functions in SA-AMG, cf. Fig. 6(c). When looking at the combined time  $T_{\text{total}} = T_{\text{setup}} + T_{\text{solve}}$  as shown in Fig. 6(d), both types of transfer operators result in very similar timings.

With a moderate increase of  $T_{\text{setup}}$  for PA-AMG for an increasing number of parallel processes, the increase in solver time  $T_{\text{solve}}$  as well as total time  $T_{\text{total}}$  of the PA-AMG scheme appears to be directly linked to the number of iterations required to achieve the desired tolerance of the iterative linear solver. In contrast, SA-AMG requires a rather constant number of iterations for all problem sizes and spends most of its time in the preconditioner setup, thus resulting in total solver timings  $T_{\text{total}}$  that are dominated by the preconditioner setup time. This hints at potential savings when setting up the preconditioner once and then reusing it to solve multiple subsequent linear systems, e.g., through the course of a Newton scheme.

We now look at the option of building the preconditioner only in the first Newton step and then reusing it throughout an entire load step. In the present study, each load step requires two Newton steps, hence we expect to save 50% of  $T_{\text{setup}}$  and hope to not worsen in terms of iteration numbers and solver time  $T_{\text{solve}}$ . When looking at the setup time in Fig. 6(b), setup costs for both SA-AMG and PA-AMG are reduced by  $\approx 50\%$  as expected. Furthermore, the iteration numbers as well as the solver time  $T_{\text{solve}}$  stay nearly the same for both SA-AMG and PA-AMG, cf. Figs. 6(a) and 6(c). Overall, the reuse of the preconditioner positively impacts the total solver time  $T_{\text{total}}$  with the best option being SA-AMG with reuse of the preconditioner, which appears to be roughly 30% faster than the variant without reuse of the preconditioner. We note that the actual benefit of reusing the preconditioner depends on the number of nonlinear solver iterations per load step: The more Newton steps are required, the greater savings are to be expected from reusing the preconditioner.

In conclusion, this example has shown weak scalability of the proposed preconditioner. The iteration numbers remain perfectly constant for SA-AMG. Due to the overlap  $\delta = 1$  of the ILUT smoother in the MG hierarchy, the setup time  $T_{\text{setup}}$  increases by a factor of  $\approx 5x$  when increasing the problem size by 120x, i.e., from mesh ID 1 to mesh ID 9, cf. Table 4. We stress that the choice of transfer operators has a great influence on the total weak scalability of the preconditioner. In the presented test cases, the best scalability in terms of the iteration count were obtained by SA-AMG transfer operators and when reusing the preconditioner though all Newton steps of a load step. In sum, SA-AMG appears as the method of choice to demonstrate weak scalability and keep the iteration count low. For more complex application scenarios however, it might be beneficial to fall back to PA-AMG due to its reduced fill-in during coarsening. Given the intricate nature of beam-solid applications and their arising systems of linear equations, we deem the present weak scaling behavior acceptable and adequate.

### 5.3. Robustness of the preconditioner under varying physical parameters

To assess the preconditioner's robustness, we now study its behavior w.r.t. iteration numbers of the linear solver under changes of critical physical parameters. Therefore, a composite plate with four fiber layers is considered, where the beam-solid stiffness ratio as well as the beam radius are varied. This allows to cover a wide range of possible parameter combinations for the coupled problem.

The geometrical setup is identical to the composite plate presented in our prior work [1] with a length of 2 m, a width of 1 m and a total thickness of  $t = 0.04$  m, where two layers are oriented in  $45^\circ$  and  $-45^\circ$  angles, respectively. The solid bulk domain is modeled as St.-Venant–Kirchhoff material with fixed constitutive properties ( $E^S = 1$  GPa,  $\nu^S = 0.3$ ). The embedded fibers are modeled as

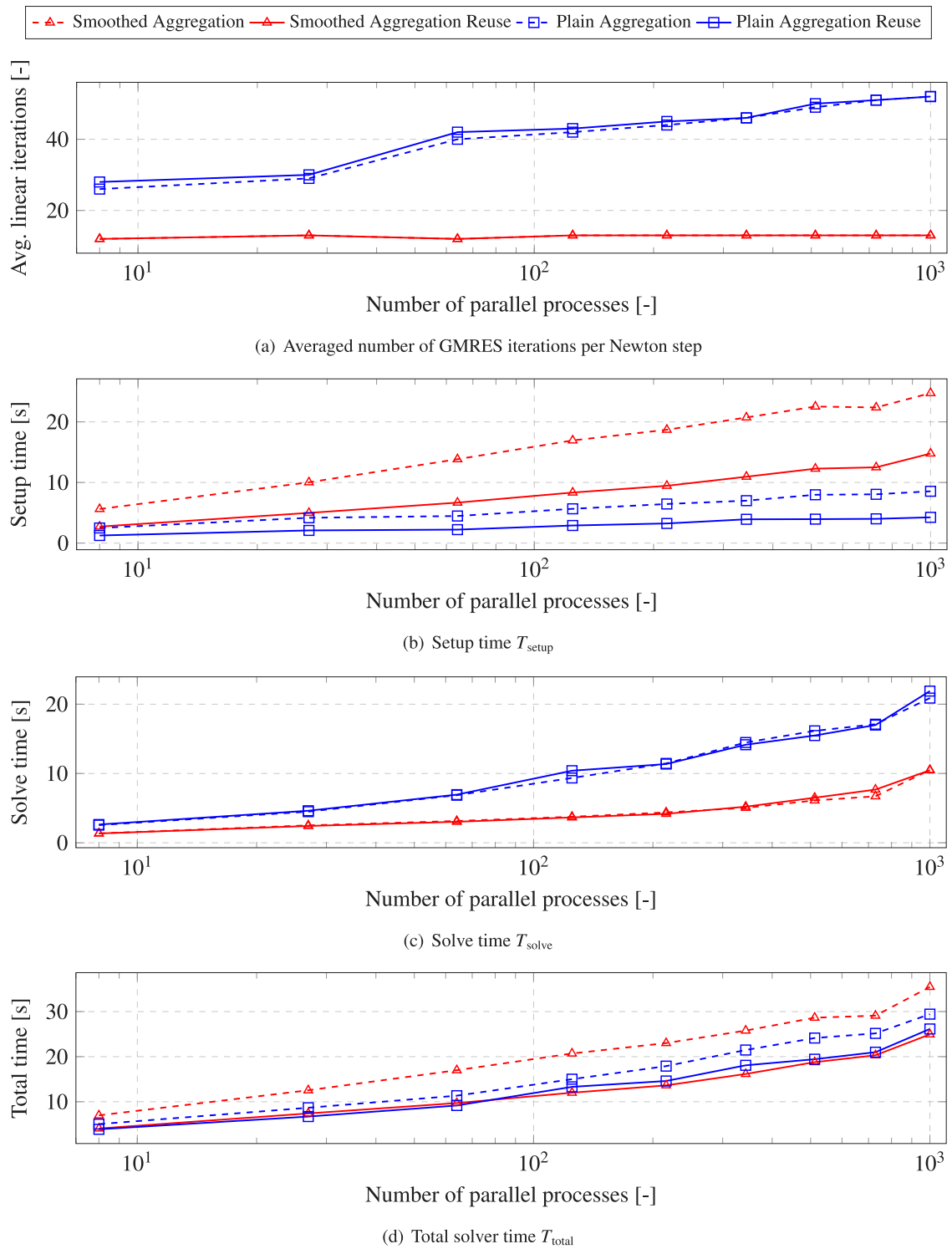


Fig. 6. Weak scaling behavior.



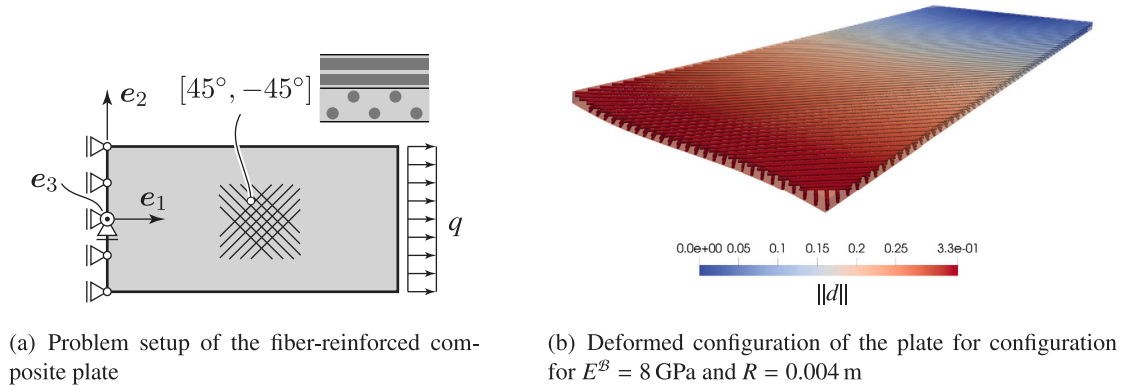


Fig. 7. Model problem for the robustness study. The left figure is adapted from our previous work [1], permissions granted under the Creative Commons (CC BY) license.

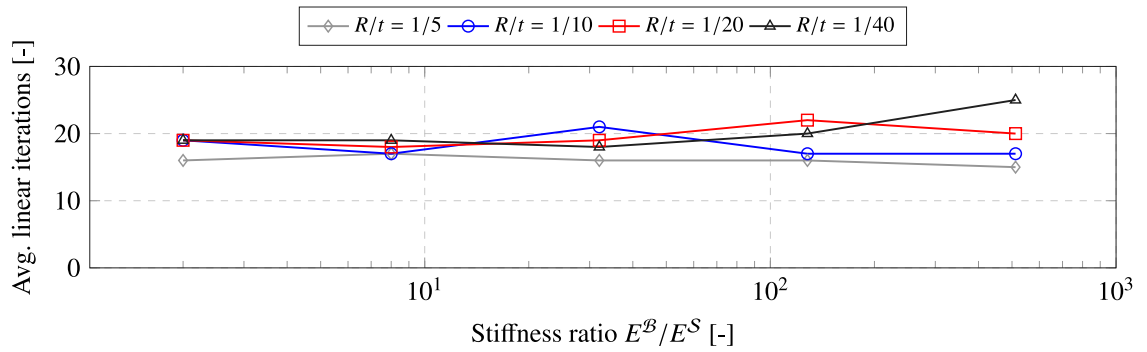


Fig. 8. Robustness study of the preconditioning regarding the beam to solid stiffness ratio and plate thickness to beam radius ratio: Iteration numbers are not affected by changes in physical parameters.

torsion-free Kirchhoff–Love beams, where we vary the beam radius  $R \in \{0.001 \text{ m}, 0.002 \text{ m}, 0.004 \text{ m}, 0.008 \text{ m}\}$  and the beam’s Young’s modulus  $E^B \in \{2 \text{ GPa}, 8 \text{ GPa}, 32 \text{ GPa}, 128 \text{ GPa}, 256 \text{ GPa}, 512 \text{ GPa}\}$ . The examined beam-solid stiffness ratios  $E^B/E^S$  span a wide array of practical applications, ranging from natural fiber composites with low ratios to steel-reinforced concrete and carbon fiber composites with high ratios. The penalty parameter is chosen as  $e^V = E^B$  to properly enforce the positional coupling constraints. Considering boundary conditions, the left side of the plate is fixed and a distributed tensile load  $q$  is applied to the right side. The problem setup and the deformed configuration (exemplary for  $E^B = 8 \text{ GPa}$  and  $R = 0.004 \text{ m}$ ) are depicted in Fig. 7. For a more detailed description of the problem, the reader is referred to [1]. Since the overall compound plate stiffness changes with the different parameter combinations described above, the load  $q$  is adapted such that the axial deformations of the plate are the same for each parameter combination and match the example shown in [1]. The overall simulation is of quasi-static nature and the load is applied incrementally over the course of 10 load steps.

The nonlinear solver converges for all test cases if the nonlinear residual  $\|\mathbf{f}\|_2$  drops below  $10^{-6}$  and the displacement increment  $\|\Delta \mathbf{d}\|_2$  falls below  $10^{-6}$ . For the solution of the linear system arising in each nonlinear iteration, a preconditioned GMRES method is applied with the proposed preconditioner. Hereby, the parameters for the block method are similar to test case I from Section 5.1: We set the number of applications of the preconditioner per linear solver iteration to  $k = 1$ , build the preconditioner once per load step and then reuse it in every nonlinear iteration of this load step. Due to the rather long fibers appearing in this example, the settings for the SPAI smoother are changed to  $m = 3$ . In addition, the SPAI computation for the beam sub-matrix  $\mathbf{A}$  uses a drop tolerance  $\sigma = 10^{-8}$  and an increased refinement level  $\ell = 4$  to handle the larger individual beam sub-blocks properly. Similar to before, the Schur complement equation is solved with an aggregation-based AMG method. Coarsening is performed until the number of unknowns on the coarsest level drops below 6500. The AMG hierarchy is traversed using a V-cycle with level transfer operators arising from SA-AMG. On all but the coarsest level, the level smoother is chosen as ILUT with overlap  $\delta = 1$ , an increased fill-in level  $p = 2.5$  (again necessary due to the length of the fibers and their discretization with sometimes more than ten beam elements, resulting in a denser Schur complement matrix) and drop-off tolerance  $\tau = 10^{-4}$ . The coarse level is solved with a direct solver using the distributed memory version of SUPERLU [59]. The linear solver is assumed to be converged if the relative residual  $\|\mathbf{r}^n\|_2 / \|\mathbf{r}^0\|_2$  falls below  $10^{-6}$ . All simulations are done in serial on a single processor.

Averaged iteration counts of the linear solver are shown in Fig. 8. At global scope, the number of iterations appears to be independent of the stiffness ratio  $E^B/E^S$  as well as the geometric ratio  $R/t$ . This is particularly true for a larger beam radius, i.e.,

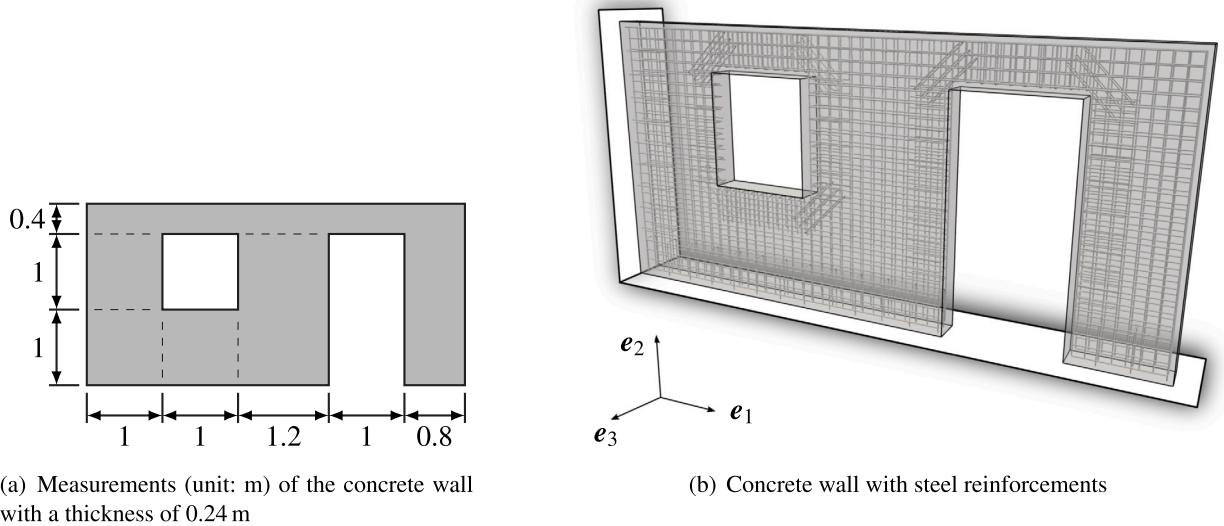


Fig. 9. Geometric configuration of a steel-reinforced concrete wall.

$R/t = 1/5$ . While smaller beam radii are expected to be more challenging to handle, their impact on the solver performance is very limited: the increase in iterations is very small compared to the case of  $R/t = 1/5$ . Intermediate geometric ratios  $R/t = \{1/10, 1/20\}$  exhibit slight changes in the iteration number under an increasing stiffness ratio  $E^B/E^S$ , though neither outliers nor a trend towards increasing iteration numbers has been observed. Considering the smallest beam radius, i.e.,  $R/t = 1/40$ , the iteration count remains rather constant for all but the largest stiffness ratio. Only the largest stiffness ratio  $E^B/E^S = 512$  GPa in combination with the smallest radius results in an outlier in iteration numbers that is slightly above all other cases. The challenges of small radii for the iterative solvers have also been reported in [30]. Overall, this study reveals a satisfying robustness of the proposed preconditioner in relevant application scenarios.

#### 5.4. Application: Concrete wall with steel reinforcements

To show the applicability of the proposed block preconditioner to real-world problems, we study an example from civil engineering, in particular the loading of a steel-reinforced concrete wall. The problem setup and its dimensions are shown in Fig. 9.

Since we are focusing on the performance of the linear solver and the proposed preconditioner, we restrict ourselves in this example to an idealized, fully elastic constitutive behavior of concrete. Therefore, it suffices to model the solid with a St.-Venant–Kirchhoff material ( $E^S = 30$  GPa,  $\nu^S = 0.3$ ). In this study, we, thus, refrain from using more elaborate constitutive models, that also cover inelastic effects, such as the Drucker–Prager model [60], which could serve as a smeared, phenomenological model for concrete undergoing damage or crack initiation. Yet, we study a complex reinforcement design: The initially curved reinforcement bars are modeled as Simo–Reissner beams ( $E^B = 210$  GPa,  $\nu^B = 0.0$ ) with a fiber cross section radius of  $R = 0.005$  m. The bottom and left side of the wall are clamped, restricting the displacement of the solid and fibers. Additionally, a distributed load of  $-3 \times 10^7$  N/m<sup>2</sup> is applied on the top surface in  $e_2$  direction and  $1.5 \times 10^5$  N/m<sup>2</sup> on the back surface in  $e_3$  direction. The reinforced concrete wall is discretized with first-order hexahedral finite elements for the solid domain and Simo–Reissner beam elements for the reinforcement fibers, respectively. We study this problem for three different mesh sizes (coarse, medium, fine), where each refinement quadruples the total number of unknowns. Details on these meshes including the resulting number of degrees of freedom for each field are given in Table 5 along with the computational resources, i.e., number of MPI ranks  $n^{\text{proc}}$ , used for each mesh. The penalty parameters for positional and rotational couplings are set to  $\epsilon^V = 2.1 \times 10^{11}$  N/m<sup>2</sup> and  $\epsilon^R = 5.25 \times 10^6$  Nm/m, respectively. We perform a quasi-static simulation and impose the total load over the course of four load steps.

Exemplarily, Fig. 10 depicts the sparsity pattern of the system matrix for the coarse mesh. The  $2 \times 2$  blocking as introduced in (8) is highlighted through colors. The beam sub-block A (orange) internally exhibits block diagonal structure as illustrated in Fig. 4, which allows for an efficient construction of the SPAI, since the sparsity pattern of the inverse can be estimated very well based on the block-diagonal structure of A. Arising from the penalty contributions assembled into C, the solid sub-block C (white) contains many entries far away from its diagonal, exemplifying the attested lack of diagonal dominance, cf. Section 3.2. Both beams and solid are connected through the coupling blocks  $\mathbf{B}_1^T$  and  $\mathbf{B}_2$  (gray), which are the main reasons for the loss of block diagonal dominance, again see Section 3.2.

For all simulations, the nonlinear solver converges if the nonlinear residual  $\|\mathbf{f}\|_2$  drops below  $10^{-8}$  and the displacement increment  $\|\Delta \mathbf{d}\|_2$  falls below  $10^{-8}$ . To get an idea of the effectiveness of the proposed preconditioner in an application scenario, we compare different methods to solve the arising linear system in each Newton step, in particular a direct solver (by applying the

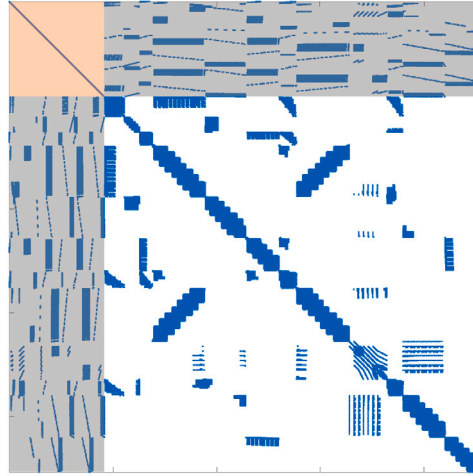


Fig. 10. Sparsity pattern of the linear system arising from the coarse mesh of the concrete wall (cf. Table 5) with the block-diagonal beam sub-block A (orange), the solid sub-block C (white), and the off-diagonal coupling blocks  $B_1^T$  and  $B_2$  (gray). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5

Comparison of averaged linear solver timings per nonlinear iteration for a steel-reinforced concrete wall (methods: *Direct* — direct solver based on an LU factorization, *Naive* — a naive-preconditioned GMRES solver using an ILU factorization as preconditioner, *Block* — a GMRES solver using the block preconditioner proposed in Section 4).

$n^{\text{proc}}$	$n_{\text{DOF}}^S$	$n_{\text{DOF}}^B$	$n_{\text{DOF}}^{\text{total}}$	Method	Reuse	#iter	CPU time (s)			Speedup $\frac{T_{\text{total}}^{\text{direct}}}{T_{\text{total}}^{\text{naive}}}$	
							$T_{\text{setup}}$	$T_{\text{solve}}$	$T_{\text{total}}$		
8	180 888	45 894	226 782	Direct	no	–	1070	2.8	1072.8		
				Naive	no			no convergence			
				Block	no	25	100.9	13.9	114.8	$\approx 9$	
				Block	yes	26	19.4	13.3	32.7	$\approx 33$	
32	844 545	45 894	890 439	Direct	no	–	5358	8.3	5366.3		
				Naive	no			no convergence			
				Block	no	25	145.9	17.6	163.5	$\approx 33$	
				Block	yes	26	29.2	18.9	48.1	$\approx 112$	
128	3 323 565	45 894	3 369 459	Direct	no			not feasible			
				Naive	no			no convergence			
				Block	no	22	198.3	20.4	218.7	n/a	
				Block	yes	23	38.5	21.1	59.6	n/a	

distributed memory version of SUPERLU [59]), a naive approach represented by a GMRES solver with an ILU preconditioner, and a GMRES solver with the proposed block preconditioner from Section 4. Where applicable, we also study the effect of reusing the preconditioner over multiple invocations of the linear solver within a single load step to better amortize the potentially expensive setup of the block preconditioner. In case of GMRES, we assume convergence of the linear solver, if the full relative residual norm  $\|\mathbf{r}^n\|_2 / \|\mathbf{r}^0\|_2$  drops to at least  $10^{-6}$ .

In case of the block preconditioner, we apply  $k = 3$  sweeps of the proposed preconditioner within each GMRES iteration. The SPAI computation for the beam sub-matrix A uses a drop tolerance  $\sigma = 10^{-8}$  and a refinement level  $\ell = 4$  to enrich the sparsity pattern. For the predictor and corrector step, the described SPAI smoother is applied with  $m = 3$  sweeps. The Schur complement equation is tackled with a PA-AMG hierarchy with a maximum size of the coarse level problem of 6500 unknowns, which results in three MG levels for all meshes. Prolongator smoothing (as proposed by [52] and usually beneficial for problems in solid mechanics) is explicitly disabled to reduce the fill-in of the coarse level matrices; see [53] for a detailed comparison. An ILUT method with  $\delta = 1$ ,  $\rho = 2.5$ , and  $\tau = 10^{-4}$  is used as level smoother. The coarse level equations are solved directly using the distributed memory version of SUPERLU [59].

The solver options and their iteration counts and timings are summarized in Table 5. Each of the four load steps requires five Newton iterations to reach convergence of the nonlinear solver. The reported iteration counts and timings have been averaged over all load steps and Newton steps, such that the numbers now give a good estimate for the cost of a single invocation of the linear solver. For the direct solver, the coarse and medium mesh could be solved, while the fine mesh was infeasible, i.e., one load step taking more than three days of wall clock time on a cluster, such that we do not report the final result. For the out-of-the-box iterative solver using a GMRES method preconditioned with an incomplete LU factorization, the iterative solver was not able to reach

convergence within 1000 iterations. Only GMRES with the proposed block preconditioner from Section 4 was able to solve all three levels of mesh refinement. Moreover, the iterative method outperforms the direct solver in the sense that it is roughly  $9\times$  faster than the direct method for the coarse discretization. When moving to the medium-sized mesh, the discrepancy in solver timings increases even further: the direct method requires a total solver time  $T_{\text{total}} = 5366.3$  s for a single solve, while the total solver time  $T_{\text{total}}$  of the preconditioned GMRES only takes 163.5 s, resulting in a speed-up of approximately  $33\times$ . Increasing the problem size even further makes the application of a direct method infeasible, as memory consumption and computing time become prohibitively high, leaving the iterative approach as the sole viable option. In all working cases, the linear iterations for the preconditioned iterative method per Newton step remain almost constant over the whole simulation for each problem and in addition also stay almost constant for each mesh size. For the coarse and medium meshes, approximately 25 iterations are necessary to achieve the desired tolerance, whereas the large setup requires 22 iterations on average. Still, the discrepancy between  $T_{\text{setup}}$  and  $T_{\text{solve}}$  is rather large for the iterative method, as most of the computation time is spent in the construction of the factorization of the ILUT level smoother. To better amortize the expensive setup cost, one can build the preconditioner only once per load step and then reuse it for each Newton step with the goal of decreasing the overall simulation time. This shows to be an effective option, as it reduces the setup time  $T_{\text{setup}}$  by a factor of  $5\times$  in each case, which is perfectly in line with using the preconditioner five times, but only building it once per load step. Due to the reuse, the preconditioner is not perfectly fitting the system matrix anymore, occasionally resulting in a slight increase in iteration numbers (#iter) and solver time  $T_{\text{solve}}$ . Overall though, the total solver time  $T_{\text{total}}$  is reduced, which also manifests itself in the speed-up factors of  $33\times$  and  $112\times$  in Table 5 for the coarse and medium mesh. The infeasibility of a direct solver for the fine mesh prevents the calculation of a speed-up factor (n/a), however clearly testifies to the beneficial impact of the proposed preconditioner on the solvability of large-scale application examples. For both rebuilding or reusing the preconditioner, the iteration counts appear to be independent of the mesh size also in this example from engineering practice.

Overall, this example demonstrates not only the applicability of the proposed preconditioner in engineering problems, but also its benefits in terms of efficiency and speed-up, ultimately enabling the analysis of large and complex fiber-solid systems, which have not been accessible with existing linear solvers so far.

## 6. Concluding remarks

In this paper, we have proposed a physics-based multi-level block preconditioner for the scalable solution of mixed-dimensional models in beam-solid interaction, specifically tailored to systems with many independent fibers being embedded into a solid domain. The regularized mortar-type coupling approach leads to  $2 \times 2$  block systems exhibiting particular properties, most prominently the lack of block diagonal dominance stemming from the penalty terms on the off-diagonal coupling blocks, which render classical block relaxation preconditioners inapplicable. To precondition an outer Krylov solver, we utilize an approximate block factorization to enable the tackling of the individual blocks and their coupling within the preconditioner. To this end, we exploit the beam-related sub-block's sparsity structure resembling a block diagonal matrix to explicitly construct a sparse approximate inverse (SPAI) by solving a minimization problem over the Frobenius norm on a given sparsity pattern, which in practice is decomposed into row-wise minimizations to be solved in parallel. To increase its robustness and approximation quality, the SPAI computation is equipped with pre-processing steps such as a filtering of small entries and a static enrichment of the sparsity pattern as well as a post-filtering of small entries. This approximation has then not only been used for the explicit formation of an approximation to the Schur complement, but also as a smoother in the Block-LU's prediction and correction steps. To solve the Schur complement equation, we have employed an AMG hierarchy. Due to the Schur complement's fill-in stemming from the SPAI matrix as well as the penalty contributions, an ILUT factorization with fill-in  $p$ , threshold  $\tau$ , and overlap  $\delta$  serves as level smoother on all levels except for the direct solver on the coarse level. All building blocks of the proposed preconditioner have been implemented in TRILINOS and are available as open-source software to the entire scientific community.

We have studied the influence of the SPAI algorithm's parameters and have found, that a static enrichment of the graph  $\mathcal{J}(\mathbf{A})$  of at least  $\ell = 2$  greatly improves the quality of the approximation as well as the performance of the iterative solver, while more enrichment might be required for particularly challenging problems. Regarding the scaling behavior, we were able to demonstrate weak scalability up to 1000 MPI ranks. While the iteration count is completely independent of the problem size and number of MPI ranks for SA-AMG, the setup and solver time exhibit a minor increase with an increasing problem size. This is mainly attributed to the use of an ILUT level smoother within the Block-LU's Schur complement step. We have demonstrated the robustness of the preconditioner by showing that the iteration counts remain constant when changing critical physical parameters such as the stiffness ratio between fibers and bulk field or the fiber radii. Finally, we have investigated an application example from civil engineering, in particular a steel-reinforced concrete wall, and have compared the performance of the proposed multi-level preconditioner to established one-level preconditioners and direct solvers. Even for a coarse mesh, the one-level preconditioner failed to converge. The proposed multi-level preconditioner delivered a speed-up by a factor up to  $112\times$  compared to the direct solver on small and medium sized meshes, whereas the application of a direct method on the finest mesh was not feasible anymore, leaving the proposed preconditioned iterative method as the only working option. Again, the iteration count is independent of the mesh size. While intractable for existing solvers, the proposed preconditioner enables the analysis of mixed-dimensional fiber-solid systems with complex reinforcement structures for the first time. Considering computational performance, the option of building the preconditioner only once per load step and then reusing it in every iteration of the nonlinear solver has shown to cut down the setup time  $T_{\text{setup}}$  at the expected rate, while still delivering a strong preconditioning effect, such that the convergence of the linear solver is not impeded and the total solver time  $T_{\text{total}}$  is reduced significantly.

In future work, the proposed preconditioner and its building blocks can be extended to other types of beam-solid interaction phenomena such as the coupling of beams onto a solid's surface [14] or the contact between beams and solid bodies [34]. Similarly, other mixed-dimensional multi-physics systems such as fiber-fluid interaction are likely to be amenable to such a methodology. It seems worthwhile to use semi-structured grids to discretize the solid domain, if applicable, which in turn allow one to further enhance its computational performance in many application scenarios [61]. Performance and scalability bottlenecks associated with the evaluation of the beam-solid coupling terms on a distributed-memory parallel computing cluster need to be addressed, e.g., following ideas from mortar methods for contact mechanics to improve data locality and load balancing [62].

### CRediT authorship contribution statement

**Max Firmbach:** Writing – original draft, Software, Methodology, Data curation. **Ivo Steinbrecher:** Writing – original draft, Software. **Alexander Popp:** Writing – review & editing, Supervision, Funding acquisition. **Matthias Mayr:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

### Declaration of competing interest

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

### Data availability

Building blocks of the AMG preconditioner developed and applied in this study are openly available in Trilinos at <https://github.com/trilinos/Trilinos> [55,56]. All other data that support the findings of this study are available from the corresponding author upon reasonable request.

### Acknowledgments

The work described in this contribution has been funded by dtec.bw - Digitalization and Technology Research Center of the Bundeswehr under the project “hpc.bw - Competence Platform for High Performance Computing”. dtec.bw is funded by the European Union – NextGenerationEU. Partial support has been provided by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the project “Stable discretization methods and scalable solvers for embedded fiber/solid coupling” (project number 528397555) and by the Bayerisches Verbundforschungsprogramm (BayVFP) – Digitalisierung within the project “IFB-Individuelle Fließfertigung für Betonfertigteile” (project number DIK-2201-0016/DIK0411/03). The authors gratefully acknowledge the computing resources provided by the Data Science & Computing Lab at the University of the Bundeswehr Munich.

Some sketches in this work have been created using the Adobe Illustrator plug-in LaTeX2AI (<https://github.com/isteinbrecher/latex2ai>).

### References

- [1] I. Steinbrecher, M. Mayr, M.J. Grill, J. Kremheller, C. Meier, A. Popp, A mortar-type finite element approach for embedding 1D beams into 3D solid volumes, *Comput. Mech.* 66 (6) (2020) 1377–1398.
- [2] I. Steinbrecher, A. Popp, C. Meier, Consistent coupling of positions and rotations for embedding 1D Cosserat beams into 3D solid volumes, *Comput. Mech.* 69 (2022) 701–732.
- [3] B.D. Agarwal, L.J. Broutman, K. Chandrashekhara, *Analysis and Performance of Fiber Composites*, Wiley, 2017.
- [4] P. Raghavan, S. Ghosh, A continuum damage mechanics model for unidirectional composites undergoing interfacial debonding, *Mech. Mater.* 37 (9) (2005) 955–979.
- [5] M.A. Puso, A 3D mortar method for solid mechanics, *Internat. J. Numer. Methods Engrg.* 59 (3) (2004) 315–336.
- [6] B. Flemisch, B.I. Wohlmuth, Stable Lagrange multipliers for quadrilateral meshes of curved interfaces in 3D, *Comput. Methods Appl. Mech. Engrg.* 196 (8) (2007) 1589–1602.
- [7] É. Béchet, N. Moës, B.I. Wohlmuth, A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method, *Internat. J. Numer. Methods Engrg.* 78 (8) (2009) 931–954.
- [8] M. Hautefeuille, C. Annavarapu, J.E. Dolbow, Robust imposition of Dirichlet boundary conditions on embedded surfaces, *Internat. J. Numer. Methods Engrg.* 90 (1) (2012) 40–64.
- [9] M.A. Puso, T.A. Laursen, A mortar segment-to-segment contact method for large deformation solid mechanics, *Comput. Methods Appl. Mech. Engrg.* 193 (6–8) (2004) 601–629.
- [10] M.A. Puso, T.A. Laursen, A mortar segment-to-segment frictional contact method for large deformations, *Comput. Methods Appl. Mech. Engrg.* 193 (45–47) (2004) 4891–4913.
- [11] A. Popp, M. Gitterle, M.W. Gee, W.A. Wall, A dual mortar approach for 3D finite deformation contact with consistent linearization, *Internat. J. Numer. Methods Engrg.* 83 (11) (2010) 1428–1465.
- [12] T.A. Wiesner, M. Mayr, A. Popp, M.W. Gee, W.A. Wall, Algebraic multigrid methods for saddle point systems arising from mortar contact formulations, *Internat. J. Numer. Methods Engrg.* 122 (15) (2021) 3749–3779.
- [13] D. Durville, Finite element simulation of textile materials at mesoscopic scale, in: *Finite Element Modelling of Textiles and Textile Composites*, Saint Petersburg, 2007.
- [14] I. Steinbrecher, N. Hagemeyer, C. Meier, A. Popp, A consistent mixed-dimensional coupling approach for 1D cosserat beams and 2D solid surfaces, 2022, preprint on arXiv: <https://arxiv.org/abs/2210.16010>.
- [15] S. Kakaletsis, E. Lejeune, M. Rausch, The mechanics of embedded fiber networks, *J. Mech. Phys. Solids* 181 (2023) 105456.

- [16] U. Khristenko, S. Schuß, M. Krüger, F. Schmidt, B. Wohlmuth, C. Hesch, Multidimensional coupling: A variationally consistent approach to fiber-reinforced materials, *Comput. Methods Appl. Mech. Engrg.* 382 (2021) 113869.
- [17] B. Lé, G. Legrain, N. Moës, Mixed dimensional modeling of reinforced structures, *Finite Elem. Anal. Des.* 128 (2017) 1–18.
- [18] J. Ao, M. Zhou, B. Zhang, A dual mortar embedded mesh method for internal interface problems with strong discontinuities, *Internat. J. Numer. Methods Engrg.* 123 (22) (2022) 5652–5681.
- [19] N. Hagemeyer, M. Mayr, I. Steinbrecher, A. Popp, One-way coupled fluid-beam interaction: Capturing the effect of embedded slender bodies on global fluid flow and vice versa, *Adv. Model. Simul. Eng. Sci.* 9 (2022) 9.
- [20] N. Hagemeyer, M. Mayr, A. Popp, A fully coupled regularized mortar-type finite element approach for embedding one-dimensional fibers into three-dimensional fluid flow, *Internat. J. Numer. Methods Engrg.* (2024) e7435, published online ahead of print.
- [21] F. Lespagnol, C. Grandmont, P. Zunino, M.A. Fernandez, A Mixed-Dimensional Formulation for the Simulation of Slender Structures Immersed in an Incompressible Flow, Technical Report 98/2023, MOX Politecnico di Milano, 2023.
- [22] M. Mayr, T. Klöppel, W.A. Wall, M.W. Gee, A Temporal Consistent Monolithic Approach to Fluid–Structure Interaction Enabling Single Field Predictors, *SIAM J. Sci. Comput.* 37 (1) (2015) B30–B59.
- [23] M. Mayr, M. Noll, M.W. Gee, A hybrid interface preconditioner for monolithic fluid-structure interaction solvers, *Adv. Model. Simul. Eng. Sci.* 7 (2020) 15.
- [24] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, USA, 2003.
- [25] M. Kuchta, M. Nordaas, J.C.G. Verschaeve, M. Mortensen, K.-A. Mardal, Preconditioners for Saddle Point Systems with Trace Constraints Coupling 2D and 1D Domains, *SIAM J. Sci. Comput.* 38 (6) (2016).
- [26] M. Kuchta, K.-A. Mardal, M. Mortensen, Preconditioning trace coupled 3d-1d systems using fractional Laplacian, *Numer. Methods Partial Differential Equations* 35 (1) (2019) 375–393.
- [27] T. Baerland, M. Kuchta, K.-A. Mardal, Multigrid Methods for Discrete Fractional Sobolev Spaces, *SIAM J. Sci. Comput.* 41 (2) (2019) A948–A972.
- [28] A. Budiša, X. Hu, M. Kuchta, K.-A. Mardal, L. Zikatanov, Algebraic multigrid methods for metric-perturbed coupled problems, *SIAM J. Sci. Comput.* 46 (3) (2024) A1461–A1486.
- [29] K.-A. Mardal, R. Winther, Preconditioning discretizations of systems of partial differential equations, *Numer. Linear Algebra Appl.* 18 (1) (2011) 1–40.
- [30] N. Dimola, M. Kuchta, K.A. Mardal, P. Zunino, Robust preconditioning of mixed-dimensional PDEs on 3d-1d domains coupled with Lagrange multipliers, in: A. Linninger, K.A. Mardal, P. Zunino (Eds.), *Quantitative Approaches to Microcirculation*, in: SEMA SIMAI Springer Series, vol. 36, Springer, Cham, 2024.
- [31] D. Cerroni, F. Laurino, P. Zunino, Mathematical analysis, finite element approximation and numerical solvers for the interaction of 3D reservoirs with 1D wells, *Int. J. Geomath.* 10 (2019) 4.
- [32] M.J. Grote, T. Huckle, Parallel preconditioning with sparse approximate inverses, *SIAM J. Sci. Comput.* 18 (3) (1997) 838–853.
- [33] L. Berger-Vergiat, C.A. Glusa, G. Harper, J.J. Hu, M. Mayr, A. Prokopenko, C.M. Siefert, R.S. Tuminaro, T.A. Wiesner, MueLu User's Guide, Technical Report SAND2023-12265, Sandia National Laboratories, Albuquerque, NM (USA) 87185, 2023.
- [34] I.S. Steinbrecher, Mixed-Dimensional Finite Element Formulations for Beam-To-Solid Interaction (Ph.D. thesis), Universität der Bundeswehr München, 2022.
- [35] C. Meier, A. Popp, W.A. Wall, A locking-free finite element formulation and reduced models for geometrically exact Kirchhoff rods, *Comput. Methods Appl. Mech. Engrg.* 290 (2015) 314–341.
- [36] C.A. Meier, A. Popp, W.A. Wall, Geometrically Exact Finite Element Formulations for Slender Beams: Kirchhoff–Love Theory Versus Simo–Reissner Theory, *Arch. Comput. Methods Eng.* 26 (1) (2019) 163–243.
- [37] D. Feingold, R. Varga, Block diagonally dominant matrices and generalizations of the Gershgorin Theorem, *Pacific J. Math.* 12 (4) (1962) 1241–1250.
- [38] L. Chacón, An optimal, parallel, fully implicit Newton–Krylov solver for three-dimensional viscoelastic magnetohydrodynamics, *Phys. Plasmas* 15 (2008) 056103.
- [39] E.C. Cyr, J.N. Shadid, R.S. Tuminaro, Teko: A Block Preconditioning Capability with Concrete Example Applications in Navier–Stokes and MHD, *SIAM J. Sci. Comput.* 38 (5) (2016) S307–S331.
- [40] S. Patankar, D. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *Int. J. Heat Mass Transfer* 15 (10) (1972) 1787–1806.
- [41] H.C. Elman, V.E. Howle, J.N. Shadid, R. Shuttleworth, R.S. Tuminaro, A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 227 (3) (2008) 1790–1808.
- [42] U. Langer, H. Yang, Robust and efficient monolithic fluid-structure-interaction solvers, *Internat. J. Numer. Methods Engrg.* 108 (4) (2016) 303–325.
- [43] D. Jodlbauer, U. Langer, T. Wick, Parallel block-preconditioned monolithic solvers for fluid-structure interaction problems, *Internat. J. Numer. Methods Engrg.* 117 (6) (2019) 623–643.
- [44] E.G. Phillips, H.C. Elman, E.C. Cyr, J.N. Shadid, R.P. Pawlowski, A Block Preconditioner for an Exact Penalty Formulation for Stationary MHD, *SIAM J. Sci. Comput.* 36 (6) (2014) B930–B951.
- [45] E.G. Phillips, J.N. Shadid, E.C. Cyr, H.C. Elman, R.P. Pawlowski, Block Preconditioners for Stable Mixed Nodal and Edge finite element Representations of Incompressible Resistive MHD, *SIAM J. Sci. Comput.* 38 (6) (2016) B1009–B1031.
- [46] J.P. Van Doormaal, G.D. Raithby, Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer* 7 (2) (1984) 147–163.
- [47] E. Chow, A. Patel, Fine-Grained Parallel Incomplete LU Factorization, *SIAM J. Sci. Comput.* 37 (2) (2015) C169–C193.
- [48] M. Sedlacek, Sparse Approximate Inverse for Preconditioning, Smoothing, and Regularization (Ph.D. thesis), Technische Universität München, 2012.
- [49] E. Chow, Parallel implementation and practical use of sparse approximate inverse preconditioners with a priori sparsity patterns, *Int. J. High Perform. Comput. Appl.* 15 (1) (2001) 56–74.
- [50] O. Bröker, M.J. Grote, Sparse approximate inverse smoothers for geometric and algebraic multigrid, *Appl. Numer. Math.* 41 (1) (2002) 61–80.
- [51] Y. Saad, ILUT: A dual threshold incomplete LU factorization, *Numer. Linear Algebra Appl.* 1 (4) (1994) 387–402.
- [52] P. Vaněk, J. Mandel, M. Brezina, Algebraic Multigrid By Smoothed Aggregation For Second And Fourth Order Elliptic Problems, *Computing* 56 (1996) 179–196.
- [53] S.J. Thomas, S. Ananthan, S. Yellapantula, J.J. Hu, M. Lawson, M.A. Sprague, A comparison of classical and aggregation-based algebraic multigrid preconditioners for high-fidelity simulation of wind turbine incompressible flows, *SIAM J. Sci. Comput.* 41 (5) (2019) S196–S219.
- [54] 4C: A Comprehensive Multi-Physics Simulation Framework, 2024, <https://www.4c-multiphysics.org>. (Accessed 07 June 2024).
- [55] M.A. Heroux, R.A. Bartlett, V.E. Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, Lehocq, K.R. Long, R.P. Pawlowski, E.T. Phipps, A.G. Salinger, H.K. Thornquist, R.S. Tuminaro, J.M. Willenbring, A. Williams, K.S. Stanley, An Overview of the Trilinos Project, *ACM Trans. Math. Software* 31 (3) (2005) 397–423.
- [56] The Trilinos Project, 2024, <https://trilinos.github.io>. (Accessed 07 June 2024).
- [57] I. Steinbrecher, A. Popp, MeshPy – A general purpose 3D beam finite element input generator, 2021, <https://imcs-compsim.github.io/meshpy>.
- [58] Y. Saad, M.H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.* 7 (3) (1986) 856–869.
- [59] X.S. Li, J.W. Demmel, SuperLU\_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems, *ACM Trans. Math. Softw.* 29 (2) (2003) 110–140.
- [60] D.C. Drucker, W. Prager, Soil mechanics and plastic analysis for limit design, *Quart. Appl. Math.* 10 (2) (1952) 157–166.
- [61] M. Mayr, L. Berger-Vergiat, P. Ohm, R.S. Tuminaro, Non-invasive multigrid for semi-structured grids, *SIAM J. Sci. Comput.* 44 (4) (2022) A2734–A2764.
- [62] M. Mayr, A. Popp, Scalable computational kernels for mortar finite element methods, *Eng. Comput.* 39 (2023) 3691–3720.