



# Highly scalable two-level monolithic overlapping Schwarz preconditioners for thermo-elastoplastic laser beam welding problems

Tommaso Bevilacqua<sup>1</sup> · Axel Klawonn<sup>1,2</sup> · Martin Lanser<sup>1,2</sup>

Received: 28 May 2025 / Accepted: 16 July 2025 / Published online: 5 August 2025  
© The Author(s) 2025

## Abstract

A thermo-elastoplastic finite element approach is used to perform the simulation of a Laser Beam Welding (LBW) process. This results in a nonlinear, nonsymmetric saddle point multiphysics system, for which the nonlinearity is handled via the Newton method. The iterative solution of the arising linear system is accelerated by using robust and highly scalable, overlapping Schwarz domain decomposition preconditioners. It is well-known that a one-level method of this type is not scalable and therefore a second level has to be added. Therefore, the construction and numerical analysis of monolithic, two-level overlapping Schwarz preconditioners with different variants of the GDSW (Generalized Dryja–Smith–Widlund) coarse space are presented here. A new and parallel efficient implementation of several variants of GDSW, that is, GDSW, RGDSW, and GDSW\*, in PETSc (Portable, Extensible Toolkit for Scientific Computing), is introduced, which is usable for multiphysics problems, as, for example, the thermo-mechanical LBW problem considered here. Different combinations of the GDSW variants for the different fields (temperature and displacements) are compared and parallel scalability for realistic problems is shown.

**Keywords** Laser beam welding · Saddle point problems · Schwarz methods · GDSW · RGDSW · GDSW\* · Thermo-elastoplasticity · Monolithic preconditioner

## 1 Introduction

Overlapping Schwarz domain decomposition methods are a class of preconditioners for the large linear systems obtained from the discretization of many linear and nonlinear partial differential equations. Originally, they were introduced for uniformly elliptic partial differential equations. In [24] monolithic overlapping Schwarz methods were introduced

for saddle point problems originating from the discretization of Stokes' equations discretized with mixed finite elements. Here, the overlapping domain decomposition is applied directly to the saddle point problem without reducing it to sub-blocks; this requires the solution of smaller saddle point problems on each subdomain. Additionally, a coarse problem based on a saddle point formulation was introduced in [24]. For a brief review of monolithic overlapping Schwarz preconditioners, see the discussion further below in this introduction. In the present paper, we develop a monolithic overlapping Schwarz preconditioner for a challenging multiphysics problem from Laser Beam Welding (LBW).

LBW is a noncontact joining process that recently has become considerably more important in the course of the increasing degree of automation in industrial production. The advantages of this process are its short cycle times and the small heat-affected zones. However, the high cooling rate of the metal specimen inherent in the process can lead to solidification cracks, due to a residual melt oversaturated with certain alloy elements. It is therefore crucial to try to

✉ Tommaso Bevilacqua  
tommaso.bevilacqua@uni-koeln.de

✉ Axel Klawonn  
axel.klawonn@uni-koeln.de

✉ Martin Lanser  
martin.lanser@uni-koeln.de

<sup>1</sup> Department of Mathematics and Computer Science, University of Cologne, Cologne, Germany

<sup>2</sup> Center for Data and Simulation Science, University of Cologne, Cologne, Germany

quantitatively understand the development mechanisms of these fractures and their correlation to process parameters. In particular, this is the focus of the DFG research group 5134 "Solidification Cracks During Laser Beam Welding– High Performance Computing for High Performance Processing"<sup>1</sup>, where experiments, advanced multi-scale finite element modelling, and scientific computing are brought together to obtain new insights. A nice example for the interplay of experiments and high-performance simulations within this project is presented in [4], where a Controlled Tensile Weldability (CTW) test is performed both experimentally and virtually using finite element simulations. The CTW test is used to check if a material is susceptible for cracks during welding by measuring its ductility in the mushy zone, i.e. the region where a material is partially solid and partially liquid during cooling and phase transition. The probability of cracks is increased in the CTW test by applying different strain conditions and then temperature and strain are tracked optically to see how they affect cracking. This is a crucial analysis to improve welding processes and prevent defects. In [4], our PETSc (Portable, Extensible Toolkit for Scientific Computing) [1]-based software library FE2TI [21, 23] and its numerical solver infrastructure was used to perform the finite element simulations of the CTW test on high-performance computers. In the present work here we provide an accompanying in-depth study of the performance and scalability of our FE2TI software for LBW problems and, to be more precise, we investigate the robustness and scalability of the domain decomposition methods (DDMs) used to solve the large finite element problems arising in LBW simulations.

The thermomechanical properties of LBW can be represented by a coupled system of time dependent thermo-elastoplasticity equations [30]. Due to the nature of the problem, the linear system that arises from a finite element discretization in space, a Backward Euler discretization in time, and a linearization by Newton's method is ill-conditioned. As a consequence, here we discuss robust and parallel scalable preconditioners to accelerate the convergence of the Generalized Minimal Residual (GMRES) method. Since we are predominantly interested in analyzing the performance of the domain decomposition preconditioners and not in a precise modelling of realistic laser simulations, we here consider finite element problems where the action of the laser beam is simplified. In particular, we assume that the melting pool, that is, the portion of the material melted by the laser, has the shape of a simple cylinder that goes vertically through the entire material. Actually, the exact form of the melting pool does not play an important role for the solver performance and simulations of thermo-elastic and

thermo-elastoplastic LBW processes with more accurate representation of the melting pool [2] can be found in [4] and [5].

The action of the laser is applied as a volumetric constraint to the temperature field in the region of the melting pool that moves over time with a problem dependent speed. More realistic heat source models for the laser can also be used in order to obtain more accurate simulations [11, 18, 19], but again we do not expect a significant influence on the performance of our preconditioners.

As already mentioned, in this work, we focus on the DDMs used to iteratively solve the linearized systems arising in the simulation of LBW processes. To be more precise, we make use of a monolithic overlapping two-level additive Schwarz preconditioner for saddle point problems. Such monolithic domain decomposition preconditioners have been first introduced in [24]; see also [25] for a comparison to block-diagonal and block-triangular preconditioners using overlapping Schwarz methods on the sub-blocks. In order to obtain numerical scalability, a coarse level needs to be introduced. In [24] Lagrangian coarse spaces that operate on the saddle point problem defined on the discretization introduced by the domain decomposition have been used. Such a construction of the coarse space using the domain decomposition as a coarse finite element triangulation is quite restrictive. To obtain more flexibility in the construction of the coarse space, especially for more general shapes of subdomains, we use variants of the Generalized Dryja-Smith-Widlund (GDSW) coarse spaces, in which coarse basis functions are defined using problem dependent extension operators. In the case of symmetric positive definite matrices these extensions are discrete harmonic or, more generally, energy minimizing. The GDSW coarse space was originally introduced in [7, 8] for scalar diffusion problems and has been extended to saddle point problems arising from the discretization of incompressible fluid flow problems in [12, 16]. Let us note that in [12] a fully monolithic approach was chosen, using saddle point problems to compute the extensions. Despite being very robust, a disadvantage of this type of coarse space is the rapid growth of its size when scaling to a large number of subdomains, especially for three dimensional problem. Several strategies to reduce the size of the coarse space have been introduced, for example, GDSW\* [16] and Reduced-GDSW [9] coarse spaces. In the case of multi-physics problems one can combine the different coarse spaces to reduce its overall size. For instance, in case of fluids, choosing a larger coarse space for the velocity block and a smaller one for the pressure space is often beneficial. This idea has been introduced and analyzed in [13] for Navier–Stokes problems where many combinations of coarse spaces have been considered. Parallel implementations of monolithic Schwarz preconditioners have also been

<sup>1</sup> <https://www.for5134.science/en/>

applied to fully-coupled nonlinear chemo-mechanics problems; see [26] for details.

In our present work, for the first time, we give an exhaustive computational analysis of similar combinations of different coarse spaces of GDSW-type for thermo-elastoplasticity problems. For this purpose, we implemented all variants, that are, GDSW, GDSW\*, and RGDSW, in PETSc. For the nonlinear and time-dependent coupled thermo-elastoplasticity problems multiple linear systems arise from Newton’s method. To deliver a fair comparison and sound results, we also integrated well-known strategies to improve the efficiency, such as, an adaptive time stepping approach, truncation of the coarse matrix, and the possibility to recycle certain solver components from previous Newton steps. We show efficient parallel scalability of our new solvers for complex and realistic LBW problems.

The remainder of the paper is organized as follows. In Sect. 2 we present the thermo-elastoplasticity equations, in Sect. 3 we introduce the discretization, in Sect. 4 we present the domain decomposition preconditioners and all coarse spaces, in Sect. 5 we describe the implementation in PETSc, and, finally, in Sect. 6 we show several numerical results and a detailed comparison of the performance of different monolithic coarse spaces.

## 2 Thermo-elastoplasticity equations

In our work we focus on studying a coupled system of thermo-elasto-plasticity equations

$$\begin{aligned} \operatorname{div}(\boldsymbol{\sigma}(\mathbf{u}, \theta)) &= 0, \\ \operatorname{div}(\mathbf{q}) + \gamma \operatorname{div}(\dot{\mathbf{u}}) \theta + \rho c_p \dot{\theta} &= 0. \end{aligned}$$

Here  $\mathbf{u}(x, t)$  and  $\theta(x, t)$  represent the displacement vector and the absolute temperature respectively. These equations are simplified versions of the ones that can be found in [30], since in our case we assume to not have any volume acceleration and no external heat source. The action of a laser

beam is modeled by a volumetric constraint, directly fixing the temperature to the melting temperature of the material in the melting pool, which is an unusual but still effective approach.

The other variables, that represent the material-specific and temperature-dependent material parameters are:  $c_p$  the heat capacity,  $\mathbf{q}$  the heat flux vector given by  $\mathbf{q} = -\lambda \nabla \theta$ , and  $\gamma = 3\alpha_T \kappa$  the stress temperature modulus. Moreover,  $\alpha_T$  denotes the coefficient of linear thermal expansion,  $\kappa = E/3(1 - 2\nu)$  the bulk modulus and  $E$  the Young’s modulus. These temperature-dependent parameters have been chosen conformely to model an austenitic chrome-nickel steel (1.4301) as shown in Table 1. Finally,  $\rho = 7919 \text{ kg/m}^3$  is the density of the material, assumed to be constant [27].

Here, the thermo-elastoplastic model assumes an additive decomposition of total strains  $\boldsymbol{\varepsilon} = [\varepsilon_{11} \ \varepsilon_{22} \ \varepsilon_{33} \ \varepsilon_{13} \ \varepsilon_{12} \ \varepsilon_{23}]$  (in Voigt notation), into elastic, plastic, and thermal components. Based on this, the free energy function  $\psi$  is formulated as

$$\begin{aligned} \psi(\boldsymbol{\varepsilon}^{te}, \theta, \alpha) &= \frac{1}{2} \kappa (\operatorname{tr} \boldsymbol{\varepsilon}^{te})^2 + \mu \|\operatorname{dev} \boldsymbol{\varepsilon}^{te}\|^2 \\ &\quad - c_p \left( \theta \ln \frac{\theta}{\theta_0} - \theta + \theta_0 \right) \\ &\quad - 3\alpha_T \kappa (\theta - \theta_0) \operatorname{tr}(\boldsymbol{\varepsilon}^{te}) + f(\alpha), \end{aligned} \tag{1}$$

in terms of the thermo-elastic strains  $\boldsymbol{\varepsilon}^{te}$ , temperature, and the strain-like internal variable  $\alpha$  used in the hardening function  $f$ . This leads to the constitutive equation  $\boldsymbol{\sigma} = \partial_{\boldsymbol{\varepsilon}^{te}} \psi$ , which defines the stress–strain relationship. The model also considers the balance of momentum and energy and to characterize the plastic behavior, a von Mises yield criterion is employed, incorporating the deviatoric stress  $\operatorname{dev} \boldsymbol{\sigma}$ . For further details related to the model, see [29, 30] and for the same specific problem [4].

The weak formulation of the coupled thermo-elastoplasticity equations is

**Table 1** Parameters of the material austenitic chrome-nickel steel (1.4301) at different temperatures. The values, taken from [19], have been provided by Bundesanstalt für Materialforschung und -prüfung (BAM). A linear interpolation is used between the temperature intervals

Parameter	Value (Temperature [C])								
$E \cdot 1e4 \text{ [N/mm}^2\text{]}$	20.0 (20)	19.1 (170)	17.5 (400)	12.5 (800)	7.2 (1000)	1.6 (1100)	0.1 (1500)	–	–
$\nu$	0.271 (20)	0.284 (183)	0.300 (484)	0.319 (799)	0.329 (994)	0.364 (1994)	0.364 (2000)	–	–
$\alpha_T \cdot 1e - 5 \text{ [K}^{-1}\text{]}$	1.6 (20)	1.81 (200)	1.98 (580)	2.13 (1000)	2.23 (1200)	2.23 (1500)	2.33 (2000)	–	–
$\lambda \text{ [W/(m} \cdot \text{K)]}$	15.6 (20)	18.1 (200)	21.0 (400)	23.8 (600)	26.6 (800)	34.4 (1350)	35.0 (1393)	60 (1460)	–
$c_p \cdot 1e5 \text{ [J/kg} \cdot \text{K]}$	5.11 (20)	5.42 (200)	5.75 (400)	6.05 (600)	6.30 (800)	6.85 (1350)	7.30 (1427)	20.20 (1442)	50.00 (1460)

$$\begin{aligned}
 \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}, \theta) : \boldsymbol{\varepsilon}(\mathbf{v}) \, dx &= 0 \\
 \forall \mathbf{v} \in [H^1(\Omega)]^3, \\
 \int_{\Omega} \mathbf{q} \cdot \nabla q \, dx + \int_{\Omega} \{-3\alpha_T \kappa \operatorname{tr}[\dot{\boldsymbol{\varepsilon}}(\mathbf{u})] \theta - \rho c_\rho \dot{\theta}\} q \, dx &= 0 \\
 \forall q \in H^1(\Omega),
 \end{aligned} \tag{2}$$

with Dirichlet boundary conditions  $\mathbf{u} := (u_x, u_y, u_z) = \mathbf{u}_D$  on  $\partial\Omega_{D,u} \subset \partial\Omega$ . As already mentioned, the only heat entry is caused by the laser beam, which we implement as a volume of molten material with fixed melting temperature  $\theta_l$ . Therefore we have the additional constraint  $\theta := \theta_l \gg 0$  in  $\Omega_\theta \subset \bar{\Omega}$ . Let us note that the displacement is not fixed within  $\Omega_\theta$  and the material can still move. This is a nonlinear and nonsymmetric saddle point system for which the theoretical framework is still an object of studies, for example, an investigation of pairs of inf-sup-stable finite elements. A theoretical investigation of this problem is beyond the scope of this work.

### 3 Finite element discretization

The equations in (2) are linearized by applying the Newton method to the nonlinear boundary value problem. This yields, in each nonlinear iteration, a linear problem which spatially is discretized by using a mixed finite element method; see [11] for further details on the linearization. Next, we provide details on the mixed finite element discretization.

Let  $\tau_h$  be a uniform mesh of  $N_e$  hexahedral elements  $\Omega_e$  of  $\Omega$  with characteristic mesh size  $h$ . We introduce the conforming discrete piecewise linear displacement and temperature spaces

$$\begin{aligned}
 V^h &= V^h(\Omega) = \{\mathbf{u} \in [C^0(\bar{\Omega}) \cap H^1(\Omega)]^3 : \mathbf{u}|_T \in Q_1 \, \forall T \in \tau_h\}, \\
 Q^h &= Q^h(\Omega) = \{\theta \in C^0(\bar{\Omega}) \cap H^1(\Omega) : \theta|_T \in Q_1 \, \forall T \in \tau_h\}
 \end{aligned}$$

respectively, of Q1-Q1 mixed finite elements, where  $C^0(\bar{\Omega})$  denotes the space of continuous functions on  $\bar{\Omega}$  and  $H^1(\Omega)$  the usual Sobolev space. Here, the displacement space is discretized using the  $B_{bar}$  formulation to prevent the phenomenon of volumetric locking [31]. This technique modifies the discrete gradient operator by treating the volumetric and deviatoric strain components separately, with the volumetric part computed using reduced integration at the element centroid. This approach is particularly important for elastoplastic problems where volume-preserving plastic flow can introduce additional constraints that lead to locking phenomena. Since we do not have a stable theoretical framework, we can not ensure that this choice of finite elements satisfies an inf-sup condition [3]. As a time discretization we use a Backward Euler method with time step

$\Delta t$ . Therefore, in each time step, we have to solve a linearized system until the global absolute residual of Newton’s method falls below a fixed tolerance. The resulting discrete saddle point problem has the form

$$K \Delta \mathbf{d} = \begin{bmatrix} K_{uu} & K_{u\theta} \\ K_{\theta u} & K_{\theta\theta} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{d}_u \\ \Delta \mathbf{d}_\theta \end{bmatrix} = \begin{bmatrix} R_u \\ R_\theta \end{bmatrix} = R,$$

where  $\Delta \mathbf{d}_u$  and  $\Delta \mathbf{d}_\theta$  represent the Newton update for the displacement and temperature, and  $R_u$  and  $R_\theta$  the vectors of the residual, respectively. The block matrices are obtained by finite element assembly and we have

$$\begin{aligned}
 K_{uu} &= \mathbb{A}_{e=1}^{N_e} \left[ \int_{\Omega_e} B_u^T \mathbb{C} B_u \, dx \right], \\
 K_{u\theta} &= \mathbb{A}_{e=1}^{N_e} \left[ - \int_{\Omega_e} B_u^T (3\alpha_T \kappa \mathbf{1}) N_\theta \, dx \right], \\
 K_{\theta u} &= \mathbb{A}_{e=1}^{N_e} \left[ - \frac{1}{\Delta t} \int_{\Omega_e} \bar{\theta} N_\theta^T (3\alpha_T \kappa \mathbf{1}^T) B_u \, dx \right], \\
 K_{\theta\theta} &= \mathbb{A}_{e=1}^{N_e} \left[ - \int_{\Omega_e} B_\theta^T \boldsymbol{\lambda} B_\theta \, dx - \int_{\Omega_e} N_\theta^T \operatorname{tr}[\dot{\boldsymbol{\varepsilon}}(\bar{\mathbf{u}})] N_\theta \, dx \right. \\
 &\quad \left. - \frac{1}{\Delta t} \int_{\Omega_e} N_\theta^T \rho c_\rho N_\theta \, dx \right],
 \end{aligned}$$

where  $N_u$  and  $N_\theta$  are the finite element nodal basis functions for displacement and temperature,  $B_u$  and  $B_\theta$  denote their derivatives in tensor notation, and  $\bar{\mathbf{u}}$  and  $\bar{\theta}$  are the displacement and temperature components resulting from the previous Newton iteration. The nature of this linear system requires suitable preconditioners to accelerate the convergence of an iterative solver, as, in our case, the GMRES method. Our choice is to use two-level monolithic Schwarz type preconditioners.

## 4 Monolithic overlapping Schwarz preconditioners

Here, we exclusively consider Schwarz DDMs to precondition the linearized systems. They have the advantage that they can be used to directly precondition the original system without any modifications and it is additionally possible to add monolithic second levels or coarse spaces to obtain robustness and scalability.

### 4.1 Monolithic two-level Schwarz preconditioner with GDSW coarse space

More precisely, we consider a monolithic overlapping Schwarz domain decomposition preconditioner of GDSW (Generalized Dryja-Smith-Widlund) type for the saddle point problem. This is a two-level Schwarz preconditioner

where the coarse space and the local problems have the same block structure as the original one. We refer the interested reader to [7, 8] for a complete definition of the GDSW preconditioner for elliptic problems and to [12] for saddle point problems resulting from Stokes’ equations.

We introduce  $\{\Omega_i\}_{i=1}^N$ , a nonoverlapping decomposition of the domain  $\Omega$  into  $N$  subdomains of characteristic diameter  $H$ , and  $\{\Omega'_i\}_{i=1}^N$ , the corresponding overlapping domain decomposition with  $k$  finite element layers of overlap, generated by geometric or graph techniques. We define the interface  $\Gamma$  of the nonoverlapping domain decomposition  $\Gamma = \{x \in (\Omega_i \cap \Omega_j) \setminus \partial\Omega_D : i \neq j, 1 \leq i, j \leq N\}$ , that is divided into  $M$  connected components, specifically into  $M_V$  vertices,  $M_E$  edges, and  $M_F$  faces. Here, all interface nodes belonging to the same two subdomains build a face and all nodes belonging to the same set of more than just two subdomains build an edge. Finally, edges which consist of just one node are considered as vertices. These edges, faces, and vertices are used later on to define the basis functions of the GDSW coarse space. In general, the definition of an interface allows for the partitioning of the degrees of freedom (DOFs) into interface ( $\Gamma$ ) and interior ( $I$ ) ones and, after a reordering, an equivalent partitioning of  $K$ , that is,

$$K := \begin{bmatrix} K_{II} & K_{I\Gamma} \\ K_{\Gamma I} & K_{\Gamma\Gamma} \end{bmatrix}.$$

We further introduce the operators  $R_{u,i} : V^h \rightarrow V_i^h$  and  $R_{\theta,i} : Q^h \rightarrow Q_i^h$  for  $i = 1, \dots, N$ . These are restrictions from the global finite element spaces to the local finite element spaces defined on the overlapping subdomains. The monolithic restriction operators are then denoted by  $R_i : V^h \times Q^h \rightarrow V_i^h \times Q_i^h$  for  $i = 1, \dots, N$ , with

$$R_i := \begin{bmatrix} R_{u,i} & 0 \\ 0 & R_{\theta,i} \end{bmatrix}.$$

The transposed operators  $R_i^T, R_{u,i}^T$ , and  $R_{\theta,i}^T$  are the corresponding prolongation operators. Now, the additive monolithic two-level GDSW preconditioner can be written as

$$\hat{B}_{\text{GDSW}}^{-1} = \Phi K_0^{-1} \Phi^T + \sum_{i=1}^N R_i^T K_i^{-1} R_i,$$

where the local stiffness matrices  $K_i$  are extracted from  $K$  by

$$K_i = R_i K R_i^T, \quad \text{for } i = 1, \dots, N,$$

and the coarse operator is defined by

$$K_0 = \Phi^T K \Phi. \tag{3}$$

Here,  $\Phi$  is the matrix that contains the coarse space basis functions  $\phi^i, i = 1, \dots, N_\phi$ , column-wise. These functions are constructed as the discrete saddle point extensions of functions  $\phi_\Gamma^i$  defined on the interface by solving the linear system

$$\begin{bmatrix} K_{II} & K_{I\Gamma} \\ 0 & I \end{bmatrix} \begin{bmatrix} \phi_I^i \\ \phi_\Gamma^i \end{bmatrix} = \begin{bmatrix} 0 \\ \phi_\Gamma^i \end{bmatrix}, \tag{4}$$

where we decompose the interface values of the coarse basis into displacements ( $u_0$ ) and temperature ( $\theta_0$ ) as  $\phi_\Gamma^i = [\phi_{\Gamma,u_0}^i, \phi_{\Gamma,\theta_0}^i]^T$ . The entries of  $\phi_\Gamma^i$  are usually set to 1 or 0 and generally represent the restriction of the null space of the operators  $K_{uu}$  and  $K_{\theta\theta}$  on the interface components  $\Gamma_j$  with  $j = 1, \dots, M$ .

In three dimensions the null space of the  $K_{uu}$  block is given by the six rigid body modes (three translations and three rotations) defined on the interface components. For the  $K_{\theta\theta}$  operator, these consist of the functions that have constant temperature on  $\Omega$ , so we restrict the constant function 1 to vertices, edges, and faces.

Specifically, each node of our mesh can be associated to a vector with four components, where the first three entries represent the displacements and the last one the temperature. We introduce the vectors

$$\begin{aligned} v_{u,1} &= [1 \ 0 \ 0 \ 0]^T, & v_{u,2} &= [0 \ 1 \ 0 \ 0]^T, & v_{u,3} &= [0 \ 0 \ 1 \ 0]^T, \\ v_\theta &= [0 \ 0 \ 0 \ 1]^T \end{aligned} \tag{5}$$

where the first three are the translations into  $x$ -,  $y$ -, and  $z$ -directions and the last one is the restriction of the constant temperature on the node. Moreover we introduce the vectors

$$\begin{aligned} v_{u,4} &= [y \ -x \ 0 \ 0]^T, & v_{u,5} &= [-z \ 0 \ x \ 0]^T, \\ v_{u,6} &= [0 \ z \ -y \ 0]^T, \end{aligned} \tag{6}$$

that represent the three rotation with center placed in the origin  $(0, 0, 0)$ . For each vertex, edge, and face  $\Gamma_j, j = 1, \dots, M$ , of the interface  $\Gamma$ , we then construct four basis functions  $\phi_{\Gamma_j}^k, k = 1, \dots, 4$ , by filling the DOFs associated to the nodes of  $\Gamma_j$  with  $v_{u,k}$  for  $k = 1, 2, 3$  and  $v_\theta$  for  $k = 4$ . If rotations are used, we add  $\phi_{\Gamma_j}^k, k = 5, 6, 7$ , using  $v_{u,k-1}$  analogously. For the nodes in  $\Gamma \setminus \Gamma_j, \phi_{\Gamma_j}^k$  is set to zero. Finally, we collect all  $\phi_{\Gamma_j}^k, j = 1, \dots, M, k = 1, \dots, 4$ , as the columns of  $\Phi_\Gamma$  and thus obtain  $N_\phi = 4M$  coarse basis functions and columns in  $\Phi$ . In case of using rotations, we analogously obtain  $N_\phi = 7M$  coarse basis functions.

After having solved Eq. (4) for each basis function of the coarse space, the matrix  $\Phi$  has the following block representation

$$\Phi = \begin{bmatrix} \Phi_{u,u_0} & \Phi_{u,\theta_0} \\ \Phi_{\theta,u_0} & \Phi_{\theta,\theta_0} \end{bmatrix}. \quad (7)$$

With this construction, the presence of nonzero blocks on the off-diagonal can deteriorate the numerical scalability, especially if nonzero Dirichlet boundary conditions for the temperature are present (cf. [12]), as we have with the laser beam. Therefore, these blocks need to be removed, obtaining

$$\Phi = \begin{bmatrix} \Phi_{u,u_0} & 0 \\ 0 & \Phi_{\theta,\theta_0} \end{bmatrix}. \quad (8)$$

Let us remark that the extensions from the interface to the interior parts of the subdomains, that is, solving Equation (4), can be done in parallel in a perfectly scalable way, since  $K_{II}$  is a block diagonal matrix. The factorization of  $K_{II}$  can thus be done block-wise on each subdomain simultaneously using a sparse direct solver. Additionally, it is not necessary to compute the extension to all subdomains for all basis functions. For example, for the four or seven coarse basis functions belonging to a face, only the extension to the two neighboring subdomains are necessary, the extensions to all other subdomains are zero and must not be computed. Consequently, computing  $\Phi$  is a parallel scalable process.

## 4.2 GDSW\* and RGDSW coarse spaces

In the GDSW coarse space, the number of coarse basis functions depends on the number of components in the equivalence classes into which the interface between subdomains is divided—namely, vertices, edges, and faces. This number can grow rapidly, especially in 3D domain decomposition. To manage this complexity, alternative techniques such as GDSW\* and RGDSW have been introduced.

We briefly outline the ideas behind these alternative coarse spaces and refer to [13, 16] for further details. The key difference lies in how the components of the equivalence classes are treated. In the GDSW\* coarse space, unlike the original approach, only two interface components are used; the first one is obtained by each vertex along with its adjacent edges and the latter by faces. Since two adjacent vertices share an edge, the resulting interface components overlap over this edge. In order to maintain the robustness of these coarse spaces a partition of unity of the basis functions defined across the interface, according to (5) and (6) has to be preserved, therefore an inverse multiplicity scaling (Option 1 in Section 3 [9]) is applied.

Similarly, in the RGDSW coarse space, only one type of interface component is considered, that is each vertex along with its adjacent edges and faces. Again, multiplicity scaling is necessary to handle the overlap between interface components, which in this case happens for edges

and faces. Having partitioned the interface into edges, vertices, and faces, the coarse space size of GDSW is thus  $C \cdot (M_E + M_V + M_F)$ . The size of the GDSW\* coarse space is smaller with  $C \cdot (M_V + M_F)$  and finally the size of the RGDSW coarse space is the smallest one with only  $C \cdot M_V$  coarse basis functions. Here,  $C$  is either 4 or 7, depending on the usage of rotations.

Finally, we point out that for a two-dimensional domain decomposition of a three-dimensional problem, the GDSW and GDSW\* coarse spaces coincide, as no vertices are available in the decomposition.

## 4.3 Combination of coarse spaces

In Sects. 4.1 and 4.2, we explained the monolithic approach to construct GDSW, GDSW\*, and RGDSW coarse spaces for the complete saddle point problem. However, in this particular instance and for more general multi-field problems, it is feasible to amalgamate these approaches by independently selecting the most suitable one for each field. In fact, the sole requirement is to provide a partition of unity of the basis function across the interface. Examining the construction of these in eqs. (5) and (6) reveals that the problems in eq. (4) are independent of each other.

Let us give an explicit example for the construction of a GDSW\*(T)-RGDSW coarse space, that means to choose GDSW\* that includes only translations for the elastoplastic block and RGDSW for the thermal block. First of all, we note that in the matrix  $\Phi$  in (7) we have, columnwise represented,  $N_u$  and  $N_\theta$  basis function respectively associated to the displacements and temperature fields, with  $N_\phi = N_u + N_\theta$ . As we saw before, the number of coarse basis functions  $N_\phi$  depends on the number of components into which the interface is divided. The choice GDSW\*(T)-RGDSW means that the interface is split into  $M_V + M_F$  components for the elastoplastic block and into  $M_V$  components for the thermal block. Since in this coarse space, only translations are considered, we construct 3 interface basis functions for each interface component by using the vectors  $v_{u,i}, i = 1, \dots, 3$  in (5) and we then extend them to the subdomain by solving (4), obtaining  $N_u = 3 \cdot (M_V + M_F)$ . In a similar way we define the interface basis function for the temperature by only using the vector  $v_\theta$  and again we extend the solution to internal part again solving (4), obtaining  $N_\theta = M_V$ .

In this way, for instance in our particular problem, one could opt for a more extensive space for the elastoplastic block eventually deciding to introduce also the rotations and a more limited space for the thermal one. These available choices contribute to the flexibility of the preconditioners, enabling their design and adaptation for different problems, for example by leveraging the most economical coarse space

when feasible and reserving the more costly one where it is necessary. As we will show in the numerical results, a more extensive selection for the elastoplastic block is required in our particular problem, while the more limited coarse space can be utilized for the temperature.

## 5 Implementation

All Schwarz preconditioners and GDSW-type preconditioners discussed here are incorporated into the FE2TI software package, a C/C++ library that utilizes a hybrid MPI/OpenMP [6] parallelization strategy, leveraging the PETSc library [1]. To enhance its capabilities, we have integrated an interface with the FEAP library [32], allowing access to FEAP's finite element formulations and material models. FE2TI has already demonstrated an excellent strong and weak scalability behavior, efficiently handling simulations with over one million parallel ranks for realistic dual-phase steel deformation scenarios; for further details, see [21, 23]. In FE2TI, efficient implementations of the domain decomposition approaches BDDC (Balancing Domain Decomposition by Constraints) and FETI-DP (Finite Element Tearing and Interconnecting - Dual Primal) are already integrated (see [14, 21, 22]) and we were able to reuse some of the efficient building blocks for the new Schwarz preconditioners. For example, faces, edges, and vertices of the domain decomposition are identified in a scalable way using the implementation from our BDDC implementation.

Within the Schwarz preconditioners, our software defaults to the more efficient restricted option [10] with an overlap of one layer of finite elements for the first level, although the classical additive alternative is also available. Regarding the second (or coarse) level, several options are supported, that are, GDSW, GDSW\*, and RGDSW. While this work focuses on solving a thermo-elastoplasticity problem, that is, a two-by-two block system, we note that our coarse solver implementation is also applicable to general multifield problems. In fact, as described in Sect. 4.3, we can select different approaches for each field, allowing for flexible combinations in the choice of the coarse space to better adapt to the nature of the problem. We provide a detailed comparison of different combinations in our numerical results.

The coarse basis functions are constructed by solving local problems and then assembled into an MPI parallel matrix  $\Phi$  (see Equation (8)) shared among all the ranks. During this construction the truncation tolerance is applied by zeroing out small entries of  $\Phi$ . The coarse matrix  $K_0$  (see (3)) is then obtained by using the PtAP product from PETSc. However, the resulting parallel matrix is spread over all the MPI ranks, which makes its factorization using a sparse

direct solver inefficient. Therefore  $K_0$  is then redistributed only among a reduced number of MPI ranks within a separate sub-communicator. Since the coarse space is typically expensive to compute, we have incorporated the ability to recycle it after its initial computation, enabling a reuse in subsequent applications of the preconditioner across different linear systems. This feature is particularly beneficial in cases such as ours, where a Newton method is applied at each time step, requiring the solution of multiple spectrally similar linear systems. Different strategies for the recycling can be applied: one can decide to reuse the matrix  $\Phi$  and recompute  $K_0$  and its factorization, or one can recycle all,  $\Phi$ ,  $K_0$ , and its factorization. Clearly, the cheapest option is to reuse both basis functions and coarse matrix, saving an extra factorization. This leads to a slightly worse approximation of the preconditioner that can lead to an increasing number in the GMRES iteration in later Newton iterations. Anyway, we will see in the numerical results that this effect is minimal. Therefore, here we only consider the complete recycling.

All local problems are solved exactly using the sparse direct solver package PARDISO from the Math Kernel Library (MKL) [20, 28], while the coarse problem is solved using MKL Cluster PARDISO (see Sect. 6.1). Therefore, the coarse matrix is distributed to the ranks of an MPI sub-communicator of a reasonable size.

In summary, we employ the highly efficient GMRES solver from PETSc to iteratively solve large sparse linear systems. To ensure controlled convergence, we use our highly scalable implementation of Schwarz preconditioners with GDSW-type coarse spaces. Additionally, our interface to FEAP enables seamless integration with the models described in Sect. 3.

## 6 Numerical results

In our numerical simulations, we consider thermo-elastoplasticity problems that emulate real experiments in which a steel plate of domain  $\Omega = [0, l_x] \times [0, l_y] \times [0, l_z]$ , with  $l_x, l_y, l_z > 0$ , is melted by a laser beam. As anticipated in Sect. 2, the effect of the laser is modeled by fixing the temperature field in the melting pool. This is enforced as a gradient with a ratio of  $\dot{\theta}_l = 14\,400^\circ\text{C}/\text{s}$ . Specifically, at the beginning of the simulation each DOF is initialized with a temperature  $\theta_0 = 20^\circ\text{C}$ , then in each time step we identify the DOFs that are located inside the region heated by the laser, which is modeled as a cylinder of radius  $2\text{ mm}$  passing vertically through the entire material. The temperature of these latter DOFs is then increased by a value  $\dot{\theta}_l \cdot \Delta t$  until the maximum value of  $\theta_l = 1460^\circ\text{C}$  is reached. We note that, this heating procedure (initialization phase),

where the laser remains fixed in its initial position, takes a total of 0.1s and will not be completed during the first set of experiments, since only few first time iterations are taken into account. This aspect is relevant for the simulations in Sects. 6.6 and 6.7, where after the initialization phase the laser moves along the plate.

For the displacements, additional Dirichlet boundary conditions are applied to simulate an external load acting on the plate during the welding process. To avoid higher localization of the strain in the corners of the domain, the boundary conditions are applied in the following way: on the surface with  $y = 0$ , we prevent the movement along the  $y$  axis by enforcing  $u_y = 0$  and the rotation of the material by setting  $u_x = 0$  along the  $z$  axis and  $u_z = 0$  along the  $x$  axis. The external load is then applied to the face with  $y = l_y$  by enforcing  $u_y = u_D(t)$ , defined as  $u_D(t) := \min(\varepsilon_{22}, \dot{\varepsilon}_{22} \cdot t)$  with the external strain  $\varepsilon_{22}$  and strain rate  $\dot{\varepsilon}_{22}$  specified before the simulations. We consider two different setups for our simulations. First, to validate our implementation and assess the performance of the solver, we analyze a steel cube subjected to both the laser and an external load using a 3D domain decomposition strategy (Fig. 1), that is, a decomposition into  $N \times N \times N$  cubical subdomains. Once the optimal solver is identified, we apply it to a more realistic case involving a thin steel plate using 2D domain decomposition, that is, a decomposition into  $N \times N$  thin subdomains. For an application of our solver to reproduce a physically realistic experiment, we refer the interested reader to [4].

If not explicitly specified, by default all the simulations use a time step size of  $\Delta t = 0.001$  s and the external strain is applied from the beginning of the simulation with  $\varepsilon_{22} = 0.03$  and  $\dot{\varepsilon}_{22} = 0.06$ . For each time step, Newton's method is applied until the global residual falls below an absolute tolerance of  $10^{-4}$ , while, for the linear systems, the GMRES method is employed, with a stopping criterion of either a relative residual error of  $10^{-6}$  or an absolute residual error of  $10^{-10}$  based on the unpreconditioned norm. In our numerical results and tables we report the following statistics:  $it_{GMRES}$  is the average count of GMRES iteration over all the linear systems,  $it_N$  is the average count of Newton iteration per time step,  $T_{PC}$  is the total time to assemble the preconditioner,  $T_{Ass}$  is the total time to assemble the linear systems,  $T_{Sol}$  is the total time to solve the linear systems,  $T_{Tot}$  is the total time for the simulation,  $T_{N_{avg}}$  is the

average time per Newton iteration,  $T_{Tot}/\#Timeit$  is the average time per time iteration,  $P_E$  is the parallel efficiency,  $Cores$  is the number of cores and MPI ranks used,  $Subcomm$  is the number of MPI ranks in the sub-communicator used for the coarse solve,  $PCType$  is the type of preconditioner used,  $(T + R)$  indicates that in the coarse space for the elastoplasticity field translations and rotations are included while  $(T)$  indicates that only translations are considered.

## 6.1 General solver setup

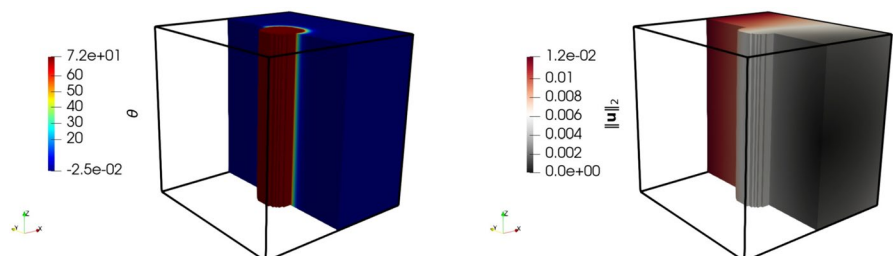
In this section we exclusively consider the simplified LBW problem on a cube geometry to find out the optimal setup for our Schwarz preconditioner. Before comparing and analyzing the performance of different combinations of GDSW-type coarse spaces to identify the optimal one, we first investigate two key aspects of the solver that will remain fixed afterwards.

One of the main challenges of GDSW-type preconditioners is the rapid growth of the coarse space, which results in a large coarse matrix which tends to be much denser than a typical finite element stiffness matrix. This growth significantly increases the computational cost of the coarse factorization using a sparse direct solver, especially since a forward-backward substitution has to be applied in each GMRES iteration. To mitigate this issue, we can adjust the construction of the coarse matrix. Recall that the coarse matrix is given by  $K_0 = \Phi^T K \Phi$ .

A first strategy to improve efficiency is to remove small entries in the coarse basis matrix  $\Phi$  that fall below a certain truncation tolerance. This results in a sparser and more computationally efficient coarse matrix. To evaluate this approach, we consider the GDSW\*(T+R)-RGDSW coarse space and perform five iteration steps. In Table 2, we analyze the solver's behavior for different threshold values applied to the entries of  $\Phi$ . Our results indicate that we can be fairly aggressive, using a relatively high tolerance of  $1e - 4$  without increasing the iteration count. This significantly reduces both the cost of preconditioner construction and the overall solving time. However, as expected, setting the truncation tolerance too high degrades the accuracy of the coarse space, making it a poor approximation.

Additionally, in Table 3, we report the timings for the same experiment as before fixing a truncation tolerance of

**Fig. 1** Section of temperature field (left) and norm of the displacements (right) for the cube test



**Table 2** Solver performance with respect to truncation tolerance.  $10 \times 10 \times 10$  subdomains with  $20 \times 20 \times 20$  finite elements. All values in  $\Phi$  which are lower than the given tolerance are set to zero before building the coarse matrix. All times are given in seconds

Tolerance	$it_{GMRES}$	$it_N$	$T_{PC}$	$T_{Sol}$	$T_{Tot}$
0	18.9	3	425.5	82.3	548.6
$1e-4$	18.9	3	326.2	58.5	416.8
$1e-2$	42.3	3	314.3	128.8	475.4

$1e-4$  while varying the number of computational cores and the number of MPI ranks in the sub-communicator for the coarse solve. As expected, solving the coarse problem on a single MPI rank is not feasible, especially as the number of MPI processes for the global problem increases, leading to a larger coarse space. Based on the results in Table 3, we determine the optimal number of MPI ranks within the coarse sub-communicator for all further tests. The best setup for the different number of subdomains is marked in bold face in Table 3.

### 6.2 Comparison of different coarse spaces

Fixing the optimal basic setup from the previous section, that is, truncating  $\Phi$  with a tolerance of  $1e-4$  and choosing the optimal size of the coarse sub-communicator, we now compare different coarse spaces. We again consider five time iteration steps of the thermo-elastoplastic problem on the cube by using  $10 \times 10 \times 10$  cores and subdomains with

$20 \times 20 \times 20$  finite elements each. In Table 4 we report the effect of different coarse space choices on the solver performance in terms of the average iteration counts in each time step and the total timings. We recall that the first coarse space is used for the elastoplastic part of the coupled system, while the second one for the temperature, for example, GDSW-RGDSW means that we use GDSW for the mechanics and RGDSW for the temperature. We also investigate the choice to include only translations (T) or both translations and rotations (T+R) in the coarse space for the elastoplastic block.

At first sight, we note that, as expected, the time to construct the preconditioner increases with the size of the coarse space. Additionally, we observe that including rotational modes significantly improves convergence by reducing the number of GMRES iterations. This pattern is observed across all configurations, confirming that rotations enhance the preconditioners effectiveness. However, these improvements in iteration counts come at the cost of an increment of the coarse space size, which affects the setup time of the preconditioner as well as the solution time. Among the configurations that include rotations, GDSW\*(T+R)-RGDSW provides the best trade-off, maintaining a moderate coarse space size (21 303) while achieving one of the lowest total solution times (416.5s). On the other hand, RGDSW(T+R)-RGDSW, despite having the smallest coarse space (5 103), suffers from a significantly higher iteration count (48.6),

**Table 3** Solver performance with respect to the number of subcommunicator used for the coarse solver. Each subdomain contains  $10 \times 10 \times 10$  finite elements. All values in  $\Phi$  which are lower than  $1e-4$  are truncated to zero before building the coarse matrix. All times are given in seconds

Cores (Coarse)	512(10 465)		1 728(37 730)		4 096(92 610)	
	$T_{PC+Sol}$	$T_{Tot}$	$T_{PC+Sol}$	$T_{Tot}$	$T_{PC+Sol}$	$T_{Tot}$
Subcomm						
1	49.5	54.4	211.2	216.9	x	x
4	37.7	42.4	103.1	108.3	247.3	254.9
8	35.5	40.2	85.1	90.3	159.5	206.0
16	35.7	40.4	76.7	81.8	173.7	181.8
32	<b>35.3</b>	<b>40.0</b>	<b>68.1</b>	<b>73.5</b>	157.5	164.6
64	37.1	42.0	85.4	96.0	<b>150.4</b>	<b>157.6</b>

Bold values represent optimal performance for each configuration

**Table 4** Solver performance with respect to different choices for the coarse space.  $10 \times 10 \times 10$  subdomains with  $20 \times 20 \times 20$  finite elements. The truncation tolerance is set to  $1e-4$ . All times are given in seconds

No recycling						
PCType	Coarse	$it_{GMRES}$	$it_N$	$T_{PC}$	$T_{Sol}$	$T_{Tot}$
GDSW(T+R)-GDSW	36 225	18.8	3	481.3	68.4	582.2
GDSW(T+R)-RGDSW	31 266	18.8	3	443.7	65.1	540.6
GDSW*(T+R)-GDSW*	23 954	18.9	3	338.5	60.1	430.5
GDSW*(T+R)-RGDSW	21 303	18.8	3	326.2	58.5	416.5
RGDSW(T+R)-RGDSW	5 103	48.6	3	254.2	133.1	418.9
GDSW(T)-GDSW	23 265	29.4	3	342.8	90.0	465.1
GDSW(T)-RGDSW	18 306	29.1	3	309.2	86.8	427.6
GDSW*(T)-GDSW*	13 667	29.5	3	259.6	83.4	374.7
GDSW*(T)-RGDSW	11 016	29.7	3	249.2	82.5	363.2
RGDSW(T)-RGDSW	2 916	53.8	3	216.7	142.9	391.4

leading to an overall longer solution time (418.9s), regardless its lower setup cost.

A similar trend is observed in the translation-only (T) configurations. While reducing the coarse space leads to lower preconditioner setup times, the number of iterations increases, impacting the overall efficiency. For example, GDSW\*(T)-RGDSW, with a coarse space size of 11 016, takes 29.7 iterations and has a total computation time of 363.2s, which is an improvement over larger coarse spaces and its (T+R) counterpart, despite having a higher number of iterations. So far, these results indicate that using only translations appear to be a valid option. Especially when reducing the coarse space size has priority. Often, even if we have an increased number of iterations, the total time to solution is lower. Overall, GDSW(T)-RGDSW emerges as the most efficient configuration, achieving a strong balance between coarse space size, iteration count, and total computational time.

### 6.3 Recycling option

In Table 5, we further analyze solver performance repeating the previous test but now activating the recycling option for the coarse operator and the coarse basis functions. A clear improvement is observed across all configurations, particularly regarding the preconditioner setup time, which is significantly reduced compared to Table 4. We recall that with this approach, in each time step, the coarse space is constructed only once and in the subsequent Newton iterations within the same time step only the sparse direct solvers for the local subdomains have to be updated. This strategy mostly affects the setup time for the preconditioner, for instance, with GDSW\*-RGDSW (T+R) we have a reduction in the setup time from 326.2s to 178.6s and at the same time its total solution time drops from 416.5s to 268.9s. We stress out that, by recycling the coarse space and the basis functions, we are not using anymore the exact two-level Schwarz preconditioner, but an approximation. This is

highlighted by the increase in iteration count (18.8 to 18.9 iterations), that anyway is minimal. This trend is consistent across other configurations, confirming that recycling effectively mitigates the computational overhead associated with the coarse space factorization while maintaining solver robustness.

Since, as already mentioned, the most expensive part of the setup of the preconditioner is the construction of the coarse space, the benefits of the recycling are more evident especially when the plastic effect is stronger and, as a result, the number of Newton iterations per time step is increased. Moreover, since the coarse space is assembled just once, i.e. in the first Newton iteration, the time spent to assemble the preconditioner in the further ones is affected only by the direct solver applied on the subdomain, levelling the timings also for the most expensive coarse space. In that case the time for setting up the preconditioner loses importance, and thus solver setups with fast times the solution get more and more favorable. The results confirm that GDSW\*-RGDSW (T+R) remains the optimal configuration, particularly when combined with recycling, as it provides the best trade-off between coarse space size, computational cost, and solver efficiency.

### 6.4 Strong scalability

Table 6 and Fig. 2 (Left) provide the strong scalability results for the GDSW\*-RGDSW coarse space, with and without the recycling option. We performed the test for a mesh of 12 194 500 DOFs over the first 5 time iteration steps. To increase the plasticity effect we apply an external strain of  $\varepsilon_{22} = 0.06$  with a strain rate of  $\dot{\varepsilon}_{22} = 0.12$ .

In both cases, as the number of cores increases from 64 to 2048, the total time to solution significantly decreases, demonstrating effective parallelization where the significant improvement from 64 to 216 is mainly due to the reduced sizes of the subdomains. However, the parallel efficiency shows a decline at higher core counts from 2 048 to 4 096

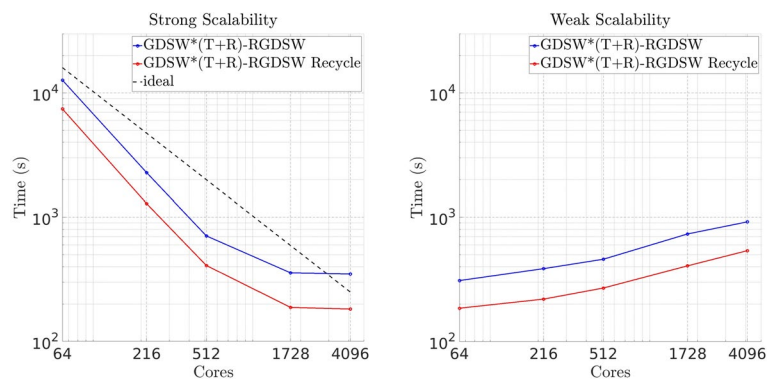
**Table 5** Solver performance with respect to different choices for the coarse space with the recycling option active.  $10 \times 10 \times 10$  subdomains with  $20 \times 20 \times 20$  finite elements. The truncation tolerance is set to  $1e - 4$ . All times are given in seconds

With recycling						
<i>PCT</i> type	<i>Coarse</i>	<i>it<sub>GMRES</sub></i>	<i>it<sub>N</sub></i>	<i>T<sub>PC</sub></i>	<i>T<sub>Sol</sub></i>	<i>T<sub>Tot</sub></i>
GDSW(T+R)-GDSW	36225	18.9	3	230.6	68.9	331.3
GDSW(T+R)-RGDSW	31266	18.9	3	217.8	65.5	314.8
GDSW*(T+R)-GDSW*	23954	19.1	3	183.0	60.8	275.2
GDSW*(T+R)-RGDSW	21303	18.9	3	178.6	58.9	268.9
RGDSW(T+R)-RGDSW	5103	48.7	3	154.4	134.3	320.1
GDSW(T)-GDSW	23265	29.8	3	183.9	92.8	308.6
GDSW(T)-RGDSW	18306	29.1	3	172.9	87.3	291.5
GDSW*(T)-GDSW*	13667	29.9	3	156.2	85.0	272.5
GDSW*(T)-RGDSW	11016	29.7	3	152.6	83.3	267.0
RGDSW(T)-RGDSW	2916	53.8	3	141.9	143.8	317.2

**Table 6** Strong scalability for GSDW\*(T+R)-RGDSW without recycling option (top) and with recycling option (bottom). 12 194 500 DOFs. Computed the first 5 time iterations. All times are given in seconds

No recycling								
Cores	Coarse	$it_{GMRES}$	$it_N$	$T_{PC}$	$T_{Sol}$	$T_{Ass}$	$T_{Tot}$	$P_E$
64	1 053	29.2	7.6	10 898	1 512	251.1	12 681	1.00
216	4 115	27.4	7.6	1 810	348.6	116.1	2 282	2.77
512	1 0465	27.0	7.6	517.2	139.0	50.4	708.4	2.23
1728	37 730	27.0	7.8	232.2	107.4	15.7	356.5	1.32
4096	92 610	26.6	7.8	213.7	126.2	7.8	349.5	0.56
With recycling								
Cores	Coarse	$it_{GMRES}$	$it_N$	$T_{PC}$	$T_{Sol}$	$T_{Ass}$	$T_{Tot}$	$P_E$
64	1 053	29.6	7.6	5 643	1 519	250.1	7 431	1.00
216	4 115	27.9	7.6	810.1	356.6	115.2	1 288	2.88
512	10 465	27.9	7.6	213.9	144.7	49.8	409.5	2.27
1728	37 730	28.3	7.8	59.5	112.2	15.6	187.9	1.46
4096	92 610	28.6	7.8	39.1	135.2	7.8	182.2	0.63

**Fig. 2** Strong and weak scalability for GSDW\*-RGDSW with and without recycling option over 5 time iterations



due to the huge size of the coarse space that reflects into communication overhead and load imbalance. Notably, the recycling option improves performance, reducing the preconditioner computation time by a significant factor, that contributes to a faster solution. While both configurations exhibit strong scalability, we notice that the recycling option does not affect the number of the iterations and does not increase substantially the solving time. This yields better parallel efficiency and reduced computational cost, making it a favorable choice for large-scale simulations.

### 6.5 Weak scalability

With the same simulation set up of the Sect. 6.4, in Table 7 and Fig. 2 (Right) we report a weak scalability test again performed with GSDW\*(T+R)-RGDSW coarse space. We keep fixed the subdomain sizes with  $16 \times 16 \times 16$  elements and we analyze the performance when increasing proportionally with the number of cores. Both configurations exhibit a gradual increase in the computational time, with a more pronounced growth in the case without the recycling option.

Notably, the recycling option improves performance across all core counts, significantly reducing the

preconditioner time. We note in fact, that from 64 to 4096 cores this increases by less than a factor of 2. Time for assembling the linear systems shows almost perfect weak scalability, while the time to compute the solution does not scale due to the increasing number of both, iteration per linear system and Newton iteration per time step.

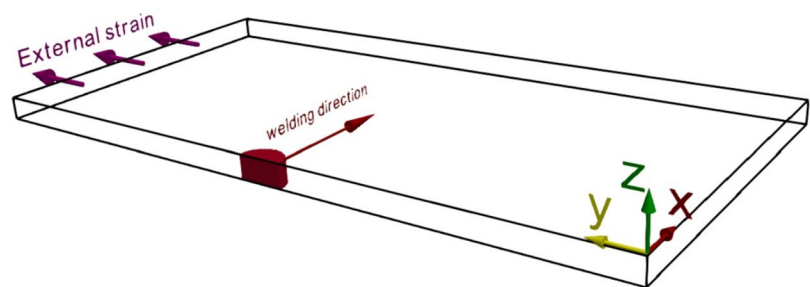
For the reasons just described above, the average time per Newton iteration  $T_{N_{avg}}$  increase with the number of the subdomains. Despite this, the recycling strategy appears significantly faster, indicating improved efficiency in the iterative solution process. Considering the time per Newton iteration, we have a parallel efficiency of 46% on 4096 cores. This is an effect of slightly increasing GMRES iteration counts per linear solve and an increase of the size of the coarse space. Considering instead the time per GMRES iteration, we obtain an excellent parallel efficiency of 79%, which is satisfying due to the increasing coarse space size.

### 6.6 2D domain decomposition

To replicate the practical application of a LBW process, we perform a series of numerical simulations on a thin plate using a 2D domain decomposition approach. The plate has dimensions  $30 \times 15 \times 1$ , and the simulation runs for 0.13s

**Table 7** Weak scalability for GSDW\*(T+R)-RGDSW without recycling option (top) and with recycling option (bottom).  $16 \times 16 \times 16$  elements per subdomain. Computed the first 5 time iterations. All times are given in seconds

No recycling								
Cores	Coarse	$it_{GMRES}$	$it_N$	$T_{PC}$	$T_{Sol}$	$T_{Ass}$	$T_{Tot}$	$T_{N_{avg}}$
64	1 053	21.2	6.6	225	52	29.1	309.7	9.4
216	4 115	23.4	7.2	279.4	70.3	31.3	385.4	10.7
512	10 465	25.6	7.8	320.2	95.5	41.4	459.6	11.8
1728	37 730	30.9	8.4	494.4	195.9	42.6	734.2	17.5
4096	92 610	34.7	8.8	567.8	299.7	47.2	918.4	20.9
With recycling								
Cores	Coarse	$it_{GMRES}$	$it_N$	$T_{PC}$	$T_{Sol}$	$T_{Ass}$	$T_{Tot}$	$T_{N_{avg}}$
64	1 053	21.5	6.6	100.6	53.4	29.3	185.4	5.6
216	4 115	23.9	7.2	114.1	71.9	30.7	219.1	6.1
512	10 465	26.4	7.8	129.4	98.5	41.0	269.3	6.9
1728	37 730	32.4	8.4	159.5	203.6	42.5	405.8	9.7
4096	92 610	36.8	8.8	171.6	317.5	47.2	538.2	12.2

**Fig. 3** Sketch of the thin plate geometry and displacements boundary conditions. In red the melting pool, initially placed with the center on the edge of the plate**Table 8** Solver performance with respect to different choices for the coarse space.  $16 \times 64 \times 1$  subdomains with  $8 \times 4 \times 9$  finite elements. The truncation tolerance is set to  $1e - 4$ . All times are given in seconds

PCType	Coarse	$it_{GMRES}$	$it_N$	$T_{PC}$	$T_{Sol}$	$T_{Tot}$
GDSW(T+R)-GDSW	19 445	28.8	6.5	159.4	569.2	805.7
GDSW(T+R)-RGDSW	17 478	28.2	6.5	147.5	501.2	722
RGDSW(T+R)-RGDSW	3 777	294.3	6.5	114.7	3 590	3 779
GDSW(T)-GDSW	1 1642	35.4	6.5	128.5	496.3	700.2
GDSW(T)-RGDSW	9 681	35.4	6.5	120.3	459.6	654.7
RGDSW(T)-RGDSW	6 615	302.9	6.5	107.5	3 222	3 405

real time. Initially, during the first 0.1s of the initialization phase the laser is positioned at the edge of the plate (see Fig. 3) and no external strain is applied. After this, the laser moves along the x-direction at a velocity of  $16.67 \text{ mm/s}$  for 0.2s, while an external strain is enforced with  $\varepsilon_{22} = 0.03$  and a strain rate of  $\dot{\varepsilon}_{22} = 0.06$ .

Table 8 presents a comparison of different coarse spaces in this experiment, with the recycling option enabled. The results indicate that the GDSW-RGDSW approach, both with and without rotation, provides the best balance between computational cost and iteration count. As expected, the RGDSW(T)-RGDSW and RGDSW(T+R)-RGDSW configurations exhibit a significant increase in GMRES iterations, leading to longer solving times. Overall, even if GDSW(T)-RGDSW has slightly more GMRES iterations than its rotational counterpart, it stands out as the most efficient configuration. Its superior performance, combined

with a smaller coarse space that is half the size, makes it the optimal choice.

For this choice of the coarse space we perform two strong scalability tests reported in Table 9 and Fig. 4. The two test are both computed with the same setup described just before using a finer discretization in space, namely 9 492 552 and 30 790 760 DOFs. The smaller problem shows a perfect scalability up to 2 048 cores and then deteriorates when switching to 4 096 cores with a parallel efficiency of 55%, due to the coarse space that affect mainly the solution time. As showed in the same table, when moving to a higher number of DOFs, we observe an excellent parallel efficiency of 76% even with the larger number of subdomains. This is a well known phenomena in DDMs, since strong scalability always deteriorates when the coarse space size gets large in comparison to the subdomain sizes. In fact, a direct solver application of the coarse level becomes prohibitive, due

**Table 9** Strong scalability for GDSW(T)-RGDSW with recycling option over 130 time iterations

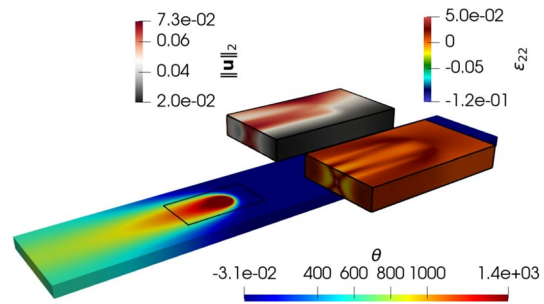
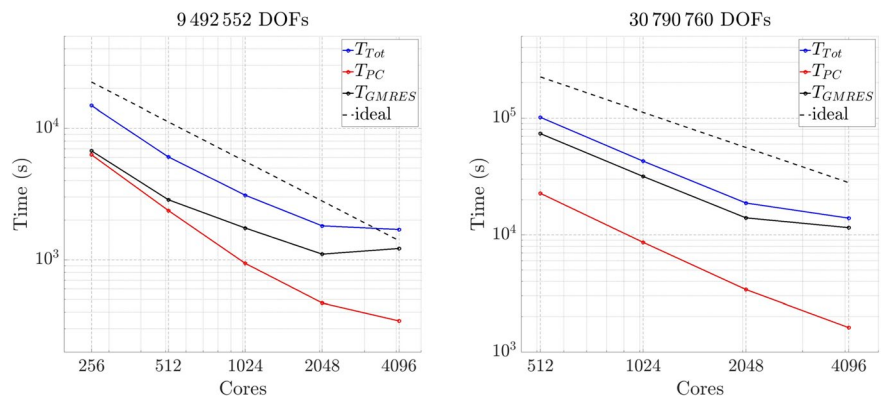
9 492 552		DOFs		
Cores	$T_{tot}(T_{PC} + T_{Sol})$	$it_{GMRES}$	$T_{Tot}/\#T_{inP}$	$\frac{T_{Tot}}{\#T_{inP}}$
256	14 846(6 295+6 740)	49.7	114.2	1.00
512	6 061(2 371+2 863)	40.9	46.6	1.20
1024	3 099(939+1 744)	44.7	23.8	1.19
2048	1 812(470+1 104)	39.2	13.9	1.01
4096	1 700(344+1 219)	42.2	13.1	0.55
30 790 760		DOFs		
Cores	$T_{tot}(T_{PC} + T_{Sol})$	$it_{GMRES}$	$T_{Tot}/\#T_{inP}$	$\frac{T_{Tot}}{\#T_{inP}}$
512	10 1647(22 691+73 668)	188.5	781.9	1.00
1024	42 789(8 633+31 642)	137.0	329.1	1.18
2048	18 739(3 404+13 985)	114.7	144.1	1.35
4096	13 145(1 613+11 532)	113.2	107.0	0.91

to the superlinear memory complexity of the sparse direct solvers. In such cases, on the coarse space can be recursively applied a GDSW-type preconditioner, resulting in a three level method [15, 17].

### 6.7 Application to LBW simulation

In this section we mimic a physically realistic simulation of a LBW process. In particular we reproduce a simplified CTW test to assess the weldability of thin plates in the LBW process. We choose a plate of size  $44.6 \times 8 \times 1\text{mm}$  (see Fig. 5) and we split the domain into  $128 \times 32 \times 1$  subdomain, each made of  $7 \times 5 \times 20$  finite elements for a total of 12 131 028 DOFs. As before the laser is placed with the center on the edge of the plate, and no external strain is applied neither during the 0.1s of the initialization phase nor during the following 0.7s. The external strain is then applied for 0.3s with  $\epsilon_{22} = 0.03$  and  $\dot{\epsilon}_{22} = 0.03$ , leading to a total of 1.0s real time until the end of the simulation. With a time step  $\Delta t = 0.001$  this corresponds to 1000 iterations. This simulation, performed with the solver previously described, i.e. GMRES methods preconditioned with a two-level additive Schwarz preconditioner with GDSW(T)-RGDSW coarse space, with the recycling option active, and a truncation tolerance of  $1e - 4$ . The coarse solver is performed

**Fig. 4** Strong scalability for GDSW(T)-RGDSW with recycling option over 130 time iterations



**Fig. 5** Distribution of the temperature field  $\theta$ , the norm of the displacements  $\|u\|_2$  and the strain component into the direction of the external strain applied  $\epsilon_{22}$  at the end of the simulation

on 64 MPI ranks using cluster PARDISO. The simulation ran a total of 40 684s of runtime. This approximately corresponds to 11 hours and 20 minutes of computation with 40.6s per time iteration.

## 7 Conclusions

We have introduced different variants of monolithic GDSW coarse spaces for thermo-elastoplastic LBW problems as well as a new and efficient parallel implementation in PETSc. Overall, the suggested preconditioners show a very high robustness and a good parallel scalability, which enables large scale simulations of fully-coupled thermo-mechanical simulations in the future. As a result of our numerical tests, the best candidate is a combination of GDSW or GDSW\* for the mechanical problem with an RGDSW variant of reduced size for the temperature field.

**Acknowledgements** This project has received funding from the Deutsche Forschungsgemeinschaft (DFG) as part of the Forschungsgruppe (Research Unit) 5134 “Solidification Cracks During Laser Beam Welding– High Performance Computing for High Performance Processing” under DFG project number 434946896. The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project k109be10. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU

hardware is partially funded by the German Research Foundation (DFG) - 440719683. We would like to thank our project partners from Forschungsgruppe 5134 L. Scheunemann, J. Schröder, and P. Hartwig for support with the interface to FEAP and providing the thermo-elasto-plasticity formulation and all parameters in FEAP.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## Declarations

**Conflict of interest** This project has received funding from the Deutsche Forschungsgemeinschaft (DFG) as part of the Forschungsgruppe (Research Unit) 5134 “Solidification Cracks During Laser Beam Welding– High Performance Computing for High Performance Processing” under DFG project number 434946896.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Balay S, Abhyankar S, Adams MF, Benson S, Brown J, Brune P, Buschelman K, Constantinescu E, Dalcin L, Dener A, Eijkhout V, Faibussovich J, Gropp WD, Hapla V, Isaac T, Jolivet P, Karpeev D, Kaushik D, Knepley MG, Kong F, Kruger S, May DA, McInnes LC, Mills RT, Mitchell L, Munson T, Roman JE, Rupp K, Sanan P, Sarich J, Smith BF, Suh H, Zampini S, Zhang H, Zhang H, Zhang J (2024) PETSc/TAO users manual. Technical Report ANL-21/39 - Revision 3.22, Argonne National Laboratory. <https://doi.org/10.2172/2205494>
- Bakir N, Artinov A, Gumenyuk A, Bachmann M, Rethmeier M (2018) Numerical Simulation on the Origin of Solidification Cracking in Laser Welded Thick-Walled Structures. *Metals* 8(6):406. <https://doi.org/10.3390/met8060406>
- Boffi D, Brezzi F, Fortin M et al (2013) *Mixed Finite Element Methods and Applications* vol. 44. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-36519-5>
- Bevilacqua T, Gumenyuk A, Habibi N, Hartwig P, Klawonn A, Lanser M, Rethmeier M, Scheunemann L, Schröder J (2025) Large-scale thermo-mechanical simulation of laser beam welding using high-performance computing: A qualitative reproduction of experimental results. arXiv preprint [arXiv:2503.09345](https://arxiv.org/abs/2503.09345)
- Bevilacqua T, Klawonn A, Lanser M, Wasiak A (2025) Monolithic overlapping Schwarz preconditioners for nonlinear finite element simulations of laser beam welding processes. In: Marmo, F., Cuomo, S., Cutolo, A. (eds.) *Computational Mechanics and Applied Mathematics: Perspectives from Young Scholars*, pp. 123–133. Springer, Cham. [https://doi.org/10.1007/978-3-031-76591-9\\_12](https://doi.org/10.1007/978-3-031-76591-9_12)
- Chandra R (2001) *Parallel Programming in OpenMP*. Morgan Kaufmann San Francisco CA United States. <https://doi.org/10.5555/355074>
- Dohrmann CR, Klawonn A, Widlund OB (2008) Domain decomposition for less regular subdomains: Overlapping Schwarz in two dimensions. *SIAM J Numer Anal* 46(4):2153–2168. <https://doi.org/10.1137/070685841>
- Dohrmann CR, Klawonn A, Widlund OB (2008) A Family of Energy Minimizing Coarse Spaces for Overlapping Schwarz Preconditioners. In: *Domain Decomposition Methods in Science and Engineering XVII*, pp. 247–254. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-75199-1\\_28](https://doi.org/10.1007/978-3-540-75199-1_28)
- Dohrmann CR, Widlund OB (2017) On the design of small coarse spaces for domain decomposition algorithms. *SIAM J Sci Comput* 39(4):1466–1488. <https://doi.org/10.1137/17M11427>
- Frommer A, Szyld DB (2001) An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM J Numer Anal* 39(2):463–479. <https://doi.org/10.1137/S0036142900370824>
- Hartwig P, Bakir N, Scheunemann L, Gumenyuk A, Schröder J, Rethmeier M (2024) A Physically Motivated Heat Source Model for Laser Beam Welding. *Metals* 14(4):430. <https://doi.org/10.3390/met14040430>
- Heinlein A, Hochmuth C, Klawonn A (2019) Monolithic Overlapping Schwarz Domain Decomposition Methods with GDSW Coarse Spaces for Incompressible Fluid Flow Problems. *SIAM J Sci Comput* 41(4):291–316. <https://doi.org/10.1137/18M1184047>
- Heinlein A, Klawonn A, Knepper J, Saßmannshausen L (2025) Monolithic and Block Overlapping Schwarz Preconditioners for the Incompressible Navier–Stokes Equations. arXiv preprint [arXiv:2506.16179](https://arxiv.org/abs/2506.16179)
- Heinlein A, Klawonn A, Lanser M, Weber J (2020) A frugal FETI-DP and BDDC coarse space for heterogeneous problems, 562–591 [https://doi.org/10.1553/etna\\_vol53s562](https://doi.org/10.1553/etna_vol53s562)
- Heinlein A, Klawonn A, Rheinbach O, Röver F (2019) A three-level extension of the GDSW overlapping Schwarz preconditioner in two dimensions. In: *Advanced Finite Element Methods with Applications: Selected Papers from the 30th Chemnitz Finite Element Symposium 2017* 30, pp. 187–204. [https://doi.org/10.1007/978-3-030-14244-5\\_10](https://doi.org/10.1007/978-3-030-14244-5_10). Springer
- Hochmuth C (2020) *Parallel overlapping schwarz preconditioners for incompressible fluid flow and fluid-structure interaction problems*. PhD thesis, Universität zu Köln. <http://kups.ub.uni-koeln.de/id/eprint/11345>
- Heinlein A, Rheinbach O, Röver F (2023) A multilevel extension of the GDSW overlapping Schwarz preconditioner in two dimensions. *Computational Methods in Applied Mathematics* 23(4):953–968. <https://doi.org/10.1515/cmam-2022-0168>
- Hartwig P, Scheunemann L, Schröder J (2023) On the numerical treatment of heat sources in laser beam welding processes. *PAMM* 23(1):202200220. <https://doi.org/10.1002/pamm.202200220>
- Hartwig P, Scheunemann L, Schröder J (2023) A volumetric heat source model for the approximation of the melting pool in laser beam welding. *PAMM* 23(4):202300173. <https://doi.org/10.1002/pamm.202300173>
- Intel Corporation: Intel® oneAPI Math Kernel Library PARDISO. (2024). Accessed: January 17, 2025. <https://www.intel.com/content/www/us/en/docs/onemkl/developer-reference-c/2024-0/onemkl-pardiso-parallel-direct-sparse-solver-iface.html>
- Klawonn A, Köhler S, Lanser M, Rheinbach O (2020) Computational homogenization with million-way parallelism using domain decomposition methods. *Comput Mech* 65(1):1–22. <https://doi.org/10.1007/s00466-019-01749-5>

22. Klawonn A, Lanser M, Rheinbach O (2018) Nonlinear BDDC methods with approximate solvers, 244–273 [https://doi.org/10.1553/etna\\_vol49s244](https://doi.org/10.1553/etna_vol49s244)
23. Klawonn A, Lanser M, Rheinbach O, Uran M (2021) Fully-coupled micro-macro finite element simulations of the Nakajima test using parallel computational homogenization. *Comput Mech* 68(5):1153–1178. <https://doi.org/10.1007/s00466-021-02063-9>
24. Klawonn A, Pavarino LF (1998) Overlapping Schwarz methods for mixed linear elasticity and Stokes problems. *Comput Methods Appl Mech Eng* 165(1–4):233–245. [https://doi.org/10.1016/S0045-7825\(98\)00059-0](https://doi.org/10.1016/S0045-7825(98)00059-0)
25. Klawonn A, Pavarino LF (2000) A comparison of overlapping Schwarz methods and block preconditioners for saddle point problems. *Num Linear Algebra Appl* 7(1):1–25
26. Kiefer B, Prüger S, Rheinbach O, Röver F (2023) Monolithic parallel overlapping Schwarz methods in fully-coupled nonlinear chemo-mechanics problems. *Comput Mech* 71:765–788. <https://doi.org/10.1007/s00466-022-02254-y>
27. Richter F (2010) The physical properties of steels: The 100-steel program. Thermophysics Working Group in GEFTA: Berlin, Germany [https://doi.org/10.1016/S0921-5093\(02\)00526-9](https://doi.org/10.1016/S0921-5093(02)00526-9)
28. Schenk O, Gärtner K, Fichtner W, Stricker A (2001) Pardiso: a high-performance serial and parallel sparse linear solver in semiconductor device simulation. *Futur Gener Comput Syst* 18(1):69–78. [https://doi.org/10.1016/S0167-739X\(00\)00076-5](https://doi.org/10.1016/S0167-739X(00)00076-5)
29. Simo JC, Hughes TJR (1998) *Interdisciplinary applied mathematics*, vol. v. 7. Springer, New York. <https://doi.org/10.1007/b98904>
30. Simo JC, Miehe C (1992) Associative coupled thermoplasticity at finite strains: Formulation, numerical analysis and implementation. *Comput Methods Appl Mech Eng* 98(1):41–104. [https://doi.org/10.1016/0045-7825\(92\)90170-O](https://doi.org/10.1016/0045-7825(92)90170-O). Accessed 2024-12-16
31. Simo JC, Taylor RL, Pister KS (1985) Variational and projection methods for the volume constraint in finite deformation elastoplasticity. *Comput Methods Appl Mech Eng* 51(1–3):177–208. [https://doi.org/10.1016/0045-7825\(85\)90033-7](https://doi.org/10.1016/0045-7825(85)90033-7)
32. Taylor RL (2014) FEAP - Finite Element Analysis Program. University of California, Berkeley. <http://www.ce.berkeley/feap>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.