



INSTITUT DE FRANCE
Académie des sciences

Comptes Rendus

Mécanique

Frédéric Nataf and Pierre-Henri Tournier

A GenEO Domain Decomposition method for Saddle Point problems

Published online: 27 March 2023

<https://doi.org/10.5802/crmeca.175>

Part of Special Issue: The scientific legacy of Roland Glowinski

Guest editors: Gregoire Allaire (CMAP, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, France), Jean-Michel Coron (Laboratoire Jacques-Louis Lions, Sorbonne Université) and Vivette Girault (Laboratoire Jacques-Louis Lions, Sorbonne Université)



This article is licensed under the
CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.
<http://creativecommons.org/licenses/by/4.0/>



*Les Comptes Rendus. Mécanique sont membres du
Centre Mersenne pour l'édition scientifique ouverte*

www.centre-mersenne.org

e-ISSN : 1873-7234



The scientific legacy of Roland Glowinski / *L'héritage scientifique de Roland Glowinski*

A GenEO Domain Decomposition method for Saddle Point problems

Une méthode de décomposition de domaine GenEO pour les problèmes de points selle

Frédéric Nataf^{*, a} and Pierre-Henri Tournier^a

^a Laboratoire J.L. Lions, Sorbonne Université, 4 place Jussieu France

E-mails: frederic.nataf@sorbonne-universite.fr (F. Nataf),

pierre-henri.tournier@sorbonne-universite.fr (P.-H. Tournier)

Abstract. We introduce an adaptive element-based domain decomposition (DD) method for solving saddle point problems defined as a block two by two matrix. The algorithm does not require any knowledge of the constrained space. We assume that all sub matrices are sparse and that the diagonal blocks are spectrally equivalent to a sum of positive semi definite matrices. The latter assumption enables the design of adaptive coarse space for DD methods that extends the GenEO theory (Spillane et al., 2014) to saddle point problems. Numerical results on three dimensional elasticity problems for steel-rubber structures discretized by a finite element with continuous pressure are shown for up to one billion degrees of freedom.

Résumé. Nous présentons une méthode de décomposition de domaine (DD) adaptative basée pour résoudre les problèmes de points selle définis comme une matrice bloc 2x2. L'algorithme ne nécessite aucune connaissance de l'espace contraint. Nous supposons que toutes les sous-matrices sont creuses et que les blocs diagonaux sont spectralement équivalents à une somme de matrices semi-définies positives. Cette dernière hypothèse permet de concevoir un espace grossier adaptatif pour les méthodes DD qui étend la théorie de GenEO (Spillane et al., 2014) aux problèmes de points de selle. Des résultats numériques sur des problèmes d'élasticité tridimensionnels pour des structures acier-caoutchouc discrétisées avec une pression continue sont montrés jusqu'à un milliard de degrés de liberté.

Keywords. domain decomposition method, nearly incompressible elasticity, high performance computing, saddle point problem, coarse space, multiscale finite element, Schur complement.

Mots-clés. Méthode de décomposition de domaine, élasticité quasi incompressible, calcul haute performance, problème de point selle, espace grossier, éléments finis multi échelle, complément de Schur.

Funding. The HPC resources of OCCIGEN@CINES were given under the allocation 2021-067730 granted by GENCI..

Published online: 27 March 2023

* Corresponding author.

1. Introduction

Solving saddle point problems with parallel algorithms is very important for many branches of scientific computing: fluid and solid mechanics especially for incompressible or nearly incompressible materials, computational electromagnetism, multi-physics problems, inverse problems and optimization. They also arise when analyzing realistic problems with the incorporation of a set of linear multipoint constraints (MPC) into the linear solver. Examples of the problems described by the MPCs are the non-penetration condition in the contact problem, connection of different types of finite elements, modelling of rigid bodies, and so on. Using Lagrange multipliers to enforce them, yields linear saddle point systems.

For small enough problems, direct solvers are the method of choice since they are robust and have a predictable efficiency. But for large problems the memory requirements are very large and at some point the solver runs out of memory, see e.g. the tests performed in § 5.5.1. Then iterative solvers are needed. Domain decomposition methods are, alongside multigrid methods, one of the dominant paradigms for defining efficient and robust preconditioners in modern large-scale applications dealing with partial differential equations and algebraic systems of equations.

We are interested in domain decomposition (DD) methods since they are naturally well-fitted to modern parallel architectures.

Recently adaptive spectral coarse spaces have been developed successfully for symmetric positive definite systems, see [1–5] and the book [6] and references therein. In [7, § 6.2.], it is shown that in the case of nearly incompressible elasticity in 2D with a $\mathbb{P}2/\mathbb{P}0$ discretization where the pressure unknown can be condensed and the resulting velocity matrix is sparse and SPD, the adaptive GenEO coarse space fails. Note that for a homogeneous nearly incompressible elasticity problem, the extension of GenEO to FETI [5] works well but not in the highly heterogeneous case considered in this paper. It is thus necessary to keep the saddle point formulation. Note also that the method presented here applies as well to a $\mathbb{P}2/\mathbb{P}1$ discretization with a continuous pressure.

For specific systems of partial differential equations with a saddle point formulation, efficient DD methods have been designed, see e.g., [8–10] and [11] references therein. Also in [12], a GenEO coarse space is introduced for the PL. Lions’ algorithm and its efficiency is mathematically proved for symmetric definite positive problems. In the above article, numerical experiments are conducted on three dimensional elasticity problems for steel-rubber structures discretized by a finite element with continuous pressure. Although the method works well in practice, it lacks theoretical convergence guarantees and also demands the design of specific absorbing conditions as interface conditions.

As for a convergence rate analysis for a discretization with a continuous pressure, the recent article [13] generalizes the theory developed in [14] to the case of nonzero pressure block but under the assumption that the discontinuities are resolved by the subdomains. Compared to the above mentioned works, the method we propose has a provable control on the condition number for zero or non zero pressure block with a continuously discretized pressure also in the case of arbitrary heterogeneities and, in contrast to [12], it bypasses the need for absorbing boundary conditions.

Here as in [15–18], we consider the problem in the form of a two by two block matrix. Let m and n be two integers with $m < n$. Let A $n \times n$ be a SPD matrix, B a sparse $m \times n$ full rank matrix of constraints and C a $m \times m$ non negative matrix (in particular, $C = 0$ is allowed), we consider the following saddle point matrix:

$$\mathcal{A} := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}. \quad (1)$$

When the kernel of matrix B is known, very efficient multigrid methods have been designed in the context of finite element methods, see e.g., [19–24]. Without this knowledge, it is nevertheless

also possible to design efficient geometric multigrid methods as in [25] where the fine mesh is obtained by several uniform mesh refinements. In the case where matrix C is zero, augmented Lagrangian methods [26–28] may be used. Applications to incompressible fluid dynamics and incompressible elasticity are numerous, see e.g. in [24, 29] and references therein.

Here we do not assume any knowledge on the kernel of matrix B nor that C is zero and we work with arbitrary meshes. The following three factor factorization, see e.g., [30]:

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} = \begin{pmatrix} I & 0 \\ BA^{-1} & I \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & -(C + BA^{-1}B^T) \end{pmatrix} \begin{pmatrix} I & A^{-1}B^T \\ 0 & I \end{pmatrix},$$

shows that solving the linear system with \mathcal{A} can be performed by solving sequentially linear systems with A and one with the Schur complement $C + BA^{-1}B^T$. In order to build a scalable method, we assume that all three matrices A , B and C are sparse and that A and C are the sum of positive semi definite matrices. This is easily achieved in finite element or finite volume contexts for partial differential equations. The latter assumption enables the design of adaptive coarse space for DD methods, see [6].

The highlights of the article are:

- new provably scalable DD method that is a blend of two-level Schwarz and Neumann–Neumann domain decomposition methods.
- successfully application to the notorious difficult problem of large scale (1 billion d.o.f’s) nearly incompressible elasticity with heterogeneities not resolved by the domain decomposition and discretized with a continuous pressure.

The paper is organized as follows. In § 2, we recall the two-level additive Schwarz method denoted M_A used to precondition the matrix A (the primal-primal block of the saddle point problem). Then in § 3, we introduce the operator $P_S := C + BM_A^{-1}B^T$ which is spectrally equivalent to the Schur complement S . Its preconditioning is studied in § 3.2. In § 4, we combine these different components to define in a compact way the parallel saddle point preconditioner. In § 5, we present weak and strong scaling experiments on large scale elasticity problems for steel-rubber structures discretized by a finite element with continuous pressure. Comparisons with direct and multigrid solvers are also given. These problems are highly heterogeneous since the Lamé–Poisson coefficients of the rubber are $(E_1, \nu_1) = (1e + 7, 0.4999)$ and those of the steel are $(E_2, \nu_2) = (2e + 9, 0.35)$.

2. Preconditioning of the primal problem

The sparse $n \times n$ SPD matrix A is preconditioned by a two-level Schwarz type DD method:

$$M_A^{-1} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^N R_i^T (R_i A R_i^T)^{-1} R_i, \quad (2)$$

where R_0 is full rank $\dim(V_0) \times n$ with V_0 denoting the space spanned by the columns of R_0^T . The following assumptions are crucial to ensure the final method is scalable:

Assumption 1 (dimension and structure of the coarse space).

- The coarse space dimension, $\dim(V_0)$, is $O(N)$ typically 10-20 times N .
- The coarse space is made of extensions by zero of local vectors.

Using the GenEO method [31], it is possible to specify in advance two constants $0 < \lambda_m < 1 < \lambda_M$ and then build a coarse space V_0 such that M_A^{-1} is spectrally equivalent to A^{-1} :

$$\frac{1}{\lambda_M} M_A^{-1} \leq A^{-1} \leq \frac{1}{\lambda_m} M_A^{-1}. \quad (3)$$

The dimension of the coarse space V_0 is typically proportional to the number of subdomains. This corresponds to Assumption 1. More precisely, for each subdomain $1 \leq i \leq N$, let D_i be a non negative diagonal matrix that defines a discrete partition of unity, i.e.:

$$\sum_{i=1}^N R_i^T D_i R_i = I,$$

and A_i^{Neu} be a symmetric semi-definite positive matrix such that for the maximum multiplicity of the intersection of subdomains denoted k_1 , we have:

$$\sum_{i=1}^N R_i^T A_i^{Neu} R_i \leq k_1 A. \quad (4)$$

Then, the GenEO eigenvalue problem is local to each subdomain and reads:

Find $(\lambda_{ik}, V_{ik}) \in \mathbb{R} \times \mathbb{R}^{\text{rank}(R_i)}$ such that:

$$(D_i R_i A R_i^T D_i) V_{ik} = \lambda_{ik} A_i^{Neu} V_{ik}. \quad (5)$$

Let $\tau > 0$ be a positive threshold, the coarse space is the vector space spanned by the vectors $R_i^T D_i V_{ik}$ for all $\lambda_{ik} > \tau$. Then inequality (3) holds with $\lambda_m := (1 + k_1 \tau)^{-1}$ and $\lambda_M := k_0$ where k_0 is the maximal number of neighbours of a subdomain including itself.

Our aim in the next section is to precondition the Schur complement $-S$ of matrix \mathcal{A} (eq. (1)) where

$$S := C + B A^{-1} B^T. \quad (6)$$

This is achieved by a series of spectrally equivalent matrices or preconditioners, the first one being P_S defined as follows:

$$P_S := C + B M_A^{-1} B^T \quad (7)$$

and the final one being N_S^{-1} introduced in § 3.3, see eq. (24). Finally in § 4.2 we will introduce the preconditioner of the saddle point matrix \mathcal{A} .

3. Schur complement preconditioning

3.1. First spectrally equivalent preconditioner

Note that P_S is by definition a sum of $N + 2$ positive semi definite matrices

$$P_S := B R_0^T (R_0 A R_0^T)^{-1} R_0 B^T + C + \sum_{i=1}^N B R_i^T (R_i A R_i^T)^{-1} R_i B^T. \quad (8)$$

Since B is a sparse matrix, it is interesting to introduce, for all $0 \leq i \leq N$, \tilde{R}_i the restriction operator on the support of $\Im(B R_i^T)$ (the image of $B R_i^T$) so that:

$$\tilde{R}_i^T \tilde{R}_i B R_i^T = B R_i^T. \quad (9)$$

Then, by defining for $0 \leq i \leq N$,

$$\tilde{B}_i := \tilde{R}_i B R_i^T,$$

the operator P_S is rewritten as

$$P_S := \tilde{R}_0^T \tilde{B}_0 (R_0 A R_0^T)^{-1} \tilde{B}_0^T \tilde{R}_0 + C + \sum_{i=1}^N \tilde{R}_i^T \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \tilde{R}_i. \quad (10)$$

We consider a partition of unity on $H := \mathbb{R}^m$ defined with local diagonal matrices $(\tilde{D}_i)_{1 \leq i \leq N} \in \mathbb{R}^{\dim(\Im(B R_i^T)) \times \dim(\Im(B R_i^T))}$.

$$\sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \tilde{R}_i = I_H.$$

We make the following assumption

Assumption 2. *There exist symmetric positive semi-definite matrices $(\tilde{C}_i)_{1 \leq i \leq N}$ such that for some constant \tilde{k}_1*

$$C \leq \sum_{i=1}^N \tilde{R}_i^T \tilde{C}_i \tilde{R}_i \leq \tilde{k}_1 C. \quad (11)$$

This assumption is not so restrictive. Indeed, for a minimization problem with constraints enforced exactly without penalization nor relaxation, we have $C = 0$ and the assumption is automatically satisfied. Moreover, we have:

Lemma 3. *If C is a diagonal matrix, Assumption 2 is satisfied with $\tilde{k}_1 = 1$.*

Proof. If C is a diagonal matrix, it suffices to take

$$\tilde{C}_i := \tilde{R}_i C \tilde{R}_i^T \tilde{D}_i,$$

which is a diagonal non negative matrix. Indeed, we then have:

$$C = \sum_{i=1}^N C \tilde{R}_i^T \tilde{D}_i \tilde{R}_i = \sum_{i=1}^N \tilde{R}_i^T (\tilde{R}_i C \tilde{R}_i^T \tilde{D}_i) \tilde{R}_i. \quad \square$$

Remark 4. Note also that in the finite element case it suffices to restrict the variational form that defines C to the subdomains. In this case \tilde{k}_1 is the multiplicity of the intersections of the subdomains used to define the \tilde{C}_i 's.

Let us define the operator M_S as the sum of a non local but low rank matrix S_0 :

$$S_0 := \tilde{R}_0^T \tilde{B}_0 (R_0 A R_0^T)^{-1} \tilde{B}_0^T \tilde{R}_0,$$

and of S_1 which is a sum of N local positive semi definite matrices:

$$S_1 := \sum_{i=1}^N \tilde{R}_i^T \left(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \right) \tilde{R}_i,$$

that is

$$M_S := S_0 + S_1.$$

By Assumption 2, the operator M_S is spectrally equivalent to P_S which is also spectrally equivalent to S . Note that we may assume that S_1 is invertible whereas it does not make sense for S_0 . Note that if it is not the case, since we build a preconditioner, S_1 can be regularized by a small diagonal term with little effect on the efficiency of the preconditioner.

We consider next the construction of a preconditioner $M_{S_1}^{-1}$ to S_1 leveraging the fact that S_1 is a sum of symmetric semidefinite positive matrices. Let us stress that this property stems from the domain decomposition structure of the preconditioner for matrix A which apart from the coarse level is block diagonal.

3.2. Preconditioning of S_1

It is well known that one-level domain decomposition methods are in most cases not scalable. Nevertheless, the study of a one-level method in § 3.2.1 enables the identification of a suitable coarse space that will be efficiently embedded in a scalable two-level domain decomposition method in § 3.2.2.

Our studies of the spectrum of the DD preconditioners are based on the Fictitious Space lemma which is recalled here, see [32] for the original paper and [33] for a modern presentation.

Lemma 5 (Fictitious Space Lemma, Nepomnyaschikh 1991). *Let H and H_D be two Hilbert spaces, with the scalar products denoted by (\cdot, \cdot) and $(\cdot, \cdot)_D$. Let the symmetric positive bilinear forms $a : H \times H \rightarrow \mathbb{R}$ and $b : H_D \times H_D \rightarrow \mathbb{R}$ be generated by the s.p.d. operators $A : H \rightarrow H$ and $B : H_D \rightarrow H_D$, respectively (i.e. $(Au, v) = a(u, v)$ for all $u, v \in H$ and $(Bu_D, v_D)_D = b(u_D, v_D)$ for all $u_D, v_D \in H_D$). Suppose that there exists a linear operator $\mathcal{R} : H_D \rightarrow H$ that satisfies the following three assumptions:*

- (i) \mathcal{R} is surjective.
- (ii) Continuity of \mathcal{R} : there exists a positive constant c_R such that

$$a(\mathcal{R}u_D, \mathcal{R}u_D) \leq c_R \cdot b(u_D, u_D) \quad \forall u_D \in H_D. \quad (12)$$

- (iii) Stable decomposition: there exists a positive constant c_T such that for all $u \in H$ there exists $u_D \in H_D$ with $\mathcal{R}u_D = u$ and

$$c_T \cdot b(u_D, u_D) \leq a(\mathcal{R}u_D, \mathcal{R}u_D) = a(u, u). \quad (13)$$

We introduce the adjoint operator $\mathcal{R}^* : H \rightarrow H_D$ by $(\mathcal{R}u_D, u) = (u_D, \mathcal{R}^*u)_D$ for all $u_D \in H_D$ and $u \in H$.

Then, we have the following spectral estimate

$$c_T \cdot a(u, u) \leq a(\mathcal{R}B^{-1}\mathcal{R}^*Au, u) \leq c_R \cdot a(u, u), \quad \forall u \in H, \quad (14)$$

which proves that the eigenvalues of operator $\mathcal{R}B^{-1}\mathcal{R}^*A$ are bounded from below by c_T and from above by c_R .

The Fictitious Space Lemma (FSL) can also conveniently be related to the book [11]: the first assumption corresponds to equation (2.3), page 36, where the global Hilbert space is assumed to satisfy a decomposition into subspaces, the second assumption is related to Assumptions 2.3 and 2.4, page 40, and the third assumption corresponds to the Stable decomposition Assumption 2.2 page 40.

3.2.1. One-level DD for S_1

As in [6, Chapter 7], we begin with a one-level Neumann–Neumann [34] type DD method defined in terms of the Fictitious Space Lemma (FSL). This study will be the basis for constructing a scalable two-level preconditioner in § 3.2.2. Recall the formula for the one-level preconditioner $M_{S_1,1}^{-1}$ for S_1 :

$$M_{S_1,1}^{-1} := \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \left(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \right)^\dagger \tilde{D}_i \tilde{R}_i, \quad (15)$$

where the superscript \dagger denotes a pseudo inverse in case the operator in brackets is not invertible. For the sake of simplicity, we assume that they are invertible so that the following framework enables the study of $M_{S_1,1}$ with the fictitious space lemma. Let

$$H := \mathbb{R}^m$$

and let a be the following bilinear form:

$$a : H \times H \rightarrow \mathbb{R} \quad a(\mathbf{P}, \mathbf{Q}) := (S_1 \mathbf{P}, \mathbf{Q}).$$

Let

$$H_D := \prod_{i=1}^N \mathbb{R}^{\text{rank}(\tilde{B}_i)},$$

and b be the following bilinear form:

$$b : H_D \times H_D \rightarrow \mathbb{R} \quad b((\mathbf{P}_i)_{1 \leq i \leq N}, (\mathbf{Q}_i)_{1 \leq i \leq N}) := \sum_{i=1}^N \left(\left(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \right) \mathbf{P}_i, \mathbf{Q}_i \right). \quad (16)$$

We define \mathcal{R} :

$$\begin{aligned} \mathcal{R} : H_D &\rightarrow H \\ (\mathbf{P}_i)_{1 \leq i \leq N} &\mapsto \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i. \end{aligned}$$

We now check the three assumptions of the FSL.

Surjectivity of \mathcal{R} . For any $\mathbf{P} \in H$, we have using the partition of unity that:

$$\mathbf{P} = \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \tilde{R}_i \mathbf{P},$$

so that

$$\mathbf{P} = \mathcal{R}((\tilde{R}_i \mathbf{P})_{1 \leq i \leq N}). \quad (17)$$

Continuity of \mathcal{R} . On the one hand, we have using k_0 the number of neighbours of a subdomain (including itself), $k_0 := \max_{1 \leq i \leq N} \#\mathcal{O}(i)$ where $\mathcal{O}(i) := \{1 \leq j \leq N \mid \tilde{R}_i \tilde{D}_i S_1 \tilde{D}_j \tilde{R}_j^T \neq 0\}$:

$$\begin{aligned} a(\mathcal{R}(\mathcal{P}), \mathcal{R}(\mathcal{P})) &= \left\| \sum_{i=1}^N \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i \right\|_a^2 \leq k_0 \sum_{i=1}^N \|\tilde{R}_i^T \tilde{D}_i \mathbf{P}_i\|_a^2 \\ &\leq k_0 \left(\left(\sum_{j \in \mathcal{O}(i)} \tilde{R}_j^T \left(\tilde{C}_j + \tilde{B}_j (R_j A R_j^T)^{-1} \tilde{B}_j^T \right) \tilde{R}_j \right) \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i, \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i \right). \end{aligned}$$

On the other hand, we have by definition:

$$b(\mathcal{P}, \mathcal{P}) = \sum_{i=1}^N \left(\left(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \right) \mathbf{P}_i, \mathbf{P}_i \right).$$

We can take:

$$c_R := \max_{1 \leq i \leq N} \max_{\mathbf{P}_i \in \mathbb{R}^{\text{rank}(\tilde{B}_i)}} \frac{\left(\sum_{j \in \mathcal{O}(i)} \tilde{R}_i \tilde{R}_j^T \left(\tilde{C}_j + \tilde{B}_j (R_j A R_j^T)^{-1} \tilde{B}_j^T \right) \tilde{R}_j \tilde{R}_i^T \tilde{D}_i \mathbf{P}_i, \tilde{D}_i \mathbf{P}_i \right)}{\left(\left(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \right) \mathbf{P}_i, \mathbf{P}_i \right)}, \quad (18)$$

but we have no control on c_R which may be large. This motivates the introduction of a spectral coarse space in § 3.2.2 with the generalized eigenvalue problem (19).

Stable decomposition. Let $\mathbf{P} \in H$, we start from its decomposition (17) and estimate its b -norm

$$b(\mathcal{P}, \mathcal{P}) = \sum_{i=1}^N \left(\left(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \right) \tilde{R}_i \mathbf{P}, \tilde{R}_i \mathbf{P} \right) = a(\mathbf{P}, \mathbf{P}),$$

so that we can take $c_T = 1$ which is an optimal value.

3.2.2. Two-level DD for S_1

In order to control the value of c_R defined above, we introduce a two-level preconditioner similarly to what is done for Schur complement methods in [6, § 7.8.3, p. 197] or in [5]. The generalized eigenvalue problem in each subdomain $1 \leq i \leq N$ to be solved in order to build the coarse space is inferred from the definition of the constant c_R in eq. (18):

$$\begin{aligned} \tilde{D}_i \tilde{R}_i \left(\sum_{j \in \mathcal{O}(i)} \tilde{R}_j^T \left(\tilde{C}_j + \tilde{B}_j (R_j A R_j^T)^{-1} \tilde{B}_j^T \right) \tilde{R}_j \right) \tilde{R}_i^T \tilde{D}_i \mathbf{P}_{ik} \\ = \lambda_{ik} \left(\tilde{C}_i + \tilde{B}_i (R_i A R_i^T)^{-1} \tilde{B}_i^T \right) \mathbf{P}_{ik}. \end{aligned} \quad (19)$$

This generalized eigenvalue problem contains inverses of some local matrices on both sides and in order to solve it via e.g. Arpack, we have to factorize one of them. This difficulty can be

overcome since the right matrix is the inverse of the Schur complement of the extended sparse matrix (23). It is thus amenable to a factorization using only sparse matrix factorizations. Note that matrix-vector products on the left hand side involve neighbor to neighbor communications and thus synchronization points. Consequently, solving these generalized eigenvalue problem is a little more involved to code than in the standard GenEO.

The coarse space is defined as follows. Let τ_{S_1} be a user-defined threshold; for each subdomain $1 \leq i \leq N$, we introduce a subspace $\widetilde{W}_i \subset \mathbb{R}^{\text{rank}(\widetilde{B}_i)}$:

$$\widetilde{W}_i := \text{Span} \{ \mathbf{P}_{ik} \mid \lambda_{ik} > \tau_{S_1} \}. \quad (20)$$

Then the coarse space \widetilde{W}_0 is defined (with some abuse of notation) as

$$\widetilde{W}_0 := \bigoplus_{1 \leq i \leq N} \widetilde{R}_i^T \widetilde{D}_i \widetilde{W}_i.$$

Let Z_{S_1} be a rectangular matrix whose columns span the coarse space \widetilde{W}_0 . Let \widetilde{P}_0 be the S_1 orthogonal projection from \mathbb{R}^m on \widetilde{W}_0 whose formula is

$$\widetilde{P}_0 = Z_{S_1} \left(Z_{S_1}^T S_1 Z_{S_1} \right)^{-1} Z_{S_1}^T S_1. \quad (21)$$

In order to avoid a too cumbersome analysis, we make the following assumption:

Assumption 6. *We assume that for all subdomains $1 \leq i \leq N$, $\widetilde{C}_i + \widetilde{B}_i (R_i A R_i^T)^{-1} \widetilde{B}_i^T$ is invertible.*

Finally, the preconditioner for S_1 reads

$$M_{S_1}^{-1} := Z_{S_1} \left(Z_{S_1}^T S_1 Z_{S_1} \right)^{-1} Z_{S_1}^T + (I - \widetilde{P}_0) \times \left(\sum_{i=1}^N \widetilde{R}_i^T \widetilde{D}_i \left(\widetilde{C}_i + \widetilde{B}_i (R_i A R_i^T)^{-1} \widetilde{B}_i^T \right)^\dagger \widetilde{D}_i \widetilde{R}_i \right) (I - \widetilde{P}_0^T). \quad (22)$$

If Assumption 6 is not satisfied for some subdomain i , we should incorporate the kernel of $\widetilde{C}_i + \widetilde{B}_i (R_i A R_i^T)^{-1} \widetilde{B}_i^T$ into the coarse space and make use of a pseudo inverse in the definition of the preconditioner as it is done for the FETI method, see [35] or [6, §7.8.2] and references therein. Recall that from [6, Chapter 7], we have for $\alpha := \max(1, \frac{k_0}{\tau_{S_1}})$:

$$\frac{1}{\alpha} M_{S_1}^{-1} \leq S^{-1} \leq M_{S_1}^{-1}.$$

A careful implementation of (22) requires two coarse solves. Note that the application of $M_{S_1}^{-1}$ can be done using only sparse solvers since solving a linear system with a local Schur complement

$$\left(\widetilde{C}_i + \widetilde{B}_i (R_i A R_i^T)^{-1} \widetilde{B}_i^T \right) \mathbf{P}_i = \mathbf{G}_i,$$

amounts to solving an augmented sparse system which has the form of a local saddle point system:

$$- \begin{pmatrix} R_i A R_i^T & \widetilde{B}_i^T \\ \widetilde{B}_i & -\widetilde{C}_i \end{pmatrix} \begin{pmatrix} \mathbf{U}_i \\ \mathbf{P}_i \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{G}_i \end{pmatrix}. \quad (23)$$

3.3. Final Preconditioner for the Schur complement

From the spectrally equivalent preconditioner M_{S_1} to S_1 , we define N_S a spectrally equivalent preconditioner to M_S and thus to S as well:

$$N_S := S_0 + M_{S_1}. \quad (24)$$

We now consider the application of the preconditioner N_S , that is for some right-hand side $\mathbf{G} \in \mathbb{R}^m$, solving the following system for \mathbf{P} :

$$N_S \mathbf{P} = \mathbf{G}, \quad (25)$$

by a Krylov solver with $M_{S_1}^{-1}$ as a preconditioner.

4. Recap

4.1. Setup for the Schur complement preconditioner

We have a setup phase which is composed of:

- (1) Build the two-level preconditioner M_A^{-1} for A , see eq. (2),
- (2) Build the two-level preconditioner $M_{S_1}^{-1}$ for S_1 , see eq. (22).

Once the setup is complete, applying preconditioner N_S^{-1} can be performed following Algorithm 1

Algorithm 1 N_S^{-1} matvec product

INPUT: $\mathbf{G} \in \mathbb{R}^m$ OUTPUT: $\mathbf{P} = N_S^{-1} \mathbf{G}$

1. Solve eq. (25) in \mathbf{P} by a Krylov method with M_S^{-1} as preconditioner.
-

4.2. DD solver for the saddle point system

Algorithm 2 DD saddle point solver

INPUT: $\begin{pmatrix} \mathbf{F}_U \\ \mathbf{F}_P \end{pmatrix} \in \mathbb{R}^{n+m}$ OUTPUT: $\begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix}$ the solution to (26).

1. Solve $A\mathbf{G}_U = \mathbf{F}_U$ by a PCG with M_A^{-1} as a preconditioner
 2. Compute $\mathbf{G}_P := \mathbf{F}_P - B\mathbf{G}_U$
 3. Solve $(C + BA^{-1}B^T)\mathbf{P} = -\mathbf{G}_P$ by a PCG with N_S^{-1} as a preconditioner, see Algorithm 1
 4. Compute $\mathbf{G}_U := \mathbf{F}_U - B^T\mathbf{P}$
 5. Solve $A\mathbf{U} = \mathbf{G}_U$ by a PCG with M_A^{-1} as a preconditioner
-

The solution to the saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_U \\ \mathbf{F}_P \end{pmatrix} \quad (26)$$

is obtained by Algorithm 2. Note that Step 3 requires the matrix-vector product with matrix $C + BA^{-1}B^T$ which is done by an iterative solve for matrix A^{-1} .

5. Numerical experiments

In this section, we perform 3D experiments to illustrate the theory and the performance of the method. We are interested in a heterogeneous elasticity problem with nearly incompressible material, typically rubber-steel structures. First, we recall the definition of the coefficients and the corresponding variational formulation.

The mechanical properties of a solid are characterized by its Young modulus E and Poisson ratio ν , or alternatively by its Lamé coefficients λ and μ . They verify the following relations:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1+\nu)}. \quad (27)$$

For the discretization, we choose a continuous pressure space and take the lowest order Taylor–Hood finite element $C0P2 - C0P1$ whose stability is proved, see e.g., [36]. The domain Ω is a beam and the variational problem consists in finding $(\mathbf{u}_h, p_h) \in \mathcal{V}_h := \mathbb{P}_2^3 \cap (H_0^1(\Omega))^3 \times \mathbb{P}_1$ with Dirichlet boundary conditions on the four lateral faces and Neumann boundary conditions on the other two faces such that for all $(\mathbf{v}_h, q_h) \in \mathcal{V}_h$:

$$\begin{cases} \int_{\Omega} 2\mu \underline{\underline{\varepsilon}}(\mathbf{u}_h) : \underline{\underline{\varepsilon}}(\mathbf{v}_h) dx - \int_{\Omega} p_h \operatorname{div}(\mathbf{v}_h) dx = \int_{\Omega} \mathbf{f} \mathbf{v}_h dx \\ - \int_{\Omega} \operatorname{div}(\mathbf{u}_h) q_h dx - \int_{\Omega} \frac{1}{\lambda} p_h q_h = 0. \end{cases} \quad (28)$$

Letting \mathbf{u} denote the degrees of freedom of \mathbf{u}_h and p that of p_h , the problem can be written in matrix form as:

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}. \quad (29)$$

The matrix C is a mass matrix arising from the discretization of a variational form. This enables us to satisfy Assumption 2 with \tilde{C}_i the corresponding mass matrix but only defined on subdomain $\tilde{\Omega}_i$ which is the extension by a layer of direct neighbors of subdomain Ω_i .

In the following numerical experiments, we consider a heterogeneous beam composed of 10 alternating layers of rubber material $(E_1, \nu_1) = (1e+7, 0.4999)$ and steel material $(E_2, \nu_2) = (2e+9, 0.35)$. Note that the decomposition into subdomains is generated automatically, independently of the distribution of the coefficients, see Figure 1. As a result, the discontinuities are not resolved by the domain decomposition.

5.1. Software, hardware, implementation details

In the following numerical experiments, the iteration counts assess the control of the condition number via the adaptive coarse spaces. We also report timings along with iteration counts. We also showcase the fact that the size of the GenEO coarse space adapts automatically to the difficulty of the problem at hand, for example when going from a homogeneous to a heterogeneous problem. We illustrate the efficiency of the method by performing weak and strong scalability tests, using the automatic graph partitioner *Metis* [37] for the subdomain partitioning.

The problem is discretized and solved with the open-source parallel finite element software *FreeFEM* [38]. *FreeFEM* is a domain specific language (DSL) where the problem to be solved is defined in terms of its variational formulation. Then the local matrices $(A_i^{Neu})_{1 \leq i \leq N}$ (see eq. (4)) and $(\tilde{C}_i)_{1 \leq i \leq N}$ (see eq. (11)) are easily obtained by restricting the corresponding variational formulations to adequate local subdomains. Note that these matrices are different from the restriction of the global matrices A and C to the local degrees of freedom. The domain decomposition algorithm presented in this paper is implemented on top of the *ffddm* framework, a set of parallel *FreeFEM* scripts implementing Schwarz domain decomposition methods. *ffddm* already implements the GenEO method [31] for SPD problems, and its building blocks are designed to simplify the implementation and prototyping of new domain decomposition methods such as the saddle point solver presented in this paper. The *ffddm* documentation is available on the *FreeFem.org* web page, see [39].

Numerical results are obtained on the french GENCI supercomputer *Occigen*, on the Haswell partition composed of 50544 cores of Intel Xeon E5-2690V3 processors clocked at 2.6GHz. The interconnect is an InfiniBand FDR 14 pruned fat tree. We use Intel compilers, the Intel Math Kernel Library and Intel MPI.

As is usually done in domain decomposition methods, we assign one subdomain per MPI process. Our implementation is pure MPI and no multithreading is done ; we assign one MPI process per computing core. The mesh of the computational domain is partitioned using the automatic graph partitioner *Metis* [37] (see Figure 1). Local subdomain matrices are factorized by

the sparse direct solver *MUMPS* [40]. Local eigenvalue problems are solved with *Arpack* [41]; both libraries are interfaced with *FreeFEM*. GenEO coarse space matrices $R_0 A R_0^T$ in (2) and $Z_{S_1}^T S_1 Z_{S_1}$ in (21) are assembled and factorized in a distributed manner on a few cores (24 in most of the experiments) using the parallel solver *MUMPS*.

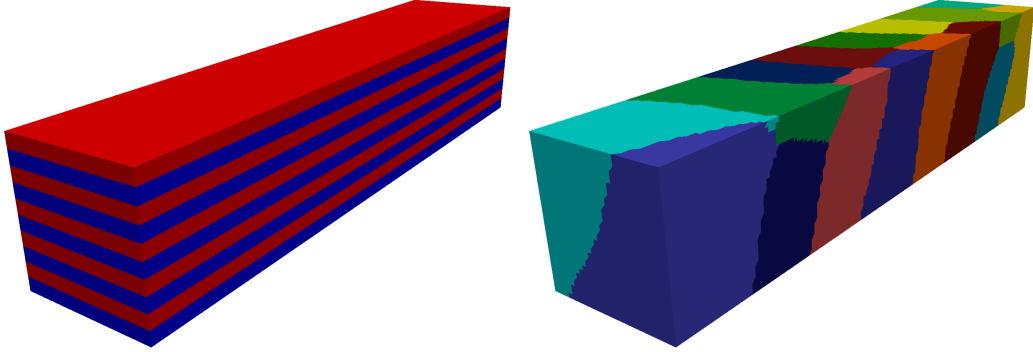


Figure 1. Heterogeneous beam composed of 10 alternating layers of rubber and steel. Coefficient distribution (left) and mesh partitioning into 16 subdomains by the automatic graph partitioner *Metis* (right).

For illustration purposes, we represent in Figure 2 (top) the eigenvalues of the local GenEO eigenvalue problems for both coarse spaces, V_0 for A and \tilde{W}_0 for S_1 (the former corresponding to eq. (5) and the latter to eq. (19)), for the heterogeneous beam problem with 16 million degrees of freedom, corresponding to the first row of Table 1. The figures show the inverse of the first 40 largest eigenvalues for 10 of the subdomains for the experiment corresponding to the first row of Table 1, so that the eigenvectors corresponding to the smallest values on the graphs (below the dashed line) will be selected to enter the coarse space. For comparison, we also solve the constant coefficient problem corresponding to an homogeneous steel (and compressible) beam and show the eigenvalues in Figure 2 (bottom).

We can see the effect of heterogeneities on the spread of eigenvalues for different subdomains compared to the homogeneous case. In addition, we retrieve the 6 eigenvalues corresponding to the rigid body modes for A in the homogeneous case, and we see that we need a larger set of eigenvectors in order to build a robust coarse space in the heterogeneous case. Figure 2 also shows that there is no need for a coarse space for S_1 in the compressible homogeneous case. A strong feature of the GenEO method is that relevant eigenvectors to enter the coarse space are selected automatically, adapting to the problem at hand and its spatial heterogeneity. Moreover, the robustness of the preconditioner does not rely on a specific partitioning, which allows the use of automatic graph partitioners such as *Metis* [37] or *Scotch* [42].

5.2. Parameters of the method

The method has a few parameters in play:

- The number of layers of mesh elements in the overlap region between subdomains is 2 for the velocity blocks (corresponding to R_i in (2)) and 4 for the pressure blocks (corresponding to \tilde{R}_i in (10)). This corresponds to the minimum overlap that satisfies relation (9) for a symmetric construction of the overlap between subdomains.

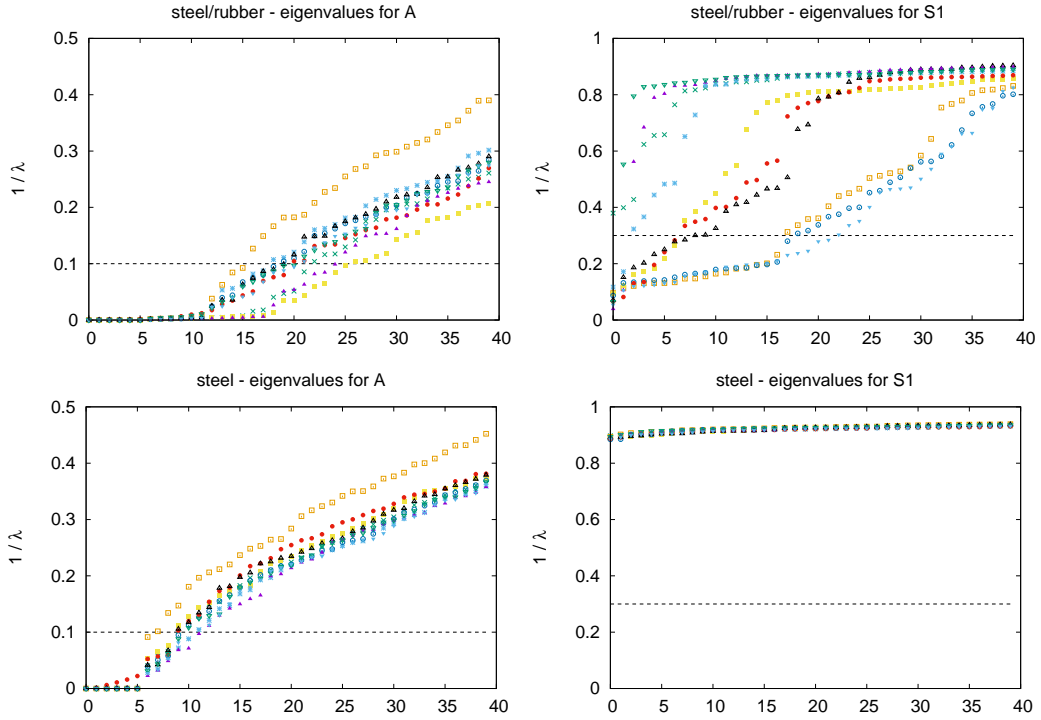


Figure 2. Top: Heterogeneous steel and rubber beam. Bottom: Homogeneous steel only beam. Inverse of the eigenvalues of the local GenEO eigenvalue problems for both coarse spaces, V_0 for A (left) and \widetilde{W}_0 for S_1 (right), for 10 of the subdomains.

- For the heterogeneous beam problem, we set the threshold τ_A for selecting the local eigenvectors entering the coarse space V_0 to $\tau_A = 10$ (corresponding to $1/\lambda = 0.1$ on Figure 2, left). The threshold τ_{S_1} for selecting the local eigenvectors entering the coarse space \widetilde{W}_0 is set to $\tau_{S_1} = 3.33$ (corresponding to $1/\lambda = 0.3$ on Figure 2, right).

The selection of these thresholds is based on the fact that the iteration counts are very stable in both weak and strong scaling experiments. It means that for a given physics the parameters τ_A and τ_{S_1} can be tuned for a small test case and then used in large scale experiments.

5.3. Weak scalability test for heterogeneous steel and rubber beam

Here we present weak scalability results for the heterogeneous beam composed of 10 alternating layers of rubber $(E_1, \nu_1) = (1e + 7, 0.4999)$ and steel $(E_2, \nu_2) = (2e + 9, 0.35)$. The local problem size is kept roughly constant as N grows, and the total number of dofs n goes from 16 million on 262 cores to 1 billion on 16800 cores.

We report in Table 1 the iteration counts and computing times for the DD saddle point solver Algorithm 2. Note that in Algorithm 2 we replace PCG by right-preconditioned GMRES for step 3. The stopping criterion is a tolerance smaller than 10^{-5} . Moreover, we use flexible GMRES, as we solve (25) inexactly using GMRES with a tolerance of 10^{-2} in order to apply N_S^{-1} . In order, columns correspond to: number of cores, number of dofs n , size of the coarse space for A $\dim(V_0)$, size of the coarse space for S_1 $\dim(\widetilde{W}_0)$, setup time corresponding to the assembly and factorization of

the various local and coarse operators, number of outer GMRES iterations, GMRES computing time, total computing time (setup + GMRES) and average number of inner GMRES iterations for each solution of (25). All timings are reported in seconds.

Table 1. Weak scaling experiment for 3D heterogeneous elasticity: beam with 10 alternating layers of steel and rubber. Reported iteration counts and timings for DD saddle point algorithm 2.

#cores	n	$\dim(V_0)$	$\dim(\bar{W}_0)$	setup(s)	#It	gmres(s)	total(s)	#It N_S^{-1}
262	15987380	5383	3319	710.7	24	631.6	1342.3	11
525	27545495	9959	2669	526.6	21	519.5	1046.1	12
1050	64982431	17837	4587	675.2	22	665.9	1341.1	11
2100	126569042	32361	7995	689.2	25	733.8	1423.0	10
4200	218337384	59704	13912	593.0	27	705.4	1298.4	10
8400	515921881	141421	25949	735.8	32	1152.5	1888.3	10
16800	1006250208	260348	41341	819.2	29	1717.9	2537.1	12

Iteration counts

We first discuss iteration counts. We see that the outer iteration count remains stable, between 21 and 32. The inner iteration count is also stable and remains around 11. We also observed (figures are not reported here) that the inner GMRES tolerance of 10^{-2} does not affect the outer iteration count compared to an accurate solution with a stricter tolerance of 10^{-5} , and allows a significant reduction in inner iteration count. For example, 11 iterations on average instead of 28 on 1050 cores for the same outer iteration count of 22, leading to a decrease from 1178.2 to 665.9 seconds in GMRES timing.

Timings

In terms of setup timings, the computing time remains relatively stable, with roughly 15% increase for a factor of 64 in problem size. Around 60% of the setup time is spent in the solution of the eigenvalue problems (19) for S_1 .

The solution time stays relatively stable up to 4200 cores, where it starts to degrade. This can be related to the increased cost of the coarse space solves with matrices $R_0 A R_0^T$ and $Z_S^T S_1 Z_S$ as their size increases: total time spent in coarse space solves is 14.7, 62.1 and 679.3 seconds on 262, 4200 and 16800 cores respectively. A possible improvement would be to use a multi-level method to solve the coarse problems iteratively.

5.4. Strong scalability test for heterogeneous steel and rubber beam

Strong scalability results for the same test case as above consisting of an heterogeneous beam composed of 10 alternating layers of rubber and steel are presented in Table 2. The problem size is 27.5 million and the strong scaling test ranges from 525 to 4200 cores. Iteration counts and computing times for the DD saddle point solver Algorithm 2 are reported.

Iteration counts

Outer iteration count remains stable, with a slight increase from 21 to 23. Inner iteration counts remains also stable, even slightly decreasing from 12 to 9.

Table 2. Strong scaling experiment for 3D heterogeneous elasticity: beam with 10 alternating layers of steel and rubber. Reported iteration counts and timings for DD saddle point Algorithm 2.

#cores	n	$\dim(V_0)$	$\dim(\tilde{W}_0)$	setup(s)	#It	gmres(s)	total(s)	#It N_S^{-1}
525	27545495	9959	2669	526.6	21	519.5	1046.1	12
1050	27545495	15078	4082	265.7	21	224.7	490.4	11
2100	27545495	23172	6453	168.8	23	131.1	299.9	10
4200	27545495	37768	11152	103.8	23	91.3	195.1	9

Timings

We see that the setup timing decreases accordingly as the subdomains shrink in size, from 526.6 seconds on 525 cores to 103.8 seconds on 4200 cores; the speedup efficiency with respect to 525 cores ranges from 99% on 1050 cores to 63% on 4200 cores.

We see a similar trend for the solution time, ranging from 519.5 seconds on 525 cores to 91.3 seconds on 4200 cores ; the speedup efficiency with respect to 525 cores ranges from 116% on 1050 cores to 71% on 4200 cores. This decrease in efficiency can be explained by the increased relative cost of coarse space solves as subdomains get smaller: from 3% on 525 cores to 30% on 4200 cores. The added overlap also plays a greater role in the loss of efficiency as subdomains get smaller.

5.5. Comparison with other algorithms

5.5.1. Comparison with a direct solver

Table 3. Comparison with the parallel sparse direct solver *MUMPS* for 3D heterogeneous elasticity: beam with 10 alternating layers of steel and rubber. Reported timings for four discretization levels while also varying the number of cores (OOM means the computation ran out of available memory).

n	#cores	MUMPS			DD saddle point solver			
		setup(s)	solve(s)	total(s)	setup(s)	#It	gmres(s)	total(s)
139809	16	7.1	0.1	7.2	27.1	18	19.7	46.8
1058312	32	85.7	0.8	86.5	166.2	20	137.2	303.4
1058312	65	71.0	0.6	71.6	91.0	21	77.1	168.1
1058312	131	63.2	0.5	63.7	59.7	24	49.7	109.4
3505582	55	477.8	3.7	481.5	404.1	24	430.1	834.2
3505582	110	392.3	2.3	394.6	242.5	23	212.8	455.3
3505582	221	387.0	2.1	389.1	134.8	23	109.4	244.2
3505582	442	453.9	2.2	456.1	88.2	24	68.6	156.8
8235197	262	OOM	/	/	278.5	25	264.3	542.8
8235197	525	1622.1	6.1	1628.2	172.1	24	136.0	308.1
8235197	1050	1994.3	7.4	2001.7	136.5	25	99.7	236.2

In Table 3, we compare the performance of the solver to the parallel sparse direct solver *MUMPS* for the heterogeneous steel and rubber beam test case with four discretization levels, while also varying the number of cores. As we can see, *MUMPS* is comparatively more efficient for smaller problems, with for example a total time of 86.5 seconds compared to 303.4 seconds for

our saddle point solver for 1 million unknowns on 32 cores. However, as expected, we see a large increase in memory and computational cost as the size of the system gets larger: for 8.2 million unknowns, *MUMPS* runs out of memory on 262 cores and solves the problem in 1628.2 seconds on 525 cores, compared to 308.1 seconds for the DD solver. Moreover, we can see from Table 3 that the DD saddle point solver offers much better strong scalability.

5.5.2. Comparison with other Schwarz methods

In Figure 3, we plot the convergence history of GMRES for both compressible homogeneous and heterogeneous steel-rubber cases with 27.5 million unknowns on 525 cores (left), with a comparison to the standard one-level Additive Schwarz Method (ASM) from PETSc [43] on the global problem (right), illustrating the difficulty of the test case at hand. Our saddle point solver needs 15 and 21 iterations to converge for the homogeneous and heterogeneous cases respectively, compared to 856 and 2880 for the one-level method, with significant undesired plateaux.

Note also that for the same physical test case, iteration counts are better than in [12] but timings are not as good. As mentioned in the introduction, our method has the advantage of having a provable convergence estimate and of not depending on the design of specific absorption conditions for the elasticity system.

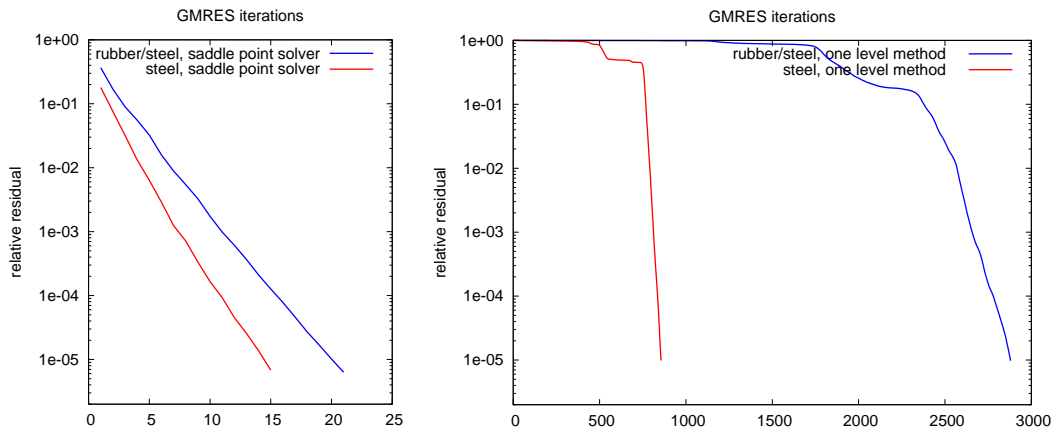


Figure 3. GMRES convergence history of the saddle point solver (left) compared to the one-level Additive Schwarz Method (right) for the homogeneous steel beam and heterogeneous rubber/steel beam problems discretized with 27.5 million unknowns (corresponding to the first row of Table 2), on 525 cores.

5.5.3. Comparison with algebraic multigrid method

We also performed comparisons with the Geometric Algebraic Multigrid (GAMG) preconditioner from PETSc. We were not able to find a suitable tuning of parameters for GAMG for the saddle point formulation. However, we performed comparisons between GAMG and standard GenEO for the velocity formulation on the homogeneous beam, varying the Poisson ratio ν from 0.48 to 0.499. The near kernel of the problem which corresponds to rigid body modes is provided to the multigrid method. The GenEO threshold τ is set to 3.33, and we select at most 80 eigenvectors in each subdomain. Even though GAMG is faster for $\nu \leq 0.49$, we can see that GenEO is more robust as ν increases. In particular, GAMG fails to converge in 2000 iterations for $\nu \geq 0.495$.

Table 4. GAMG versus standard GenEO for the velocity formulation on the homogeneous beam discretized with 7.9 million unknowns, using 131 and 525 cores. Reported iteration counts and timings for different values of the Poisson ratio ν ranging from 0.48 to 0.499.

131 cores		GAMG		GenEO			
ν	#It	total(s)	$dim(V_0)$	setup(s)	#It	gmres(s)	total(s)
0.48	60	67.1	10480	200.7	24	11.2	212.0
0.485	109	89.0	10480	199.5	27	12.7	212.2
0.49	210	137.0	10480	202.0	32	15.0	217.0
0.495	>2000	/	10480	199.9	43	20.2	220.1
0.499	>2000	/	10480	199.2	99	48.6	247.7

525 cores		GAMG		GenEO			
ν	#It	total(s)	$dim(V_0)$	setup(s)	#It	gmres(s)	total(s)
0.48	56	25.5	41766	60.4	18	5.0	65.4
0.485	60	26.1	41984	60.9	20	5.3	66.2
0.49	116	33.3	42000	60.4	23	5.9	66.3
0.495	>2000	/	42000	60.4	32	7.6	68.1
0.499	>2000	/	42000	60.6	95	20.3	81.0

6. Conclusion and outlook

Under the assumption that the diagonal block matrices of a saddle point problem are spectrally equivalent to a sum of positive semi definite matrices, we have introduced an adaptive domain decomposition (DD) method. The key ingredients of the algorithm are a two level Schwarz method for the primal unknowns block and a two-level Neumann–Neuman for the Schur complement problem on the dual unknowns.

Two coarse spaces are built by solving generalized eigenvalue problems, one for the primal unknowns and the second one for the dual unknowns. The robustness of the method was assessed on a notoriously difficult three dimensional elasticity problem for a steel-rubber structure discretized with continuous pressure.

Several issues deserve further investigations. First a multilevel method with more than two levels would enable even larger and possibly faster simulations. Also, the tests were performed with *FreeFEM* scripts using the standalone *ffddm* [39] framework written in the *FreeFEM* language. The integration of the method in the C++/MPI library *hpddm* [44], which already implements the GenEO methods [12, 31], could lead to faster codes and a more general diffusion of the saddle point preconditioner since *hpddm* is callable [45] from the PETSc library [46]. In a different setting, the design of adaptive coarse space is strongly connected to multiscale finite element (MFE) methods (see e.g., [47–49] and references therein) and this work could be used in designing MFE methods for saddle point problems.

Code reproducibility

Our numerical results can be reproduced running the script https://github.com/FreeFem/FreeFem-sources/blob/master/examples/ffddm/elasticity_saddlepoint.edp available in the *FreeFem* distribution starting from version 4.10.

Conflicts of interest

The authors have no conflict of interest to declare.

Acknowledgment

This work was granted access to the HPC resources of OCCIGEN@CINES under the allocation 2021-067730 granted by GENCI.

References

- [1] J. Galvis, Y. Efendiev, “Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces”, *Multiscale Model. Simul.* **8** (2010), no. 5, p. 1621-1644.
- [2] F. Nataf, H. Xiang, V. Dolean, N. Spillane, “A coarse space construction based on local Dirichlet to Neumann maps”, *SIAM J. Sci. Comput.* **33** (2011), no. 4, p. 1623-1642.
- [3] Y. Efendiev, J. Galvis, R. Lazarov, J. Willems, “Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms”, *ESAIM, Math. Model. Numer. Anal.* **46** (2012), no. 5, p. 1175-1199.
- [4] P. Jolivet, V. Dolean, F. Hecht, F. Nataf, C. Prud’homme, N. Spillane, “High performance domain decomposition methods on massively parallel architectures with freefem++”, *J. Numer. Math.* **20** (2012), no. 3-4, p. 287-302.
- [5] N. Spillane, D. J. Rixen, “Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms”, *Int. J. Numer. Methods Eng.* (2013), no. 11, p. 953-990.
- [6] V. Dolean, P. Jolivet, F. Nataf, *An Introduction to Domain Decomposition Methods: algorithms, theory and parallel implementation*, Other Titles in Applied Mathematics, vol. 144, Society for Industrial and Applied Mathematics, 2015.
- [7] N. Spillane, “Robust domain decomposition methods for symmetric positive definite problems”, PhD Thesis, Université Pierre et Marie Curie - Paris VI, Paris, France, 2014.
- [8] J. E. Pasciak, J. Zhao, “Overlapping Schwarz methods in $H(\text{curl})$ on polyhedral domains”, *J. Numer. Math.* **10** (2002), no. 3, p. 221-234.
- [9] A. Klawonn, “An optimal preconditioner for a class of saddle point problems with a penalty term”, *SIAM J. Sci. Comput.* **19** (1998), no. 2, p. 540-552.
- [10] L. F. Pavarino, O. B. Widlund, “Balancing Neumann-Neumann methods for incompressible Stokes equations”, *Commun. Pure Appl. Math.* **55** (2002), no. 3, p. 302-335.
- [11] A. Toselli, O. B. Widlund, *Domain Decomposition Methods - Algorithms and Theory*, Springer Series in Computational Mathematics, vol. 34, Springer, 2005.
- [12] R. Haferssas, P. Jolivet, F. Nataf, “An Additive Schwarz Method Type Theory for Lions’s Algorithm and a Symmetrized Optimized Restricted Additive Schwarz Method”, *SIAM J. Sci. Comput.* **39** (2017), no. 4, p. A1345-A1365.
- [13] O. B. Widlund, S. Zampini, S. Scacchi, L. F. Pavarino, “Block FETI-DP/BDDC preconditioners for mixed isogeometric discretizations of three-dimensional almost incompressible elasticity”, *Math. Comput.* **90** (2021), no. 330, p. 1773-1797.
- [14] X. Tu, J. Li, “A FETI-DP type domain decomposition algorithm for three-dimensional incompressible Stokes equations”, *SIAM J. Numer. Anal.* **53** (2015), no. 2, p. 720-742.
- [15] M. F. Murphy, G. H. Golub, A. J. Wathen, “A note on preconditioning for indefinite linear systems”, *SIAM J. Sci. Comput.* **21** (2000), no. 6, p. 1969-1972.
- [16] M. Benzi, G. H. Golub, J. Liesen, “Numerical solution of saddle point problems”, *Acta Numer.* **14** (2005), p. 1-137.
- [17] E. de Sturler, J. Liesen, “Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. I. Theory”, *SIAM J. Sci. Comput.* **26** (2005), no. 5, p. 1598-1619.
- [18] T. Rees, M. Wathen, “An element-based preconditioner for mixed finite element problems”, *SIAM J. Sci. Comput.* **43** (2020), no. 5, p. S884-S907.
- [19] J. Cahouet, J.-P. Chabard, “Some fast 3d finite element solvers for the generalized stokes problem”, *Int. J. Numer. Methods Fluids* **8** (1988), no. 8, p. 869-895.
- [20] R. Hiptmair, “Multigrid Method for $H(\text{div})$ in Three Dimensions”, *Electron. Trans. Numer. Anal.* **6** (1997), p. 133-152.
- [21] R. Hiptmair, “Multigrid method for Maxwell’s equations”, *SIAM J. Numer. Anal.* **36** (1998), no. 1, p. 204-225.
- [22] D. N. Arnold, R. S. Falk, R. Winther, “Multigrid in $H(\text{div})$ and $H(\text{curl})$ ”, *Numer. Math.* **85** (2000), no. 2, p. 197-217.
- [23] S. Reitzinger, J. Schöberl, “An algebraic multigrid method for finite element discretizations with edge elements”, *Numer. Linear Algebra Appl.* **9** (2002), no. 3, p. 223-238.
- [24] P. E. Farrell, L. Mitchell, F. Wechsung, “An Augmented Lagrangian Preconditioner for the 3D Stationary Incompressible Navier–Stokes Equations at High Reynolds Number”, *SIAM J. Sci. Comput.* **41** (2019), no. 5, p. A3073-A3096.
- [25] D. Drzisga, L. John, U. Rüde, B. Wohlmuth, W. Zulehner, “On the analysis of block smoothers for saddle point problems”, *SIAM J. Matrix Anal. Appl.* **39** (2018), no. 2, p. 932-960.
- [26] R. Glowinski, P. Le Tallec, “Numerical Solution of Problems in Incompressible Finite Elasticity by Augmented Lagrangian Methods. I. Two-dimensional and Axisymmetric Problems”, *SIAM J. Appl. Math.* **42** (1982), p. 400-429.
- [27] R. Glowinski, P. Le Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, SIAM Studies in Applied Mathematics, vol. 9, Society for Industrial and Applied Mathematics, 1989.

- [28] M. Fortin, R. Glowinski, *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*, Studies in Mathematics and its Applications, vol. 15, Elsevier, 2000.
- [29] R. Glowinski, P. Le Tallec, "Numerical Solution of Problems in Incompressible Finite Elasticity by Augmented Lagrangian Methods II. Three-Dimensional Problems", *SIAM J. Appl. Math.* **44** (1984), p. 710-733.
- [30] M. Benzi, A. J. Wathen, "Some preconditioning techniques for saddle point problems", in *Model order reduction: theory, research aspects and applications*, Mathematics in Industry, vol. 13, Springer, 2008, p. 195-211.
- [31] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, R. Scheichl, "Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps", *Numer. Math.* **126** (2014), no. 4, p. 741-770.
- [32] S. V. Nepomnyaschikh, "Mesh theorems of traces, normalizations of function traces and their inversions", *Sov. J. Numer. Anal. Math. Model.* **6** (1991), p. 223-242.
- [33] M. Griebel, P. Oswald, "On the abstract theory of additive and multiplicative Schwarz algorithms", *Numer. Math.* **70** (1995), no. 2, p. 163-180.
- [34] J. F. Bourgat, R. Glowinski, P. Le Tallec, M. Vidrascu, "Variational formulation and algorithm for trace operator in domain decomposition calculations", in *Domain Decomposition Methods* (T. Chan, R. Glowinski, J. Périaux, O. Widlund, eds.), Society for Industrial and Applied Mathematics, 1989, p. 3-16.
- [35] C. Farhat, F.-X. Roux, "A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm", *Int. J. Numer. Meth. Eng.* **32** (1991), no. 6, p. 1205-1227.
- [36] F. Brezzi, M. Fortin, *Mixed and hybrid finite element methods*, Springer Series in Computational Mathematics, vol. 15, Springer, 1991.
- [37] G. Karypis, V. Kumar, "METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices", Tech. report, Department of Computer Science, University of Minnesota, 1998, <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [38] F. Hecht, "New development in Freefem++", *J. Numer. Math.* **20** (2012), no. 3-4, p. 251-265.
- [39] P.-H. Tournier, F. Nataf, "FFDDM: Freefem domain decomposition method", 2019, <https://doc.freefem.org/documentation/ffddm/index.html>.
- [40] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, J. Koster, "A fully asynchronous multifrontal solver using distributed dynamic scheduling", *SIAM J. Matrix Anal. Appl.* **23** (2001), no. 1, p. 15-41.
- [41] R. B. Lehoucq, D. C. Sorensen, C. Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, Software - Environments - Tools, vol. 6, Society for Industrial and Applied Mathematics, 1998.
- [42] C. Chevalier, F. Pellegrini, "PT-SCOTCH: a tool for efficient parallel graph ordering", *Parallel Comput.* **34** (2008), no. 6-8, p. 318-331.
- [43] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, "Efficient management of parallelism in object oriented numerical software libraries", in *Modern Software Tools in Scientific Computing*, Birkhäuser, 1997, p. 163-202.
- [44] P. Jolivet, F. Nataf, "HPDDM: High-Performance Unified framework for Domain Decomposition methods, MPI-C++ library", 2014, <https://github.com/hpddm/hpddm>.
- [45] P. Jolivet, J. E. Roman, S. Zampini, "KSPHPDDM and PCHPDDM: Extending PETSc with advanced Krylov methods and robust multilevel overlapping Schwarz preconditioners", *Comput. Math. Appl.* **84** (2021), p. 277-295.
- [46] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, H. Zhang, "PETSc users manual", Tech. Report ANL-95/11, Argonne National Laboratory, 2014.
- [47] Y. Efendiev, J. Galvis, T. Y. Hou, "Generalized multiscale finite element methods (GMsFEM)", *J. Comput. Phys.* **251** (2013), p. 116-135.
- [48] P. Jenny, S. H. Lee, H. A. Tchelepi, "Adaptive multiscale finite-volume method for multiphase flow and transport in porous media", *Multiscale Model. Simul.* **3** (2005), no. 1, p. 50-64.
- [49] C. Ma, R. Scheichl, T. Dodwell, "Novel design and analysis of generalized FE methods based on locally optimal spectral approximations", <https://arxiv.org/abs/2103.09545>, 2021.