INSTITUT DE FRANCE
Académie des sciences

Comptes Rendus

Mécanique

Frédéric Nataf and Pierre-Henri Tournier

Recent advances in domain decomposition methods for large-scale saddle point problems

Published online: 3 October 2022

https://doi.org/10.5802/crmeca.130

Part of Special Issue: More than a half century of Computational Fluid Dynamics

Guest editor: Mohammed El Ganaoui (Professeur des Universités, Spécialiste : Mécanique des Fluides et Transferts de Chaleur et de Masse, Université de Lorraine)

Les Comptes Rendus. Mécanique *sont membres du Centre Mersenne pour l'édition scientifique ouverte*
www.centre-mersenne.org
e-ISSN : 1873-7234

Comptes Rendus
Mécanique
Published online: 3 October 2022
https://doi.org/10.5802/crmeca.130

More than a half century of Computational Fluid Dynamics / *Plus d'un demi-siècle de mécanique des fluides numérique*

Recent advances in domain decomposition methods for large-scale saddle point problems

Progrès récents dans les méthodes de décomposition de domaine pour le problème du point de selle à grande échelle

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. Scalability of parallel solvers for problems with high heterogeneities relies on adaptive coarse spaces built from generalized eigenvalue problems in the subdomains. The corresponding theory is powerful and flexible but the development of an efficient parallel implementation is challenging. We report here on recent advances in adaptive coarse spaces and on their open source implementations in domain specific languages such as FreeFem, focusing on a new domain decomposition for saddle point formulations with some numerical tests.

Résumé. L'extensibilité des solveurs parallèles pour les problèmes à forte hétérogénéité repose sur des espaces grossiers adaptatifs construits à partir de problèmes de valeurs propres généralisés dans les sous-domaines. La théorie correspondante est puissante et flexible mais le développement d'une implémentation parallèle efficace est un défi. Nous présentons ici les avancées récentes en matière d'espaces grossiers adaptatifs et leurs implémentations open source dans des langages spécifiques au domaine tels que FreeFem, en nous concentrant sur une nouvelle décomposition de domaine pour les formulations de points de selle avec des tests numériques.

Keywords. High performance computing, Saddle point problem, Domain decomposition methods, Nearly incompressible elasticity, Multilevel methods.

Mots-clés. Calcul haute performance, Problème de point selle, Méthodes de décomposition de domaine, Elasticité presque incompressible, Méthode multi niveaux.

Published online: 3 October 2022

* Corresponding author.

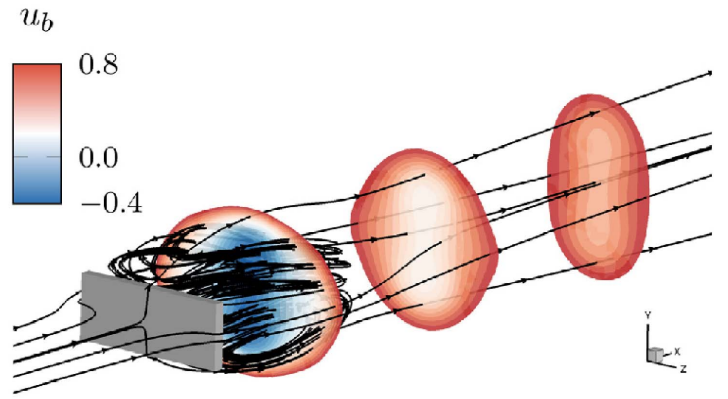
1. Introduction

The clock speed of cores have stagnated at 2–3 GHz since approximately the year 2005. The increase in performance is almost entirely due to the increase in the number of cores per processor. All processor vendors are producing multicore chips and now every machine is a parallel machine. This assertion is true for personal computers (with few cores) as well as for high performance computers (with tens to hundreds of thousands of cores). The evolution of GPU graphics cards is another example of this strong tendency. Waiting for the next generation machine does not guarantee anymore a better performance of a software. To keep doubling performance, parallelism must double. The consequence is that the development of parallel algorithms is mandatory in order to take advantage of the current and next generations of computers. It implies a huge effort in algorithmic development. Scientific computing is only one illustration of this general need in computer science. Visualization, data storage, mesh generation, operating systems, languages,... must be designed with parallelism in mind.

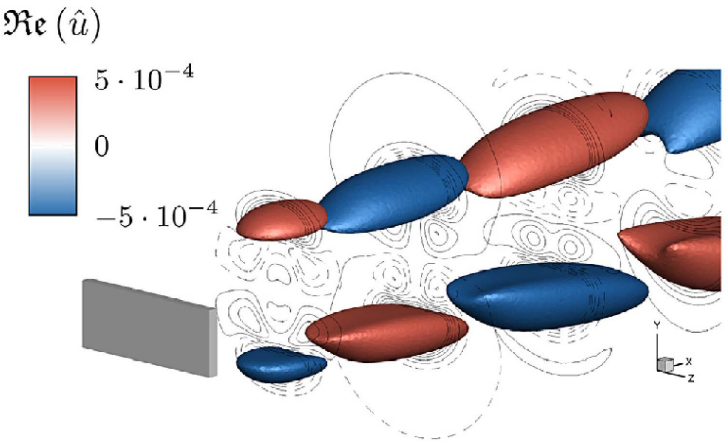
For computational fluid dynamics simulation codes as well, parallelism is nowdays "a has to have". For developers, it means integrating in their codes more and more complex tools such as parallel mesh generation, graph partitioners (e.g., METIS or Scotch) for mesh decomposition, parallel linear solvers (e.g., PETSc, Trilinos, MUMPS, PARDISO, SuperLU DIST, parallel multigrid or domain decomposition (DD)r science tools such as OpenMP for shared parallelism and/or MPI (Message Passing Interface) for distributed parallelism as well as OpenCL or Cuda for taking advantage of GPUs. Such parallel codes are thus more and more difficult to develop in small or middle size teams and demand expertise in fields that are complex and often remote from specialists in fluid dynamics. Commercial software and domain specific languages (DSL) are two solutions to this problem as they provide a user interface that hides a great deal of all these difficulties. We focus here on DSL which are more adapted to new technological or environmental challenges. Indeed, by their very nature, the implied new modelisations are easy to integrate by the end user. They also give a great flexibility in the exploitation of simulation results since the end user can write their own analysis scripts.

Simulation DSLs such as opensource libraries e.g., FreeFem, OpenFOAM, FEniCS, FireDrake, Feel++, GetFEM or commercial packages e.g., Comsol are now well established to easily handle multiphysics simulations as well as parallelism. We focus here on FreeFEM since it will be used in the sequel for the implementation and test of a new DD method for saddle point problem. Specifically a FreeFEM script is very close to the mathematical variational formulation of the equations. This allows non-computational experts to solve non-trivial PDEs on parallel computers through a user friendly language. This is one of the reasons why it can be used for both teaching and prototyping, in academia as well as in industry. As an example, it was used in [1] for performing large-scale hydrodynamic stability analysis, see Figure 1 or also Figure 2 taken from [2] as an example of parallel computations made with the DSL FreeFem.

In this paper, we focus on linear solvers based on DD methods for solving large-scale saddle point problems. These problems are ubiquitous in solid, fluid and fluid structure mechanics, inverse problems and optimization. For incompressible or nearly incompressible materials (solids or fluids), the stability of the approximation scheme makes it necessary to introduce an unknown pressure. When the problem is small enough, a direct method is the one of choice. They are readily available via well established open source libraries e.g., MUMPS, SUPER_LU DIST or commercial ones e.g., PARDISO or the MKL. To some extent, they offer some degree of parallelism but only for some tens of cores at best. But for large problems of say tens of millions of dofs or more, their memory requirements are so large that they cannot be used in practice, see Section 3.2.1. As for iterative solvers, when the kernel of matrix B



(a) Steady solution



(b) High-frequency unstable eigenmode

Figure 1. Linear stability analysis at $Re = 100$ with 75 M dofs. Streamwise velocity contours of (a) the steady solution and (b) the high-frequency unstable eigenmode, see Moulin *et al.* [1].

(see (12)) is known, very efficient multigrid methods have been designed in the context of finite element methods, see e.g., [3–8]. But for problems with high heterogeneities, this kernel is generally not known. Without this knowledge, it is nevertheless also possible to design efficient geometric multigrid methods as in [9] where the fine mesh is obtained by several uniform mesh refinements. We focus here on DD methods which have the advantage of being naturally parallel. Moreover, DD preconditioners easily interoperate with other multilevel solvers, e.g., for fluid–structure interaction, as it has been done in the context of shape optimization, see e.g., [10].

In Section 2, we recall the basics of overlapping DD methods i.e., one-level method and the need for a second level in order to achieve scalability as well as the more advanced GenEO coarse space [11]. Then in Section 3 we present a novel DD method that extends the GenEO method to saddle point problems. We also provide numerical tests for very large-scale problems on thousands of computers. In Section 4, we explain why the use of a domain specific language makes the method available to non-experts in parallel computing.

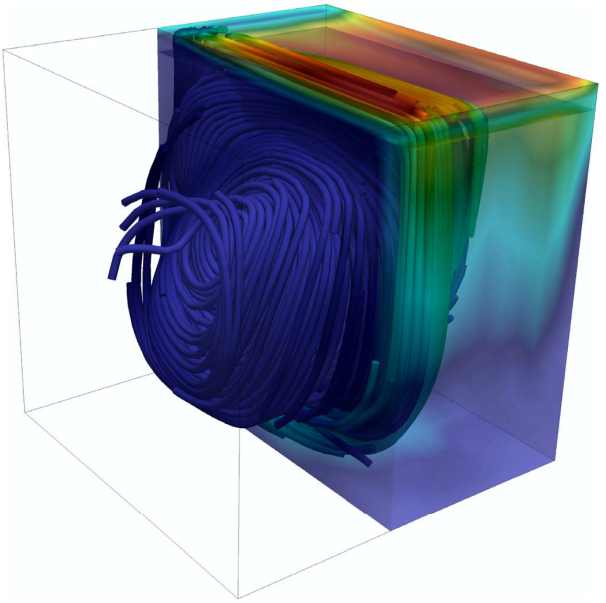


Figure 2. Flow stream lines for the driven cavity at $Re = 7500$ (Haferssas, Jolivet and Rubino, see [2]). At each iteration, a saddle point problem is solved with the method explained below.

2. Domain decomposition methods

2.1. *One-level Schwarz methods*

Hermann Schwarz was a German analyst of the 19th century. He was interested in proving the existence and uniqueness of the Poisson problem. At his time, there were no Sobolev spaces nor Lax–Milgram theorem. The only available tool was the Fourier transform, limited by its very nature to simple geometries. In order to consider more general situations, Schwarz devised an iterative algorithm for solving the Poisson problem set on a union of simple geometries, see [12]. For a historical presentation of these kinds of methods see [13].

Let the domain Ω be the union of a disk and a rectangle, see Figure 3. Consider the Poisson problem which consists in finding $u : \Omega \to \mathbb{R}$ such that:

$$\begin{aligned} -\Delta(u) &= f \text{ in } \Omega \\ u &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{1}$$

Definition 1 (Original Schwarz algorithm). The Schwarz algorithm is an iterative method based on solving alternatively sub-problems in domains Ω_1 and Ω_2. It updates $(u_1^n, u_2^n) \to (u_1^{n+1}, u_2^{n+1})$ by:

$$\begin{aligned} -\Delta(u_1^{n+1}) &= f && \text{in } \Omega_1 \\ u_1^{n+1} &= 0 && \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} &= u_2^n && \text{on } \partial\Omega_1 \cap \overline{\Omega_2}. \end{aligned} \quad \text{then,} \quad \begin{aligned} -\Delta(u_2^{n+1}) &= f && \text{in } \Omega_2 \\ u_2^{n+1} &= 0 && \text{on } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} &= u_1^{n+1} && \text{on } \partial\Omega_2 \cap \overline{\Omega_1}. \end{aligned} \tag{2}$$

Schwarz proved the convergence of the algorithm and thus the well-posedness of the Poisson problem in complex geometries.

With the advent of digital computers, this method also acquired a practical interest as an iterative linear solver. Subsequently, parallel computers became available and a small modification

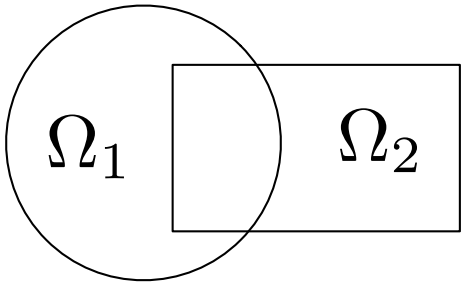


Figure 3. A complex domain made from the union of two simple geometries.

of the algorithm [14] made it suited to these architectures. Its convergence can be proved using the maximum principle [15].

Our main focus is high performance computing which implies the presence of thousands of subdomains and also to be able to work with third-party linear solver libraries. Let N be the number of subdomains, the domain Ω is decomposed into N overlapping subdomains $(\Omega_i)_{1\le i\le N}$. Rather than iterating on the collection of local approximate solutions $(u_i^n)_{1\le i\le N}$, it is then more convenient to consider related algorithms where we iterate on a global approximate solution $u^n : \Omega \to \mathbb{R}$. In order to do this, we introduce a partition of unity. We proceed here in an informal way.

Definition 2 (Extension operators and partition of unity). Let the extension operator E_i be such that $E_i(w_i) : \Omega \to \mathbb{R}$ is the *extension* of a function $w_i : \Omega_i \mapsto \mathbb{R}$, by zero outside Ω_i. We also define the *partition of unity functions* $\chi_i : \Omega_i \to \mathbb{R}$, $\chi_i \ge 0$ and $\chi_i(x) = 0$ for $x \in \partial\Omega_i \setminus \partial\Omega$ and such that:

$$w = \sum_{i=1}^{N} E_i(\chi_i \, w_{|\Omega_i}) \tag{3}$$

for any function $w : \Omega \mapsto \mathbb{R}$.

A natural generalization of (2) is the RAS (Restricted Additive Schwarz) algorithm [16]:
For an approximate solution, u^n to (1), we first solve in parallel subproblems

$$\begin{aligned} -\Delta(u_i^{n+1}) &= f && \text{in } \Omega_i \\ u_i^{n+1} &= 0 && \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^{n+1} &= u^n && \text{on } \partial\Omega_i \setminus \partial\Omega. \end{aligned} \tag{4}$$

$$\text{followed by } u^{n+1} := \textstyle\sum_{i=1}^{N} E_i(\chi_i \, u_i^{n+1}) \quad \text{in } \Omega.$$

Note that in the original algorithm (2) local solves are performed sequentially and that local approximate solutions do not match in the overlap whereas in the RAS algorithm local solves are performed in parallel and the iterate u^n is univocally defined on the global domain Ω.

2.2. *Need for global transfer of informations*

When the number of subdomains is large, plateaus appear in the convergence of Schwarz DD methods. This is the case even for a simple model such as the Poisson problem (1). The problem of the one-level method arises from the fact that in the Schwarz method there is a lack of a global exchange of information. Data are exchanged only from one subdomain to its direct neighbors. But the solution in each subdomain depends on the right-hand side in all subdomains. Let us denote by N_d the number of subdomains in one direction. Then, for instance, the leftmost

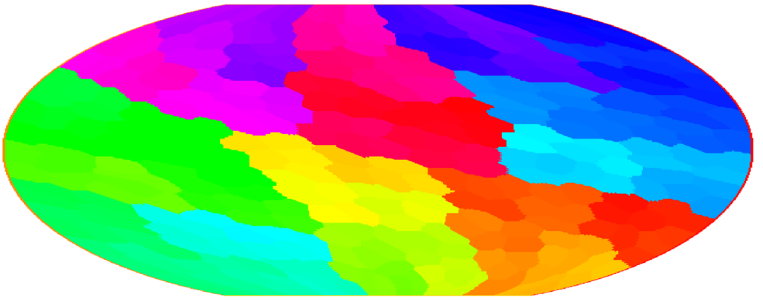


Figure 4. Decomposition into 324 subdomains. One color for each subdomain.

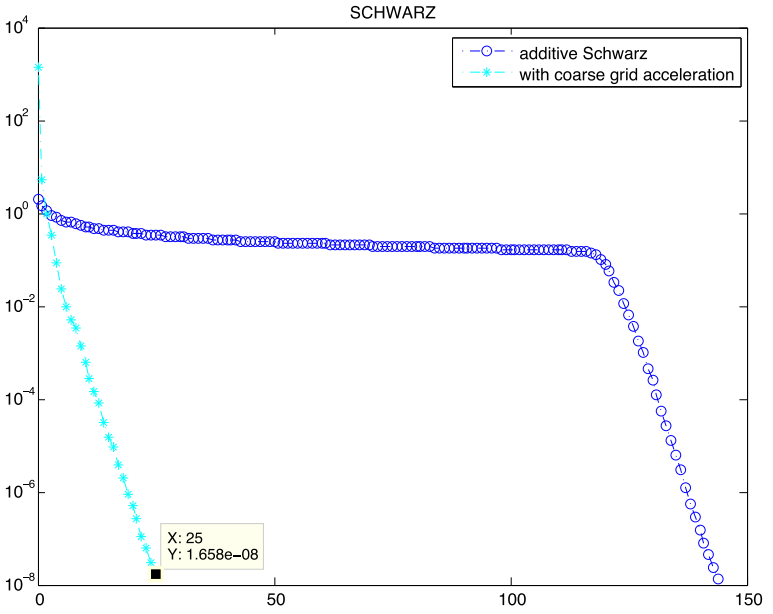


Figure 5. Convergence curves with and without a coarse space correction for a decomposition into 64 strips.

domain of Figure 4 needs at least N_d iterations before being aware about the value of the right-hand side f in the rightmost subdomain. The length of the plateau is thus typically related to the number of subdomains in one direction as exemplified in Figure 5.

2.3. *Coarse Spaces constructions*

In order to analyze and fix this weakness of the one-level method, it is necessary to go beyond the above qualitative explanation relating the length of the delay in the convergence to the diameter of the graph of the connections between subdomains. In [17] an adaptive method was proposed at the continuous level. In order to generalize it, a first key tool for this is the Fictitious Space Lemma (see [18] for the original paper and [19] for a modern exposition) which can be seen as the *Lax–Milgram theorem* of DD methods. It is formulated as an abstract result of the theory of Hilbert spaces but actually it encompasses almost all (if not all) known DD methods: Schwarz methods, Lions algorithm, Balancing Neumann–Neuman and FETI methods. The second key tool

is the deflation technique that comes more for the linear algebra community, see e.g., [20–22] and references therein. This is connected as well to augmented or recycled Krylov space methods, see e.g., [23–25] or [26] and references therein.

For instance in [11] and in [27] and references therein, it is explained how to add a so-called coarse space correction to the original one-level method in order to get a scalable method with a guaranteed condition number for the preconditioned system. More precisely, let (1) (or any symmetric positive definite system) be discretized by a finite element method so that the resulting linear to be solved reads:

$$A\mathbf{U} = \mathbf{F}, \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{\#\mathcal{N}}$ is the vector of degrees of freedom for a set of indices denoted by $\mathcal{N}$.

In order to define the DD method at the discrete level, we first decompose the global mesh $\mathcal{T}_h$ into overlapping sub meshes $(\mathcal{T}_{hi})_{1\le i\le N}$. This induces a decomposition of the global set of degrees of freedom $\mathcal{N}$ into N sub sets $(\mathcal{N}_i)_{1\le i\le N}$. Let $(R_i)_{1\le i\le N}$ be the restriction operator from the global set of indices $\mathcal{N}$ to the local one $\mathcal{N}_i$. The one-level ASM (additive Schwarz method) preconditioner which is a symmetrized version of the RAS method reads:

$$M_{\mathrm{ASM}}^{-1} := \sum_{i=1}^{N} R_i^T \, (R_i \, A R_i^T)^{-1} \, R_i. \tag{6}$$

The coarse space is defined in several steps.

First, let D_i be diagonal square matrices of size $\#\mathcal{N}_i$ that define a partition of unity, that is for all $\mathbf{U} \in \mathbb{R}^{\#\mathcal{N}}$:

$$\mathbf{U} = \sum_{i=1}^{N} R_i^T \, D_i \, R_i \, \mathbf{U}. \tag{7}$$

A simple choice for D_i is that for each degree of freedom $k \in \mathcal{N}_i$, we set $(D_i)_{kk} := 1/\mu_k$ where μ_k denotes the number of subsets k belongs to. Note that other choices are possible and popular particularly the ones for which D_i has a zero entry for degrees of freedom that live on the interface.

Now, in each subdomain, let A_i^{Neu} be the matrix associated to the restriction of the variational formulation to the sub mesh $\mathcal{T}_{hi}$. We define a generalized eigenvalue problem

$$D_i R_i \, A R_i^T \, D_i \mathbf{V}_{ik} = \lambda_{ik} A_i^{\mathrm{Neu}} \mathbf{V}_{ik}. \tag{8}$$

Let τ be a user-defined threshold, we assume that the rectangular matrix Z_0 defined by the concatenation of the vectors $R_i^T D_i \mathbf{V}_{ik}$ for all $1 \le i \le N$ such that $\lambda_{ik} > 1/\tau$ is full rank. The coarse space is defined as the range of Z_0.

Following [28], a two-level preconditioner is defined as follows:

$$\boxed{M_{2,\mathrm{HSM}}^{-1} := Z_0 (Z_0^T A Z_0)^{-1} Z_0^T + (I - P_0) \, M_{\mathrm{ASM}}^{-1} (I - P_0^T),} \tag{9}$$

where P_0 is the A orthogonal projection on the coarse space V_0:

$$P_0 := Z_0 (Z_0^T A Z_0)^{-1} Z_0^T A. \tag{10}$$

It is then possible to have a full control of the spectrum of the preconditioned operator, see Theorem 7.23 in [27]:

Theorem 3 (Hybrid Schwarz algorithm). *Let τ be a user-defined parameter to build the GenEO coarse space as above.*

The eigenvalues of the hybrid Schwarz preconditioned system satisfy the following estimate

$$\boxed{\frac{1}{1 + k_1 \, \tau} \le \lambda(M_{2,\mathrm{HSM}}^{-1} A) \le k_0,} \tag{11}$$

where k_0 is the number of neighbors of a subdomain plus one and k_1 is the maximum multiplicity of the intersection between subdomains.

We have considered here only the additive Schwarz method but this kind of result has been extended to FETI method [29, 30], P. L. Lions algorithm [31], inexact coarse solve [32], boundary element methods [33], multiscale finite element methods [34], time-dependent Maxwell system [35], least square problems [36], purely algebraic settings [36, 37] and very recently to saddle point problems [38] that we develop below.

3. Saddle point

Solving saddle point problems with parallel algorithms is very important for many branches of scientific computing: fluid and solid mechanics, computational electromagnetism, inverse problems and optimization.

We are interested in DD methods since they are naturally well-fitted to modern parallel architectures. For specific systems of partial differential equations with a saddle point formulation, efficient DD methods have been designed, see e.g., [39–41] and [42] references therein. Also in [43], a GenEO coarse space is introduced for the P. L. Lions' algorithm and its efficiency is mathematically proved for symmetric definite positive problems. In the above article, numerical experiments are conducted on 3D elasticity problems for steel–rubber structures discretized by a finite element with continuous pressure. Although the method works well in practice, the method 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, a recent article [44] generalizes the theory developed in [45] to the case of non-zero 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 bypasses the need for absorbing boundary conditions.

Here as in [46–49], 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 be an $n \times n$ SPD matrix and B be a sparse $m \times n$ full rank matrix of constraints and C an $m \times m$ non-negative matrix (in particular, $C = 0$ is allowed), we consider the following saddle point matrix:

$$\mathscr{A} := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}. \tag{12}$$

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., [3–8]. Without this knowledge, it is nevertheless also possible to design efficient geometric multigrid methods as in [9] where the fine mesh is obtained by several uniform mesh refinements.

Here we do not assume any knowledge on the kernel of matrix B and we work with arbitrary meshes. The following three factor factorization, see e.g., [50]:

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

shows that solving the linear system with $\mathscr{A}$ can be performed by solving sequentially two 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 [27].

3.1. *Two-level DD for Saddle Point problems*

We sketch here the extension of the GenEO preconditioner to the Saddle Point problem, more details are given in [38]. We consider the preconditioning of a saddle point matrix $\mathscr{A}$ arising from the discretization of e.g., Stokes, nearly incompressible elasticity systems:

$$\mathscr{A} := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}.$$

It is well known that it is equivalent to the precondition A and $S := C + BA^{-1}B^T$, see e.g., [50]. Starting with $A^{-1} \approx M^{-1}_{\mathrm{ASM}2}$, defined as follows where Z_0 is defined above in Section 2.3 by solving generalized eigenvalue problems as in (8):

$$\boxed{M^{-1}_{\mathrm{ASM}2} := Z_0 (Z_0^T A Z_0)^{-1} Z_0^T + M^{-1}_{\mathrm{ASM}},} \tag{13}$$

we have

$$S \approx C + B M^{-1}_{\mathrm{ASM}2} B^T = B\, R_0^T\, (R_0 A R_0^T)^{-1}\, R_0\, B^T + C + \sum_{i=1}^{N} B R_i^T\, (R_i A R_i^T)^{-1}\, R_i B^T.$$

Matrices B and C are sparse so that we can introduce sparse matrices $(\tilde{C}_i, \tilde{R}_i)_{1\le i\le N}$ so that

$$S \approx C + B M^{-1}_{\mathrm{ASM}2} B^T \approx S_0 + \underbrace{\sum_{i=1}^{N} \tilde{R}_i^T (\tilde{C}_i + \tilde{B}_i\, (R_i A R_i^T)^{-1}\, \tilde{B}_i^T) \tilde{R}_i}_{M_{S_1}},$$

where $S_0 := B\, R_0^T\, (R_0 A R_0^T)^{-1}\, R_0\, B^T$. The operator M_{S_1} is dense and has to be preconditioned which at first glance seems difficult. But as a sum of local Schur complements, it can be preconditioned by a GenEO Neumann–Neumann type method where the rectangular matrix Z_{S_1} is built by solving local generalized eigenvalue problems in the pressure space similar to what is done in the GenEO method:

$$M^{-1}_{S_1} := Z_{S_1}\, (Z_{S_1}^T S_1 Z_{S_1})^{-1}\, Z_{S_1}^T + \left(\sum_{i=1}^{N} \tilde{R}_i^T \tilde{D}_i\, (I - \xi_i)(\tilde{C}_i + \tilde{B}_i\, (R_i A R_i^T)^{-1}\, \tilde{B}_i^T)^{\dagger}\, (I - \xi_i^T) \tilde{D}_i \tilde{R}_i \right),$$

where the $(\xi_i)_{1\le i\le N}$ are local projections. We can then define N_S as a spectrally equivalent preconditioner to S:

$$N_S := S_0 + M_{S_1}.$$

The application of the preconditioner N_S consists in solving:

$$N_S \mathbf{P} = \mathbf{G}, \tag{14}$$

by a Krylov solver with $M^{-1}_{S_1}$ as a preconditioner. Having spectrally equivalent preconditioners to both A and S is sufficient to have a spectrally equivalent preconditioner to the saddle point matrix $\mathscr{A}$, see e.g., [51].

3.2. *Numerical results*

We perform here numerical tests on a saddle point problem arising in solid mechanics that has the ingredients that make a saddle point difficult to solve: highly heterogeneous coefficients and near incompressibility. More precisely, we present weak scalability results for the heterogeneous beam composed of ten alternating layers of rubber $(E_1, \nu_1) = (1 \times 10^7, 0.4999)$ and steel $(E_2, \nu_2) = (2 \times 10^9, 0.35)$. 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 16,800 cores.

Table 1. Weak scaling experiment for 3D heterogeneous elasticity: beam with ten alternating layers of steel and rubber

#Cores	n	$\dim(V_0)$	$\dim(\tilde{W}_0)$	Setup (s)	#It	GMRES (s)	Total (s)	#It N_S^{-1}
262	15,987,380	5383	3319	710.7	24	631.6	1342.3	11
525	27,545,495	9959	2669	526.6	21	519.5	1046.1	12
1050	64,982,431	17,837	4587	675.2	22	665.9	1341.1	11
2100	126,569,042	32,361	7995	689.2	25	733.8	1423.0	10
4200	218,337,384	59,704	13,912	593.0	27	705.4	1298.4	10
8400	515,921,881	141,421	25,949	735.8	32	1152.5	1888.3	10
16,800	1006,250,208	260,348	41,341	819.2	29	1717.9	2537.1	12

Reported iteration counts, coarse space dimensions and timings for DD saddle point.

We report in Table 1 the iteration counts and computing times for the DD saddle point. The stopping criterion is a tolerance smaller than 10^{-5}. Moreover, we use flexible GMRES, as we solve (14) 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 $\dim(\tilde{W}_0)$, 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 (14). All timings are reported in seconds.

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 for the GenEO coarse space 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 $Z_{S_1}^T S_1 Z_{S_1}$ as their size increases: total time spent in coarse space solves is 14.7, 62.1 and 679.3 seconds on 262, 4200 and 16,800 cores respectively. A possible improvement would be to use a multi-level method to solve the coarse problems iteratively.

3.2.1. *Comparison with direct and multigrid solvers*

In Table 2, 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 2 that the DD saddle point solver offers much better strong scalability.

Table 2. Comparison with the parallel sparse direct solver *MUMPS* for 3D heterogeneous elasticity: beam with ten alternating layers of steel and rubber

n	#Cores	MUMPS			DD saddle point solver			
		Setup (s)	Solve (s)	Total (s)	Setup (s)	#It	GMRES (s)	Total (s)
139,809	16	7.1	0.1	7.2	27.1	18	19.7	46.8
1058,312	32	85.7	0.8	86.5	166.2	20	137.2	303.4
1058,312	65	71.0	0.6	71.6	91.0	21	77.1	168.1
1058,312	131	63.2	0.5	63.7	59.7	24	49.7	109.4
3505,582	55	477.8	3.7	481.5	404.1	24	430.1	834.2
3505,582	110	392.3	2.3	394.6	242.5	23	212.8	455.3
3505,582	221	387.0	2.1	389.1	134.8	23	109.4	244.2
3505,582	442	453.9	2.2	456.1	88.2	24	68.6	156.8
8235,197	262	OOM	/	/	278.5	25	264.3	542.8
8235,197	525	1622.1	6.1	1628.2	172.1	24	136.0	308.1
8235,197	1050	1994.3	7.4	2001.7	136.5	25	99.7	236.2

Reported timings for four discretization levels while also varying the number of cores (OOM means the computation ran out of available memory).

Table 3. GAMG versus standard GenEO for the velocity formulation on the homogeneous beam discretized with 7.9 million unknowns, using 131 and 525 cores

ν	GAMG		DD saddle point solver				
	#It	Total (s)	$\dim(V_0)$	Setup (s)	#It	GMRES (s)	Total (s)
131 cores							
0.48	60	67.1	10,480	200.7	24	11.2	212.0
0.485	109	89.0	10,480	199.5	27	12.7	212.2
0.49	210	137.0	10,480	202.0	32	15.0	217.0
0.495	>2000	/	10,480	199.9	43	20.2	220.1
0.499	>2000	/	10,480	199.2	99	48.6	247.7
525 cores							
0.48	56	25.5	41,766	60.4	18	5.0	65.4
0.485	60	26.1	41,984	60.9	20	5.3	66.2
0.49	116	33.3	42,000	60.4	23	5.9	66.3
0.495	>2000	/	42,000	60.4	32	7.6	68.1
0.499	>2000	/	42,000	60.6	95	20.3	81.0

Reported iteration counts and timings for different values of the Poisson ratio ν ranging from 0.48 to 0.499.

We also performed comparisons with the Geometric Algebraic Multigrid (GAMG) preconditioner from PETSc, see Table 3. 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 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 $\nu \geq 0.495$.

4. Libraries for large-scale computations

We have thus a strong set of mathematical techniques that enable to build efficient DD methods. But it comes at the expense of some difficulties since the implementation of these methods is not so easy due to the need to have access to the variational formulation to build the local matrices $(A_i^{\text{Neu}})_{1\leq i\leq N}$ in the generalized eigenvalue problem for computing the coarse space, see (8). Also, an efficient parallel implementation needs a careful organization of the code and the use of the message passing interface MPI on distributed memory machines.

This motivated the development of open source implementation of the GenEO coarse space:

- the C++/MPI library *hpddm* [52] either as an autonomous library or interfaced with FreeFem [53] or PETSc [54]. The library also provides access to the GCRODR method [24] useful when solving linear systems with multiple right-hand sides.
- the FreeFem DD library *ffddm* [55]
- a Dune solver as described in [56]

It is worth noticing that the access to the variational formulation is made easy if one makes use of domain specific languages or libraries such as *FreeFem* [53], *Dune* [57] or *Firedrake* [58]. It facilitates the encapsulation of two-level methods. The resulting scripts are then quite compact and do not require writing of MPI lines of code although they enable the parallel solving of all kinds of equations with the above mentioned methods.

5. Conclusion

We have introduced adaptive DD methods efficient for highly heterogeneous problems. For a saddle point problem, 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 3D problem discretized with continuous pressure. The computations were made using the FreeFem DSL that meets the ever growing need to hide from the specialist in the physical field these complexities, while still giving them the freedom to propose and test new modelizations and/or new ways to analyze simulation results.

Several issues deserve further investigations for saddle point problems. 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* [55] framework. The integration of the method in the C++/MPI library *hpddm* [52] could lead to faster codes and a more general diffusion of the saddle point preconditioner. In a different setting, the design of adaptive coarse space is strongly connected to multiscale finite element (MFE) methods (see e.g., [34, 59, 60] and references therein) and this work could be used in designing MFE or Reduced Order Methods for saddle point problems.

Conflicts of interest

The authors declare no competing financial interest.

Dedication

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Acknowledgments

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

References

[1] J. Moulin, P. Jolivet, O. Marquet, "Augmented Lagrangian preconditioner for large-scale hydrodynamic stability analysis", *Comput. Methods Appl. Mech. Eng.* **351** (2019), p. 718-743.

[2] R. Haferssas, P. Jolivet, S. Rubino, "Efficient and scalable discretization of the Navier–Stokes equations with LPS modeling", *Comput. Methods Appl. Mech. Eng.* **333** (2018), p. 371-394.

[3] 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.

[4] R. Hiptmair, "Multigrid method for H(div) in three dimensions", *Electron. Trans. Numer. Anal.* **6** (1997), no. 1, p. 133-152.

[5] R. Hiptmair, "Multigrid method for Maxwell's equations", *SIAM J. Numer. Anal.* **36** (1998), no. 1, p. 204-225.

[6] D. N. Arnold, R. S. Falk, R. Winther, "Multigrid in H(div) and H(curl)", *Numer. Math.* **85** (2000), no. 2, p. 197-217.

[7] 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.

[8] 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.

[9] D. Drzisga, L. John, U. Rude, 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.

[10] F. Feppon, G. Allaire, C. Dapogny, P. Jolivet, "Topology optimization of thermal fluid–structure systems using body-fitted meshes and parallel computing", *J. Comput. Phys.* **417** (2020), article no. 109574.

[11] 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.

[12] H. A. Schwarz, "Über einen Grenzübergang durch alternierendes Verfahren", *Vierteljahrsschr. Nat.forsch. Ges. Zür.* **15** (1870), p. 272-286.

[13] M. J. Gander, "Schwarz methods over the course of time", *Electron. Trans. Numer. Anal.* **31** (2008), p. 228-255.

[14] P.-L. Lions, "On the Schwarz alternating method. III: A variant for nonoverlapping subdomains", in *First International Symposium on Domain Decomposition Methods for Partial Differential Equations* (Philadelphia, PA) (T. F. Chan, R. Glowinski, J. Périaux, O. Widlund, eds.), SIAM, 1990.

[15] P.-L. Lions, "On the Schwarz alternating method. II", in *Domain Decomposition Methods* (T. Chan, R. Glowinski, J. Périaux, O. Widlund, eds.), SIAM, Philadelphia, PA, 1989, p. 47-70.

[16] X.-C. Cai, M. Sarkis, "A restricted additive Schwarz preconditioner for general sparse linear systems", *SIAM J. Sci. Comput.* **21** (1999), p. 239-247.

[17] 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.

[18] S. V. Nepomnyaschikh, "Mesh theorems of traces, normalizations of function traces and their inversions", *Sov. J. Numer. Anal. Math. Model.* **6** (1991), p. 1-25.

[19] M. Griebel, P. Oswald, "On the abstract theory of additive and multiplicative Schwarz algorithms", *Numer. Math.* **70** (1995), no. 2, p. 163-180.

[20] R. A. Nicolaides, "Deflation of conjugate gradients with applications to boundary value problems", *SIAM J. Numer. Anal.* **24** (1987), no. 2, p. 355-365.

[21] Y. A. Erlangga, R. Nabben, "Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices", *SIAM J. Matrix Anal. Appl.* **30** (2008), no. 2, p. 684-699.

[22] J. Tang, R. Nabben, C. Vuik, Y. Erlangga, "Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods", *J. Sci. Comput.* **39** (2009), no. 3, p. 340-370.

[23] J. Erhel, F. Guyomarc'h, "An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems", *SIAM J. Matrix Anal. Appl.* **21** (2000), no. 4, p. 1279-1299.

[24] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, S. Maiti, "Recycling Krylov subspaces for sequences of linear systems", *SIAM J. Sci. Comput.* **28** (2006), no. 5, p. 1651-1674 (electronic).

[25] Y. Saad, "Analysis of augmented Krylov subspace methods", *SIAM J. Matrix Anal. Appl.* **18** (1997), no. 2, p. 435-449.

[26] A. Chapman, Y. Saad, "Deflated and augmented Krylov subspace techniques", *Numer. Linear Algebra Appl.* **4** (1997), no. 1, p. 43-66.

[27] V. Dolean, P. Jolivet, F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory and Parallel Implementation*, SIAM, Philadelphia, PA, 2015.

[28] J. Mandel, "Balancing domain decomposition", *Commun. Appl. Numer. Methods* **9** (1992), p. 233-241.
[29] N. Spillane, D. Rixen, "Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms", *Int. J. Numer. Methods Eng.* **95** (2013), no. 11, p. 953-990.
[30] P. Gosselet, D. Rixen, F.-X. Roux, N. Spillane, "Simultaneous FETI and block FETI: Robust domain decomposition with multiple search directions", *Int. J. Numer. Methods Eng.* **104** (2015), no. 10, p. 905-927.
[31] 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.
[32] F. Nataf, "Mathematical analysis of robustness of two-level domain decomposition methods with respect to inexact coarse solves", *Numer. Math.* **144** (2020), p. 811-833.
[33] P. Marchand, X. Claeys, P. Jolivet, F. Nataf, P.-H. Tournier, "Two-level preconditioning for the h-version boundary element approximation of hypersingular operator with GenEO", *Numer. Math.* **146** (2020), no. 3, p. 597-628.
[34] C. Ma, R. Scheichl, T. Dodwell, "Novel design and analysis of generalized finite element methods based on locally optimal spectral approximations", *SIAM J. Numer. Anal.* **60** (2022), no. 1, p. 244-273.
[35] N. Bootland, V. Dolean, F. Nataf, P.-H. Tournier, "Two-level DDM preconditioners for positive Maxwell equations", 2020, preprint, https://arxiv.org/abs/2012.02388.
[36] H. Al Daas, P. Jolivet, J. A. Scott, "A robust algebraic domain decomposition preconditioner for sparse normal equations", *SIAM J. Sci. Comput.* **44** (2022), no. 3, p. A1047-A1068.
[37] L. Gouarin, N. Spillane, "Fully algebraic domain decomposition preconditioners with adaptive spectral bounds", 2021, preprint, https://arxiv.org/abs/2106.10913.
[38] F. Nataf, P.-H. Tournier, "A GenEO domain decomposition method for saddle point problems", 2021, preprint, https://arxiv.org/abs/1911.01858.
[39] J. E. Pasciak, J. Zhao, "Overlapping Schwarz methods in H(curl) on polyhedral domains", *J. Numer. Math.* **10** (2002), no. 3, p. 221-234.
[40] 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.
[41] 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.
[42] A. Toselli, O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer Series in Computational Mathematics, vol. 34, Springer, Berlin, Heidelberg, 2005.
[43] 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.
[44] O. Widlund, S. Zampini, S. Scacchi, L. 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.
[45] 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.
[46] 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.
[47] M. Benzi, G. H. Golub, J. Liesen, "Numerical solution of saddle point problems", *Acta Numer.* **14** (2005), p. 1-137.
[48] 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.
[49] T. Rees, M. Wathen, "An element-based preconditioner for mixed finite element problems", *SIAM J. Sci. Comput.* **43** (2021), no. 5, p. S884-S907.
[50] 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, Berlin, 2008, p. 195-211.
[51] Y. Notay, "Convergence of some iterative methods for symmetric saddle point linear systems", *SIMAX* **40** (2019), p. 122-146.
[52] P. Jolivet, F. Nataf, "HPDDM: High-Performance Unified framework for Domain Decomposition methods, MPI-C++ library", 2014, https://github.com/hpddm/hpddm.
[53] F. Hecht, "New development in FreeFem++", *J. Numer. Math.* **20** (2012), no. 3–4, p. 251-265.
[54] 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* (E. Arge, A. M. Bruaset, H. P. Langtangen, eds.), Birkhäuser Press, Boston, MA, 1997, p. 163-202.
[55] P.-H. Tournier, F. Nataf, "FFDDM: FreeFem domain decomposition method", 2019, https://doc.freefem.org/documentation/ffddm/index.html.
[56] R. Butler, T. Dodwell, A. Reinarz, A. Sandhu, R. Scheichl, L. Seelinger, "High-performance dune modules for solving large-scale, strongly anisotropic elliptic problems with applications to aerospace composites", *Comput. Phys. Commun.* **249** (2020), article no. 106997.

[57] P. Bastian, F. Heimann, S. Marnach, "Generic implementation of finite element methods in the distributed and unified numerics environment (DUNE)", *Kybernetika* **46** (2010), no. 2, p. 294-315.

[58] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. McRae, G.-T. Bercea, G. R. Markall, P. H. Kelly, "Firedrake: automating the finite element method by composing abstractions", *ACM Trans. Math. Softw. (TOMS)* **43** (2016), no. 3, p. 1-27.

[59] Y. Efendiev, J. Galvis, T. Y. Hou, "Generalized multiscale finite element methods (GMsFEM)", *J. Comput. Phys.* **251** (2013), p. 116-135.

[60] 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.