



High Performance Krylov Subspace Solvers with Preconditioning and Deflation

Nicole Spillane

► To cite this version:

Nicole Spillane. High Performance Krylov Subspace Solvers with Preconditioning and Deflation. Numerical Analysis [math.NA]. Institut polytechnique de Paris, 2025. <tel-05344757>

HAL Id: tel-05344757

<https://hal.science/tel-05344757v1>

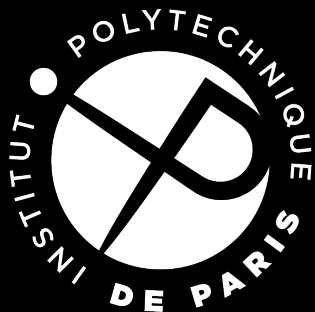
Submitted on 3 Nov 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



Habilitation à Diriger des Recherches
Institut Polytechnique de Paris
CMAP (CNRS, École polytechnique)

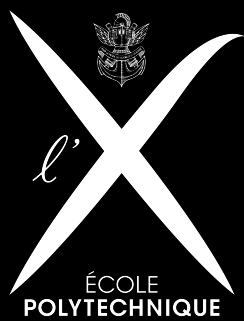
Soutenue le 24 octobre 2025

High Performance Krylov Subspace Solvers with Preconditioning and Deflation

by Nicole SPILLANE

Jury Members

Grégoire ALLAIRE Professor at École polytechnique	Reviewer & Examiner
Stéphanie CHAILLAT CNRS Senior Researcher at ENSTA	Examiner
Mark EMBREE Professor at Virginia Tech	Examiner
Virginie ERLACHER Professor at École des Ponts	Examiner
Martin GANDER Professor at Université de Genève	Reviewer & Examiner
Luc GIRAUD Senior Researcher at Inria Bordeaux	Examiner
Laura GRIGORI Professor at EPFL	Examiner
Axel KLAWONN Professor at University of Cologne	Examiner
Yousef SAAD Professor at University of Minnesota	Reviewer



HIGH PERFORMANCE KRYLOV SUBSPACE SOLVERS WITH PRECONDITIONING AND DEFLATION**Abstract**

Linear solvers are an essential tool for the accurate numerical simulation of large scale problems. My work addresses the analysis, development and application of linear solvers for problems that arise in science or the industry. My particular focus is on a class of iterative linear solvers called the Krylov subspace methods. Together with the accelerators that are preconditioning and deflation, Krylov subspace methods offer a flexible framework for a wide range of problem. The contributions in this manuscript divide into three main topics.

A first set of results is for symmetric positive definite matrices solved by the conjugate gradient method preconditioned by domain decomposition. A unified theory of spectral coarse spaces is developed, a new fully algebraic solver is introduced, and open-source software is made available to apply these methods.

The second set of results addresses multipreconditioning, a technique that allows users to exploit not just one, but several preconditioners. At each iteration, the space in which the solution is optimized grows much faster than with classical preconditioning. This can significantly accelerate convergence but multipreconditioned iterations are also more expensive. A significant contribution is adaptive multipreconditioning in which it is decided on the fly whether to apply a preconditioner or a multipreconditioner. This work is most advanced for symmetric positive definite problems but non-symmetric preconditioners and non-symmetric problems are also considered.

Finally, the last set of results is for non-Hermitian problems solved by GMRES. Convergence of GMRES is still not fully predictable for general matrices despite it having been an active field of research for many years already. The contributions in this manuscript are new convergence bounds that make apparent the role played by the preconditioner, the deflation operator and the choice of a weighted inner product. An important finding is that, if the matrix is positive definite, it is a good strategy to precondition its Hermitian part with a Hermitian positive definite preconditioner. Convergence then depends on how well the Hermitian part is preconditioned and on how non-Hermitian the problem is. Finally, a new spectral deflation space is proposed to improve the term in the bound that depends on the non-Hermitianness of the problem. This has the effect of accelerating convergence in practice too.

Keywords: linear system, Krylov subspace solver, GMRES, deflation, convergence, preconditioning, parallel computing

“Harder, Better, Faster, Stronger”

Daft Punk, on linear solvers I think

Acknowledgements

This work was supported in part by the ANR JCJC project DARK (research grant ANR-24-CE46-1633).

Contents

Abstract	3
Acknowledgements	7
Contents	9
1 Context and Introduction	11
1.1 Abbreviations and notation used throughout the manuscript	11
1.2 Krylov Subspace Methods with Preconditioning and Deflation	12
1.3 Curriculum Vitae	19
1.4 Main contributions and organization of the manuscript	21
2 Spectral coarse spaces for symmetric positive definite problems	25
2.1 Introduction	25
2.2 GenEO coarse spaces in the Abstract Schwarz framework [6]	26
2.2.1 Two-level DD in the Abstract Schwarz framework	27
2.2.2 Spaces spanned by eigenvectors	29
2.2.3 GenEO coarse spaces and convergence results	30
2.2.4 Some elements of the proofs	33
2.2.5 Application of the abstract result to three DD preconditioners	34
2.2.6 Scope and impact of this work	39
2.3 Fully algebraic preconditioner [12, 4]	42
2.3.1 A new matrix A_+ and its two-level GenEO preconditioners	43
2.3.2 Woodbury matrix identity for $A = A_+ - A_-$	44
2.3.3 AWG preconditioner for A with inexact coarse solve	44
2.3.4 AWG preconditioner for A with exact coarse solve	46
2.3.5 Scope and impact of this work	46
2.4 Open-source software [13]	47
2.4.1 Presentation of the code	47
2.4.2 Numerical results for AWG	47
2.4.3 Scope and impact of this work	47
3 Multipreconditioning of Krylov Subspace Methods	49
3.1 A word about results not included: Block FETI [3]	49
3.2 The adaptive multipreconditioned CG algorithm (AMPCG) [5]	49
3.2.1 Theory of PPCG	50
3.2.2 Main Result: Adaptive MPCG (AMPCG)	51
3.2.3 AMPCG optimized for BDD	55

3.2.4 Numerical Results	58
3.2.5 Parallel results: Adaptive Multipreconditioning of FETI [2]	61
3.3 Non-symmetric and algebraic preconditioner: RAS [11]	69
3.3.1 Multipreconditioning RAS	69
3.3.2 First adaptive algorithm: select search directions with a τ -test	70
3.3.3 Second adaptive algorithm: aggregate search directions	71
3.3.4 Numerical Results	71
3.4 Non-symmetric problem: Orthomin and BiCG [1]	72
3.4.1 Multipreconditioned Orthomin and multipreconditioned BiCG	72
3.4.2 Numerical results: application to non-symmetric FETI	75
3.5 Scope and impact of this work	80
4 GMRES: new convergence analysis and acceleration	83
4.1 Introduction	83
4.2 Weighted and Preconditioned GMRES [7]	84
4.2.1 General case	84
4.2.2 Hermitian positive definite preconditioning	85
4.2.3 Positive definite A	86
4.2.4 Numerical Illustration: Convection-Diffusion-Reaction	87
4.3 Weighted, Deflated and Preconditioned GMRES [8]	90
4.3.1 General case	90
4.3.2 Hermitian positive definite preconditioning for positive definite A	91
4.3.3 A new spectral deflation space	92
4.3.4 Numerical illustration: Jordan block	93
4.3.5 Numerical illustration: Convection-Diffusion-Reaction	95
4.4 Scope and impact of this work	100
5 Conclusions and Perspectives	101
5.1 Conclusions	101
5.2 Perspectives – Ongoing Work	104
5.3 Perspectives – Open Problems	106
6 Bibliography	109
6.1 List of my publications	109
6.2 Bibliography	111

Chapter 1

Context and Introduction

1.1 Abbreviations and notation used throughout the manuscript

- DD: domain decomposition;
- spd: symmetric positive definite;
- hpd: Hermitian positive definite;
- spsd: symmetric positive semi-definite;
- pd: positive definite, *i.e.*, the symmetric (respectively, Hermitian) part is spd (respectively, hpd);
- $\llbracket 1, N \rrbracket$: the set of all integers in $[1, N]$;
- I : identity matrix;
- $\cdot^\top$: transpose of a vector or matrix;
- $\cdot^*$: conjugate transpose of a vector or matrix;
- $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^\top \mathbf{x}$, and $\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x} \rangle^{1/2}$, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$;
- $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^* \mathbf{x}$, and $\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x} \rangle^{1/2}$, for any $\mathbf{x}, \mathbf{y} \in \mathbb{C}^m$;
- if M is an order m spd matrix, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$,

$$\langle \mathbf{x}, \mathbf{y} \rangle_M = \mathbf{y}^\top M \mathbf{x}, \|\mathbf{x}\|_M = \langle \mathbf{x}, \mathbf{x} \rangle_M^{1/2}, \text{ and } \mathbf{x} \perp^M \mathbf{y} \text{ if } \langle \mathbf{x}, \mathbf{y} \rangle_M = 0;$$

- if M is an order m hpd matrix, for any $\mathbf{x}, \mathbf{y} \in \mathbb{C}^m$,

$$\langle \mathbf{x}, \mathbf{y} \rangle_M = \mathbf{y}^* M \mathbf{x}, \|\mathbf{x}\|_M = \langle \mathbf{x}, \mathbf{x} \rangle_M^{1/2}, \text{ and } \mathbf{x} \perp^M \mathbf{y} \text{ if } \langle \mathbf{x}, M \mathbf{y} \rangle_M = 0;$$

- if M is an order n spsd matrix $|\mathbf{x}|_M = \langle \mathbf{x}, M \mathbf{y} \rangle^{1/2}$ for any $\mathbf{x} \in \mathbb{R}^m$;
- if M is a matrix, $\lambda(M)$ is any one of its eigenvalues;
- if M is a matrix, $M^\dagger$ denotes its pseudo-inverse (also called its Moore-Penrose inverse) as defined *e.g.* in [101, Problem 7.3.P7].

1.2 Krylov Subspace Methods with Preconditioning and Deflation

Problem

The problem considered is the solution of linear systems of the form

$$A\mathbf{x} = \mathbf{b}, \text{ where } A \in \mathbb{C}^{n \times n} \text{ is non-singular.}$$

The applications to bear in mind are ones for which A is sparse and the number n of unknowns is very large ($n \gg 10^6$). Hence, parallel solvers are a necessity. Although sparsity does not directly play a role in the proofs, it sets the scope in terms of scientific computing and programming. Indeed, sparse and dense numerical linear algebra are quite distinct fields.

On paper, it is not hard to solve these linear systems. Gaussian elimination (or LU -factorization as seen from the matrix point of view) overcomes any well posed linear system up to some permutation of its rows. In a first step, the matrix is factorized. In a second step the linear system is solved by forward and backward substitution (see Figure 1.1). With infinite time, memory and numerical accuracy there would be no need for any more algorithmic development. In the real world however, computations are tarnished by round-off errors. Time, as well as computational power and memory, are finite resources.

$$\begin{pmatrix} \boxed{\mathbf{A}} \end{pmatrix} = \begin{pmatrix} \boxed{\mathbf{L}} \end{pmatrix} \begin{pmatrix} \boxed{\mathbf{U}} \end{pmatrix} \quad \begin{pmatrix} \boxed{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} \boxed{\mathbf{U}} \end{pmatrix} \setminus \begin{pmatrix} \boxed{\mathbf{L}} \end{pmatrix} \setminus \begin{pmatrix} \boxed{\mathbf{b}} \end{pmatrix}$$

Figure 1.1: Direct Solver. Step 1: LU -factorization of A (up to a permutation of the rows). Step 2: Forward and Backward substitutions.

Solvers that rely on the *factorize then solve* paradigm are called direct solvers [61]. They have been thoroughly optimized for supercomputers. However, even state of the art direct solvers such as MUMPS [62, 25, 124, 23], PaStiX [98, 52], or SuperLU [121, 120] reach a limit where linear systems become too large for the direct approach to be the most efficient. For this reason, we turn to iterative solvers and accelerate them by applying direct solvers to well chosen subproblems.

Krylov Subspace Methods [149, Chapters 5, 6]. Starting with an initial guess $\mathbf{x}_0 \in \mathbb{C}^n$, let $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ denote the initial residual. Then, the m -th Krylov subspace is defined as

$$\mathcal{K}_m(A, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots, A^{m-1}\mathbf{r}_0\}.$$

The approximate solution $\mathbf{x}_m$ of a Krylov subspace method at iteration m is defined to be the unique element that satisfies

$$\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m(A, \mathbf{r}_0), \text{ and } \mathbf{r}_m := \mathbf{b} - A\mathbf{x}_m \in \mathcal{L}_m^\perp, \quad (1.1)$$

where $\mathcal{L}_m$ is an m -dimensional subspace of $\mathbb{C}^n$. The particular choice of $\mathcal{L}_m$ as well as the properties of A (e.g. its symmetry) define each Krylov subspace method formally. What is meant by formally is that there exist pairs of Krylov subspace methods which share the same characterization (1.1) but compute $\mathbf{x}_m$ via different algorithms (e.g., GMRES [148] and GCR [67]). Written in the residual variable (1.1) reads

$$\mathbf{r}_m \in \mathbf{r}_0 + A\mathcal{K}_m(A, \mathbf{r}_0), \text{ and } \mathbf{r}_m \in \mathcal{L}_m^\perp,$$

which is to say that $(\mathbf{r}_0 - \mathbf{r}_m)$ is the projection of the initial residual $\mathbf{r}_0$ onto $A\mathcal{K}_m(A, \mathbf{r}_0)$ parallel to $\mathcal{L}_m^\perp$. If, moreover, $\mathcal{L}_m = A\mathcal{K}_m(A, \mathbf{r}_0)$, i.e., the projection is orthogonal, then the m -th iteration is also

characterized by a minimization property as

$$\|\mathbf{r}_m\| = \|\mathbf{r}_0 - (\mathbf{r}_0 - \mathbf{r}_m)\| = \min_{\mathbf{y} \in \mathcal{K}_m(A, \mathbf{r}_0)} \|\mathbf{r}_0 - \mathbf{y}\| = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(A)\mathbf{r}_0\|,$$

where $\mathbb{P}_m$ is the set of polynomials of degree at most m .

In this manuscript, almost all of the focus is on the conjugate gradient (CG) [100] and GMRES methods. They can both be interpreted as the choice $\mathcal{L}_m = A\mathcal{K}_m(A, \mathbf{r}_0)$ but the orthogonality condition in (1.1) is ensured in two different inner products:

- GMRES uses the Euclidean inner product $\langle \mathbf{z}, \mathbf{y} \rangle_I = \mathbf{y}^* \mathbf{z}$ with the result that

$$\|\mathbf{r}_m\|_I = \min_{\mathbf{y} \in A\mathcal{K}_m(A, \mathbf{r}_0)} \|\mathbf{r}_0 - \mathbf{y}\|_I = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(A)\mathbf{r}_0\|_I.$$

- CG applies only to Hermitian positive definite (hpd) matrices A and uses the A^{-1} inner product $\langle \mathbf{z}, \mathbf{y} \rangle_{A^{-1}} = \mathbf{y}^* A^{-1} \mathbf{z}$, so the quantity that is minimized is

$$\|\mathbf{r}_m\|_{A^{-1}} = \min_{\mathbf{y} \in A\mathcal{K}_m(A, \mathbf{r}_0)} \|\mathbf{r}_0 - \mathbf{y}\|_{A^{-1}} = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(A)\mathbf{r}_0\|_{A^{-1}},$$

which can be rewritten with respect to the error vector $\mathbf{d}_m := \mathbf{x} - \mathbf{x}_m$ as

$$\|\mathbf{d}_m\|_A = \min_{\mathbf{y} \in \mathcal{K}_m(A, \mathbf{r}_0)} \|\mathbf{d}_0 - \mathbf{y}\|_A = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(A)\mathbf{d}_0\|_A.$$

This presentation of CG is purely formal and CG codes, of course, do not require the knowledge of A^{-1} .

A generalization of both of these algorithms is weighted GMRES, a term coined in [72]. Given a hpd matrix W , the W -inner product is defined by $\langle \mathbf{z}, \mathbf{y} \rangle_W = \mathbf{y}^* W \mathbf{z}$ and the orthogonality condition in (1.1) is ensured in the W -inner product. This is referred to as W -GMRES.

W -GMRES

The m -th iterate $\mathbf{x}_m$ is characterized by the following minimization property of the residual $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$:

$$\|\mathbf{r}_m\|_W = \min_{\mathbf{y} \in A\mathcal{K}_m(A, \mathbf{r}_0)} \|\mathbf{r}_0 - \mathbf{y}\|_W = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(A)\mathbf{r}_0\|_W, \quad (1.2)$$

where $\mathcal{K}_m(A, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots, A^{m-1}\mathbf{r}_0\}$ is the Krylov subspace and $\mathbb{P}_m$ denotes the set of all polynomials of degree at most m .

In this framework, GMRES is I -GMRES and CG is A^{-1} -GMRES.

Deflation. Krylov subspace solvers may also be applied in cases where A is singular. Of particular interest to the work in this manuscript is the following result from [45, Theorem 2.6]. As long as $\text{range}(A) \cap \ker(A) = \{0\}$ and the linear system $A\mathbf{x} = \mathbf{b}$ is consistent, GMRES breaks down only if the solution has been found.¹ So, under the aforementioned condition, characterization (1.2) holds until the solution has been found.

The singular systems that are addressed in this manuscript result from deflation, a procedure that projects $A\mathbf{x} = \mathbf{b}$ into two supplementary spaces and leaves only one of the projected problems to the Krylov subspace method. With notation from [80] the idea is as follows.

¹The proof in [45] is for unweighted GMRES but generalizes to W -GMRES.

- Let $Y, Z \in \mathbb{C}^{n \times p}$ be two full rank matrices. Under the assumption that $\ker(Y^*) \cap \text{range}(AZ) = \{\mathbf{0}\}$, let

$$P_D := I - AZ(Y^*AZ)^{-1}Y^* \text{ and } Q_D := I - Z(Y^*AZ)^{-1}Y^*A. \quad (1.3)$$

These are projection operators called the deflation operators.² Now,

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow \begin{cases} P_DA\mathbf{x} = P_D\mathbf{b} \\ (I - P_D)A\mathbf{x} = (I - P_D)\mathbf{b} \end{cases} \Leftrightarrow \begin{cases} AQ_D\mathbf{x} = P_D\mathbf{b} \\ A(I - Q_D)\mathbf{x} = (I - P_D)\mathbf{b}. \end{cases}$$

- The solution $\mathbf{x}$ splits into $\mathbf{y} := Q_D\mathbf{x}$ and $\mathbf{z} := (I - Q_D)\mathbf{x}$ and is computed in two steps:
 - The inverse in $\mathbf{z} = Z(Y^*AZ)^{-1}Y^*\mathbf{b}$ is computed by a direct solver.
 - $\mathbf{y}$ is computed by solving $P_DA\tilde{\mathbf{x}} = P_D\mathbf{b}$ with a Krylov subspace method and setting $\mathbf{y} = Q_D\tilde{\mathbf{x}}$.

The no-breakdown sufficient condition for the deflated system is $\text{range}(P_DA) \cap \ker(P_DA) = \{\mathbf{0}\}$ or equivalently $\text{range}(Z) \cap \ker(Y^*) = \{\mathbf{0}\}$.

Deflated W -GMRES

Let $Y, Z \in \mathbb{C}^{n \times p}$ be two full rank matrices. Under the assumption that $\ker(Y^*) \cap \text{range}(AZ) = \ker(Y^*) \cap \text{range}(Z) = \{\mathbf{0}\}$, let $P_D := I - AZ(Y^*AZ)^{-1}Y^*$. The application of W -GMRES to $P_DA\tilde{\mathbf{x}} = P_D\mathbf{b}$ produces iterates $\mathbf{x}_m$ and residuals $\mathbf{r}_m = P_D\mathbf{b} - P_DA\mathbf{x}_m$ characterized by

$$\|\mathbf{r}_m\|_W = \min_{\mathbf{y} \in A\mathcal{K}_m(P_DA, \mathbf{r}_0)} \|\mathbf{r}_0 - \mathbf{y}\|_W = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(P_DA)\mathbf{r}_0\|_W, \quad (1.4)$$

where $\mathcal{K}_m(P_DA, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, P_D A \mathbf{r}_0, \dots, (P_D A)^{m-1} \mathbf{r}_0\}$; $\mathbf{r}_0 = P_D \mathbf{b} - P_D A \mathbf{x}_0$.

There are two requirements for deflation to accelerate GMRES. First, p must be sufficiently small for the direct solve $(Y^*AZ)^{-1}$ to be fast. Second, W -GMRES must solve the projected problem in significantly fewer iterations than the original problem.

Deflation [136, 60, 147], augmentation [55] and balancing [105] are closely related. A review and analysis of these methods can be found in [158, 81, 153], or [116] with an emphasis on Domain Decomposition. In practice, the deflation space is often computed as an approximation of the eigenvectors corresponding to isolated eigenvalues of the preconditioned operator, for example, by recycling information from a previous linear solve [132, 139, 83, 161, 153]. This strategy is somewhat justified below.

Convergence for A diagonalizable. Convergence of (W) -GMRES is a vast, still open, topic about which more will be said later. In this introduction, let's consider the case where A is diagonalizable, *i.e.*, there exists an invertible matrix P and a diagonal matrix D such that $A = PDP^{-1}$. The minimization property can be rewritten as

$$\|\mathbf{r}_m\|_W = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(PDP^{-1})\mathbf{r}_0\|_W = \min_{q \in \mathbb{P}_m; q(0)=1} \|Pq(D)P^{-1}\mathbf{r}_0\|_W \quad (1.5)$$

and, by sub-multiplicativity of $\|\cdot\|_W$,

$$\|\mathbf{r}_m\|_W \leq \min_{q \in \mathbb{P}_m; q(0)=1} \|P\|_W \|P^{-1}\|_W \|q(D)\|_W \|\mathbf{r}_0\|_W.$$

²In [153], the projectors P_D and Q_D are called *sibling projectors*. What is meant is that defining one also defines the other unambiguously.

The quantity $\|P\|_W\|P^{-1}\|_W$ is the W -condition number of the matrix of eigenvectors P , an impractical quantity. Moreover, the usual trick of replacing $\|q(D)\|_W$ by the maximal entry of $q(D)$ does not go through unless W is diagonal. Instead, [142, Theorem 3.8] proposes to factorize W as $W = C^*C$ and operate the sub-multiplicativity in the I -norm. This leaves us with

$$\begin{aligned}\|\mathbf{r}_m\|_W &\leq \min_{q \in \mathbb{P}_m; q(0)=1} \|CPq(D)P^{-1}C^{-1}C\mathbf{r}_0\|_I \\ &\leq \min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in \sigma(A)} |q(\lambda)| \|CP\|_I \|(CP)^{-1}\|_I \|\mathbf{r}_0\|_W,\end{aligned}$$

where $\sigma(A)$ is the spectrum of A . Indeed, it can easily be checked that $\|q(D)\|_I = \max_{\lambda \in \sigma(A)} |q(\lambda)|$ since the entries in the diagonal matrix $q(D)$ are the $q(\lambda_i)$ with λ_i an eigenvalue of A . The quantity $\|CP\|_I \|(CP)^{-1}\|_I$ is the I -condition number of the matrix of weighted eigenvectors CP , an impractical quantity. Starting back from (1.5), another possibility is to apply sub-multiplicativity in the P^*WP -norm

$$\|\mathbf{r}_m\|_W = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(D)P^{-1}\mathbf{r}_0\|_{P^*WP} \leq \min_{q \in \mathbb{P}_m; q(0)=1} \|q(D)\|_{P^*WP} \|\mathbf{r}_0\|_W.$$

Finally, a less general result is for cases where the eigenvectors of A are W -orthogonal. Then, P^*WP is a diagonal matrix and the norm of $q(D)$ can be replaced by a maximum over its diagonal entries to get

$$\|\mathbf{r}_m\|_W \leq \min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in \sigma(A)} |q(\lambda)| \|\mathbf{r}_0\|_W. \quad (1.6)$$

This bound holds true in three important cases:

- $W = I$, and P unitary, which is GMRES for normal matrices,
- $W = A^{-1}$, and A hpd, which is CG,
- $W = (PP^*)^{-1}$, which is proposed in [142, Corollary 3.10].

The solution of the min-max problem is not trivial but there are at least two well-known cases where bounds are available. The first is $\sigma(A) \subset [\lambda_{\min}(A), \lambda_{\max}(A)]$ a real positive interval, then [149, Theorem 6.25] tell us that

$$\min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in \sigma(A)} |q(\lambda)| \leq \min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in [\lambda_{\min}(A), \lambda_{\max}(A)]} |q(\lambda)| = \frac{1}{C_m \left(1 + \frac{2\lambda_{\min}(A)}{\lambda_{\max}(A) - \lambda_{\min}(A)}\right)} \quad (1.7)$$

where C_m is the Chebyshev polynomial of the first kind of degree m . It follows from a property of C_m (see [149, equation (6.127)] that

$$\min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in \sigma(A)} |q(\lambda)| \leq 2 \left[\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right]^m, \text{ where } \kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)},$$

a quantity familiar to CG specialists. The second well-known case is when $\sigma(A)$ can be enclosed in an ellipse that does not contain the origin (see [149, equation (6.119)]).

Eliminating the influence of some eigenvalues. With the objective of analyzing the convergence of deflated Krylov subspace methods, let's consider the case where the initial residual is spanned only by a subset of the eigenvectors of A . In the current formalism, this corresponds to the assumption that the vector $\boldsymbol{\rho}_0 := P^{-1}\mathbf{r}_0$ has at least one zero entry. The convergence bound (1.5) holds with the entries in $q(D)$ corresponding to the zeros of $\boldsymbol{\rho}_0$ set to zero. This allows us to remove the influence of some

eigenvectors and eigenvalues in the convergence bounds. We write only the case where the eigenvectors of A are W -orthogonal.

Convergence W -GMRES

If A is diagonalizable into PDP^{-1} and the eigenvectors of A are W orthogonal, let $\sigma_{\text{act}}(A, \mathbf{r}_0, W)$ denote the subset of eigenvalues of A defined by

$$\sigma_{\text{act}}(A, \mathbf{r}_0, W) = \{\lambda_i \in \sigma(A); \langle \mathbf{y}_i, \mathbf{r}_0 \rangle_W \neq 0\},$$

where $\mathbf{y}_i$ is the i -th column of P and $A\mathbf{y}_i = \lambda_i\mathbf{y}_i$. This can be viewed as the active spectrum of A for the starting vector $\mathbf{r}_0$. Then the m -th residual of W -GMRES satisfies

$$\|\mathbf{r}_m\|_W \leq \min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in \sigma_{\text{act}}(A, \mathbf{r}_0, W)} |q(\lambda)| \|\mathbf{r}_0\|_W. \quad (1.8)$$

In particular, if A is a singular operator and $\mathbf{r}_0 \perp^W \text{Ker}(A)$ then, the convergence bound depends only on non-zero eigenvalues of A .

The eigenvalues in σ_{act} are referred to as the active eigenvalues following the terminology in [161].

Preconditioning Krylov Subspace Methods. Preconditioning [32, 163] is a technique that aims at improving the performance of an iterative solver. The idea is to provide the solver with an approximation³ H of A^{-1} . The first requirement is that applications of H must not be too costly since they occur at each iteration of the solver. Moreover, if H is a *good* guess for the inverse of A , then the preconditioned linear system is *easier* to solve than the original linear system. Providing a mathematical definition of the words *good* and *easy* in the previous sentence is a core question addressed by my work. The answer depends very much on the nature of the problem (symmetry, definiteness, ...) and is still an open problem in many cases.

Left, right and split preconditioning are three options that I choose to introduce simultaneously. Let (H_L, H_R) be a pair of matrices in $\mathbb{C}^{n \times n}$ which we call the left and right preconditioner respectively. Preconditioned GMRES is the application of GMRES to the problem

$$H_L A H_R \mathbf{u} = H_L \mathbf{b} \text{ followed by } \mathbf{x} = H_R \mathbf{u}.$$

By the characterization of the iterates of W -GMRES for the above problem, and with the adequate change of variables to go back to the $\mathbf{x}$ variable, it can be shown that the iterates satisfy the following property.

W -GMRES with (H_L, H_R)

The m -th iterate $\mathbf{x}_m$ is characterized by the following minimization property of the residual $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$:

$$\|H_L \mathbf{r}_m\|_W = \min_{\mathbf{y} \in A\mathcal{K}_m(HA, H\mathbf{r}_0)} \|H_L(\mathbf{r}_0 - \mathbf{y})\|_W = \min_{q \in \mathbb{P}_m; q(0)=1} \|H_L q(AH) \mathbf{r}_0\|_W, \quad (1.9)$$

where $H = H_R H_L$ is the combined preconditioner, $\mathcal{K}_m(HA, H\mathbf{r}_0) := \text{span}\{H\mathbf{r}_0, HA H\mathbf{r}_0, \dots, (HA)^{m-1} H\mathbf{r}_0\}$ is the Krylov subspace and $\mathbb{P}_m$ denotes the set of all polynomials of degree at most m .

The Preconditioned CG algorithm (PCG). Formally, PCG with hpd preconditioner $H = LL^*$ is the application of CG to the matrix $L^* A L$ [149, Section 9.2.1]. We deduce that PCG has the same

³It is more common to use M^{-1} as notation for the preconditioner (i.e., an approximation of A^{-1}). I prefer to use H so that it is immediately clear that H is easy to compute and H^{-1} is not necessarily.

iterates as GMRES for $L^*AL\mathbf{u} = L^*\mathbf{b}$ ($\mathbf{x} = L\mathbf{u}$) in the $(L^*AL)^{-1}$ -norm. This is restricted to A and H hpd. After cancellation of all the terms in $L^{\pm 1}$, the m -th iterate $\mathbf{x}_m$ of PCG is characterized by the following minimization property of the residual $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$:

$$\|\mathbf{r}_m\|_{A^{-1}} = \min_{\mathbf{y} \in A\mathcal{K}_m(HA, H\mathbf{r}_0)} \|\mathbf{r}_0 - \mathbf{y}\|_{A^{-1}} = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(AH)\mathbf{r}_0\|_{A^{-1}}, \quad (1.10)$$

which can also be written in the $\mathbf{x}$ variable as

$$\|\mathbf{x} - \mathbf{x}_m\|_A = \min_{\mathbf{y} \in \mathbf{x}_0 + K_m(HA, H\mathbf{r}_0)} \|\mathbf{x} - \mathbf{y}\|_A = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(HA)(\mathbf{x} - \mathbf{x}_0)\|_A.$$

PCG fits the framework for (1.8): L^*AL is diagonalizable and its eigenvectors are $(L^*AL)^{-1}$ -orthogonal. They are also (L^*AL) -orthogonal. Moreover, it can be remarked that, $L^*AL\mathbf{y} = \lambda\mathbf{y}$ if and only if $HA(L\mathbf{y}) = \lambda(L\mathbf{y})$ so HA is diagonalizable and its eigenvectors are A -orthogonal.

Convergence PCG

Suppose A and H are hpd, let $\sigma_{\text{act}}(HA, \mathbf{r}_0, I)$ denote the subset of eigenvalues of HA defined by

$$\sigma_{\text{act}}(HA, \mathbf{r}_0, I) = \{\lambda_i \in \sigma(HA); \langle \mathbf{y}_i, \mathbf{r}_0 \rangle_I \neq 0\}$$

where $(\lambda_i, \mathbf{y}_i)$ is a complete set of eigenpairs of HA . This can be viewed as the active spectrum of HA for the starting vector $\mathbf{r}_0$. Then the m -th error vector $\mathbf{d}_m = \mathbf{x} - \mathbf{x}_m$ of PCG satisfies

$$\|\mathbf{d}_m\|_A \leq \min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in \sigma_{\text{act}}(HA, \mathbf{r}_0, I)} |q(\lambda)| \|\mathbf{d}_0\|_A. \quad (1.11)$$

The PCG algorithm can be run successfully on a singular system [106] if the preconditioned problem is consistent. Here we address the particular combination of PCG with deflation.

The projected PCG algorithm (PPCG). PPCG [60, 136] is a combination of PCG and deflation. Let A be the hpd problem matrix and let us setup deflation operators following (1.3). In the hpd setting, it is natural to choose $U := Y = Z$, in which case Q_D is denoted by Π , i.e.,

$$\Pi := Q_D = I - U(U^*AU)^{-1}U^*A.$$

This entails two strong properties:

- First, Π is an A -orthogonal projection because $\ker(\Pi) = \text{range}(U) \perp^A \ker(U^*A) = \text{range}(\Pi)$,
- second, the sibling projector satisfies $P_D = \Pi^*$ as well as $A\Pi = \Pi^*A = \Pi A \Pi^*$.

Denoting by H the hpd preconditioner, Algorithm 1 describes the PPCG algorithm [60]. In the first line of Algorithm 1, the contribution $U(U^*AU)^{-1}U^*\mathbf{b}$ to $\mathbf{x}_0$ equals $(I - \Pi)\mathbf{x}$. The Krylov iterations then solve the projected problem $\Pi^*A(\Pi\mathbf{x}) = \Pi^*\mathbf{b}$ starting with $\mathbf{r}_0 = \Pi^*\mathbf{b} - \Pi^*A\mathbf{x}_0$. All search directions (the $\mathbf{p}_i$) are projected so that $\mathbf{x}_i$ remains an approximation to the solution $\mathbf{x}$ of the original problem. The usual PCG is recovered by setting $\Pi = I$, which corresponds to an empty deflation space.

It can be proved that

$$\|\mathbf{x} - \mathbf{x}_m\|_A = \min_{\mathbf{y} \in \mathbf{x}_0 + K_m(\Pi H A, \Pi H \mathbf{r}_0)} \|\mathbf{x} - \mathbf{y}\|_A = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(\Pi H A)(\mathbf{x} - \mathbf{x}_0)\|_A. \quad (1.12)$$

In fact, only non-zero eigenvalues of $HA\Pi$ influence convergence so that the following result (that is proved in Section 3.2.1 and can be found in [76]) holds.

Algorithm 1 : PPCG algorithm for $A\mathbf{x} = \mathbf{b}$ preconditioned by H and deflated by $\Pi = I - U(U^*AU)^{-1}U^*A$ for initial guess $\mathbf{x}_{00}$

```

 $\mathbf{x}_0 = \Pi\mathbf{x}_{00} + U(U^*AU)^{-1}U^*\mathbf{b};$   $\leftarrow$  Improved initial guess
 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0;$ 
 $\mathbf{z}_0 = H\mathbf{r}_0;$ 
 $\mathbf{p}_0 = \Pi\mathbf{z}_0;$   $\leftarrow$  Projected initial search direction
for  $i = 0, 1, \dots, \text{convergence}$  do
     $\mathbf{q}_i = A\mathbf{p}_i;$ 
     $\alpha_i = \frac{\langle \mathbf{r}_i, \mathbf{z}_i \rangle}{\langle \mathbf{q}_i, \mathbf{p}_i \rangle};$ 
     $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i;$ 
     $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{q}_i;$ 
     $\mathbf{z}_{i+1} = H\mathbf{r}_{i+1};$ 
     $\beta_i = \frac{\langle \mathbf{z}_{i+1}, A\mathbf{p}_i \rangle}{\langle \mathbf{p}_i, A\mathbf{p}_i \rangle};$ 
     $\mathbf{p}_{i+1} = \Pi\mathbf{z}_{i+1} - \beta_i \mathbf{p}_i;$   $\leftarrow$  Projected search direction
end
Return  $\mathbf{x}_{i+1};$ 

```

Convergence PPCG

If A and H are hpd, and $U \in \mathbb{C}^{n \times p}$ is a full rank matrix. Let $\Pi = I - U(U^*AU)^{-1}U^*A$ be the corresponding deflation operator, $\sigma(HA\Pi)$ denote the spectrum of $HA\Pi$ and

$$\kappa_{\text{eff}}(HA\Pi) := \frac{\max(\sigma(HA\Pi))}{\min(\sigma(HA\Pi) \setminus \{0\})},$$

denote the effective condition number of $HA\Pi$. The m -th error vector $\mathbf{d}_m = \mathbf{x} - \mathbf{x}_m$ of PPCG (Algorithm 1) satisfies

$$\|\mathbf{x} - \mathbf{x}_m\|_A = \min_{q \in \mathbb{P}_m; q(0)=1} \max_{\lambda \in \sigma(HA\Pi) \setminus \{0\}} \|\mathbf{x} - \mathbf{x}_0\|_A \leq 2 \left[\frac{\sqrt{\kappa_{\text{eff}}(HA\Pi)} - 1}{\sqrt{\kappa_{\text{eff}}(HA\Pi)} + 1} \right]^m \|\mathbf{x} - \mathbf{x}_0\|_A. \quad (1.13)$$

Consequently, a strategy for choosing U is to compute, or approximate, all isolated eigenvalues of HA and use the corresponding eigenvectors as a basis for the deflation space $\text{range}(U)$. This way, the spectrum of $HA\Pi$ is clustered and (1.13) guarantees good convergence.

Domain Decomposition (DD) Preconditioning. A form of preconditioning that plays an important role in my research is DD. As indicated by its name, DD partitions the simulation domain Ω into N , possibly overlapping, subspaces $(\Omega^s)_{1 \leq s \leq N}$ called the subdomains (see Figure 1.2). Each subdomain is characterized by a prolongation matrix $R^{s\top}$ and it is assumed that $\sum_{s=1}^N \text{range}(R^{s\top}) = \mathbb{R}^n$ (or $\mathbb{C}^n$). In the abstract Schwarz framework ([159][Chapter 2]), a one-level DD preconditioner takes the form

$$H = \sum_{s=1}^N R^{s\top} (\tilde{A}^s)^{-1} R^s.$$

$\begin{array}{c} \uparrow \\ \text{prolongation to } \Omega \\ \uparrow \\ \text{local solve} \\ \uparrow \\ \text{restriction to } \Omega^s \end{array}$

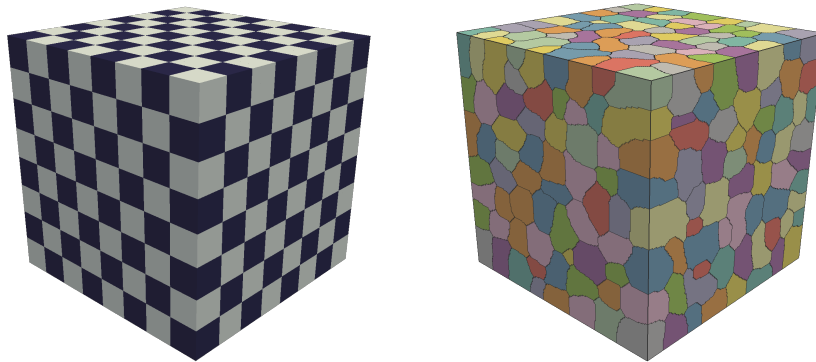


Figure 1.2: Partition of a cube into 512 subdomains. The partition can be regular (left) or irregular (right). The irregular partition was computed by the automatic graph partitioner Metis [108].

To say it with words, DD preconditioners approximate A^{-1} by a sum (interpolated by the $R^{s\top}$) of inverses of local problems $\tilde{A}^s$. DD preconditioners are particularly well suited to solving large scale problems on parallel computers. Indeed the local solves $\tilde{A}^{s-1}$ can be performed in parallel, and communication is only between neighbouring subdomains. Another advantage of DD methods is that state of the art direct solvers can be applied to the local solves. In this sense, Krylov subspace methods preconditioned by DD are hybrid (direct/iterative) methods.

The combination of DD preconditioning and deflation is a two-level DD method. Two-level DD is, both, an already well developed and a still very active field of research. In DD, the deflation space is referred to as the coarse space. During my PhD I developed spectral coarse spaces for DD preconditioners applied to symmetric positive definite problems. Very broadly speaking, since my PhD, my interest has developed to include the following challenges.

Research Questions

- Analysis of Krylov subspace methods accelerated by preconditioning and deflation. Analysis of the role played by the weight matrix. Proof of new results that make apparent the role of each accelerator.
- Development of new Krylov subspace methods. Development of new preconditioners and deflation spaces with the objective of achieving scalability, robustness and efficiency.
- Application of the methods to linear systems that arise in science and engineering.

1.3 Curriculum Vitae

Since 2016, I have been a CNRS researcher at CMAP, or Centre de Mathématiques Appliquées, of École polytechnique in Palaiseau. I am a member of two teams at CMAP

- **MMOF : Mécanique, Matériaux et Optimisation de forme**, meaning mechanics, materials and shape optimization
- **HPC : High Performance computing**.

This shows my attachment both to the field of solvers for supercomputers and to problems arising from

the applications (such as structural mechanics). Since the defense of my PhD, I have collaborated with the following list of people:

- Christophe Bovet: Researcher at ONERA, the French aerospace agency,
- Élise Fressart: PhD student at CMAP, École polytechnique (UMR 7641),
- Pierre Gosselet: CNRS senior researcher (DR) at Lamcube, Université de Lille (UMR 9013),
- Loïc Gouarin: CNRS research engineer (IR) at CMAP, École polytechnique (UMR 7641),
- Marc Massot: Full Professor at CMAP, École polytechnique (UMR 7641),
- Pierre Matalon: Research engineer (IR) at CMAP, École polytechnique (UMR 7641),
- Daniel J. Rixen: Full Professor at Technische Universität of Munich in Germany,
- François-Xavier Roux: Researcher at ONERA and Full Professor at LJLL, Université Paris Sorbonne (UMR 7598),
- Daniel Szyld: Full Professor at Temple University in the USA,

in academia, and

- Michel Nowak: Thales Research and Technology,
- Augustin Parret-Fréaud: Safran Tech,

in industry.

Mentorship activities.

- 2015-2018: Co-supervised the PhD thesis of Aloïs Bissuel with Grégoire Allaire (CMAP, École polytechnique). Topic: Solution of the linearized Navier Stokes equations for aeroelasticity, shape optimization and aeroacoustics.
- 2023 (6 months): Co-advised the Master thesis of Élise Fressart with Marc Massot (CMAP, École polytechnique) and Michel Nowak (Thales Research and Technology). Topic: Adaptive mesh refinement for an elliptic PDE.
- since 2023: Co-advising the PhD of Élise Fressart with Marc Massot (CMAP, École polytechnique) and Michel Nowak (Thales Research and Technology). Topic: Adaptive mesh refinement for harmonic wave problems on quantum computers.
- since april 2025: Co-supervision of the master thesis of Lucien Gontier with Olivier Le Maitre (CNRS, CMAP, École polytechnique and Inria). Topic: Domain decomposition for partial differential equations with stochastic coefficients.
- starting october 2025: Co-supervision of PhD thesis with Olivier Le Maitre (CNRS, CMAP, École polytechnique and Inria). Topic: Domain decomposition for partial differential equations with stochastic coefficients.

Scientific expertise (a selection).

- Program Committee member of the International Conference for Preconditioning Techniques for Scientific and Industrial Applications in Atlanta, USA. (twice in 2024 and 2026)
- Member of Maître de Conférences hiring committee; Université Paris Saclay, Sorbonne Université, Université Paris Dauphine, Université de Lille. (5 times)
- Evaluator in PhD committee (membre du jury): Institut Polytechnique de Paris - Université Paris-Saclay - Université Paris Sorbonne - Université Paris sciences et lettres - Université de Bordeaux - Liège University/ Belgium. (12 times)

Funding and contracts.

- In 2019, together with V. Dolean (TU Eindhoven, The Netherlands) and D. Szyld (Temple University, USA) we organized the conference *Parallel Solution Methods for Systems Arising from PDEs* at CIRM (Centre international de Rencontres Mathématiques), Marseille, France. We obtained funding for approximately 50 participants from CIRM, FMJH (Fondation Mathématique Jacques Hadamard), CNRS, École polytechnique and Université Côte d'Azur.
- In 2024, the master thesis of Élise Fressart at CMAP and Thales Research and Technology was funded by Thales Research and Technology.
- The PhD of Élise Fressart is a CIFRE thesis with an accompanying package allocated to CMAP.
- I am the principal investigator of ANR JCJC grant DARK, a four year project starting in February 2025.

Miscellaneous.

- In 2017, I received the **Leslie Fox Prize for Numerical Analysis** (First Prize) from the UK Institute for Mathematics and its Applications, known as the IMA. A brief presentation of the corresponding research was published in the French bulletin of the Applied Math society (SMAI) :
Matapli 116 (2018) <http://smai.emath.fr/IMG/pdf/Matapli116.pdf>
Une méthode de décomposition de domaine améliorée par multipréconditionnement dans la rubrique Leslie Fox Prize 2017 - Nicole Spillane.
- I have had three career breaks.

1.4 Main contributions and organization of the manuscript

I have organized my work since my PhD into three main lines of research. They are presented in Chapters 2, 3 and 4. Chapter 5 describes some conclusions as well as perspectives opened up by this work and/or new search directions that I plan to explore. Here is an overview of the contents.

Chapter 2 Spectral coarse spaces for symmetric positive definite problems

In this chapter, I present extensions of the GenEO coarse space.

- The first set of results (Section 2.2 – adapted from [6]) is a unified theory of DD methods with spectral coarse spaces applied to spd linear systems. Given a DD method in the abstract framework, a recipe is given to define spectral coarse spaces (of the GenEO kind) and the

resulting spectral bounds are proved. The analysis requires generalizing the usual abstract Schwarz results (stable splitting and stability of the local solvers) to deflated local spaces and singular local solvers. The new results are applied to some standard DD methods (Additive Schwarz and Neumann-Neumann) as well as one for which spectral deflation had not yet been considered (Inexact Schwarz).

- The second result for spd DD methods (Section 2.3 – adapted from [12, 4] in collaboration with Loïc Gouarin) is both a new solver and a set of preconditioners that have the particular feature of being algebraic. By this, it is meant that the only knowledge that the solver has of the problem is the matrix and right hand side. This is in opposition to the usual GenEO coarse space that requires access to some spsd submatrices from which A has been assembled. Algebraic solvers are a necessity because many users can only apply a black box linear solver. This can be either because they are not scientific computing experts or because they, themselves, don't have access to the code that produced the linear system. The new methods are called AWG for Algebraic Woodbury GenEO. The core idea is to algebraically construct an auxiliary matrix A_+ which is closely related to A and well suited to apply GenEO. Then, the Woodbury matrix identity (also known as the Sherman-Morrison-Woodbury formula) makes the connection between solving for A and for A_+ .
- The final result in this chapter is the open-source code that we have developed together with Loïc Gouarin (Section 2.4). The code is available on github [13]. It is a Python code that uses petsc4py (a Python wrapper to the PETSc library) to implement abstract DD methods with spectral coarse spaces. As an illustration, results obtained for AWG by this code are presented. Adaptive multipreconditioning, the method which I present next is also implemented in our code.

Chapter 3 Multipreconditioning of Krylov Subspace Methods

Multipreconditioning is a technique that was introduced in 2006 [43] but whose potential had not been fully exploited. The idea is to accelerate a Krylov subspace method by enlarging the space in which each approximate solution is computed. At each iteration, Krylov subspace methods such as preconditioned GMRES and PCG, minimize the residual in a space that has dimension m , see e.g., (1.9) where the minimization space is A times

$$\mathcal{K}_m(HA, H\mathbf{r}_0) = \text{span}\{H\mathbf{r}_0, HA H\mathbf{r}_0, \dots, (HA)^{m-1} H\mathbf{r}_0\} = \text{span}\{H\mathbf{r}_0, H\mathbf{r}_1, \dots, H\mathbf{r}_{m-1}\}.$$

If instead of just one, a whole set of preconditioners $\{H_s\}_{s=1,\dots,N}$ is available, a multipreconditioned Krylov subspace method would change that to a space N times larger:

$$\text{span}\{H_1\mathbf{r}_0, \dots, H_N\mathbf{r}_0, H_1\mathbf{r}_1, \dots, H_N\mathbf{r}_1, \dots, H_1\mathbf{r}_{m-1}, \dots, H_N\mathbf{r}_{m-1}\}.$$

The intuition is that an $m \times N$ -dimensional minimization space will give a much better approximation than an m -dimensional space.

- Although multipreconditioning can significantly reduce the number of iterations needed to achieve convergence, each iteration is more expensive because the minimization problem is N times larger. Since Krylov subspace methods typically construct an orthonormal basis of the minimization space, it is the orthogonalization that can become prohibitively expensive. My first contribution to the field of multipreconditioning is the Adaptive Multipreconditioned CG algorithm (AMPCG). It is presented in Section 3.2 adapted from [5]. At each iteration, a computationally cheap test is performed in order to determine whether multipreconditioning is a good idea. Based on the result of this test (which is justified by the analysis), the

algorithm mixes more expensive (and more efficient) multipreconditioned iterations together with classically preconditioned iteration for the aggregated preconditioner $\sum_{s=1}^N H_s$.

- In the original AMPCG work, the algorithm is applied to a substructuring DD method called BDD. Since this pointed toward the efficiency of AMPCG, the algorithm was then applied to industrial test cases. This work, in collaboration with Christophe Bovet, Pierre Gosselet and Augustin Parret Fréaud is presented in Section 3.2.5 (adapted from [2]). Scalability and robustness of AMPCG are validated with the adaptive multipreconditioned version of the FETI [75] DD method.
- A natural next question is whether adaptive multipreconditioning can be applied to a wider range of problems and whether it can be applied algebraically (in the same sense as above). The first extension, presented in Section 3.3 and adapted from [11], is for non-symmetric preconditioners, and more particularly the Restricted Additive Schwarz (RAS) DD preconditioner. When the problem matrix is symmetric, even if the preconditioner is not, the same iterations as multipreconditioned CG can be run. However, the idea that led to the adaptivity step no longer goes through in this setting. The main issue is that for algebraic preconditioners such as RAS, it is small eigenvalues that hamper convergence (as opposed to BDD and FETI that are slowed down by isolated large values). Two alternative ways of making multipreconditioned RAS adaptive are presented and tested.
- Finally, Section 3.4.1, adapted from [1], addresses non-symmetric linear systems. It is advocated that multipreconditioning should be applied either to Orthomin (GCR) or to biCG. The algorithms are defined, their constitutive properties are proved and a test problem is solved. This work opens up more perspectives for the future of adaptive multipreconditioning and non-symmetric linear solvers.

Chapter 4 Non-Hermitian linear systems solved by W -GMRES Non-Hermitian Krylov subspace solvers are notoriously difficult to analyze and to accelerate. The convergence of GMRES has been studied extensively but, in the general case, it is still unclear how to predict what a good choice of preconditioner is. The analysis in the introduction, which considered a diagonalization of A , is helpful only for normal matrices. In all other cases, although eigenvalues play a role in convergence they do not govern it alone. A striking illustration of this is the well-known result from [86] where the authors prove that prescribing a convergence curve for GMRES does not impose any condition at all on the eigenvalues of the problem matrix.

- My first contribution to GMRES is presented in Section 4.2 (adapted from [7]). GMRES is considered with two accelerators: preconditioning and weighting. The objective is to derive a convergence bound in which the role of each is apparent so that they can be chosen wisely. A new convergence bound is proved. It has particularly interesting implications when the preconditioner is hpd, the weight matrix is chosen to be the preconditioner, and the problem matrix is positive definite. Indeed, the convergence bound is then dependent only on whether the Hermitian part is well preconditioned and on whether the problem matrix is *strongly* or *mildly* non-Hermitian. Both the analysis and numerical experiments on a convection-reaction-diffusion problem point towards scalability of the method and h -independence if these properties are satisfied for the preconditioned Hermitian part. And this can in turn be achieved, *e.g.*, by the methods from Chapter 2.
- The drawback of the previous work is that if the problem is strongly non-Hermitian convergence slows down. In collaboration with Daniel Szyld, we have proposed in [8] to improve this by deflation. Our work, the main results of which are described in Section 4.3, starts

by adding deflation to the analysis above. In the same special case (hpd preconditioning, preconditioner as weight matrix and positive definite problem matrix), a new deflation space is defined. It consists of low-frequency eigenvectors of the eigenvalue problem $N\mathbf{x} = \lambda M\mathbf{x}$ where $A = M + N$ is the splitting of the problem matrix A into its Hermitian and non-Hermitian parts. The threshold for selecting low-frequency eigenvalues enters into the new bound. Hence the algorithm can be made more efficient by adding more vectors to the deflation space.

In the interest of making the manuscript insightful but not too long, many results are stated along with the ideas that lead to them but without full proofs. Numerical results are insightful to understand the methods as well as their strengths and weaknesses. For this reason, there is an overrepresentation of numerical results in the manuscript compared to the usual proportion that they occupy in my work.

Chapter 2

Spectral coarse spaces for symmetric positive definite problems

2.1 Introduction

The problem considered is still the solution of linear systems of the form

$$A\mathbf{x} = \mathbf{b}, \text{ where } A \in \mathbb{R}^{n \times n} \text{ is symmetric positive and definite (spd).}^1$$

The purpose of this chapter is to propose, analyse and test new coarse spaces in domain decomposition. For readers unfamiliar with the concept, coarse spaces can in a first approach be viewed as the domain decomposition (DD) version of a deflation space (although deflation is not the only form of two-level DD method – more on this later on). The choice of a coarse space can have a huge, positive, effect on the convergence of the method. It is with the introduction of coarse spaces that domain decomposition methods became scalable. Indeed, the first coarse spaces already ensured that convergence could be bounded independently of the number of subdomains and with only a weak dependency on the number of elements in each subdomain (see *e.g.*, [74, 118]). A consensus seems to have occurred that it is worth enlarging, even quite significantly, the coarse space if this enlargement allows to achieve robustness and scalability. One popular way of doing this is to compute the coarse space by solving generalized eigenvalue problems in the subdomains. These generalized eigenvalue problems are chosen to seek out the vectors that make convergence slow. A first group of methods was tailored to the scalar elliptic problem with a varying conductivity in the Additive Schwarz² framework. Among these are the two articles [78, 77] on one hand, and [135, 17, 16] on the other. The same two groups of authors contributed, with collaborators, to the set of articles [64] and [20, 18]. This time, the methods apply to a much wider range of PDEs that include the linear elasticity equations. The method in [20, 18], to which I contributed during my PhD, is called GenEO for Generalized eigenvalues in the overlaps.

A different version of the GenEO coarse space was proposed for FETI³ and BDD⁴ in [19]. The problems there are reduced to the interfaces between subdomains but we kept the name GenEO since these interfaces in some sense constitute an overlap between subdomains. The family of GenEO coarse

¹Although the chapter is for real-valued linear systems, all the developments extend to complex hpd matrices.

²Readers who are not yet fluent in DD need not worry. The purpose here is to convey, with details for the specialists, that spectral coarse spaces apply in many settings. In the remainder of the manuscript, when DD methods are applied they are introduced.

³See footnote 2 above.

⁴See footnote 2 above.

spaces has grown since, with *e.g.*, the contributions [90] for Optimized Schwarz⁵ and [129] in the context of boundary element methods.

The idea of solving generalized eigenvalue problems to design coarse spaces with guaranteed good convergence had in fact already been proposed, unknowingly to the authors previously mentioned. Indeed, the pioneering work [128] proposes such a technique for FETI-DP⁶ and BDDC⁷. The authors make use of a ‘Local Indicator of the Condition Number Bound’ to fill a gap in what would be an otherwise complete proof of a condition number bound. The follow-up article [154] illustrates the efficiency of the method for BDDC in a multilevel framework, and [115] (by different authors) makes the proof complete in two dimensions. It must also be noted that, as early as 1999, the authors of [41] proposed a multigrid smoothed aggregation algorithm with an enrichment technique that includes low frequency eigenmodes of the operator in the aggregate (which is like a subdomain). Thanks to this procedure, any convergence rate chosen *a priori* can be achieved. Spectral enrichment is also at the heart of the spectral algebraic multigrid method [56]. The field of coarse spaces based on generalized eigenproblems in subdomains has been so active that it is not realistic to list all contributions here. A very incomplete overview is [111, 51, 140, 79, 93, 94, 140, 112, 170, 109, 110]. An innovative approach is the combination of two-level domain decomposition and machine learning techniques as proposed by [95, 96, 113, 114].

During my PhD, I contributed to the DtN coarse space in [17, 16] for the scalar elliptic problem as well as the GenEO coarse space [20, 18, 19] for Additive Schwarz, BDD and FETI. In the remainder of this chapter, I present work done since.

- Section 2.2 proposes a unified framework for spectral coarse spaces in the abstract Schwarz framework. This is a generalization of GenEO to all methods that can be written in the abstract Schwarz framework. It also serves the construction of the method presented in the next section. [Adapted from [6]]
- In Section 2.3 an algebraic spectral coarse space is proposed: the AWG (Algebraic Woodbury GenEO) coarse space. Algebraic solvers have the particularity that they require only the knowledge of A and $\mathbf{b}$ so they are fully black box. [Adapted from [12] and, [4] in collaboration with Loïc Gouarin]
- Section 2.4 presents the open-source code which we have developed in collaboration with Loïc Gouarin and some numerical results. [<https://github.com/hpc-maths/GenEO> – [13]]

2.2 GenEO coarse spaces in the Abstract Schwarz framework [6]

The purpose of this section is to provide unified definitions and analysis for two-level domain decomposition methods with spectral coarse spaces. This is done in the *abstract Schwarz framework* by which we refer to the formalism presented in Chapters 2 and 3 of the book by Toselli and Widlund [159]. This framework provides both a way of defining two-level domain decomposition preconditioners and to analyze condition number bounds that involve them. It is fair to mention that the article [21] proposes a setting with the same objective. One difference is that, here, all abstract results are proved from scratch to fit exactly the framework that is considered.

⁵Readers who are not yet fluent in DD need not worry. The purpose here is to convey, with details for the specialists, that spectral coarse spaces apply in many settings. In the remainder of the manuscript, when DD methods are applied they are introduced.

⁶See footnote 5 above.

⁷See footnote 5 above.

The outline of the section is the following. In Section 2.2.1, the two-level Abstract Schwarz framework is presented. Section 2.2.2 considers spaces spanned by eigenvectors and their properties. The definitions of the GenEO coarse spaces and the convergence results are stated in Section 2.2.3 followed by some elements of the proofs (Section 2.2.4). This includes generalized versions of the technical lemmas from [159] (stable splitting and stability of local solver). As an illustration, Section 2.2.5 considers a two-dimensional linear elasticity problem and presents exactly how the abstract framework applies to the Additive Schwarz, Neumann-Neumann, and inexact Schwarz preconditioners with numerical results.

2.2.1 Two-level DD in the Abstract Schwarz framework

We start with the components of the one-level abstract Schwarz preconditioner.

Assumption 2.2.1 (Local Setting). *Let $n \in \mathbb{N}$ denote the order of the problem matrix A . Let $N \in \mathbb{N}$ denote the chosen number of subdomains and $n^s \in \mathbb{N}$ ($s \in \llbracket 1, N \rrbracket$) denote the cardinality of each one. It is assumed that restriction operators $R^s \in \mathbb{R}^{n^s \times n}$ and local operators $\tilde{A}^s \in \mathbb{R}^{n^s \times n^s}$ have been defined for each $s \in \llbracket 1, N \rrbracket$ in such a way that:*

- $R^s R^{s\top} = I$ (i.e., the rows of R^s form an orthonormal basis of $\text{range}(R^{s\top})$),
- $\mathbb{R}^n = \sum_{s=1}^N \text{range}(R^{s\top})$ (i.e., the set of local subspaces forms a cover of the global space),
- each $\tilde{A}^s$ is an spsd matrix,
- with $\tilde{A}^{s\top}$ denoting the pseudo-inverses of the matrices $\tilde{A}^s$, the one-level preconditioner $(\sum_{s=1}^N R^{s\top} \tilde{A}^{s\top} R^s)$ (later denoted by H – see Definition 2.2.6) is non-singular. Hence it is symmetric positive definite.

The coloring constant, whose definition is recalled next following [159][Section 2.5.1], plays an important role in the numerical performance and the theory of domain decomposition methods. For a given A , it depends only on the choice of local subspaces.

Definition 2.2.2 (Coloring constant). *Let $\mathcal{N} \in \mathbb{N}$ be such that there exists a set $\{\mathcal{C}_j; 1 \leq j \leq \mathcal{N}\}$ of subsets of $\llbracket 1, N \rrbracket$ satisfying*

$$\llbracket 1, N \rrbracket = \bigcup_{1 \leq j \leq \mathcal{N}} \mathcal{C}_j \text{ and } \forall j \in \llbracket 1, \mathcal{N} \rrbracket, \forall s, t \in \mathcal{C}_j, s \neq t \Rightarrow R^s A R^{t\top} = 0.$$

One can always choose $\mathcal{N} = N$ but in general there are values of $\mathcal{N}$ that are significantly smaller than the number N of subdomains. The number $\mathcal{N}$ is often referred to as the coloring constant since it can be viewed as the number of colors needed to color each subdomain in such a way that any two subdomains with the same color are A -orthogonal.

Assumption 2.2.3 (Splitting of A). *Assume that there exist a family of N spsd matrices $M^s \in \mathbb{R}^{n^s \times n^s}$ (for $s = 1, \dots, N$), and a real number $\mathcal{N}' > 0$ such that any $\mathbf{x} \in \mathbb{R}^n$ can be decomposed as*

$$\mathbf{x} = \sum_{s=1}^N R^{s\top} \mathbf{y}^s, \mathbf{y}^s \in \mathbb{R}^{n^s}, \text{ and } \sum_{s=1}^N \langle \mathbf{y}^s, M^s \mathbf{y}^s \rangle \leq \mathcal{N}' \langle \mathbf{x}, A \mathbf{x} \rangle, \text{ for every } \mathbf{x} \in \mathbb{R}^n. \quad (2.1)$$

The usual way of defining the GenEO coarse spaces is slightly less general and involves a family of matrices that form a partition of unity. Here, the partition of unity matrices are not assumed to be diagonal.

Assumption 2.2.4 (Simplification of Assumption 2.2.3). Let $D_s \in \mathbb{R}^{n^s \times n^s}$ for $s = 1, \dots, N$ be a family of spd matrices that form a partition of unity in the sense that $I = \sum_{s=1}^N R^s \top D_s R^s$. Assume that there exist a set of N spd matrices $M^s \in \mathbb{R}^{n^s \times n^s}$ (for $s = 1, \dots, N$), and a real number $\mathcal{N}' > 0$ such that

$$\sum_{s=1}^N \langle D_s R^s \mathbf{x}, M^s D_s R^s \mathbf{x} \rangle \leq \mathcal{N}' \langle \mathbf{x}, A \mathbf{x} \rangle, \text{ for every } \mathbf{x} \in \mathbb{R}^n. \quad (2.2)$$

Assumption 2.2.4 implies Assumption 2.2.3 by choosing the splitting $\mathbf{x} = \sum_{s=1}^N R^s \top \mathbf{y}^s$ with $\mathbf{y}^s = D_s R^s \mathbf{x}$.

To inject a second level into the preconditioners, a coarse space and a coarse solver must be chosen. The coarse space is the central topic of this work. It is denoted by V^0 and the following assumption is made.

Assumption 2.2.5 (Coarse Setting). Given a coarse space $V^0 \subset \mathbb{R}^n$, a basis for V^0 is stored in the rows of a matrix denoted R^0 :

$$V^0 = \text{range}(R^0 \top); \quad R^0 \in \mathbb{R}^{\dim(V^0) \times n}.$$

A solver must be chosen for the coarse space. Here we focus on the case where the coarse solver is the exact solver on the coarse space: $(R^0 A R^0 \top)^{-1}$.

Definition 2.2.6 (Abstract Schwarz preconditioners). Under Assumption 2.2.1, the one-level abstract Schwarz preconditioner for linear system $A \mathbf{x} = \mathbf{b}$ is defined by:

$$H := \sum_{s=1}^N R^s \top \tilde{A}^s \top R^s.$$

Under Assumptions 2.2.1 and 2.2.5, the two-level hybrid Schwarz preconditioner is defined by

$$H_{\text{hyb}} := \Pi H \Pi \top + R^0 \top (R^0 A R^0 \top)^{-1} R^0,$$

where

$$\Pi := I - R^0 \top (R^0 A R^0 \top)^{-1} R^0 A$$

is the A -orthogonal projection satisfying $\text{Ker}(\Pi) = V^0$.

Under Assumptions 2.2.1 and 2.2.5, the two-level Additive Schwarz preconditioner is defined by

$$H_{\text{ad}} := H + R^0 \top (R^0 A R^0 \top)^{-1} R^0.$$

A simple and well-known result is that there is a connection between spectral bounds for the deflated operator and spectral bounds for the hybrid operator. This is recalled below.

Theorem 2.2.7 (Spectral bounds for $H A \Pi \Rightarrow$ Spectral bounds for $H_{\text{hyb}} A$). Let Assumptions 2.2.1 and 2.2.5 hold. If the eigenvalues of the projected and preconditioned operator satisfy

$$\lambda(H A \Pi) \in \{0\} \cup [\lambda_{\min}, \lambda_{\max}],$$

then the eigenvalues of the operator preconditioned by H_{hyb} from Definition 2.2.6 satisfy

$$\lambda(H_{\text{hyb}} A) \in [\min(1, \lambda_{\min}), \max(1, \lambda_{\max})].$$

Proof. The connection between the spectra of the projected and hybrid/balanced preconditioned operators

is well known (see e.g., [158, 116]) and easy to verify. For any $\mathbf{x} \in \mathbb{R}^n$, it holds that

$$\begin{aligned}\langle \mathbf{x}, AH_{\text{hyb}}A\mathbf{x} \rangle &= \langle \mathbf{x}, A\Pi H\Pi^\top A\mathbf{x} \rangle + \langle \mathbf{x}, AR^{0\top}(R^0AR^{0\top})^{-1}R^0A\mathbf{x} \rangle \\ &= \langle \Pi\mathbf{x}, AH\Pi\mathbf{x} \rangle + \langle (I - \Pi)\mathbf{x}, A(I - \Pi)\mathbf{x} \rangle,\end{aligned}$$

so, with the result for the projected preconditioned operator:

$$\left\{ \begin{array}{l} \langle \mathbf{x}, AH_{\text{hyb}}A\mathbf{x} \rangle \geq \lambda_{\min} \langle \Pi\mathbf{x}, A\Pi\mathbf{x} \rangle + \langle (I - \Pi)\mathbf{x}, A(I - \Pi)\mathbf{x} \rangle \\ \text{and } \langle \mathbf{x}, AH_{\text{hyb}}A\mathbf{x} \rangle \leq \lambda_{\max} \langle \Pi\mathbf{x}, A\Pi\mathbf{x} \rangle + \langle (I - \Pi)\mathbf{x}, A(I - \Pi)\mathbf{x} \rangle \end{array} \right.$$

and the result follows by recalling that Π is an A -orthogonal projection so $\langle \mathbf{x}, A\mathbf{x} \rangle = \langle \Pi\mathbf{x}, A\Pi\mathbf{x} \rangle + \langle (I - \Pi)\mathbf{x}, A(I - \Pi)\mathbf{x} \rangle$. $\square$

2.2.2 Spaces spanned by eigenvectors

The GenEO coarse spaces are defined by choosing a basis for the coarse space that consists of high or low frequency eigenvectors of certain generalized eigenvalue problems. We introduce notation to describe these spaces in a compact manner.

Definition 2.2.8 ($\mathcal{Y}_L(\tau, M_A, M_B)$ and $\mathcal{Y}_H(\tau, M_A, M_B)$). *Let M_A be an spsd matrix, let M_B be an spd matrix, and let $\tau > 0$ be a scalar. We define $\mathcal{Y}_L(\tau, M_A, M_B)$ and $\mathcal{Y}_H(\tau, M_A, M_B)$ to be the spaces of, respectively, low and high frequency eigenvectors of the generalized eigenvalue problem $M_A\mathbf{y} = \lambda M_B\mathbf{y}$:*

$$\mathcal{Y}_L(\tau, M_A, M_B) := \text{span}\{\mathbf{y}; M_A\mathbf{y} = \lambda M_B\mathbf{y} \text{ with } \lambda < \tau\},$$

and

$$\mathcal{Y}_H(\tau, M_A, M_B) := \text{span}\{\mathbf{y}; M_A\mathbf{y} = \lambda M_B\mathbf{y} \text{ with } \lambda \geq \tau\}.$$

Subsets that are spanned by eigenvectors of well-chosen generalized eigenvalue problems have very useful properties. Some of these are recalled below. They show how projecting onto a subspace spanned by eigenvectors implies bounds for well chosen Rayleigh quotients. These in turn provide bounds for eigenvalues which inform us about convergence.

Lemma 2.2.9. *Let M_A be an spsd matrix, let M_B be an spd matrix, and let $\tau > 0$. With the notation from Definition 2.2.8, the two following properties hold*

- *spectral estimates:*

$$\left\{ \begin{array}{l} |y|_{M_A}^2 < \tau \|y\|_{M_B}^2 \text{ for any } \mathbf{y} \in \mathcal{Y}_L(\tau, M_A, M_B), \\ |y|_{M_A}^2 \geq \tau \|y\|_{M_B}^2 \text{ for any } \mathbf{y} \in \mathcal{Y}_H(\tau, M_A, M_B), \end{array} \right. \quad (2.3)$$

- *conjugacy:*

$$\left\{ \begin{array}{l} (\mathcal{Y}_L(\tau, M_A, M_B))^\perp = M_B \mathcal{Y}_H(\tau, M_A, M_B) \\ (\mathcal{Y}_H(\tau, M_A, M_B))^\perp = M_B \mathcal{Y}_L(\tau, M_A, M_B), \end{array} \right. \quad (2.4)$$

in which the orthogonality is with respect to the Euclidean inner product.

Proof. By assumption M_B is spd, so the generalized eigenvalue problem $M_A\mathbf{y} = \lambda M_B\mathbf{y}$ from Definition 2.2.8 is equivalent to the classical eigenvalue problem

$$M_B^{-1/2} M_A M_B^{-1/2} \mathbf{z} = \lambda \mathbf{z}, \text{ with } \mathbf{z} = M_B^{1/2} \mathbf{y}.$$

By the spectral theorem [101, Theorem 4.1.5] applied to the spsd matrix $M_B^{-1/2} M_A M_B^{-1/2}$, this matrix is unitarily diagonalizable so there exists an orthonormal basis of $\mathbb{R}^n$ formed of eigenvectors $\mathbf{z}_k$. Conse-

quently, the vectors $\mathbf{y}_k = M_B^{-1/2} \mathbf{z}_k$ form an M_B -orthonormal basis of $\mathbb{R}^n$ and everything else follows by analogy with the classical eigenvalue problem. $\square$

A particular role is played by the kernel of the local operators $\tilde{A}^s$ and the notation below is necessary to study their role in convergence.

Definition 2.2.10 (Constant $C_\#$). *Let $C_\# > 0$ be such that, for every $s = 1, \dots, N$ and every $\mathbf{x}^s \in \mathbb{R}^{n^s}$,*

$$\|\tilde{I}^s R^{s\top} \mathbf{x}^s\|_A^2 \leq C_\#^{-1} |\mathbf{x}^s|_{\tilde{A}^s}^2,$$

where $\tilde{I}^s$ is the A -orthogonal projection characterized⁸ by $\text{Ker}(\tilde{I}^s) = R^{s\top} \text{Ker}(\tilde{A}^s)$. (If $\tilde{A}^s$ is non-singular then $\tilde{I}^s$ is the identity matrix.)

The definition of $C_\#$ is not very natural. The following result motivates its existence. The proof is omitted.

Lemma 2.2.11 (Natural coarse space as a spectral coarse space). *Let Assumptions 2.2.1 hold, then*

$$\mathcal{Y}_L(C_\#, \tilde{A}^s, R^s A R^{s\top}) = \text{Ker}(\tilde{A}^s), \text{ for any } s \in \llbracket 1, N \rrbracket$$

where $C_\# > 0$ comes from Definition 2.2.10.

2.2.3 GenEO coarse spaces and convergence results

Recall that for any matrix B , the notation $\lambda(B)$ refers to any of its eigenvalues. In the next two theorems, GenEO coarse spaces are defined and bounds for the eigenvalues of the preconditioned operators are provided. These do not depend on the number of subdomains or on any parameters in A . The reason eigenvalue bounds are so important is that they imply a convergence bound for PCG and PPCG (recalled in (1.13), see also [159, Lemma C.10], [76], and the proof in Section 3.2.1). Table 2.1 also summarizes almost all of the results in a very compact form.

Theorem 2.2.12 (Spectral results for the projected and hybrid preconditioners). *For all results in the theorem, it is assumed that Assumptions 2.2.1 and 2.2.5 hold. Let $\tau > 0$ be a user-chosen threshold. (Recall Definition 2.2.2 for the coloring constant $\mathcal{N}$ and Definition 2.2.8 for the spaces $\mathcal{Y}_L$ and $\mathcal{Y}_H$ of low and high frequency eigenvectors of a certain generalized eigenvalue problem.)*

1. (Natural coarse space) With $C_\#$ from Definition 2.2.10, if for any $s \in \llbracket 1, N \rrbracket$, $(R^{s\top} \text{Ker}(\tilde{A}^s)) \subset V^0$ then

$$\lambda(HA\Pi) \leq \mathcal{N} C_\#^{-1} \text{ and } \lambda(H_{\text{hyb}}A) \leq \max(1, \mathcal{N} C_\#^{-1}). \quad (2.5)$$

2. (GenEO for $\lambda_{\max}$) If for any $s \in \llbracket 1, N \rrbracket$, $(R^{s\top} \mathcal{Y}_L(\tau, \tilde{A}^s, R^s A R^{s\top})) \subset V^0$ then

$$\lambda(HA\Pi) \leq \mathcal{N} \tau^{-1} \text{ and } \lambda(H_{\text{hyb}}A) \leq \max(1, \mathcal{N} \tau^{-1}). \quad (2.6)$$

3. (GenEO for $\lambda_{\min}$) Under Assumption 2.2.3 (or 2.2.4), if for any $s \in \llbracket 1, N \rrbracket$,

- (a) either $\tilde{A}^s$ is non-singular, and $(R^{s\top} \mathcal{Y}_L(\tau^{-1}, M^s, \tilde{A}^s)) \subset V^0$,
- (b) or M^s is non-singular and $(R^{s\top} [\text{Ker}(\tilde{A}^s) + \mathcal{Y}_H(\tau, \tilde{A}^s, M^s)]) \subset V^0$,

⁸If the columns in $\tilde{Z}^s$ form a basis for $R^{s\top} \text{Ker}(\tilde{A}^s)$ then $\Pi^s := I - \tilde{Z}^s (\tilde{Z}^{s\top} A \tilde{Z}^s)^{-1} \tilde{Z}^{s\top} A$

All the result below are under Assumptions 2.2.1 and 2.2.5.

Local contributions to GenEO coarse space	Requirement		Spectral bounds for two-level preconditioners	Ref.
	Ass. 2.2.3 (or 2.2.4)	$\tilde{A}^s$ spd		
$\text{Ker}(\tilde{A}^s)$			$\lambda(HA\Pi) \leq \frac{\mathcal{N}}{C_\#}$ $\lambda(H_{\text{hyb}}A) \leq \max\left(1, \frac{\mathcal{N}}{C_\#}\right)$	(2.5)
$\emptyset$		✓	$\lambda(H_{\text{ad}}A) \leq \frac{\mathcal{N}}{C_\#} + 1$	(2.8)
$\mathcal{Y}_L(\tau, \tilde{A}^s, R^s A R^{s\top})$ i.e., low frequency eigenvectors of $\tilde{A}^s \mathbf{x}^s = \lambda^s R^s A R^{s\top} \mathbf{x}^s$			$\lambda(HA\Pi) \leq \frac{\mathcal{N}}{\tau}$ $\lambda(H_{\text{hyb}}A) \leq \max\left(1, \frac{\mathcal{N}}{\tau}\right)$	(2.6)
$\mathcal{Y}_L(\tau^{-1}, M^s, \tilde{A}^s)$ i.e., low frequency eigenvectors of $M^s \mathbf{x}^s = \lambda^s \tilde{A}^s \mathbf{x}^s$	✓	✓	$\lambda(HA\Pi) \geq \frac{1}{\tau \mathcal{N}'}$ $\lambda(H_{\text{hyb}}A) \geq \min\left(1, \frac{1}{\tau \mathcal{N}'}\right)$ $\lambda(H_{\text{ad}}A) \geq \left[\max\left(2, 1 + 2\frac{\mathcal{N}}{C_\#}\right) \max(1, \mathcal{N}'\tau)\right]^{-1}$	(2.7) and (2.9)
$\text{Ker}(\tilde{A}^s) + \mathcal{Y}_H(\tau, \tilde{A}^s, M^s)$ i.e., high frequency eigenvectors of $\tilde{A}^s \mathbf{x}^s = \lambda^s M^s \mathbf{x}^s$	✓ and M^s spd		$\lambda(HA\Pi) \geq \frac{1}{\tau \mathcal{N}'}$ $\lambda(H_{\text{hyb}}A) \geq \min\left(1, \frac{1}{\tau \mathcal{N}'}\right)$	(2.7)
$\text{Ker}(\tilde{A}^s) + \text{Ker}(M^s) +$ $W^s \mathcal{Y}_H(\tau, W^{s\top} \tilde{A}^s W^s, W^{s\top} M^s W^s)$ i.e., high frequency eigenvectors of $W^{s\top} \tilde{A}^s W^s \mathbf{x}^s = \lambda^s W^{s\top} M^s W^s \mathbf{x}^s$	✓		Same two bounds as in the cell directly above	(2.7)
$\mathcal{Y}_L(v, \tilde{A}^s, R^s A R^{s\top}) + \mathcal{Y}_L(\tau^{-1}, M^s, \tilde{A}^s)$ i.e., low frequency eigenvectors of $\tilde{A}^s \mathbf{x}^s = \lambda^s R^s A R^{s\top} \mathbf{x}^s$ and of $M^s \mathbf{x}^s = \lambda^s \tilde{A}^s \mathbf{x}^s$	✓	✓	$\lambda(H_{\text{ad}}A) \geq \left[\max\left(2, 1 + 2\frac{\mathcal{N}}{v}\right) \max(1, \mathcal{N}'\tau)\right]^{-1}$	(2.10)

Summary of Notation:

- A : problem matrix in the linear system $A\mathbf{x} = \mathbf{b}$,
- V^0 : coarse space
- Π : coarse projector (Definition 2.2.6),
- $\tilde{A}^s$: local solvers in the definitions of the preconditioners $H, H_{\text{hyb}}, H_{\text{ad}}$ (Definition 2.2.6),
- $\mathcal{N}$: coloring constant (Definition 2.2.2),
- $C_\#$: constant that depends on the kernel of $\tilde{A}^s$ (Definition 2.2.10),
- $R^s A R^{s\top}$: restriction of the problem to subdomain number s where R^s is the restriction to the subdomain (see Assumption 2.2.1),
- $\tau, v > 0$: thresholds chosen by the user,
- $\mathcal{Y}_L$ and $\mathcal{Y}_H$: low and high frequency eigenvectors of a certain generalized eigenvalue problem (Definition 2.2.8),
- $\mathcal{N}'$ and M^s : from Assumption 2.2.3 (or 2.2.4),
- $W^s \in \mathbb{R}^{n \times \text{rank}(M^s)}$ is a matrix whose columns form an L -orthonormal basis of $\text{range}(M^s)$.

Table 2.1: Summary of most of the results from Theorems 2.2.12 and 2.2.13. (Not included are the variants (labelled (b) and (c)) of (2.10)). For the projected operator $HA\Pi$, the lower bounds are to be understood for any non-zero eigenvalue.

- (c) or $W^s \in \mathbb{R}^{n^s \times \text{rank}(M^s)}$ is a matrix whose columns form an I-orthonormal basis of $\text{range}(M^s)$ and $\left(R^{s\top} \left[\text{Ker}(\tilde{A}^s) + \text{Ker}(M^s) + W^s \mathcal{Y}_H(\tau, W^{s\top} \tilde{A}^s W^s, W^{s\top} M^s W^s) \right] \right) \subset V^0$

then, excluding the zero eigenvalues of $H\Lambda\Pi$,

$$\tau \mathcal{N}'^{-1} \leq \lambda(H\Lambda\Pi) \text{ and } \min(1, (\tau \mathcal{N}')^{-1}) \leq \lambda(H_{\text{hyb}}A), \quad (2.7)$$

(with $\mathcal{N}'$ from Assumption 2.2.3 (or 2.2.4)).

Proof. By Lemma 2.2.11, (2.5) is a consequence of (2.6) with $\tau = C_{\#}$. We also notice that in the Assumptions for (2.7), Case 3b is included in case 3c (setting $W^s = \mathbb{I}$). It remains to prove (2.6) and (2.7) without the assumption labelled 3b. This proof is a core result of [6]. The results for $H_{\text{hyb}}A$ are deduced from the results for $H\Lambda\Pi$ by applying Theorem 2.2.7. In the end, the proof comes down to showing (2.6) and (2.7) for $H\Lambda\Pi$. This is addressed in Section 2.2.4. $\square$

The presence of W^s in the last case is to ensure that the matrix on the right hand side of the generalized eigenvalue problem is spd. Although the notation is slightly awkward, it is standard in practice for eigensolvers to apply special treatment to null spaces.

Theorem 2.2.13 (Spectral results for the additive preconditioner). *For all results in the theorem, it is assumed that Assumptions 2.2.1 and 2.2.5 hold. It is also assumed that all local operators $\tilde{A}^s$ are non-singular (for every $s \in \llbracket 1, N \rrbracket$). Finally, let $\tau, v > 0$ be two user-chosen thresholds. (Recall Definition 2.2.2 for the coloring constant $\mathcal{N}$ and Definition 2.2.8 for the spaces $\mathcal{Y}_L$ and $\mathcal{Y}_H$ of low and high frequency eigenvectors of a certain generalized eigenvalue problem.)*

1. (Natural bound for $\lambda_{\max}$) With $C_{\#}$ from Definition 2.2.10, it holds that

$$\lambda(H_{\text{ad}}A) \leq \mathcal{N}C_{\#}^{-1} + 1. \quad (2.8)$$

2. (GenEO for $\lambda_{\min}$) Under Assumption 2.2.3 (or 2.2.4), with $C_{\#}$ from Definition 2.2.10, if for any $s \in \llbracket 1, N \rrbracket$,

- (a) either $\left(R^{s\top} \mathcal{Y}_L(\tau^{-1}, M^s, \tilde{A}^s)\right) \subset V^0$,
- (b) or M^s is non-singular, and $\left(R^{s\top} \mathcal{Y}_H(\tau, \tilde{A}^s, M^s)\right) \subset V^0$,
- (c) or $W^s \in \mathbb{R}^{n^s \times \text{rank}(M^s)}$ is a matrix whose columns form an I-orthonormal basis of $\text{range}(M^s)$ and $\left(R^{s\top} \left[\text{Ker}(M^s) + W^s \mathcal{Y}_H(\tau, W^{s\top} \tilde{A}^s W^s, W^{s\top} M^s W^s) \right] \right) \subset V^0$

then

$$\left[\max(2, 1 + 2\mathcal{N}C_{\#}^{-1}) \max(1, \mathcal{N}'\tau) \right]^{-1} \leq \lambda(H_{\text{ad}}A), \quad (2.9)$$

(with $\mathcal{N}'$ from Assumption 2.2.3 (or 2.2.4)).

3. (Double GenEO for $\lambda_{\min}$) Under Assumption 2.2.3 (or 2.2.4), if for any $s \in \llbracket 1, N \rrbracket$, $R^{s\top} \mathcal{Y}_L(v, \tilde{A}^s, R^s A R^{s\top}) \subset V_0$ and

- (a) either $\left(R^{s\top} \mathcal{Y}_L(\tau^{-1}, M^s, \tilde{A}^s)\right) \subset V^0$,
- (b) or M^s is non-singular, and $\left(R^{s\top} \mathcal{Y}_H(\tau, \tilde{A}^s, M^s)\right) \subset V^0$,
- (c) or $W^s \in \mathbb{R}^{n^s \times \text{rank}(M^s)}$ is a matrix whose columns form an I-orthonormal basis of $\text{range}(M^s)$ and $\left(R^{s\top} \left[\text{Ker}(M^s) + W^s \mathcal{Y}_H(\tau, W^{s\top} \tilde{A}^s W^s, W^{s\top} M^s W^s) \right] \right) \subset V^0$

then

$$\left[\max(2, 1 + 2\mathcal{N}v^{-1}) \max(1, \mathcal{N}'\tau) \right]^{-1} \leq \lambda(H_{\text{ad}}A), \quad (2.10)$$

(with $\mathcal{N}'$ from Assumption 2.2.3 (or 2.2.4)).

Proof. It has been assumed that the $\tilde{A}^s$ are non-singular so equation (2.5) in Theorem 2.2.12 applies to the one-level preconditioner to give $\lambda(HA) \leq \mathcal{N}C_{\#}^{-1}$. Moreover, $H_{\text{ad}}A - HA = R^{0\top}(R^0AR^{0\top})^{-1}R^0A$ which is a projection. Projections have eigenvalues in $\{0, 1\}$ so (2.8) holds with no restriction on the coarse space.

By Lemma 2.2.11, (2.9) is a consequence of (2.10) with $v = C_{\#}$. We also notice that in the Assumptions for (2.10), Case 3b is included in case 3c (setting $W^s = I$). It remains to prove (2.10) without the assumption labelled 3b. This follows from the result in Theorem 2.2.14 (see below) with the bound for the hybrid operator ($\min(1, (\tau\mathcal{N}')^{-1}) \leq \lambda(H_{\text{hyb}}A)$) coming from (2.7) in Theorem 2.2.12. $\square$

2.2.4 Some elements of the proofs

Additive preconditioner. The following theorem is my best shot at providing a bound for the smallest eigenvalue of the Additive preconditioned operator with respect to either the deflated operator or the hybrid operator. There is an additional assumption which is that for some $v > 0$, the spaces $\mathcal{Y}_L(v, \tilde{A}^s, R^sAR^{s\top})$ contribute to the coarse space. This result is not widely known and may be new. Usually, the two-level additive preconditioner is applied with exact local solves $\tilde{A}^s = R^sAR^{s\top}$, in which case $\mathcal{Y}_L(1, \tilde{A}^s, R^sAR^{s\top}) = \emptyset$. The proof of the theorem is rather technical and not included. It can be found in [6, Section 3.4]. The result is important because it allows us to derive the proof of Theorem 2.2.13 (spectral bound for H_{ad}) from Theorem 2.2.12 (spectral bounds for H_{hyb} and $H\Lambda\Pi$).

Theorem 2.2.14 (Lower bound for $\lambda(H_{\text{hyb}}A) \Rightarrow$ Lower bound for $\lambda(H_{\text{ad}}A)$). *Let Assumptions 2.2.1, 2.2.5, and 2.2.3 (or 2.2.4) hold. Moreover, assume that the matrices $\tilde{A}^s$ are non-singular. If for any $s \in \llbracket 1, N \rrbracket$, $(R^{s\top}\mathcal{Y}_L(v, \tilde{A}^s, R^sAR^{s\top})) \subset V^0$ and $0 < \lambda_{\min, \text{hyb}} \leq \lambda(H_{\text{hyb}}A)$ is a lower bound for the eigenvalues of the hybrid preconditioned operator then*

$$\lambda_{\min, \text{hyb}} \left[\max(2, 1 + 2\mathcal{N}v^{-1}) \right]^{-1} \leq \lambda(H_{\text{ad}}A).$$

Projected preconditioner. In the proof of Theorem 2.2.12, it remains to prove (2.6) and (2.7) for $H\Lambda\Pi$. The full proofs can be found in [6, Sections 3.1, 3.2 and 3.3]). The gist of the proofs is that they combine two arguments

- Generalizations of the usual lemmas that are used to analyze the convergence of DD in the abstract framework. These are the properties often referred to as the stability of the local solver (generalized in Lemma 2.2.15) and the existence of a stable splitting (generalized in Lemma 2.2.16). They are stated below.
- Properties of spaces formed by low- or high-frequency eigenvectors of a generalized eigenvalue problem (recalled in Lemma 2.2.9).

Below are the generalizations of two fundamental results in the abstract Schwarz framework. First, the abstract result that bounds the spectrum from above is given. A difference with [159][Assumption 2.4] is that the result must be proved for vectors in $\text{range}(\tilde{A}^{s\top}R^s\Pi^\top)$, instead of $\text{range}(\tilde{A}^{s\top}R^s)$. This subtlety is what allows us to choose the coarse space. Another difference, is the presence of a projection operator $\tilde{\Pi}^s$ in the assumption of the lemma. This weakens the assumption as long as the kernel of $\tilde{A}^s$ (once extended to the global space) is in the coarse space.

Lemma 2.2.15 (Upper bound for $\lambda(HA\Pi)$). *Assume that the kernels of the local solvers $\tilde{A}^s$ contribute to the coarse space in the sense that $\sum_{s=1}^N R^s{}^\top \text{Ker}(\tilde{A}^s) \subset V^0$. For each $s = 1, \dots, N$, let $\tilde{\Pi}^s$ be the A -orthogonal projection characterized by $\text{Ker}(\tilde{\Pi}^s) = R^s{}^\top \text{Ker}(\tilde{A}^s)$. Assume that there exists $\omega > 0$ such that*

$$\|\tilde{\Pi}^s R^s{}^\top \mathbf{x}^s\|_A^2 \leq \omega \|\mathbf{x}^s\|_{A^s}^2 \text{ for every } s = 1, \dots, N \text{ and every } \mathbf{x}^s \in \text{range}(\tilde{A}^s{}^\dagger R^s \Pi^\top).$$

Then the eigenvalues of $HA\Pi$ satisfy $\lambda(HA\Pi) \leq \mathcal{N}\omega$, where $\mathcal{N}$ is as in Definition 2.2.2.

The abstract Schwarz theory ([159][Theorem 2.13 under Assumption 2.12]) also provides a result for bounding the spectrum of the two-level operator from below. The result proved in the next lemma is similar up to the differences pointed out below the lemma.

Lemma 2.2.16 (Lower bound for $\lambda(HA\Pi)$). *Assume that the kernels of the local solvers contribute to the coarse space in the sense that $\sum_{s=1}^N R^s{}^\top \text{Ker}(\tilde{A}^s) \subset V^0$. If, for any $\mathbf{x} \in \text{range}(\Pi)$, there exist $\mathbf{z}^1, \dots, \mathbf{z}^n$ such that*

$$\mathbf{x} = \sum_{s=1}^N \Pi R^s{}^\top \mathbf{z}^s \text{ and } \sum_{s=1}^N \langle \mathbf{z}^s, \tilde{A}^s \mathbf{z}^s \rangle \leq C_0^2 \langle \mathbf{x}, A\mathbf{x} \rangle \quad (\text{stable splitting of } \mathbf{x}),$$

then, the eigenvalues of $HA\Pi$, excluding zero, satisfy $\lambda(HA\Pi) \geq C_0^{-2}$.

The differences compared to [159][Theorem 2.13] are the possible singularity of $\tilde{A}^s$, the extra presence of Π in the definition of a splitting, and the extra assumption on the minimal coarse space.

2.2.5 Application of the abstract result to three DD preconditioners

In this Section, the abstract framework is made concrete. Its setup, analysis and numerical performance are described for a two-dimensional linear elasticity problem.

Geometry and PDE

Let $\Omega = [0, 2] \times [0, 1] \subset \mathbb{R}^2$ be the computational domain. Let $\partial\Omega_D$ be the left hand side boundary of Ω and let $\mathcal{V} = \{\mathbf{v} \in H^1(\Omega)^2; \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}$. The linear elasticity equations posed in Ω with mixed boundary conditions are considered. A solution $\mathbf{u} \in \mathcal{V}$ is sought such that

$$\int_{\Omega} 2\mu \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) dx + \int_{\Omega} L \text{div}(\mathbf{u}) \text{div}(\mathbf{v}) dx = \int_{\Omega} \mathbf{g} \cdot \mathbf{v} dx, \text{ for all } \mathbf{v} \in \mathcal{V}, \quad (2.11)$$

where, for $i, j = 1, 2$, $\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$, $\mathbf{g} = (0, 1)^\top$ and the Lamé coefficients are functions of Young's modulus E and Poisson's ratio ν : $\mu = \frac{E}{2(1+\nu)}$, $L = \frac{E\nu}{(1+\nu)(1-2\nu)}$. The material parameters are constant $\nu = 0.4$ and varying E following two configurations, 'with layers' and 'without layers', described in Figure 2.1.

Discretization

The computational domain is discretized by a uniform mesh with element size $h = 1/42$ and the boundary value problem is solved numerically with standard piecewise linear ($\mathbb{P}_1$) Lagrange finite elements. Let $\mathcal{V}_h$ be the space of $\mathbb{P}_1$ finite elements that satisfy the Dirichlet boundary condition. Let $\{\phi_k\}_{k=1}^n$ be a basis of $\mathcal{V}_h$. The linear system that is to be solved is

$$\text{Find } \mathbf{x} \in \mathbb{R}^n \text{ such that } A\mathbf{x} = \mathbf{b},$$

with $A_{ij} = \int_{\Omega} [2\mu\epsilon(\phi_i) : \epsilon(\phi_j) + L \operatorname{div}(\phi_i) \operatorname{div}(\phi_j)] dx$ and $\mathbf{b}_i = \int_{\Omega} \mathbf{g} \cdot \phi_i dx$. The dimension of the global problem is $n = 43 \times 84 \times 2 = 7224$ where it has been taken into account that there are two degrees of freedom at each grid point (the $\mathbf{x}$ and $\mathbf{y}$ displacements) and that there are no degrees of freedom where a Dirichlet boundary condition has been prescribed.

Domain Decomposition

The computational domain Ω is partitioned into $N = 8$ non-overlapping subdomains with Metis [108] (see Figure 2.1–left). The geometric subdomains are assumed to be mesh-conforming and they are denoted Ω^s for $s \in \llbracket 1, N \rrbracket$. Let $V^s = \{S_1^s, \dots, S_{n^s}^s\}$ be the set of mesh nodes that are in each Ω^s (for $s \in \llbracket 1, N \rrbracket$). These are also the local degrees of freedom. The restriction matrices $R^s \in \mathbb{R}^{n^s \times n}$ are defined by

$$(R^s)_{ij} = 1 \text{ if } j = S_i^s \text{ and } (R^s)_{ij} = 0 \text{ otherwise.}$$

Each R^s has exactly one 1 per row. These are in agreement with Assumption 2.2.1. By construction the degrees of freedom that are on the interfaces between subdomains are duplicated (and only these ones).

We may also assemble the matrices that correspond to the discretization of the problem (2.11) restricted to each subdomain $s \in \llbracket 1, N \rrbracket$: $A|_{\Omega^s} \in \mathbb{R}^{n^s \times n^s}$ such that

$$(A|_{\Omega^s})_{ij} := \int_{\Omega^s} [2\mu\epsilon(\phi_{S_i^s}) : \epsilon(\phi_{S_j^s}) + L \operatorname{div}(\phi_{S_i^s}) \operatorname{div}(\phi_{S_j^s})] dx \text{ for all } i, j. \quad (2.12)$$

These are frequently referred to as the local Neumann matrices as they arise from assembling the original problem over the subdomain Ω^s with natural boundary conditions. They can't be computed from the global matrix A . Since we consider non-overlapping subdomains they satisfy the very useful property that

$$A = \sum_{s=1}^N R^{s\top} A|_{\Omega^s} R^s.$$

It is chosen to fulfill Assumption 2.2.3 through the use of a partition of unity as proposed in Assumption 2.2.4. Let the partition of unity be

- either the multiplicity scaling (also known as μ -scaling)

$$D_s := R^s \left(\sum_{t=1}^N R^{t\top} R^t \right)^{-1} R^{s\top} \quad \forall s \in \llbracket 1, N \rrbracket, \quad (2.13)$$

- or the stiffness scaling (also known as k -scaling)

$$D_s \in \mathbb{R}^{n^s \times n^s} \text{ diagonal with entries } (D_s)_{ii} := (A|_{\Omega^s})_{ii} / (R^s A R^{s\top})_{ii}, \quad \forall i \in \llbracket 1, n^s \rrbracket. \quad (2.14)$$

Assumption 2.2.3 is fulfilled with $\mathcal{N}' = 1$ by setting

$$M^s := D_s^{-1} A|_{\Omega^s} D_s^{-1}; \text{ and } \mathbf{y}^s := D_s \mathbf{x}, \quad \forall \mathbf{x} \in \mathbb{R}^n. \quad (2.15)$$

These matrices are typically singular and their kernel is the set of rigid body modes on the subdomain weighted by the partition of unity.

Additive Schwarz (AS), Neumann-Neumann (NN) and Inexact Schwarz (IS)

Three DD preconditioners are tested (AS, NN and IS). Their definitions, GenEO coarse spaces and spectral bounds are summarized in Table 2.2.

DD Preconditioner	Local problem $\tilde{A}^s$	Local contributions to GenEO coarse space	Spectral bounds for two-level prec.
AS Additive Schwarz	$R^s A R^{s\top}$	$\mathcal{Y}_L(\tau^{-1}, D_s^{-1} A_{ \Omega^s} D_s^{-1}, R^s A R^{s\top})$ i.e. low frequency eigenvectors of $D_s^{-1} A_{ \Omega^s} D_s^{-1} \mathbf{x}^s = \lambda^s R^s A R^{s\top} \mathbf{x}^s$	$\lambda(HA) \leq \mathcal{N}$ $1/\tau \leq \lambda(HA\Pi) \leq \mathcal{N}$ $1/\tau \leq \lambda(H_{\text{hyb}}A) \leq \mathcal{N}$ $1/((1+2\mathcal{N})\tau) \leq \lambda(H_{\text{ad}}A) \leq \mathcal{N} + 1,$
NN Neumann-Neumann	$D_s^{-1} A_{ \Omega^s} D_s^{-1}$	Same as above	$1 \leq \lambda(HA\Pi) \leq \mathcal{N}\tau$ $1 \leq \lambda(H_{\text{hyb}}A) \leq \mathcal{N}\tau$
IS Inexact Schwarz	$L^s L^{s\top} \approx R^s A R^{s\top}$ (no-fill incomplete Cholesky fact.)	$\mathcal{Y}_L(v, L^s L^{s\top}, R^s A R^{s\top}) +$ $\mathcal{Y}_L(\tau^{-1}, D_s^{-1} A_{ \Omega^s} D_s^{-1}, L^s L^{s\top})$ i.e. low frequency eigenvectors of $L^s L^{s\top} \mathbf{x}^s = \lambda^s R^s A R^{s\top} \mathbf{x}^s$ and of $D_s^{-1} A_{ \Omega^s} D_s^{-1} \mathbf{x}^s = \lambda^s L^s L^{s\top} \mathbf{x}^s$	$1/\tau \leq \lambda(HA\Pi) \leq \mathcal{N}/v$ $1/\tau \leq \lambda(H_{\text{hyb}}A) \leq \mathcal{N}/v$ $\frac{1}{\tau} \left(1 + 2\frac{\mathcal{N}}{v}\right)^{-1} \leq \lambda(H_{\text{ad}}A) \leq \frac{\mathcal{N}}{C_\#} + 1$

Summary of Notation:

- $\tilde{A}^s$: local solvers in the definitions of the preconditioners $H, H_{\text{hyb}}, H_{\text{ad}}$ (Definition 2.2.6),
- Π : coarse projector (Definition 2.2.6),
- $\mathcal{N}$: coloring constant (Definition 2.2.2),
- $A_{|\Omega^s}$: local Neumann matrix (defined in (2.12)),
- D_s : partition of unity (either defined by (2.13) or by (2.14)),
- $R^s A R^{s\top}$: restriction of the problem to subdomain number s where R^s is the restriction to the subdomain (see Assumption 2.2.1),
- L^s : factor in the no-fill incomplete Cholesky factorization $L^s L^{s\top} \approx R^s A R^{s\top}$ [54] (triangular matrix).
- $\tau > 1, v \in]0, 1[$: thresholds chosen by the user,
- $\mathcal{Y}_L$: low frequency eigenvectors of a certain generalized eigenvalue problem (Definition 2.2.8),
- $C_\#$: constant defined in Definition 2.2.10.

Table 2.2: Three DD preconditioners, their GenEO coarse spaces and spectral bounds. For the projected operator $HA\Pi$, the lower bounds hold for any non-zero eigenvalue. These bounds are for non-overlapping subdomains.

A remarkable feature is that the GenEO coarse spaces for AS and NN are identical: there is a set of coarse vectors that *fixes* both the Neumann-Neumann preconditioners and the Additive Schwarz preconditioners. This was already pointed out in [21]. The coarse space V^0 is formed by contributions coming from each subdomain $s \in \llbracket 1, N \rrbracket$. They can be computed as follows. First, a Cholesky factorization with pivoting of the matrix $M^s = D_s^{-1} A_{|\Omega^s} D_s^{-1}$ is performed. This gives both a factorization of M^s and an orthonormal basis Z^s for the kernel of M^s . Then the eigenvalue problem is solved to compute the lowest frequency eigenvectors. The factorization of M^s is crucial since an application of $(M^s)^\dagger$ is necessary at each iteration.

For IS, the contributions to the coarse space that come from each subdomain $s \in \llbracket 1, N \rrbracket$ require solving two generalized eigenvalue problems. For the first one, application of $(L^s L^{s\top})^{-1}$ are computed, a

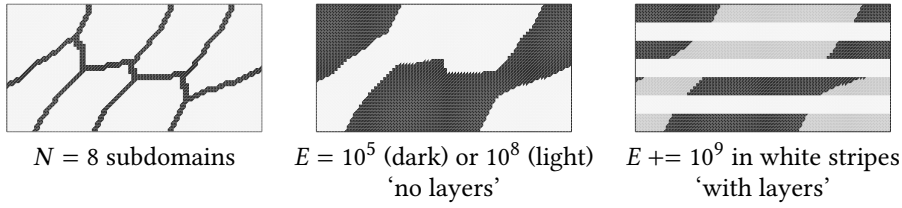


Figure 2.1: Partition into $N = 8$ subdomains (left). Two distributions of Young's modulus are considered (center and right). In the first, E is constant per subdomain: $E = 10^5$ if s is odd and $E = 10^8$ if s is even. The second distribution of E is obtained by adding some rigid layers to the first one: E is augmented by 10^9 if $y \in [1/7, 2/7] \cup [3/7, 4/7] \cup [5/7, 6/7]$. These test cases are referred to as 'no layers' and 'with layers'.

cost-effective task since L^s is triangular. The second eigenvalue problem resembles the one for Additive Schwarz (and Neumann-Neumann), in that the matrix that must be (pseudo-) inverted is the singular matrix M^s . However, it is possible to use a shifting procedure in order to solve instead for the largest eigenvalues of

$$[2\lambda_{\max}I - L^{s-1}M^s(L^{s\top})^{-1}]\mathbf{x}^s = \lambda^s\mathbf{x}^s,$$

where $\lambda_{\max}$ is the largest eigenvalue of $M^s\mathbf{x}^s = \lambda^s L^s L^{s\top} \mathbf{x}^s$, which is itself approximated using only linear solves by triangular matrices.

Numerical results

The results in this paragraph were obtained with the software libraries FreeFem++ [92] and GNU Octave [63]. The value of the coloring constant from Definition 2.2.2 is $\mathcal{N} = 3$ and there are $n_T = 546$ degrees of freedom that belong to more than one subdomain.

Results for AS. In Figure 2.2, the condition numbers for the Additive and Hybrid preconditioners are plotted with respect to the threshold τ for both coefficient distributions and both choices of scaling. The theoretical upper bounds are also plotted and never exceeded. The theoretical bound is less sharp when τ becomes larger. As expected, preconditioning by the Hybrid preconditioner always leads to a lower condition number than preconditioning by the fully Additive variant. Table 2.3 gives a lot more information. Only the test case with layers is considered. In all four configurations (hybrid/additive and μ -scaling/ k -scaling), the choice $\tau = 10$ seems to offer a good compromise between the condition number (or number of iterations) and the size of the coarse space. With the same value $\tau = 10$ the size of the coarse space with multiplicity scaling is 241 versus only 68 with k -scaling. This is due to the fact that the k -scaling already handles coefficient jumps that are across the subdomain interfaces (the ones that are already present in the 'no layers' test case). With multiplicity scaling, it is the coarse space that must handle also these jumps. To better illustrate this behaviour, Figure 2.3 shows the eigenvalues of the generalized eigenvalue problem. It is clear that the even-numbered subdomains (in which E is three orders of magnitude larger) must contribute many more vectors. (Recall that it is the high-frequency eigenvectors that get selected). This is not a failure of the GenEO method. In practice it is highly unlikely that an automatic graph partitioner would produce such a configuration. A human partitioner could produce such a configuration. If so, she would be aware of it and would choose the scaling accordingly.

Results for AS, NN and IS. These are presented in a lot less detail. Figure 2.4 (test cases with and without layers) shows the condition numbers with respect to the size of the coarse space for several

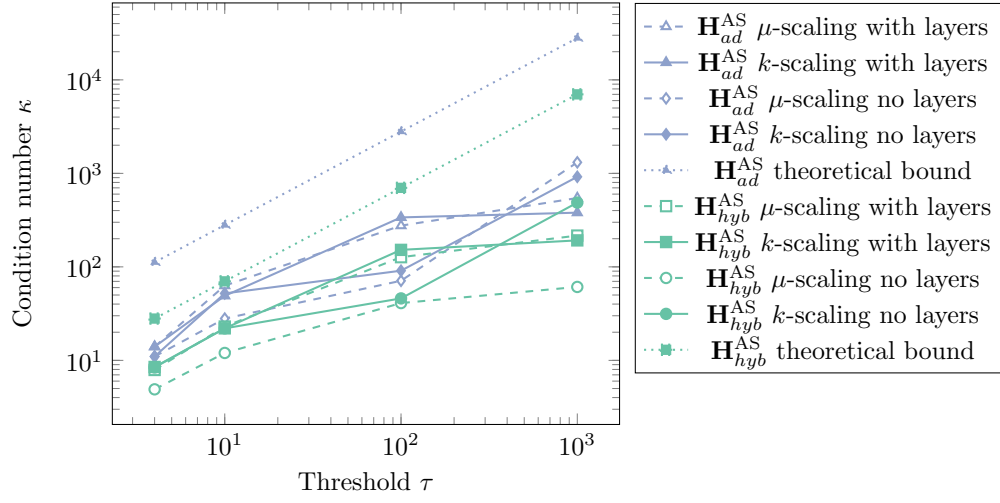


Figure 2.2: Condition numbers for Additive Schwarz preconditioners: additive and hybrid; μ -scaling and k -scaling; with and without layers; $\tau \in [4; 10; 100; 1000]$. All condition numbers are below the theoretical bound. The condition number is controlled by the value, set by the user, of the parameter τ .

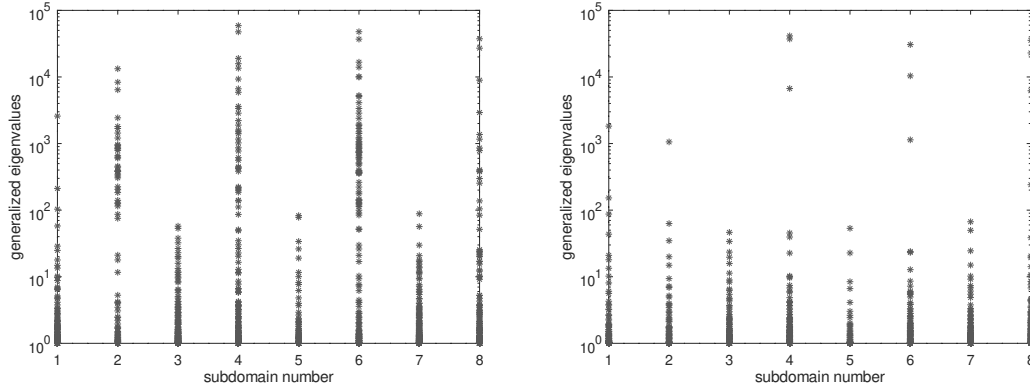


Figure 2.3: For each subdomain, solution of the generalized eigenvalue problem for computing V^0 in the case ‘with layers’. Left: μ -scaling, right: k -scaling.

values of the thresholds (that are not reported here). In this plot, the best methods are the ones which have data points closest to the origin (small condition number with a small coarse space). It appears clearly that k -scaling gives better results. This was previously explained.

With k -scaling the methods from most to least efficient rank as follows: Neumann-Neumann, Additive Schwarz with hybrid preconditioner, Additive Schwarz with additive preconditioner, Inexact Schwarz. Again, this does not tell the whole story as the cost of one iteration depends on the choice of method. With inexact Schwarz, the local solves are cheapest. With additive Schwarz in the additive variant, the coarse solve can be done in parallel to the local solves. With Neumann-Neumann, the matrices that must be handled with most care numerically (the ones that are singular) are the local solvers whereas they only appear in the generalized eigenvalue problem for the other methods. A second word of caution about Neumann-Neumann with k -scaling is that the scaling matrices D_s can be very ill-conditioned and make the whole method less efficient again. This problem is well known and independent of GenEO.

As a final remark on these results, let us comment on the *zig-zags* in the Inexact Schwarz data in Figure 2.4–Bottom. Since there are two parameters for the Inexact Schwarz coarse space, there are several choices of parameters that lead to coarse spaces that have the same dimension with possibly very different condition numbers. It is mostly only adding vectors to $Y_H(\tau, L^s L^{s\top}, M^s)$, that makes the method more efficient while adding vectors to $\mathcal{Y}_L(v, L^s L^{s\top}, R^s A R^{s\top})$ makes the coarse space grow very fast and the condition number decrease very little. Additional numerical results for Inexact Schwarz with an additive coarse correction can be found in [6].

Additive Schwarz with k -scaling						
		$\lambda_{\min}$	$\lambda_{\max}$	κ	It (final error)	$\#V^0$ (min $\#V_s^0$; max $\#V_s^0$)
one-level		$7.7 \cdot 10^{-4}$	3.0	3875	$> 100 (6 \cdot 10^{-3})$	0 (0 ; 0)
$\tau = 10^{10}$ (only $\text{Ker}(M^s)$)	hyb	0.0030	3.0	1003	$> 100 (2 \cdot 10^{-5})$	18 (0 ; 3)
	ad	0.0025	3.2	1271	$> 100 (2 \cdot 10^{-4})$	18 (0 ; 3)
$\tau = 1000$	hyb	0.016	3.0	192	98	29 (1 ; 6)
	ad	0.0087	3.3	380	$> 100 (3 \cdot 10^{-7})$	29 (1 ; 6)
$\tau = 100$	hyb	0.02	3.0	152	93	31 (1 ; 7)
	ad	0.098	3.3	338	$> 100 (2 \cdot 10^{-7})$	31 (1 ; 7)
$\tau = 10$	hyb	0.13	3.0	22	43	68 (5 ; 13)
	ad	0.069	3.37	49	63	68 (5 ; 13)
$\tau = 4$	hyb	0.35	3.0	8.5	26	118 (8 ; 20)
	ad	0.25	3.4	14	34	118 (8 ; 20)
Theory (τ)	hyb	$1/\tau$	$\mathcal{N} = 3$	3τ		
	ad	$1/(7\tau)$	$\mathcal{N} + 1 = 4$	28τ		

Table 2.3: Test case with hard layers — AS — $\lambda_{\min}$ and $\lambda_{\max}$: extreme eigenvalues, κ : condition number, It: iteration count (with relative error at iteration 100 in parenthesis if the method has not converged), $\#V^0$: dimension of the coarse space, min $\#V_s^0$: number of coarse vectors contributed by the subdomain that contributes the fewest vectors, max $\#V_s^0$: number of coarse vectors contributed by the subdomain that contributes the most eigenvectors. The one-level method does not satisfy any theoretical bound for $\lambda_{\min}$. When τ is decreased, the coarse space grows, $\lambda_{\min}$ increases and It decreases.

2.2.6 Scope and impact of this work

GenEO coarse spaces (and other spectral coarse spaces) are usually introduced for particular problems and/or particular domain decomposition methods. This work provides a very concise abstract framework for spectral deflation spaces in domain decomposition. It is written both for specialists who will find all results proved from scratch (which fills a gap in the theory), and for users of linear solvers. Indeed, the *recipe* for defining a spectral coarse space is given with examples of usual and unusual domain decomposition methods.

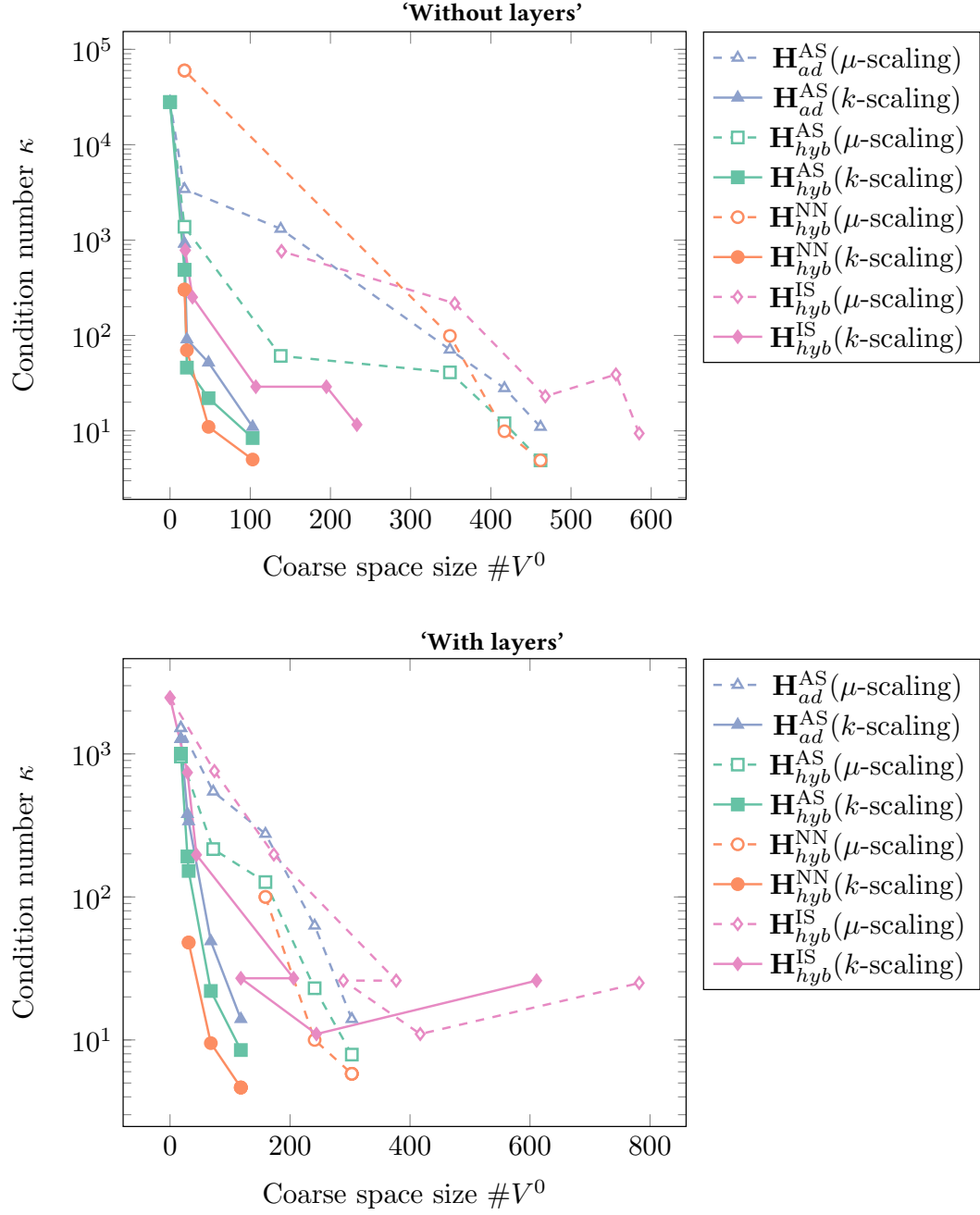


Figure 2.4: Efficiency of all methods (Condition number versus coarse space size). Top: the test case without layers in E . Bottom: test case with layers in E . These plots illustrate the trade-off between the condition number and the dimension of the coarse space.

The idea for the algebraic preconditioner proposed in the next section as well as its analysis are made possible by the abstract GenEO framework.

2.3 Fully algebraic preconditioner [12, 4]

This work is in collaboration with Loïc Gouarin (CNRS, CMAP, École polytechnique). A new family of preconditioners is introduced for spd linear systems. The new preconditioners, called the AWG preconditioners (for Algebraic-Woodbury-GenEO) are constructed algebraically. By this, we mean that only the knowledge of the matrix A for which the linear system is being solved is required. This is the same meaning as in Algebraic Multigrid (AMG) [39]. Thanks to the GenEO spectral coarse space technique, the condition number of the preconditioned operator is bounded theoretically from above. This upper bound can be made smaller by enriching the coarse space with more spectral modes.

Preexisting spectral coarse spaces require the knowledge of a set of symmetric positive semi-definite (spsd) matrices N^s that satisfy

$$\exists C > 0 \text{ such that } \sum_{s=1}^N \langle \mathbf{x}, R^{s\top} N^s R^s \mathbf{x} \rangle \leq C \langle \mathbf{x}, A \mathbf{x} \rangle \quad \forall \mathbf{x} \in \mathbb{R}^n. \quad (2.16)$$

(This is Assumption 2.2.4 in the abstract framework from Section 2.2 (with $N^s = D_s M^s D_s$.) These matrices N^s enter directly into the coarse space construction *via* the choice of matrix pencil for the generalized eigenvalue problems. In other words they play an essential role.

For matrices arising from discretized PDEs, the above assumption is far from unnatural and the matrices N^s are not expensive to compute as long as it is known at assembly that they are required (see $A|_{\Omega^s}$ in Section 2.2.5). Partial assembly over subdomains is neither hard conceptually nor expensive computationally and the purpose of this chapter is not to rule it out when it is possible. There are however many cases where only A is known or available. Then, the unassembled information is simply lost and the GenEO coarse spaces can't be computed. This is quite a common scenario: the problem may have been assembled by another user or with another piece of software. In this case only black box, algebraic, algorithms can be used.

Direct solvers, like MUMPS [24, 23], belong to the category of black box solvers and they are the most efficient up to a certain problem size. In the field of domain decomposition, the authors of [119] propose an algebraic preconditioner under the name DD-LR (for Domain Decomposition based Low-Rank). The original matrix is rewritten in a particular form inspired by domain decomposition. The inverse of A is expressed in terms of the components in that formulation thanks to the Woodbury matrix identity. Finally, a low rank approximation of one of the terms is performed in order to get an approximation of A^{-1} that can serve as a preconditioner. Our AWG preconditioners also exploit the Woodbury matrix identity but the modification of A that it is applied to is entirely different. Another competing DD method is [22].

Also closely related to DD are the multigrid algorithms, a very well-established set of solvers that solves the problem by iterating over the original (fine) problem as well as coarser and coarser approximations of it. The original multigrid algorithm [40, 44] is often referred to as geometric multigrid as it requires more information about the problem than just the matrix A . Our objective with this algorithm is not unlike the objective of Algebraic Multigrid (AMG) which is to make multigrid applicable in cases where less information about the problem is accessible or some assumptions are not satisfied (see [39, 146] for the original contributions and [168] for a unified presentation and theory of many multigrid algorithms). Our ambition here is to propose an algorithm as easy to apply as are the Algebraic Multigrid algorithms.

The methodology to construct the new AWG preconditioners is the following:

- The problem matrix A is split into symmetric, but possibly indefinite, matrices B^s as $A = \sum_{s=1}^N R^{s\top} B^s R^s$. This can always be done algebraically.
- The positive parts A_+^s of the matrices B^s are computed and assembled to form a global matrix

$A_+ = \sum_{s=1}^N R^s{}^\top A_+^s R^s$. By construction, (2.16) is satisfied with $N^s = A_+^s$ and $C = 1$. Consequently, two-level preconditioners with GenEO coarse spaces can be computed for A_+ by applying the abstract theory in [6], previously presented in Section 2.2.

- Finally, the Woodbury matrix identity relates the inverses of A and of A_+ and makes apparent that a good preconditioner for A can be obtained by adding a second coarse space to a good preconditioner for A_+ .

2.3.1 A new matrix A_+ and its two-level GenEO preconditioners

We place ourselves in the Abstract Schwarz framework for DD (Assumption 2.2.1). For simplicity, we assume that the partition into subdomains corresponds to a partition of the degrees of freedom into subsets Ω^s of $\llbracket 1, N \rrbracket$. We defer the choice of A^s . Since it is not known how to algebraically find local spd matrices that satisfy (2.16), it is chosen to relax the assumption by allowing the matrices to be indefinite.

Assumption 2.3.1. *Let's assume that there exists a family of symmetric matrices $B^s \in \mathbb{R}^{n^s \times n^s}$ for $s = 1, \dots, N$ such that*

$$A = \sum_{s=1}^N R^s{}^\top B^s R^s.$$

Such a family of matrices can always be chosen under Assumption 2.3.2 (minimal overlap). What's more, the minimal overlap condition can always be satisfied by adding a layer of overlap to the partition into subdomains.

Assumption 2.3.2 (Minimal overlap). *For any pair of indices $(i, j) \in \llbracket 1, n \rrbracket^2$, denoting by A_{ij} the coefficient of A at the i -th line and j -th column,*

$$A_{ij} \neq 0 \Rightarrow (\exists s \in \llbracket 1, N \rrbracket \text{ such that } \{i, j\} \subset \Omega^s).$$

Under the minimal overlap condition, the following choice of B^s provides a splitting of A into symmetric matrices.

$$B^s := R^s B R^s{}^\top \text{ with } B \in \mathbb{R}^{n \times n}; \quad B_{ij} := \begin{cases} \frac{A_{ij}}{\#\{s: \{i, j\} \subset \Omega^s\}} & \text{if } A_{ij} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

The first step in defining the very important matrix A_+ is to split B^s into a positive part and a negative semi-definite part.

Definition 2.3.3 (Splitting of B^s). *Let B^s , for $s \in \llbracket 1, N \rrbracket$, be a family of matrices that satisfy Assumption 2.3.1. For each s , let a diagonalization of B^s be written as*

$$B^s = V^s L^s V^s{}^\top; \text{ with } V^s \text{ orthogonal and } L^s \text{ diagonal.}$$

It can be assumed that the eigenvalues of B^s are sorted into positive and non-positive so that

$$L^s = \begin{pmatrix} L_-^s & \mathbf{0} \\ \mathbf{0} & L_+^s \end{pmatrix}, \quad V^s = [V_-^s | V_+^s], \quad L_+^s \text{ is spd, } -L_-^s \text{ is spsd.}$$

Finally, the two following matrices are defined in $\mathbb{R}^{n^s \times n^s}$:

$$A_+^s := V_+^s L_+^s V_+^s{}^\top \text{ and } A_-^s := -V_-^s L_-^s V_-^s{}^\top.$$

It is clear that, for each $s \in \llbracket 1, N \rrbracket$, both matrices A_+^s and A_-^s are spd matrices and that $B^s = A_+^s - A_-^s$. Next, global matrices are computed by assembling the local components with the usual restriction and prolongation operators R^s and $R^{s\top}$.

Definition 2.3.4 (New matrices A_+ and A_-). *Let the global matrices $A_+ \in \mathbb{R}^{n \times n}$ and $A_- \in \mathbb{R}^{n \times n}$ be defined by*

$$A_+ := \sum_{s=1}^N R^{s\top} A_+^s R^s, \text{ and } A_- := \sum_{s=1}^N R^{s\top} A_-^s R^s.$$

The matrices A_+ and A_- from Definition 2.3.4 satisfy the following three properties

- (i) $A = A_+ - A_-$,
- (ii) the matrix A_- is symmetric positive semi-definite,
- (iii) the matrix A_+ is symmetric positive definite.

By construction, the spd matrix A_+ also satisfies (2.16) with $N^s = A_+^s$ and $C = 1$. In other words: the abstract GenEO theory from Section 2.2 applies to defining and analyzing two-level preconditioners for A_+ . Three of these preconditioners are briefly presented in Table 2.4. No more detail is given here. Instead the next step is presented: how any good preconditioner H_2 for A_+ is adapted into a preconditioner for A with the Woodbury matrix identity.

2.3.2 Woodbury matrix identity for $A = A_+ - A_-$

The new preconditioner for A arises from the realization that A can be viewed as a low rank modification of A_+ and adding a term to H_2 accordingly. Let $L_- \in \mathbb{R}^{n_- \times n_-}$ and $V_- \in \mathbb{R}^{n \times n_-}$ be the diagonal matrix and the orthogonal matrix that are obtained by removing the null part of A_- from its diagonalization in such a way that

$$A_- := A_+ - A = V_- L_- V_-^\top.$$

It now holds that $A = A_+ - V_- L_- V_-^\top$ with A , A_+ , L_- spd matrices and V_- a full rank matrix. The Woodbury matrix identity [166] applied to computing the inverse of A gives

$$A^{-1} = A_+^{-1} + A_+^{-1} V_- (L_-^{-1} - V_-^\top A_+^{-1} V_-)^{-1} V_-^\top A_+^{-1}. \quad (2.17)$$

Remark 2.3.5. *The Woodbury matrix identity is also called the Sherman-Morrison-Woodbury formula (e.g., in [82][Section 2.1.4]). The formula is correct since A_+ is non-singular and $L_-^{-1} - V_-^\top A_+^{-1} V_-$ is also non-singular.*

2.3.3 AWG preconditioner for A with inexact coarse solve

The Woodbury matrix identity leads, rather straightforwardly, to a new preconditioner for the original matrix A that is defined in the following theorem.

Theorem 2.3.6 (AWG preconditioner for A with inexact coarse space). *Given a preconditioner H_2 for A_+ such that the eigenvalues of $H_2 A_+$ are in the interval $[\lambda_{\min}(H_2 A_+), \lambda_{\max}(H_2 A_+)]$, let the inexact AWG preconditioner for A be defined as*

$$H_{3,\text{inex}} := H_2 + A_+^{-1} V_- (L_-^{-1} - V_-^\top A_+^{-1} V_-)^{-1} V_-^\top A_+^{-1}.$$

The eigenvalues of the new preconditioned operator satisfy

$$\min(1, \lambda_{\min}(H_2 A_+)) \leq \lambda(H_{3,\text{inex}} A) \leq \max(1, \lambda_{\max}(H_2 A_+)). \quad (2.18)$$

DD Preconditioner	Local contributions to GenEO coarse space	Spectral bounds for two-level preconditioners
$H_+^{AS} := \sum_{s=1}^N R^s (R^s A_+ R^{s\top})^{-1} R^s$	$\mathcal{Y}_L(\tau, D_s^{-1} A_+^s D_s^{-1}, R^s A_+ R^{s\top})$ i.e. low frequency eigenvectors of $D_s^{-1} A_+^s D_s^{-1} \mathbf{x}^s = \lambda^s R^s A_+ R^{s\top} \mathbf{x}^s$	$\tau \leq \lambda(H_{+, \text{hyb}}^{AS} A_+) \leq \mathcal{N}(A_+)$ $\tau/(1 + 2\mathcal{N}(A_+)) \leq \lambda(H_{+, \text{ad}}^{AS} A_+) \leq \mathcal{N}(A_+) + 1$
$H^{NN} := \sum_{s=1}^N R^{s\top} D_s (A_+^s)^\dagger D_s R^s$ $= \sum_{s=1}^N R^{s\top} D_s V_+^s (L_+^s)^{-1} V_+^{s\top} D_s R^s$	Same as above	$1 \leq \lambda(H_{\text{hyb}}^{NN} A_+) \leq \mathcal{N}(A_+)/\tau$
$H^{AS} := \sum_{s=1}^N R^{s\top} (R^s A R^{s\top})^{-1} R^s$	$\mathcal{Y}_L(v, D_s^{-1} A_+^s D_s^{-1}, R^s A R^{s\top}) +$ $\mathcal{Y}_L(\tau, R^s A R^{s\top}, R^s A_+ R^{s\top})$ i.e. low frequency eigenvectors of $D_s^{-1} A_+^s D_s^{-1} \mathbf{x}^s = \lambda^s R^s A R^{s\top} \mathbf{x}^s$ and of $R^s A R^{s\top} \mathbf{x}^s = \lambda^s R^s A_+ R^{s\top} \mathbf{x}^s$	$v \leq \lambda(H_{\text{hyb}}^{AS} A_+) \leq \mathcal{N}(A_+)/\tau$

Summary of Notation:

- A : problem matrix in the linear system $A\mathbf{x} = \mathbf{b}$,
- A_+ : matrix (from Definition 2.3.4) for which we are defining a preconditioner,
- $\cdot_{\text{hyb}}$ (resp. $\cdot_{\text{ad}}$): subindices that denote the hybrid (resp. additive) two level preconditioner following Definition 2.2.6 for the given one-level preconditioner and coarse space,
- R^s : the restriction to the subdomain (see Assumption 2.2.1),
- $\mathcal{N}(A_+)$: coloring constant with respect to matrix A_+ (following Definition 2.2.2 for A_+),
- D_s : partition of unity – a set of local spd matrices that satisfy $I = \sum_{s=1}^N R^{s\top} D_s R^s$,
- A_+^s , V_+^s , and L_+^s are as in Definition 2.3.3,
- $\tau, v \in]0; 1[$: thresholds chosen by the user,
- $\mathcal{Y}_L$: low frequency eigenvectors of a certain generalized eigenvalue problem (Definition 2.2.8).

Table 2.4: Three DD preconditioners for A_+ as well as their GenEO coarse spaces and spectral bounds. The notation from Section 2.2 is employed. Additionally, the matrices D_s form a partition of unity. Once more, the GenEO coarse space is the same for H^{NN} (local solvers come from the splitting of A_+) and H_+^{AS} (exact local solvers for A_+).

Proof. The estimate for the eigenvalues of $H_2 A_+$ is equivalent to

$$\lambda_{\min}(H_2 A_+) \langle \mathbf{x}, A_+^{-1} \mathbf{x} \rangle \leq \langle \mathbf{x}, H_2 \mathbf{x} \rangle \leq \lambda_{\max}(H_2 A_+) \langle \mathbf{x}, A_+^{-1} \mathbf{x} \rangle, \forall \mathbf{x} \in \mathbb{R}^n.$$

Adding, $\langle \mathbf{x}, A_+^{-1} V_- (D_-^{-1} - V_-^\top A_+^{-1} V_-)^{-1} V_-^\top A_+^{-1} \mathbf{x} \rangle$ to each term, it holds that

$$\min(1, \lambda_{\min}(H_2 A_+)) \langle \mathbf{x}, A_+^{-1} \mathbf{x} \rangle \leq \langle \mathbf{x}, H_{3, \text{inex}} \mathbf{x} \rangle \leq \max(1, \lambda_{\max}(H_2 A_+)) \langle \mathbf{x}, A_+^{-1} \mathbf{x} \rangle, \forall \mathbf{x} \in \mathbb{R}^n,$$

where the Woodbury matrix identity (2.17) was applied. This is equivalent to (2.18). $\square$

The name *inexact* comes from the fact that the term $A_+^{-1} V_- (L_-^{-1} - V_-^\top A_+^{-1} V_-)^{-1} V_-^\top A_+^{-1}$ in $H_{3, \text{inex}} A$ has the form of a coarse space correction with an inexact coarse operator $(L_-^{-1} - V_-^\top A_+^{-1} V_-)$. Next, two other AWG preconditioners are defined which have an exact coarse solve in the sense of the Abstract Schwarz theory: their coarse operator is of the form $R^0 A R^{0\top}$.

2.3.4 AWG preconditioner for A with exact coarse solve

From the Woodbury identity, it follows that $\text{range}(A_+^{-1}V_-) = \text{range}(A^{-1}V_-)$. Consequently, if it is computationally possible to compute $\text{range}(A_+^{-1}V_-)$, it is also possible to project A -orthogonally onto the space that is I -orthogonal to $\text{range}(V_-)$ which is exactly $\text{Ker}(A_-)$ (see Π_3 in Definition 2.3.7 below). On that space, $A|_{\text{Ker}(A_-)} = (A_+)|_{\text{Ker}(A_-)}$ and we can fall back onto a known and efficient preconditioner H_2 for A_+ .

Definition 2.3.7 (AWG preconditioners for A). *Let $W \in \mathbb{R}^{n \times n_-}$ be such that $\text{range}(W) = \text{range}(A_+^{-1}V_-)$ and let*

$$\Pi_3 := I - W(W^\top AW)^{-1}W^\top A.$$

Assume that H_2 is a given preconditioner for A_+ such that the eigenvalues of H_2A_+ are in the interval $[\lambda_{\min}(H_2A_+), \lambda_{\max}(H_2A_+)]$. Let two new preconditioners for A be defined as

$$H_{3,\text{ad}} := H_2 + W(W^\top AW)^{-1}W^\top \text{ (Additive AWG preconditioner),}$$

and

$$H_{3,\text{hyb}} := \Pi_3 H_2 \Pi_3^\top + W(W^\top AW)^{-1}W^\top \text{ (Hybrid AWG preconditioner).}$$

Theorem 2.3.8 (Convergence of AWG). *Let Π_3 , $H_{3,\text{ad}}$ and $H_{3,\text{hyb}}$ be as in Definition 2.3.7. The operator Π_3 is an A -orthogonal projection operator that satisfies*

$$\text{Ker}(\Pi_3) = \text{range}(A^{-1}V_-) \text{ and } \text{range}(\Pi_3) = \text{Ker}(A_-). \quad (2.19)$$

Moreover, the new preconditioned operators satisfy the spectral bounds :

$$\lambda_{\min}(H_2A_+) \leq \lambda(H_2A\Pi_3) \leq \lambda_{\max}(H_2A_+) \text{ if } \lambda(H_2A\Pi_3) \neq 0, \quad (2.20)$$

$$\min(1, \lambda_{\min}(H_2A_+)) \leq \lambda(H_{3,\text{hyb}}A) \leq \max(1, \lambda_{\max}(H_2A_+)), \quad (2.21)$$

$$\min(1, \lambda_{\min}(H_2A_+)) \leq \lambda(H_{3,\text{ad}}A) \leq (\lambda_{\max}(H_2A_+) + 1). \quad (2.22)$$

Note that a bound for the projected and preconditioned operator ($H_2A\Pi_3$) has also been included in the theorem (equation (2.20)).

Proof. We begin by proving (2.19). For $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}$ is in the kernel of Π_3 if:

$$\Pi_3 \mathbf{x} = 0 \Leftrightarrow \mathbf{x} = W(W^\top AW)^{-1}W^\top A\mathbf{x} \Leftrightarrow \mathbf{x} \in \text{range}(W) = \text{range}(A_+^{-1}V_-) = \text{range}(A^{-1}V_-).$$

Moreover, Π_3 is an A -orthogonal projection so

$$\text{range}(\Pi_3) = (\text{Ker}(\Pi_3))^{\perp_A} = (\text{range}(A^{-1}V_-))^{\perp_A} = (\text{range}(V_-))^{\perp_{\ell_2}} = \text{Ker}(V_-^\top) = \text{Ker}(A_-),$$

by definition of V_- . A direct consequence of this result, that is frequently used in the remainder of the proof, is the identity $A_+\Pi_3 = A\Pi_3$.

The remainder of the (quite long) proof is skipped. It consists mostly in manipulating spectral equivalences. $\square$

2.3.5 Scope and impact of this work

Algebraic linear solvers are the only type of solver that many scientists and engineers can rely on. This can be because they are not expert enough in numerical linear algebra to set up a non-algebraic solver, or because assembling matrices of the PDE over a subdomain is more intrusive than what can be achieved

with the software. Having a strong impact with the AWG algorithm requires two things. On one hand, the algorithm must be continuously improved to be competitive with algebraic multigrid methods. On the other hand, it is necessary to propose an open-source code so that users can test the method and apply it. With this in mind, our open-source software is presented next.

2.4 Open-source software [13]

2.4.1 Presentation of the code

In collaboration with Loïc Gouarin, we develop open source software [13] that can be downloaded from Github (under the BSD license). The code is a Python code that relies on the `petsc4py` library [58] (a Python interface for the C++ library PETSc [29, 28, 30]). Eigenvalue problems are solved by SLEPc [99] and all matrix factorizations are performed by MUMPS [24, 23].

2.4.2 Numerical results for AWG

Let's examine a result obtained with our code. It compares the new AWG algorithm (presented in the previous section) to other domain decomposition methods with a GenEO coarse space (ones that are non-algebraic). The test case is a finite element Q_1 discretization of a linear elasticity problem posed over the domain $\omega = [0, 3] \times [0, 3]$ (with homogeneous Dirichlet boundary conditions on the left hand side and a free boundary everywhere else). Poisson's ratio is set to $\nu = 0.3$. Young's modulus is heterogeneous and represented in Figure 2.5. The mesh size is $h = 1/21$ so that the problem size is $n = 8064$. The partition is into $N = 9$ subdomains. PCG is run up to a tolerance of 10^{-10} on the preconditioned residual. The solution is shown in Figure 2.5 where the domain ω has been deformed by the solution to the linear system. This makes sense because the unknown represents a displacement.

The results are presented in Table 2.5. Several variants of the AWG preconditioners and the more classical preconditioners are presented. Their definitions can be found in Table 2.2 for the non-algebraic preconditioners, Table 2.4 for H_2 and Definition 2.3.7 for the resulting AWG preconditioners. The bottom row in the table corresponds to a one-level preconditioner (that has no coarse space). This is the only solver considered for which there is no scalability or robustness result. It serves as a comparison to confirm that the problems considered are numerically difficult. All condition numbers for methods that have a coarse space (either AWG or the non-algebraic coarse space) are small. As a result, the iteration counts are also small. The good news is that the dimension of the AWG preconditioner is close to the dimension of the non-algebraic version. Let us recall that the objective with AWG was not to propose a better coarse space than the usual GenEO but to propose an algebraic version that is almost as efficient. We conclude that these numerical results point towards efficiency of AWG.

2.4.3 Scope and impact of this work

The first ambition of this code is to test both existing domain decomposition methods as well as the ones that we develop (e.g., AWG from the previous section). This has been achieved since the numerical results in [4] were produced by our code. As with any code, stellar parallel performances are necessary to encourage other users both to use our code and the AWG algorithms. Further developments to the code are an important part of ANR DARK project which will take place during the next four years under my responsibility.

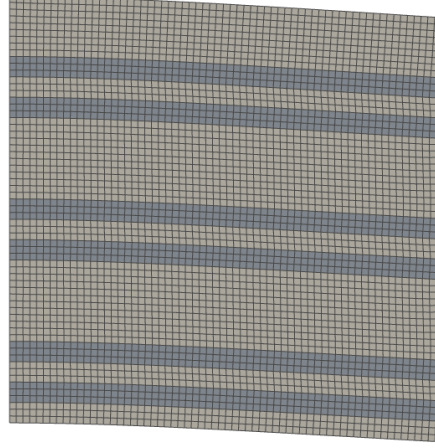


Figure 2.5: Dark-colored: $E = E_1 = 10^{11}$ (harder material); Light-colored : $E = E_2 = 10^7$ (softer material). Solution of the two-dimensional elasticity test case.

	κ	It	$\#V_0$	n_-
AWG Preconditioners (algebraic):				
$H_{3,ad}$ with $H_2 = H_{hyb}^{NN}$	9.09	26	57	48
$H_{3,ad}$ with $H_2 = H_{+,hyb}^{AS}$	12.3	25	57	48
$H_{3,ad}$ with $H_2 = H_{+,ad}^{AS}$	16.8	31	57	48
$H_{3,hyb}$ with $H_2 = H_{hyb}^{NN}$	9.09	27	57	48
$H_{3,hyb}$ with $H_2 = H_{+,hyb}^{AS}$	12.2	25	57	48
$H_{3,hyb}$ with $H_2 = H_{+,ad}^{AS}$	16.7	29	57	48
Non-algebraic preconditioners:				
Hybrid AS + GenEO ($\tau = 10$)	26.5	43	55	0
Additive AS + GenEO ($\tau = 10$)	50.0	58	55	0
NN + GenEO ($\tau = 10$)	11.1	29	55	0
One-level preconditioner (not robust, not scalable):				
One-level AS	34772	> 150	0	0

Table 2.5: κ : condition number of preconditioned operator; It : iteration count; $\#V_0$: dimension of GenEO coarse space ; n_- : dimension of second coarse space. All methods with a spectral coarse space have a low condition number and iteration count. The coarse space for AWG is almost the same dimension as for (non-algebraic) classical GenEO. The additional coarse space also is roughly of the same dimension.

Chapter 3

Multipreconditioning of Krylov Subspace Methods

3.1 A word about results not included: Block FETI [3]

Block Krylov subspace methods [138] are a natural extension of Krylov subspace methods for solving linear systems with multiple right-hand sides. By splitting a single right-hand side, they can also accelerate the solution of a single system. Particularly well suited are problems preconditioned by DD with the right-hand side split over the subdomains. Before initiating work on adaptive multipreconditioning, together with Pierre Gosselet, Daniel Rixen, and François-Xavier Roux, we considered applying either multipreconditioned CG or block CG to the FETI DD method. Our results are published in [3]. We show, on examples from structural mechanics, that both methods are robust and provide good convergence in the presence of high heterogeneities, even when the interface is jagged or when the domains have a bad aspect ratio. The multipreconditioned algorithm was efficiently implemented in an optimized parallel code and exhibited excellent performance compared to the regular FETI method. The conclusion of this work is in favor of multipreconditioning which further motivated the remainder of the work in this chapter.

3.2 The adaptive multipreconditioned CG algorithm (AMPCG) [5]

In 2006 the Multi-Preconditioned Conjugate Gradient (MPCG) algorithm was introduced by Bridson and Greif [43]. It is an iterative linear solver, adapted from the Preconditioned Conjugate Gradient (PCG) algorithm [149] which can be used in cases where several preconditioners are available or the usual preconditioner is a sum of contributions. MPCG is related to the block CG algorithm [138, 137] and the multiparameter descent method [42]. In [43] it was already pointed out that Domain Decomposition algorithms are ideal candidates to benefit from MPCG. This was further studied in [88] which considers Additive Schwarz preconditioners in the Multi Preconditioned GMRES (MPGMRES) [87] setting. In 1997, Rixen had proposed in his thesis [144] the Simultaneous FETI algorithm which turns out to be MPCG applied to FETI. The algorithm is more extensively studied in [3] where its interpretation as an MPCG solver is made explicit.

The strength of the multipreconditioned methods is that they generate a large number of search directions. Their drawback is that the cost of minimizing over these becomes increasingly expensive. Instead, the newly proposed adaptive multipreconditioning considers each contribution arising from the application of one preconditioner (denoted H^s) as a *candidate* [38] to augment the space in which the

solution is sought for (called the minimization space). A theoretical estimate (3.3) predicts whether this candidate should augment the minimization space or whether it should only contribute to it through a global preconditioner $\sum_{s=1}^N H^s$. The estimate is only practical if a lower bound for the eigenvalues of the preconditioned operator is known (e.g., for the substructuring DD methods, BDD and FETI, $\lambda_{\min} \geq 1$ [126, 118, 159]).

The original motivation that led to the Adaptive multipreconditioned CG algorithm (AMPCG) was to compute the same deflation space as GenEO without needing to solve generalized eigenvalue problems.

3.2.1 Theory of PPCG

PPCG [60], a combination of PCG and deflation, has already been discussed in the introduction. It is presented once more in Algorithm 2. Let A be the spd problem matrix, let H be the spd preconditioner and let $U \in \mathbb{R}^{n \times m}$ be a full-rank matrix that enters into the definition of the deflation operator as follows

$$\Pi := I - U(U^\top AU)^{-1}U^\top A.$$

Algorithm 2 : PPCG algorithm for $A\mathbf{x} = \mathbf{b}$ preconditioned by H and deflated by $\Pi = I - U(U^\top AU)^{-1}U^\top A$ for initial guess $\mathbf{x}_{00}$

```

 $\mathbf{x}_0 = \Pi \mathbf{x}_{00} + U(U^\top AU)^{-1}U^\top \mathbf{b};$   $\leftarrow \mathbf{x}_0 = \Pi(\mathbf{x} - \mathbf{x}_{00})$ 
 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0;$ 
 $\mathbf{z}_0 = H\mathbf{r}_0;$ 
 $\mathbf{p}_0 = \Pi \mathbf{z}_0;$   $\leftarrow$  Projected initial search direction
for  $i = 0, 1, \dots$ , convergence do
     $\mathbf{q}_i = A\mathbf{p}_i;$ 
     $\alpha_i = \frac{\langle \mathbf{r}_i, \mathbf{z}_i \rangle}{\langle \mathbf{q}_i, \mathbf{p}_i \rangle};$ 
     $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i;$ 
     $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{q}_i;$ 
     $\mathbf{z}_{i+1} = H\mathbf{r}_{i+1};$ 
     $\beta_i = \frac{\langle \mathbf{z}_{i+1}, A\mathbf{p}_i \rangle}{\langle \mathbf{p}_i, A\mathbf{p}_i \rangle};$ 
     $\mathbf{p}_{i+1} = \Pi \mathbf{z}_{i+1} - \beta_i \mathbf{p}_i;$   $\leftarrow$  Projected search direction
end
Return  $\mathbf{x}_{i+1};$ 

```

One way to estimate the relative error of PPCG is to use the following convergence result from [76] (that generalizes the classical convergence result for PCG of [130, 107], see also [149, Theorem 6.29]):

$$\|\mathbf{x} - \mathbf{x}_i\|_A = \min_{q \in \mathbb{P}_i; q(0)=1} \max_{\lambda \in \sigma(HA\Pi) \setminus \{0\}} |q(\lambda)| \|\mathbf{x} - \mathbf{x}_0\|_A \leq 2 \left[\frac{\sqrt{\kappa_{\text{eff}}(HA\Pi)} - 1}{\sqrt{\kappa_{\text{eff}}(HA\Pi)} + 1} \right]^i \|\mathbf{x} - \mathbf{x}_0\|_A, \quad (3.1)$$

where $\sigma(HA\Pi)$ is the spectrum of $HA\Pi$, $\mathbb{P}_i$ is the set of polynomials of degree at most i , and

$$\kappa_{\text{eff}}(HA\Pi) := \frac{\max(\sigma(HA\Pi))}{\min(\sigma(HA\Pi) \setminus \{0\})},$$

denotes the effective condition number of $HA\Pi$.

Proof of (3.1) (also labelled (1.13) in the introduction). Recall from the introduction that equation (1.12) characterizes the iterates of PPCG as

$$\|\mathbf{x} - \mathbf{x}_i\|_A = \min_{q \in \mathbb{P}_i; q(0)=1} \|q(\Pi H A)(\mathbf{x} - \mathbf{x}_0)\|_A.$$

Recall also that $\Pi^\top A \Pi = \Pi^\top A = A \Pi$. Consider the preconditioned and projected operator $H A \Pi$. If $(\lambda, \mathbf{y})$ is an eigenpair of $H A \Pi$ then $(\lambda, \mathbf{y})$ also solves the generalized eigenvalue problem $\Pi^\top A \Pi \mathbf{y} = \lambda H^{-1} \mathbf{y}$. The matrix on the right hand side is spd while the matrix on the left is spsd. Consequently, there exists a basis of eigenvectors that are H^{-1} -orthonormal.¹ Let these be denoted by $\mathbf{y}_i$, for $i = 1, \dots, n$ and let λ_i be the eigenvalue corresponding to $\mathbf{y}_i$, i.e.,

$$H A \Pi \mathbf{y}_i = \lambda_i \mathbf{y}_i, \quad \|\mathbf{y}_i\|_{H^{-1}} = 1; \quad \langle \mathbf{y}_i, H^{-1} \mathbf{y}_j \rangle = \langle \mathbf{y}_i, \Pi^\top A \Pi \mathbf{y}_j \rangle = 0 \text{ if } i \neq j.$$

There exist coefficients $\xi_i \in \mathbb{R}$ such that $\mathbf{x} - \mathbf{x}_0 = \sum_{i=1}^n \xi_i \mathbf{y}_i$. Moreover, $(\mathbf{x} - \mathbf{x}_0) \in \text{range}(\Pi)$ so $\mathbf{x} - \mathbf{x}_0 = \sum_{i=1}^n \xi_i \Pi \mathbf{y}_i = \sum_{\{i; \lambda_i \neq 0\}} \xi_i \Pi \mathbf{y}_i$ and

$$\begin{aligned} \|q(\Pi H A)(\mathbf{x} - \mathbf{x}_0)\|_A^2 &= \|\Pi q(H A \Pi) \sum_{\{i; \lambda_i \neq 0\}} \xi_i \mathbf{y}_i\|_A^2 = \|\Pi \sum_{\{i; \lambda_i \neq 0\}} q(\lambda_i) \xi_i \mathbf{y}_i\|_A^2 \\ &= \sum_{\{i; \lambda_i \neq 0\}} |q(\lambda_i) \xi_i|^2 \|\Pi \mathbf{y}_i\|_A^2, \end{aligned}$$

which is bounded from above by

$$\|q(\Pi H A)(\mathbf{x} - \mathbf{x}_0)\|_A^2 \leq \max_{\lambda \in \sigma(H A \Pi) \setminus \{0\}} |q(\lambda)| \sum_{\{i; \lambda_i \neq 0\}} |\xi_i|^2 \|\Pi \mathbf{y}_i\|_A^2 = \max_{\lambda \in \sigma(H A \Pi) \setminus \{0\}} |q(\lambda)| \|\mathbf{x} - \mathbf{x}_0\|_A^2.$$

The second bound in (3.1) is the usual upper bound for the min-max problem posed on a segment in $\mathbb{R}^+$ (see, e.g., [149, equation (6.127)]) that was already mentioned about (1.7) of the introduction. $\square$

Consequently, a strategy for choosing U is to compute, or approximate, all isolated eigenvalues of $H A$ and use the corresponding eigenvectors as a basis for the deflation space $\text{range}(U)$. This way, the spectrum of $H A \Pi$ is clustered and (1.13) guarantees good convergence.

3.2.2 Main Result: Adaptive MPCG (AMPCCG)

The algorithms proposed here are closely related to deflation except that U is initialized with vectors chosen *a priori* and then augmented with vectors selected on the fly. Let $n \in \mathbb{N}$. The five assumptions are the following.

- (A1) $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix.
- (A2) $H \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix.
- (A3) $U \in \mathbb{R}^{n \times m}$ is a full rank matrix with $m < n$.
- (A4) A lower bound $\lambda_{\min}$ for the spectrum of $H A$ is known.
- (A5) The preconditioner H is a sum of N contributions : $H = \sum_{s=1}^N H^s$, with each H^s symmetric and positive semi-definite.

¹To prove this, consider the diagonalization of the spsd matrix $H^{1/2} \Pi^\top A \Pi H^{1/2}$.

Motivation for the τ -test in AMPCG. With the notation from Algorithm 2, and denoting by $\mathbf{d}_i$ the error at iteration i : $\mathbf{d}_i = \mathbf{x} - \mathbf{x}_i$, the authors in [26] prove that²

$$\|\mathbf{d}_i\|_A^2 = \|\mathbf{d}_{i-1}\|_A^2 - \alpha_{i-1}^2 \|\mathbf{p}_{i-1}\|_A^2, \text{ for all } i = 1, \dots, n - m - 1. \quad (3.2)$$

The proof consists of three steps: first, by the finite termination property, the exact solution can be written as

$$\mathbf{x} = \mathbf{x}_0 + \sum_{j=0}^{n-m-1} \alpha_j \mathbf{p}_j = \mathbf{x}_i + \sum_{j=i}^{n-m-1} \alpha_j \mathbf{p}_j.$$

Then, the A -conjugacy between search directions implies that $\|\mathbf{d}_i\|_A^2 = \|\mathbf{x} - \mathbf{x}_i\|_A^2 = \sum_{j=i}^{n-m-1} \alpha_j^2 \|\mathbf{p}_j\|_A^2$. Finally, (3.2) follows easily by subtraction.

The authors use this to derive some *a posteriori* error estimates and stopping criteria. Here, from the same starting point, two adaptive algorithms are derived with the objective of accelerating convergence when necessary. Assume that at iteration i the exact solution has not yet been found (i.e., $\mathbf{d}_i \neq \mathbf{0}$) then (3.2) can be rewritten as

$$\frac{\|\mathbf{d}_{i-1}\|_A^2}{\|\mathbf{d}_i\|_A^2} = 1 + \frac{\|\alpha_{i-1} \mathbf{p}_{i-1}\|_A^2}{\|\mathbf{d}_i\|_A^2} = 1 + \frac{\|\alpha_{i-1} \mathbf{p}_{i-1}\|_A^2}{\|\mathbf{r}_i\|_H^2} \frac{\|\mathbf{r}_i\|_H^2}{\|\mathbf{d}_i\|_A^2} = 1 + \frac{\|\alpha_{i-1} \mathbf{p}_{i-1}\|_A^2}{\|\mathbf{r}_i\|_H^2} \frac{\|\mathbf{d}_i\|_{HA}^2}{\|\mathbf{d}_i\|_A^2}.$$

Noting that the last factor is related to a Rayleigh quotient for HA it follows that

$$\frac{\|\mathbf{d}_i\|_A^2}{\|\mathbf{d}_{i-1}\|_A^2} \leq \left(1 + \lambda_{\min} \frac{\|\alpha_{i-1} \mathbf{p}_{i-1}\|_A^2}{\|\mathbf{r}_i\|_H^2} \right)^{-1}. \quad (3.3)$$

Remark 3.2.1. In the recent work [133], it was remarked that $\lambda_{\min}$ can be replaced by the smallest non-zero eigenvalue of the projected operator $H\Pi$.

From estimate (3.3) it can be deduced that, if there exists $\tau > 0$ such that $\tau \|\mathbf{r}_i\|_H^2 \leq \|\alpha_{i-1} \mathbf{p}_{i-1}\|_A^2$ at every iteration $i = 1, \dots, j$, then $\|\mathbf{d}_j\|_A / \|\mathbf{d}_0\|_A \leq (1 + \lambda_{\min} \tau)^{-j/2}$. Conversely, to guarantee that the error decreases at least linearly with a given contraction factor ρ (i.e., $\|\mathbf{d}_i\| / \|\mathbf{d}_{i-1}\| \leq \rho$), it is sufficient to check that:

$$\frac{\|\alpha_{i-1} \mathbf{p}_{i-1}\|_A^2}{\|\mathbf{r}_i\|_H^2} \geq \tau \text{ with } \tau := \frac{1 - \rho^2}{\lambda_{\min} \rho^2}. \quad (3.4)$$

The adaptive step in AMPCG is based on evaluating, at each iteration, whether (3.4) holds or not. In the case where it doesn't, convergence is accelerated by multipreconditioning.

Presentation of AMPCG

Algorithm 3 presents AMPCG and introduces some new notation. AMPCG is designed to adapt automatically if convergence is too slow. More precisely, given a threshold $\tau \in \mathbb{R}^+$ chosen by the user, the adaptive steps are between lines 7 and 12. The test in line 8 is referred to as the τ -test.

- If the τ -test returns $t_i \geq \tau$, then the algorithm predicts that there is no need to adapt and performs a step of PPCG with preconditioner $\sum_{s=1}^N H^s$.
- If the τ -test returns $t_i < \tau$, then the PPCG algorithm is not reducing the error sufficiently and the algorithm performs one step of the (projected) MPCG algorithm [43] for the N preconditioners H^s .

²The proof in [26] is for the non-projected PCG ($m = 0$) but goes through in the same way when considering PPCG with the difference that the process terminates in at most $n - m$ iterations (instead of n).

This automatically improves the next approximate solution since the error is minimized over a larger search space that includes the usual search space (see Property 2 in Theorem 3.2.3).

There are two extreme choices for τ : if $\tau = 0$ then the usual PPCG iterations are recovered and if $\tau = \infty$ then the (projected) MPCG algorithm is recovered.

Algorithm 3 : AMPCG with global τ -test for $A\mathbf{x} = \mathbf{b}$ preconditioned by $(\sum_{s=1}^N H^s)$ and deflated by Π for initial guess $\mathbf{x}_0$. $\tau \in \mathbb{R}^+$: chosen by user.

```

1  $\mathbf{x}_0 = \Pi\mathbf{x}_{00} + U(U^\top AU)^{-1}U^\top \mathbf{b}$ ;  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ ;  $Z_0 = H\mathbf{r}_0$ ;  $P_0 = \Pi Z_0$ ;
2 for  $i = 0, 1, \dots$ , convergence do
3    $Q_i = AP_i$ ;
4    $\Delta_i = Q_i^\top P_i$ ;  $\gamma_i = P_i^\top \mathbf{r}_i$ ;  $\alpha_i = \Delta_i^\dagger \gamma_i$ ;
5    $\mathbf{x}_{i+1} = \mathbf{x}_i + P_i \alpha_i$ ;
6    $\mathbf{r}_{i+1} = \mathbf{r}_i - Q_i \alpha_i$ ;
7    $t_i = \frac{\gamma_i^\top \alpha_i}{\mathbf{r}_{i+1}^\top H \mathbf{r}_{i+1}}$ ;
8   if  $t_i < \tau$  then
9      $Z_{i+1} = [H^1 \mathbf{r}_{i+1} \mid \dots \mid H^N \mathbf{r}_{i+1}]$ ;
10  else
11     $Z_{i+1} = H \mathbf{r}_{i+1}$ ;
12  end
13   $\Phi_{i,j} = Q_j^\top Z_{i+1}$ ;  $\beta_{i,j} = \Delta_j^\dagger \Phi_{i,j}$  for each  $j = 0, \dots, i$ ;
14   $P_{i+1} = \Pi Z_{i+1} - \sum_{j=0}^i P_j \beta_{i,j}$ ;
15 end
16 Return  $\mathbf{x}_{i+1}$ ;
```

Adaptive Multipreconditioning

$\leftarrow \tau$ -test
 $\leftarrow$ Concatenate the N vectors

Each time $t_i < \tau$, an $N \times N$ matrix $\Delta_i = P_i^\top A P_i$ must be inverted. Since P_i is the concatenation of contributions from the components H^s in the preconditioner, it is reasonable to expect that Δ_i be full-rank. If it is not, then Δ_i is only positive semi-definite and pseudo-inversion (denoted by $\Delta_i^\dagger$) is necessary. In any case, both γ_i and the columns in $\Phi_{i,j}$ are in $\text{range}(P_i^\top) = \text{range}(\Delta_i)$ so that the iteration is always well defined.

The presented algorithm is with full reorthogonalization (lines 13 and 14). It is known that for MPCG this is necessary but for PPCG it is not ($\beta_{i,j} = 0$ as soon as $j \neq i$) so some reorthogonalization steps may be skipped.

Remark 3.2.2. For clarity the size of the different variables is given (recall that n is the size of the global problem and the preconditioner has N components):

- $A, \Pi, H, H^1, \dots, H^N \in \mathbb{R}^{n \times n}$,
- $\mathbf{x}, \mathbf{x}_i, \mathbf{r}_i, \mathbf{b} \in \mathbb{R}^n$,
- $Z_{i+1}, P_{i+1}, Q_{i+1} \in \mathbb{R}^{n \times N}$ or $\mathbb{R}^n$ depending on the iteration,
- $\Delta_i \in \mathbb{R}^{N \times N}$ or $\mathbb{R}$ depending on the iteration,
- $\gamma_i, \alpha_i \in \mathbb{R}^N$ or $\mathbb{R}$ depending on the iteration number i ,

- $\Phi_{i,j}, \beta_{i,j} \in \mathbb{R}^{N \times N}$ or $\mathbb{R}^N$ or $\mathbb{R}^{1 \times N}$ or $\mathbb{R}$ depending on the iteration numbers i and j ,

Note that the quantities $P_i \alpha_i$ and $AP_i \alpha_i$ are always vectors in $\mathbb{R}^n$. Note also that the notation $\langle \cdot, \cdot \rangle$ is only used for computing the inner product between two vectors.

The usual PPCG properties hold for Algorithm 3

In Theorem 3.2.3, results similar to the usual properties of PPCG are proved for Algorithm 3.

Theorem 3.2.3. *Algorithm 3 satisfies the five properties:*

1. The exact solution is achieved after at most $n-m$ iterations (m is the dimension of the deflated subspace).
2. $\|\mathbf{x} - \mathbf{x}_i\|_A = \min \{ \|\mathbf{x} - \hat{\mathbf{x}}\|_A; \hat{\mathbf{x}} \in \mathbf{x}_{00} + \text{range}(U) + \sum_{j=0}^{i-1} \text{range}(P_j) \}$.
3. Blocks of search directions are pairwise A -orthogonal: $P_j^\top AP_i = \mathbf{0}$ ($i \neq j$).
4. Residuals are ℓ_2 -orthogonal to previous search directions: $P_j^\top \mathbf{r}_i = \mathbf{0}$ ($i > j$).
5. Residuals are pairwise H -orthogonal: $\langle H\mathbf{r}_j, \mathbf{r}_i \rangle = 0$ ($i \neq j$).

Convergence Results

The following theorems hold.

Theorem 3.2.4. *If the exact solution has not yet been achieved at iteration $i-1$ of Algorithm 3 and $t_{i-1} \geq \tau$ then the relative error is bounded by*

$$\frac{\|\mathbf{x} - \mathbf{x}_i\|_A}{\|\mathbf{x} - \mathbf{x}_{i-1}\|_A} \leq \left(\frac{1}{1 + \lambda_{\min} \tau} \right)^{1/2}.$$

(Recall that $\tau \in \mathbb{R}^+$ is the threshold chosen by the user and $\lambda_{\min}$ is a lower bound for the smallest eigenvalue of the preconditioned operator HA .)

Proof. The proof follows the same lines of reasoning as the results in Section 3.2.1 with handling of blocks. $\square$

The following corollary provides some insight into how to choose the parameter τ by choosing a targeted contraction factor ρ . This is related to the discussion that yielded (3.4).

Corollary 3.2.5. *Let $0 < \rho < 1$. Under the assumptions of Theorem 3.2.4 it holds that*

$$\frac{\|\mathbf{x} - \mathbf{x}_i\|_A}{\|\mathbf{x} - \mathbf{x}_{i-1}\|_A} \leq \rho \text{ if } \tau = \frac{1 - \rho^2}{\lambda_{\min} \rho^2}.$$

The next theorem states that if the problem is well-conditioned, then no adaptive step will be performed and the additional cost compared to PPCG is just the cost of performing the τ -test (two inner products and one scalar division) which is negligible.

Theorem 3.2.6. *If all eigenvalues of the preconditioned operator HA are smaller than $1/\tau$ then the result of the τ -test is $t_i \geq \tau$ at each iteration. Consequently, Algorithm 3 performs the usual PPCG iterations.*

Proof. Begin with $\mathbf{r}_i = \mathbf{r}_{i-1} - Q_{i-1}\alpha_{i-1}$ and take the inner product by $H\mathbf{r}_i$:

$$\langle H\mathbf{r}_i, \mathbf{r}_i \rangle = -\langle H\mathbf{r}_i, AP_{i-1}\alpha_{i-1} \rangle \text{ (by Property 5 in Theorem 3.2.3).}$$

An application of the Cauchy-Schwarz inequality in the A -norm gives

$$\langle H\mathbf{r}_i, \mathbf{r}_i \rangle \leq \langle H\mathbf{r}_i, AH\mathbf{r}_i \rangle^{1/2} \langle P_{i-1}\alpha_{i-1}, AP_{i-1}\alpha_{i-1} \rangle^{1/2},$$

or equivalently

$$\frac{\langle H\mathbf{r}_i, \mathbf{r}_i \rangle}{\langle H\mathbf{r}_i, AH\mathbf{r}_i \rangle} \leq \frac{\langle P_{i-1}\alpha_{i-1}, AP_{i-1}\alpha_{i-1} \rangle}{\langle H\mathbf{r}_i, \mathbf{r}_i \rangle}. \quad (3.5)$$

By assumption all eigenvalues of HA are smaller than $1/\tau$ so $\frac{\langle H\mathbf{r}_i, \mathbf{r}_i \rangle}{\langle H\mathbf{r}_i, AH\mathbf{r}_i \rangle} \geq \tau$ and the τ -test returns $t_i \geq \tau$. $\square$

Corollary 3.2.7. *If the τ -test returns $t_{i-1} < \tau$ then $\|H\mathbf{r}_i\|_{H^{-1}}^2 < \tau \|H\mathbf{r}_i\|_A^2$. This implies that there is at least one eigenvalue of HA larger than $1/\tau$. Moreover, it holds that $\langle H\mathbf{r}_i, A\mathbf{v} \rangle \neq 0$ where $\mathbf{v}$ is an eigenvector corresponding to that eigenvalue. This explains why it makes sense to augment the minimization space with the components of $H\mathbf{r}_i$.*

AMPCG Algorithm with local τ -test (Algorithm 4)

One further assumption is made:

(A6) the operator A is a sum of N contributions : $A = \sum_{s=1}^N A^s$, with each A^s spsd.

In cases where the number N of components in the preconditioner is very large, it may occur that the cost of factorizing Δ_i becomes excessive. In this case, the *global* τ -test in Algorithm 3 (line 8) is switched out for N *local* tests in order to deflate only the *local* components that are problematic. This is presented in Algorithm 4 and the adaptive steps are between lines 8 and 13.

The remarks from Section 3.2.2 about the choice of τ , factorization of Δ_i and full reorthogonalization also apply here. Additionally, if at a given iteration $H^s \mathbf{r}_{i+1} = \mathbf{0}$, then t_i^s is not defined, but it cannot occur that $H^s \mathbf{r}_{i+1} = \mathbf{0}$ for all values of s simultaneously unless convergence is achieved ($\mathbf{r}_{i+1} = \mathbf{0}$).

If the local τ -tests return $t_i^s < \tau$ for every $s = 1, \dots, N$ at a given iteration, then it holds that $Z_{i+1} = [H\mathbf{r}_{i+1} | H^1 \mathbf{r}_{i+1} | \dots | H^N \mathbf{r}_{i+1}]$ and the first column is obviously linearly redundant so any efficient implementation of the algorithm would delete it immediately.

It makes sense to look for additional search directions locally if the preconditioner is constructed as a sum of approximate inverses H^s of the components A^s in the operator. This is illustrated with an example of substructuring DD solvers in the next section.

The theoretical properties of Algorithm 4 are stated in the following theorem and are similar to the ones of Algorithm 3.

Theorem 3.2.8. *The five properties proved in Theorem 3.2.3 hold. The convergence bound in Theorem 3.2.4 holds if the local τ -tests return $t_i^s \geq \tau$ for every $s = 1, \dots, N$ at a given iteration i .*

3.2.3 AMPCG optimized for BDD

In this section, the focus is on one of the so called substructuring methods: and particularly on Balancing Domain Decomposition, or BDD [126].

Algorithm 4 : AMPCG with local τ -test for $(\sum_{s=1}^N A^s) \mathbf{x} = \mathbf{b}$ preconditioned by $(\sum_{s=1}^N H^s)$ and deflated by Π for initial guess $\mathbf{x}_0$. $\tau \in \mathbb{R}^+$: chosen by user.

```

1  $\mathbf{x}_0 = \Pi \mathbf{x}_0 + U(U^\top AU)^{-1}U^\top \mathbf{b}$ ;  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ ;  $Z_0 = H\mathbf{r}_0$ ;  $P_0 = \Pi Z_0$ ;
2 for  $i = 0, 1, \dots$ , convergence do
3    $Q_i = AP_i$ ;
4    $\Delta_i = Q_i^\top P_i$ ;  $\gamma_i = P_i^\top \mathbf{r}_i$ ;  $\alpha_i = \Delta_i^\dagger \gamma_i$ ;
5    $\mathbf{x}_{i+1} = \mathbf{x}_i + P_i \alpha_i$ ;
6    $\mathbf{r}_{i+1} = \mathbf{r}_i - Q_i \alpha_i$ ; (Locally) Adaptive Multipreconditioning
7    $Z_{i+1} = H\mathbf{r}_{i+1}$ ;  $\leftarrow$  initialize  $Z_{i+1}$ 
8   for  $s = 1, \dots, N$  do
9      $t_i^s = \frac{\langle P_i \alpha_i, A^s P_i \alpha_i \rangle}{\mathbf{r}_{i+1}^\top H^s \mathbf{r}_{i+1}}$ ;
10    if  $t_i^s < \tau$  then  $\leftarrow$  local  $\tau$ -test
11       $Z_{i+1} = [Z_{i+1} \mid H^s \mathbf{r}_{i+1}]$ ;  $\leftarrow$  concatenate  $Z_{i+1}$  and  $H^s \mathbf{r}_{i+1}$ 
12    end
13  end
14   $\Phi_{i,j} = Q_j^\top Z_{i+1}$ ;  $\beta_{i,j} = \Delta_j^\dagger \Phi_{i,j}$  for each  $j = 0, \dots, i$ ;
15   $P_{i+1} = \Pi Z_{i+1} - \sum_{j=0}^i P_j \beta_{i,j}$ ;
16 end
17 Return  $\mathbf{x}_{i+1}$ ;

```

Notation and Introduction of the BDD formulation

Assume that a linear system $K\mathbf{u} = \mathbf{f}$ is given for an spd matrix K which corresponds to the finite element discretization of a partial differential equation (PDE) posed in an open subset Ω of $\mathbb{R}^2$ or $\mathbb{R}^3$. Further assume that Ω has been partitioned into N , non-overlapping and mesh conforming, subdomains Ω^s and that Γ is the set of boundaries between subdomains:

$$\bar{\Omega} = \bigcup_{s=1,\dots,N} \bar{\Omega}^s; \quad \Omega^s \cap \Omega^t = \emptyset \text{ for all } s \neq t; \quad \Gamma = \bigcup_{s=1,\dots,N} \partial\Omega^s \setminus \partial\Omega.$$

The original linear system admits the following block form

$$\begin{pmatrix} K_{\Gamma\Gamma} & K_{\Gamma I} \\ K_{I\Gamma} & K_{II} \end{pmatrix} \begin{pmatrix} \mathbf{u}_\Gamma \\ \mathbf{u}_I \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\Gamma \\ \mathbf{f}_I \end{pmatrix} \Leftrightarrow \begin{cases} K_{\Gamma\Gamma}\mathbf{u}_\Gamma + K_{\Gamma I}\mathbf{u}_I = \mathbf{f}_\Gamma \\ K_{I\Gamma}\mathbf{u}_\Gamma + K_{II}\mathbf{u}_I = \mathbf{f}_I \end{cases},$$

where the subscript $*_\Gamma$ denotes the restriction to the set of degrees of freedom on Γ and $*_I$ to the remaining unknowns. From the second line it can be deduced that $\mathbf{u}_I = K_{II}^{-1}(\mathbf{f}_I - K_{I\Gamma}\mathbf{u}_\Gamma)$ and, by injecting this into the first line, the problem is reduced to one on the interfaces between subdomains: Find $\mathbf{u}_\Gamma \in \mathbb{R}^n$ ($n := \#(\Gamma)$) such that

$$A\mathbf{u}_\Gamma = \mathbf{b}, \text{ where } A := K_{\Gamma\Gamma} - K_{\Gamma I}K_{II}^{-1}K_{I\Gamma} \text{ and } \mathbf{b} := \mathbf{f}_\Gamma - K_{\Gamma I}K_{II}^{-1}\mathbf{f}_I. \quad (3.6)$$

This is the linear system that is solved in the BDD method. The result is an approximation $\mathbf{u}_\Gamma$ of the solution on Γ and the remaining part of the solution is computed as $\mathbf{u}_I = K_{II}^{-1}(\mathbf{f}_I - K_{I\Gamma}\mathbf{u}_\Gamma)$. To understand why BDD is ideal for parallel computing and fits the framework for adaptive multipreconditioning, (3.6)

is rewritten into a form that makes the local contributions apparent. First, let K^s be the local matrices corresponding to the discretization of the same PDE but restricted to each subdomain $\bar{\Omega}^s$ and write them in block formulation as

$$K^s = \begin{pmatrix} K_{\Gamma^s \Gamma^s}^s & K_{\Gamma^s \Gamma^s}^s \\ K_{\Gamma^s \Gamma^s}^s & K_{\Gamma^s \Gamma^s}^s \end{pmatrix} \text{ where } \begin{cases} *_{\Gamma^s}: \text{degrees of freedom on } \Gamma \cap \partial\Omega^s, \\ *_{\Gamma^s}: \text{remaining degrees of freedom in } \bar{\Omega}^s. \end{cases}$$

Then, define the local Schur complements $\mathbf{S}^s := K_{\Gamma^s \Gamma^s}^s - K_{\Gamma^s \Gamma^s}^s (K_{\Gamma^s \Gamma^s}^s)^{-1} K_{\Gamma^s \Gamma^s}^s$. Finally, these can be assembled into the BDD operator already defined in (3.6). Indeed it holds that

$$A = \sum_{s=1}^N A^s, \text{ where for all } s = 1, \dots, N : \quad A^s := R^{s\top} \mathbf{S}^s R^s \quad (3.7)$$

and $R^s \in \mathbb{R}^{(\Gamma^s) \times n}$ is the boolean matrix that, given a vector in $\mathbb{R}^n$, selects the entries in Γ^s . The fact that A is a sum of local contributions has now been made apparent and the preconditioner exploits this since it is chosen as

$$H := \sum_{s=1}^N H^s, \text{ where for all } s = 1, \dots, N : \quad H^s := R^{s\top} D^s \mathbf{S}^{s\top} D^s R^s, \quad (3.8)$$

and $\{D^s\}_{s=1, \dots, N}$ is a family of positive definite diagonal matrices that form a partition of unity (i.e., they satisfy $\sum_{s=1}^N R^{s\top} D^s R^s = I$). The deflation, or *coarse*, space is chosen as:

$$\text{range}(U) = \sum_{s=1}^N R^{s\top} D^s \text{Ker}(\mathbf{S}^s). \quad (3.9)$$

This offers the double advantage of making all applications of ΠH and $H \Pi^\top$ uniquely defined and of improving convergence significantly. An alternative approach is the Balancing Domain Decomposition by Constraints (or BDDC) solver [59].

AMPCG for BDD

There were six assumptions in Sections 3.2.1 and 3.2.2. With the notation introduced in (3.7), (3.8) and (3.9) all six of these assumptions hold: U is a full rank matrix, A and H are symmetric positive definite matrices [126] assembled as the sum of N symmetric positive semidefinite matrices and all eigenvalues of the preconditioned operator HA are larger than $\lambda_{\min} = 1$ [126, 159]. As a result, the two new AMPCG algorithms (namely, Algorithm 3 with the global τ -test, and Algorithm 4 with the local τ -test) can straightforwardly be applied to the BDD linear system (3.6). Moreover, the theoretical results in Theorems 3.2.3, 3.2.4, 3.2.6, 3.2.8 and Corollary 3.2.7 hold.

It is well known that the majority of time and resource consuming operations in a BDD algorithm are the local solves required by any application of the operator A and the preconditioner H (specifically, the Dirichlet solves $K_{\Gamma^s \Gamma^s}^{s-1}$ in $\mathbf{S}^s$ and the Neumann solves $K^{s\top}$ used to compute an application of $\mathbf{S}^{s\top}$). We observe that the cost of preconditioning is the same in one iteration of AMPCG as in an iteration of PPCG for BDD. However, in iterations where multiple search directions are selected, additional applications of A are needed if the original formulation of the algorithms is implemented (P_i is dense after orthogonalization and projection). Since the interest is in high performance computing, optimized versions of the algorithms are proposed for BDD which fix this. In exact arithmetic the modifications make no difference to the sequence of approximate solutions but they save a lot of computational time. This is not presented here.

3.2.4 Numerical Results

The results in this section illustrate the behaviour of AMPCG. They were obtained with the software libraries FreeFem++ [91] and GNU Octave [63]. Further performance results will be presented next in Section 3.2.5. The linear elasticity equations posed in $\Omega = [0, 1]^2$ with mixed boundary conditions are considered. A solution $\mathbf{u} = (u_1, u_2)^\top \in H^1(\Omega)^2$ is sought such that

$$\begin{cases} -\operatorname{div}(\sigma(\mathbf{u})) = (0, 10)^\top, & \text{in } \Omega, \\ \mathbf{u} = (0, 0)^\top, & \text{on } \{(x, y) \in \partial\Omega : x = 0\}, \\ \sigma(\mathbf{u}) \cdot \mathbf{n} = 0, & \text{on the rest of } \partial\Omega \text{ (with } \mathbf{n} \text{ denoting the outward normal)}. \end{cases} \quad (3.10)$$

The stress tensor $\sigma(\mathbf{u})$ is defined by $\sigma_{ij}(\mathbf{u}) = 2\mu\varepsilon_{ij}(\mathbf{u}) + \lambda\delta_{ij}\operatorname{div}(\mathbf{u})$ for $i, j = 1, 2$ where $\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$, δ_{ij} is the Kronecker symbol and the Lamé coefficients are functions of Young's modulus E and Poisson's ratio ν : $\mu = \frac{E}{2(1+\nu)}$, $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$. It is well known (see, e.g., [141]) that the solution of (3.10) in a heterogeneous medium is challenging due to ill-conditioning. In the following experiments, after setting $\nu = 0.4$ in all of the domain for all test cases, a checkerboard distribution is chosen for E (see Figure 3.1c) where the two values are $E_1 = 10^7$ and, unless specified otherwise, $E_2 = 10^{12}$.

The computational domain is discretized by a uniform mesh and the boundary value problem is solved numerically with standard piecewise linear (P_1) Lagrange finite elements. The resulting linear system is then rewritten in BDD formulation. To this end, the computational domain is partitioned into N subdomains (N is given). The partition is either regular (into squares of size $1/\sqrt{N} \times 1/\sqrt{N}$) or generated automatically by the graph partitioner Metis [108] (see Figures 3.1a and 3.1b). The mesh parameter is inversely proportional to the number of subdomains: $h = 1/(10N)$. The material in each subdomain is either homogeneous (if the partition is regular as in Section 3.2.4) or heterogeneous (if the partition is computed by Metis as in Section 3.2.4).

Five algorithms for solving the linear system are compared:

AMPCG-Glob Algorithm 3: AMPCG for BDD with Global τ -test

AMPCG-Loc Algorithm 4: AMPCG for BDD with Local τ -test

MPCG Algorithm 3 with $\tau = \infty$ such that all iterations are multipreconditioned BDD iterations,

PPCG Algorithm 2 where U is full rank and $\operatorname{range}(U) = \sum_{s=1}^N R^s{}^\top D^s \operatorname{Ker}(\mathbf{S}^s)$ as per (3.9),

GenEO Algorithm 2 where U is the GenEO coarse space for BDD from [19, Theorem 2.11 and Remark 2.9]. In detail, $\operatorname{range}(U) = \operatorname{span}\{R^s{}^\top \mathbf{y}^s; s = 1, \dots, N \text{ and } \lambda^s \leq \tau\}$ where $(\mathbf{y}^s, \lambda^s)$ solve the generalized eigenvalue problem $(D^{s-1} \mathbf{S}^s D^{s-1}) \mathbf{y}^s = \lambda^s (R^s A R^s{}^\top) \mathbf{y}^s$. The eigenvalues of the preconditioned operator are in $[1, \mathcal{N}/\tau]$ where $\mathcal{N}$ is the usual coloring constant (Definition 2.2.2).

The stopping criterion is always that the error $\|\mathbf{x}_i - \mathbf{x}\|_A$ be smaller than $10^{-6} \|\mathbf{x}\|_A$. This allows us to compare all algorithms fairly. Finally, for all three adaptive methods (both AMPCG methods, cf. Algorithms 3 and 4, and GenEO), $\tau = 0.1$ is chosen except in one explicitly stated test, where τ is varied. According to Corollary 3.2.5 (with $\lambda_{\min} = 1$ for BDD) this guarantees that, at iterations where only one search direction is used, the error has decreased by at least a factor $\rho = 0.95$.

We examine three quantities with respect to the iteration count: the relative error $\|\mathbf{x}_i - \mathbf{x}\|_A / \|\mathbf{x}\|_A$, the dimension of the minimization space (see Property 2 in Theorem 3.2.3), and the number of local solves (number of applications of $\mathbf{S}^s$ or $\mathbf{S}^{s^\dagger}$). In the case of GenEO, results on the number of local solves are never included since this would also require counting the number of iterations needed for the eigensolver to converge. Optimizing this is not the purpose here. We leave the study in terms of CPU time for Section 3.2.5.

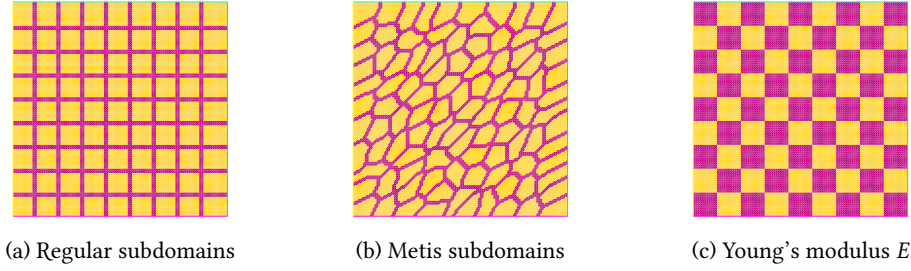


Figure 3.1: Geometry of the test problem. There are 19800 degrees of freedom in the original problem and 3056 (respectively, 3458) in its BDD formulation on the regular (respectively, Metis) partition into 81 subdomains.

Regular partition: Homogeneous Subdomains

The partition is into $N = 81$ subdomains (see Figure 3.1a) in such a way that the material parameters are constant in each subdomain. The size of the linear system is $n = 3056$. We consider two choices for the partition of unity matrices D^s in the preconditioner. Either D^s is proportional to the diagonal values of the local matrix K^s (k-scaling) [127, 145, 118, 141] or it is equal to the inverse of the multiplicity of the degrees of freedom (multiplicity scaling). These were previously defined in (2.13) and (2.14)

Figure 3.2a shows the results when k-scaling is used. Theoretical results [126, 118, 159] guarantee that PPCG converges fast in this case. We observe that this is indeed the case and that the adaptive algorithms detect that they do not need to do any extra work (or very little in the case of AMPCG with the local τ -test which selects just 4 extra search directions). The MPCG algorithm converges even faster but at the cost of about twice as many local solves.

When multiplicity scaling is used, see Figure 3.2b, convergence of PPCG is not as good. Both of the AMPCG algorithms adapt by selecting significantly larger minimization spaces (626 and 783 versus 268). This allows for a reduction in the number of iterations from over 50 to under 10 and the number of local solves decreases by about a factor 2 from 8586 to 4302 and 4176. The algorithm with the global τ -test augments the minimization space at each iteration so in this case it is the same as the MPCG algorithm. The coarse space computed by GenEO was prohibitively large so GenEO was not included, a phenomenon already encountered and explained in the comments of Figure 2.3.

As predicted, in each case the AMPCG algorithms adapt correctly to the difficulty of the problem: for the harder problem, convergence is (almost) as good as with MPCG, and for the easier problem (almost) no extra work is performed compared to PPCG. Next a more complex geometry is considered for which neither MPCG or PPCG are expected to be optimal.

Metis partition: Heterogeneous Subdomains

We partition the domain into $N = 81$ subdomains with Metis (see Figure 3.1b). This way, the partition into subdomains is no longer aligned with the heterogeneities and the domains are non-regular. Subdomains with these properties are known to cause problems for BDD even with k-scaling (which is used here). The size of the linear system for this problem is $n = 3458$.

We first plot the results for all test cases in Figure 3.3a, and then without PPCG in Figure 3.3b in order to zoom in on the other methods. The convergence history of PPCG (first plot in Figure 3.3a) illustrates the typical behaviour of conjugate gradient algorithms for ill-conditioned systems. For all other methods the stagnation is completely eliminated (GenEO) or significantly decreased. This confirms that the problematic eigenvectors are well-handled by augmenting the minimization space with local

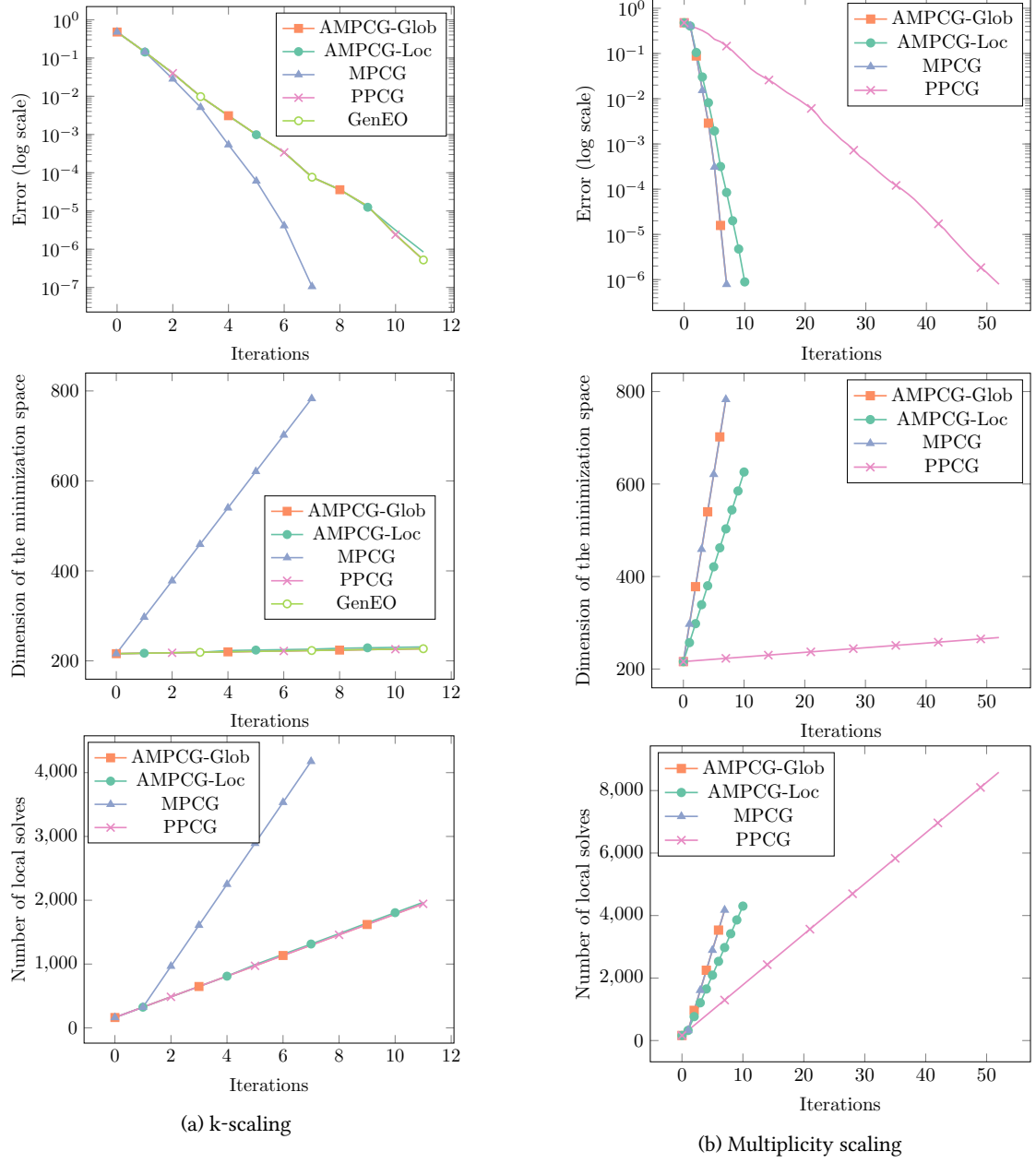


Figure 3.2: Regular partition: the subdomains are homogeneous.

quantities. Observe that with AMPCG the minimization space is augmented only during the first few iterations. Then they behave as fast converging PPCG methods and in the end they require roughly 4 times fewer local solves than PPCG. For this harder test case, MPCG also converges faster and with fewer local solves than PPCG.

Influence of τ . The threshold τ plays a very important role in AMPCG. The global ($t_i < \tau$) or local ($t_i^s < \tau$) τ -tests determine whether to perform an iteration with one or several search directions, respectively. In Corollary 3.2.5, it is shown that τ can be related to a targeted contraction factor ρ that has a more natural interpretation. We have performed the simulation with both AMPCG algorithms for the same test problem as previously but with values of τ ranging between 10^{-4} and 10. In Figure 3.4, the contraction factor ρ given by Corollary 3.2.5 is plotted, as well as the iteration count, dimension of the minimization space, and number of local solves required by each algorithm (all as functions of τ). For the method with the global τ -test, there is a whole range of values ($0.06 \leq \tau \leq 0.4$ or equivalently $0.846 \leq \rho \leq 0.997$) which is optimal in the sense of minimizing the number of local solves. This illustrates that the method is robust with respect to variations in τ . In fact, for all values of τ considered, the AMPCG algorithms outperforms both non-adaptive algorithms: PPCG (22842 local solves) and MPCG (8360 local solves). One (heuristic) explanation as to why there are several *good* choices for τ is that there are two regimes in the convergence history of PPCG (first plot in Figure 3.3a): a long stagnation during the first 100 or so iterations followed by fast convergence once the large isolated eigenvalues of the preconditioned BDD operator have been well approximated by the Ritz values [161]. The transition between both regimes is quite abrupt and that is why the best choice of τ is not unique.

For the local version of the algorithm, the minimal number of local solves is 4743 instead of 5212 with the global τ -test. As expected, the minimization space is smaller than with the global version of the algorithm but sometimes this leads to a larger iteration count and more local solves overall. In the end, the local version of the algorithm appears to be better for the best choice of τ but also more sensitive to the choice of τ .

Influence of E_2/E_1 and N . In Table 3.1, the influence of other problem parameters is presented for all methods applied to the same testcase with $\tau = 0.1$. First, details on the number of iterations and local solves needed for convergence when the heterogeneity E_2/E_1 varies (both for k-scaling and multiplicity scaling) are given. Then the ratio is set back to $E_2/E_1 = 10^5$ and the number N of Metis subdomains is varied. Note that the number of squares in the checkerboard is also N and the mesh parameter is $h = 1/(10N)$. The last line of each table gives the maximal dimension of the coarse space for that column. As a general rule, only PPCG suffers in terms of number of iterations when the problems become harder. For all methods, the number of local solves becomes larger as the problems become harder but the impact is a lot less significant for the two AMPCG algorithms illustrating that they adapt well to more difficult problems. One final argument in favor of the AMPCG algorithms is that block operations are often proportionally much less expensive than single vector operations because the computation time is driven by the memory access. Note that this point was previously made in [3] for MPCG.

3.2.5 Parallel results: Adaptive Multipreconditioning of FETI [2]

The AMPCG algorithm has been implemented at ONERA and Safran Tech, the Research & Technology Center of the aerospace, defence and security corporation Safran. This particular implementation is in the finite element suite Z-Set 8.6 [169], a simulation code that is developed by ONERA, Safran Tech and the LMT Cachan (now a part of Laboratoire de Mécanique Paris-Saclay). Together with Christophe Bovet, Pierre Gosselet and Augustin Parret Fréaud, in [2], we have used this code to perform a scalability and robustness assessment of AMPCG on industrial test cases.

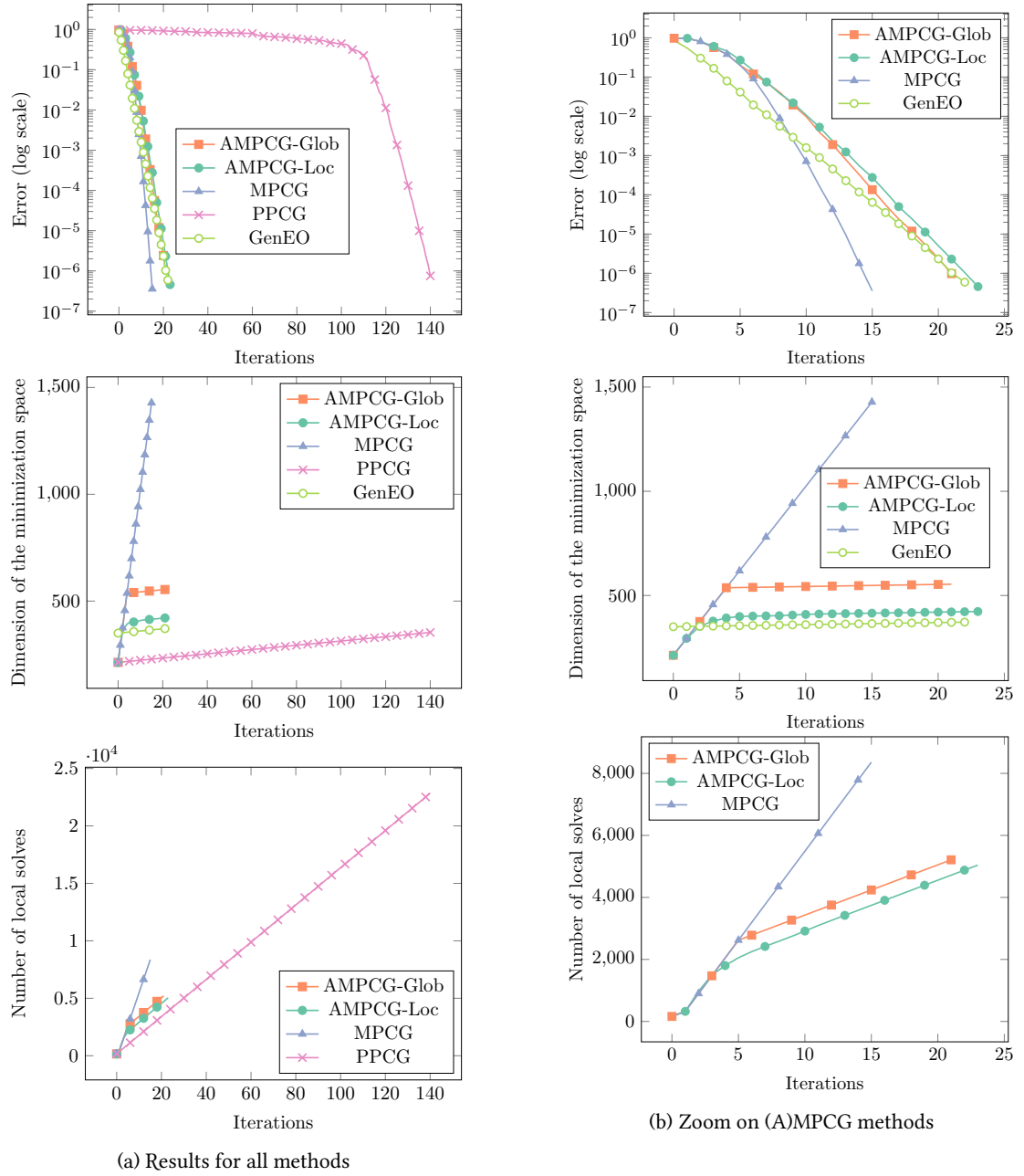
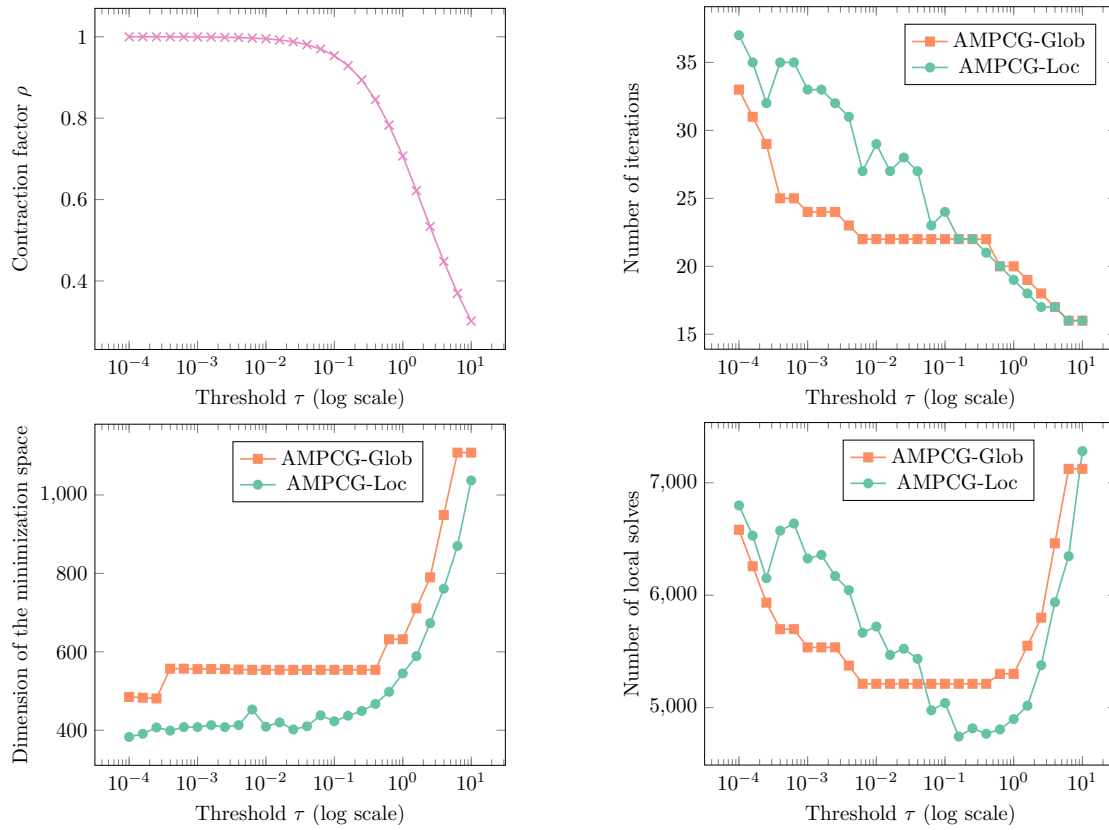


Figure 3.3: Metis Partition : the subdomains are heterogeneous.

Figure 3.4: Influence of the threshold τ on AMPCG.

Variable heterogeneity for $N = 81$ (k-scaling) :									
E_2/E_1	AMPCG-Glob		AMPCG-Loc		MPCG		PPCG		GenEO
	it	solves	it	solves	it	solves	it	solves	it
1	26	4624	25	4602	14	7212	36	5832	23
10	26	5036	28	5213	14	7212	44	7128	23
1×10^2	30	6096	25	5164	15	7786	76	12 312	21
1×10^3	23	5374	25	5133	16	8360	126	20 412	22
1×10^4	22	5212	25	5176	16	8360	139	22 518	22
1×10^5	22	5212	24	5041	16	8360	141	22 842	23
	$dim < 554$		$dim < 423$		$dim < 1428$		$dim < 353$		$dim < 372$

Variable heterogeneity for $N = 81$ (multiplicity scaling) :									
E_2/E_1	AMPCG-Glob		AMPCG-Loc		MPCG		PPCG		GenEO
	it	solves	it	solves	it	solves	it	solves	it
1	30	5272	30	5626	20	10 656	35	5670	23
10	32	6832	30	5941	21	11 230	51	8262	23
1×10^2	39	9202	34	8276	24	12 952	100	16 200	23
1×10^3	34	11 688	34	8890	24	12 952	204	33 048	22
1×10^4	31	11 202	34	8872	25	13 526	299	48 438	23
1×10^5	33	11 114	35	9089	25	13 526	336	54 432	23
	$dim < 1365$		$dim < 808$		$dim < 2157$		$dim < 548$		$dim < 662$

Variable number of subdomains for $E_2/E_1 = 10^5$ (k-scaling) :									
N	AMPCG-Glob		AMPCG-Loc		MPCG		PPCG		GenEO
	it	solves	it	solves	it	solves	it	solves	it
25	20	1784	22	1447	17	2530	69	3450	20
36	24	2392	23	2150	16	3476	87	6264	20
49	20	3364	24	3146	16	4844	110	10 780	20
64	21	5264	24	4137	17	7006	152	19 456	20
	$dim < 693$		$dim < 379$		$dim < 1193$		$dim < 320$		$dim < 327$

Table 3.1: Comparison of the five methods for variable heterogeneity and number of Metis subdomains. it: number of iterations. solves: number of local solves. dim: dimension of minimization space.

These runs were performed on the *Cobalt* supercomputer, which is a 1.299-petaflop Bull system at the french Computing Center for Research and Technology (CCRT, <http://www-ccrt.cea.fr>, ranked 75 in the November 2016 TOP500 list). Cobalt is composed of 1422 Intel Broadwell nodes inter-connected with an Infiniband EDR network and providing a total of 34 816 available cores. Seven cores have been assigned to each subdomain. In all configurations, the MUMPS solver [62, 25, 124, 23] in association with the BLAS library provided by the Intel MKL package has been used for local solves. Because of the high material contrast, the local rigid body modes of subdomains have been computed using the geometric-algebraic method [73]. For all computations, the partition of the domain has been obtained by the automatic graph partitioner Metis [108]. The solver is considered to have converged when the residual has decreased by a factor 10^6 . The DD method is FETI [74]. We do not present it here. For all useful purposes, AMPCG applied to FETI, or AMPFETI, is very similar to AMPCG (Algorithms 3 and 4) applied to BDD. AMPFETI is described in [2, Algorithm 7].

Out of many results in the article, only one is shown here: the woven Organic Matrix Composite material. This type of material is used in the automotive and aircraft industries to maximize strength while decreasing total mass. In the following, we consider a cuboid volume of composite material with an aspect ratio of $19 \times 19 \times 3$, made of a 26 yarn weave pattern inside a polymer matrix (see Figure 3.5). A typical use case is numerical homogenization, where a representative elementary volume is used to identify equivalent macroscopic properties by averaging the mechanical response under specific load conditions (e.g. periodic, mixed, uniform) [151]. In order to accurately describe the architecture at the mesoscopic scale, given the complexity of the underlying weave, this type of problem requires a large number of degrees of freedom (dofs). Furthermore, as the weave pattern complexity increases, its characteristic scale may be too high compared to that of the macroscopic structure such that no scale separation hypothesis can be used anymore. As a result, to get an accurate description of the medium, the mesoscopic model has to be used on the whole structure which, again, leads to a discretized problem with a high number of dofs.

The present weave pattern has been obtained by simulation [165] and a voxel-based finite element description is used to model the woven mesostructure [97]. The cuboid volume is discretized into a structured mesh of 30 223 800 regular eight-node fully integrated hexahedral elements (c3d8), which leads to a global size of 91 995 540 dofs. Depending on its spatial position, each element is either associated to one of the yarns, or to the matrix. All yarns share the same material behavior, chosen to be an orthotropic linear elastic law with a primary direction corresponding to the orientation of the yarn. The behavior of the matrix is chosen to be isotropic linear elastic. The maximal ratio between all Young's moduli related to the yarns and the matrix is $6.9 \cdot 10^3$. The cuboid is clamped on one side and subject to prescribed displacements corresponding to pure tension on the opposite side.

The problem, once written in the FETI formulation is solved by three algorithms:

FETI is the usual FETI method,

AMPCG-Glob is AMPCG with a global τ -test for FETI, where the threshold has been set to $\tau = 10^{-2}$.

AMPCG-Loc is AMPCG with a local τ -test for FETI, where the threshold has been set to $\tau = 10^{-2}$.

Three partitions of the domain are considered: $N = 175$; 250 and 500 subdomains (see Figure 3.5 for $N = 500$). The average number of dofs per subdomain is approximately 555 000; 390 000 and 200 000 dofs, respectively. The results are summarized in Table 3.2. Five specific timers have been defined:

- The *Total* timer is the Wall time (not CPU time). It is the actual time of the computation and takes into consideration the message passing latency and a perhaps perfectible synchronization of processes.
- The *Orthog.* timer represents the time spent orthogonalizing the search directions.

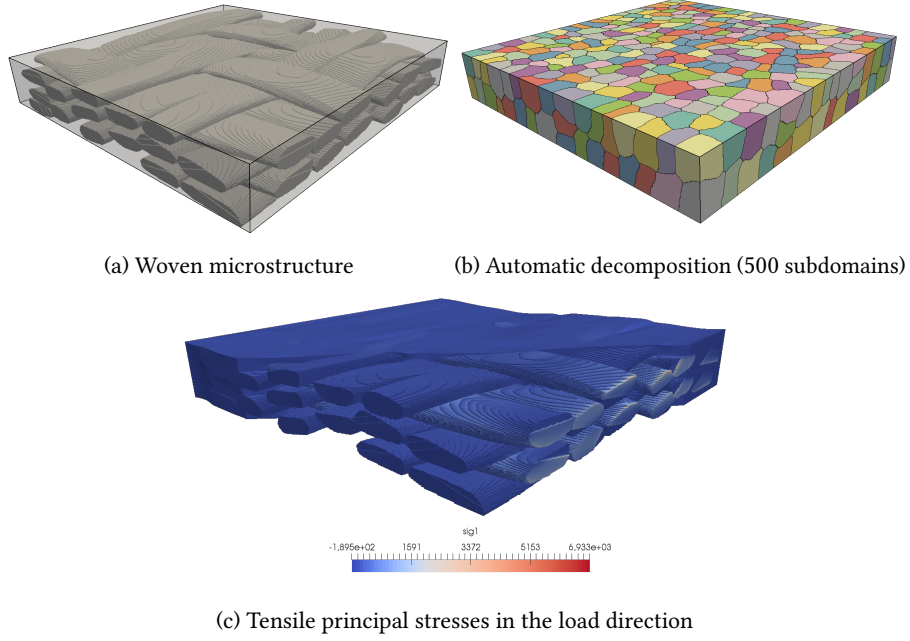


Figure 3.5: Representative elementary volume of woven composite (test-case for AMPFETI)

- The *Precond.* timer regroups the time spent computing the preconditioned residual and selecting search directions within the adaptive algorithms.
- The *Operator* timer is the time spent applying the FETI operator.
- The *Remaining* timer is the rest of the computational time (including initialization).

Table 3.2 shows that the time repartition of the different steps is uniformly decreasing as the number of subdomains increases, except for the orthogonalization step, which is logical since the size of the minimization space increases. Table 3.2 also shows that the size of the minimization space for AMPCG-Loc and AMPCG-Glob is about ten times the number of subdomains, for all splittings.

Figure 3.6 presents the global computational time of all three methods. It clearly emphasizes the gain provided by the AMPCG variants. Indeed, the AMPCG-Glob method reduces the time to solution by a factor of more than two in comparison with FETI. AMPCG-Loc shows slightly less efficiency compared to AMPCG-Glob, with a computational time about 20% larger.

Figure 3.7 shows the speedup of all three methods with respect to the 175 subdomain configuration, as defined by:

$$\text{Speedup}(N \text{ subdomains}) = \frac{\text{Wall time}(175 \text{ subdomains})}{\text{Wall time}(N \text{ subdomains})}.$$

We note that speedup curves for all three methods have nearly identical behavior, with a slight advantage for AMPCG-Glob.

As a conclusion, AMPCG variants have shown their ability to provide the solution to this engineering problem twice as fast as the FETI method, exhibiting a numerical behavior consistent with what has been previously observed. Although the local τ -test did not outperform the global τ -test on this example, it is easy to design a test problem for which it would (see e.g., Figure 3.8 and details in [2, Section 3.2.2]).

$N = 175$	Counters		Timers (s)			
	it	dim	Total	Precond.	Operator	Orthog.
FETI	371	371	1553.00	52.91%	29.77%	2.59%
AMPCG-Loc	103	1627	876.30	49.93%	15.97%	2.63%
AMPCG-Glob	120	1686	726.60	40.69%	24.49%	2.85%
$N = 250$	Counters		Timers (s)			
	it	dim	Total	Precond.	Operator	Orthog.
FETI	405	405	1109.00	50.97%	28.71%	3.52%
AMPCG-Loc	103	2026	635.10	51.74%	14.52%	3.49%
AMPCG-Glob	121	2113	499.10	40.88%	23.87%	3.83%
$N = 500$	Counters		Timers (s)			
	it	dim	Total	Precond.	Operator	Orthog.
FETI	544	544	721.60	49.40%	26.70%	7.75%
AMPCG-Loc	104	3907	408.20	43.73%	11.60%	12.25%
AMPCG-Glob	122	4613	319.52	30.75%	22.25%	12.64%

Table 3.2: Woven composite: summary of results. “it” is the number of iterations to reach convergence. “dim” is the dimension of the minimization space (total number of search directions). The percentages are to be understood as a fraction of the Total time. (AMPFETI)

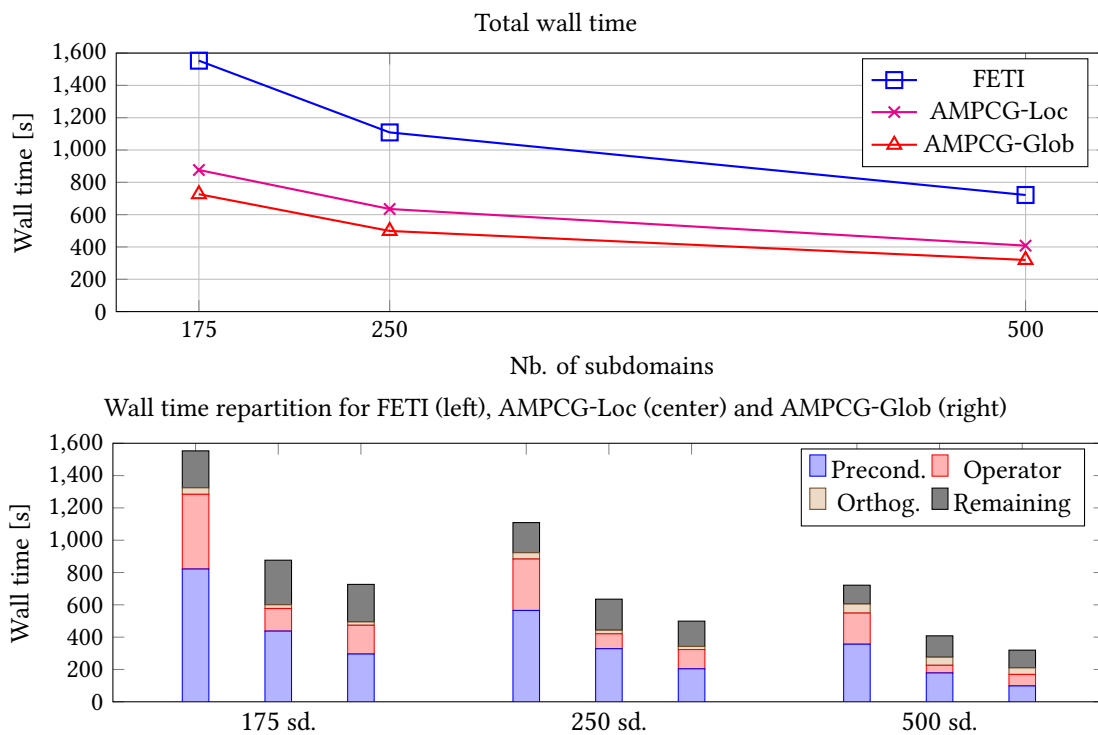


Figure 3.6: Woven composite: wall time vs. number of subdomains (AMPFETI)

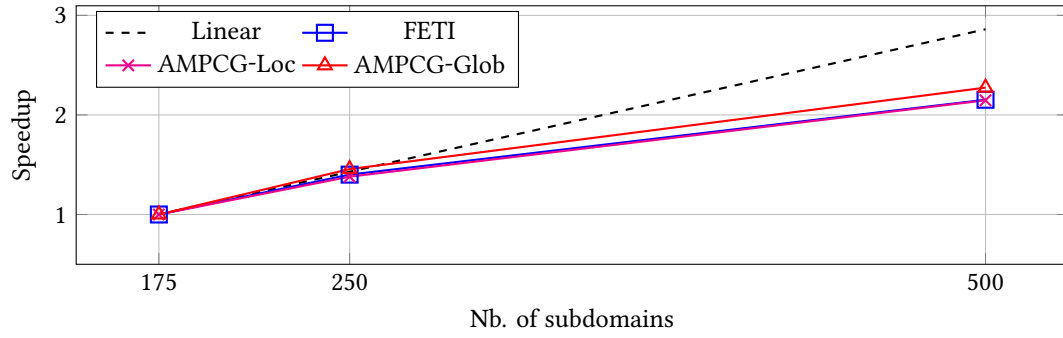


Figure 3.7: Woven composite: speedup vs. number of subdomains (AMPFETI)

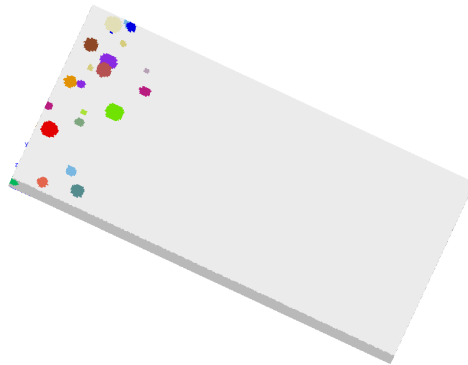


Figure 3.8: Example of a test case (taken from [2, Section 3.2.2]) for which AMPCG with the local τ -test is particularly well suited. The colors represent heterogeneities in the material that are gathered in a small part of the domain.

3.3 Non-symmetric and algebraic preconditioner: RAS [11]

It has been shown that AMPCG gives good results with FETI and BDD. However the theory behind AMPCG does not apply to Additive Schwarz. Indeed, Assumption (A4) (from Section 3.2.2) is that the smallest eigenvalue of the (globally) preconditioned operator is known. The test (called the τ -test), which chooses at each iteration which contributions should be kept, heavily relies on it. More precisely, the quantity that is examined by the τ -test can be related to a Rayleigh quotient and the vectors that are selected to form the next minimization space correspond to large frequencies of the (globally) preconditioned operator. These are exactly the ones that are known to slow down convergence of BDD and FETI. Moreover they are generated by the first few iterations of PCG [161]. These two reasons make BDD and FETI ideal for the AMPCG framework.

The question addressed here is whether an AMPCG algorithm can be developed for Additive Schwarz type preconditioners. The goal is to design an adaptive algorithm that is robust at a minimal cost. One great feature of Additive Schwarz is that it is algebraic (all the components in the preconditioner can be computed from the knowledge of matrix A) and we will aim to preserve this property. Since the short recurrence property is lost anyway in the MPCG setting, we will consider the more efficient [65] Restricted Additive Schwarz preconditioner (RAS) [49], in our numerical experiments instead of its symmetric counterpart the Additive Schwarz preconditioner (see [159]). RAS is a non-symmetric preconditioner but, provided that full recurrence is used, conjugate gradient based algorithms apply and still have nice properties (in particular the global minimization property).

3.3.1 Multipreconditioning RAS

To compute the RAS preconditioner, a non-overlapping partition of the degrees of freedom must first be chosen and then overlap must be added to each subdomain to get an overlapping partition. Denoting, for each $s = 1, \dots, N$, by $\tilde{R}^s$ (respectively, R^s) the restriction operator from $\llbracket 1, n \rrbracket$ into the s -th non-overlapping (respectively, overlapping) subdomain, the preconditioner is defined as:

$$H := \sum_{s=1}^N H^s \text{ with } H^s := \tilde{R}^{s\top} A^{s-1} R^s \text{ and } A^s := R^s A R^{s\top}.$$

In Algorithm 5, the MPCG iterations correspond to the non-adaptive algorithm, *i.e.* the condition in line 8 is not satisfied and N search directions are added to the minimization space at each iteration (namely the columns in Z_{i+1}).

Although RAS is a non-symmetric preconditioner the following properties hold:

- $\|\mathbf{x} - \mathbf{x}_i\|_A = \min \{\|\mathbf{x} - \mathbf{y}\|_A; \mathbf{y} \in \mathbf{x}_0 + \sum_{j=0}^{i-1} \text{range}(P_j)\},$
- $P_j^\top A P_i = \mathbf{0}$ ($i \neq j$), $\mathbf{r}_i^\top P_j = \mathbf{0}$ ($i > j$), and $\mathbf{r}_i^\top H \mathbf{r}_j = \mathbf{0}$ ($i > j$).

This can be proved easily following similar proofs as in textbook [149] or the AMPCG article [5]. The difference with the symmetric case is that the two last properties only hold for $i > j$ and not for every pair $i \neq j$.

Two methods to adaptively reduce the number of columns in Z_{i+1} are proposed. These fill the blank in line 9 of Algorithm 5. In Subsection 3.3.2, a τ -test is proposed that measures the relevance of every *candidate* $H^s \mathbf{r}_{i+1}$ and only keeps the most relevant contributions. In Subsection 3.3.3 it is proposed to form m coarser subdomains (which are *agglomerates* of the initial N subdomains) and *aggregate* the N *candidates* into only m search directions. Note that there is a definite connection with the field of multigrid (see [162, 56, 38] and many references therein) from where some vocabulary has been borrowed.

Algorithm 5 : AMPCG for $A\mathbf{x} = \mathbf{b}$. Preconditioners: $\{H^s\}_{s=1,\dots,N}$ (possibly non-symmetric).

Initial guess: $\mathbf{x}_0$.

```

1  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ ;  $Z_0 = [H^1\mathbf{r}_0 \dots | H^N\mathbf{r}_0]$ ;  $P_0 = Z_0$ ;
2 for  $i = 0, 1, \dots$ , convergence do
3    $Q_i = AP_i$ ;
4    $\Delta_i = Q_i^\top P_i$ ;  $\gamma_i = P_i^\top \mathbf{r}_i$ ;  $\alpha_i = \Delta_i^\dagger \gamma_i$ ;
5    $\mathbf{x}_{i+1} = \mathbf{x}_i + P_i \alpha_i$ ;
6    $\mathbf{r}_{i+1} = \mathbf{r}_i - Q_i \alpha_i$ ;
7    $Z_{i+1} = [H^1\mathbf{r}_{i+1} \dots | H^N\mathbf{r}_{i+1}]$ ; ← Generate  $N$  search directions.
8   if Adaptive Algorithm then
9     | Reduce number of columns in  $Z_{i+1}$  (see Sections 3.3.2 and 3.3.3);
10  end
11   $\Phi_{i,j} = Q_j^\top Z_{i+1}$ ;  $\beta_{i,j} = \Delta_j^\dagger \Phi_{i,j}$  for each  $j = 0, \dots, i$ ;
12   $P_{i+1} = Z_{i+1} - \sum_{j=0}^i P_j \beta_{i,j}$ ;
13 end
14 Return  $\mathbf{x}_{i+1}$ ;
```

3.3.2 First adaptive algorithm: select search directions with a τ -test

The proposed approach to select local contributions is different from the usual AMPCG τ -test. It is based on the knowledge that each approximate solution returned by PCG is the A -orthogonal projection of the exact solution $\mathbf{x}$ onto the minimization space. Here the property satisfied by the update between iterations i and $i + 1$ is

$$\|\mathbf{x} - \mathbf{x}_{i+1}\|_A = \min \{\|\mathbf{x} - \mathbf{y}\|_A; \mathbf{y} \in \mathbf{x}_i + \text{range}(P_i)\},$$

where P_i forms a basis of $\text{range}(Z_i)$ after orthogonalization against previous search spaces (line 12 in Algorithm 5).

For this reason, the τ -test that is proposed aims at evaluating, for each $s = 1, \dots, N$, the ratio between the norm of the error projected onto the global vector $H\mathbf{r}_{i+1}$ and the norm of the error projected onto the local candidate $H^s\mathbf{r}_{i+1}$. More precisely, we compute (with $\langle \cdot, \cdot \rangle$ denoting the ℓ_2 inner product)

$$t_i^s = \frac{\langle \mathbf{r}_{i+1}, H\mathbf{r}_{i+1} \rangle^2}{\langle H\mathbf{r}_{i+1}, AH\mathbf{r}_{i+1} \rangle} \times \frac{\langle H^s\mathbf{r}_{i+1}, AH^s\mathbf{r}_{i+1} \rangle}{\langle \mathbf{r}_{i+1}, H^s\mathbf{r}_{i+1} \rangle^2}. \quad (3.11)$$

This is indeed the announced quantity, since the squared A -norm of the A orthogonal projection of $\mathbf{x} - \mathbf{x}_i$ onto any vector $\mathbf{v}$ is

$$\|\mathbf{v}(\mathbf{v}^\top A\mathbf{v})^{-1}\mathbf{v}^\top \underbrace{A(\mathbf{x} - \mathbf{x}_i)}_{=\mathbf{r}_i}\|_A^2 = \frac{\langle \mathbf{r}_i, \mathbf{v} \rangle^2}{\langle \mathbf{v}, A\mathbf{v} \rangle}.$$

Then, given a threshold τ , the number of columns in Z_{i+1} is reduced by eliminating all the ones such that $t_i^s > \tau$. In order for the global preconditioned residual $\sum_{s=1}^N H^s\mathbf{r}_{i+1}$ to be included in the search space (as is always the case in PCG) it is added to Z_{i+1} in a separate column. This i -th minimization space $\text{range}(P_i)$ is of dimension anything between 1 and N

An important question is, as usual, how to choose τ . Considering that t_i^s measures the (inverse of the) impact of one of N contributions compared to the impact of the sum of the N contributions, it is quite

natural to choose $\tau \approx N$. In the next section, the behaviour of the adaptive algorithm with the τ -test is illustrated for values of τ ranging between $N/10$ and $10N$ with satisfactory results.

3.3.3 Second adaptive algorithm: aggregate search directions

Here, a completely different, and much more simple, way of reducing the number of vectors in Z_{i+1} is proposed. It is to choose a prescribed number $m \leq N$ of search directions per iteration and a partition of $\llbracket 1, N \rrbracket$ into m subsets. Then, the columns of Z_{i+1} that correspond to the same subset are simply replaced by their sum leaving m vectors. This is referred to as aggregation. The choice of m could be a certain fraction of N or the maximal dense matrix order that the user is prepared to factorize. In the next section, values ranging from $N/20$ to N are considered.

3.3.4 Numerical Results

The result in this section are computed with FreeFem++ [91] and GNU Octave [63]. The linear elasticity equations posed in $\Omega = [0, 1]^2$ with mixed boundary conditions are considered again. We search for $\mathbf{u} = (u_1, u_2)^\top \in H^1(\Omega)^2$ such that

$$\begin{cases} -\operatorname{div}(\sigma(\mathbf{u})) = (0, 0)^\top, & \text{in } \Omega, \\ \mathbf{u} = (1/2(y(1-y)), 0)^\top, & \text{on } \{(x, y) \in \partial\Omega : x = 0\}, \\ \mathbf{u} = (-1/2(y(1-y)), 0)^\top, & \text{on } \{(x, y) \in \partial\Omega : x = 1\}, \\ \sigma(\mathbf{u}) \cdot \mathbf{n} = 0, & \text{on the rest of } \partial\Omega \text{ (n: outward normal)}. \end{cases}$$

The stress tensor $\sigma(\mathbf{u})$ is defined by $\sigma_{ij}(\mathbf{u}) = 2\mu\epsilon_{ij}(\mathbf{u}) + \lambda\delta_{ij}\operatorname{div}(\mathbf{u})$ for $i, j = 1, 2$ where $\epsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$, δ_{ij} is the Kronecker symbol and the Lamé coefficients are functions of Young's modulus E and Poisson's ratio ν : $\mu = \frac{E}{2(1+\nu)}$, $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$. In all test cases ν is uniform and equal either to 0.4 (compressible test case) or 0.49999 (almost incompressible test case) while E varies between 10^6 and 10^{12} in a pattern presented in Figure 3.9–left. The geometries of the solutions are also presented in this figure.

The computational domain is discretized into a uniform mesh with step: $h = 1/60$ and partitioned into $N = 100$ subdomains by the automatic graph partitioner Metis [108]. One layer of overlap is added to each subdomain. In the compressible case, the system is discretized by piecewise second order polynomial (P_2) Lagrange finite elements. In the almost incompressible setting, it is known that the locking phenomenon occurs rendering the solution unreliable. To remedy this, the problem is rewritten in a mixed formulation with an additional unknown $p = \operatorname{div}(u)$, and then discretized. Although the $P_2 - P_0$ mixed finite element does not satisfy the discrete inf-sup condition it is often used in practice and chosen here. Finally, the pressure unknowns are eliminated by static condensation.

In both cases the problem has 28 798 degrees of freedom (once the degrees of freedom corresponding to Dirichlet boundary conditions have been eliminated). The initial guess is computed from a random vector $\mathbf{v}$ that is scaled to form $\mathbf{x}_0 = \frac{\mathbf{b}^\top \mathbf{v}}{\|\mathbf{v}\|_A}$, according to what is proposed in [156]. This guarantees that $\|\mathbf{x} - \mathbf{x}_0\|_A \leq \|\mathbf{x}\|_A$: the initial error is at most as large as it would be with a zero initial guess.

Table 3.3 reports the number of iterations needed to reduce the initial error $\|\mathbf{x} - \mathbf{x}_0\|_A$ by a factor 10^{-7} and the dimension of the minimization space that is constructed to do this ($\sum_i \operatorname{rank}(P_i)$). Note that, although they are presented in the same table, the compressible and incompressible test cases cannot be compared since they are simply not the same problem. Figures 3.10, 3.11, 3.12 and 3.13 show in more detail the convergence behaviour of each method.

The first point to be made is that MPCG does an excellent job at reducing the number of iterations. This can be observed by looking at the data for $m = 100 = N$ directions per iteration in Figures 3.12 and 3.13. The iteration counts are reduced from 889 to 60 and from over 999 to 56 compared to the classical

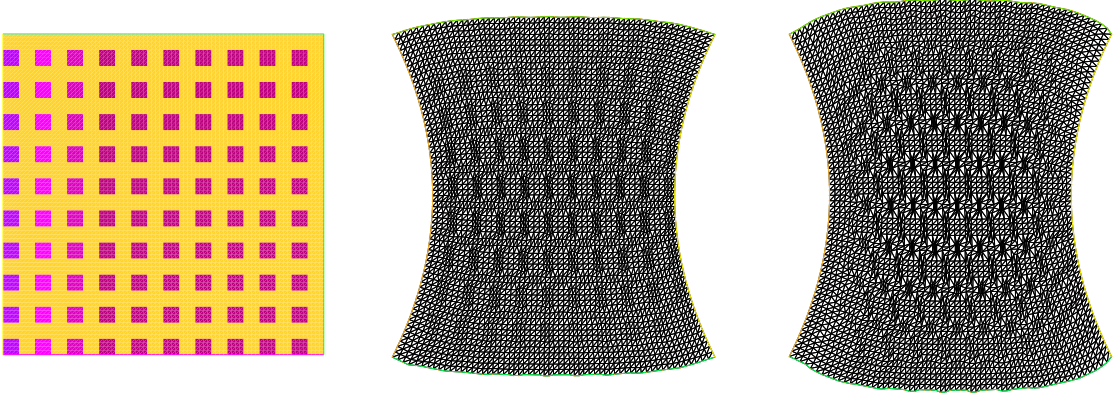


Figure 3.9: Test case setup (all three configurations are drawn to scale). Left: Young's modulus – $E = 10^6$ with square inclusions of larger E , up to 10^{12} . Middle: Solution for $\nu = 0.4$. Right: Solution for $\nu = 0.49999$.

PCG iterations ($m = 1$ direction per iteration). Secondly, the newly introduced adaptation steps seem to do their job since they ensure fast convergence with smaller minimization spaces. In particular, all of these adaptive methods converge in fewer than 512 iterations even for the incompressible case (for which the usual PCG still has a relative error of $8 \cdot 10^{-4}$ after 999 iterations).

With the τ -test, the number of iterations is always reduced by a factor at least 8 compared to PCG even with the smallest threshold $\tau = 10 = N/10$. With $\tau = 10N$ the number of iterations is almost the same as with the full MPCG. On these test cases the choice $\tau = N$ advocated in Section 3.3.2 seems to be a good compromise.

With the aggregation procedure, convergence is achieved even when the coarsening is quite aggressive (5 vectors per iteration means that 20 local contributions get aggregated into one search direction). As expected, keeping more vectors per iteration yields significantly better results in terms of iteration count.

3.4 Non-symmetric problem: Orthomin and BiCG [1]

We now consider solving non-symmetric linear systems. Algorithms based on PCG do not apply. MPMGRES, in its original form, leads to an exponential increase in the size of the minimization space. Although it is pointed out in [87] that this can be decreased to a linear growth by selecting some vectors, we propose to look at methods where linear growth is by construction: biCG and orthomin. In these two methods, the residual is readily available at each iteration and the preconditioner (or set of preconditioners) is applied to it (and not to the previous search direction(s)). This section is adapted from [1].

3.4.1 Multipreconditioned Orthomin and multipreconditioned BiCG

We consider the problem of finding $\mathbf{x} \in \mathbb{R}^n$ such that $A\mathbf{x} = \mathbf{b}$ where A is a non-singular (possibly non-symmetric) matrix and $\mathbf{b}$ is a given right hand side. A family of N preconditioners (also non-singular matrices) denoted by $\{H^s\}_{s=1,\dots,N}$ is used. The multipreconditioned orthomin (MPorthomin) and multipreconditioned biCG (MPbiCG) algorithms are presented in Algorithms 6 and 7 respectively.

The two properties that define the coefficients in MPorthomin are:

- the $(A^\top A)$ -conjugacy of the search directions (i.e., $Q_i^\top Q_j = \mathbf{0}$ for any $j \neq i$),

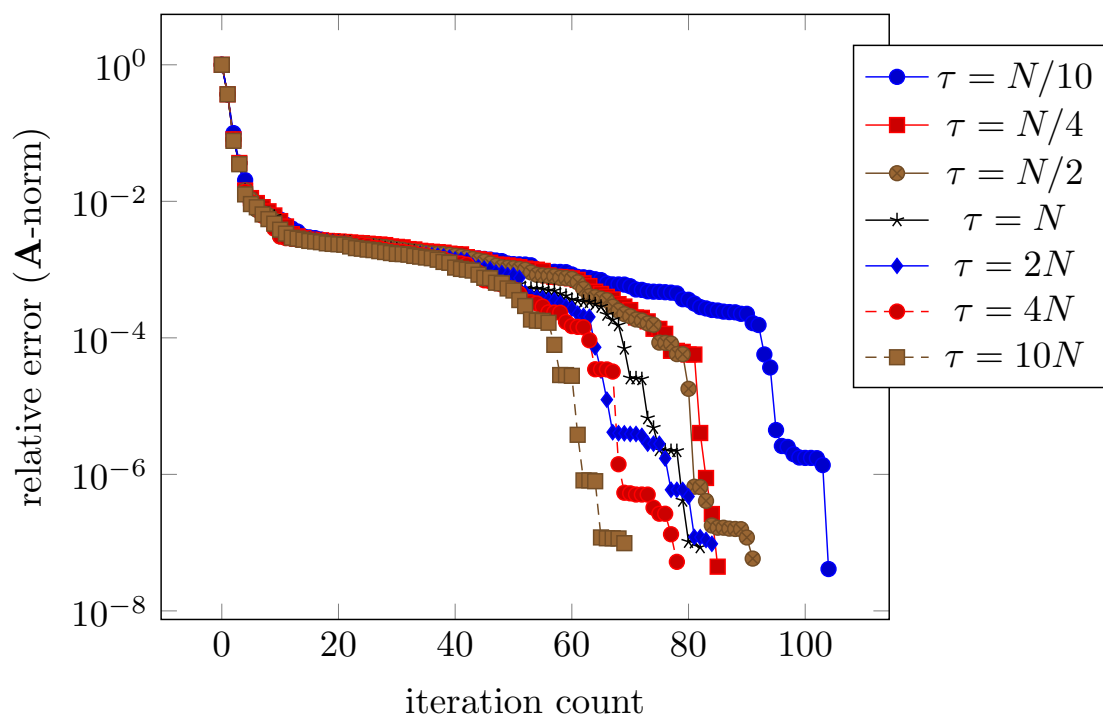


Figure 3.10: AMPCG for RAS (non-symmetric preconditioner). Compressible test case – Reduce number of directions with the τ -test – error norm versus iteration count for different values of τ

Compressible test case:					
τ -test (see Fig. 3.10)			Aggregates (see Fig. 3.12)		
τ	it	dim	m	it	dim
10	104	6059	1	889	890
25	85	5769	5	381	1910
50	91	6625	10	277	2780
100	82	6339	20	186	3740
200	84	6876	40	111	4480
400	78	6817	100	60	6100
1000	69	6153			

Incompressible test case:					
τ -test (see Fig. 3.11)			Aggregates (see Fig. 3.13)		
τ	it	dim	m	it	dim
10	124	4865	1	>999	>1000
25	99	4889	5	512	2565
50	79	4621	10	345	3460
100	72	4521	20	194	3900
200	68	4593	40	125	5040
400	65	4552	100	56	5700
1000	68	5156			

Table 3.3: AMPCG for RAS (non-symmetric preconditioner). Summary of all numerical results presented. it: number of iterations needed to reduce the initial error by a factor 10^{-7} . dim: dimension of the minimization space. There are two test cases: Compressible and Incompressible, and for each there are two ways of reducing the number of search directions at each iteration: with the τ -test (as proposed in Subsection 3.3.2) or by aggregating into m directions (as proposed in Subsection 3.3.3).

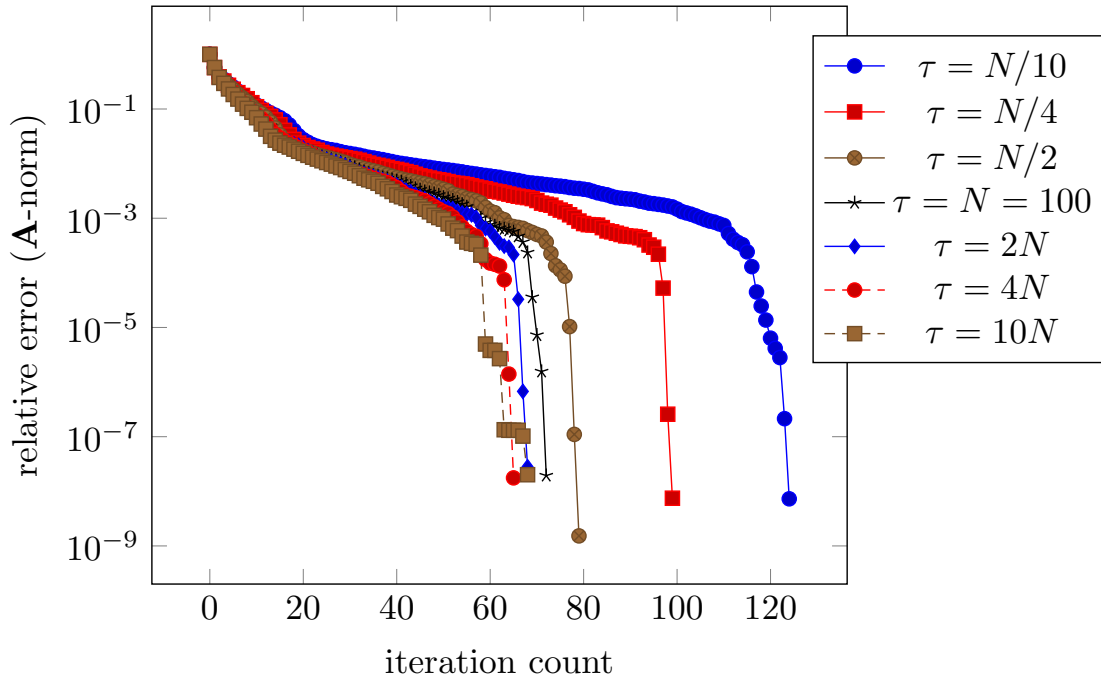


Figure 3.11: AMPCG for RAS (non-symmetric preconditioner). Incompressible test case – Reduce number of directions with the τ -test – error norm versus iteration count for different values of τ

- the minimization property: $\mathbf{x}_i$ minimizes the $A^\top A$ -norm of the error $\|\mathbf{x} - \mathbf{y}\|_{A^\top A}$ over all $\mathbf{y} \in \mathbf{x}_0 + \sum_{k < i} \text{span}(P_k)$ (i.e., $\mathbf{r}_i^\top Q_j = 0$ for any $j < i$).

The properties that define the coefficients in MPbiCG are:

- the biconjugacy of search directions (i.e., $\hat{P}_i^\top Q_j = \mathbf{0}$ if $i \neq j$),
- $\hat{\mathbf{r}}_i^\top P_j = \mathbf{0}$ and $\mathbf{r}_i^\top \hat{P}_j = \mathbf{0}$ if $j < i$. Note that this is not a minimization property: with MPbiCG, as with biCG, there is no error norm that is guaranteed to decrease at every iteration.

MPbiCG is expected to inherit the lack of stability of biCG (an issue addressed in biCGstab) but there exist many configurations where it remains competitive in spite of the need to store twice as many vectors at each iteration. We encountered unexpected poor conditioning of the orthogonalization step of MPorthomin; as a first answer we used a modified Gram-Schmidt method with reorthogonalization which allowed to achieve convergence. We also note that the preconditioning step in MPorthomin could have been $Z_{i+1} = [H^1(Q_i \alpha_i) \mid \dots \mid H^N(Q_i \alpha_i)]$ leading to an MPorthodir algorithm.

3.4.2 Numerical results: application to non-symmetric FETI

The algorithms presented in the previous sections have been applied to the FETI method [74]. As in Section 3.2.5, the methods have been implemented in the finite element suite Z-Set 8.6 [169]. The test case aims at representing a laminated material (a soft material reinforced by a stiffer one): a five-layer plate occupies the domain $\Omega = [0, 10] \times [0, 5] \times [0, 0.5]$ (see Figure 3.14). It is clamped on the $x = 0$ side and a displacement is imposed on the opposite side. Stiff layers are made of a non-associated elastoviscoplastic material (which causes the nonsymmetry of the tangent systems). Soft layers follow a

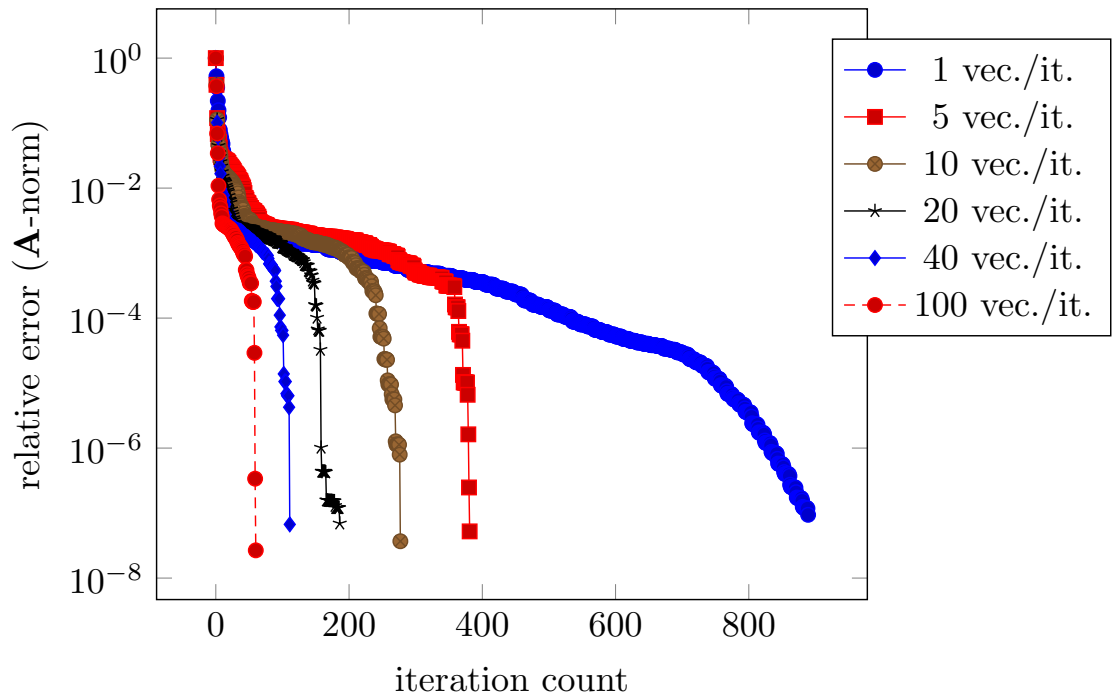


Figure 3.12: AMPCG for RAS (non-symmetric preconditioner). Compressible test case – Reduce number of directions by aggregating them into m vectors – error norm versus iteration count for different values of m

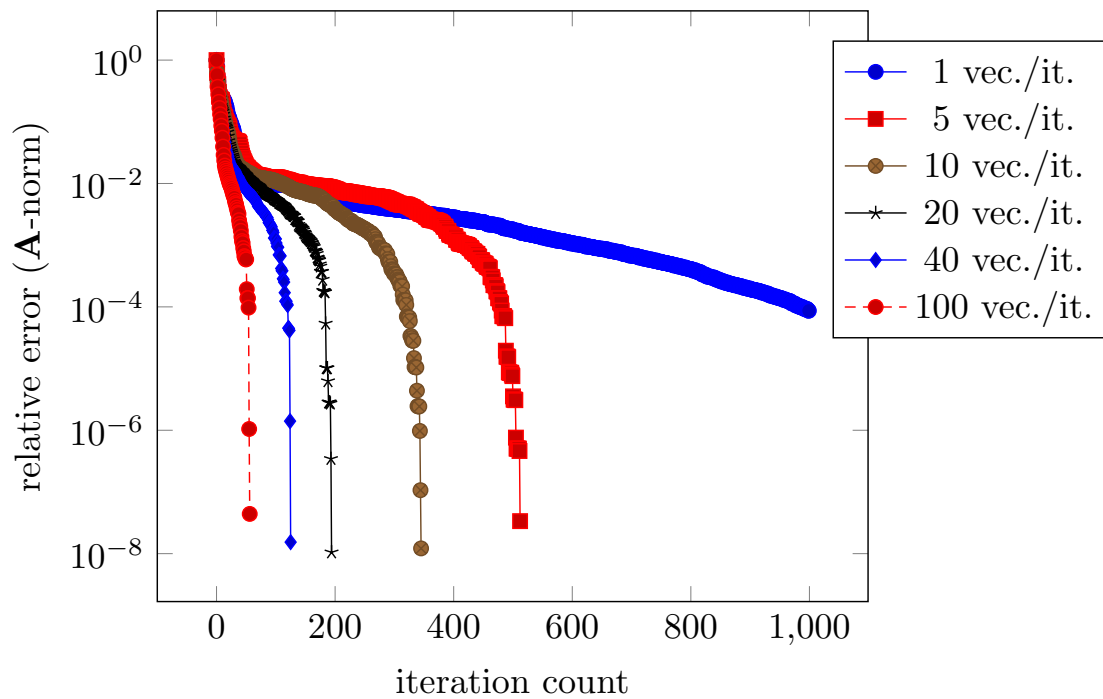


Figure 3.13: AMPCG for RAS (non-symmetric preconditioner). Incompressible test case – Reduce number of directions by aggregating them into m vectors – error norm versus iteration count for different values of m

Algorithm 6 : Multipreconditioned orthomin (MPorthomin)

```

x0 given ;                                     ← Initial Guess
r0 = b - Ax0;
Z0 = [H1r0 | ... | HNr0];   P0 = Z0;           ← Initial search directions
Q0 = AP0;
for i = 0, 1, ..., convergence do
    αi = (Qi⊤Qi)-1 (Qi⊤ri);
    xi+1 = xi + Piαi;           ← Update approximate solution
    ri+1 = ri - Qiαi;           ← Update residual
    Zi+1 = [H1ri+1 | ... | HNri+1];   ← Multi Precondition
    βi,j = (Qj⊤Qj)-1 (Qj⊤(AZi+1)),   j = 0, ..., i;
    Pi+1 = Zi+1 - ∑j=0i Pjβi,j;           ← Orthogonalize
    Qi+1 = (AZi+1) - ∑j=0i Qjβi,j;
end
Return xi+1;

```

classical linear elastic constitutive law. Small displacements are assumed. The prescribed displacement grows linearly with time, the computation ends at $t = 0.05$. A Newton scheme with a constant time step 0.01 is used. It is small enough for all Newton steps to converge in 1 iteration. The mechanical problem is:

$$\begin{cases} \operatorname{div}(\sigma(\mathbf{u})) = \mathbf{0} \text{ in } \Omega, & \mathbf{u} = \frac{xt}{10000} \mathbf{e}_x \text{ for } x \in \{0, 10\} \text{ with } \mathbf{e}_x \text{ the unit vector along the } x \text{ direction,} \\ \text{constitutive law,} & \sigma(\mathbf{u}) \cdot \mathbf{n} = 0 \text{ on } \partial\Omega \text{ with } x \notin \{0, 10\}. \end{cases} \quad (3.12)$$

The soft material follows an isotropic linear elastic constitutive law characterized by its Young's modulus $E_1 = 100$ and the Poisson's coefficient $\nu = 0.3$. The hard material follows a non-associated elastoviscoplastic constitutive law which combines a non-symmetric yield criterion, a Norton flow rule and a Ziegler kinematic hardening [33]:

$$\begin{cases} \sigma(\mathbf{u}) = \frac{E}{1+\nu}(\varepsilon(\mathbf{u}) - \varepsilon^{vp}) + \frac{\nu E}{(1+\nu)(1-2\nu)} \operatorname{Tr}(\varepsilon(\mathbf{u}) - \varepsilon^{vp}) I \text{ with } \varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^\top), \\ \dot{\varepsilon}^{vp} = \frac{3}{2} \frac{\sigma_D - X}{J_2(\sigma - X)} \dot{p} \text{ with } \dot{p} = \left(\frac{\langle f \rangle}{K} \right)^m \text{ and } f = (1-a)J(\sigma - X) + a \operatorname{Tr}(\sigma), \\ X = \frac{2}{3} C \alpha \text{ and } \dot{\alpha} = \dot{p} \left\{ \frac{\sigma - X}{J_2(\sigma - X)} - \frac{D}{C} X \right\}, \end{cases} \quad (3.13)$$

where ε^{vp} is the viscoplastic strain tensor, X is the hardening variable, $\dot{p}$ is the viscoplastic multiplier. J, J_2, I_2 are tensor invariants and σ_D is the deviatoric part of σ . The material parameters are given in the table below:

E_2	ν	K	m	C	D	a
varies	0.3	400	7	6000	100	0.3

In order to assess the robustness of the solvers, we make the ratio of Young's moduli E_2/E_1 vary between 1 and 10^5 . Since a prescribed displacement is applied, the increase of the heterogeneity ratio produces more (incompressible) viscoplastic flow inside the stiff layers, which contributes to the bad

Algorithm 7 : Multipreconditioned biconjugate gradient algorithm (MPbiCG)

```

 $\mathbf{x}_0$  given ; ← Initial Guess
 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$  and  $\hat{\mathbf{r}}_0 = \mathbf{r}_0$ ;
 $Z_0 = [H^1 \mathbf{r}_0 \mid \dots \mid H^N \mathbf{r}_0]$ ;  $P_0 = Z_0$ ; ← Initial search directions
 $\hat{Z}_0 = [H^{1^\top} \hat{\mathbf{r}}_0 \mid \dots \mid H^{N^\top} \hat{\mathbf{r}}_0]$ ;  $\hat{P}_0 = \hat{Z}_0$ ; ← Initial conjugate search directions
for  $i = 0, 1, \dots$ , convergence do
     $Q_i = AP_i$  and  $\hat{Q}_i = A^\top \hat{P}_i$ ;
     $\alpha_i = (\hat{P}_i^\top Q_i)^{-1} (\hat{P}_i^\top \mathbf{r}_i)$  and  $\hat{\alpha}_i = (P_i^\top \hat{Q}_i)^{-1} (P_i^\top \hat{\mathbf{r}}_i)$ ;
     $\mathbf{x}_{i+1} = \mathbf{x}_i + P_i \alpha_i$ ; ← Update approximate solution
     $\mathbf{r}_{i+1} = \mathbf{r}_i - Q_i \alpha_i$  and  $\hat{\mathbf{r}}_{i+1} = \hat{\mathbf{r}}_i - \hat{Q}_i \hat{\alpha}_i$ ; ← Update residual
     $Z_{i+1} = [H^1 \mathbf{r}_{i+1} \mid \dots \mid H^N \mathbf{r}_{i+1}]$  and  $\hat{Z}_{i+1} = [H^{1^\top} \hat{\mathbf{r}}_{i+1} \mid \dots \mid H^{N^\top} \hat{\mathbf{r}}_{i+1}]$ ; ← Multi Precondition
     $\beta_{i,j} = (\hat{P}_j^\top Q_j)^{-1} (\hat{Q}_j^\top Z_{i+1})$  and  $\hat{\beta}_{i,j} = (\hat{P}_j^\top Q_j)^{-T} (Q_j^\top \hat{Z}_{i+1})$  for  $j = 0, \dots, i$ ;
     $P_{i+1} = Z_{i+1} - \sum_{j=0}^i P_j \beta_{i,j}$  and  $\hat{P}_{i+1} = \hat{Z}_{i+1} - \sum_{j=0}^i \hat{P}_j \hat{\beta}_{i,j}$ ; ← Orthogonalize
end
Return  $\mathbf{x}_{i+1}$ ;

```

condition number of the tangent systems.

The problem is discretized into $150 \times 75 \times 15$ regular twenty-node brick elements (c3d20), the problem size is 2 158 068 degrees of freedom. The mesh is split into 64 domains using the Metis automatic graph partitioner. Subdomains are composed on average of 38 000 dofs and 2600 elements. A 10^{-6} relative convergence threshold is used on the Euclidean norm of the residual.

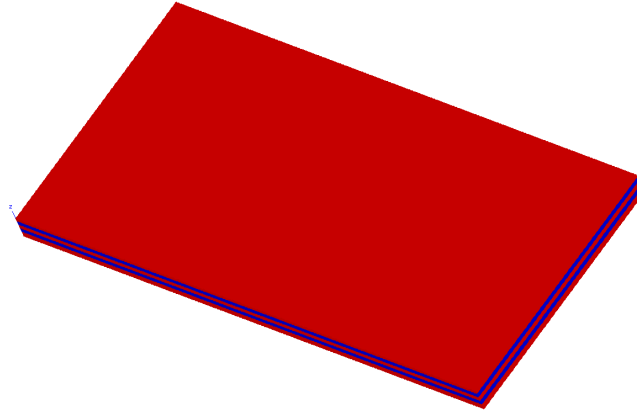


Figure 3.14: Heterogeneous composite – Soft material is in red, stiff material in blue.

Figure 3.15 summarizes all the results. Classical GMRES and orthomin are compared to our new algorithms. MPorthomin² stands for MPorthomin with a modified Gram-Schmidt with reorthogonalization (see [149, page 162]). In terms of Krylov subspace iterations, all multipreconditioned algorithms are less sensitive to the increase in the heterogeneity. The lack of stability of MPbiCG is however observed for $E_2/E_1 \geq 10^4$. The poor conditioning of the Orthomin orthogonalization step is visible for $E_2/E_1 = 10^4$. Thanks to the reorthogonalization, MPorthomin² achieves convergence even for $E_2/E_1 = 10^5$. When they converge, GMRES and orthomin remain faster than the multipreconditioned algorithms. This indicates

that the best algorithm should mix multipreconditioned iterations (that help with robustness) and usual preconditioned iterations (that are less expensive).

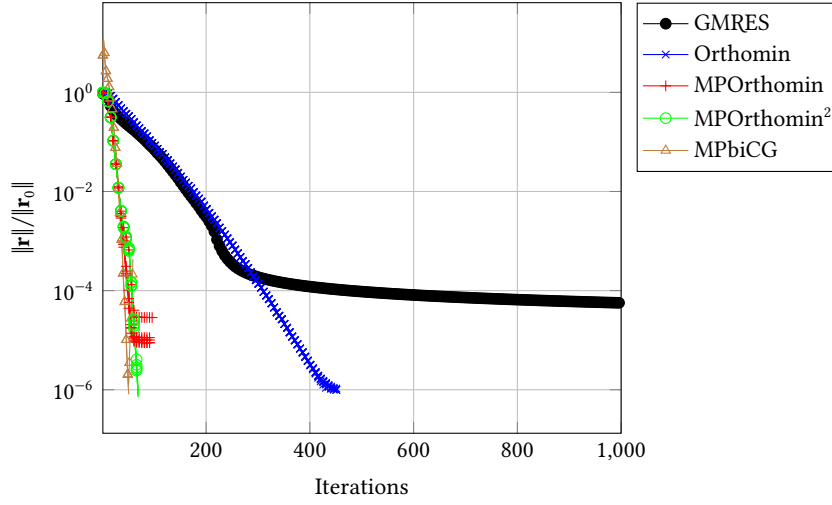
3.5 Scope and impact of this work

Adaptive multipreconditioning has received quite a lot of attention. It is the work for which I received the Leslie Fox Prize in Numerical Analysis. It is a very general framework which allows to get the advantage of multiple preconditioners at the same time as remaining cost effective. AMPCG, for spd problems, is the most advanced version of adaptive multipreconditioning. It must compete with other algorithms (DD with spectral coarse spaces, multigrid, ...). One strong advantage of adaptive multipreconditioning is that it extends quite naturally to non-symmetric preconditioners and non-symmetric problems. These extensions have been initiated but there are still many open perspectives.

Performance for various heterogeneity ratios. A dash means that the solver did not converge.

E_2/E_1	Total # Krylov iter.					Time to solution (s)				
	10^1	10^2	10^3	10^4	10^5	10^1	10^2	10^3	10^4	10^5
GMRES	315	445	825	-	-	165	198	295	-	-
orthomin	305	445	830	2231	-	153	185	265	621	-
MPbiCG	130	185	235	-	-	374	524	684	-	-
MPomin	125	185	240	472	-	221	303	386	870	-
MPomin ²	125	185	240	345	534	251	360	488	803	1565

Evolution of the residual for $h = 10^4$. All Newton steps are superposed.



Same plot as above but zoom on 100 iterations

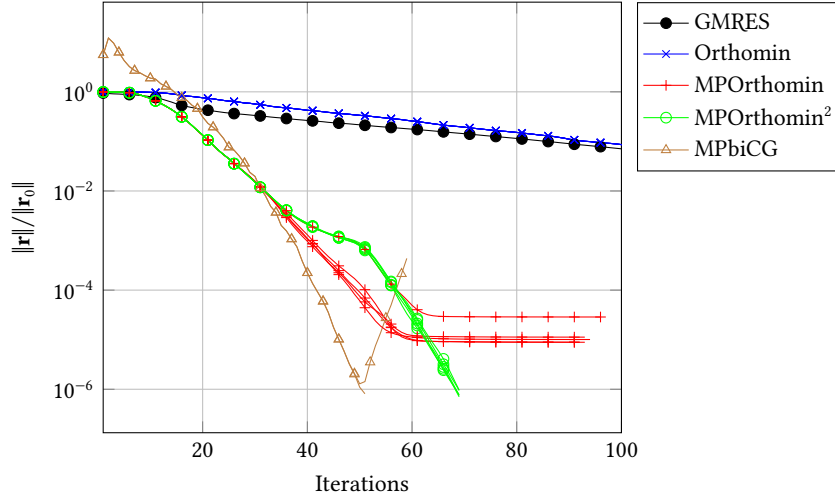


Figure 3.15: Heterogeneous composite. MPomin² uses a modified Gram-Schmidt orthogonalization with reorthogonalization.

Chapter 4

GMRES: new convergence analysis and acceleration

4.1 Introduction

GMRES, or the Generalized Minimal Residual Method, is a method of choice for solving general linear systems. First introduced by [148], the convergence of GMRES has since been extensively studied [67, 68, 122, 31, 131, 134]. The linear systems in this chapter are of the form

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{K}^{n \times n}, \quad (4.1)$$

where $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$ and A is a large, sparse, non-singular matrix. Particular emphasis will be on the cases where A has a positive definite (pd) Hermitian part. Such matrices A are referred to as pd matrices. The major change compared to my previous work is that we now consider non-Hermitian problems. Another change is that the problems can be complex-valued, but this is not the major source of difficulty.

There are essentially three components for a successful acceleration strategy, which can be used alone or be combined: preconditioning, weighting, and deflation. All three were presented in the introduction (Section 1.2). Standard references for preconditioning include [32, 149, 152]; for weighted GMRES [69, 72, 89]; and for deflation [46, 55, 71, 81], and more recently [80]. There are some equivalences between weighting and preconditioning, see, e.g., [125, Chapter 4], but here the weight and the preconditioner are kept separate for further flexibility and generality.

The objective is to propose new convergence bounds that are sufficiently explicit to indicate how to choose the preconditioner, weight matrix, and deflation spaces. Special emphasis is on Hermitian preconditioning and on applying W -GMRES in the preconditioner norm. For the cases we consider, the new bounds improve upon the more general bound of Elman [67, 68].

Background on W -GMRES. In his PhD thesis [47], Cai proposes to select for GMRES applied to a matrix P , an inner product that ‘is chosen to take advantage of some special properties of P ’. In collaboration with Widlund [50], they propose domain decomposition preconditioners for non-symmetric and indefinite second order PDEs. GMRES is considered in the norm induced by the highest order term in the variational form: the energy norm. More is said about these results below. Another choice that has been proposed in [155] is to apply GMRES in the inner product induced by the inverse of an spd preconditioner. Later, a different set of authors justify this choice in [53] by the argument that, if H is spd and A is nearly symmetric then HA is nearly H^{-1} -self adjoint. In [117], preconditioning of saddle point problems is tackled and the bounds from [155] are applied in an inner product that is derived

from a non-symmetric triangular preconditioner. The authors of [143] address the problem of finding inner products that make a general preconditioned system nearly normal or nearly non-symmetric. A detailed presentation of what is now called weighted GMRES, with results both on convergence and implementation strategy, can be found in [150]. In [72], the author reinvents the idea of changing the norm within GMRES in a general framework and coins the term *weighted GMRES*. The inner products considered by [72] are diagonal matrices of weights that changes at each restart of the method. Larger weights are associated to the larger components of the residual at the end of the previous cycle, leading to a faster convergence. This idea is compared to other accelerators for GMRES in [89].

Background on preconditioning with information coming from the Hermitian part of A . The idea of separating the Hermitian and skew-Hermitian part of a matrix in order to approximate the solution of a linear system has been exploited in the Hermitian and skew-Hermitian splitting methods introduced in [27]. The field of domain decomposition for non-spd problems was paved by [47, 50] (see also [159, Chapter 11]). The authors solve convection-diffusion-reaction with a two-level additive Schwarz preconditioner where the coarse space is based on a coarse grid. The one level component in the preconditioner takes two forms: either the local solvers come from the original (non-symmetric and/or indefinite) matrix, or the local solvers come from an spd part of the problem matrix. GMRES is applied in the norm induced by that spd part. In both cases the coarse projector is for the original operator. The algorithm is generalized in [167] to any matrix that can be viewed as a perturbation of an spd matrix. Their proposed preconditioner is the combination of a very good preconditioner for the spd part and a coarse solve. A parameter δ_0 qualifies how efficient the coarse space is at filling the gap between the original problem and the spd one. It enters into the convergence estimate. The algorithm is later called CSPD for Coarse Grid Plus SPD Preconditioning in a numerical comparison with other algorithms [48]. More recently, domain decomposition for Helmholtz has been studied [84, 85]. This case is symmetric indefinite and applying GMRES in the energy norm is a crucial part of the proof. Weighted GMRES was also applied to study the spd GenEO eigenproblem applied to indefinite and non-self-adjoint problems in [36, 37]. It is proved and observed numerically that GenEO performs well also on a family of non-spd second-order problems. Finally, the authors of [35] prove an abstract framework for one-level additive Schwarz for non-Hermitian or indefinite problems. They illustrate their results by solving the convection-diffusion-reaction equation. The present work takes a more algebraic route and proves results that are not restricted to domain decomposition.

Outline. The outline of the rest of this chapter is as follows. Section 4.2 considers preconditioned W -GMRES. Section 4.3 adds deflation to the framework. In both cases the analysis starts in a general setting before focusing on the case of hpd preconditioning, weighting by the preconditioner and solving pd problems. The new analysis in Section 4.2 shows that convergence is bounded with respect to how well the hermitian part of A is preconditioned and how non-Hermitian the problem is. The new analysis in Section 4.3 proposes to improve the bound by deflating eigenmodes of a particular generalized eigenvalue problem. The motivation is to accelerate convergence. Numerical results confirm and illustrate the results.

4.2 Weighted and Preconditioned GMRES [7]

4.2.1 General case

We first consider the convergence of W -GMRES applied to a non-singular matrix A and preconditioned on the right by H . It is also assumed that H is non-singular. From (1.9) in the introduction, the m -th

residual $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$ satisfies:

$$\|\mathbf{r}_m\|_W = \min_{q \in \mathbb{P}_m; q(0)=1} \|q(AH)\mathbf{r}_0\|_W,$$

where $\mathbb{P}_m$ denotes the set of all polynomials of degree at most m . Two subsequent residuals hence satisfy the relation

$$\|\mathbf{r}_m\|_W \leq \min_{q \in \mathbb{P}_1; q(0)=1} \|q(AH)\mathbf{r}_{m-1}\|_W = \min \{\|\mathbf{r}_{m-1} - \alpha AH\mathbf{r}_{m-1}\|_W; \alpha \in \mathbb{C}\},$$

of which the solution is $\alpha = \frac{\langle AH\mathbf{r}_{m-1}, \mathbf{r}_{m-1} \rangle_W}{\|AH\mathbf{r}_{m-1}\|_W^2}$ to get the W -orthogonal projection of $\mathbf{r}_{m-1}$ onto $\text{span}(AH\mathbf{r}_{m-1})$. This leads to

$$\frac{\|\mathbf{r}_m\|_W^2}{\|\mathbf{r}_{m-1}\|_W^2} \leq 1 - \frac{|\langle AH\mathbf{r}_{m-1}, \mathbf{r}_{m-1} \rangle_W|^2}{\|AH\mathbf{r}_{m-1}\|_W^2 \|\mathbf{r}_{m-1}\|_W^2},$$

which brings us to the first result.

Theorem 4.2.1 (Convergence of preconditioned W -GMRES). *Assume that the operator A and preconditioner H are non-singular, and that the weight matrix W is hpd. The m -th iterate of W -GMRES right-preconditioned by H satisfies*

$$\frac{\|\mathbf{r}_m\|_W}{\|\mathbf{r}_0\|_W} \leq \left(1 - \inf_{\mathbf{y} \neq \mathbf{0}} \frac{|\langle AH\mathbf{y}, \mathbf{y} \rangle_W|^2}{\|AH\mathbf{y}\|_W^2 \|\mathbf{y}\|_W^2} \right)^{m/2}. \quad (4.2)$$

The Cauchy-Schwarz inequality ensures that the square root is well defined. The terms in (4.2) can be grouped as

$$\left[\frac{\langle AH\mathbf{y}, \mathbf{y} \rangle_W}{\|\mathbf{y}\|_W^2} \times \frac{\|\mathbf{y}\|_W}{\|AH\mathbf{y}\|_W} \right]^2 \geq \left[\frac{d(0, W_W(AH))}{\|AH\|_W} \right]^2, \quad (4.3)$$

where

- $d(0, W_W(AH)) = \inf \left\{ \frac{|\langle AH\mathbf{u}, \mathbf{u} \rangle_W|}{\langle \mathbf{u}, \mathbf{u} \rangle_W}; \mathbf{u} \in \mathbb{C}^n \setminus \{0\} \right\}$ is the distance to zero of the W -field of values of AH ,
- $\|AH\|_W$ denotes the matrix norm of AH induced by the vector norm W .

This results in the Elman bound, a result (out of many) from [67]

$$\frac{\|\mathbf{r}_m\|_W}{\|\mathbf{r}_0\|_W} \leq \left(1 - \left[\frac{d(0, W_W(AH))}{\|AH\|_W} \right]^2 \right)^{m/2}. \quad (4.4)$$

It is noted in [123] and [66] that the new bound from Theorem 4.2.1 is stronger than the Elman bound.

The convergence study now focuses on some less general cases.

4.2.2 Hermitian positive definite preconditioning

Two strong assumptions are made:

1. the preconditioner H is hpd,

2. the inner product is induced by the preconditioner: $W = H$. With this choice, the operations can be arranged so that the preconditioner is applied only once per iteration.

To advance the convergence analysis, starting from Theorem 4.2.1, the fraction in the convergence bound is studied:

$$\begin{aligned}
\inf_{\mathbf{y} \neq 0} \frac{|\langle A\mathbf{H}\mathbf{y}, \mathbf{y} \rangle_H|^2}{\|A\mathbf{H}\mathbf{y}\|_H^2 \|\mathbf{y}\|_H^2} &= \inf_{\mathbf{y} \neq 0} \frac{|\langle H A \mathbf{H} \mathbf{y}, \mathbf{y} \rangle|^2}{\langle H A \mathbf{H} \mathbf{y}, A \mathbf{H} \mathbf{y} \rangle \langle \mathbf{H} \mathbf{y}, \mathbf{y} \rangle} \\
&= \inf_{\mathbf{y} \neq 0} \frac{|\langle A \mathbf{y}, \mathbf{y} \rangle|^2}{\langle H A \mathbf{y}, A \mathbf{y} \rangle \langle H^{-1} \mathbf{y}, \mathbf{y} \rangle} \quad (\text{by } \mathbf{y} \leftarrow H \mathbf{y}) \\
&\geq \inf_{\mathbf{y} \neq 0} \frac{|\langle A \mathbf{y}, \mathbf{y} \rangle|}{\langle H A \mathbf{y}, A \mathbf{y} \rangle} \times \inf_{\mathbf{y} \neq 0} \frac{|\langle A \mathbf{y}, \mathbf{y} \rangle|}{\langle H^{-1} \mathbf{y}, \mathbf{y} \rangle} \\
&= \inf_{\mathbf{y} \neq 0} \frac{|\langle A^{-1} \mathbf{y}, \mathbf{y} \rangle|}{\langle \mathbf{H} \mathbf{y}, \mathbf{y} \rangle} \times \inf_{\mathbf{y} \neq 0} \frac{|\langle A \mathbf{y}, \mathbf{y} \rangle|}{\langle H^{-1} \mathbf{y}, \mathbf{y} \rangle} \quad (\text{by } \mathbf{y} \leftarrow A \mathbf{y}) \\
&\geq \inf_{\mathbf{y} \neq 0} \frac{|\langle M(A^{-1}) \mathbf{y}, \mathbf{y} \rangle|}{\langle \mathbf{H} \mathbf{y}, \mathbf{y} \rangle} \times \inf_{\mathbf{y} \neq 0} \frac{|\langle M(A) \mathbf{y}, \mathbf{y} \rangle|}{\langle H^{-1} \mathbf{y}, \mathbf{y} \rangle}
\end{aligned}$$

where, for any matrix B , $M(B)$ is the Hermitian part of B ,

$$M(B) := \frac{B + B^*}{2},$$

which satisfies $|\langle B \mathbf{y}, \mathbf{y} \rangle| \geq |\operatorname{Re}(\langle B \mathbf{y}, \mathbf{y} \rangle)| = |\langle M(B) \mathbf{y}, \mathbf{y} \rangle|$. Consequently, the m -th iterate of preconditioned W -GMRES satisfies

$$\frac{\|\mathbf{r}_m\|_H}{\|\mathbf{r}_0\|_H} \leq \left[1 - \inf_{\mathbf{y} \neq 0} \frac{|\langle M(A^{-1}) \mathbf{y}, \mathbf{y} \rangle|}{\langle \mathbf{H} \mathbf{y}, \mathbf{y} \rangle} \times \inf_{\mathbf{y} \neq 0} \frac{|\langle M(A) \mathbf{y}, \mathbf{y} \rangle|}{\langle H^{-1} \mathbf{y}, \mathbf{y} \rangle} \right]^{m/2}. \quad (4.5)$$

The same result holds for all truncated and restarted versions of W -GMRES.

4.2.3 Positive definite A

Assuming that the problem matrix A is positive definite, *i.e.*, that $M(A)$ is hpd, the calculations from the previous paragraph are resumed:

$$\begin{aligned}
\inf_{\mathbf{y} \neq 0} \frac{\langle A \mathbf{H} \mathbf{y}, \mathbf{y} \rangle_W^2}{\|A \mathbf{H} \mathbf{y}\|_W^2 \|\mathbf{y}\|_W^2} &\geq \inf_{\mathbf{y} \neq 0} \frac{|\langle M(A^{-1}) \mathbf{y}, \mathbf{y} \rangle|}{|\langle M(A)^{-1} \mathbf{y}, \mathbf{y} \rangle|} \times \inf_{\mathbf{y} \neq 0} \frac{|\langle M(A)^{-1} \mathbf{y}, \mathbf{y} \rangle|}{\langle \mathbf{H} \mathbf{y}, \mathbf{y} \rangle} \times \inf_{\mathbf{y} \neq 0} \frac{|\langle M(A) \mathbf{y}, \mathbf{y} \rangle|}{\langle H^{-1} \mathbf{y}, \mathbf{y} \rangle} \\
&\geq \inf_{\mathbf{y} \neq 0} \frac{\langle M(A^{-1}) \mathbf{y}, \mathbf{y} \rangle}{\langle M(A)^{-1} \mathbf{y}, \mathbf{y} \rangle} \times \inf_{\mathbf{y} \neq 0} \frac{\langle M(A)^{-1} \mathbf{y}, \mathbf{y} \rangle}{\langle \mathbf{H} \mathbf{y}, \mathbf{y} \rangle} \times \inf_{\mathbf{y} \neq 0} \frac{\langle M(A) \mathbf{y}, \mathbf{y} \rangle}{\langle H^{-1} \mathbf{y}, \mathbf{y} \rangle},
\end{aligned}$$

where division by $\langle M(A)^{-1} \mathbf{y}, \mathbf{y} \rangle$ is not by zero and removing the absolute values does not change the result. Indeed, by [104][Property (1.6) on page 10], $M(A)$ being positive definite implies that A^{-1} is well defined and that $M(A^{-1})$ is also positive definite.

By the Rayleigh-Ritz characterization of eigenvalues, it can be realized that the last two terms in the product are related to the smallest and largest eigenvalues of $HM(A)$, denoted by $\lambda_{\min}(HM(A))$ and

$\lambda_{\max}(HM(A))$ in such a way that

$$\inf_{\mathbf{y} \neq 0} \frac{\langle AHy, \mathbf{y} \rangle_W^2}{\|AHy\|_W^2 \|\mathbf{y}\|_W^2} \geq \inf_{\mathbf{y} \neq 0} \frac{\langle M(A^{-1})\mathbf{y}, \mathbf{y} \rangle}{\langle M(A)^{-1}\mathbf{y}, \mathbf{y} \rangle} \times \frac{\lambda_{\min}(HM(A))}{\lambda_{\max}(HM(A))}.$$

For the remaining term, a result by Johnson provides a bound.

Lemma 4.2.2 (Corollary 3 in [103], see also [102, 104]). *Let A be positive definite and $c \in \mathbb{R}$. The matrix $cM(A^{-1}) - M(A)^{-1}$ is positive definite if and only if $c > 1 + \rho(M(A)^{-1}N(A))^2$, with*

$$N(A) = \frac{A - A^*}{2} \text{ and } \rho(M(A)^{-1}N(A)) \text{ the spectral radius of } M(A)^{-1}N(A).$$

An equivalent reformulation of the lemma is that

$$\inf_{\mathbf{y} \neq 0} \frac{\langle M(A^{-1})\mathbf{y}, \mathbf{y} \rangle}{\langle M(A)^{-1}\mathbf{y}, \mathbf{y} \rangle} = [1 + \rho(M(A)^{-1}N(A))^2]^{-1}. \quad (4.6)$$

Putting everything together, we get:

Theorem 4.2.3 (Convergence of W -GMRES preconditioned by W for positive definite A). *Assume that the operator A is positive definite, that the (right) preconditioner H is hpd and that the weight in W -GMRES is H . Then*

$$\frac{\|\mathbf{r}_m\|_H}{\|\mathbf{r}_0\|_H} \leq \left[1 - \frac{\lambda_{\min}(HM(A))}{\lambda_{\max}(HM(A))} \times \frac{1}{1 + \rho(M(A)^{-1}N(A))^2} \right]^{m/2}. \quad (4.7)$$

The estimate in the theorem has split the residual bound into two components:

- the condition number of the Hermitian part of A preconditioned by H ,
- a measure $\rho(M(A)^{-1}N(A))$ of the strength of non-Hermitianness of A that is independent of H .

In particular, if H is a domain decomposition preconditioner such that $\kappa(HM(A))$ can be bounded independently of the number of subdomains, then the convergence bound above does not depend on the number of subdomains either. In other words, if a scalable domain decomposition method is known for the Hermitian part of A , the proposed algorithm for the non-Hermitian problem is also scalable.

For many problems arising from the discretization of PDEs, $M(A)$ is derived from a differential operator of higher order than that producing $N(A)$ so $\rho(M(A)^{-1}N(A))$ is bounded independently of the discretization step (an observation already made in a different context in [164] and [68][Section 10]). If H can also be chosen so that the preconditioned Hermitian part is well conditioned independently of h then the residual bound is h -independent.

4.2.4 Numerical Illustration: Convection-Diffusion-Reaction

In this section, the problem considered is the convection-diffusion-reaction problem posed in $\Omega = [0, 1]^2$. It is a real-valued problem ($\mathbb{K} = \mathbb{R}$), so Hermitian means symmetric. The presentation, notation and test case are inspired by [35, 34]. The strong formulation of the problem is:

$$\begin{aligned} c_0 u + \operatorname{div}(\mathbf{a}u) - \operatorname{div}(\nu \nabla u) &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega. \end{aligned}$$

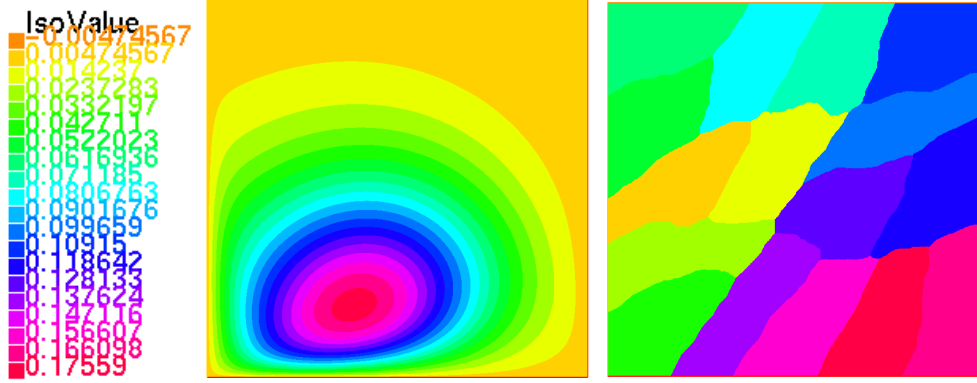


Figure 4.1: Left: Solution for $c_0 = \nu = 0.1$. Right: Partition into $N = 16$ subdomains computed by Metis.

The variational formulation is: Find $u \in H_0^1(\Omega)$ such that

$$\underbrace{\int_{\Omega} \left(\left(c_0 + \frac{1}{2} \operatorname{div} \mathbf{a} \right) uv + \nu \nabla u \cdot \nabla v \right)}_{\text{symmetric part}} + \int_{\Omega} \left(\frac{1}{2} \mathbf{a} \cdot \nabla uv - \frac{1}{2} \mathbf{a} \cdot \nabla vu \right) = \int_{\Omega} f v,$$

for all $v \in H_0^1(\Omega)$.

The right hand side and the convection field are chosen as

$$f(x, y) = \exp(-10((x - 0.5)^2 + (y - 0.1)^2)) \text{ and } \mathbf{a}(x, y) = 2\pi[-(y - 0.1), x - 0.5].$$

The reaction coefficient $c_0 > 0$ and viscosity $\nu > 0$ are chosen to be constant over Ω . Together with $\operatorname{div} \mathbf{a} = 0$, positivity of c_0 and ν ensures that A is positive-definite. Varying c_0 and ν inside Ω would not cause any additional difficulty since the proposed preconditioner handles heterogeneous c_0 and ν (in the Hermitian part). The problem is discretized by Lagrange $\mathbb{P}_1$ finite elements on a regular triangular mesh of characteristic length h . GMRES is implemented as GCR in FreeFem++ [92] with the `ffddm` library [160]. All iteration counts correspond to the number of iterations needed to reach $\|\mathbf{r}_i\|_H < 10^{-6} \|\mathbf{b}\|_H$ starting from a zero initial guess. The Dirichlet boundary condition has been enforced by penalization. The solution computed by a direct solve on a 501×501 degree of freedom (dof) mesh has been plotted in Figure 4.1 (left) for values of $c_0 = \nu = 0.1$.

Let $M(A)$ be the matrix corresponding to the discretization of the symmetric part of the problem. $M(A)$ is preconditioned by the Additive Schwarz domain decomposition method with the GenEO coarse space with threshold $\tau = 0.15$ (see Table 2.2 or [20, 18]). The partition of Ω into N subdomains Ω^s is computed automatically by Metis. One layer of overlap is added to each Ω^s . The `ffddm` options are

```
-ffddm_schwarz_method asm
-ffddm_geneo_threshold 0.15
-ffddm_schwarz_coarse_correction BNN.
```

From the theory of GenEO, the condition number of the preconditioned symmetric part of the problem is bounded from above by $\kappa(HM(A)) \leq 3(1 + 3/0.15) = 63$. Injecting this into the bound from Theorem 4.2.3, it is obtained that the residuals satisfy

$$\frac{\|\mathbf{r}_i\|_H}{\|\mathbf{r}_0\|_H} \leq \left[1 - \frac{0.016}{1 + \rho(M(A)^{-1}N(A))^2} \right]^{i/2}.$$

Discretization step h	1/500	1/200	1/30	1/10
$\rho(M(A)^{-1}N(A))$	0.3391	0.3389	0.3380	0.3136

Table 4.1: Computation of $\rho(M(A)^{-1}N(A))$ when the discretization step h varies. Case $\nu = c_0 = 1$.

Number of subdomains	4	8	16	32
Iteration count for $h = 1/200$	19	20	20	20
Iteration count for $h = 1/500$	18	18	19	20

Table 4.2: Scalability. In each line, the same problem is solved for an increasing number of subdomains. The iteration count remains nearly constant. Case $\nu = c_0 = 1$.

If, for example, $\rho(M(A)^{-1}N(A)) \leq 1$ the bound gives

$$\frac{\|\mathbf{r}_i\|_H}{\|\mathbf{r}_0\|_H} \leq \sqrt{\left[1 - \frac{1}{126}\right]^i} = 0.996^i; \quad \frac{\|\mathbf{r}_{500}\|_H}{\|\mathbf{r}_0\|_H} \leq 0.14 \text{ and } \frac{\|\mathbf{r}_{3468}\|_H}{\|\mathbf{r}_0\|_H} < 1.0 \cdot 10^{-6}.$$

The residual is guaranteed to decrease only by 0.4% at each iteration. Luckily, this turns out to be pessimistic.

Value of $\rho(M(A)^{-1}N(A))$. It remains to consider the value of $\rho(M(A)^{-1}N(A))$. With arguments from the proof of [35][Lemma 4.5], it is derived that

$$\rho(M(A)^{-1}N(A)) \leq \frac{1}{2} \frac{\|\mathbf{a}\|_{L^\infty(\Omega)}}{\sqrt{\inf(\nu) \inf(c_0 + \frac{1}{2} \operatorname{div}(\mathbf{a}))}}. \quad (4.8)$$

Bound (4.8) for $\rho(M(A)^{-1}N(A))$ does not depend on the discretization step h so neither does the overall convergence bound. In Table 4.1, the actual value of $\rho(M(A)^{-1}N(A))$ computed by Octave's *eigs* is given in the case where $c_0 = \nu = 1$. The discretization step varies between $h = 1/10$ and $h = 1/200$ for which $\rho(M(A)^{-1}N(A))$ varies only between 0.31 and 0.34. In comparison, bound (4.8) gives $\rho(M(A)^{-1}N(A)) \leq 3.23$. The bound is approximately 10 times larger than the computed value of the spectral radius.

Scalability (Table 4.2). Since Additive Schwarz with the GenEO coarse space is scalable, it has been proved that the overall algorithm is scalable. This is checked by solving the same problem for different partitions into subdomains (all computed by Metis). Two discretizations are considered: $h = 1/200$ and $h = 1/500$. For this test $\nu = c_0 = 1$. It is observed that the method is indeed scalable: the iteration counts reported in Table 4.2 do not depend on the number of subdomains.

Dependency on h (Table 4.3). The influence of the discretization step h on the iteration count is studied in Table 4.3. The partition is set to $N = 8$ subdomains. For, three different values of $c_0 = \nu$, the mesh size h varies between $1/100$ and $1/2000$. It is observed that, as predicted by the theory, the number of iterations remains almost constant when h varies.

Dependency on strength of non-symmetry (Table 4.4). For this test, $h = 1/500$ and $N = 8$. The value of ν and c_0 are varied and the corresponding iteration counts are reported in Table 4.4. The problem converges very fast when the symmetric part dominates and not so fast otherwise. This is expected by design of the algorithm and addressed next. A word of warning is that when c_0 and ν go to zero, a

	$1/h$	2000	1000	500	200	100
	# dofs (n)	4 004 001	1 002 001	251 001	40 401	10 201
$c_0 = \nu = 10$	it. count	16	16	17	17	20
$c_0 = \nu = 1$	it. count	17	18	19	20	21
$c_0 = \nu = 0.1$	it. count	39	40	42	43	41

Table 4.3: Dependency on mesh size h . Each line corresponds to a value of (ν, c_0) .

$c_0 = \nu$	0.001	0.01	0.1	1	10	only symmetric part
Iteration count	> 500 ($1.1 \cdot 10^{-4}$)	161	42	19	17	17

Table 4.4: Influence of the relative importance of the symmetric term and the skew-symmetric term. Case $N = 8$ subdomains and $h = 1/500$. The value in parenthesis corresponds to the relative residual $\|\mathbf{r}_{500}\|_H / \|\mathbf{b}\|_H$ when the algorithm stopped after 500 iterations.

stabilized discretization scheme should be preferred to avoid instabilities. In this case, the new results apply to the stabilized system.

4.3 Weighted, Deflated and Preconditioned GMRES [8]

In the previous section, both theoretical and numerical results agree that convergence deteriorates if the problem becomes more strongly non-Hermitian. In joint work with Daniel Szyld (Temple University), we have proposed to address this by spectral deflation. First we extend the new convergence bounds to include a deflation operator.

4.3.1 General case

Deflation has already been presented in the introduction. Recall that the deflation operators (projectors) are characterized¹ by two full rank matrices $Y, Z \in \mathbb{K}^{n \times m}$ and defined as

$$P_D := I - AZ(Y^*AZ)^{-1}Y^* \text{ and } Q_D := I - Z(Y^*AZ)^{-1}Y^*A. \quad (4.9)$$

We now focus on the application of W -GMRES to the deflated and right-preconditioned system. Given a non-singular preconditioner H , this means that the system solved by GMRES is

$$P_D A H \mathbf{u} = P_D \mathbf{b} \text{ with } \mathbf{w} = H \mathbf{u}. \quad (4.10)$$

Theorem 4.3.1. *The deflation operators are well defined and weighted GMRES does not break down when solving the deflated and preconditioned linear system (4.10) if*

$$\ker(Y^*) \cap \text{range}(AZ) = \{\mathbf{0}\} \text{ and } \ker(Y^*) \cap \text{range}(H^{-1}Z) = \{\mathbf{0}\}.$$

For future reference the following lemma states some easy to prove but very useful properties.

¹The projection operators P_D and Q_D are entirely defined through their range and their kernel. This means that, Y and Z need only be defined up to their ranges, not necessarily for the particular choice of their columns.

Lemma 4.3.2. *The deflation operators satisfy $P_D A = A Q_D = P_D A Q_D$, and*

$$\begin{aligned} \ker(P_D) &= \text{range}(AZ), & \text{range}(P_D) &= (\ker(P_D^*))^\perp = \ker(Y^*), \\ \ker(P_D^*) &= \text{range}(Y), & \text{range}(P_D^*) &= (\ker(P_D))^\perp = \ker(Z^* A^*), \\ \ker(Q_D) &= \text{range}(Z), & \text{range}(Q_D) &= (\ker(Q_D^*))^\perp = \ker(Y^* A), \\ \ker(Q_D^*) &= \text{range}(A^* Y), & \text{range}(Q_D^*) &= (\ker(Q_D))^\perp = \ker(Z^*). \end{aligned}$$

Two cases stand out that will be useful further on in this work. The proofs are not included.

Lemma 4.3.3. *If H is hpd and $Y = HAZ$, then the projection operator P_D is orthogonal in the H inner product. Moreover the condition $\ker(Y^*) \cap \text{range}(AZ) = \{0\}$ in Theorem 4.3.1 is automatically satisfied.*

Lemma 4.3.4. *If Y is an invariant subspace of $H^* A^*$, then $Q_D H P_D = H P_D$. Moreover, the two conditions in Theorem 4.3.1 are equivalent.*

The following result extends Theorem 4.2.1 (also [7, Theorem 3]) to the case where we have a deflation space.

Theorem 4.3.5 (Convergence of preconditioned and deflated W-GMRES). *Assume that the operator A and preconditioner H are non-singular, that the weight matrix W is hpd, and that the two conditions from Theorem 4.3.1 are satisfied. The m -th iterate of W-GMRES right-preconditioned by H and deflated by P_D satisfies*

$$\frac{\|\mathbf{r}_m\|_W^2}{\|\mathbf{r}_0\|_W^2} \leq \left[1 - \inf_{\mathbf{y} \in \text{range}(P_D) \setminus \{0\}} \frac{|\langle P_D A H \mathbf{y}, \mathbf{y} \rangle_W|^2}{\|P_D A H \mathbf{y}\|_W^2 \|\mathbf{y}\|_W^2} \right]^{m/2}.$$

From now on we use the notation

$$\theta(A, H, W, Y, Z) := \inf_{\mathbf{y} \in \text{range}(P_D) \setminus \{0\}} \frac{|\langle P_D A H \mathbf{y}, \mathbf{y} \rangle_W|^2}{\|P_D A H \mathbf{y}\|_W^2 \|\mathbf{y}\|_W^2}, \quad (4.11)$$

where $\theta(A, H, W, Y, Z)$ is indexed by the operator A , the preconditioner H , the weight matrix W as well as the deflation spaces represented by Y and Z .

4.3.2 Hermitian positive definite preconditioning for positive definite A

We make the same three assumptions as previously.

1. The preconditioner H is hpd,
2. W-GMRES is applied in the inner product induced by the preconditioner, i.e., $W = H$,
3. The coefficient matrix A is positive definite (in the sense that its Hermitian part is hpd).

The quantity in the convergence bound of Theorem 4.3.5 can now be rewritten as

$$\begin{aligned} \theta(A, H, H, Y, Z) &= \inf_{\mathbf{y} \in \text{range}(P_D) \setminus \{0\}} \frac{|\langle H P_D A H \mathbf{y}, \mathbf{y} \rangle|^2}{\|P_D A H \mathbf{y}\|_H^2 \|\mathbf{y}\|_H^2} \\ &= \inf_{\mathbf{y} \in \text{range}(H P_D) \setminus \{0\}} \frac{|\langle P_D A \mathbf{y}, \mathbf{y} \rangle|^2}{\|P_D A \mathbf{y}\|_H^2 \|\mathbf{y}\|_{H^{-1}}^2}. \end{aligned} \quad (4.12)$$

From here, two cases are considered that differ by the constraint imposed on the deflation spaces. In each case, the objective is to make explicit a condition that must be satisfied by H , Y and Z in order to ensure fast convergence. The results are summed up in the next theorem. It extends Theorem 4.2.3 (also [7, Theorem 6]) to the case where we have a deflation space.

Theorem 4.3.6. *Let us assume that A is positive definite, H is hpd and $W = H$. Let*

$$M = \frac{A + A^*}{2}$$

denote the Hermitian part of A , and $\lambda_{\min}(HM)$ and $\lambda_{\max}(HM)$ denote the extreme eigenvalues of HM . The quantity θ that comes into the convergence bound of Theorem 4.3.5 can be bounded as follows.

1. *If $Y = HAZ$, i.e., P_D is H -orthogonal then*

$$\theta(A, H, H, HAZ, Z) \geq \frac{\lambda_{\min}(HM)}{\lambda_{\max}(HM)} \times \inf_{\mathbf{y} \in \ker(Z^* A^* A^{-1}) \setminus \{0\}} \frac{|\langle A^{-1} \mathbf{y}, \mathbf{y} \rangle|}{\langle \mathbf{y}, M^{-1} \mathbf{y} \rangle}.$$

2. *If Y is an invariant subset of HA^* and $\ker(Y^*) \cap \text{range}(H^{-1}Z) = \{0\}$, then*

$$\theta(A, H, H, Y, Z) \geq \frac{\lambda_{\min}(HM)}{\lambda_{\max}(HM)} \times \inf_{\mathbf{y} \in \ker(Y^*) \setminus \{0\}} \frac{|\langle A^{-1} \mathbf{y}, \mathbf{y} \rangle|}{\langle M^{-1} \mathbf{y}, \mathbf{y} \rangle}.$$

(M was previously denoted $M(A)$.) We observe that since the matrices H and M are hpd, the eigenvalues of HM are real and positive. The proof is similar to the proof of Theorem 4.2.3. Instead of inserting the estimate by Johnson, the last term has been kept and the space of vectors for which it must be bounded is found by applying results from Lemma 4.3.2. Next a spectral deflation space is proposed in order to bound exactly that term, i.e., $\inf \frac{|\langle A^{-1} \mathbf{y}, \mathbf{y} \rangle|}{\langle M^{-1} \mathbf{y}, \mathbf{y} \rangle}$ on the appropriate space.

4.3.3 A new spectral deflation space

Our objective is to compute Y and Z in such a way that the convergence of preconditioned and deflated W -GMRES is bounded explicitly. More precisely, following the results in Theorem 4.3.9 we aim to find a subset of vectors that satisfy

$$\frac{|\langle A^{-1} \mathbf{y}, \mathbf{y} \rangle|}{\langle \mathbf{y}, M^{-1} \mathbf{y} \rangle} \geq \gamma,$$

for some choice of γ . This can be done by computing eigenvectors of a particular generalized eigenvalue problem. Then these eigenvectors enter into the choice of Y and Z .

Let N be the skew-Hermitian part of A

$$N := \frac{A - A^*}{2},$$

so that $A = M + N$. (N was previously denoted $N(A)$.)

Definition 4.3.7 (Deflation Space). *Given a positive definite matrix $A = M + N$ (as above), let $(\lambda_j, \mathbf{z}^{(j)})_{j=1, \dots, n}$ denote a set of eigenpairs of the generalized eigenvalue problem*

$$N \mathbf{z}^{(j)} = \lambda_j M \mathbf{z}^{(j)}, \quad (4.13)$$

chosen so that the eigenvectors form an M -orthonormal basis of $\mathbb{C}^n$. This is possible because M is hpd and N is skew-Hermitian (hence iN is Hermitian). Moreover the non-zero eigenvalues are purely imaginary.

Let $\tau > 0$, and consider the space

$$\mathcal{Z} = \text{span}\{\mathbf{z}^{(j)}; |\lambda_j| > \tau\}.$$

Theorem 4.3.8. *Let $\mathcal{Z}$ be as in Definition 4.3.7. Its orthogonal complement is the space $\mathcal{Z}^\perp = \text{span}\{M\mathbf{z}^{(k)}; |\lambda_k| \leq \tau\}$. Moreover, any $\mathbf{y} \in \mathcal{Z}^\perp \setminus \{\mathbf{0}\}$ satisfies*

$$\frac{|\langle A^{-1}\mathbf{y}, \mathbf{y} \rangle|}{\langle M^{-1}\mathbf{y}, \mathbf{y} \rangle} \geq \frac{1}{1 + \tau^2}.$$

Proof. The fact that $\mathcal{Z}^\perp = \text{span}\{M\mathbf{z}^{(k)}; |\lambda_k| \leq \tau\}$, comes from the M -orthonormality of the eigenvectors. Let $\mathbf{y} = M\mathbf{z}^{(k)} \in \mathcal{Z}^\perp$ with $\mathbf{z}^{(k)}$ one of the eigenvectors,

$$|\langle A^{-1}\mathbf{y}, \mathbf{y} \rangle| = |\langle A^{-1}M\mathbf{z}^{(k)}, M\mathbf{z}^{(k)} \rangle| = |\langle (I + M^{-1}N)^{-1}\mathbf{z}^{(k)}, M\mathbf{z}^{(k)} \rangle|$$

since $A^{-1} = (I + M^{-1}N)M^{-1}$. Moreover, by definition of the eigenproblem $(I + M^{-1}N)^{-1}\mathbf{z}^{(k)} = (1 + \lambda_k)^{-1}\mathbf{z}^{(k)}$, so

$$|\langle A^{-1}\mathbf{y}, \mathbf{y} \rangle| = |(1 + \lambda_k)^{-1}\langle \mathbf{z}^{(k)}, M\mathbf{z}^{(k)} \rangle| \geq \frac{1}{1 + \tau} \langle \mathbf{z}^{(k)}, M\mathbf{z}^{(k)} \rangle.$$

To end the proof it suffices to show that this generalizes to linear combinations of vectors $M\mathbf{z}^{(k)} \in \mathcal{Z}^\perp \setminus \{\mathbf{0}\}$ which is skipped. $\square$

Theorem 4.3.9 (Convergence with the new deflation space). *Let us assume that A is positive definite, H is hpd, and $W = H$. With $\mathcal{Z}$ as introduced in Definition 4.3.7, assume either*

1. $\text{range}(Z) = \mathcal{Z}$ and $Y = HAZ$, i.e., P_D is H -orthogonal, or
2. $\text{range}(Y) = \mathcal{Z}$, Y is an invariant subset of HA^* , and $\ker(Y^*) \cap \text{range}(H^{-1}Z) = \{\mathbf{0}\}$.

Then, convergence is bounded with respect to the choice of the threshold τ for selecting eigenvectors as

$$\frac{\|\mathbf{r}_m\|_W^2}{\|\mathbf{r}_0\|_W^2} \leq \left[1 - \frac{\lambda_{\min}(HM)}{\lambda_{\max}(HM)} \times \frac{1}{1 + \tau^2} \right]^{m/2}.$$

Real-valued Case. We consider now the case where A and $\mathbf{b}$ are real-valued. The solution is also real-valued and the iterative solver should be applied in $\mathbb{R}$. In this case, the deflation space from Definition 4.3.7, can be replaced by

$$\mathcal{Z} = \text{span}\{\{\Re(\mathbf{z}^{(k)}), \Im(\mathbf{z}^{(k)})\}; |\lambda_k| > \tau\}.$$

Since $\lambda_k = \pm i\mu$, there are two eigenvectors with $|\lambda_k| = \mu$. Therefore, it suffices to choose the real and imaginary part of one of them to span the same subspace. The Octave command is $\mathbf{Z} = [\text{real}(\mathbf{V}(:, 1:2:m)), \text{imag}(\mathbf{V}(:, 1:2:m))]$, where the columns of $\mathbf{V}$ are the sorted eigenvectors and m is the (even) number of vectors that should form the deflation space.

4.3.4 Numerical illustration: Jordan block

Our problems are real-valued ($\mathbb{K} = \mathbb{R}$), so Hermitian means symmetric. We place ourselves in the framework of Theorem 4.3.9 in the case $Y = HAZ$. Moreover, we apply the strategy just described to stay in $\mathbb{R}$. We start from a zero initial vector, and stop when $\|\mathbf{r}_i\|_H < 10^{-10} \|\mathbf{b}\|_H$.

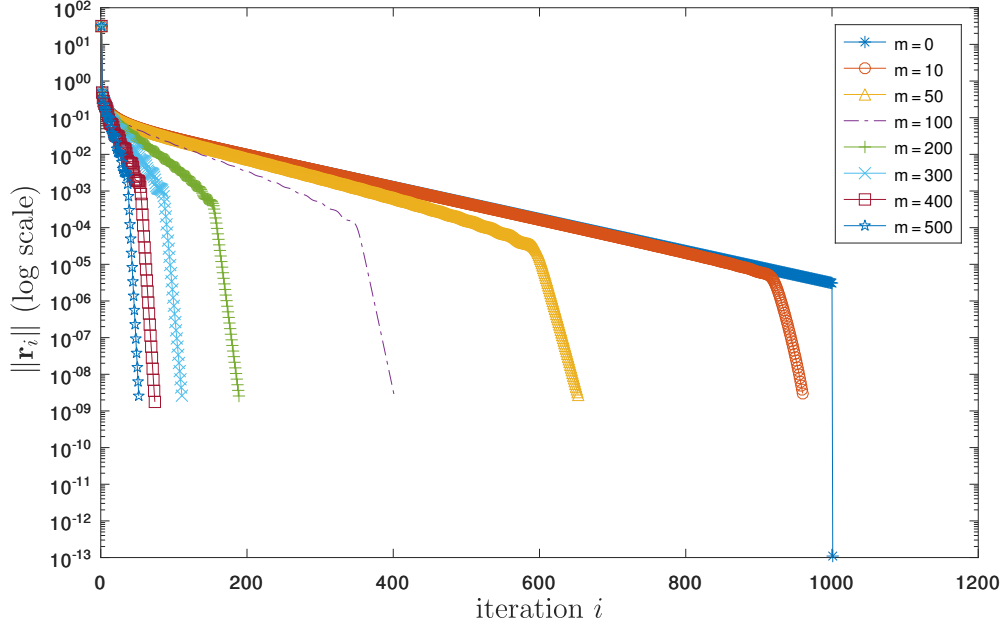


Figure 4.2: Convergence for the scaled Jordan block example. The number m of deflated vectors varies. Increasing m accelerates convergence as predicted by the theory.

Quantities of interest. We first consider an academic problem. Namely, a 1000×1000 (bidiagonal) scaled Jordan block of the form

$$A = \begin{pmatrix} 1 & \alpha & & & \\ & 1 & \alpha & & \\ & & \ddots & \ddots & \\ & & & 1 & \alpha \\ & & & & 1 \end{pmatrix}$$

for which the symmetric part M is a tridiagonal Toeplitz matrix. Choosing $\alpha = 0.99$ guarantees that all eigenvalues of the symmetric part M are positive so A is positive definite; see, e.g., [101, Problem 1.4.P16]. The condition number of the symmetric part is $\kappa(M) = 199$. Our objective is to illustrate the new deflation space. For this reason we do not consider preconditioning or weighting, i.e., $W = H = I$. Figure 4.2 shows the convergence curves obtained without deflation and with deflation of a varying number m of vectors. This confirms that the vectors arising from the generalized eigenvalue problem (4.13) are good candidates for deflation. Deflating 50 eigenvectors decreases the iteration count by 348 and deflating 100 eigenvectors decreases the iteration count by 600. In Table 4.5 we show for various values of m , the dimension of the deflation space, number of iterations to converge, the quantities θ_{exp} calculated from the experiment, and θ_{th} its lower bound from Theorem 4.3.6 (case 1):

$$\theta_{th} = \frac{1}{\kappa(HM)} \times \frac{1}{1 + |\lambda_{m+1}|^2} \text{ and } \theta_{exp} = \min_i \left\{ 1 - \frac{\|\mathbf{r}_{i+1}\|_H^2}{\|\mathbf{r}_i\|_H^2} \right\}.$$

The threshold τ in the theoretical bound has been substituted for $|\lambda_{m+1}|$.

m	iter	θ_{th}	θ_{exp}
0	1000	1.00e-04	1.99e-02
10	959	1.02e-04	1.99e-02
50	652	1.33e-04	1.99e-02
100	400	2.25e-04	1.99e-02
200	188	5.79e-04	2.00e-02
300	110	1.13e-03	1.99e-02
400	73	1.81e-03	2.09e-02
500	51	2.58e-03	2.38e-02

Table 4.5: Scaled Jordan block example. m is the (chosen) number of deflated vectors, iter is the iteration count, θ_{th} and θ_{exp} are the theoretical lower bound and observed worse value of θ .

4.3.5 Numerical illustration: Convection-Diffusion-Reaction

We now solve the same problem as in Section 4.2.4. Since the analysis does not require the mesh to be regular, we discretize the problem with a mesh generated automatically by FreeFem++. Two example meshes are shown in Figure 4.3 with different levels of refinement. They are good representatives of the meshes used throughout our numerical testing. We set $c_0 = \nu = 1$ over the domain and vary the strength of the convective field with a parameter η as follows.

Let $(\phi_i)_{1 \leq i \leq n}$ denote the P_1 finite element basis corresponding to the mesh. Recalling that $\underline{a}$ is divergence-free, the problem matrix splits into

$$A = M + \eta \tilde{N}, \text{ with } M \text{ spd and } \tilde{N} \text{ skew-symmetric,}$$

where the entries of M and $\tilde{N}$ are

$$M_{ij} = \int_{\Omega} (\phi_i \phi_j + \nabla \phi_i \cdot \nabla \phi_j),$$

and

$$\tilde{N}_{ij} = \int_{\Omega} \left(\frac{1}{2} \underline{a} \cdot \nabla \phi_i \phi_j - \frac{1}{2} \underline{a} \cdot \nabla \phi_j \phi_i \right).$$

Choice of preconditioners. For our numerical study, three preconditioners are considered:

- $H = I$ the identity matrix,
- $H = M^{-1}$ the inverse of M , the symmetric part of A , and
- $H = H_{DD}$, the DD preconditioner from Section 4.2.4 (two-level hybrid Additive Schwarz with GenEO coarse space).

The choice of $H = M^{-1}$ corresponds to the CGW method [57, 157, 164].

Deflation Operator. We aim to illustrate the convergence result in Theorem 4.3.9. Once M and N have been assembled by FreeFem++, they are imported into Octave. The generalized eigenvalue problem (4.13) (i.e., $N\mathbf{z}^{(j)} = \lambda_j M\mathbf{z}^{(j)}$) is partially solved by *eigs*: the eigenpairs corresponding to the eigenvalues of largest magnitude are approximated. The problem size is 31 502.

Results for $\eta = 1$. For this test case, we use a mesh with 63 658 triangles and 32 158 vertices. Once the homogeneous Dirichlet boundary condition has been treated by elimination, the problem has 31 502

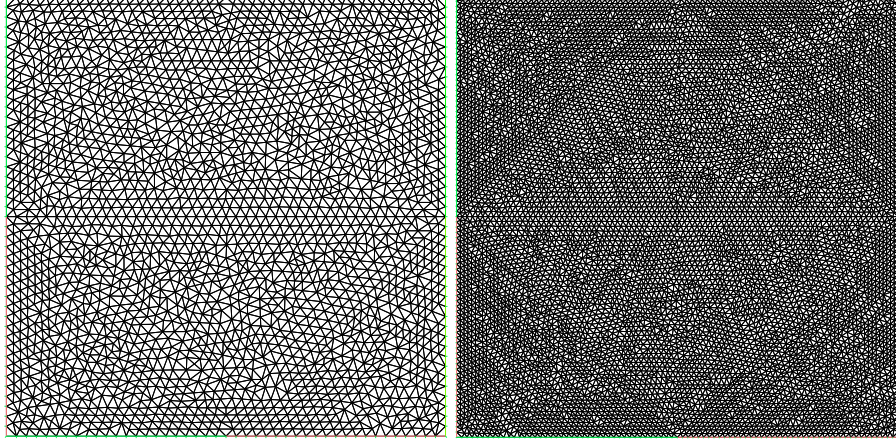


Figure 4.3: Example meshes for the convection-diffusion-reaction problem. Left: 2373 vertices and 4568 triangles. Right: 8643 vertices and 16 948 triangles.

m	0	10	50	100	200	500	1000
$ \lambda_{m+1} $	0.65	0.32	0.18	0.13	0.09	0.06	0.04
θ_{th} for $H = M^{-1}$	0.71	0.91	0.97	0.98	0.991	0.997	0.998
θ_{th} for $H = H_{DD}$	0.043	0.056	0.0597	0.0605	0.0610	0.0614	0.0615
θ_{th} for $H = I$	7.1e-05	9.2e-05	9.8e-05	9.9e-05	1.001e-04	1.007e-04	1.009e-04

Table 4.6: Convection-diffusion-reaction problem. With m as the rank of the deflation space, the second line is the modulus of the first eigenvalue not included in the deflation space, and the next three lines correspond to the theoretical bound θ_{th} for out three choices of preconditioner when $\eta = 1$.

degrees of freedom. The parameter in the convection field is $\eta = 1$ so that $A = M + N$, with $N = \tilde{N}$. We first solve generalized eigenproblem (4.13). The spectrum is represented in Figure 4.4 where the magnitudes of the 3000 largest (purely imaginary) eigenvalues are represented. Unfortunately, the distribution of eigenvalues does not consist in a cluster of large eigenvalues and a tail of eigenvalues tightly clustered around 0. This would indeed have been ideal for selecting a value of τ to compute the deflation space with the formula in Definition 4.3.7. Table 4.6 gives the value of $|\lambda_{m+1}|$ for a few choices of m as well as the resulting bounds for θ_{th} . In these bounds we have injected the approximate condition number $\kappa(HM)$ which takes the following values

$$\kappa(IM) = 9896.4, \quad \kappa(H_{DD}M) = 16.241, \quad \text{and} \quad \kappa(M^{-1}M) = 1.$$

We now solve the problem with all three preconditioners and varying ranks m of the deflation spaces. The case $m = 0$ corresponds to weighted and preconditioned GMRES with no deflation. The largest deflation space has rank 1000 which corresponds to 3.2% of the total number of degrees of freedom (dofs). The convergence curves are shown in Figure 4.5. We observe that our choice of deflation space indeed improves convergence: when the deflation space gets larger, the number of iterations is reduced. Since the case is only mildly nonsymmetric ($\rho(M^{-1}N) = 0.65$), the two *good* preconditioners for M also give good results for the full problem even with $m = 0$. For $H = I$, though, the effect of deflation is welcome: deflating 10 vectors reduces the iteration count from 671 to 543 and deflating 50 vectors reduces the iteration count from 671 to 440.

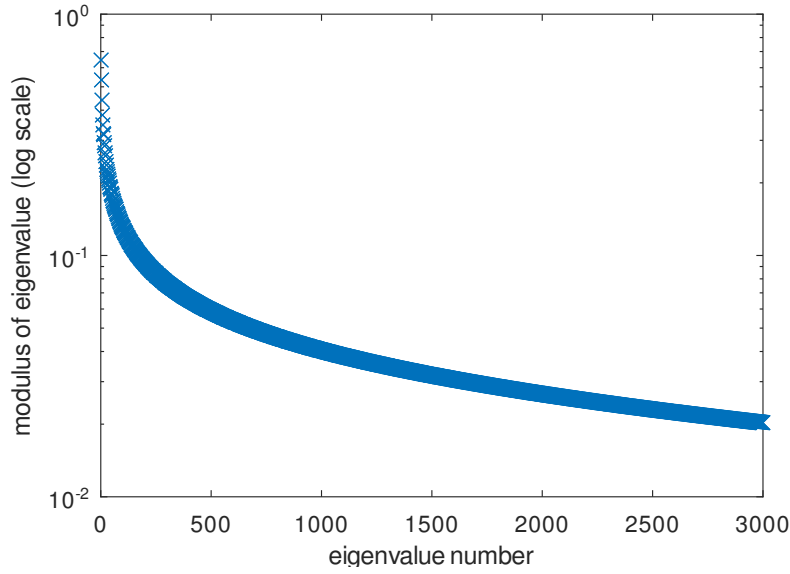
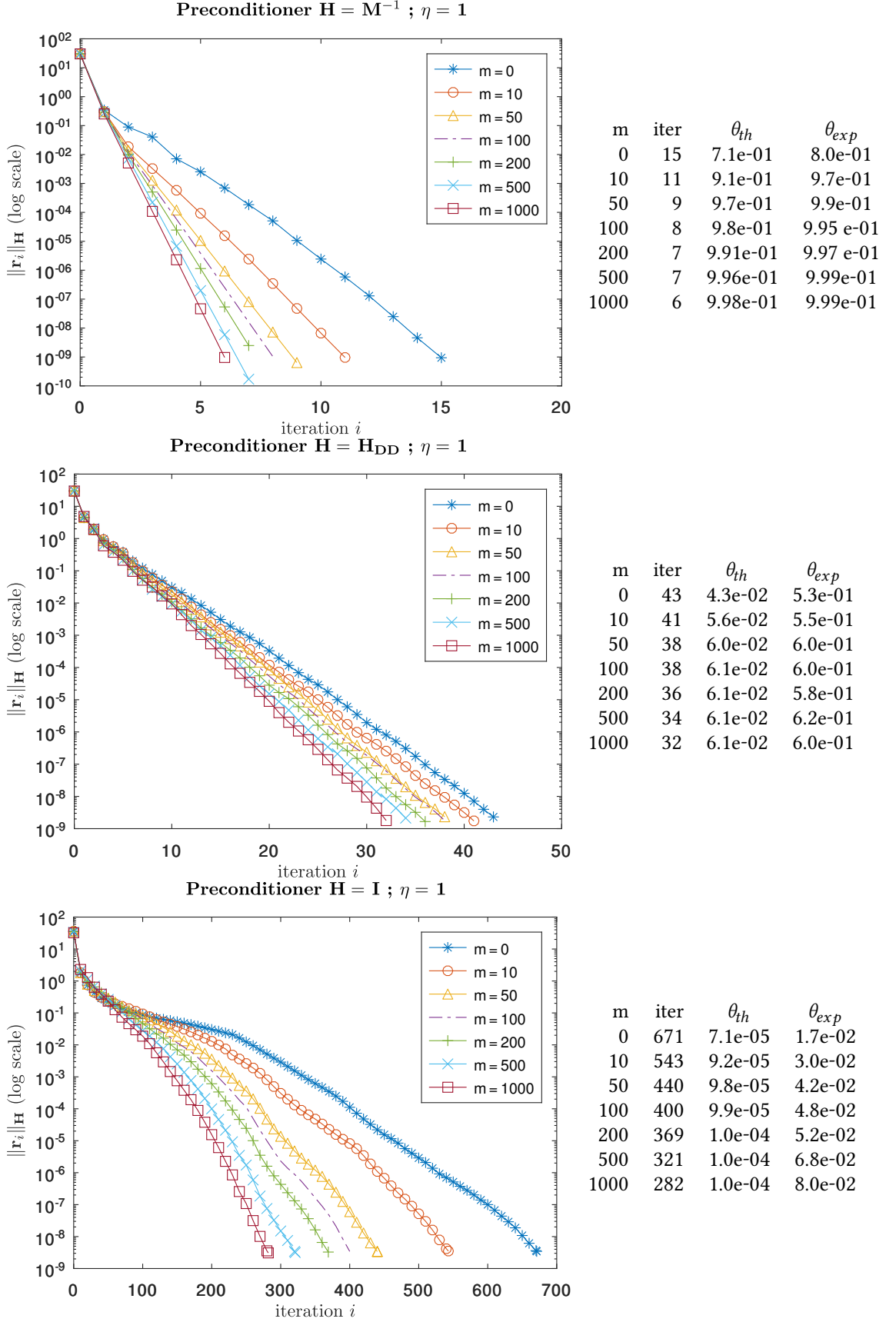
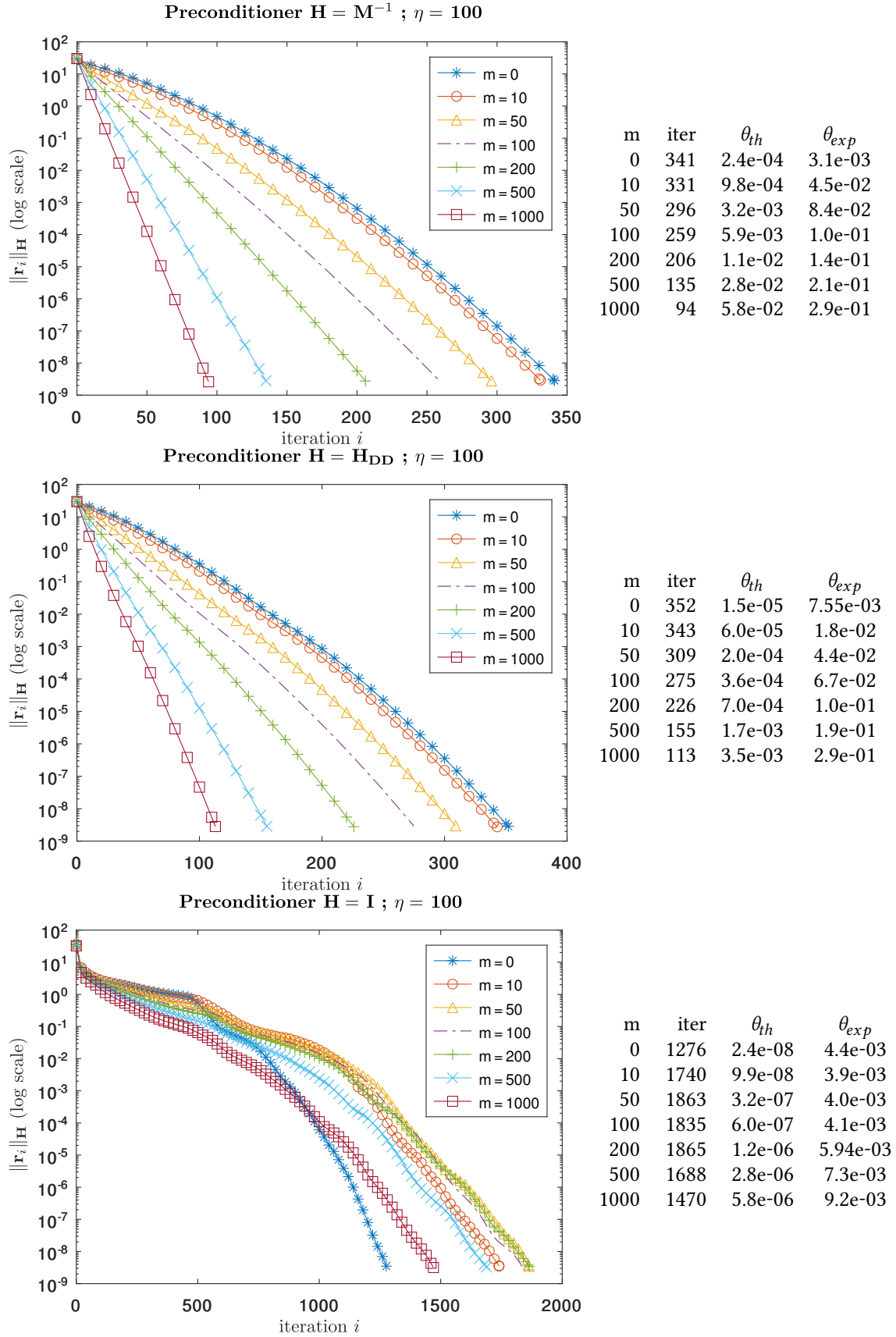


Figure 4.4: Convection-diffusion-reaction problem. $|\lambda_1|$ to $|\lambda_{3000}|$ in log scale (solution of (4.13) for $\eta = 1$). The problem size is 31 502

Results for $\eta = 100$. Next we change the value of η to 100. The global matrix is $A = M + N$ with $N = 100\tilde{N}$, and the problem is much more nonsymmetric. The eigenvalues arising from (4.13) get multiplied by 100 which has the effect of seriously deteriorating the bounds for θ_{th} . Figure 4.6 shows the convergence curves in this case. Again, we confirm that applying a preconditioner tailored for the symmetric part is a good idea (with 341 or 352 iterations instead of 1276). In the non-preconditioned case, it occurs that the non-deflated problem converges the fastest. This is surprising but does not contradict the theory. Figure 4.7 shows the first 1000 residuals in this case for $m = 0$ and $m = 100$. It can be observed that deflation initially accelerates convergence, particularly where it is slowest. However, after roughly 500 iterations, the non-deflated algorithm is faster. For $H = H_{DD}$ or M^{-1} , convergence improves when more vectors are added to the deflation space. With $H = M^{-1}$, deflating 10 vectors reduces the iteration count from 341 to 331 and deflating 100 vectors reduces it further to 259. With $H = H_{DD}$, deflating 10 vectors reduces the iteration count from 352 to 343 and deflating 100 vectors reduces it further to 275.

Influence of the mesh. We consider the case where $\eta = 100$ and $H = H_{DD}$ with varying mesh size. The stopping criterion is as before, $\|\mathbf{r}_i\|_H / \|\mathbf{r}_0\|_H = \|\mathbf{r}_i\|_H / \|\mathbf{b}\|_H < 10^{-10}$. The three considered meshes are the 32158 vertex mesh that was used in the tests above as well as the two less refined meshes represented in Figure 4.3 (2373 and 8643 vertices). After elimination of the degrees of freedom on the boundary, the resulting linear systems have respectively 31 502, 8307 and 2197 dofs. The problem is solved by WPD-GMRES without deflation ($m = 0$) and with deflation of $m = 100$ vectors. The results are presented in Table 4.7. We notice that the iteration counts increase weakly with the mesh size. For example, from 236 for 2,197 dofs to 275 for 32,502 dofs in the case where 100 vectors are deflated. We also report on the quantities that appear in the convergence bounds: $\kappa(H_{DD}M)$, $|\lambda_1| = \rho(M^{-1}N)$ and $|\lambda_{101}|$. We know from the GenEO theory that $\kappa(H_{DD}M)$ is bounded independently of the mesh size and we indeed notice that it does not depend very much on the mesh. In (4.8) it was stated (and proved in [7, Section 5]) that for this particular PDE, $|\lambda_1| = \rho(M^{-1}N)$ is also bounded independently of the mesh size. The bound is satisfied here and $\rho(M^{-1}N)$ does not depend on the mesh. For $|\lambda_{101}|$, we have no theoretical results (except of course that $|\lambda_{101}| \leq \rho(M(A)^{-1}N(A))$) and we observe a small increase when the number of

Figure 4.5: Convergence curves when $\eta = 1$ with varying preconditioner and rank of deflation space.

Figure 4.6: Convergence curves when $\eta = 100$ with varying preconditioner and rank of deflation space.

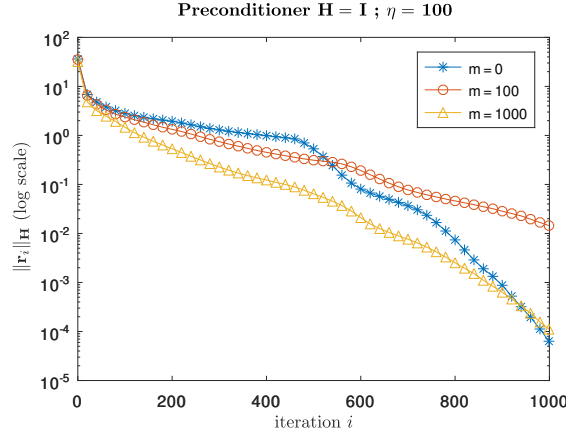


Figure 4.7: Zoom on the first 1000 iterations in the case $H = I$ and $\eta = 100$.

# dofs	Iter		$\kappa(H_{\text{DD}}M)$	$ \lambda_1 = \rho(M^{-1}N)$	$ \lambda_{101} $
	m = 0	m = 100			
2,197	316	236	14.4	64.4	11.5
8,307	343	267	14.2	64.5	12.7
32,502	352	275	13.0	64.6	13.0

Table 4.7: For $\eta = 100$ and $H = H_{\text{DD}}$, three mesh sizes are considered.

dofs increases.

More numerical results for a Stokes problem are presented in [8].

4.4 Scope and impact of this work

We have proved a very general convergence bound for preconditioned GMRES when any inner product is used, and when it is deflated. This bound inspired us to produce an effective deflation space, namely, the eigenvectors of the generalized eigenvalue problem $Nz = \lambda Mz$, where M is the Hermitian part of A , which is assumed to be positive definite, and N is the skew-Hermitian part of A . Numerical experiments illustrate the potential for these ideas.

The impact of this work is two-fold. First, there is a concrete strategy of preconditioning and deflation for positive definite problems. Second, the new analysis of GMRES could have a strong impact in itself since this is a very active area of research and has been for decades. This contribution has already received interest from the domain decomposition and numerical linear algebra communities in the form of invitations.

Chapter 5

Conclusions and Perspectives

5.1 Conclusions

Krylov subspace solvers are a very diverse family of methods, even if one restricts themselves to methods that minimize some residual at each iteration¹. Indeed, on one hand, changing the preconditioner modifies the minimization space. On the other hand, weighting the inner product modifies the norm in which the minimization occurs. Deflation is almost a form of cheating: a well-identified part of the problem is solved directly. To make it fair, the deflation procedure must stay cheap relative to the overall computation time. Other modifications, such as block methods and multipreconditioned methods, make the scope of Krylov subspaces even larger.

Through this manuscript, I have tried to convey that my work started with one particular solver, PCG, preconditioned by DD with the GenEO coarse space and expanded in many different directions: algebraic solvers, non-Hermitian solvers, non-Hermitian preconditioners, multipreconditioners, block Krylov subspace methods, general results for GMRES, acceleration of GMRES by Hermitian preconditioning and by spectral deflation. Some of my contributions have been theoretical: introduction of new methods and their analysis. Other contributions are numerical: development of an open source code, application of the new methods to problems from structural mechanics, transfer to the industry. I find the interface between applied mathematics, scientific computing and high performance computing to be a never-ending source of challenges, even on such a well-established problem as solving a linear system. My best attempt at summarizing my contributions is Figure 5.1.

[Abstract GenEO] The strategy for constructing the GenEO coarse space, which is presented in its most abstract form in Section 2.2 of this manuscript, is the following. First, the theory of PCG preconditioned by DD is pushed as far as possible. This means up to the point where the assumptions required for the analysis to go through are too stringent for the diversity of problems addressed. This is where the generalized eigenvalue problem comes from. Instead of proving a result, a threshold is set for a well identified spectral equivalence property. The generalized eigenvalue problem computes the space on which the estimate does not hold. The problematic eigenvectors are grouped into the coarse space. This has had an impact on the community of DD and beyond. However, as with any method, there are limitations to GenEO. I see three: it applies only to spd problems, the setup time can become prohibitive, and a set of local spsd matrices from which A is assembled must be known.

[AMPCG] Multipreconditioning is a different way of accelerating convergence of a Krylov subspace method by adding N search directions to the minimization space simultaneously. The combination of multipreconditioning and DD is natural by considering each local solve in the DD preconditioner to be its

¹which I have done in this manuscript except for the brief appearance of biCG in Section 3.4.1.

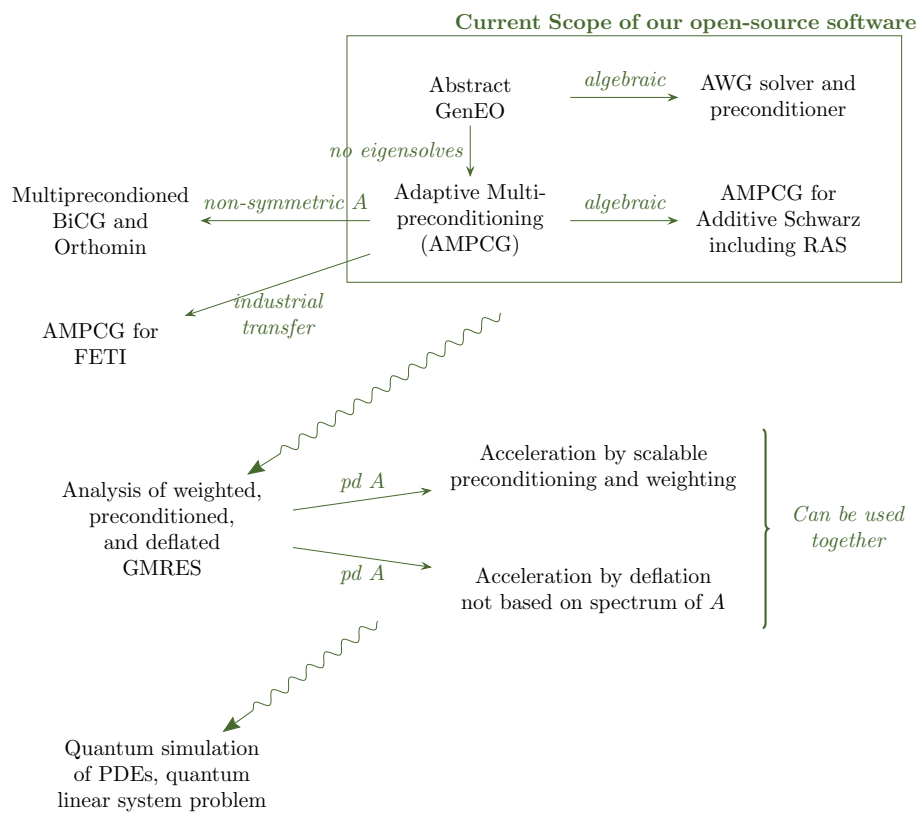


Figure 5.1: Summary of my research since my PhD

own preconditioner. In [3] (mentioned in Section 3.1), multipreconditioned FETI is introduced (under the name Simultaneous FETI). Quite significant work goes into reorganizing the operations in order to limit the computational cost of handling many search directions. It is observed that multipreconditioned FETI is scalable in terms of iteration count, and robust on the same class of problems as GenEO: heterogeneous coefficients, jagged interfaces between subdomains...The problem is that there is no short recurrence property for multipreconditioned CG so iterations become more and more expensive as the algorithm progresses (because of orthogonalization).

Motivated by the fact that, *via* GenEO, deflating a small coarse space is often enough to accelerate convergence, I introduced the adaptive multipreconditioned CG algorithm (Section 3.2). The idea is to sort through the multitude of search directions proposed by multipreconditioned CG and keep only the ones that accelerate convergence. A test, called the τ -test and based on the convergence theory of PCG, is performed at each iteration to either precondition classically or multiprecondition. This method has had quite a significant impact and I was awarded the Leslie Fox Prize for the AMPCG paper and presentation that I gave of it. AMPCG combined with FETI is applied frequently in structural mechanics, which was initiated in [2], a flavour of which can be found in Section 3.2.5.

[AMPCG for RAS, Orthomin and BiCG] Of course, as a scientist and a mathematician, I thrive to solve even more problems, and to weaken the assumptions of every result. So, I considered applying adaptive multipreconditioning to non-symmetric preconditioners and non-symmetric problems (Sections 3.3 and 3.4). The combination of AMPCG and the additive Schwarz preconditioners was slightly dissatisfying from the theoretical point of view because the proposed τ -test is not based on the CG convergence theory. This is because the τ -test in AMPCG requires the knowledge of the smallest eigenvalue of the preconditioned operator but, with additive Schwarz, the spectrum is naturally bounded only from above. In fact, there is no algebraic DD method for which the spectrum is naturally bounded from below. This encouraged me to develop a new algebraic DD method with a controlled spectrum, and this is how I arrived at the AWG solver presented in Section 2.3.

[AWG solver and preconditioner] In order to make GenEO algebraic, the assumption that needs weakening is that A is assembled from a sum of known local spsd matrices. The way that I found around this, is to build an auxiliary problem to which GenEO applies and then go back to the original problem by the Woodbury matrix identity (hence the name AWG for Algebraic Woodbury GenEO). The AWG work is made possible by the very abstract framework for GenEO.

[Analysis of GMRES] The next leap that I have taken is to move from spd problems to general linear systems. In order to do this, the natural solver is GMRES. My intention was to repeat the GenEO procedure by taking the DD theory as far as possible before applying deflation. However, I have found that there is no well-established abstract Schwarz (or DD) theory for non-symmetric problems. In fact, even the theory of GMRES is still an active field of research: for general problems, it is not known which is the full set of properties of a (possibly preconditioned) matrix that make it easy or hard to solve by GMRES. This was the motivation to take a step back and work not on DD preconditioning, but on convergence of GMRES.

[Acceleration of GMRES] My idea is to study preconditioned, deflated and weighted GMRES in a way that makes apparent the influence of each. The first significant results are for positive definite matrices and hpd preconditioning (Section 4.2). They have already led to a scalable strategy for non-symmetric problems. The new deflation space that I proposed with Daniel Szyld (Section 4.3) is a rupture with existing deflation methods that are usually based on the spectrum of A . This result could, in itself, have a strong impact in the numerical linear algebra community. It will also serve as the starting point for an abstract DD (or multigrid, or multilevel Krylov, ...) theory in the linear solver community.

[Quantum Simulation of PDEs] Finally, in very recent work (not presented in the manuscript), I have begun to explore the realm of quantum linear system problems (QLSP):

$$A|x\rangle = |b\rangle.$$

Compared to the classical linear system $A\mathbf{x} = \mathbf{b}$, there are several major differences. First of all, quantum states are normalized, so $|x\rangle$ and $|b\rangle$ are not direct representations of $\mathbf{x}$ and $\mathbf{b}$ but they are proportional to these vectors. What this means, is that the entries in $\mathbf{x}/\|\mathbf{x}\|$ and $\mathbf{b}/\|\mathbf{b}\|$ are encoded in the amplitudes of $|x\rangle$ and $|b\rangle$. A second difference is that it must be assumed that the action of A can be encoded. Finally, the output of a QLSP solver is not the knowledge of all entries of $\mathbf{x}$ (even up to a scaling constant). Measurements must be performed in order to retrieve scalar quantities from $|x\rangle$. Quantum computers have shown promise with exponential speedup for some classes of linear system problems and their range is quickly evolving. Quantum algorithms rely very heavily on numerical linear algebra which is my area of expertise. Existing quantum linear solvers are based on ideas such as diagonalization (HHL), the singular value decomposition, or even Krylov subspace methods. In the current state of the art, accelerators such as preconditioners are definitely a next step. With my collaborators, Elise Fressart and Michel Nowak, we have not yet considered the acceleration of quantum linear solvers. Instead, we have addressed how an efficient linear solver (VQLS in our application) could be taken advantage of in a full simulation loop where the linear solve is applied at every iteration of an outer mesh refinement loop. We have proposed a technique to compute a posteriori estimates in the quantum setting. Our first results have recently been submitted and this is very much work to be continued [9].

5.2 Perspectives – Ongoing Work

To begin with perspectives, I present three ongoing projects.

Sequences of systems. Spectral coarse spaces and adaptive multipreconditioning are two state of the art options for solving spd linear systems that arise from PDEs. There are many cases in which these spd systems arise, not by themselves, but as part of a sequence. This is the case, for example, if the linear solves correspond to different discretization steps of a Newton method for a non-linear problem. This is also the case if the systems arise from a discretization in time (a problematic raised in [10]), or a shape optimization loop. Finally, sampling of PDEs with stochastic coefficients also produces large numbers of systems. In collaboration with Olivier Le Maître (CNRS, CMAP, École polytechnique and Inria), one of the objectives of ANR project DARK is to propose an optimal solver for whole sequence of systems. Funding for a PhD student to work on this topic has been secured through ANR DARK.

Analysis of weighted GMRES. It has been observed that the convergence of weighted GMRES is often similar to the convergence of GMRES. That is, when GMRES converges fast in one norm, it does in other norms too. As an illustration of this, I include one more numerical result for the problem presented in Section 4.2.4. The same problem is solved twice: once by W -GMRES with the spectral deflation space, preconditioned and weighted by H_{DD} and once by GMRES with the spectral deflation space, preconditioned on the left by H_{DD} . In both cases, the stopping criterion is set to $\|H\mathbf{r}_i\|/\|H\mathbf{r}_0\| = \|H\mathbf{r}_i\|/\|H\mathbf{b}\| < 10^{-10}$ where $\|\cdot\|$ is the Euclidean norm. As an illustration, we choose the problem with 8,307 dofs and set $H = H_{DD}$. Problems that are weakly or strongly non-symmetric ($\eta \in \{0.1, 1, 10, 100\}$) are solved without deflation and with deflation of 100 vectors and of 1000 vectors. The iteration counts can be found in Table 5.1. In Figure 5.2 we plot the history of residual $\|H\mathbf{r}_i\|/\|H\mathbf{r}_0\|$ for $\eta = 100$. We observe a similarity up to a point that has me very intrigued.

In collaboration with Pierre Matalon (CMAP, École polytechnique), we have set out to understand the exact connection between the residuals produced by GMRES in different inner products. A first result [14] is the following.

Theorem (Simultaneous prescription of two convergence curves). *Let W be a hpd matrix, and consider two prescribed convergence curves: $r_0 > r_1 > r_2 > \dots$ and $r'_0 > r'_1 > r'_2 > \dots$. The following statements are equivalent:*

η	$m = 0$		$m = 100$	
	$W = I$	$W = H$	$W = I$	$W = H$
0.1	36	37	33	33
1	40	40	33	34
10	89	90	54	54
100	327	329	253	255

Table 5.1: Iteration counts for weighted and unweighted algorithms. Both algorithms use the stopping criterion coming from the unweighted algorithm.

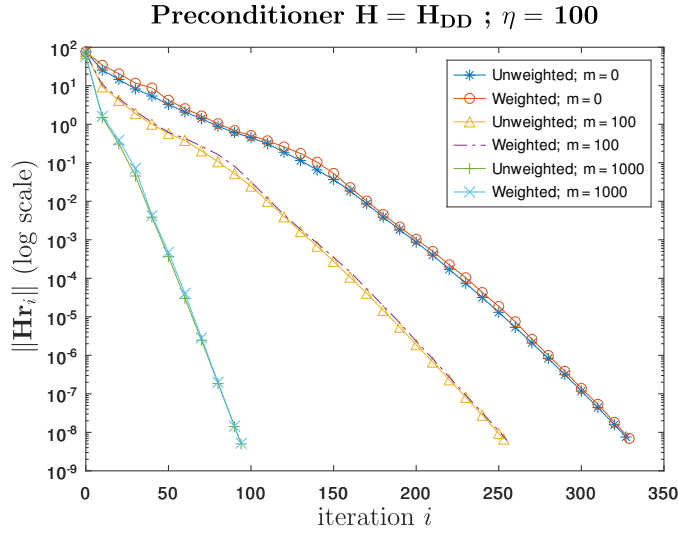


Figure 5.2: Comparison of weighted and unweighted algorithms, without deflation and with deflation of 100 vectors and of 1000 vectors. The displayed norm $\|Hr_i\|$ is the one that is minimized by the unweighted algorithm.

1. There exists a system (A, b) such that I -GMRES(A, b) produces residuals r_i and W -GMRES(A, b) produces residuals r'_i .
2. There exists a non-singular triangular matrix T such that the singular values of T are the inverse square roots of the eigenvalues of W , and

$$\begin{pmatrix} \sqrt{r_0^2 - r_1^2} \\ \sqrt{r_1^2 - r_2^2} \\ \vdots \end{pmatrix} = T \begin{pmatrix} \sqrt{r_0'^2 - r_1'^2} \\ \sqrt{r_1'^2 - r_2'^2} \\ \vdots \end{pmatrix}.$$

Moreover, the eigenvalues of A can be freely prescribed.

Computationally efficient deflation of GMRES. My recent work on GMRES with Daniel Szyld (Temple University) has opened up many new perspectives for DD preconditioning and deflation of non-spd problems. In the work presented in this manuscript, the spectral deflation space consists of high-frequency eigenvectors of $M^{-1}N$ (where $A = M + N$ is the Hermitian + skew-Hermitian splitting of A). This is expensive to compute if solving problems for M is costly. A consequence is that, even though

the proposed spectral deflation space has the advantage of improving robustness, it does not yet allow to solve problems very fast. In even more recent work [15], we propose polynomial bounds for GMRES and a deflation space that no longer requires applications of M^{-1} .

Theorem. Consider W -weighted GMRES applied to $A\mathbf{x} = \mathbf{b}$, split-preconditioned by (H_L, H_R) and deflated by $P_D = I - AZ(Y^*AZ)^{-1}Y^*$. We make the four assumptions that A is pd, $H := H_R H_L$ is hpd, $H = H_L^* W H_L$ and $Y = HAZ$.

Let $(\lambda^{(k)}, \mathbf{x}^{(k)})_{1 \leq k \leq n}$ be the eigenpairs of the generalized eigenvalue problem

$$N\mathbf{x}^{(k)} = \lambda^{(k)} H^{-1} \mathbf{x}^{(k)}.$$

If, for a given $\tau > 0$, the columns of Y are set to be the vectors $\{\mathbf{x}^{(k)}; |\lambda^{(k)}| > \tau\}$, then the k -th residual is bounded by

$$\frac{\|\mathbf{r}_k\|_H}{\|\mathbf{r}_0\|_H} \leq (1 + \sqrt{2}) \min_{q \in \mathbb{P}_k; q(0)=1} \max_{z \in \Omega_1^c} |q(z)|,$$

where

$$\Omega_1^c := [\lambda_{\min}(HM), \lambda_{\max}(HM)] + i[-\tau, \tau].$$

with $[\lambda_{\min}(HM), \lambda_{\max}(HM)]$ a real positive interval that contains all eigenvalues of HM .

5.3 Perspectives – Open Problems

The long-term ambition with respect to linear solvers is to propose a fully algebraic solver that is scalable and robust enough to solve any linear system. This is not as completely out of reach as may appear at first. Krylov subspace methods offer a very diverse set of methods, and Krylov accelerators are even more varied. There have been contributions (including my own) that are scalable, algebraic, robust, or that apply to non-Hermitian problems. It remains to propose a generalization that is all of these things at once. With the knowledge and perspective that I have gained from my work, I present three steps in the direction of the silver bullet that solves any $A\mathbf{x} = \mathbf{b}$.

Analysis of indefinite Krylov Subspace Methods. The GMRES bounds that provide the most information for preconditioning, deflation and weighting rely on the field of values. If it contains the origin, the resulting bound no longer gives any information. In recent work [70], Embree has shown that as long as all eigenvalues are in a half-plane that does not contain the origin, a weight can be defined such that the weighted field of values does not include zero. This is already a significant generalization. However, indefinite problems require convergence results based on different quantities and this is still an open problem.

Inexactness. In the manuscript, I have focused on exact arithmetic and exact solution of local problems, coarse problems, and eigenvalue problems. In real-life none of these operations are performed up to a zero tolerance and it is necessary to introduce uncertainty into the analysis in order to design the most efficient algorithms. This is all the more important as supercomputers have more and more levels of parallelism and are accelerated by GPUs.

Algebraic solvers that have eigenvalues close to zero. Computing, or deflating, large frequency eigenvectors is in general much less computationally expensive than doing so for eigenvalues that are close to zero. However, solvers that are algebraic (e.g. Additive Schwarz preconditioning) have eigenvalues that are not bounded away from zero. This is the limitation encountered with adaptive

multipreconditioning applied to Additive Schwarz. It is my opinion that development of algebraic solvers should go in that spectral direction in order to make a significant contribution.

Beyond linear systems. As an additional perspective, I would like to mention eigensolvers. Indeed solving eigenproblems has many similarities with solving linear systems and is a perfect next step for my research. One difference is that eigensolvers don't have right hand sides and the solutions are invariant subspaces.

Beyond classical computers. Taking advantage of quantum computers for numerical simulations, including solving linear systems is a very appealing idea. Although the hardware is not widely available yet, it is already foreseen what some of the characteristics of quantum algorithms should be. Quantum algorithms are a new and exciting era of linear solvers.

Chapter 6

Bibliography

6.1 List of my publications

Peer-reviewed journal articles (after PhD)

- [1] C. Bovet, P. Gosselet, and N. Spillane. Multipreconditioning for nonsymmetric problems: the case of orthomin and biCG. *C. R., Math., Acad. Sci. Paris*, 355(3):354–358, 2017. doi: 10.1016/j.crma.2017.01.010. <https://hal.science/hal-01348633v2/document>.
- [2] C. Bovet, A. Parret-Fréaud, N. Spillane, and P. Gosselet. Adaptive multipreconditioned FETI: scalability results and robustness assessment. *Computers & Structures*, 193:1–20, 2017. doi: <https://doi.org/10.1016/j.compstruc.2017.07.010>. <https://hal.science/hal-01458725/document>.
- [3] P. Gosselet, D. Rixen, F.-X. Roux, and N. Spillane. Simultaneous FETI and block FETI: robust domain decomposition with multiple search directions. *Int. J. Numer. Methods Eng.*, 104(10):905–927, 2015. doi: 10.1002/nme.4946. <https://hal.science/hal-01056928/document>.
- [4] L. Gouarin and N. Spillane. Fully algebraic domain decomposition preconditioners with adaptive spectral bounds. *ETNA, Electron. Trans. Numer. Anal.*, 60:169–196, 2024. doi: 10.1553/etna_vol60s169. <https://hal.science/hal-03258644/document>.
- [5] N. Spillane. An adaptive multipreconditioned conjugate gradient algorithm. *SIAM J. Sci. Comput.*, 38(3):a1896–a1918, 2016. doi: 10.1137/15M1028534. <https://hal.science/hal-01170059/document>.
- [6] N. Spillane. GenEO spectral coarse spaces in SPD domain decomposition. *Numerical Algorithms*, 2025. doi: 10.1007/s11075-025-02166-x. <https://hal.science/hal-03186276/document>.
- [7] N. Spillane. Hermitian preconditioning for a class of non-Hermitian linear systems. *SIAM J. Sci. Comput.*, 46(3):a1903–a1922, 2024. doi: 10.1137/23M1559026. <https://hal.science/hal-04028590v2/document>.
- [8] N. Spillane and D. B. Szyld. New convergence analysis of GMRES with weighted norms, preconditioning, and deflation, leading to a new deflation space. *SIAM J. Matrix Anal. Appl.*, 45(4):1721–1745, 2024. doi: 10.1137/23M1622398. <https://hal.science/hal-04328504/document>.

Peer-reviewed conference proceedings (after PhD)

- [9] E. Fressart, M. Nowak, and N. Spillane. Adaptive mesh refinement quantum algorithm for Maxwell's equations. In *2025 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2025. To appear. <https://arxiv.org/abs/2504.01646>.
- [10] P. Hauret, E. Lignon, N. Spillane, and B. Pouliot. Two-scale space-time methods for computational solid mechanics. In *Proceedings of the 7th ECM (7ECM) congress, Berlin, Germany, July 18–22, 2016*, pages 777–794. Zürich: European Mathematical Society (EMS), 2018. doi: 10.4171/176-1/34. <https://ems.press/content/book-chapter-files/21791>.
- [11] N. Spillane. Algebraic adaptive multipreconditioning applied to restricted additive Schwarz. In *Proceedings of the 23rd international conference on domain decomposition methods, Jeju Island, South Korea, July 6–10, 2015*, pages 93–104. Cham: Springer, 2017. doi: 10.1007/978-3-319-52389-7_8. <https://hal.science/hal-01276446/document>.
- [12] N. Spillane. Toward a new fully algebraic preconditioner for symmetric positive definite problems. In *Proceedings of the 26th international conference on domain decomposition methods, virtual, December 7–12, 2020*, pages 745–752. Cham: Springer, 2022. doi: 10.1007/978-3-030-95025-5_81. <https://hal.science/hal-03187092/document>.

Open Source Code (after PhD)

- [13] L. Gouarin and N. Spillane. GenEO Software, 2021–2024. <https://github.com/hpc-maths/GenEO>.

Preprints (after PhD)

- [14] P. Matalon and N. Spillane. Any nonincreasing convergence curves are simultaneously possible for GMRES and weighted GMRES, as well as for left and right preconditioned GMRES. <https://arxiv.org/abs/2506.17193>, 2025.
- [15] N. Spillane and D. B. Szyld. Improved Polynomial Bounds and Acceleration of GMRES by Solving a min-max Problem on Rectangles, and by Deflating. <https://hal.science/hal-05022856>, 2025.

Publications from my PhD

- [16] V. Dolean, F. Nataf, R. Scheichl, and N. Spillane. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. *Comput. Methods Appl. Math.*, 12(4):391–414, 2012. <https://hal.science/hal-00491919/document>.
- [17] F. Nataf, H. Xiang, V. Dolean, and N. Spillane. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.*, 33(4):1623–1642, 2011. <https://hal.science/hal-00491919/document>.
- [18] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer. Math.*, 126(4):741–770, 2014. doi: 10.1007/s00211-013-0576-y. <https://people.bath.ac.uk/masrs/2011-07.pdf>.

- [19] N. Spillane and D. J. Rixen. Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. *Int. J. Numer. Methods Eng.*, 95(11):953–990, 2013. doi: [10.1002/nme.4534](https://doi.org/10.1002/nme.4534). <https://hal.science/hal-00756994/document>.
- [20] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. A robust two-level domain decomposition preconditioner for systems of PDEs. *C. R., Math., Acad. Sci. Paris*, 349(23–24):1255–1259, 2011. doi: [10.1016/j.crma.2011.10.021](https://doi.org/10.1016/j.crma.2011.10.021). https://people.bath.ac.uk/masrs/CRAS_SDHNPS_2011.pdf.

6.2 Bibliography

- [21] E. Agullo, L. Giraud, and L. Poirel. Robust preconditioners via generalized eigenproblems for hybrid sparse linear solvers. *SIAM Journal on Matrix Analysis and Applications*, 40(2):417–439, 2019.
- [22] H. Al Daas, P. Jolivet, and T. Rees. Efficient algebraic two-level Schwarz preconditioner for sparse matrices. *SIAM J. Sci. Comput.*, 45(3):a1199–a1213, 2023. doi: [10.1137/22M1469833](https://doi.org/10.1137/22M1469833).
- [23] P. Amestoy, A. Buttari, J.-Y. L’Excellent, and T. Mary. Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures. *ACM Transactions on Mathematical Software*, 45:2:1–2:26, 1, 2019.
- [24] P. Amestoy, I. S. Duff, J. Koster, and J.-Y. L’Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.
- [25] P. R. Amestoy, I. S. Duff, J.-Y. L’Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001. doi: [10.1137/S0895479899358194](https://doi.org/10.1137/S0895479899358194).
- [26] O. Axelsson and I. Kaporin. Error norm estimation and stopping criteria in preconditioned conjugate gradient iterations. *Numer. Linear Algebra Appl.*, 8(4):265–286, 2001. doi: [10.1002/nla.244](https://doi.org/10.1002/nla.244).
- [27] Z.-Z. Bai, G. H. Golub, and M. K. Ng. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM J. Matrix Anal. Appl.*, 24(3):603–626, 2003. doi: [10.1137/S0895479801395458](https://doi.org/10.1137/S0895479801395458).
- [28] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Users Manual. Technical report ANL-95/11 - Revision 3.15, Argonne National Laboratory, 2021.
- [29] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Web page. <https://www.mcs.anl.gov/petsc>, 2021.
- [30] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
- [31] B. Beckermann, S. A. Goreinov, and E. E. Tyrtyshnikov. Some remarks on the Elman estimate for GMRES. *SIAM J. Matrix Anal. Appl.*, 27(3):772–778, 2006. doi: [10.1137/040618849](https://doi.org/10.1137/040618849).

- [32] M. Benzi. Preconditioning techniques for large linear systems: A survey. *J. Comput. Phys.*, 182(2):418–477, 2002. DOI: [10.1006/jcph.2002.7176](https://doi.org/10.1006/jcph.2002.7176).
- [33] J. Besson, G. Cailletaud, J.-L. Chaboche, and S. Forest. *Non-Linear Mechanics of Materials*, volume 167 of *Solid Mechanics and Its Applications*. Springer Netherlands, 2010. DOI: [10.1007/978-90-481-3356-7](https://doi.org/10.1007/978-90-481-3356-7).
- [34] M. Bonazzoli, X. Claeys, F. Nataf, and P.-H. Tournier. How does the partition of unity influence SORAS preconditioner? In *Proceedings of the 27th international conference on domain decomposition methods, Prague, Czech Republic, July 25–29, 2022*, pages 61–68. Cham: Springer, 2024. DOI: [10.1007/978-3-031-50769-4_6](https://doi.org/10.1007/978-3-031-50769-4_6).
- [35] M. Bonazzoli, X. Claeys, F. Nataf, and P.-H. Tournier. Analysis of the SORAS domain decomposition preconditioner for non-self-adjoint or indefinite problems. *J. Sci. Comput.*, 89(1):27, 2021. DOI: [10.1007/s10915-021-01631-8](https://doi.org/10.1007/s10915-021-01631-8). Id/No 19.
- [36] N. Bootland, V. Dolean, I. G. Graham, C. Ma, and R. Scheichl. GenEO coarse spaces for heterogeneous indefinite elliptic problems. In *Proceedings of the 26th international conference on domain decomposition methods, virtual, December 7–12, 2020*, pages 117–125. Cham: Springer, 2022. DOI: [10.1007/978-3-030-95025-5_10](https://doi.org/10.1007/978-3-030-95025-5_10).
- [37] N. Bootland, V. Dolean, I. G. Graham, C. Ma, and R. Scheichl. Overlapping Schwarz methods with GenEO coarse spaces for indefinite and nonself-adjoint problems. *IMA J. Numer. Anal.*, 43(4):1899–1936, 2023. DOI: [10.1093/imanum/drac036](https://doi.org/10.1093/imanum/drac036).
- [38] A. Brandt, J. Brannick, K. Kahl, and I. Livshits. Bootstrap AMG. *SIAM J. Sci. Comput.*, 33(2):612–632, 2011. DOI: [10.1137/090752973](https://doi.org/10.1137/090752973).
- [39] A. Brandt. Algebraic multigrid (amg) for sparse matrix equations. *Sparsity and its Applications*:257–284, 1984.
- [40] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.
- [41] M. Brezina, C. Heberton, J. Mandel, and P. Vaněk. An iterative method with convergence rate chosen a priori. Technical report 140, University of Colorado Denver, Apr. 1999.
- [42] C. Brezinski. Multiparameter descent methods. *Linear Algebra Appl.*, 296(1-3):113–141, 1999. DOI: [10.1016/S0024-3795\(99\)00112-3](https://doi.org/10.1016/S0024-3795(99)00112-3).
- [43] R. Bridson and C. Greif. A multipreconditioned conjugate gradient algorithm. *SIAM J. Matrix Anal. Appl.*, 27(4):1056–1068 (electronic), 2006. DOI: [10.1137/040620047](https://doi.org/10.1137/040620047).
- [44] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, 2000.
- [45] P. N. Brown and H. F. Walker. GMRES on (nearly) singular systems. *SIAM J. Matrix Anal. Appl.*, 18(1):37–51, 1997. DOI: [10.1137/S0895479894262339](https://doi.org/10.1137/S0895479894262339).
- [46] K. Burrage, J. Erhel, B. Pohl, and A. Williams. A deflation technique for linear systems of equations. *SIAM J. Sci. Comput.*, 19:1245–1260, 1998.
- [47] X.-C. Cai. *Some domain decomposition algorithms for nonselfadjoint elliptic and parabolic partial differential equations*. PhD thesis, New York University, 1989.
- [48] X.-C. Cai, W. D. Gropp, and D. E. Keyes. A comparison of some domain decomposition algorithms for nonsymmetric elliptic problems. In *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA*, 1992.
- [49] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21(2):792–797 (electronic), 1999. DOI: [10.1137/S106482759732678X](https://doi.org/10.1137/S106482759732678X).

- [50] X.-C. Cai and O. B. Widlund. Domain decomposition algorithms for indefinite elliptic problems. *SIAM J. Sci. Stat. Comput.*, 13(1):243–258, 1992. DOI: [10.1137/0913013](https://doi.org/10.1137/0913013).
- [51] J. G. Calvo and O. B. Widlund. An adaptive choice of primal constraints for BDDC domain decomposition algorithms. *Electron. Trans. Numer. Anal.*, 45:524–544, 2016.
- [52] L. Cambier, C. Chen, E. G. Boman, S. Rajamanickam, R. S. Tuminaro, and E. Darve. An algebraic sparsified nested dissection algorithm using low-rank approximations. *SIAM J. Matrix Anal. Appl.*, 41(2):715–746, 2020. DOI: [10.1137/19M123806X](https://doi.org/10.1137/19M123806X).
- [53] T. F. Chan, E. Chow, Y. Saad, and M. C. Yeung. Preserving symmetry in preconditioned Krylov subspace methods. *SIAM J. Sci. Comput.*, 20:568–581, 1999. DOI: [10.1137/S1064827596311554](https://doi.org/10.1137/S1064827596311554).
- [54] T. F. Chan and H. A. Van Der Vorst. Approximate and incomplete factorizations. In *Parallel numerical algorithms*, pages 167–202. Springer, 1997.
- [55] A. Chapman and Y. Saad. Deflated and augmented Krylov subspace techniques. *Numer. Linear Algebra Appl.*, 4(1):43–66, 1997. DOI: [10.1002/\(SICI\)1099-1506\(199701/02\)4:1<43::AID-NLA99>3.3.CO;2-Q](https://doi.org/10.1002/(SICI)1099-1506(199701/02)4:1<43::AID-NLA99>3.3.CO;2-Q).
- [56] T. Chartier, R. D. Falgout, V. E. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, and P. S. Vassilevski. Spectral AMGe (ρ AMGe). *SIAM J. Sci. Comput.*, 25(1):1–26, 2003. DOI: [10.1137/S106482750139892X](https://doi.org/10.1137/S106482750139892X).
- [57] P. Concus and G. H. Golub. A generalized conjugate gradient method for nonsymmetric systems of linear equations. In R. Glowinski and J.-L. Lions, editors, *Computing methods in applied sciences and engineering (Second Internat. Sympos., Versailles, 1975), Part 1*. Volume 134, Lecture Notes in Econom. and Math. Systems, pages 56–65. Springer, Berlin, 1976.
- [58] L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.
- [59] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258 (electronic), 2003. DOI: [10.1137/S1064827502412887](https://doi.org/10.1137/S1064827502412887).
- [60] Z. Dostál. Conjugate gradient method with preconditioning by projector. *International Journal of Computer Mathematics*, 23(3-4):315–323, 1988.
- [61] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices*. Numer. Math. Sci. Comput. Oxford: Oxford University Press, 2nd edition edition, 2017. DOI: [10.1093/acprof:oso/9780198508380.001.0001](https://doi.org/10.1093/acprof:oso/9780198508380.001.0001).
- [62] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Trans. Math. Softw.*, 9:302–325, 1983. DOI: [10.1145/356044.356047](https://doi.org/10.1145/356044.356047).
- [63] J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring. *GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations*. 2020.
- [64] Y. Efendiev, J. Galvis, R. Lazarov, and J. Willems. Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. *ESAIM Math. Model. Numer. Anal.*, 46(5):1175–1199, 2012. DOI: [10.1051/m2an/2011073](https://doi.org/10.1051/m2an/2011073).
- [65] E. Efstathiou and M. J. Gander. Why restricted additive Schwarz converges faster than additive Schwarz. *BIT*, 43(suppl.):945–959, 2003. DOI: [10.1023/B:BITN.0000014563.33622.1d](https://doi.org/10.1023/B:BITN.0000014563.33622.1d).
- [66] M. Eiermann and O. G. Ernst. Geometric aspects of the theory of Krylov subspace methods. *Acta Numerica*, 10:251–312, 2001. DOI: [10.1017/S0962492901000046](https://doi.org/10.1017/S0962492901000046).
- [67] S. C. Eisenstat, H. C. Elman, and M. H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, 20:345–357, 1983. DOI: [10.1137/0720023](https://doi.org/10.1137/0720023).

- [68] H. C. Elman. *Iterative methods for large, sparse, nonsymmetric systems of linear equations*. PhD thesis, Computer Science Dept., Yale University, 1982. Research Report 229.
- [69] M. Embree, R. B. Morgan, and H. V. Nguyen. Weighted inner products for GMRES and GMRES-DR. *SIAM J. Sci. Comput.*, 39:S610–S632, 2017.
- [70] M. Embree. Extending Elman’s Bound for GMRES. Preprint, arXiv:2312.15022 [math.NA] (2023), 2023.
- [71] Y. A. Erlangga and R. Nabben. Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices. *SIAM J. Matrix Anal. Appl.*, 30:684–699, 2008.
- [72] A. Essai. Weighted FOM and GMRES for solving nonsymmetric linear systems. *Numer. Algorithms*, 18:277–292, 1998. doi: [10.1023/A:1019177600806](https://doi.org/10.1023/A:1019177600806).
- [73] C. Farhat and M. G  radin. On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures. *International Journal for Numerical Methods in Engineering*, 41(4):675–696, 1998. doi: [10.1002/\(SICI\)1097-0207\(19980228\)41:4<675::AID-NME305>3.0.CO;2-8](https://doi.org/10.1002/(SICI)1097-0207(19980228)41:4<675::AID-NME305>3.0.CO;2-8).
- [74] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Methods Eng.*, 32(6):1205–1227, 1991. doi: [10.1002/nme.1620320604](https://doi.org/10.1002/nme.1620320604).
- [75] C. Farhat and F.-X. Roux. The dual Schur complement method with well-posed local Neumann problems. *Contemporary Mathematics*, 157:193, 1994.
- [76] J. Frank and C. Vuik. On the construction of deflation-based preconditioners. *SIAM J. Sci. Comput.*, 23(2):442–462, 2001. doi: [10.1137/S1064827500373231](https://doi.org/10.1137/S1064827500373231).
- [77] J. Galvis and Y. Efendiev. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Model. Simul.*, 8(5):1621–1644, 2010. doi: [10.1137/100790112](https://doi.org/10.1137/100790112).
- [78] J. Galvis and Y. Efendiev. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Model. Simul.*, 8(4):1461–1483, 2010. doi: [10.1137/090751190](https://doi.org/10.1137/090751190).
- [79] M. J. Gander and A. Loneland. Shem: an optimal coarse space for ras and its multiscale approximation. In *Domain decomposition methods in science and engineering XXIII*, pages 313–321. Springer, 2017.
- [80] L. Garc  a Ramos, R. Kehl, and R. Nabben. Projections, deflation, and multigrid for nonsymmetric matrices. *SIAM J. Matrix Anal. Appl.*, 41:83–105, 2020. doi: [10.1137/18M1180268](https://doi.org/10.1137/18M1180268).
- [81] A. Gaul, M. H. Gutknecht, J. Liesen, and R. Nabben. A framework for deflated and augmented Krylov subspace methods. *SIAM J. Matrix Anal. Appl.*, 34(2):495–518, 2013.
- [82] G. H. Golub and C. F. van Loan. *Matrix Computations*. JHU Press, fourth edition, 2013.
- [83] P. Gosselet, C. Rey, and J. Pebrel. Total and selective reuse of Krylov subspaces for the resolution of sequences of nonlinear structural problems. *Internat. J. Numer. Methods Engrg.*, 94(1):60–83, 2013. doi: [10.1002/nme.4441](https://doi.org/10.1002/nme.4441).
- [84] I. G. Graham, E. A. Spence, and E. Vainikko. Domain decomposition preconditioning for high-frequency Helmholtz problems with absorption. *Math. Comput.*, 86(307):2089–2127, 2017. doi: [10.1090/mcom/3190](https://doi.org/10.1090/mcom/3190).
- [85] I. G. Graham, E. A. Spence, and J. Zou. Domain decomposition with local impedance conditions for the Helmholtz equation with absorption. *SIAM J. Numer. Anal.*, 58(5):2515–2543, 2020. doi: [10.1137/19M1272512](https://doi.org/10.1137/19M1272512).

- [86] A. Greenbaum, V. Pták, and Z. Strakoš. Any nonincreasing convergence curve is possible for GMRES. *SIAM J. Matrix Anal. Appl.*, 17(3):465–469, 1996. DOI: [10.1137/S0895479894275030](https://doi.org/10.1137/S0895479894275030).
- [87] C. Greif, T. Rees, and D. B. Szyld. MPGMRES: a generalized minimum residual method with multiple preconditioners. Technical report 11-12-23, Temple University, 2014.
- [88] C. Greif, T. Rees, and D. Szyld. Additive Schwarz with variable weights. In *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014.
- [89] S. Güttel and J. Pestana. Some observations on weighted GMRES. *Numer. Algorithms*, 67(4):733–752, 2014. DOI: [10.1007/s11075-013-9820-x](https://doi.org/10.1007/s11075-013-9820-x).
- [90] R. Haferssas, P. Jolivet, and F. Nataf. An additive Schwarz method type theory for Lions’s algorithm and a symmetrized optimized restricted additive Schwarz method. *SIAM Journal on Scientific Computing*, 39(4):A1345–A1365, 2017.
- [91] F. Hecht. *FreeFem++*. Numerical Mathematics and Scientific Computation. Laboratoire J.L. Lions, Université Pierre et Marie Curie, <http://www.freefem.org/ff++/>, 3.23 edition, 2013.
- [92] F. Hecht. New development in FreeFem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [93] A. Heinlein, A. Klawonn, J. Knepper, and O. Rheinbach. Adaptive GDSW coarse spaces for overlapping Schwarz methods in three dimensions. *SIAM J. Sci. Comput.*, 41(5):a3045–a3072, 2019. DOI: [10.1137/18M1220613](https://doi.org/10.1137/18M1220613).
- [94] A. Heinlein, A. Klawonn, J. Knepper, O. Rheinbach, and O. B. Widlund. Adaptive GDSW coarse spaces of reduced dimension for overlapping Schwarz methods. *SIAM J. Sci. Comput.*, 44(3):a1176–a1204, 2022. DOI: [10.1137/20M1364540](https://doi.org/10.1137/20M1364540).
- [95] A. Heinlein, A. Klawonn, M. Lanser, and J. Weber. Combining machine learning and domain decomposition methods for the solution of partial differential equations – a review. *GAMM-Mitt.*, 44(1):28, 2021. DOI: [10.1002/gamm.202100001](https://doi.org/10.1002/gamm.202100001). Id/No 202100001.
- [96] A. Heinlein, A. Klawonn, M. Lanser, and J. Weber. Machine learning in adaptive domain decomposition methods – predicting the geometric location of constraints. *SIAM J. Sci. Comput.*, 41(6):a3887–a3912, 2019. DOI: [10.1137/18M1205364](https://doi.org/10.1137/18M1205364).
- [97] G. Hello, J. Schneider, and Z. Aboura. Generation of Voxel-FE Models for Complex 3D Composite Architectures. In *8th European Solid Mechanics Conference (ESMC-2012)*, Graz, Austria, 2012.
- [98] P. Hénon, P. Ramet, and J. Roman. PaStiX: A high-performance parallel direct solver for sparse symmetric positive definite systems. *Parallel Comput.*, 28(2):301–321, 2002. DOI: [10.1016/S0167-8191\(01\)00141-7](https://doi.org/10.1016/S0167-8191(01)00141-7).
- [99] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software*, 31(3):351–362, 2005.
- [100] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49:409–436, 1952. DOI: [10.6028/jres.049.044](https://doi.org/10.6028/jres.049.044).
- [101] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, second edition, 2013.
- [102] C. R. Johnson. An inequality for matrices whose symmetric part is positive definite. *Linear Algebra Appl.*, 6:13–18, 1973. DOI: [10.1016/0024-3795\(73\)90003-7](https://doi.org/10.1016/0024-3795(73)90003-7).
- [103] C. R. Johnson. Inequalities for a complex matrix whose real part is positive definite. *Trans. Am. Math. Soc.*, 212:149–154, 1975. DOI: [10.2307/1998618](https://doi.org/10.2307/1998618).
- [104] C. R. Johnson. *Matrices whose hermitian part is positive definite*. PhD thesis, California Institute of Technology, 1972.

- [105] W. D. Joubert and T. A. Manteuffel. Iterative methods for nonsymmetric linear systems. *Academic Press, New York*:149–171, 1990.
- [106] E. F. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *J. Comput. Appl. Math.*, 24(1-2):265–275, 1988. DOI: [10.1016/0377-0427\(88\)90358-5](https://doi.org/10.1016/0377-0427(88)90358-5).
- [107] S. Kaniel. Estimates for some computational techniques in linear algebra. *Mathematics of Computation*, 20(95):369–378, 1966.
- [108] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392 (electronic), 1998. DOI: [10.1137/S1064827595287997](https://doi.org/10.1137/S1064827595287997).
- [109] A. Klawonn, P. Radtke, and O. Rheinbach. Adaptive coarse spaces for BDDC with a transformation of basis. In *Twenty Second International Conference on Domain Decomposition Methods*, 2014.
- [110] A. Klawonn, P. Radtke, and O. Rheinbach. FETI-DP methods with an adaptive coarse space. *SIAM J. Numer. Anal.*, 53(1):297–320, 2015. DOI: [10.1137/130939675](https://doi.org/10.1137/130939675).
- [111] A. Klawonn, M. Kuhn, and O. Rheinbach. Adaptive coarse spaces for FETI-DP in three dimensions. *SIAM Journal on Scientific Computing*, 38(5):A2880–A2911, 2016.
- [112] A. Klawonn, M. Lanser, and O. Rheinbach. Toward extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations. *SIAM Journal on Scientific Computing*, 37(6):C667–C696, 2015.
- [113] A. Klawonn, M. Lanser, and J. Weber. Learning adaptive coarse basis functions of FETI-DP. *J. Comput. Phys.*, 496:17, 2024. DOI: [10.1016/j.jcp.2023.112587](https://doi.org/10.1016/j.jcp.2023.112587). Id/No 112587.
- [114] A. Klawonn, M. Lanser, and J. Weber. Machine learning and domain decomposition methods – a survey. Preprint, arXiv:2312.14050 [math.NA] (2023), 2023.
- [115] A. Klawonn, P. Radtke, and O. Rheinbach. A comparison of adaptive coarse spaces for iterative substructuring in two dimensions. *Electron. Trans. Numer. Anal.*, 45:75–106, 2016.
- [116] A. Klawonn and O. Rheinbach. Deflation, projector preconditioning, and balancing in iterative substructuring methods: connections and new results. *SIAM J. Sci. Comput.*, 34(1):A459–A484, 2012. DOI: [10.1137/100811118](https://doi.org/10.1137/100811118).
- [117] A. Klawonn and G. Starke. Block triangular preconditioners for nonsymmetric saddle point problems: Field-of-values analysis. *Numer. Math.*, 81(4):577–594, 1999. DOI: [10.1007/s002110050405](https://doi.org/10.1007/s002110050405).
- [118] A. Klawonn and O. B. Widlund. FETI and Neumann-Neumann iterative substructuring methods: connections and new results. *Comm. Pure Appl. Math.*, 54(1):57–90, 2001. DOI: [10.1002/1097-0312\(200101\)54:1<57::AID-CPA3>3.0.CO;2-D](https://doi.org/10.1002/1097-0312(200101)54:1<57::AID-CPA3>3.0.CO;2-D).
- [119] R. Li and Y. Saad. Low-rank correction methods for algebraic domain decomposition preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 38(3):807–828, 2017.
- [120] X. S. Li. An overview of SuperLU: algorithms, implementation, and user interface. 31(3):302–325, Sept. 2005.
- [121] X. S. Li and J. W. Demmel. SuperLU DIST: a scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Mathematical Software*, 29(2):110–140, June 2003.
- [122] J. Liesen and Z. Strakoš. *Krylov subspace methods. Principles and analysis*. Numer. Math. Sci. Comput. Oxford: Oxford University Press, reprint of the 2013 hardback edition edition, 2015. DOI: [10.1093/acprof:oso/9780199655410.001.0001](https://doi.org/10.1093/acprof:oso/9780199655410.001.0001).
- [123] J. Liesen and P. Tichý. The field of values bound on ideal GMRES. *arXiv preprint arXiv:1211.5969*, 2012. DOI: [10.48550/ARXIV.1211.5969](https://doi.org/10.48550/ARXIV.1211.5969).

- [124] J. W. H. Liu. The multifrontal method for sparse matrix solution: Theory and practice. *SIAM Rev.*, 34(1):82–109, 1992. doi: [10.1137/1034004](https://doi.org/10.1137/1034004).
- [125] J. Málek and Z. Strakoš. *Preconditioning and the conjugate gradient method in the context of solving PDEs*. SIAM, Philadelphia, 2015.
- [126] J. Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9(3):233–241, 1993. doi: [10.1002/cnm.1640090307](https://doi.org/10.1002/cnm.1640090307).
- [127] J. Mandel and M. Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65(216):1387–1401, 1996. doi: [10.1090/S0025-5718-96-00757-0](https://doi.org/10.1090/S0025-5718-96-00757-0).
- [128] J. Mandel and B. Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1389–1399, 2007. doi: [10.1016/j.cma.2006.03.010](https://doi.org/10.1016/j.cma.2006.03.010).
- [129] P. Marchand, X. Claeys, P. Jolivet, F. Nataf, and P.-H. Tournier. Two-level preconditioning for h h -version boundary element approximation of hypersingular operator with GenEO. *Numerische Mathematik*, 146(3):597–628, 2020.
- [130] G. Meinardus. *Approximation of functions: Theory and numerical methods*. Expanded translation of the German edition. Translated by Larry L. Schumaker. Springer Tracts in Natural Philosophy, Vol. 13. Springer-Verlag New York, Inc., New York, 1967, pages viii+198.
- [131] G. Meurant and J. D. Tebbens. The role eigenvalues play in forming GMRES residual norms with non-normal matrices. *Numer. Algorithms*, 68(1):143–165, 2015. doi: [10.1007/s11075-014-9891-3](https://doi.org/10.1007/s11075-014-9891-3).
- [132] R. B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.*, 16(4):1154–1171, 1995. doi: [10.1137/S0895479893253975](https://doi.org/10.1137/S0895479893253975).
- [133] R. Nabben and J. Schramm. Improved error bounds for the deflated multi-preconditioned CG method. *Linear Algebra Appl.*, 712:29–48, 2025. doi: [10.1016/j.laa.2025.02.022](https://doi.org/10.1016/j.laa.2025.02.022).
- [134] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen. How fast are nonsymmetric matrix iterations? *SIAM J. Matrix Anal. Appl.*, 13(3):778–795, 1992. doi: [10.1137/0613049](https://doi.org/10.1137/0613049).
- [135] F. Nataf, H. Xiang, and V. Dolean. A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps. *C. R. Math. Acad. Sci. Paris*, 348(21-22):1163–1167, 2010. doi: [10.1016/j.crma.2010.10.007](https://doi.org/10.1016/j.crma.2010.10.007).
- [136] R. A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24(2):355–365, 1987. doi: [10.1137/0724027](https://doi.org/10.1137/0724027).
- [137] A. A. Nikishin and A. Y. Yeremin. Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers. I. General iterative scheme. *SIAM J. Matrix Anal. Appl.*, 16(4):1135–1153, 1995. doi: [10.1137/S0895479893247679](https://doi.org/10.1137/S0895479893247679).
- [138] D. P. O’Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra Appl.*, 29:293–322, 1980. doi: [10.1016/0024-3795\(80\)90247-5](https://doi.org/10.1016/0024-3795(80)90247-5).
- [139] M. L. Parks, E. De Sturler, G. Mackey, D. D. Johnson, and S. Maiti. Recycling Krylov subspaces for sequences of linear systems. *SIAM J. Sci. Comput.*, 28(5):1651–1674, 2006. doi: [10.1137/040607277](https://doi.org/10.1137/040607277).
- [140] C. Pechstein and C. R. Dohrmann. A unified framework for adaptive BDDC. *Electron. Trans. Numer. Anal.*, 46(273-336):3, 2017.
- [141] C. Pechstein and R. Scheichl. Analysis of FETI methods for multiscale PDEs. Part II: interface variation. *Numer. Math.*, 118(3):485–529, 2011. doi: [10.1007/s00211-011-0359-2](https://doi.org/10.1007/s00211-011-0359-2).

- [142] J. Pestana. *Nonstandard inner products and preconditioned iterative methods*. PhD thesis, Oxford University, UK, 2011.
- [143] J. Pestana and A. J. Wathen. On the choice of preconditioner for minimum residual methods for non-Hermitian matrices. *J. Comput. Appl. Math.*, 249:57–68, 2013. DOI: 10.1016/j.cam.2013.02.020.
- [144] D. Rixen. *Substructuring and Dual Methods in Structural Analysis*. PhD thesis, Université de Liège, Collection des Publications de la Faculté des Sciences appliquées, n.175, 1997.
- [145] D. J. Rixen and C. Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *Int. J. Numer. Meth. Engng.*, 44(4):489–516, 1999. DOI: 10.1002/(SICI)1097-0207(19990210)44:4<489::AID-NME514>3.0.CO;2-Z.
- [146] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid methods*, pages 73–130. SIAM, 1987.
- [147] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc’h. A deflated version of the conjugate gradient algorithm. *SIAM J. Sci. Comput.*, 21(5):1909–1926, 2000. DOI: 10.1137/S1064829598339761.
- [148] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986. DOI: 10.1137/0907058.
- [149] Y. Saad. *Iterative methods for sparse linear systems*. Philadelphia, PA: SIAM Society for Industrial and Applied Mathematics, 2nd ed. Edition, 2003.
- [150] M. Sarkis and D. B. Szyld. Optimal left and right additive Schwarz preconditioning for minimal residual methods with Euclidean and energy norms. *Comput. Methods Appl. Mech. Eng.*, 196(8):1612–1621, 2007. DOI: 10.1016/j.cma.2006.03.027.
- [151] J. Schneider, G. Hello, Z. Aboura, M. Benzeggagh, and D. Marsal. A Meso-FE voxel model of an interlock woven composite. In *17th International Conference on Composite Materials (ICCM-17)*, Edinburgh, Scotland, 2009.
- [152] V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numer. Linear Algebra Appl.*, 14:1–59, 2007.
- [153] K. M. Soodhalter, E. de Sturler, and M. E. Kilmer. A survey of subspace recycling iterative methods. *GAMM-Mitt.*, 43(4):29, 2020. DOI: 10.1002/gamm.202000016. Id/No e202000016.
- [154] B. Sousedík, J. Šístek, and J. Mandel. Adaptive-Multilevel BDDC and its parallel implementation. *Computing*, 95(12):1087–1119, 2013. DOI: 10.1007/s00607-013-0293-5.
- [155] G. Starke. Field-of-values analysis of preconditioned iterative methods for nonsymmetric elliptic problems. *Numer. Math.*, 78:103–117, 1997. DOI: 10.1007/s002110050306.
- [156] Z. Strakoš and P. Tichý. Error estimation in preconditioned conjugate gradients. *BIT*, 45(4):789–817, 2005. DOI: 10.1007/s10543-005-0032-1.
- [157] D. B. Szyld and O. B. Widlund. Variational analysis of some conjugate gradient methods. *East-West J. Numer. Math.*, 1:51–74, 1993.
- [158] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *J. Sci. Comput.*, 39(3):340–370, 2009. DOI: 10.1007/s10915-009-9272-6.
- [159] A. Toselli and O. Widlund. *Domain decomposition methods – algorithms and theory*. Volume 34 of *Springer Ser. Comput. Math.* Berlin: Springer, 2005.
- [160] P.-H. Tournier, P. Jolivet, and F. Nataf. FFDDM: freefem domain decomposition method. <https://doc.freefem.org/documentation/ffddm/index.html>, 2019.

- [161] A. van der Sluis and H. A. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986. DOI: [10.1007/BF01389450](https://doi.org/10.1007/BF01389450).
- [162] P. S. Vassilevski. *Multilevel block factorization preconditioners*. Springer, New York, 2008, pages xiv+529. Matrix-based analysis and algorithms for solving finite element equations.
- [163] A. J. Wathen. Preconditioning. *Acta Numerica*, 24:329–376, 2015. DOI: [10.1017/S0962492915000021](https://doi.org/10.1017/S0962492915000021).
- [164] O. Widlund. A Lanczos method for a class of nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, 15:801–812, 1978. DOI: [10.1137/0715053](https://doi.org/10.1137/0715053).
- [165] Y. Wielhorski and D. Durville. Finite element simulation of a 3d woven fabric: determination of the initial configuration and characterization of the mechanical behavior. In *13th International conference on textile composites (TexComp-12)*, Rayleigh, USA, 2015.
- [166] M. A. Woodbury. *Inverting modified matrices*. Statistical Research Group, 1950.
- [167] J. Xu and X.-C. Cai. A preconditioned GMRES method for nonsymmetric or indefinite problems. *Math. Comput.*, 59(200):311–319, 1992. DOI: [10.2307/2153059](https://doi.org/10.2307/2153059).
- [168] J. Xu and L. Zikatanov. Algebraic multigrid methods. *Acta Numerica*, 26:591–721, 2017.
- [169] *Z-set 8.6 user manual*. Transvalor S.A. 2015.
- [170] S. Zampini. PCBDDC: a class of robust dual-primal methods in PETSc. *SIAM Journal on Scientific Computing*, 38(5):S282–S306, 2016.