Sharpened CG iteration bound for Schwarz-preconditioned high-contrast heterogeneous elliptic PDEs

Going beyond condition number

WI5005: Thesis Project (Interim Thesis)

Philip Soliman

Delft University of Technology

TU Delft

Sharpened CG iteration bound for Schwarz-preconditioned high-contrast heterogeneous elliptic PDEs

Going beyond condition number

by

Philip Soliman

To obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on T.B.A.

Student number:	4945255	
Project duration:	December 2024 – September 2025	
Thesis committee:	Prof. H. Schuttelaars,	TU Delft, responsible supervisor
	Dr. A. Heinlein,	TU Delft, daily supervisor
	F. Camaru,	TU Delft, daily co-supervisor

This thesis is confidential and cannot be made public until December 31, 2025.

TU Delft

Contents

Nomenclature

Table 1: List of symbols and their descriptions from the background section.

Symbol	Description
Ω	Bounded domain in $\mathbb{R}^d$ with Lipschitz boundary.
$\partial\Omega$	Boundary of Ω.
$\mathbb{R}^d$	d-dimensional Euclidean space.
$\mathcal{C}$	Scalar coefficient in the Darcy problem equation in $L^\infty(\Omega)$
u	Exact solution of the elliptic problem.
f	Source term in $L^2(\Omega)$.
u_D	Dirichlet boundary data.
$\kappa_{\min}$	Lower bound of κ.
$\kappa_{\max}$	Upper bound of κ.
u_h	Finite element approximation of u.
V_h	Finite-dimensional subspace of $H_0^1(\Omega)$.
$\{\varphi_i\}_{i=1}^n$	Basis functions spanning V_h.
A	Stiffness matrix from the Galerkin method.
$\mathbf{u}$	Discrete solution vector.
$\mathbf{b}$	Load vector in the linear system.
v_h	Test function in V_h.
$\mathbf{u}_0$	Initial guess for the solution.
$\mathbf{r}_0$	Initial residual, defined as $\mathbf{b} - A\mathbf{u}_0$.
$\mathbf{r}_j$	Residual vector at iteration j.
c_i	Coefficients in the solution polynomial.
$q_{m-1}(A)$	Polynomial of degree $m-1$ used in the CG approximation.
$\{\mathbf{v}_i\}$	Orthonormal Lanczos vectors.
$\mathcal{K}_m(A, \mathbf{r}_0)$	Krylov subspace spanned by $\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{m-1}\mathbf{r}_0\}$.
$\mathcal{L}_m$	Constraint subspace in variants of CG.
A^T	Transpose of A.
μ	Grade of a vector with respect to A.
α_j	CG step size, computed as $(\mathbf{r}_j, \mathbf{r}_j)/(Ap_j, p_j)$.
p_j	CG search direction at iteration j.
β_j	CG coefficient, computed as $(\mathbf{r}_{j+1}, \mathbf{r}_{j+1})/(\mathbf{r}_j, \mathbf{r}_j)$.
T_m	Tridiagonal Hessenberg matrix from the Lanczos process.
δ_j	Diagonal entries of T_m.
η_j	Off-diagonal entries of T_m.
ϵ_m	Error at iteration m, defined as $x^* - \mathbf{u}_m$.
$\mathcal{P}_{m-1}$	Space of polynomials of degree at most $m-1$.
λ_i	Eigenvalues of A.
ξ_i	Components of ϵ_0 in the eigenvector basis of A.
$\sigma(A)$	Spectrum of A.
C_m	Chebyshev polynomial of degree m.
η	$\dfrac{\lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}$ (in error bound).
$\lambda_{\min}$	Minimum eigenvalue of A.
$\lambda_{\max}$	Maximum eigenvalue of A.
κ	Condition number of A, given by $\lambda_{\max}/\lambda_{\min}$.
x^*	Exact solution of the linear system.

Symbol	Description
$\mathbf{r}_{\text{test}}(t)$	Test polynomial, $\prod_{i=1}^{m} \frac{\lambda_i - t}{\lambda_i}$.
N	Number of iterations for convergence in exact arithmetic.
k	Number of distinct eigenvalues of A.
V	Matrix of eigenvectors from the eigendecomposition $A = VDV^T$.
D	Diagonal matrix of eigenvalues from the eigendecomposition of A.
$r(A)$	Residual polynomial defined as $I - Aq(A)$.
$r_m(A)$	Residual polynomial at iteration m.
I	Identity matrix.
ρ_0	Initial residual in the eigenvector basis, $V^T\mathbf{r}_0$.
$p_j(A)$	Lanczos polynomial at iteration j.
$r_j(A)$	Residual polynomial at iteration j.
$p_j(0)$	Value of the Lanczos polynomial at 0.
coeff$(p; i)$	Function that extracts the i^{th} coefficient of polynomial p.
z_j	Preconditioned residual, defined as $M^{-1}\mathbf{r}_j$.
M	Symmetric positive definite (SPD) preconditioner.
L	Lower triangular matrix from the Cholesky factorization of M ($M = LL^T$).
L^T	Transpose of L.
$(\cdot,\cdot)_M$	Inner product induced by M.
$\mathcal{T}_h$	Fine triangulation.
$\mathcal{T}_H$	Coarse triangulation.
u_c	Coarse solution, split into two components: u_I (inner) and u_Γ (interface).
u_I	Inner part of the coarse solution.
u_Γ	Interface part of the coarse solution.
Γ	Interface set, where boundary conditions are applied.
e	Edge in the triangulation.
T	Coarse element.
e_{ij}	Shared edge between subdomains Ω_i and Ω_j.
Ω_i, Ω_j	Subdomains in the domain decomposition.
η_{ij}^{kh}	Eigenmode corresponding to the generalized eigenvalue problem on a slab of width kh between edges e_{ij}.
R_0	Restriction operator derived from coarse grid basis functions.
$\pi(\alpha)$	Robustness indicator related to the scalar coefficient α.
$\gamma(\alpha)$	Another robustness indicator related to the scalar coefficient α.
G	Coarse space matrix whose columns span the rigid body modes of the subdomains.
R_Γ	Restriction operator to the interface degrees of freedom (DOFs).
R_I	Restriction operator to the interior DOFs.
R_{Γ_j}	Subdomain-specific version of the restriction operator to the interface DOFs.
Φ_Γ	Coarse solution on the interface set.
q_j	Coarse space coefficients corresponding to subdomains.
$u_{0,\Gamma}$	Coarse solution on the interface set, expressed in terms of coarse space coefficients.
Φ_I	Energy-minimized solution on the interior.

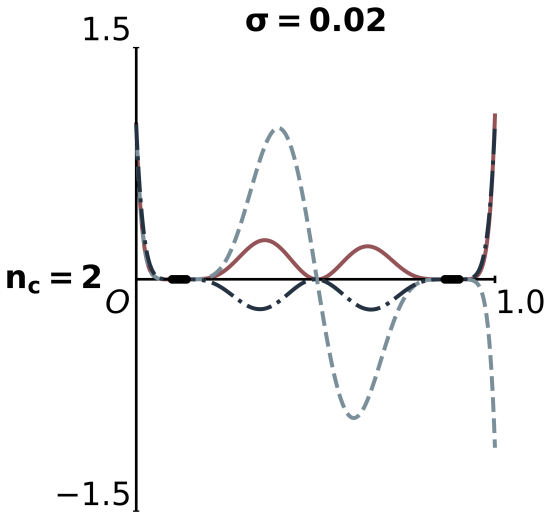
1

Introduction

To do ▶*Purpose of the review – What problem are you addressing?*◀
To do ▶*Main research question – Clearly stated in one sentence.*◀
To do ▶*Subsidiary questions – List related questions that refine the focus.*◀
To do ▶*Scope and relevance – Why is this topic important?*◀
To do ▶*Brief structure overview – What will each section cover?*◀

2

Mathematical background



In this chapter we focus on a simple Darcy problem like the one posed in [1, 9]. This problem is of the form

$$\begin{aligned} -\nabla \cdot (\mathcal{C}\nabla u) &= f \quad \text{in } \Omega, \\ u &= u_D \quad \text{on } \partial\Omega, \end{aligned} \tag{2.1}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded domain with Lipschitz boundary $\partial\Omega$, $\mathcal{C} \in L^\infty(\Omega)$ is a positive coefficient, $f \in L^2(\Omega)$ is a source term, and $u \in H_0^1(\Omega)$ is the solution. The coefficient $\mathcal{C}$ is assumed to be bounded from above and below by positive constants, i.e., $0 < \mathcal{C}_{\min} \leq \mathcal{C}(x) \leq \mathcal{C}_{\max} < \infty$ for all $x \in \Omega$. The solution u is assumed to be sufficiently smooth, i.e., $u \in H^2(\Omega) \cap H_0^1(\Omega)$, so that the problem is well-posed. The goal is to compute an approximation $u_h \in V_h$ to the solution u using a finite-dimensional subspace $V_h \subset H_0^1(\Omega)$, where V_h is spanned by a set of basis functions $\{\varphi_i\}_{i=1}^n$. The Galerkin method seeks $u_h \in V_h$ such that

$$\int_\Omega \mathcal{C}\nabla u_h \cdot \nabla v_h \, dx = \int_\Omega f v_h \, dx \quad \text{for all } v_h \in V_h. \tag{2.2}$$

The Galerkin method leads to a linear system of equations $A\mathbf{u} = \mathbf{b}$, where A is the stiffness matrix and $\mathbf{b}$ is the load vector. The stiffness matrix A is symmetric and positive definite, and the load vector $\mathbf{b}$ is determined by the source term f and the boundary conditions. The solution $\mathbf{u}$ can be computed using iterative methods like the conjugate gradient method, which is guaranteed to converge in a finite number of iterations for symmetric positive definite matrices in infinite precision arithmetic.

2.1. Conjugate Gradient Method

The CG method is a special instance of the class of Krylov subspace methods. It is derived from the Direct Lanczos (D-Lanczos) algorithm applied to the residual of linear systems [10, Algorithm 6.17]. The D-Lanczos algorithm generates a sequence of orthonormal Lanczos vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ that span the Krylov subspace $\mathcal{K}_m(A, \mathbf{r}_0)$, where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{u}_0$ is the initial residual such that

$$\mathcal{K}_m(A_0, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{m-1}\mathbf{r}_0\}, \tag{2.3}$$

or $\mathcal{K}_m$ as a shorthand. The approximate answer is then given by

$$\mathbf{u}_m = \mathbf{u}_0 + \sum_{i=0}^{m-1} c_i A^i \mathbf{r}_0 = \mathbf{u}_0 + q_{m-1}(A)\mathbf{r}_0, \tag{2.4}$$

where $q_{m-1}(A)$ is a polynomial of degree $m-1$ in A. It is shown later in this section how the coefficients c_i are obtained (equation (2.15)).

2.1.1. Variants of the CG method

Variants of the CG method differ in the way A is preconditioned (see section section 2.1.6) d the choices for the constraint subspace $\mathcal{L}_m$. The former type of variations result in the preconditioned CG method PCG and these are described in section 2.1.6. The latter type of variations branch off into two major categories:

i $\mathcal{L}_m = \mathcal{K}_m$ and $\mathcal{L}_m = A\mathcal{K}_m$;

ii $\mathcal{L}_m = \mathcal{K}_m(A^T, \mathbf{r}_0)$.

Note that item CG-type i correspond to the residual and error projection methods. The former results in Arnoldi's method, as well as variants thereof like Full Orthogonalization Method (FOM), Incomplete Orthogonalization Method (IOM) and Direct Incomplete Orthogonalization Method (DIOM). The latter on the other hand results in the Generalized Minimum Residual Method (GMRES).

2.1.2. Krylov subspaces

Definition 2.1. The grade of a vector v with respect to a matrix A is the lowest degree of the polynomial q such that $q(A)v = 0$.

Consequently,

Theorem 2.1. The Krylov subspace $\mathcal{K}_m$ is of dimension m if and only if the grade μ of v with respect to A is not less than m [10, proposition 6.2],

$$\dim(\mathcal{K}_m) = m \iff \mu \geq m,$$

such that

$$\dim(\mathcal{K}_m) = \min\{m, \text{grade}(v)\}. \tag{2.5}$$

2.1.3. CG algorithm

We can write the conjugate gradient method as algorithm 1.

Algorithm 1 Conjugate Gradient Method [10, Algorithm 6.18]

$\mathbf{r}_0 = b - A\mathbf{u}_0$, $p_0 = \mathbf{r}_0$, $\beta_0 = 0$
for $j = 0, 1, 2, \ldots, m$ **do**
  $\alpha_j = (\mathbf{r}_j, \mathbf{r}_j)/(Ap_j, p_j)$
  $\mathbf{u}_{j+1} = \mathbf{u}_j + \alpha_j p_j$
  $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j A p_j$
  $\beta_j = (\mathbf{r}_{j+1}, \mathbf{r}_{j+1})/(\mathbf{r}_j, \mathbf{r}_j)$
  $p_{j+1} = \mathbf{r}_{j+1} + \beta_j p_j$
end for

The Lanczos vectors are related through the Lanczos recurrence relation

$$\eta_{j+1}(A)\mathbf{v}_{j+1} = A\mathbf{v}_j - \delta_j \mathbf{v}_j - \eta_j \mathbf{v}_{j-1}, \tag{2.6}$$

such that

$$T_m = \mathbf{v}_m^T A \mathbf{v}_m,$$

where T_m is the tridiagonal Hessenberg matrix given by

$$T_m = \begin{pmatrix} \delta_1 & \eta_2 & 0 & \ldots & 0 \\ \eta_2 & \delta_3 & \eta_3 & \ldots & 0 \\ 0 & \eta_3 & \delta_4 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \eta_m \\ 0 & 0 & 0 & \eta_m & \delta_m \end{pmatrix}. \tag{2.7}$$

The following relations exist between the entries of T_m and the CG coefficients α_j, β_j

$$\delta_{j+1} = \begin{cases} \dfrac{1}{\alpha_j} + \dfrac{\beta_{j-1}}{\alpha_{j-1}} & j > 0, \\ \dfrac{1}{\alpha_0} & j = 0, \end{cases} \tag{2.8}$$

and

$$\eta_{j+1} = \frac{\sqrt{\beta_{j-1}}}{\alpha_{j-1}}. \tag{2.9}$$

Here we have used the definition of T_m and the fact that the residuals are multiples of the Lanczos vectors $\mathbf{r}_j = \text{scalar} \times \mathbf{v}_j$ [10, Equation 6.103].

2.1.4. CG convergence rate

It can be shown [10, lemma 6.28 and theorem 6.29] that the error of the m^{th} iterate of the CG algorithm $\epsilon_m = x^* - \mathbf{u}_m$ minimizes the A-norm of the error in the affine Krylov subspace $\mathcal{K}_m(A, \mathbf{r}_0)$, that is

$$||(I - Aq_m(A))\epsilon_0||_A = \min_{q \in \mathcal{P}_{m-1}} ||(I - Aq(A))\epsilon_0||_A = \min_{r \in \mathcal{P}_{m-1}, r(0)=1} ||r(A)\epsilon_0||_A, \tag{2.10}$$

where the equality follows, since there exists an isomorphic mapping between the affine Krylov subspace and the polynomial space $\mathcal{P}_{m-1}$ of degree $m-1$ and the polynomial $tq(t)$ equals 0 at $t=0$. The right-hand side can be further bounded by letting λ_i, ξ_i be the eigenvalues of A and the components of ϵ_0 in the eigenvector basis of A, respectively. Then

$$||r(A)\epsilon_0||_A = \sqrt{\sum_{i=1}^{n} |r(\lambda_i)|^2 |\xi_i|^2} \leq \max_{\lambda \in \sigma(A)} |r(\lambda)| ||\epsilon_0||_A,$$

where $\sigma(A)$ is the spectrum of A. This gives

$$\begin{aligned}
||e_m||_A &\leq \min_{r \in \mathcal{P}_{m-1}, r(0)=1} \max_{\lambda \in \sigma(A)} |r(\lambda)| ||\epsilon_0||_A \\
\text{Chebyshev polynomial } C_m,\ \eta = \frac{\lambda_{\text{min}}}{\lambda_{\text{max}} - \lambda_{\text{min}}} &\rightarrow \frac{||\epsilon_0||_A}{C_m(1+2\eta)} \\
&\leq \frac{2||\epsilon_0||_A}{\left(1 + 2\eta + 2\sqrt{\eta(\eta+1)}\right)^m} \\
&= \frac{2||\epsilon_0||_A}{\left(\sqrt{\eta} + \sqrt{\eta+1}\right)^{2m}} \\
&= 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m ||\epsilon_0||_A,
\end{aligned}$$

where $\sigma(A) = [\lambda_{\text{min}}, \lambda_{\text{max}}]$ and $\kappa = \lambda_{\text{max}}/\lambda_{\text{min}}$ is the condition number of (the symmetric matrix) A. To sum up

Theorem 2.2. The error of the m^{th} iterate of the CG algorithm is bounded by

$$||e_m|| \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m ||\epsilon_0||_A, \tag{2.11}$$

where $\kappa = \lambda_{\text{max}}/\lambda_{\text{min}}$ is the condition number of (symmetric matrix) A.

During the derivation of **??**, we obtain the general expression for the error of the m^{th} iterate of the CG algorithm

$$||e_m|| \leq \min_{r \in \mathcal{P}_m, r(0)=1} \max_{\lambda \in \sigma(A)} |r(\lambda)| ||\epsilon_0||_A$$

Now define,

$$\mathbf{r}_{\text{test}}(t) = \prod_{i=1}^{m} \frac{\lambda_i - t}{\lambda_i}.$$

Note that $\mathbf{r}_{\text{test}} \in \mathcal{P}_m$, since it has degree m. Also, $\mathbf{r}_{\text{test}}(0) = 1$ and $\mathbf{r}_{\text{test}}(\lambda_i) = 0$ for $i = 1, 2, \ldots, m$. Hence, $\mathbf{r}_{\text{test}}$ is a polynomial that satisfies the constraints of the minimization problem. We obtain for $m = N$ that

$$||e_N||_A = ||\epsilon_0||_A \max_{\lambda \in \sigma(A)} |\mathbf{r}_{\text{test}}(\lambda)| = 0,$$

which implies that CG converges in N iterations in exact arithmetic. Furthermore, if there are only k distinct eigenvalues, then the CG iteration terminates in at most k iterations.

2.1.5. Influence of eigenvalue distribution on CG convergence

In the derivation of the convergence rate of the CG algorithm in **??**, we used the Chebyshev polynomial to bound the error. However, we can find an expression of the error provided the eigendecomposition of A is available. Suppose $A = VDV^T$, then $r(A) = I - Aq(A) = V(I - Dq(D))V^T = Vr(D)V^T$. Also note that $e_0 = x^* - \mathbf{u}_0 = A^{-1}b - \mathbf{u}_0 = A^{-1}\mathbf{r}_0$. As seen in equation (2.10), the error of the m^{th} iterate of the CG algorithm is given by

$$||e_m||_A^2 = ||r_m(A)\epsilon_0||_A^2,$$

and

$$\begin{aligned}||r_m(A)\epsilon_0||_A^2 &= \epsilon_0^T r_m(A)^T A r_m(A)\epsilon_0 \\ &= \epsilon_0^T V r_m(D) V^T V D V^T V r_m(D) V^T \epsilon_0 \\ &= (V^T\epsilon_0)^T r_m(D) D r_m(D) V^T \epsilon_0.\end{aligned}$$

We also have

$$\begin{aligned}V^T\epsilon_0 &= V^T A^{-1}\mathbf{r}_0 \\ &= V^T V D^{-1} V^T \mathbf{r}_0 \\ &= D^{-1}\rho_0,\end{aligned}$$

where $\rho_0 = V^T\mathbf{r}_0$ is the initial residual in the eigenvector basis of A. Therefore,

$$\begin{aligned}||r_m(A)\epsilon_0||_A^2 &= \rho_0^T D^{-1} r_m(D) D r_m(D) D^{-1}\rho_0 \\ &= \rho_0^T r_m(D) D^{-1} r_m(D)\rho_0 \\ &= \sum_{i=1}^{n} \frac{r_m(\lambda_i)^2}{\lambda_i}\rho_{0,i}^2,\end{aligned}$$

which gives

$$||e_m||_A^2 = \sum_{i=1}^{n} \frac{r_m(\lambda_i)^2}{\lambda_i}\rho_{0,i}^2. \tag{2.12}$$

To obtain the residual polynomial r_m, we can use the recurrence relation between the Lanczos vectors and expressions for the Hessenberg matrix coefficients in equations (2.8) and (2.9). In particular,

$$\begin{aligned}\frac{1}{\eta_{j+1}}\mathbf{v}_{j+1} &= A\mathbf{v}_j - \delta_j\mathbf{v}_j - \eta_j\mathbf{v}_{j-1} \\ &= p_{j+1}(A)\mathbf{v}_1,\end{aligned}$$

where we define $p_{-1}(A) = 0, p_0(A) = I$. This gives

$$\begin{aligned}\eta_{j+1}p_{j+1}(A)\mathbf{v}_1 &= A\mathbf{v}_j - \delta_j\mathbf{v}_j - \eta_j\mathbf{v}_{j-1}, \\ &= \left(Ap_j(A) - \delta_j p_j(A) - \eta_j p_{j-1}(A)\right)\mathbf{v}_1,\end{aligned}$$

and therefore

$$p_{j+1}(A) = \frac{1}{\eta_{j+1}}\left((A-\delta_j)p_j(A) - \eta_j p_j(A)\right). \tag{2.13}$$

Furthermore, we have the following relation between the residual polynomial and the Lanczos polynomial [8, Section 3.2]

$$r_j(A) = (I - Aq_{j-1}(A))\mathbf{r}_0 = \frac{p_j(A)}{p_j(0)}\mathbf{r}_0. \tag{2.14}$$

This gives a way of calculating the residual polynomial r_m and thereby the error of the m^{th} iterate of the CG algorithm.

Additionally, the coefficients c_i of the solution polynomial q_m in equation (2.4) can be calculated. First we introduce a function that extracts the coefficients of a polynomial p

Definition 2.2. Let $p(t) = \sum_{i=0}^{n} c_i t^i$ be a polynomial of degree n. Then, the function coeff$(p; i)$ extracts the i^{th} coefficient of p such that coeff$(p; i) = c_i$.

Now using equation (2.14), we can write the solution polynomial as

$$Aq_{m-1}(A) = I - r_m(A)$$
$$r_m(\mathbf{0}) = I \implies A\sum_{i=1}^{m-1} c_{i-1}A^i = -\sum_{i=1}^{m} \mathsf{coeff}(r_m; i)A^i,$$

which implies

$$c_i = -\mathsf{coeff}(r_m; i+1), \quad i = 0, 1, \ldots, m-1. \tag{2.15}$$

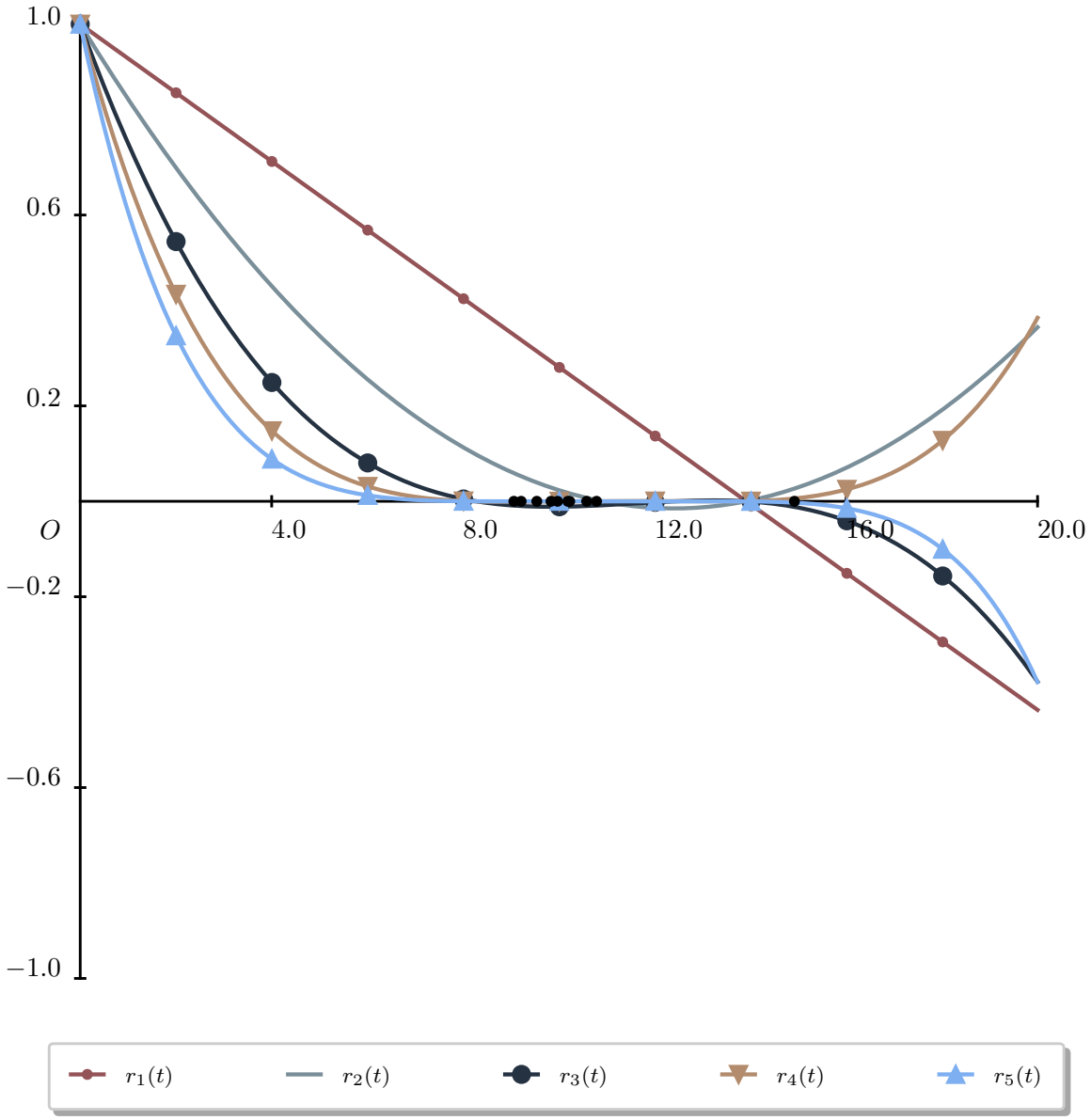


Figure 2.1: Residual polynomials resulting from successive CG iterations

The behavior of the residual polynomials is crucial for understanding the convergence properties of the CG method. In particular, the distribution of the eigenvalues of A significantly affects the convergence rate, as illustrated in figure 2.2. For all plots the lowest and highest eigenvalue in figure 2.2 are $\lambda_{\mathsf{min}} = 0.1$,

$\lambda_{\text{max}} = 0.9$ such that $f = \dfrac{\sqrt{\frac{\lambda_{\text{min}}}{\lambda_{\text{max}}}} - 1}{\sqrt{\frac{\lambda_{\text{min}}}{\lambda_{\text{max}}}} + 1}$ and the ratio $\dfrac{\|\mathbf{e}_m\|_A}{\|\mathbf{e}_0\|_A}$ is set to $\dfrac{10^{-6}}{\|\mathbf{u}_{\text{test}} - \mathbf{u}_0\|}$. The system size $N = 360$ is kept small and the system matrix A is diagonal so that it is numerically trivial to determine the exact solution $\mathbf{u}_{\text{test}}$. This results in an overall iteration bound

$$m_{\text{classical}} = \left\lceil \log_f \left(\frac{10^{-6}}{2\|\mathbf{u}_{\text{test}} - \mathbf{u}_0\|} \right) \right\rceil = 26$$

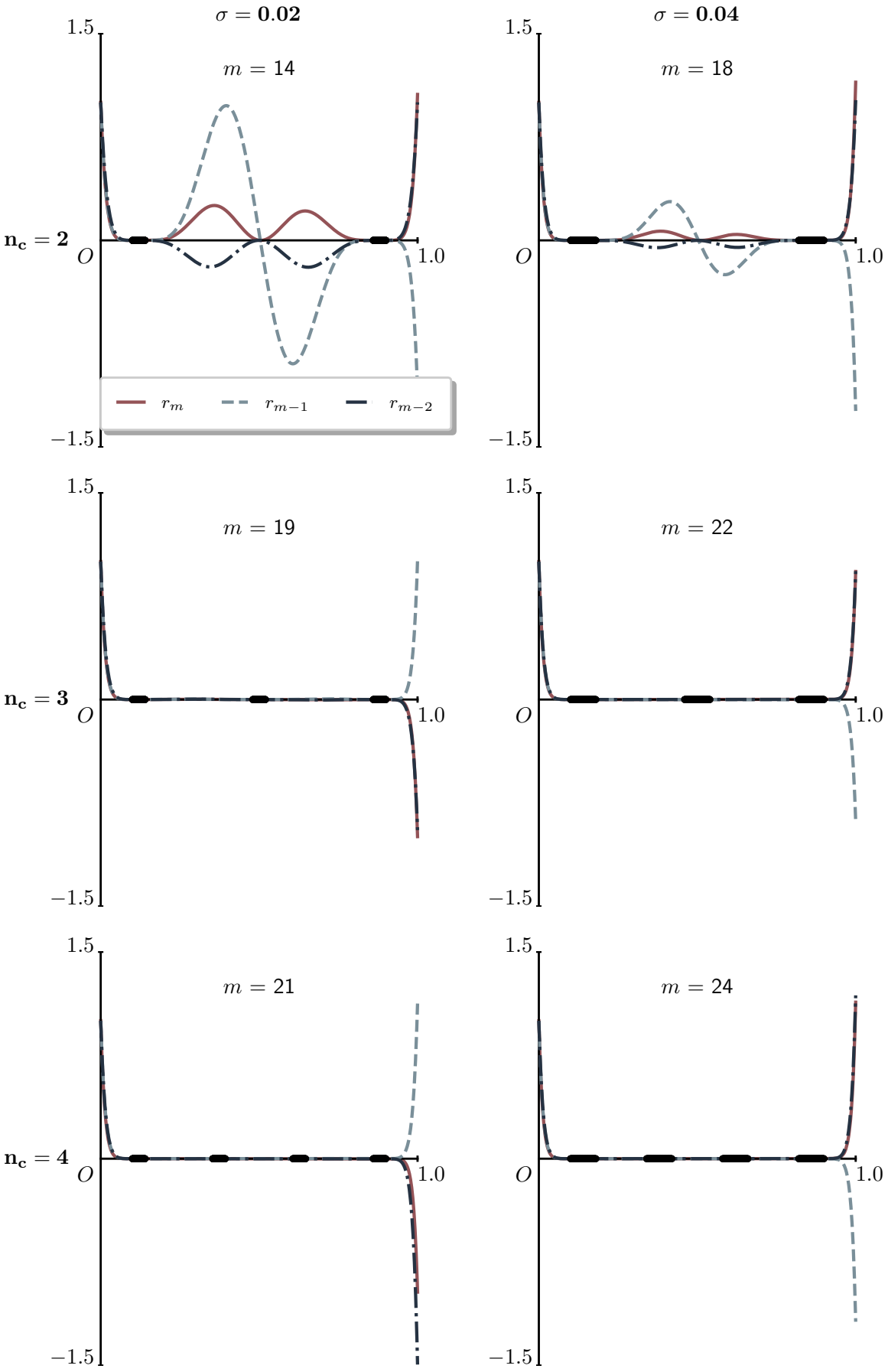


Figure 2.2: Plots of the last three CG residual polynomials for different eigenvalue distributions. n_c indicates the number of clusters and σ is the width of the cluster. The size of the system N and the condition number $\kappa(A)$ are kept constant. m indicates the number of iterations required for convergence.

Hence, the number of iterations required for convergence depends on the specific clustering of the eigenvalues, as pointed out for example in Kelley, Section 2.3.

From the behavior exhibited in figure figure 2.2 as well as from theorem 2.1 we can reason what the best and worst possible spectra for CG convergence are. That is, the best possible spectrum is one where eigenvalues are tightly clustered around distinct values, while the worst possible spectrum is one where the eigenvalues are evenly distributed across the whole range of the spectrum. This is illustrated in figure 2.3.

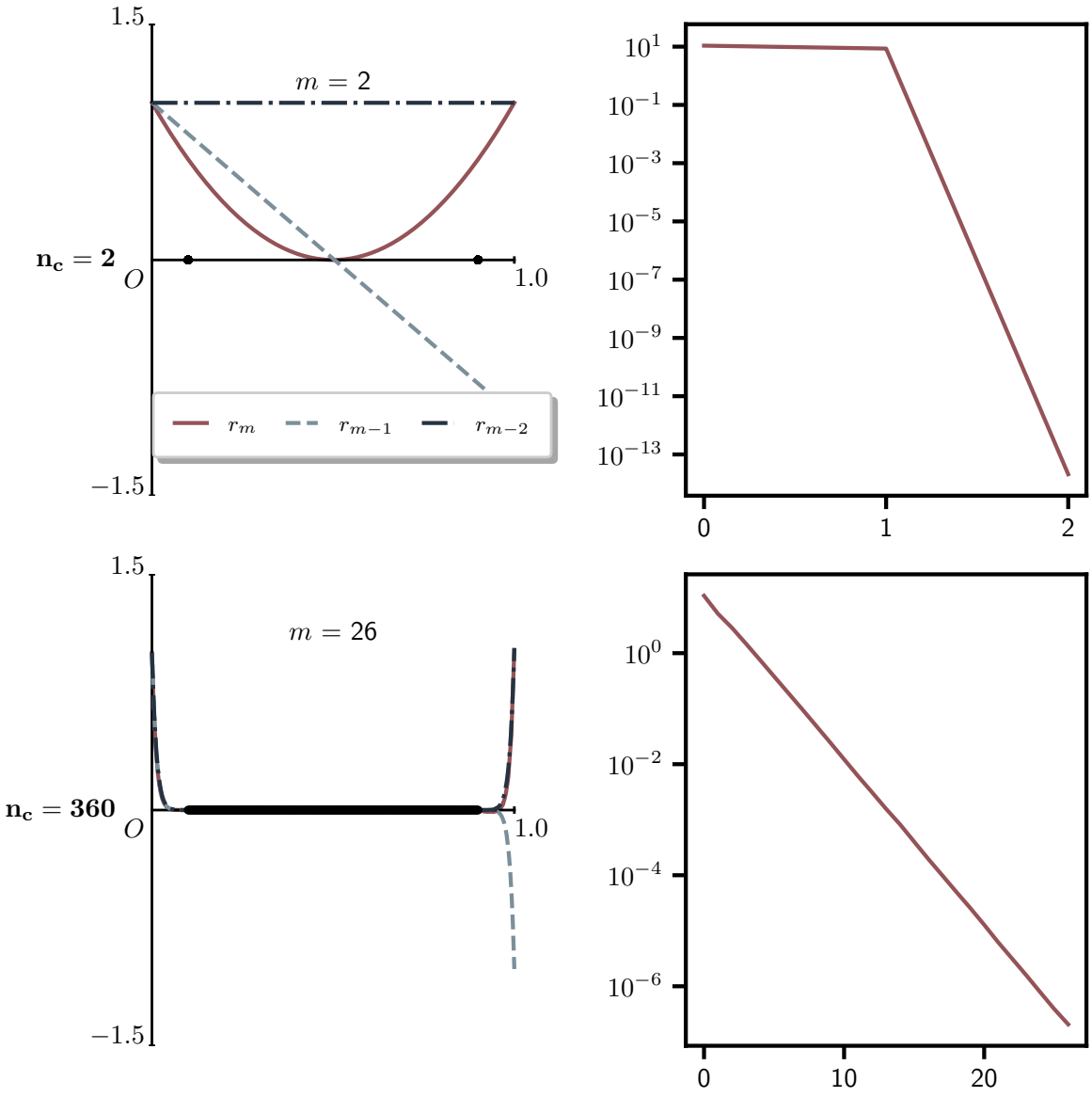


Figure 2.3: Best and worst possible spectra for CG convergence

2.1.6. Preconditioned CG

Suppose M is some SPD preconditioner, then variants of CG can be derived by applying M to the system of equations. The three main approaches are

i left

$$M^{-1}Ax = M^{-1}b$$

ii right

$$\begin{aligned} AM^{-1}u &= M^{-1}b \\ x &= M^{-1}u; \end{aligned}$$

iii symmetric or split

$$\begin{aligned} M &= LL^T \\ x &= L^{-T}u \\ L^{-1}AL^{-T}u &= L^{-1}b. \end{aligned}$$

Furthermore, all these variants are mathematically equivalent in some sense. Indeed, for the cases item preconditioner-type i and item preconditioner-type ii, we can rewrite the CG algorithm using the $M-$ or $M^{-1}-$inner products, respectively. In either case the iterates are the same. For instance for the left preconditioned CG, we define $z_j = M^{-1}\mathbf{r}_j$. Note that $M^{-1}A$ is self-adjoint with respect to the $M-$inner product, that is

$$(M^{-1}Ax, y)_M = (Ax, y) = (x, Ay) = (x, M^{-1}Ay)_M.$$

We use this to get a new expression for α_j. To that end, we write

$$\begin{aligned} 0 &= (\mathbf{r}_{j+1}, \mathbf{r}_j)_M \\ &= (z_{j+1}, \mathbf{r}_j) \\ &= (z_j - \alpha_j M^{-1}Ap_j, M^{-1}\mathbf{r}_j)_M \\ &= (z_j, M^{-1}\mathbf{r}_j)_M - \alpha_j (M^{-1}Ap_j, M^{-1}\mathbf{r}_j)_M \\ &= (z_j, z_j)_M - \alpha_j (M^{-1}Ap_j, z_j)_M \end{aligned}$$

and therefore

$$\alpha_j = \frac{(z_j, z_j)_M}{(M^{-1}Ap_j, z_j)_M}.$$

Using $p_{j+1} = z_{j+1} + \beta_j p_j$ and A-orthogonality of the search directions with respect to $M-$norm $(Ap_j, p_k)_M = 0$ (), we can write

$$\alpha_j = \frac{(z_j, z_j)_M}{(M^{-1}Ap_j, p_j)_M}.$$

Similarly, we can derive the equivalent expression of **??** as

$$\beta_j = \frac{(z_{j+1}, z_{j+1})_M}{(z_j, z_j)_M}.$$

This gives the left preconditioned CG algorithm in algorithm 2.

Algorithm 2 Left preconditioned CG [10, Algorithm 9.1]

$\mathbf{r}_0 = b - A\mathbf{u}_0$, $z_0 = M^{-1}\mathbf{r}_0$, $p_0 = z_0$, $\beta_0 = 0$
for $j = 0, 1, 2, \ldots, m$ **do**
  $\alpha_j = (z_j, z_j)_M / (M^{-1}Ap_j, p_j)_M = (\mathbf{r}_j, z_j)/(Ap_j, p_j)$
  $\mathbf{u}_{j+1} = \mathbf{u}_j + \alpha_j p_j$
  $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j Ap_j$
  $z_{j+1} = M^{-1}\mathbf{r}_{j+1}$
  $\beta_j = (z_{j+1}, z_{j+1})_M / (z_j, z_j)_M = (\mathbf{r}_{j+1}, z_{j+1})/(\mathbf{r}_j, z_j)$
  $p_{j+1} = z_{j+1} + \beta_j p_j$
end for

Furthermore it can be shown that the iterates of CG applied to the system with item preconditioner-type iii results in identical iterates [10, Algorithm 9.2].

2.2. Schwarz Methods

The content of this section is largely based on chapters 1,2, 4 and 5 of Dolean et al. about Schwarz methods.

The original Schwarz method was a way of proving that a Poisson problem on some complex domain Ω still has a solution.

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{2.16}$$

Existence is proved by splitting up the original complex domain in two (or more) simpler, possibly overlapping domains and solving the Poisson problem on each of these domains. The solution on the original domain is then the sum of the solutions on the subdomains. The method is named after Hermann Schwarz, who first introduced the method in 1869. The method has since been extended to more general problems and is now a popular method for solving partial differential equations.

Definition 2.3. The Schwarz algorithm is an iterative method based on solving subproblems alternatively in domains Ω_1 and Ω_2. It updates $(u_1^n, u_2^n) \to \left(u_1^{n+1}, u_2^{n+1}\right)$ by

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

The original Schwarz algorithm is sequential and, thereby, does not allow for parallelization. However, the algorithm can be parallelized. The Jacobi Schwarz method is a generalization of the original Schwarz method, where the subproblems are solved simultaneously and subsequently combined into a global solution. In order to combine local solutions into one global solution, an extension operator E_i, $i = 1, 2$ is used. It is defined as

$$E_i(v) = v \text{ in } \Omega_i, \quad E_i(v) = 0 \text{ in } \Omega \backslash \Omega_i.$$

Instead of solving for local solutions directly, one can also solve for local corrections stemming from a global residual. This is the additive Schwarz method (ASM). It is defined in algorithm 3.

Algorithm 3 Additive Schwarz method [4, Algorithm 1.2]

Compute residual $r^n = f - \Delta u^n$.
For $i = 1, 2$ solve for a local correction v_i^n:

$$-\Delta v_i^n = r^n \text{ in } \Omega_i, \quad v_i^n = 0 \text{ on } \partial\Omega_i$$

Update the solution: $u^{n+1} = u^n + \sum_{i=1}^{2} E_i(v_i)^n$.

The restrictive additive Schwarz method (RAS) is similar to ASM, but differs in the way local corrections are combined to form a global one. In the overlapping region of the domains it employs a weighted average of the local corrections. In particular, a partition of unity χ_i is used. It is defined as

$$\chi_i(x) = \begin{cases} 1, & x \in \Omega_i \setminus \Omega_{3-i}, \\ 0, & x \in \delta\Omega_i \setminus \delta\Omega \\ \alpha, & 0 \leq \alpha \leq 1, x \in \Omega_i \cap \Omega_{3-i} \end{cases}$$

such that for any function $w : \Omega \to \mathbb{R}$, it holds that

$$w = \sum_{i=1}^{2} E_i(\chi_i w_{\Omega_i}).$$

The RAS algorithm is defined in algorithm 4.

Algorithm 4 Restrictive additive Schwarz method [4, Algorithm 1.1]

Compute residual $r^n = f - \Delta u^n$.
For $i = 1, 2$ solve for a local correction v_i^n:

$$-\Delta v_i^n = r^n \text{ in } \Omega_i, \quad v_i^n = 0 \text{ on } \partial\Omega_i$$

Update the solution: $u^{n+1} = u^n + \sum_{i=1}^{2} E_i(\chi_i v_i^n)$.

2.2.1. Schwarz methods as preconditioners

Let $\mathcal{N}$ be set containing all indices of degrees of freedom in the domain Ω and N_{sub} be the number of subdomains such that

$$\mathcal{N} = \sum_{i=1}^{N_{\text{sub}}} \mathcal{N}_i,$$

$\mathcal{N}_i$ is the set of indices of degrees of freedom in the subdomain Ω_i.

Furthermore, let $R_i \in \mathcal{R}^{\#\mathcal{N}_i \times \#\mathcal{N}}$, R_i^T and D_i be the discrete versions of the restriction, extension and partition of unity operators such that

$$\mathcal{R}^{\#\mathcal{N}} \ni U = \sum_{i=1}^{\mathcal{N}_{\text{sub}}} R_i^T D_i R_i U.$$

Note that D_i is a diagonal matrix where the entries are the values of the partition of unity function χ_i evaluated for each degree of freedom. Consider for instance, a multidimensional FEM problem, in which $\mathcal{T}$ is the triangulation of the domain Ω and $\mathcal{T}_i$ is the triangulation of the subdomain Ω_i such that [4, Equation 1.27]

$$\Omega_i = \cup_{\tau \in \mathcal{T}_i} \tau.$$

In this case [4, Equation 1.28]

$$\mathcal{N}_i = \{k \in \mathcal{N} | \text{meas}(\text{supp}(\phi_k) \cap \Omega_i) > 0\},$$

and we can define

$$\mu_k = \#\{j | 1 \leq j \leq N_{\text{sub}} \text{ and } k \in \mathcal{N}_j\}.$$

Finally, this leads to

$$(D_i)_{kk} = \frac{1}{\mu_k}, \; k \in \mathcal{N}_i. \tag{2.17}$$

Although the original Schwarz method is not a preconditioner, the ASM and RAS methods can be used as such. Originally the Schwarz method is a fixed point one [4, Definitions 1.12 and 1.13]

$$u^{n+1} = u^n + M^{-1} r^n, \; r^n = f - Au^n,$$

where M equals, but is not limited to, one of the following matrices;

$$M_{\text{ASM}} = \sum_{i=1}^{N_{\text{sub}}} R_i^T (R_i A R_i^T)^{-1} R_i, \tag{2.18a}$$

$$M_{\text{RAS}} = \sum_{i=1}^{N_{\text{sub}}} R_i^T D_i (R_i A R_i^T)^{-1} R_i. \tag{2.18b}$$

Both M_{ASM} and M_{RAS} are symmetric and positive definite and can be used as preconditioners.

Optimized Schwarz methods and corresponding preconditioners can also be constructed by including more interface conditions (Robin or Neumann) in the subproblems. One such example is the Optimized Restrictive Additive Schwarz method (ORAS) discussed in [4, Chapter 2].

2.2.2. Convergence original Schwarz method

In this section the Schwarz problem stated in definition 2.3 is solved in the one- and two-dimensional case. The convergence of the original Schwarz method is then discussed.

1D case

Let $L > 0$ and the domain $\Omega = (0, L)$. The domain is split into two subdomains $\Omega_1 = (0, L_1)$ and $\Omega_2 = (l_2, L)$ such that $l_2 \leq L_1$. Instead of solving for $u_{1,2}$ directly, we solve for the error $e^n_{1,2} = u^n_{1,2} - u_{|\Omega_i}$, which by linearity of the Poisson problem as well as the original Schwarz algorithm satisfies

$$\begin{aligned} -\frac{e_1^{n+1}}{dx^2} &= f \text{ in } (0, L_1), \\ e_1^{n+1}(0) &= 0, \\ e_1^{n+1}(L_1) &= e_2^n(L_1); \end{aligned} \qquad \text{then} \qquad \begin{aligned} -\frac{e_2^{n+1}}{dx^2} &= f \text{ in } (l_2, L), \\ e_2^{n+1}(l_2) &= e_1^{n+1}(l_2), \\ e_2^{n+1}(L) &= 0. \end{aligned}$$

The solution to the error problem is

$$e_1^{n+1}(x) = \frac{x}{L_1} e_2^n(L_1), \quad e_2^{n+1}(x) = \frac{L - x}{L - l_2} e_1^{n+1}(l_2).$$

Thes functions increase linearly from the boundary of the domain to the boundary of the overlapping region. The error at for instance $x = L_1$ is updated as

$$e_2^{n+1}(L_1) = \frac{1 - \delta/(L - l_2)}{1 + \delta/l_2} e_2^n(L_1),$$

where $\delta = L_1 - l_2 > 0$ is the overlap. The error is reduced by a factor of

$$\rho_{\text{1D}} = \frac{1 - \delta/(L - l_2)}{1 + \delta/l_2}, \tag{2.19}$$

which indicates the convergence becomes quicker as the overlap increases [4, Section 1.5.1].

2D case

In the 2D case two half planes are considered $\Omega_1 = (-\infty, \delta) \times \mathbb{R}$ and $\Omega_2 = (\delta, \infty) \times \mathbb{R}$. Following the example of Dolean et al. the problem is

$$\begin{aligned} -(\eta - \Delta)u &= f \text{ in } \mathbb{R}^2, \\ u &\text{ bounded at infinity.} \end{aligned}$$

Proceeding in similar fashion as the one-dimensional case, the error $e^{n+1}_{1,2}$ can be solved for in the two subdomains. This is done via a partial Fourier transform of the problem in the y-direction yielding an ODE for the transformed error $\hat{e}^{n+1}_{1,2}$, which can be solved explicitly with the ansatz

$$\hat{e}_{1,2}^{n+1}(x, k) = \gamma_1(k) e^{\lambda_+(k)x} + \gamma_2(k) e^{\lambda_-(k)x},$$

where $\lambda_\pm(k) = \pm\sqrt{k^2 + \eta}$. By using the interface conditions we find

$$\gamma_i^{n+1}(k) = \rho(k; \eta, \delta)^2 \gamma_i^{n-1}(k),$$

such that the convergence factor is [4, Equation 1.36]

$$\rho_{\text{2D}}(k; \eta, \delta) = e^{-\delta\sqrt{\eta + k^2}} \tag{2.20}$$

which indicates that the convergence is quicker as the overlap increases as before. Next to this, it also shows that the convergence is quicker for higher frequencies k.

2.2.3. Need for a coarse space

Following upon the results in the previous section 2.2.2 it is clear that the convergence of the Schwarz method not only depends on the extent of the overlap between various subdomains, but on the frequency components of the solution as well. In a general sense this means that low frequency modes need for instance at least N_{sub} steps to travel from one end of a square domain to the other. This in turns causes plateaus in the convergence of the Schwarz method. To overcome this, we can perform a Galerkin projection of the error onto a coarse space. That is we solve

$$\min_{\beta} ||A(x + R_0^T \beta) - f||^2,$$

where Z is a matrix representing the coarse space. The solution to this problem is

$$\beta = (R_0 A R_0^T)^{-1} R_0 r,$$

where $r = f - Ax$ is the residual.

The coarse space R_0 can be constructed in various ways. The classical way is called the Nicolaides space [4, Section 4.2], which uses the discrete partition of unity operators D_i as examplified in equation (2.17) to get

$$R_0 = \sum_{i=1}^{N_{\text{sub}}} R_i^T D_i R_i. \tag{2.21}$$

Note that the course space has a block-diagional form.

Finally the coarse space correction term can be added to the Schwarz preconditioners equations (2.18a) and (2.18b) to get the following preconditioners

$$M_{\text{ASM,2}} = R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{N_{\text{sub}}} R_i^T (R_i A R_i^T)^{-1} R_i, \tag{2.22a}$$

$$M_{\text{RAS,2}} = R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{N_{\text{sub}}} R_i^T D_i (R_i A R_i^T)^{-1} R_i. \tag{2.22b}$$

2.2.4. Two-level additive Schwarz method

In this section the we will construct a coarse space for a Poisson problem with a constant scalar coefficient on arbitrary domain like in problem 2.16. However, the method is applicable to more general (highly) heterogeneous scalar problems, like the Darcy problem (see **??**). The coarse space is constructed using the eigenfunctions corresponding to the smallest m_j eigenvalues resulting from a local eigenproblem in each subdomain Ω_j. The coarse space is then constructed by taking the union of the m_j eigenvectors corresponding to the smallest eigenvalues in each subdomain glued together by the partition of unity functions χ_j. All of this can be found in [4, Sections 5.1-5.5].

This coarse space is subsequently used to construct the two level additive Schwarz preconditioner, and bounds for its condition number are provided as well.

Slowly convergent modes of the Dirichlet-to-Neumann map

As seen in section 2.2.2 the local error in any subdomain in the Schwarz method satisfies the original problem without forcing, i.e. right hand side $f = 0$. At the interface the local error has a Dirichlet boundary condition that equals the error of the neighbouring subdomain. Additionally, the convergence factor, e.g. ρ_{2D}, depends on the frequency of the modes present in the local error. In particular, small frequencies appear to have slow convergence. The question thus becomes how to get rid of these small frequency modes in the local errors of all subdomains.

One possible answer is the so-called Dirichlet-to-Neumann map [4, Definition 5.1]

Definition 2.4. (Dirichlet-to-Neumann map for a Poisson problem) For any function defined on the interface $u_{\Gamma_j} : \Gamma_j \mapsto \mathbb{R}$, we consider the Dirichlet-to-Neumann map

$$\mathrm{DtN}_{\Omega_j}\left(u_{\Gamma_j}\right) = \left.\frac{\partial v}{\partial \mathbf{n}_j}\right|_{\Gamma_j},$$

where $\Gamma_j := \partial\Omega_j \backslash \partial\Omega$ and v satisfies

$$\begin{aligned} -\Delta v &= 0 && \text{in } \Omega_j, \\ v &= u_{\Gamma_j} && \text{on } \Gamma_j, \\ v &= 0 && \text{on } \partial\Omega_j \cap \partial\Omega. \end{aligned} \tag{2.23}$$

The Dirichlet-to-Neumann map essentially solves for an error-like variable v that satisfies the Dirichlet local interface (or global boundary) conditions. DtN then maps the interface condition to the normal derivative of v on the interface, i.e. the Neumann condition. Now, as stated above and illustrated in [4, Figure 5.2] the low frequency modes of the error correspond to those modes that are nearly constant accross an interface, for which the Neumann condition is close to zero. So the problem of slowly convergent modes in the error of the Schwarz method is equivalent to a problem of finding eigenpairs of the DtN operator.

Hence we aim to solve the eigenvalue problem

$$\mathrm{DtN}_{\Omega_j}(v) = \lambda v,$$

which can be reformulated in the variational form. To that end let w be a test function that is zero on $\delta\Omega$. Multiply both sides of equation (2.23) by w, integrate over Ω_j and apply Green's theorem to get

$$\int_{\Omega_j} \nabla v \cdot \nabla w - \lambda \int_{\Omega_j} \frac{\partial v}{\partial \mathbf{n}_j} w, \quad \forall w.$$

Then, use the eigen property of v and fact that w is zero on $\delta\Omega$ to get the eigen problem in the variational form

$$\text{Find } (v, \lambda) \text{ s.t. } \int_{\Omega_j} \nabla v \cdot \nabla w - \lambda \int_{\Gamma_j} vw = 0, \quad \forall w. \tag{2.24}$$

FEM discretization

The discretisation is done in the context of the finite element method. To that end we consider a triangulation $\mathcal{T}$ of the domain Ω and a partition of unity χ_j. Denote by $V_{h,0} = V_h \cap H_0^1(\Omega)$ the space of piecewise continuous functions $v_h \in H_0^1(\Omega)$ with respect to $\mathcal{T}$. Let the basis of $V_{h,0}$ be given by $\{\phi_k\}_{\in\mathcal{N}}$. The FE formulation of the Poisson problem equation (2.16) follows from the variational form

$$a(u_h, v_h) = (f, v_h), \quad \forall u_h, v_h \in V_{h,0},$$

and is given by

$$A\mathbf{u} = b, \quad A_{ij} = a(\phi_j, \phi_i),\ b_i = (f, \phi_i) \quad \forall i, j \in \mathcal{N}. \tag{2.25}$$

Next to this we need a way of interpolating functions in $C(\Omega)$ to V_h. This is done by the interpolation operator $I_h : C(\Omega) \mapsto V_{h,0}$, which is defined by

$$\mathcal{I}_h v = \sum_{i\in\mathcal{N}} v(x_i)\phi_i,$$

where x_i are the nodes of the triangulation $\mathcal{T}$. $\mathcal{I}_h$ is stable with respect to the a-norm, that is

$$\|I_h(v)\|_a \leq C_{I_h}\|v\|_a.$$

As before we partition Ω into N_{sub} subdomains Ω_j, which overlap each other by one or several layers of elements in the triangulation $\mathcal{T}$. We make the the following observations

I For every degree of freedom $k \in \mathcal{N}$, there is a subdomain Ω_j such that ϕ_k has support in Ω_j [4, Lemma 5.3].

II The maximum number of subdomains a mesh element can belong to is given by

$$k_0 = \max_{\tau\in\mathcal{T}} \left(\#\{j | 1 \leq j \leq N_{\text{sub}} \text{ and } \tau \subset \Omega_j\}\right).$$

III The minimum number of colors needed to color all subdomains so that no two adjacent subdomains have the same color is given by

$$N_c \geq k_0$$

IV The minimum overlap for any subdomain Ω_j with any of its neihbouring subdomains is given by

$$\delta_j = \inf_{x \in \Omega_j \setminus \cup_{i \neq j} \bar{\Omega}_i} \mathsf{dist}(x, \partial\Omega_j \setminus \partial\Omega).$$

V The partition of unity functions $\{\chi_j\}_{j=1}^{N_{\mathsf{sub}}} \subset V_h$ are such that

V.a $\chi_j(x) \in [0,1], \quad \forall x \in \bar{\Omega}, j = 1, \ldots, N_{\mathsf{sub}}$,

V.b $\mathsf{supp}(\chi_j) \subset \bar{\Omega}_j$,

V.c $\sum_{j=1}^{N_{\mathsf{sub}}} \chi_j(x) = 1, \quad \forall x \in \bar{\Omega}$,

V.d $\|\nabla \chi_j(x)\| \leq \frac{C_\chi}{\delta_j}$,

and are given by

$$\chi_j(x) = I_h \left(\frac{d_j(x)}{\sum_{j=1}^{N_{\mathsf{sub}}} d_j(x)} \right),$$

where

$$d_j(x) = \begin{cases} \mathsf{dist}(x, \partial\Omega_j), & x \in \Omega_j, \\ 0, & x \in \Omega \setminus \Omega_j. \end{cases}$$

VI The overlap region for any subdomain is given by

$$\Omega_j^\delta = \{x \in \Omega_j | \chi_j < 1\}.$$

The extension operator $E_j : V_{h,0}(\Omega_j) \to V_h$ is defined by

$$V_h = \sum_{j=1}^{N_{\mathsf{sub}}} E_j V_{h,0}(\Omega_j),$$

which is gauranteed by item ASM observation I.

Note that using the extension operator we can show that all the local bilinear forms are positive definite as

$$a_{\Omega_j}(v, w) = a(E_j v, E_j w) \geq \alpha \|E_j v\|_a^2, \quad \forall v, w \in V_{h,0}(\Omega_j),$$

and a is positive definite.

Finally, we define the a-symmetric projection operators $\tilde{\mathcal{P}}_j : V_{h,0} \to V_h$ and $\mathcal{P}_j : V_h \to V_h$ defined by

$$a_{\Omega_j}(\tilde{\mathcal{P}}_j u, v_j) = a(u, E_j v_j) \quad \forall v_j \in V_{h,0},$$
$$\mathcal{P} = E_j \tilde{\mathcal{P}}_j.$$

then their matrix counterparts are given by

$$\tilde{P}_j = A_j^{-1} R_j^T A,$$
$$P_j = R_j^T A_j^{-1} R_j^T A,$$

where $A_j = R_j A R_j^T$. From this we can construct the two-level additive Schwarz method as

$$M_{\mathsf{ASM},2}^{-1} A = \sum_{j=1}^{N_{\mathsf{sub}}} P_j. \tag{2.26}$$

2.2.5. Convergence of two-level additive Schwarz

In the following we denote

$$\mathcal{P}_{\mathsf{ad}} = \sum_{j=1}^{N_{\mathsf{sub}}} \mathcal{P}_j,$$

and correspondingly,

$$P_{\mathsf{ad}} = \sum_{j=1}^{N_{\mathsf{sub}}} P_j.$$

In the context of this thesis the two-level additive Schwarz method is used in combination with a Krylov subspace method (**??**), in which case convergence rate depends on the entire spectrum of eigenvalues (section 2.1.5). However, an upperbound for the convergence rate (section 2.1.4) can be derived from the condition number of P_{ad} via equation (2.10).

Using the fact that P_{ad} is symmetric with respect to the a-norm, we can write

$$\kappa(P_{\mathsf{ad}}) = \frac{\lambda_{\mathsf{max}}}{\lambda_{\mathsf{min}}},$$

where

$$\lambda_{\mathsf{max}} = \sup_{v \in V_h} \frac{a(\mathcal{P}_{\mathsf{ad}})}{a(v,v)}, \quad \lambda_{\mathsf{min}} = \inf_{v \in V_h} \frac{a(\mathcal{P}_{\mathsf{ad}})}{a(v,v)}.$$

Additionally, we can employ the a-orthogonality of the projection operators to get

$$\frac{a(\mathcal{P}_j u, u)}{\|u\|_a^2} = \frac{a(\mathcal{P}_j u, \mathcal{P}_j u)}{\|u\|_a^2} \leq 1.$$

Going further, we can pose that the projection operators defined by the sum of projection operators $\mathcal{P}_j$ of like-colored subdomains are a-orthogonal to each other. This is due to the fact that the partition of unity functions χ_j are such that they are zero on the interface of like-colored subdomains (see item ASM observation III). To that end, define

$$\mathcal{P}_{\Theta_i} = \sum_{j \in \Theta_i} \mathcal{P}_j,$$

where Θ_i is the set of indices of subdomains with color i and $i = 1, \ldots, N_c$. Then, we can write [4, Lemma 5.9]

$$\begin{aligned}
\lambda_{\mathsf{max}}(\mathcal{P}_{\mathsf{ad}}) &= \sup_{v \in V_h} \sum_{i=1}^{N_c} \frac{a(\mathcal{P}_{\Theta_i} v, v)}{a(v,v)} \\
&\leq \sum_{i=1}^{N_c} \sup_{v \in V_h} \frac{a(\mathcal{P}_{\Theta_i} v, v)}{a(v,v)} \\
&\leq N_c + 1,
\end{aligned}$$

where the extra one comes from the coarse projection operator $\mathcal{P}_0$. Note that this bound can be made sharper by using item ASM observation II to get $\lambda_{\mathsf{max}}(\mathcal{P}_{\Theta_i}) \leq k_0 + 1$.

On the other hand, it can be shown that the minimum eigenvalue satisfies provided that $v \in V_h$ admits a C_0-stable decomposition [4, Theorem 5.11]

$$\lambda_{\mathsf{min}}(\mathcal{P}_{\mathsf{ad}}) \geq C_0^{-2}.$$

Finally, we can write the condition number of the two-level additive Schwarz preconditioner as

$$\kappa(P_{\mathsf{ad}}) \leq (N_c + 1)\, C_0^2. \tag{2.27}$$

The value of C_0 depends on the projection operator Π_j to the chosen coarse space V_0 for each subdomain.

I. **Nicolaides coarse space** The projection operator is defined as

$$\Pi_j^{\text{Nico}} u = \begin{cases} (\frac{1}{|\Omega_j|}\int_{\Omega_j} u)\mathbf{1}_{\Omega_j}, & \delta\Omega_j \cap \delta\Omega = \emptyset, \\ 0, & \text{otherwise}, \end{cases} \tag{2.28}$$

which gives rise to the following basis functions in $V_{h,0}$

$$\Phi_j^{\text{Nico}} = I_h(\chi_j \mathbf{1}_{\Omega_j}).$$

Then,

$$V_0 = \text{span}\{\Phi_j^{\text{Nico}}\}_{j=1}^{N_{\text{sub}}},$$

and

$$\dim V_0 = \text{the number of floating subdomains},$$

that is the number of subdomains that are not connected to the boundary of the domain Ω. In this case [4, Theorem 5.16]

$$C_0^2 = \left(8 + 8C_\chi^2 \max_{j=1}^{N_{\text{sub}}} \left[C_P^2 + C_{\text{tr}}^{-1}\frac{H_j}{\delta_j}\right] k_0 C_{I_h}(k_0+1) + 1\right), \tag{2.29}$$

where H_j is the diameter of the subdomain Ω_j, C_p the Poincaré constant following from [4, Lemma 5.18] and C_{tr} is the trace constant.

II. **Local eigenfunctions coarse space** The projection operator is defined as

$$\Pi_j^{\text{spec}} u = \sum_{k=1}^{m_j} a_{\Omega_j}(u, v_k^{(j)}) v_k^{(j)},$$

where $v_k^{(j)}$ is the k^{th} eigenfunction resulting from the eigenproblem in equation (2.24). The basis functions in $V_{h,0}$ are then given by

$$\Phi_{j,k}^{\text{spec}} = I_h(\chi_j v_k^{(j)}),$$

resulting in the coarse space

$$V_0 = \text{span}\{\Phi_{j,k}^{\text{spec}}\}_{j=1,k=1}^{N_{\text{sub}},m_j},$$

with dimension

$$\dim V_0 = \sum_{j=1}^{N_{\text{sub}}} m_j.$$

In this case [4, Theorem 5.17]

$$C_0^2 = \left(8 + 8C_\chi^2 \max_{j=1}^{N_{\text{sub}}} \left[C_P^2 + C_{\text{tr}}^{-1}\frac{1}{\delta_j \lambda_{m_j+1}}\right] k_0 C_{I_h}(k_0+1) + 1\right). \tag{2.30}$$

3

Related Work

3.1. The spectral gap arising in Darcy problems

In a Darcy problem, high-contrast $\mathcal{C}(x)$ (e.g., 10^6 in conductive channels vs. 10^{-6} in barriers) means flow concentrates in high-permeability regions, while low-permeability zones resist flow. This heterogeneity introduces modes (eigenvectors) that are nearly constant or slowly varying over low-$\mathcal{C}(x)$ regions, contributing small eigenvalues to A. These modes represent "trapped" or "isolated" behaviors disconnected by the contrast.

Small eigenvalues arise when the bilinear form $a(u,v) = \int_\Omega \mathcal{C}(x)\nabla u \cdot \nabla v \, dx$ yields low energy for certain test functions v. In high-contrast cases, if $\mathcal{C}(x)$ is tiny in a subdomain, ∇u must be large there to balance the equation, but FEM basis functions often cannot resolve this without fine meshes. Instead, coarse bases produce modes where energy is minimized, leading to eigenvalues close to zero. This is exacerbated as contrast grows, adding more such modes.

High-contrast $\mathcal{C}(x)$ can split the spectrum into clusters: large eigenvalues tied to high-$\mathcal{C}(x)$ regions (where gradients dominate) and small eigenvalues tied to low-$\mathcal{C}(x)$ regions (where flow stagnates). In [5] note that standard FEM misses these small eigenvalues unless enriched bases capture fine-scale features.

3.2. Tailored coarse spaces for high-contrast problems

Various methods for constructing a coarse space that are both scalable and robust to high contrast in a problem coefficient.

3.2.1. MsFEM

The Multiscale Finite Element Method (MsFEM), as presented in [6], constructs a coarse space based on five key assumptions (C1-C5). These assumptions ensure stability and accuracy by defining how the coarse space interacts with the fine-scale problem. Local coarse grid basis functions are obtained by solving the homogeneous version of the system equation, meaning they do not include external forcing terms. The construction of these basis functions requires specific boundary conditions, categorized as M1-M4, which control their behavior at interfaces. The method distinguishes between linear and oscillatory boundary conditions for local problems, affecting the resulting coarse space. Coarse grid basis functions are computed as harmonic extensions of basis functions restricted to edges or faces, ensuring continuity across subdomains. The restriction operator R_0 is then derived from these basis functions, as given in Equation 2.12 of [6]. Additionally, the method introduces robustness indicators, $\pi(\alpha)$ and $\gamma(\alpha)$, which quantify the stability of the coarse space and its effectiveness in capturing fine-scale features.

3.2.2. ACMS

The Approximate Component Mode Synthesis (ACMS) method, detailed in [9], introduces a separation of scales with fine and coarse triangulations, denoted as $\mathcal{T}_h$ and $\mathcal{T}_H$. The coarse problem is decomposed into two components: $u_c = u_I + u_\Gamma$, where u_I represents the inner part and u_Γ the interface contribution. This extends MsFEM by incorporating vertex-specific, edge-specific, and fixed-interface basis functions, where MsFEM corresponds solely to the vertex-specific functions. The vertex-specific basis functions are defined as harmonic extensions of trace values on the interface set Γ. Edge-specific basis functions, on the other hand, arise from an eigenvalue problem defined on an edge e, while fixed-interface basis functions correspond to eigenmodes of an eigenvalue problem within a coarse element T.

ACMS supports two types of coarse spaces, depending on whether Dirichlet (DBC) or Neumann (NBC) boundary conditions are applied. Under DBCs, MsFEM basis functions are combined with edge-specific basis functions that match on a shared edge e_{ij} between subdomains Ω_i and Ω_j. These functions are constructed from the harmonic extension of eigenmodes defined on the edge e_{ij}, with a scaled bilinear form on the right-hand side. Only eigenmodes corresponding to eigenfrequencies below a set tolerance are retained. With NBCs, both MsFEM and edge-specific basis functions are modified. MsFEM functions remain defined on an edge e_{ij} and satisfy a Kronecker-delta vertex condition but are now obtained via a generalized eigenvalue problem on a slab of width kh, denoted η_{ij}^{kh}. The edge-specific functions are similarly defined through a generalized eigenvalue problem on the slab but without enforcing DBCs. Solving these eigenvalue problems can be computationally efficient by employing mass matrix lumping techniques.

3.2.3. (R)GDSW

The Generalized Dryja-Smith-Widlund (GDSW) method, introduced by [3], partitions the computational domain into non-overlapping subdomains and further divides degrees of freedom (DOFs) into interior and interface nodes. The only required input for the method is a coarse space G, whose columns span the rigid body modes of the subdomains. The restriction operators R_Γ and R_I project onto interface and interior DOFs, respectively, with subdomain-specific versions such as R_{Γ_j}.

The coarse solution on the interface set is given by

$$u_{0,\Gamma} = \sum_{j}^{N_{\text{sub}}} R_{\Gamma_j}^T G_{\Gamma_j} q_j = \Phi_\Gamma q,$$

where q represents the coarse space coefficients. The complete coarse solution is then given by

$$u_0 = R_\Gamma^T u_{0,\Gamma} + R_I^T u_{0,I},$$

where R_I is derived from energy-minimizing extensions of $u_{0,\Gamma}$ into the subdomain interiors. Applying the energy minimization principle to the potential $u_0 A u_0$ leads to the definition of Φ_I, which governs the interior contributions.

3.2.4. AMS

The Algebraic Multiscale (AMS) method, introduced in ... studied in [1], also relies on domain decomposition into non-overlapping subdomains, followed by a further subdivision of interface nodes into edge, vertex, and face nodes (in 3D). The method eliminates lower diagonal blocks in the system matrix to facilitate efficient computation. Like (R)GDSW, AMS employs the energy minimization principle to obtain Φ_I, ensuring an optimal coarse space representation.

3.3. CG convergence in case of non-uniform spectra

To do ►*Comparison of methodologies – Different approaches used in related work.*◄

1. **CG in FP-arithmetic**: [8] Adapt Riemann-Stieltjes integral with special distribution function and Gauss quadrature to sharpen CG iteration bound instead of accounting for FP arithmetic errors and/or loss of orthogonality
2. **CG convergence rate in case of non-uniform spectra**: [11]
3. **Sharpened CG iteration bound using Chebyshev polynomials**: [2], Clever use of Chebyshev polynomials in case of non-uniform spectra caused by high-contrast $\mathcal{C}(x)$ in Darcy problems.

4

Research questions

4.1. Main research question

The main research question in this work is as follows:

Research Question. How can we sharpen the CG iteration bound for Schwarz-preconditioned high-contrast heterogeneous elliptic problems beyond the classical condition number-based bound?

For instance, in [1], the AMS and GDSW preconditioners significantly outperform the RGDSW preconditioner, despite all three having similar condition numbers. The key differences appear in their spectral gap and cluster width, highlighting the need for additional spectral characteristics to refine existing bounds.

4.2. Subsidiary research questions

Subsidiary Questions. To answer the main research question, we address the following subsidiary questions:

Q1 What spectral characteristics, like the condition number, can we define to estimate the distribution of eigenvalues in the eigenspectrum in the case of high-contrast heterogeneous problems?

Q2 How can we estimate any of the spectral characteristics defined in Q1 for the eigenspectrum in the particular case of a model Darcy problem?

Q3 Given a certain eigenspectrum, how can we sharpen the CG iteration bound?

Q4 How does the sharpened bound from Q3 perform for an unpreconditioned Darcy problem in comparison with the classical bound in equation (2.10)?

Q5 How does the performance described in Q4 of a sharpened bound vary with the measures found in Q1?

Q6 How can we employ the sharpened bound to distinguish between the performance of Schwarz-like preconditioners?

4.3. Motivation

This research is important because current studies primarily focus on selecting between different Schwarz preconditioners for the Darcy problem, yet condition numbers fail to distinguish them effectively. The ability to differentiate preconditioners based on spectral properties would improve the selection process. Applying sharpened bounds could lead to better predictions of preconditioner performance and improved efficiency in solving high-contrast problems.

4.4. Challenges

The main challenge lies in quantitatively estimating the spectrum of the preconditioned system. The literature provides a priori condition number estimates for various Schwarz preconditioners. For instance, in the simple cases of the additive Schwarz preconditioner with either a item ASM coarse space I or item ASM coarse space II an a priori estimate for the condition number is given by equation (2.27) in combination with either equation (2.29) or equation (2.30), respectively. The same can be said for the MsFEM and ACMS preconditioners. Despite this, there is no established method for estimating the full eigenspectrum. Without such an estimate, refining the CG iteration bound remains difficult. Overcoming this limitation is central to this work.

5

Preliminary Results

The results described in this chapter are adapted from the ideas discussed in [2, Section 4]. Therein Axelsson presents a sharpened CG iteration bound for two particular eigenspectra, which are described below.

5.1. Two cluster case

On the eigenspectrum of A, consider two intervals $[a, b]$ and $[c, d]$ with $a < b < c < d$ such that all eigenvalues of A are contained in the union of these two intervals. Additionally, we have $\kappa(A) = \frac{d}{a}$. We treat the following two cases simultaneously

$$\sigma_1(A) = [a, b] \bigcup [c, d] \tag{5.1}$$

$$\sigma_2(A) = [c, d] \bigcup_{\substack{i=1 \\ \lambda_i \in [a,b]}}^{N_{\text{tail}}} \lambda_i \tag{5.2}$$

where N_{tail} is the number of eigenvalues in the tail. The first case is a two-cluster eigenspectrum, while the second case has one cluster and a tail of eigenvalues.

In order to derive a CG iteration bound for these two cases we proceed as in the classical case laid out in 2.1.4. We know CG is optimal in the A-norm by equation (2.10), from which it follows that the error at the m^{th} iterate can be bounded as

$$||e_m||_A \leq \min_{r \in \mathcal{P}_m, r(0)=1} \max_{\lambda \in \sigma_i(A)} |r(\lambda)| ||\epsilon_0||_A, \text{ for } i = 1, 2, \tag{5.3}$$

To get an upper bound for m equation (5.3) suggests we look for a polynomial $r_{\bar{m}}$ of degree $\bar{m}$ that satisfies

$$\min_{r \in \mathcal{P}_{\bar{m}}, r(0)=1} \max_{\lambda \in \sigma_i(A)} |r(\lambda)| \leq \frac{||e_m||_A}{||\epsilon_0||_A} = \epsilon, \text{ for } i = 1, 2,$$

in which ϵ is the relative error.

Axelsson suggests we use not one, monolithic residual polynomial function, but a multiplication of two residual polynomial functions $\hat{r}_p^{(i)}$ and $\hat{r}_{\bar{m}-p}$ for the two clusters. Note that the superscript $^{(i)}$ corresponds to the two eigenspectra described above. The residual polynomial functions are defined as

$$\hat{r}_p^{(i)}(x) \begin{cases} C_p\left(\frac{b+a-2x}{b-a}\right) / C_p\left(\frac{b+a}{b-a}\right) & \text{, if } i = 1 \\ \prod_{i=1}^{p} (1 - x/\lambda_i) & \text{, if } i = 2, p = N_{\text{tail}} \end{cases} \tag{5.4}$$

and

$$\hat{r}_{\bar{m}-p}(x) = C_{m-p}\left(\frac{d+c-2x}{d-c}\right) / C_{\bar{m}-p}\left(\frac{d+c}{d-c}\right), \tag{5.5}$$

Indeed, the product $r_{\bar{m}} = \hat{r}_p \hat{r}_{\bar{m}-p} \in \mathcal{P}_{\bar{m}}$. Hence, we can use the residual polynomial functions to bound the error at the m^{th} iterate. Now, we obtain the following intermediate bounds

$$\max_{\lambda \in [a,b]} \|r_{\bar{m}}(\lambda)\| \leq \max_{\lambda \in [a,b]} \|\hat{r}_p^{(i)}(\lambda)\| \max_{\lambda \in [a,b]} \|\hat{r}_{\bar{m}-p}(\lambda)\| \leq \max_{\lambda \in [a,b]} \|\hat{r}_p^{(i)}(\lambda)\|, \text{ and} \tag{5.6a}$$

$$\max_{\lambda \in [c,d]} \|r_{\bar{m}}(\lambda)\| \leq \max_{\lambda \in [c,d]} \|\hat{r}_p^{(i)}(\lambda)\| \max_{\lambda \in [c,d]} \|\hat{r}_{\bar{m}-p}(\lambda)\| \leq \max_{\lambda \in [c,d]} \|\hat{r}_p(\lambda)\| / C_{\bar{m}-p}\left(\frac{d+c}{d-c}\right) \tag{5.6b}$$

where the first result follows from the fact that $\|\hat{r}_{m-p}(x)\| < 1 \; \forall x \in [a, b]$ and the second result from

$$\|C_{m-p}\left(\frac{d+c-2x}{d-c}\right)\| < 1 \; \forall x \in [c, d].$$

Furthermore, we have, using the well-known inequality

$$1/C_k\left(\frac{z_1+z_2}{z_1-z_2}\right) \leq 2\left(\frac{\sqrt{z_2}-\sqrt{z_1}}{\sqrt{z_2}+\sqrt{z_1}}\right)^k, \text{ for } z_1 > z_2 > 0 \text{ and } k \in \mathbb{N}^+, \tag{5.7}$$

that

$$\max_{\lambda\in[a,b]} \|\hat{r}_p^{(i)}(\lambda)\| \leq \begin{cases} 2\left(\dfrac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}\right)^p = \eta_1 & \text{, if } i=1, \\ \left(\dfrac{b}{a}-1\right)^p = \eta_2 & \text{, if } i=2, p=N_{\text{tail}}, \end{cases}$$

Note that if $i=1$ we can determine p by requiring that the maximum of the residual polynomial function $\hat{r}_p^{(i)}$ in $[a,b]$ is equal to ϵ. This gives the following equation

$$p = \left\lceil \frac{1}{2}\sqrt{\frac{b}{a}}\ln\epsilon + 1 \right\rceil \tag{5.8}$$

Also note that for $i=2$ $\hat{r}_p^{(2)}(\lambda)=0$ for all eigenvalues $\lambda\in[a,b]$ and thereby, bounded by ϵ.

Next, $\hat{r}_p^{(i)}$ in $[c,d]$ is bounded by its maximum value within $[a,b]$ multiplied by the polynomial that is the fastest growing polynomial outside- and bounded below 1 within $[a,b]$. This polynomial is again the (transformed) Chebyshev polynomial $C_p\left(\dfrac{2x-b-a}{b-a}\right)$. Therefore,

$$\max_{\lambda\in[c,d]} \|\hat{r}_p^{(i)}(\lambda)\| \leq \eta_i C_p\left(\frac{2d-b-a}{b+a}\right)$$

At this point we have ensured equation (5.6a) is bounded by ϵ. So it remains to bound equation (5.6b). Using above results we can write

$$\max_{\lambda\in[c,d]} \|r_{\bar{m}}(\lambda)\| < \epsilon,$$

if we require that

$$\eta_i C_p\left(\frac{2d-b-a}{b-a}\right)/C_{\bar{m}-p}\left(\frac{d+c}{d-c}\right) < \epsilon. \tag{5.9}$$

Using that for $x_1,x_2,x_3\in\mathbb{R}^+$ with $x_1>x_3$ and $z=\dfrac{x_1-x_2}{x_3}$

$$\begin{aligned} C_p(z) &\leq \left(z+\sqrt{z^2-1}\right)^p \\ &= \left(\frac{x_1-x_2}{x_3}+\sqrt{\left[\frac{x_1-x_2}{x_3}\right]^2-1}\right)^p \\ &\leq \left(\frac{x_1}{x_3}+\sqrt{\left[\frac{x_1}{x_3}\right]^2-1}\right)^p \\ &\leq \left(\frac{2x_1}{x_3}\right)^p, \end{aligned}$$

and substituting $x_1=2d$, $x_2=b+a$ and $x_3=b-a$ we obtain the following inequality

$$\eta_i\left(\frac{4d}{b-a}\right)^p/C_{\bar{m}-p}\left(\frac{d+c}{d-c}\right) < \epsilon.$$

Rewriting gives

$$1/C_{\bar{m}-p}\left(\frac{d+c}{d-c}\right) \leq \frac{\epsilon}{\eta_i\left(\frac{4d}{b-a}\right)^p} \leq \frac{\epsilon}{2\left(\frac{4d}{e_i}\right)^p},$$

where

$$e_i = \begin{cases} \sqrt{a}+\sqrt{b} & \text{, if } i=1 \\ a & \text{, if } i=2. \end{cases}$$

Again using equation (5.7) and solving for the degree $\bar{m} - p$ we obtain

$$\bar{m} - p \geq \frac{1}{2}\sqrt{\frac{d}{c}}\left(\ln \epsilon + p \ln \frac{4d}{e_i}\right),$$

which leads to the following bound for the number of iterations

$$\bar{m} = \left\lceil \frac{1}{2}\sqrt{\frac{d}{c}} \ln(2/\epsilon) + \left(1 + \frac{1}{2}\sqrt{\frac{d}{c}} \ln(4d/e_i)\right) p \right\rceil, \tag{5.10}$$

where

$$1 \leq p \leq \min\left\{ \left\lceil \frac{1}{2}\sqrt{\frac{b}{a}} \ln \epsilon + 1 \right\rceil, N_{\text{tail}} \right\}.$$

5.2. Generalization to multiple clusters

At this point we assume that we are dealing with an eigenspectrum of the form $\sigma_1(A)$, i.e. we are only treating case 1. In section 5.3 it is shown that this is indeed a very applicable case for a discretized Darcy problem.

In this case, the technique outlined in section 5.1 starts at the left most cluster $[a, b]$, finds the Chebyshev degree p satisfying inequality 5.8, moves to the neighboring cluster $[c, d]$ and finds the Chebyshev degree $p' = \bar{m} - p$ satisfying inequality 5.9. Rewriting inequality 5.9 gives the following equation for p':

$$\frac{1}{C_{p'}\left(\frac{d+c}{d-c}\right)} \leq \frac{\epsilon}{C_p^{(1)}(d)} = \epsilon', \tag{5.11}$$

where

$$C_p^{(1)}(x) = C_p\left(\frac{b+a-2x}{b-a}\right) / C_p\left(\frac{b+a}{b-a}\right),$$

is the Chebyshev polynomial corresponding to the first cluster.

Suppose there is a third cluster next to $[c, d]$, i.e. $[e, f]$. We can repeat the process and find the Chebyshev degree p'' satisfying a similar inequality as 5.11 for the third cluster.

$$\frac{1}{C_{p''}\left(\frac{f+e}{f-e}\right)} \leq \frac{\epsilon}{C_p^{(1)}(f) C_{p'}^{(2)}(f)} = \epsilon'',$$

This leads to the general equation for the Chebyshev degree p_i of the i^{th} cluster $[a_i, b_i]$

$$\frac{1}{C_{p_i}\left(\frac{b_i+a_i}{b_i-a_i}\right)} \leq \frac{\epsilon}{\prod_{j=1}^{i-1} C_{p_j}^{(j)}(b_i)} = \epsilon^{(i)}. \tag{5.12}$$

Due to the large range of the Chebyshev polynomials $\tilde{C}_p$ a computer is likely to result in floating point number overflow during calculation of the denominator of equation (5.12). Instead, we first apply inequality 5.7 and introduce the cluster condition numbers $\kappa_i = \frac{b_i}{a_i}$, where i is the index of the cluster. We can then rewrite equation (5.12) as follows

$$p_i = \left\lceil \ln \frac{\epsilon^{(i)}}{2} / \ln \frac{\sqrt{\kappa_i} - 1}{\sqrt{\kappa_i} + 1} \right\rceil,$$

and

$$\ln \frac{\epsilon^{(i)}}{2} = \ln \frac{\epsilon}{2} - \sum_{j=1}^{i-1} \ln C_{p_j}^{(j)}(b_i).$$

Let $z_1^{(i,j)} = \dfrac{b_j + a_j - 2b_i}{b_j - a_j}$ and $z_2^{(j)} = \dfrac{b_j + a_j}{b_j - a_j}$ then

$$\ln C_{p_j}^{(j)}(b_i) = \ln C_{p_j}(z_1) - \ln C_{p_j}(z_2).$$

We have, using the definition of the Chebyshev polynomial

$$\ln C_{p_j}(z_1^{(i,j)}) \lessapprox p_j \ln \left[z_1^{(i,j)} - \sqrt{\left(z_1^{(i,j)}\right)^2 - 1} \right] - \ln 2, \tag{5.13}$$

and

$$\ln C_{p_j}(z_2^{(j)}) \gtrapprox p_j \ln \left[z_2^{(j)} + \sqrt{\left(z_2^{(j)}\right)^2 - 1} \right] - \ln 2, \tag{5.14}$$

both of which become more accurate equalities as $z, m \to \infty$. Introducing $\zeta_1^{(i,j)} = z_1^{(i,j)} - \sqrt{\left(z_1^{(i,j)}\right)^2 - 1}$, $\zeta_2^{(j)} = z_2^{(j)} + \sqrt{\left(z_2^{(j)}\right)^2 - 1}$ and $f_i = \dfrac{\sqrt{\kappa_i} - 1}{\sqrt{\kappa_i} + 1}$ with κ_i the i^{th} cluster condition number, and substituting the inequalities 5.13 and 5.14 back into the equation for p_i gives

$$\begin{aligned} p_i &\leq \left\lceil \frac{\ln \frac{\epsilon}{2} - \sum_{j=1}^{i-1} p_j \left\{ \ln \zeta_1^{(i,j)} - \ln \zeta_2^{(j)} \right\}}{\ln f_i} \right\rceil \\ &= \left\lceil \log_{f_i} \frac{\epsilon}{2} - \sum_{j=1}^{i-1} p_j \left\{ \log_{f_i} \zeta_1^{(i,j)} - \log_{f_i} \zeta_2^{(j)} \right\} \right\rceil \\ &= \left\lceil \log_{f_i} \frac{\epsilon}{2} - \sum_{j=1}^{i-1} p_j \log_{f_i} \frac{\zeta_1^{(i,j)}}{\zeta_2^{(j)}} \right\rceil \end{aligned}$$

Note that in general $\zeta_1^{(i,j)} < \zeta_2^{(j)}$ and hence $\log_{f_i} \dfrac{\zeta_2^{(j)}}{\zeta_1^{(i,j)}} > 0$. This prompts us to write

$$p_i \leq \left\lceil \log_{f_i} \frac{\epsilon}{2} + \sum_{j=1}^{i-1} p_j \log_{f_i} \frac{\zeta_2^{(j)}}{\zeta_1^{(i,j)}} \right\rceil \tag{5.15}$$

Evidently, adding more clusters to the left of the interval $[a_i, b_i]$ increases the degree p_i of the Chebyshev polynomial. Next to this, equation (5.15) reduces to the classical CG iteration bound equation (2.11) for a single cluster when $i = N_{\text{clusters}} = 1$.

Equation 5.15 gives us a way to calculate the Chebyshev degree p_i of the i^{th} cluster $[a_i, b_i]$ in terms of the Chebyshev degrees of the previous clusters. To obtain a bound on the number of iterations for the CG method we sum the Chebyshev degrees of all the clusters

$$\bar{m} = \sum_{i=1}^{N_{\text{clusters}}} p_i \tag{5.16}$$

5.3. Numerical experiments

Equations 5.15 and 5.16 give a sequential algorithm for determining an upper bound on the number of iterations for the CG method. Figure 5.1 compares this bound with the classical CG iteration bound equation (2.11). As is the case for figure 2.2, $m_{\text{classical}} = 26$

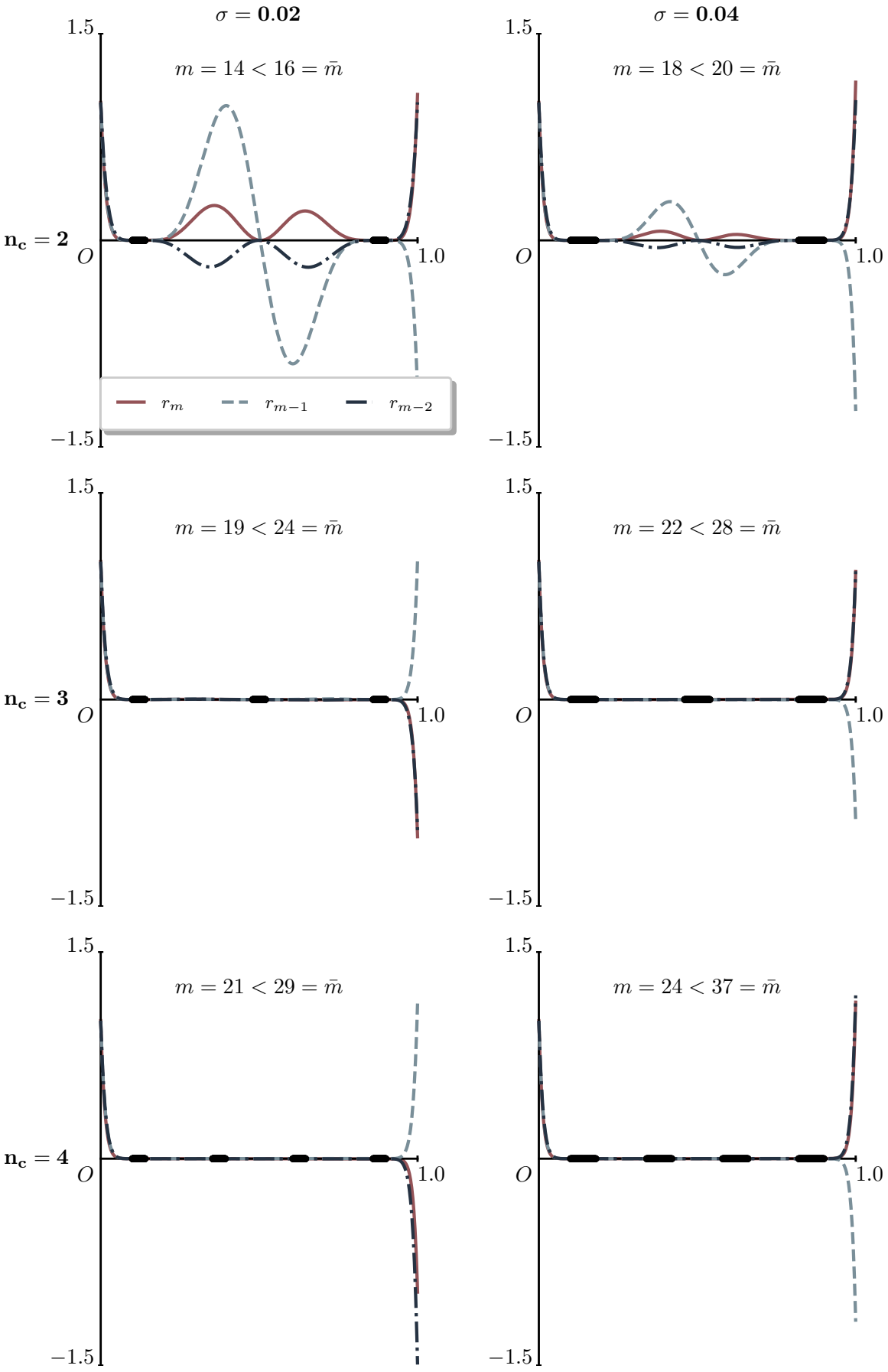


Figure 5.1: Similar to figure 2.2, but with the $\bar{m}$ as determined by equations (5.15) and (5.16).

Figure 5.1 shows that the sharpened CG iteration bound is significantly lower than the classical CG iteration bound for the two cluster case. The performance of the sharpened bound does decrease as the number of clusters increases, as is evident from equations (5.15) and (5.16). Performance also decreases as clusters become wider. So much so, that the sharpened bound is worse than the classical bound for the three cluster case with spread $\sigma = 0.04$ and for the four cluster case.

Worsening performance for the sharpened bound with increased cluster width is expected. Focussing on the two cluster case, we rediscover the ratios $\frac{d}{c}$ in equation (5.10) as well as $\frac{a}{b}$ in the corresponding equation for p. These ratios grow with increasing cluster width.

5.4. Implications for research

The preliminary results discussed in this chapter show that we can find both a priori analytic two-cluster (equation (5.10)) and numerical multiple-cluster (equation (5.16)) sharpened iteration bounds for the CG method. The sharpened bound appears to perform best in the two-cluster case, which corresponds to the eigenspectrum of a typical (preconditioned) Darcy problem. Hence, Q3 is answered positively.

With regard to Q1, the cluster condition number κ_i is introduced as a measure of the cluster width. The sharper the clusters, the smaller the cluster condition number. This is a promising result, as it suggests that we can use the cluster condition number to distinguish between different preconditioners which is a promising result for Q6. However, this is not yet fully explored in this work.

A logical next step is to more rigorously investigate how the sharpened bound depends on k_i as well as the spectral gap (Q5). Subsequently, we can simulate the eigenspectrum of a Darcy problem and compare the sharpened bound with the classical bound (Q4). This will lead to a clear understanding of the performance of the sharpened bound in the main problem context of this thesis: high-contrast, heterogeneous elliptic problems.

Furthermore, we can construct, discretize, and precondition a model Darcy problem with the methods outlined in section 3.2. Then, we both apply the sharpened bound and the CG method on the resulting systems, and investigate how sharp the new bound is (Q6).

The main challenge described in section 4.4 still stands. More work is needed to be able to use the sharpened bound for spectra that are not known or artificially constructed beforehand. The results in this chapter suggest that the cluster condition number is a good candidate for a measure of the eigenspectrum. However, it is not yet clear how to estimate the cluster condition number for a general eigenspectrum. This is an important step towards answering Q2.

6

Conclusion

To do ▶*Summary of key insights – Recap main points.*◀
To do ▶*Implications for future research – What should be studied next?*◀
To do ▶*Final thoughts – Why this research is valuable.*◀

Bibliography

Alves, F. A. C. S., Heinlein, A., & Hajibeygi, H. (2024). A computational study of algebraic coarse spaces for two-level overlapping additive schwarz preconditioners. URL.

Axelsson, O. (1976). A class of iterative methods for finite element equations. *Computer Methods in Applied Mechanics and Engineering*, *9*(2), 123–137. https://doi.org/https://doi.org/10.1016/0045-7825(76)90056-6

Dohrmann, C. R., Klawonn, A., & Widlund, O. B. (2008). A family of energy minimizing coarse spaces for overlapping schwarz preconditioners. *Domain Decomposition Methods in Science and Engineering XVII*, *60*, 247–254.

Dolean, V., Jolivet, P., & Nataf, F. (2015). *An introduction to domain decomposition methods*. Society for Industrial; Applied Mathematics. https://doi.org/10.1137/1.9781611974065

Efendiev, Y., Galvis, J., & Wu, X.-H. (2011). Multiscale finite element methods for high-contrast problems using local spectral basis functions. *Journal of Computational Physics*, *230*(4), 937–955. https://doi.org/https://doi.org/10.1016/j.jcp.2010.09.026

Graham, I. G., Lechner, P. O., & Scheichl, R. (2007). Domain decomposition for multiscale pdes. *Numerische Mathematik*, *106*, 589–626. https://doi.org/10.1007/s00211-007-0074-1

Kelley, C. T. (1995). *Iterative methods for linear and nonlinear equations*. Society for Industrial; Applied Mathematics. https://doi.org/10.1137/1.9781611970944

Meurant, G., & Strakoš, Z. (2006). The lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, *15*, 471–542. https://doi.org/10.1017/S096249290626001X

Rheinbach, O. (2018). Multiscale coarse spaces for overlapping schwarz methods based on the acms space in 2d (L. R. (Ronny Ramlau, Ed.) [Online available: https://epub.oeaw.ac.at/?arp=0x0038c0cb - Last access:7.2.2025], 156–182. URL.

Saad, Y. (2003). *Iterative methods for sparse linear systems* (Second). Society for Industrial; Applied Mathematics. https://doi.org/10.1137/1.9780898718003

Strakoš, Z. (1991). On the real convergence rate of the conjugate gradient method. *Linear Algebra and its Applications*, *154-156*, 535–549. https://doi.org/https://doi.org/10.1016/0024-3795(91)90393-B