

ROBUST ITERATIVE METHODS ON
UNSTRUCTURED MESHES

by

MARIAN BREZINA

M.S., Charles University, Prague, 1990

A thesis submitted to the
University of Colorado at Denver
in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Applied Mathematics

1997

This thesis for the Doctor of Philosophy

degree by

Marian Brezina

has been approved

by

Jan Mandel

Charbel Farhat

Thomas A. Manteuffel

John W. Ruge

Thomas F. Russell

Date _____

Brezina, Marian (Ph.D., Applied Mathematics)

Robust Iterative Methods on Unstructured Meshes

Thesis directed by Professor Jan Mandel

ABSTRACT

We propose and analyze three multilevel iterative solvers of both domain decomposition and multigrid type. All of these methods are algebraic, allowing almost or fully black-box implementation. Their development was motivated by the need to solve large algebraic systems of equations resulting from finite element discretizations of self-adjoint, second order uniformly elliptic problems on unstructured three-dimensional meshes. Two of the methods discussed perform a simple, but effective domain decomposition as a part of the solving process. This allows for a remarkable adaptivity, where the decomposition is generated depending on the difficulty of the problem without requiring an input of a different decomposition. We focus on achieving robustness features that allow using the new methods as a replacement of direct solvers for solving these systems. The new methods are superior in terms of computational complexity and storage requirements. On serial architectures, the asymptotic computational complexity of these methods for solving 3D problems is shown to be in the range of $O(n^{7/6})$ and $O(n^{49/33})$. The methods all benefit from implementation on modern parallel architectures which can reduce the computational complexity to $O(n^{7/6})$

for all three methods. The theoretical results are accompanied by computational experiments confirming the theoretically predicted convergence properties and suggesting the potential of the methods for solving a much wider variety of problems than those covered by the current theory.

This abstract accurately represents the content of the candidate's thesis.

I recommend its publication.

Signed

Jan Mandel

DEDICATION

Dedicated to Jitka Křížková.

ACKNOWLEDGEMENTS

It is my pleasant duty to acknowledge my gratitude to a number of individuals who have influenced my work. I would like to thank all the faculty and staff at the Center for Computational Mathematics at University of Colorado at Denver for creating a great environment for research and computing. My thanks go to professors Leo Franca, Tom Manteuffel, Steve McCormick and Tom Russell, whose classes I had the honor of enjoying. The joint research meetings with the members of the Department of Space Engineering of University of Colorado at Boulder motivated me to focus on the solution of the real world problems; most notably I thank Charbel Farhat for kindly providing the test data from engineering practice. I thank all my collaborators for fruitful cooperation. I want to express my gratitude to Caroline Heberton for carefully reading draft copies of this thesis and proof-reading them. If the reader finds the text enjoyable to read, it is by her merit. My sincere appreciation is due to my friend Petr Vaněk for great many conversations, both on and off the topic of numerical methods, and for teaching me how overrated cleaning one's apartment can really be. Last but not most I want to thank my advisor, Jan Mandel, for his guidance, generosity and friendship. His reading courses and informal meetings proved to be both an inspiration and fun. It was he and Petr Vaněk, who originally aroused my curiosity and enthusiasm for fully algebraic approach to iterative solvers. I am looking forward to our future cooperation.

CONTENTS

Chapter

1	Introduction and Model Problem	1
1.1	Prologue	1
1.2	Formulation of the Problem and Notation	6
1.3	Direct Solvers	11
1.4	Preconditioned Conjugate Gradient Acceleration	13
2	Overview of Some Domain Decomposition Methods	24
2.1	Abstract Schwarz Methods	24
2.2	Overlapping Schwarz	28
2.3	Non-overlapping Decomposition Methods	30
2.4	Methods Reduced to Interface	30
2.5	The Neumann-Neumann Method	34
2.6	EBE method	39
2.7	BDD as an Algebraic Method	41
2.8	DD as a 2-level Multigrid	50
2.9	Other Domain Decomposition Methods	52
3	Fully Black-box Overlapping Schwarz Method	53
3.1	Overlapping Schwarz Method with a Coarse Space.	54
3.2	Smoothed Aggregation Coarse Space and BOSS	68

3.3	Estimates for Smoothed Aggregation	76
3.4	Practical Issues	83
3.4.1	Generation of Aggregates	84
3.4.2	Nonscalar Problems	86
3.4.3	Computational Complexity	87
4	Nonoverlapping Methods with Inexact Solvers	90
4.1	Inexact Subdomain Solvers	90
4.2	The Inexact Solvers' Properties	93
4.3	Matsokin-Nepomnyaschikh Abstract Theory	95
4.3.1	Abstract Framework and Condition Number Estimate	96
4.3.2	An Application: Abstract Additive Schwarz Methods	98
4.4	Unextended Hybrid Schwarz Algorithm	101
4.4.1	BDD as a Hybrid Schwarz Algorithm	103
4.4.2	A New Look at the Conditioning of BDD	105
4.5	Extended Hybrid Schwarz Algorithm	107
4.5.1	Algorithm on Long Vectors with Exact Components	107
4.5.2	Practical Algorithm on Long Vectors and ACDD	109
4.5.3	Estimate on Long Vectors for Inexact Neumann Solvers	111
4.5.4	Inexact Coarse Space and Harmonic Extensions	113
4.5.5	Computational Complexity	117
5	Two-Level Multigrid Alternative	120
5.1	Alternative For Inexact Solvers	120
5.2	Tentative Prolongator and Standard Two-level Multigrid	121

5.3	Modified Two-level Multigrid and MLS	126
5.4	Practical Issues	135
5.4.1	Generalizations	135
5.4.2	Computational Complexity	136
6	Numerical Experiments	138
6.1	Model Problems	139
6.2	Real World Problems	148
7	Conclusions	153
<u>Appendix</u>		
A	Theoretical Results	156
A.1	Poincaré-Friedrichs Inequality	156
B	Results Used In Computational Experiments	162
B.1	The Stopping Criterion for PCG	162
<u>References</u>		165

FIGURES

Figure		
2.1	Harmonic extension of a corner peak.	35
2.2	Harmonic extension of a peak on the edge.	35
2.3	Harmonic extension of a function constant on one edge.	36
2.4	Harmonic extension of a function linear on one edge.	36
2.5	Harmonic extension of a function with random values on one edge.	37
3.1	Possible assignment of elements to different classes \mathcal{C}_i in 2D.	72
3.2	Possible assignment of elements to different classes \mathcal{C}_i in 3D.	73
4.1	Abstract framework scheme.	96
6.1	The checkerboard coefficient pattern. Dark subdomains corre- spond to values α_1 , the light ones to α_2	145
6.2	The mesh of the automobile wheel (data courtesy of Charbel Farhat, University of Colorado at Boulder).	149
6.3	The mesh of the solid with tetrahedron elements (data courtesy of Charbel Farhat, University of Colorado at Boulder).	151
6.4	The mesh of the turbine with 123,120 elements (data courtesy of John Abel, Cornell University).	152

TABLES

Table

6.1	Comparison of BDD with ACDD(k) for different values of k in case of the checkerboard coefficient pattern.	140
6.2	Comparison of BDD with ACDD(k) for problem with exponentially distributed random coefficients.	142
6.3	Comparison of BDD with ACDD(k) for problem with uniformly distributed random coefficients.	143
6.4	Comparison of BDD with BOSS and MLS for problem with coefficients jumps in a checkerboard pattern formed by 125 subdomains. Coarse spaces of dimensions 2744 and 125 were used for MLS and BOSS. Prolongation smoother of degree 1 was used, and MLS used 2 pre-smoothers, 2 post-smoothers.	144
6.5	Comparison of BDD with BOSS and MLS for problem with uniformly distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 1 was used, and MLS used 2 pre-smoothers, 2 post-smoothers.	145
6.6	Comparison of BDD with BOSS and MLS for problem with uniformly distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 4, and 4 pre-smoothers and 4 post-smoothers were used in MLS.	146

6.7	Comparison of BDD with BOSS and MLS for problem with exponentially distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 1 was used for both BOSS and MLS, and 2 pre-smoothers and 2 post-smoothers in MLS.	147
6.8	Comparison of BDD with BOSS and MLS for problem with exponentially distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 4 was used for BOSS and MLS, and 4 pre- and postsmoothers in MLS. . .	147
6.9	Comparison of BDD with BOSS and MLS for problem with uniformly distributed random coefficients and coarse space of dimension 2,744, degree 1 of prolongator smoother, and 2 pre- and postsmoothers in MLS.	147
6.10	Comparison of BDD with BOSS and MLS for problem with exponentially distributed random coefficients and coarse space of dimension 2,744, Prolonation smoother of degree 1 was used, and 2 pre-smoothers and 2 post-smoothers in MLS.	148
6.11	Comparison of BOSS and MLS for solving the shell problem on a mesh discretizing an automobile wheel with 9,915 nodes and 59,490 degrees of freedom. Prolongator smoother of degree 1 was used, and 4 pre-smoothers and 4 post-smoothers in MLS.	149

6.12 Comparison of BOSS and MLS for solving the 3D elasticity problem with 25,058 nodes and 75,174 degrees of freedom, Prolongator smoother of degree 1 was used, and 2 pre-smoothers, 2 post-smoothers in MLS.	150
6.13 Comparison of BOSS and MLS for solving a shell problem: a propeller with 41,040 nodes and 123,120 degrees of freedom. Prolongator smoother of degree 1 was used, and 4 pre-smoothers, 4 post-smoothers in MLS.	151

“The sole aim of science is the honor of the human mind,
and from this point of view a question about numbers
is as important as a question about the system of the world.”

– C. G. J. Jacobi

1. Introduction and Model Problem

1.1 Prologue

After discretization, many problems of engineering practice reduce to the numerical solution of linear systems. These systems are typically very large, sparse, unstructured and ill-conditioned. The performance of linear solvers for these discretized problems in terms of computational complexity is usually approximated by κn^β , where n is the number of degrees of freedom and κ, β are constants dependent on the particular choice of solution method.

The value of κ is smaller for direct solvers than for iterative ones, which makes them less costly for small to medium size problems. Furthermore, it is typically much less sensitive with respect to the conditioning of the problem. This robustness feature is the chief advantage direct solvers provide. Unfortunately, the exponent β is significantly larger for direct solvers, which makes their application prohibitively expensive for large problems. Further, the lack of structure in the problems makes efficient and versatile implementation of direct solvers on modern vector and parallel computers troublesome. Another problematic issue concerning direct solvers is the amount of storage needed to carry out the necessary computations, as even the most sophisticated direct methods cannot, as a matter of principle, avoid fill-in in computing factorization of a matrix.

The ever increasing demand to analyze very large finite element systems together with the drawbacks of direct solvers suggests considering iterative

solvers as an alternative for solving today's and future problems of practical interest. In order to design an iterative method as a worthy replacement of existing state of the art direct solvers for solution of the large problems, the issue of robustness has to be resolved. In order to achieve robustness features similar to those of direct methods, it is necessary to find ways of easing the dependence of their constant κ on the conditioning of the original problem. A common way to tackle this problem is an application of a conjugate gradient method with a properly chosen preconditioner. For a large class of problems, two- or multilevel methods seem to be the most appropriate choice for a preconditioner because of their convergence and computational complexity properties. It is desirable for commercial solver packages to be able to generate coarse discretizations with the property that smooth error components on a given discretization are well approximated by a consecutive coarser level space. In addition, the process should, if possible, be completely automatic, without dependence on explicit knowledge of the underlying finite element mesh. The solvers should be able to rely on as little information about the problem as possible (in extreme case, only its discrete representation), while being able to exploit any additional information that might be available. Despite the limited input, they must be able to efficiently treat practical problems.

The idea of employing a problem of reduced dimension related to the problem to be solved had been known on both discrete and continuous levels long before the advent of systematic studies of multilevel methods. It appeared in the works on the problems of economic modeling by Leontief in 1930's in a context closely related to the aggregation technique we will use in Section 3.2.

The idea of nested iterations related to multigrid method can be traced back to Southwell in 1940's. On the continuous level, the idea is known to have been employed in solving the problems of reactor physics by Enrico Fermi in early 1940's and later by Morozov [70] and others.

Despite the wealth of the multilevel literature, up until recently the known multilevel methods could not guarantee the above requirements of generality and mesh-independence of the method for general nonuniform meshes. For these reasons, algebraic multilevel iterative techniques are very attractive. The two-level domain decomposition offers one solution, mixing direct solvers at subdomain level with an iterative method on the interface level. The localization of the data in these methods makes them very suitable for modern parallel computers. Another candidate is the class of algebraic multigrid methods. These are usually harder to parallelize, but their convergence properties and possibly optimal computational complexity are very favorable. The construction of these algebraic multigrid (AMG) and domain decomposition (DD) methods has been the focus of interest in the last years, as to which the growing list of references dealing with the topic attests (e.g., the pioneering works leading to AMG of the present day by Brandt, McCormick, Ruge and Stüben [16, 77, 78, 82], and the more recent papers by Vaněk, Mandel and Brezina [87, 86].)

Our goal will be to design and study such algebraic methods. Focusing on the class of problems arising from the finite element discretization of elliptic partial differential equations, we propose several iterative solvers. Although these are efficient methods in their own right, we will view them mainly as preconditioners for the preconditioned conjugate gradient method.

The work is organized in 7 chapters. In the rest of Chapter 1 we introduce a model problem and some notation used throughout this thesis. We also recall the preconditioned conjugate gradient method as a means of efficient solution of large sparse systems. Chapter 2 gives an overview of some existing state of the art domain decomposition methods which bear relevance to the methods we will devise. Here we give comments and examples relevant to our main goal of developing robust methods for problems on unstructured meshes. In Chapter 3 we design an overlapping domain decomposition preconditioner with a new coarse space. We will call the new method BOSS. We prove optimal convergence properties of BOSS under regularity-free assumptions on the problem and very weak assumptions on the finite element discretization mesh. The main advantage of this method is its applicability to problems discretized on a variety of types of finite elements and on unstructured meshes. The method also resolves the setup difficulties usually associated with overlapping domain decomposition methods. The overlapping subdomains are generated inside the solver by simple algebraic means. Both the setup and the iteration itself are purely algebraic. One remarkable quality of the method is its adaptivity; for more difficult problems, we use more extensive overlapping achieved without ever partitioning the domain into subdomains outside the solver. The asymptotic computational complexity of the method implemented on a serial architecture in the optimal case is $O(n^{4/3})$ and $O(n^{49/33})$ in 2D and 3D, respectively. This is far below the cost of a back substitution step of typical direct solvers for sparse matrix problems, which is $O(n^{5/3})$. We also show that implementation on a parallel architecture decreases the computational complexity to as little as $O(n)$ and $O(n^{7/6})$ in 2D and 3D,

respectively. In additive setup, the method is suitable for application of inexact subdomain solvers. Chapter 4 discusses the application of inexact components in substructuring methods. The issues of the influence of approximate subdomain solvers as well as issues of approximate formulation of the method and enhancing the sparsity of the coarse level problem are studied here. Using the resulting method as a preconditioner in the conjugate gradient method, the condition number is proved to be bounded by $C(1 + \log(H/h))^2$, with constant C dependent on the quality of the approximation used. Estimates for C are given. Chapter 5 describes an algebraic multilevel technique using smoothed transfer operators. The resulting method, called MLS, is based on the same concepts as the one of Chapter 3, but no subdomain problems are solved. Instead, a combination of smoothing and coarse space correction reminiscent of the multigrid V -cycle takes place. This leads to a significantly reduced computational complexity of $O(n^{7/6})$ and $O(n^{6/5})$ for a serial implementation in 2D and 3D, respectively, as opposed to $O(n^{4/3})$ and $O(n^{49/33})$ in the case of BOSS. Optimal convergence estimate is also given. Chapter 6 presents the results of numerical experiments. The robustness of these methods with respect to a number of aspects has been tested by solving both artificial and real-world problems. The comparison of the results suggests BOSS to be the most robust of the three methods with respect to a number of influences, including discontinuities in coefficients of arbitrary pattern and complex geometries. Finally, Chapter 7 gives a brief summary and suggests the directions of the future implementation improvements and research.

1.2 Formulation of the Problem and Notation

Let Ω be a bounded domain in \mathbb{R}^d , $d \in \{2, 3\}$, with a piecewise smooth boundary $\partial\Omega$, and $\partial\Omega = \Gamma_D \cup \Gamma_N$ with Γ_D, Γ_N disjoint, $\text{meas}(\Gamma_D) > 0$. We will denote by Γ_D, Γ_N the parts of the boundary with Dirichlet and Neumann boundary conditions, respectively. A coarse triangulation is needed in some of the methods we will discuss. This can be achieved by decomposing Ω into nonoverlapping subdomains Ω_i , $i = 1, \dots, J$,

$$\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2 \cup \dots \cup \bar{\Omega}_J, \quad (1.1)$$

such that Ω_i does not cut through any element. Other forms of decomposition will also be discussed in the text whenever appropriate. Consider the model problem

$$Lu = f \text{ in } \Omega, \quad u = g \text{ on } \Gamma_D, \quad \frac{\partial u}{\partial n} = 0 \text{ on } \Gamma_N, \quad (1.2)$$

where

$$Lv = - \sum_{r,s=1}^d \frac{\partial}{\partial x_r} \left(\alpha(x) \beta_{rs}(x) \frac{\partial v(x)}{\partial x_s} \right), \quad (1.3)$$

with the coefficient matrix (β_{rs}) uniformly positive definite, bounded and piecewise smooth on Ω , smooth on each Ω_i , and $\alpha(x)$ a positive constant in each subdomain Ω_i , i.e.,

$$\alpha(x) = \alpha_i > 0 \text{ for } x \in \Omega_i.$$

Variation in the value of $\alpha(x)$ is allowed across substructure interfaces. It is possible to relax this assumption to the case when $\alpha(x)$ varies moderately within each subdomain. Problem (1.2) can be restated in terms of an equivalent variational Galerkin problem

$$u \in V : \quad a(u, v) = f(v) \quad \forall v \in V, \quad (1.4)$$

where $V = H_{\Gamma_D}^1(\Omega)$ denotes the Sobolev space (cf. [72, 1]) of $H^1(\Omega)$ functions vanishing on $\Gamma_D \subset \partial\Omega$, $\text{meas}(\Gamma_D) > C \text{meas}(\partial\Omega)$.

The finite element discretization reduces problem (1.4) to a linear system of algebraic equations

$$Ax = f, \tag{1.5}$$

where A is the stiffness matrix with entries

$$a_{ij} = a(\phi_j, \phi_i) = \sum_{r,s=1}^d \int_{\Omega} \alpha(x) \beta_{rs}(x) \frac{\partial \phi_i}{\partial x_s} \frac{\partial \phi_j}{\partial x_r} dx,$$

and $f_i = \langle f, \phi_i \rangle$ is the load vector; ϕ_i are the standard finite element functions.

We assume the finite element space to be the usual conforming P1 or Q1 space [23]. We define the subdomain decomposition interface

$$\Gamma = \bigcup_{i=1}^J \partial\Omega_i \setminus \Gamma_D \tag{1.6}$$

and, for simplicity, assume that Γ_D is the union of the closure of whole faces of some or all of the boundary substructures. We associate parameters H, h with the coarse and local mesh, respectively. We assume that the elements and subdomains are shape regular in the usual sense [23].

It is easy to see that under our assumptions the bilinear form $a(\cdot, \cdot)$ satisfies the usual assumptions of symmetry, V-ellipticity and boundedness (cf., [23]).

$$c_1 \|u\|_{H^1(\Omega)}^2 \leq a(u, u) \leq c_2 \|u\|_{H^1(\Omega)}^2 \quad \forall u \in V. \tag{1.7}$$

We note that these restrictions are assumed in order to simplify the analysis of the methods. Some of the methods discussed in the text perform well if some or even all of the aforementioned assumptions are violated. Solving problem (1.4) under these assumptions should be viewed as our minimal

goal; indeed, some of the discussed methods will be applicable to solving much more general problems with weaker assumptions. Whenever appropriate, these assumptions will be relaxed.

Let $\hat{\Omega}$ denote a reference domain of diameter $O(1)$ (e.g., square or cube in 2D or 3D, respectively) and assume that the subdomains Ω_i are of diameter $O(H)$ and shape regular, i.e.,

$$\Omega_i = F_i(\hat{\Omega}), \quad \|\partial F_i\| \leq CH, \quad \|\partial F_i^{-1}\| \leq CH^{-1}, \quad (1.8)$$

with ∂F_i the Jacobian and $\|\cdot\|$ the Euclidean \mathbb{R}^d matrix norm.

Let $V_h(\Omega)$ be a conforming linear finite element space on a quasiuniform triangulation \mathcal{T}_h of Ω with a characteristic meshsize h , such that each subdomain Ω_i is the union of some of the elements, and the usual shape regularity and inverse assumption hold (cf, [23]). All functions $v \in V_h(\Omega)$ satisfy homogeneous boundary condition $u = 0$ on Γ_D .

Let $V_h(\Omega_i)$ be the space of the restrictions of functions in $V_h(\Omega)$ to Ω_i . Throughout this thesis, C and c shall denote (unless specified otherwise) generic positive constants independent of the shape or size of Ω and Ω_i . Note that these constants may depend on the constant in (1.8) or on the regularity of the triangulation, but they are independent of h and H .

Following [11], [28] and [81], we define the scaled Sobolev norms

$$\|u\|_{1,\Omega_i}^2 = |u|_{1,\Omega_i}^2 + \frac{1}{H^2}|u|_{0,\Omega_i}^2, \quad \|u\|_{1/2,\partial\Omega_i}^2 = |u|_{1/2,\partial\Omega_i}^2 + \frac{1}{H}|u|_{0,\partial\Omega_i}^2,$$

where

$$|u|_{1,\Omega_i}^2 = \int_{\Omega_i} |\nabla u(x)|^2 dx, \quad |u|_{1/2,\partial\Omega_i}^2 = \int_{\partial\Omega_i} \int_{\partial\Omega_i} \frac{|u(t) - u(s)|^2}{|t - s|^d} dt ds.$$

The advantage of this definition is that it allows us to restrict all of our considerations to the reference domain $\hat{\Omega}$ and use the mappings F_i to obtain the results for each Ω_i from the obvious norm equivalence

$$\begin{aligned} c \|u\|_{1,\Omega_i}^2 &\leq \|u \circ F_i\|_{1,\hat{\Omega}}^2 H^{d-2} \leq C \|u\|_{1,\Omega_i}^2, \\ c \|u\|_{1/2,\partial\Omega_i}^2 &\leq \|u \circ F_i\|_{1/2,\partial\hat{\Omega}}^2 H^{d-2} \leq C \|u\|_{1/2,\partial\Omega_i}^2. \end{aligned} \quad (1.9)$$

Assume that for each Ω_i , $\Gamma_D \cap \partial\Omega_i$ is either empty or a part of $\partial\Omega_i$ of size bounded below by a fixed proportion of the size of $\partial\Omega_i$ so that the Poincaré inequality (Theorem A.1.4) holds uniformly for all Ω_i , with the constant C independent of h and H ,

$$|u|_{0,\Omega_i} \leq CH |u|_{1,\Omega_i}, \quad |u|_{0,\partial\Omega_i} \leq CH^{1/2} |u|_{1/2,\partial\Omega_i} \quad (1.10)$$

for all $u \in V_h(\Omega_i)$ if $\Gamma_D \cap \partial\Omega_i \neq \emptyset$ and for all $u \in V_h(\Omega_i)$, $\int_{\partial\Omega_i} u \, ds = 0$ if $\Gamma_D \cap \partial\Omega_i = \emptyset$.

Throughout this thesis, we will adopt the following notation: We denote by $\sigma(A)$ the set of all eigenvalues of A , by $\varrho(A)$ the spectral radius of matrix A : $\varrho(A) = \max\{|\lambda(A)|\}$. We shall use $\langle \cdot, \cdot \rangle$, $\|\cdot\|$ to denote the Euclidean inner product and Euclidean norm, respectively. If B is a symmetric positive (semi)definite matrix, we also use the (semi)norm derived from this matrix, i.e. $\|x\|_B = \langle x, x \rangle_B^{1/2} = \langle Bx, x \rangle^{1/2}$.

For a matrix A positive definite and symmetric in B -inner product, the symbol $\text{cond}_B(A)$ will denote the condition number, $\text{cond}_B(A) = \min \frac{\Lambda_M}{\Lambda_m}$, where minimum is taken over all positive Λ_m, Λ_M such that $\Lambda_m \leq \frac{\langle Ax, x \rangle_B}{\langle \mathcal{M}x, x \rangle_B} \leq \Lambda_M \quad \forall x \in \mathbb{R}^n$. For $A = A^T$ we obtain the usual spectral condition number $\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$, where λ_{\max} and λ_{\min} denote the largest and

smallest eigenvalues of A , respectively. Note that for linear systems with matrix A resulting from finite element discretization of second order partial differential equations, the condition number is proportional to $\frac{1}{h^2}$ (cf., Lemma A.1.7). The definition can also be extended to a product of two B -symmetric positive definite operators, i.e., the mutual condition number of a preconditioned system with preconditioner \mathcal{M} , $\text{cond}_B(\mathcal{M}, A)$. In this case, finding positive constants Λ_m, Λ_M such that

$$\Lambda_m \leq \frac{\langle Ax, x \rangle_B}{\langle \mathcal{M}x, x \rangle_B} \leq \Lambda_M \quad \forall x \in \mathbb{R}^n \quad (1.11)$$

provides an estimate

$$\text{cond}_B(M, A) \leq \frac{\Lambda_M}{\Lambda_m},$$

and the mutual condition number is defined as the minimum of $\frac{\Lambda_M}{\Lambda_m}$ over all Λ_m, Λ_M satisfying (1.11). Adopting this definition, we will denote by abuse of notation $\text{cond}_B(\mathcal{M}^{-1}A) = \text{cond}_B(\mathcal{M}, A)$. This follows from the identity $\sigma(\mathcal{M}^{-1}A) = \sigma(M^{-\frac{1}{2}}AM^{-\frac{1}{2}})$ and from the fact that Rayleigh quotients $\frac{\langle Ax, x \rangle_B}{\langle \mathcal{M}x, x \rangle_B}$ and $\frac{\langle M^{-\frac{1}{2}}AM^{-\frac{1}{2}}y, y \rangle_B}{\langle y, y \rangle_B}$ have the same extrema.

Symbol $A^{(i)}$ will denote the local stiffness matrix corresponding to subdomain Ω_i , $x^{(i)}$ the vector of degrees of freedom corresponding to all the elements in Ω_i and N_i the matrix with entries 0 or 1 mapping the degrees of freedom $x^{(i)}$ into global degrees of freedom, i.e., $x^{(i)} = N_i^T x$.

Each $x^{(i)}$ may be split into degrees of freedom $\bar{x}^{(i)}$ corresponding to $\partial\Omega_i \cap \Gamma$ called interface degrees of freedom and the remaining interior degrees of freedom $\hat{x}^{(i)}$ (thus the degrees of freedom on $\partial\Omega \cap \Gamma_N$ are considered interior degrees of freedom.) We will often find it useful to split local subdomain matrices

$A^{(i)}$ and the matrices N_i accordingly:

$$x^{(i)} = \begin{bmatrix} \bar{x}^{(i)} \\ \dot{x}^{(i)} \end{bmatrix}, \quad A^{(i)} = \begin{bmatrix} A_{11}^{(i)} & A_{12}^{(i)} \\ A_{12}^{(i)T} & A_{22}^{(i)} \end{bmatrix}, \quad N_i = [\bar{N}_i, \dot{N}_i], \quad (1.12)$$

where $A_{11}^{(i)}, A_{22}^{(i)}$ are the blocks corresponding to the interface and interior degrees of freedom, respectively.

Similarly, let \bar{N} and \dot{N} denote the matrices with entries 0 or 1 mapping the degrees of freedom \bar{x} on Γ into global degrees of freedom, i.e., $\bar{x} = \bar{N}^T x$ and between the degrees of freedom in $\Omega \setminus \Gamma$ and the global degrees of freedom, respectively.

1.3 Direct Solvers

In this section, we briefly remark on some direct methods of solution of linear algebraic systems commonly used in practice. Although the iterative methods had actively been used in finite element computations well into 1960's, the availability of new hardware with larger storage made it possible to use direct solvers. For the last three decades, the majority of commercial finite element codes have relied on direct methods to solve linear systems of equations. Most of today's direct solvers proceed by first factoring the matrix A (e.g., by Cholesky or Crout factorization) and then using back-substitution. The simplicity and robustness of the direct methods have gained them acceptance in the two-dimensional finite element analysis. However, their application to solving practical three-dimensional problems of even relatively modest sizes poses a challenge for the in-core storage capacity and execution capabilities of current

high-end computers. For the sake of argument let us consider two model problems: a square consisting of $n \times n$ Q1 elements and a cube of $n \times n \times n$ Q1 elements. In order to solve a simple diffusion problem on our 2D model mesh, we will need to perform $O(n^4)$ floating point operations to obtain the factorization and the storage required to carry out the computation will be $O(n^3)$ words. The demands grow severely higher for the 3D model mesh, where the requirements are $O(n^7)$ and $O(n^5)$ for the asymptotic cost of factorization and storage, respectively. The asymptotic cost of back-substitution is negligible compared to the cost of the factorization, as it is $O(n^3)$ and $O(n^5)$ for the 2D and 3D model mesh, respectively. There are direct methods requiring less work than the profile method discussed above. Optimal in terms of computational complexity is the method of nested dissection [41] requiring $O(n^3)$ and $O(n^6)$ operations for our example in 2D and 3D, respectively. Although this is less than in the case of the lexicographically ordered Cholesky factorization, it is still too expensive.

Of course, our model meshes are among those least suitable for direct solvers. Indeed, there exist 3D meshes for which application of direct solvers is more than appropriate. For instance, a single strip of 3D elements may be quite easily solved by a direct solver if the degrees of freedom are properly (re)numbered. It is intuitively clear that efficiency of direct solvers depends on the connectivity of the mesh. In order to evaluate suitability of application of direct solvers, Hughes, Ferencz and Hallquist developed the concept of fractal dimension allowing to measure mesh connectivity (cf. [49] for details). The larger the fractal dimension of the mesh, the more beneficial application of iterative methods may be. An alternative measure allowing insight into the role

of mesh connectivity and its influence on the performance of direct solvers is the Euclidean norm of the lengths of the skyline profiles.

In this thesis we focus on iterative methods as a superior replacement of direct solvers for problems on large unstructured 3D meshes with high fractal dimension, where their superiority over direct solvers can be most easily achieved. But it is the experience of the author that many of these methods prove to be worthy rivals of direct solvers even for problems with rather small fractal dimensions and in 2D. For a comprehensive study of a variety of direct methods of solution, appropriate data structures and other related topic we refer the reader to [42].

1.4 Preconditioned Conjugate Gradient Acceleration

This section provides an overview of the conjugate gradient method introduced by Hestenes and Stiefel [46]. Although a well-known method, we find the PCG method to be of such a paramount importance that we will describe it in detail. The treatment here follows mostly the exposition of Ashby, Manteuffel and Saylor [3], allowing more generality than traditional references. Also, it is the opinion of the author that this point of view is more transparent than alternative ones. Other useful discussions appear, among others, in Luenberger [60], Concus, Golub and O’Leary [24] and Golub, Van Loan [44].

Let $A : V \rightarrow V$ be a SPD operator with respect to the inner product $\langle \cdot, \cdot \rangle$. We need to solve the equation of the following type:

$$Au = b. \tag{1.13}$$

Let $n = \dim(V)$. The conjugate gradient method belongs to the family of

polynomial iterative methods of the form

$$u_{i+1} = u_i + \sum_{j=0}^i \eta_{ij} r_j,$$

where r_i denotes the residual in step i , $r_i = b - Au_i$. From here, denoting by u^* the exact solution, we easily obtain the equation for the error $e_i = u^* - u_i$,

$$e_{i+1} = R_{i+1}(A)e_0,$$

where R_{i+1} denotes the residual polynomial ($R_{i+1}(0) = 1$) of degree less or equal to $i + 1$. Because $R_{i+1}(0) = 1$, we have

$$\begin{aligned} u_{i+1} &= u^* - R_{i+1}(A)e_0 \\ &= u_0 + e_0 - R_{i+1}(A)e_0 \\ &= u_0 + (I - R_{i+1}(A))e_0 \\ &= u_0 + Q_i(A)Ae_0 \\ &= u_0 + Q_i(A)r_0. \end{aligned}$$

Let us denote by $K_i(r_0, A)$ the Krylov subspace of dimension at most i generated by r_0 and A . That is,

$$K_i = \text{span}\{r_0, Ar_0, \dots, A^{i-1}r_0\}.$$

Thus we can rewrite the last equation as

$$u_{i+1} = u_0 + \tilde{d}_i, \quad \tilde{d}_i \in K_{i+1}(r_0, A) \tag{1.14}$$

or equivalently

$$u_{i+1} = u_i + d_i, \quad d_i \in K_{i+1}(r_0, A) \tag{1.15}$$

Equations (1.14), (1.15) define a general gradient method.

Let B be a Hermitian positive definite matrix. We define CG(B,A) to be the gradient method that chooses

$$d_k \in K_{k+1}(r_0, A)$$

so that the norm $\|e_{k+1}\|_B$ is minimized over $K_{k+1}(r_0, A)$:

$$\|e_{k+1}\|_B \longrightarrow \min \text{ over } K_{k+1}(r_0, A). \quad (1.16)$$

This is the optimality property of the method. Different choices of matrix B , result in different methods. Recall that $K_k \subset K_{k+1}$. Thus for all $w \in K_k$ we have

$$\begin{aligned} 0 &= \langle Be_{k+1}, w \rangle \\ &= \langle B(e_k - d_k), w \rangle \\ &= \langle Be_k, w \rangle - \langle Bd_k, w \rangle \\ &= -\langle Bd_k, w \rangle, \end{aligned}$$

as $\langle Be_k, w \rangle = 0$ from previous step.

Therefore,

$$\langle Bd_k, w \rangle = 0 \quad \forall w \in K_k(r_0, A).$$

In other words,

$$d_k \in K_{k+1}(r_0, A) \quad \text{and} \quad d_k \perp_B K_k(r_0, A).$$

This defines d_k up to a scalar. Thus, we can let

$$d_k = \alpha_k p_k,$$

where $p_k \in K_{k+1}$ is some conveniently chosen vector and scalar α_k is yet to be determined.

To find d_k we will construct a B -orthogonal basis $\{p_j\}_{j=0}^k$ for K_{k+1} . We can use a generalization of the Gram-Schmidt process to the B inner product to do this:

$$p_0 = r_0$$

$$p_{i+1} = Ap_i - \sum_{j=0}^i \sigma_{ij} p_j, \quad \sigma_{ij} = \frac{\langle BAp_i, p_j \rangle}{\langle Bp_j, p_j \rangle}$$

By construction,

$$p_j \in K_{j+1} \text{ and } p_j \perp_B K_j$$

and so

$$d_j = \alpha_j p_j$$

for some α_j . We can once again use the orthogonality $e_{j+1} \perp_B K_{j+1}$ to determine α_j :

$$\begin{aligned} 0 &= \langle Be_{j+1}, p_j \rangle \\ &= \langle B(e_j - \alpha_j p_j), p_j \rangle \\ &= \langle Be_j, p_j \rangle - \alpha_j \langle Bp_j, p_j \rangle, \end{aligned}$$

and therefore

$$\alpha_j = \frac{\langle Be_j, p_j \rangle}{\langle Bp_j, p_j \rangle}.$$

This value is always well-defined because B is assumed Hermitian positive definite. Note, however, that this value is expressed in terms of the unknown error e_j , which raises the question of its computability. In order to compute α_i , we will need additional assumptions on B . For our purposes, it will suffice to assume

that $B = \tilde{p}(A)$, where $\tilde{p}(x)$ is an arbitrary polynomial with zero absolute term, i.e., $\tilde{p}(0) = 0$.

Thus, in practice, vector d_j satisfying the optimality property (1.16) is obtained by imposing the equivalent B -orthogonality condition

$$\langle Be_{j+1}, z \rangle = 0 \quad \forall z \in K_{j+1}(r_0, A). \quad (1.17)$$

The above discussion defines the conjugate gradient method CG(B,A). Different implementations of the method exist. These different algorithms are mathematically equivalent for a symmetric and positive definite matrix A . Also, by different choice of matrix B , we recover various known methods (setting $B = A$, for instance, yields the original method of Hestenes and Stiefel). For a systematic study we refer to Ashby, Manteuffel and Saylor [3]. As we will mostly be dealing with symmetric positive definite problems, we only recall here the so-called Orthomin implementation because of its superior computational complexity and storage requirements.

Algorithm 1 (Orthomin). For a given initial guess $u_0 \in \mathbb{R}^n$, set $r_0 = b - Au_0$, $\hat{p}_0 = r_0$, and

repeat

1. $\hat{\alpha}_i = \frac{\langle Be_i, \hat{p}_i \rangle}{\langle B\hat{p}_i, \hat{p}_i \rangle}$,
2. $u_{i+1} = u_i + \hat{\alpha}_i \hat{p}_i$,
3. $r_{i+1} = r_i - \hat{\alpha}_i A\hat{p}_i$,
4. $\beta_{ij} = \frac{\langle BAe_{i+1}, \hat{p}_j \rangle}{\langle B\hat{p}_j, \hat{p}_j \rangle}$,
5. $\hat{p}_{i+1} = r_{i+1} - \sum_{j=0}^i \beta_{ij} \hat{p}_j$,

until convergence;

This implementation of conjugate gradient method will work for any matrix A provided all the coefficients $\alpha_i \neq 0$. Algorithm 1 in general requires storing all previous direction vectors p_j in order to compute β_{ij} . Under our assumption that A be symmetric positive definite, however, the recursion in Step 4. naturally truncates so that only the last direction vector needs to be stored. The necessary and sufficient conditions of existence of this and other economical recursions in conjugate gradient method were studied in Faber and Manteuffel [35].

The advantage of looking at the conjugate gradient method in the way described above is that it simplifies the analysis of the error reduction. Note that owing to (1.15) and (1.17) we may view each step of the conjugate gradient method as eliminating the B -orthogonal Galerkin projection of the unknown error onto the current Krylov space given by r_0 and A .

Since $K_n(r_0, A) = V$ and $e_n \perp_B K_n(r_0, A)$, we have $e_n = 0$, assuming infinitely precise arithmetic. In other words, $\text{CG}(B, A)$ converges in at most n steps. This is how the conjugate gradient method was originally perceived. Computational experiments have shown that the round-off error can cause loss of orthogonality in practical applications. That is why today the method is primarily viewed as an iterative rather than direct one, as first suggested by Reid [75].

It is often useful to replace the problem $Au = b$ by a related problem $\mathcal{M}^{-1}Au = \mathcal{M}^{-1}b$, where \mathcal{M} is a symmetric positive definite matrix. Matrix \mathcal{M} is called a preconditioner. Matrix $\mathcal{M}^{-1}A$ is no longer symmetric in the $\langle \cdot, \cdot \rangle$ inner product, but we note that it is symmetric with respect to the energetic inner

product $\langle A \cdot, \cdot \rangle$ and with respect to inner product $\langle \mathcal{M} \cdot, \cdot \rangle$. Therefore, we can now apply the conjugate gradient method to the matrix $\tilde{A} = \mathcal{M}^{-1}A$ and $\tilde{b} = \mathcal{M}^{-1}b$. By doing so, we arrive at the preconditioned conjugate gradient method (PCG). The following is the Orthomin implementation of PCG:

Algorithm 2 (Orthomin implementation of PCG). For the given initial guess $u_0 \in \mathbb{R}^n$, set $r_0 = b - Au_0$, $\mathcal{M}\hat{p}_0 = r_0$,

repeat

1. $\hat{\alpha}_i = \frac{\langle Be_i, \hat{p}_i \rangle}{\langle B\hat{p}_i, \hat{p}_i \rangle}$,
2. $u_{i+1} = u_i + \hat{\alpha}_i \hat{p}_i$,
3. $r_{i+1} = r_i - \hat{\alpha}_i A \hat{p}_i$,
4. $\mathcal{M}z_{i+1} = r_{i+1}$,
5. $\beta_i = \frac{\langle Bz_{i+1}, \hat{p}_i \rangle}{\langle B\hat{p}_i, \hat{p}_i \rangle}$,
6. $\hat{p}_{i+1} = z_{i+1} - \beta_i \hat{p}_i$,

until convergence;

Note that the computational complexity of this algorithm differs from the Orthomin algorithm without preconditioning by the cost of the additional preconditioning step 4. The following theorem offers a motivation for employing a preconditioner. For simplicity, we will restrict the proof to the case of the preconditioned conjugate gradient method of Concus, Golub and O'Leary [24], i.e., we set $B = A$.

Theorem 1.4.1. For error in step k of the conjugate gradient method applied to system $\mathcal{M}^{-1}Au = \mathcal{M}^{-1}b$, it holds that

$$\|u - u_k\|_A \leq 2 \left(\frac{\sqrt{\text{cond}(\mathcal{M}^{-1}A)} - 1}{\sqrt{\text{cond}(\mathcal{M}^{-1}A)} + 1} \right)^k \|u - u_0\|_A. \quad (1.18)$$

Proof. Let us set $r_0 = \mathcal{M}^{-1}(b - Au_0)$. As u_k is a A -orthogonal projection, we have

$$\langle A(u - u_k), u - u_k \rangle \leq \langle A(u - v), u - v \rangle \quad \forall v \in u_0 + K_k(r_0, \mathcal{M}^{-1}A).$$

Now for an arbitrary polynomial p_{k-1} of degree $k - 1$, taking

$$v = u_0 + p_{k-1}(\mathcal{M}^{-1}A)r_0 = u_0 + \mathcal{M}^{-1}Ap_{k-1}(\mathcal{M}^{-1}A)(u - u_0),$$

we have

$$\begin{aligned} & \langle A(u - u_k), u - u_k \rangle \\ & \leq \min_{p_{k-1}} \langle A(I - \mathcal{M}^{-1}Ap_{k-1}(\mathcal{M}^{-1}A))(u - u_0), (I - \mathcal{M}^{-1}Ap_{k-1}(\mathcal{M}^{-1}A))(u - u_0) \rangle \\ & \leq \min_{q_k(0)=1} \langle Aq_k(\mathcal{M}^{-1}A)(u - u_0), q_k(\mathcal{M}^{-1}A)(u - u_0) \rangle \\ & \leq \min_{q_k(0)=1} \max_{\lambda \in \sigma(\mathcal{M}^{-1}A)} |q_k(\lambda)|^2 \langle A(u - u_0), u - u_0 \rangle. \end{aligned}$$

It is a well-known result of approximation theory that the solution of this minimax problem can be found in terms of the Chebyshev polynomials

$$T_k(x) = \frac{1}{2} \left((x + \sqrt{x^2 - 1})^k + (x - \sqrt{x^2 - 1})^k \right)$$

scaled to the interval of the spectrum $[\lambda_{\min}(\mathcal{M}^{-1}A), \lambda_{\max}(\mathcal{M}^{-1}A)]$. We can then deduce that

$$\min_{q_k(0)=1} \max_{\lambda \in \sigma(\mathcal{M}^{-1}A)} |q_k(\lambda)| \leq 2 \left(\frac{\sqrt{\text{cond}(\mathcal{M}^{-1}A) - 1}}{\sqrt{\text{cond}(\mathcal{M}^{-1}A) + 1}} \right)^k,$$

from where (1.18) follows. \square

By setting $B = A$, $\mathcal{M} = I$, we recover the error estimate for the original conjugate gradients method of Hestenes and Stiefel.

We notice that this estimate for the convergence rate of the conjugate gradient method depends strongly on the condition number. Namely, the number of steps required to reduce the error by a given factor is proportional to $\sqrt{\text{cond}(\mathcal{M}, A)}$. Even though the estimate (1.18) often turns out to be rather pessimistic (e.g., if the matrix A has clustered eigenvalues), we can see that the selection of the preconditioner \mathcal{M} is crucial to the performance of the method. Theorem 1.4.1 thus gives an answer to two questions: It shows that the additional cost of preconditioning can be justified and it clarifies the sense in which the preconditioner \mathcal{M} should be related to A if good convergence is to be guaranteed.

Based on these facts, we will seek preconditioners such that the problem $\mathcal{M}z = r$ is more easily obtained than that of $Au = b$. This will guarantee that the cost of applying each iterative step of (PCG) will be small. We will want \mathcal{M} to approximate spectral properties of A in the sense that $\text{cond}(\mathcal{M}^{-1}A)$ is small. This will guarantee that the number of steps necessary to reduce the error to a required size will be small. The quantity $\text{cond}(\mathcal{M}, A) = \text{cond}(\mathcal{M}^{-1}A)$ will serve as a measure of quality of a preconditioner.

We have shown that preconditioning can greatly improve the convergence properties of the conjugate gradient method. Our work will consist in finding suitable preconditioners. Sometimes these preconditioners are explicitly available in the matrix form, but most often, we will find it convenient to construct iterative methods to compute approximation of $A^{-1}r_k$ needed in step 4. of Algorithm 2. These iterative methods, if used as a preconditioner, improve

the convergence rate of PCG. We can also adopt a dual understanding of preconditioning with respect to the linear symmetric iterative methods, namely that PCG is often viewed as a means of acceleration of these methods. In order to justify this claim, let us consider a linear iterative solver in the form

$$x_{k+1} = \widehat{M}x_k + \widehat{N}b \quad (1.19)$$

consistent with the problem $Au = b$, i.e., having the same solution as $Au = b$. Here \widehat{N} is assumed symmetric, hence \widehat{M} is A -symmetric, because it follows from the consistency and the fact that A is SPD that $A^{-1}b = \widehat{M}A^{-1}b + \widehat{N}b \quad \forall b$, so $\widehat{M} = I - \widehat{N}A$ and

$$\langle A\widehat{M}x, y \rangle = \langle A(I - \widehat{N}A)x, y \rangle = \langle Ax, (I - \widehat{N}A)y \rangle = \langle Ax, \widehat{M}y \rangle.$$

Note that all the classical splitting type iterative schemes such as Jacobi, as well as the variational multigrid methods may be written in this form (see Section 4.2 for more details). The consistency condition allows us to write alternatively

$$x_{k+1} = x_k + \widehat{N}(b - Ax_k),$$

hence the method may be viewed as preconditioned Richardson iteration with preconditioner $\mathcal{M}^{-1} = \widehat{N}$.

This process converges provided that

$$\|\widehat{M}\|_A = q < 1.$$

This means $q = \|I - \widehat{N}A\|_A < 1$. Since we can easily verify that

$$\text{cond}(\widehat{N}A) \leq \frac{1+q}{1-q},$$

we can see that application of preconditioned conjugate gradient method with preconditioner \widehat{N} (in other words, the preconditioner given by one iteration of the process (1.19) with $x_0 = 0$) yields a superior convergence factor, as

$$\frac{\sqrt{\text{cond}(\widehat{N}A)} - 1}{\sqrt{\text{cond}(\widehat{N}A)} + 1} \leq \frac{\sqrt{\frac{1+q}{1-q}} - 1}{\sqrt{\frac{1+q}{1-q}} + 1} \leq \frac{1 - \sqrt{1 - q^2}}{q} < q \quad \forall q \in (0, 1). \quad (1.20)$$

Hence for a symmetric linear iterative scheme, a preconditioner for A can be found so that the convergence rate of the scheme is accelerated by using the PCG with that preconditioner. This concludes our brief discussion of the preconditioned conjugate gradient method.

2. Overview of Some Domain Decomposition Methods

In this chapter, we will give an overview of some known domain decomposition methods. Because the domain decomposition field has grown very large since the first studies, this overview cannot be exhaustive. Instead, we list a few methods we will draw on in the derivation of our new methods. Some methods are presented with the intention to contrast the differences with our approach. We also attempt to direct the reader to the appropriate sources dealing with related topics not covered in this thesis.

2.1 Abstract Schwarz Methods

In this section, we outline the theory of Schwarz methods, a useful tool for analyzing many domain decomposition methods.

In recent years a lot of effort has been put into a systematic study of various types of the domain decomposition methods, generalizations of the alternating method of Schwarz [80]. Some of the pioneers of this work include, among others, Bjørstad and Mandel [5], Bramble, Pasciak and Schatz [10], Dryja, Proskurowski and Widlund [29], Matsokin and Nepomnyaschikh [71]. The Schwarz methods form a class of iterative methods for solving problems arising from discretization of partial differential equations. These methods can

be fully described by dividing the original solution space V of (1.4) into subspaces $V_i \subset H_{\Gamma_D \cap \bar{\Omega}_i}^1(\Omega_i)$, such that

$$\forall v \in V : v = v_1 + v_2 + \dots + v_J, \quad \text{where } v_i \in V_i, \quad (2.1)$$

and defining bilinear forms $a_i(\cdot, \cdot)$ on $V_i \times V_i$. These forms are assumed coercive and bounded on V_i with respect to $a(\cdot, \cdot)$ -norm. These assumptions will be rigorously formulated in Theorem 2.1.1. Note that the decomposition in (2.1) may not be always unique. The symbol V_0 denotes the so called coarse space, whose main purpose is to expediate the global correction of the error. Sometimes this space fulfills other roles, as we will see in Section 2.7. The solution in Schwarz methods is iteratively corrected by adding the approximation of the current error on subspaces V_i . To this end we define operators $\mathcal{T}_i : V \rightarrow V_i$ as

$$a_i(\mathcal{T}_i w, v) = a(w, v) \quad \forall v \in V_i, \quad i = 0, \dots, J. \quad (2.2)$$

The operators \mathcal{T}_i are thus defined uniquely up to the possible shift in the kernel of $a_i(\cdot, \cdot)$. All the methods formulated in the text will be independent of this shift. Note that if we set $a_i(u, v) = a(u, v)$, $\forall u, v \in V_i$, \mathcal{T}_i is an $a(\cdot, \cdot)$ -orthogonal projection onto V_i . This is why we refer to \mathcal{T}_i as to generalized projections.

Using the operators \mathcal{T}_i , a variety of methods may be constructed. They differ in several aspects. The subspaces V_i may correspond to overlapping or nonoverlapping subdomains $\Omega_i \subset \Omega$, allowing unique or non-unique representation of a function $u \in V$, respectively.

Schwarz methods can naturally be classified as multiplicative or additive, based on the manner in which the correction of the error is done. The

former is done following lines similar to the Gauss-Seidel method, while the latter is reminiscent of the Jacobi method (cf. [94, 55]). In either case, we can take the benefit of the fact that the (approximate) projection of the error $u^* - u$ can be attained without the knowledge of the exact solution u^* , using the definition of \mathcal{T}_i :

$$a_i(\mathcal{T}_i(u^* - u), v) = a(u^* - u, v) = f(v) - a(u, v), \quad \forall v \in V_i. \quad (2.3)$$

The multiplicative Schwarz method starts from $u = 0$ and proceeds by updating the current approximate solution $u \leftarrow u - u_i$, where

$$u_i = \mathcal{T}_i(u^* - u) \in V_i, \quad i = 0, \dots, J$$

is computed from (2.3). This process results in a nonsymmetric operator. If the method is to be used as a preconditioner in PCG, it is desirable to work with symmetric operator instead, in which case one can apply the updates once in forward and once in backward order.

The additive Schwarz method is defined as

$$u^{k+1} = u^k + \tau \sum_{j=0}^J u_j = u^k + \tau \sum_{j=0}^J \mathcal{T}_j(u^* - u^k), \quad (2.4)$$

where τ is an additional correction parameter. With the operator C defined by

$$CA = \sum_{i=0}^J \mathcal{T}_i \equiv \mathcal{T}, \quad (2.5)$$

we can rewrite the method (2.4) as

$$u^{k+1} = u^k + \tau C(f - Au^k). \quad (2.6)$$

Hence, we may view operator C as an approximate solver to $Au = f$, defined by

$$Cf = \sum_{i=0}^J u_i = \sum_{i=0}^J \mathcal{T}_i u, \quad u_i \in V_i,$$

where

$$a_i(u_i, v) = \langle f, v \rangle \quad \forall v \in V_i.$$

Application of (2.6) by itself may be fruitful if τ is chosen so that $\varrho(\tau CA) \leq 2$. A more common approach, however, is to use C as a preconditioner in the conjugate gradient method for problem (1.4). Then we obtain

$$\sigma(\mathcal{M}^{-1}A) = \sigma(CA) = \sigma(\mathcal{T}),$$

i.e., the convergence properties of the method may be studied as the spectrum of the sum of the approximate projections \mathcal{T}_i .

This path has been taken by Bjørstad and Mandel [5], Dryja, Smith and Widlund in [30], [32] and in a different setup by Xu [94]. The following theorem summarizes the results of their research.

Theorem 2.1.1 (Dryja, Widlund [32]). Let $\mathcal{T} = \sum_{i=1}^J \mathcal{T}_i$ and
 (i) there exists a constant C_0 such that for all $u \in V$ there exists a decomposition $u = \sum_{i=0}^J u_i, u_i \in V_i$, such that

$$\sum_{i=0}^J a_i(u_i, u_i) \leq C_0^2 a(u, u)$$

(ii) there exists a constant ω such that for $i = 0, \dots, J$,

$$a(u, u) \leq \omega a_i(u, u) \quad \forall u \in V_i$$

(iii) there exist constants ε_{ij} , $i, j = 1, \dots, J$, such that

$$a(u_i, u_j) \leq \varepsilon_{ij} a(u_i, u_i)^{1/2} a(u_j, u_j)^{1/2} \quad \forall u_i \in V_i, \forall u_j \in V_j.$$

Then, \mathcal{T} is invertible and

$$C_0^{-2} a(u, u) \leq a(\mathcal{T}u, u) \leq (\varrho(\varepsilon) + 1)\omega a(u, u) \quad \forall u \in V. \quad (2.7)$$

Here $\varrho(\epsilon)$ denotes the spectral radius of the matrix $\epsilon = \{\epsilon_{ij}\}_{i,j=1}^J$.

Proof. The theorem will be proved as an application of the more general framework of Section 4.3.1. Another approach may be found in [94]. \square

Remark 2.1.2. Assumption (i) of Theorem 2.1.1 by itself guarantees that $\inf \sigma(\mathcal{T}) \geq C_0^{-2}$. This result is known as the generalized Lions' lemma. We will need it in its original form in Section 3.1, where it is stated as Lemma 3.1.8.

The scale of methods is not limited to additive and multiplicative; one may combine them together. For instance, Cai [18] proposes to apply the multiplicative approach to the local solves ($i = 1, \dots, J$), while treating the space V_0 in additive fashion with respect to V_1, \dots, V_J . This allows solving the local problems in parallel with the coarse space problem. The local problem solving takes advantage of the more rapid convergence of multiplicative methods. Such an approach, however, inhibits parallelization of the local solves. Another possibility (cf. Mandel [64]) is to treat the local problems in additive way, while a coarse problem of modest size is added in a multiplicative fashion. This approach seems preferable because all the local problems may be solved in parallel, and the multiplicative coarse grid update is more efficient than an additive one.

2.2 Overlapping Schwarz

Possibly the most studied of domain decomposition type methods is the overlapping domain decomposition method ([80], [59], [31], [27]). In this method, the original domain Ω is covered by a collection of overlapping subdomains. The additive or multiplicative update of the solution described in previous section are then applied. Using the definition of (generalized) projectors \mathcal{T}_i , the error

propagation operator of the multiplicative method can be written as

$$(I - \mathcal{T}_0)(I - \mathcal{T}_J)(I - \mathcal{T}_{J-1}) \cdots (I - \mathcal{T}_1),$$

and the error propagation operator corresponding to the additive one is

$$I - \tau \sum_{i=0}^J \mathcal{T}_i.$$

Overlapping methods with a coarse space usually offer an improvement over the optimal non-overlapping methods in the sense that in the overlapping case with a coarse space the condition number can typically be bounded (cf. [33]) as

$$\text{cond}(\mathcal{M}^{-1}A) \leq C\left(1 + \frac{H}{\delta}\right),$$

where δ denotes the size of overlap of the subdomains. This better estimate comes at a price, though, as for three dimensional problems even a small overlap amounts to a significant increase in size of the local problems. This fact is especially pronounced if the subdomains are small. Another disadvantage of these methods is that their setup is usually not very flexible. Typically, when generating a system of overlapping subdomains, one starts from a system of nonoverlapping subdomains and adds new layers of elements around each of them. This is a cumbersome procedure and also one of the reasons why substructuring methods seem to enjoy more popularity today. Yet another issue troubling practical usability of overlapping Schwarz methods is the construction of a coarse space. For unstructured meshes it is difficult or impossible, using traditional geometric approach, to construct an automatic procedure yielding a coarse space that would be a subspace of the original finite element mesh; this presents a technical difficulty of having to deal with nonnested spaces (cf. [21, 20].)

The advantage of overlapping additive Schwarz methods is that they easily lend themselves to application of inexact subdomain solvers.

In Chapter 3 we will design and analyze an efficient black-box overlapping Schwarz method with a new coarse space avoiding the pitfalls of the traditional approach.

2.3 Non-overlapping Decomposition Methods

As noted in the previous section, the application of the traditional overlapping domain decomposition methods requires an undesirably complex mesh partitioning stage if the desired measure of overlapping is to be guaranteed. It is difficult to design automatic partitioning of complex geometries unless an algebraic concept similar to one described in Chapter 3 is used.

Nonoverlapping techniques simplify the partitioning stage significantly, being able to use a variety of existing aggregation or greedy algorithms [37, 38].

2.4 Methods Reduced to Interface

Many of the substructuring methods are based on reducing the original problem

$$Ax = b$$

to a problem defined only on the interface Γ common to at least two substructures by eliminating all degrees of freedom not associated with any subdomain interface. The initial problem is thus reduced to one whose unknowns are only the interface degrees of freedom. As the interior degrees of freedom corresponding to different subdomains have no coupling in the stiffness matrix A , the elimination can be done locally on each subdomain and in parallel.

Let $A^{(i)}$ denote the stiffness matrix of the bilinear form $a_i(\cdot, \cdot)$ defined on $V \times V$, which is the restriction of the bilinear form $a(\cdot, \cdot)$ to subdomain Ω_i , $a(u, v) = \sum_{i=1}^J a_i(u, v)$. For our model problem, given by operator (1.3), this is the stiffness matrix with entries $A_{kl}^{(i)} = \sum_{r,s=1}^d \int_{\Omega_i} \alpha(x) \beta_{rs}(x) \frac{\partial \phi_k(x)}{\partial x_s} \frac{\partial \phi_l(x)}{\partial x_r} dx$, where nodes v_k, v_l lie in Ω_i . The stiffness matrix A can be obtained by the standard process of subassembly

$$A = \sum_{i=1}^J N_i A^{(i)} N_i^T.$$

The matrices $S^{(i)}$ resulting from the elimination of the interior degrees of freedom of $A^{(i)}$ are the local Schur complements. Writing the local stiffness matrices $A^{(i)}$ in the block form

$$A^{(i)} = \begin{bmatrix} A_{11}^{(i)} & A_{12}^{(i)} \\ A_{12}^{(i)T} & A_{22}^{(i)} \end{bmatrix},$$

where $A_{11}^{(i)}$ corresponds to the subdomain interface and $A_{22}^{(i)}$ to the interior, the local Schur complements have the following form:

$$S^{(i)} = A_{11}^{(i)} - A_{12}^{(i)} A_{22}^{(i)-1} A_{21}^{(i)}. \quad (2.8)$$

The Schur complement S corresponding to the global stiffness matrix A can then be also obtained by the standard process of subassembly

$$S = \sum_{i=1}^J \bar{N}_i S^{(i)} \bar{N}_i^T. \quad (2.9)$$

We can now solve the problem

$$S \bar{x} = \tilde{b}, \quad (2.10)$$

where $\tilde{b} = \bar{b} - A_{12} A_{22}^{-1} \bar{b}$ is the reduced right-hand side obtained by the elimination of the interior degrees of freedom. Methods based on solving Schur complement

problems like this one instead of solving problems with the original stiffness matrices are labeled as substructuring methods by some authors. We will, however, use the term substructuring methods for a broader class of nonoverlapping methods, including the two-grid methods based on transfer operators with multilevel smoothing described in Chapter 5.

One advantage of the reduced methods is already obtained by solving the problem with Schur complement, which is significantly better conditioned than the original matrix A (cf. Lemma A.1.7). It also has many fewer unknowns. The early engineering applications of substructuring of 1960's were solving (2.10) by a direct solver (cf. [74]). Today, the reduced system (2.10) is usually solved by a preconditioned conjugate gradient method; the preconditioner is typically based on solution of the local subdomain problems.

Throughout the text we will find useful the concept of discrete harmonic functions. These are defined by

$$a(u_h, v) = 0 \quad \forall v \in H_0^1(\Omega_i) \cap V_h, \quad i = 1, \dots, J. \quad (2.11)$$

In matrix form, this is equivalent to

$$A_{21}^{(i)} \bar{x} + A_{22}^{(i)} \dot{x} = 0, \quad i = 1, \dots, J. \quad (2.12)$$

The discrete harmonic functions are uniquely determined by their interface values \bar{x} . Discrete harmonic functions are closely related to the Schur complement in the following sense: if x is the discrete harmonic extension of \bar{x} , then from (2.12), $\dot{x} = -A_{22}^{-1} A_{21} \bar{x}$ and simple manipulations yield

$$x^T A x = \bar{x}^T S \bar{x}.$$

In addition, the following lemma demonstrates that the discrete harmonic extensions (hence also the Schur energy norm) have the lowest energy among all vectors having the same values on the interfaces.

Lemma 2.4.1. Let A be a symmetric positive definite matrix partitioned into its interface and interior components $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix}$, and let S denote the Schur complement of A . Then

$$\bar{x}^T S \bar{x} = \inf_{\bar{y}=\bar{x}} y^T A y.$$

Proof. Given y such that $\bar{y} = \bar{x}$, let us solve the following problem:

$$\text{Find } w \text{ such that } \bar{w} = 0 \text{ and } \dot{v}^T A_{22} \dot{w} = -\dot{v}^T A_{21} \bar{y} - \dot{v}^T A_{22} \dot{y} \quad \forall \dot{v}$$

As A_{22} is nonsingular, such w exists and satisfies the equivalent minimization of the functional

$$F(v) = \frac{1}{2} v^T A v + y^T A v$$

over all vectors with $\bar{v} = 0$. As this constraint is trivially satisfied by the zero vector, we have $F(v) \leq 0$. Moreover, the function $z = y + w$ is the harmonic extension of \bar{x} , as $\bar{z} = \bar{x}$ and

$$z^T A v = y^T A v + w^T A v = 0 \quad \forall v \in \{u : \bar{u} = 0\}.$$

Thus we obtain

$$\langle Az, z \rangle = \langle A(y + w), y + w \rangle = \langle Ay, y \rangle + 2F(w) \leq \langle Ay, y \rangle.$$

From here the statement follows by the definition of Schur complement. \square

Figures 2.4, 2.4, 2.4, 2.4 and 2.4 depict the harmonic extension of simple functions prescribed on the interface; the functions exhibit the “smoothing” effect of discrete harmonic extension following from Lemma 2.4.1. Note that in certain situations the trivial extension may coincide with the discrete harmonic extension (cf. Figure 2.4).

It is well known [93] that in order to achieve good convergence properties, the preconditioner has to be augmented with a mechanism of global exchange of information. Let us note that another advantage of nonoverlapping methods is that on unstructured meshes it is much easier to algebraically construct a coarse space for them than for the overlapping methods (cf. [63]). Treatment of subdomains with nonmatching grids, using currently popular so called mortar elements, is also easier and was recently studied in [4], [2], [53], [57], [19] and many others. Mortar elements form a family of nonconforming finite element methods. They are known to be as accurate as their conforming counterparts, while offering more flexibility in refinement. The fine meshes on different subdomains do not have to match over the interface, even the subdomains themselves do not have to form a conforming coarse mesh. For the study of these methods we direct the reader to the above references.

2.5 The Neumann-Neumann Method

The original Neumann-Neumann preconditioning operator \mathcal{M}_{NN}^{-1} by De Roeck and Le Tallec [26] is based on the assembly (2.9) of the global Schur complement from the local subdomain Schur complements. It seems natural to precondition the sum (2.9) by the sum of properly weighted inverses of $S^{(i)-1}$.

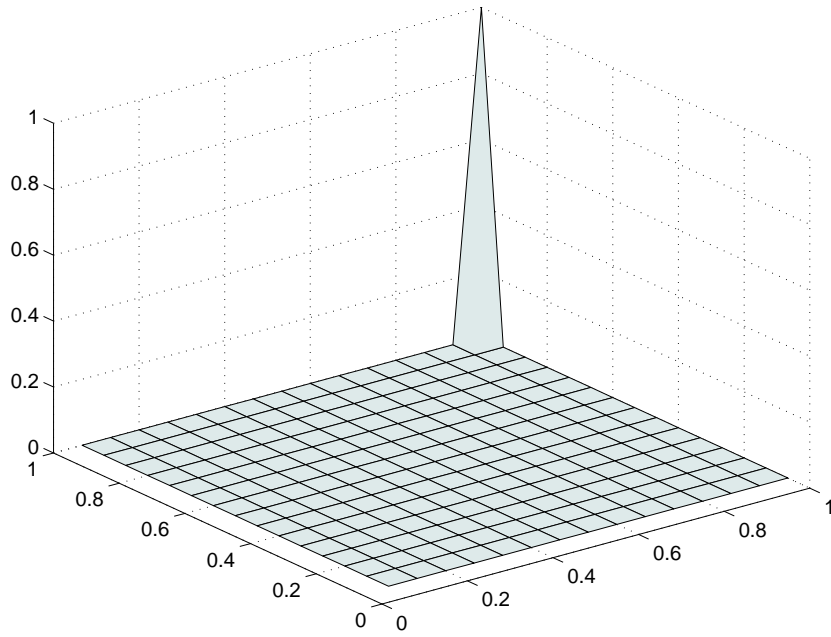


Figure 2.1: Harmonic extension of a corner peak.

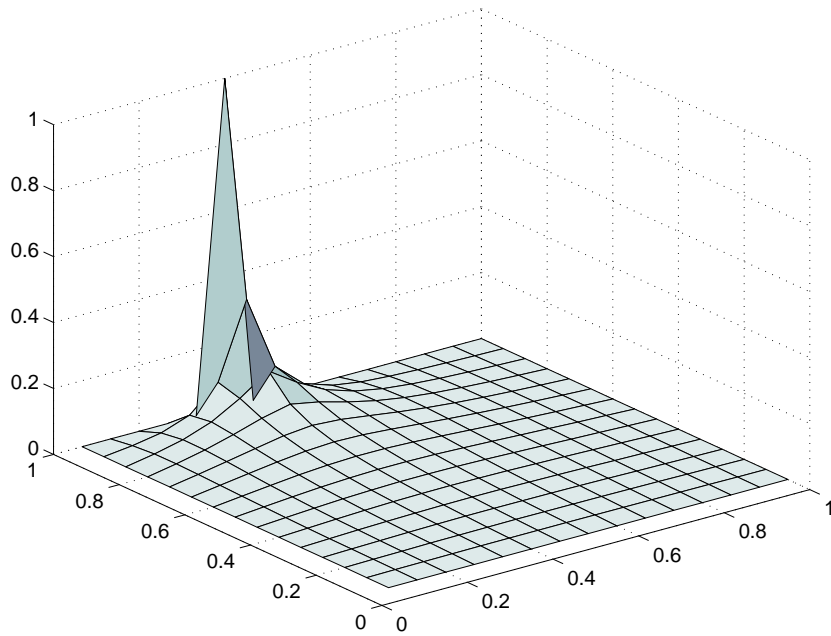


Figure 2.2: Harmonic extension of a peak on the edge.

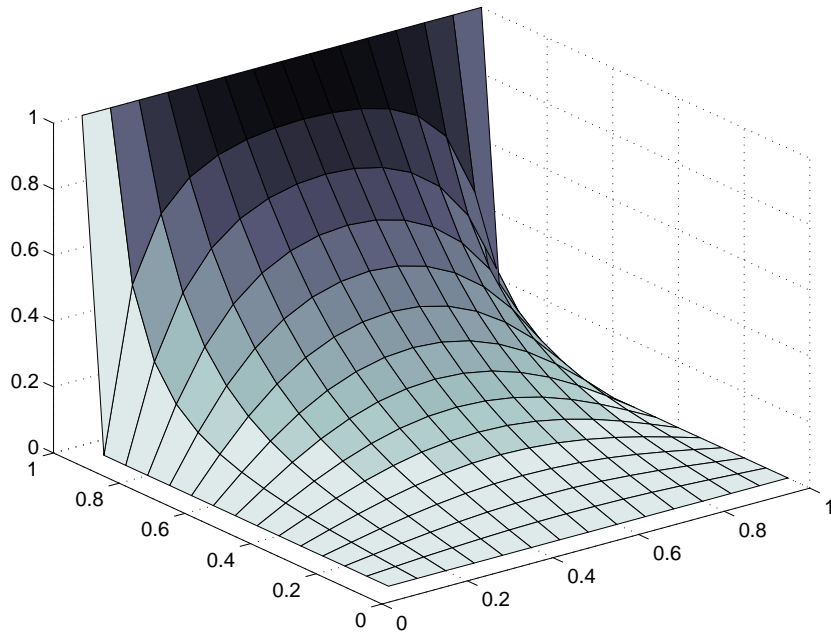


Figure 2.3: Harmonic extension of a function constant on one edge.

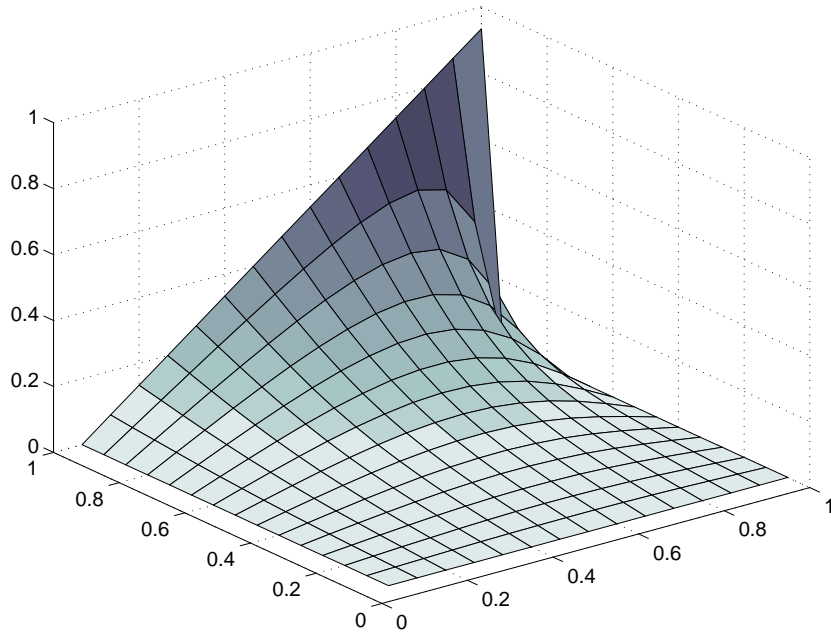


Figure 2.4: Harmonic extension of a function linear on one edge.

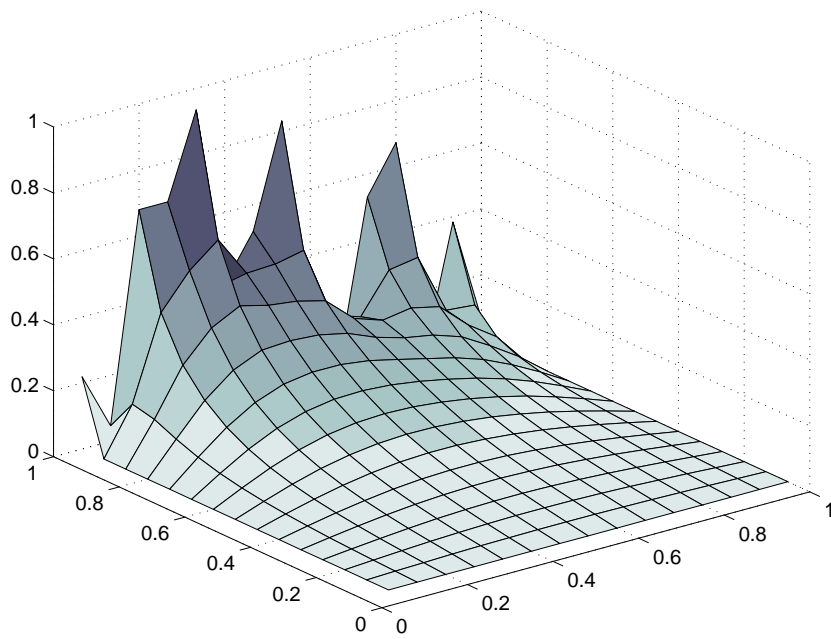


Figure 2.5: Harmonic extension of a function with random values on one edge.

If all subdomains are similar, this yields an efficient preconditioner, as

$$\text{cond} \left(\left(\sum_{i=1}^J S^{(i)} \right) \left(\sum_{i=1}^J S^{(i)-1} \right) \right) = \text{cond} \left(\sum_{i,j=1}^J S^{(j)-1} S^{(i)} \right) \approx O(1).$$

The application of inverses of the local Schur complement matrices corresponds on the continuous level to solving local Neumann problems on the subdomains. This is why preconditioners of this type are called Neumann-Neumann. The Neumann-Neumann preconditioner can be written as follows.

Algorithm 3 (Neumann-Neumann preconditioner, [26]). Given $r \in \bar{V}$, compute $z = \mathcal{M}_{NN}^{-1} r$ as follows.

1. Distribute r to subdomains,

$$r_i = D_i^T \bar{N}_i^T r.$$

2. Solve the local problems

$$S^{(i)} u_i = r_i \tag{2.13}$$

on the subdomains.

3. Average the results by

$$z = \sum_{i=1}^J \bar{N}_i D_i u_i.$$

The weight matrices D_i are an important component of this algorithm allowing to preserve good conditioning even when the subdomains differ in certain aspects, such as the values of coefficients of the continuous operator. Some appropriate choices for D_i were discussed in [26] and [66]. We will mention the choices suitable for problems with large variation of coefficients across the interfaces of the subdomains in Section 2.7 and also in Chapter 6.

Since for the model problem with operator (1.3) the local Schur complement matrices $S^{(ii)}$ are typically singular, De Roeck and Le Tallec [26] used

an “approximate” LU decomposition obtained by replacing zero pivots in the Gaussian decomposition by positive values.

This algorithm fails to provide any means of global distribution of information beyond that of block Jacobi type and, as expected, suffers from steep deterioration of convergence as the number of subdomains increases.

2.6 EBE method

In engineering practice, the solver is often very closely tied to the finite element software. One of such methods of the nonoverlapping domain decomposition type is the element-by-element method originally proposed by Hughes, Levit and Winget [50] and further studied by Hughes and Ferencz [48] and Tezduyar and Liou [84], [85].

This method, which may currently be a becoming industrial standard in the field of iterative solvers applied to nonlinear problems, assumes that the element stiffness matrices are available. Every element is treated as an independent subdomain. The method is used as a preconditioner in the conjugate gradient method. The main idea of the method is simple: The element matrices are factored and the preconditioner applied element by element based on these factorizations. The preconditioner may be written as

$$\mathcal{M} = \text{diag}(A)^{1/2} \left(\prod_{i=1}^{n_{el}} L(\tilde{A}^{(i)}) \right) \left(\prod_{i=1}^{n_{el}} D(\tilde{A}^{(i)}) \right) \left(\prod_{i=n_{el}}^{n_1} U(\tilde{A}^{(i)}) \right) \text{diag}(A)^{1/2},$$

where n_{el} denotes the number of elements in the system and L, D, U denote the factors of the Crout factorization of a matrix, $\tilde{A}^{(i)} = L(\tilde{A}^{(i)})D(\tilde{A}^{(i)})U(\tilde{A}^{(i)})$. Here all the element matrices are understood as embedded in an $n \times n$ identity matrix. The existence of the factorization is guaranteed because of the use of

positive definite local stiffness matrices $\tilde{A}^{(i)}$ obtained from $A^{(i)}$ by scaling and regularization $\tilde{A}^i = I + \text{diag}(A)^{-1/2}(A^i - \text{diag}(A^{(i)}))\text{diag}(A)^{-1/2}$.

As pointed out by Wathen [91], the method is essentially a multiplicative version of the original Neumann-Neumann method described in Algorithm 3. The difference is that the subdomain stiffness matrices in EBE are first scaled and regularized.

The chief advantage of the EBE method is its very low setup overhead compared to the typical two-level domain decomposition or multigrid methods. The method is very useful in treatment of nonlinear or time dependent problems, where refactorizations need to be performed. The cost of factorization of local element stiffness matrices is amortized over the iterations per timestep and over the number of timesteps, but because of the low factorization overhead due to the small size of element matrices, it is possible to perform refactorizations more often, which usually leads to faster overall convergence. The potential of this method is most visible in case of three-dimensional problems with low fractal dimension of the mesh, where direct factorizations are extremely inefficient.

The whole procedure is quite suitable for implementation on parallel architectures. This can be achieved by grouping the elements into noncontiguous subgroups. Although the subgroups have to be processed sequentially, the local stiffness matrices can then be factored in parallel for each element within a given subgroup. Thus, both factorization and the action of the preconditioner can be computed in parallel using this grouping of elements.

The obvious deficiency of the method is the lack of coarse space exchange of information. Despite this, computational experiments suggest that

EBE is suitable in application to nonlinear problems, where finding simple preconditioners is critical, and where EBE proved superior to diagonal preconditioning in terms of storage, convergence properties and sensitivity to penalty-type boundary conditions (cf., e.g., [48].) Another shortcoming, preventing black-box implementation, is EBE's reliance on the availability of the finite element data. In Chapter 3 we present a fully algebraic method with a coarse space, somewhat similar in spirit to EBE, with much improved convergence rate (independent of the number of subdomains) proved under regularity-free assumptions for problems on unstructured meshes.

2.7 BDD as an Algebraic Method

The BDD preconditioner [62] is based on the original Neumann-Neumann preconditioner by De Roeck and Le Tallec [26] described in Section 2.5. The algorithm proposed in [26] suffers from the lack of global distribution of information resulting in dramatic deterioration of performance with increasing number of subdomains. As noted in Le Tallec [55], the method becomes impractical when applied to problems with the number of subdomains larger than about 16. In order to defeat this drawback, Mandel [62] added a coarse problem as follows. Let $n_i = \dim V_i$, $0 \leq m_i \leq n_i$, and Z_i be $n_i \times m_i$ matrices of full column rank such that

$$\text{Ker } S^{(i)} \subset \text{Range } Z_i, \quad i = 1, \dots, J \quad (2.14)$$

and let $W \subset V$ be defined by

$$W = \{v \in V : v = \sum_{i=1}^J \bar{N}_i D_i u_i, \quad u_i \in \text{Range } Z_i\}.$$

The space W will play the role of a coarse space very much like in variational multigrid methods. We say that $s \in V$ is balanced if

$$Z_i^T D_i^T \bar{N}_i^T s = 0, \quad i = 1, \dots, J. \quad (2.15)$$

The process of replacing r by a balanced $s = r - Sw$, $w \in W$, is called balancing. We are now ready to define the action $r \mapsto z = \mathcal{M}^{-1}r$ of the BDD preconditioner.

Algorithm 4 (BDD preconditioner, [62]). Given $r \in V$, compute $\mathcal{M}^{-1}r$ as follows.

1. Pre-balance the original residual by solving the auxiliary problem for unknown vectors $\lambda_i \in \mathbb{R}^{m_i}$,

$$Z_i^T D_i^T \bar{N}_i^T (r - S \sum_{j=1}^J \bar{N}_j D_j Z_j \lambda_j) = 0, \quad i = 1, \dots, J. \quad (2.16)$$

2. Set

$$s = r - S \sum_{j=1}^J \bar{N}_j D_j Z_j \lambda_j, \quad s_i = D_i^T \bar{N}_i^T s, \quad i = 1, \dots, J. \quad (2.17)$$

3. Find any solution u_i for each of the local problems

$$S^{(i)} u_i = s_i, \quad i = 1, \dots, J. \quad (2.18)$$

4. Post-balance the residual by solving the auxiliary problem for $\mu_i \in \mathbb{R}^{m_i}$,

$$Z_i^T D_i^T \bar{N}_i^T (r - S \sum_{j=1}^J \bar{N}_j D_j (u_j + Z_j \mu_j)) = 0, \quad i = 1, \dots, J. \quad (2.19)$$

5. Average the result on the interfaces according to

$$z = \sum_{i=1}^J \bar{N}_i D_i (u_i + Z_i \mu_i). \quad (2.20)$$

If some $m_i = 0$, then Z_i as well as the block unknowns μ_i and λ_i are void and the i -th block equation is taken out of (2.16) and (2.19).

An important design choice for both the original Neumann-Neumann preconditioner and for BDD is the selection of weight matrices D_i that form a decomposition of unity on the interface space V ,

$$\sum_{i=1}^J \bar{N}_i D_i \bar{N}_i^T = I. \quad (2.21)$$

The most straightforward choice for D_i is a diagonal matrix with the diagonal elements being the reciprocal of the number of subdomains the degree of freedom is associated with. A better choice, which also guarantees a convergence bound independent of coefficient jumps between subdomains, is given in Theorem 2.7.3 below. For other possibilities, see [26] and notes in Chapter 6.

The presence of the coarse problem guarantees that the possibly singular local problems (2.18) are consistent. Indeed, after the pre-balancing Step 1. of Algorithm 4, the vector $s - \bar{N}_i D_i Z_i \quad \forall i = 1, \dots, J$. Therefore $s_i = D_i^T \bar{N}_i^T s$ is orthogonal to Z_i , and since $\text{Ker}(S_i) \subset Z_i$, $s_i \in \text{Range}(S_i)$.

The presence of the coarse problem also guarantees that the result of the algorithm does not depend on the choice of the solutions of (2.18), see [62].

In practice, the residual of the initial approximation should be balanced first as in (2.19); then the first balancing step (2.16) in every iteration can be omitted since the residual r received from the conjugate gradients algorithm is already balanced.

The addition of coarse space results in a very robust method with optimal substructuring convergence properties, with the condition number of the preconditioned operator bounded by $C(1 + \log(\frac{H}{h}))^2$, as we will demonstrate.

Mandel [63] proved the following abstract estimate as the main tool for convergence analysis of BDD.

Theorem 2.7.1. Algorithm 4 returns $z = \mathcal{M}^{-1}r$, where \mathcal{M} is symmetric positive definite and $\text{cond}(\mathcal{M}, S) \leq C$, where

$$C = \sup \left\{ \frac{\sum_{j=1}^J \|\bar{N}_j^T \sum_{i=1}^J \bar{N}_i D_i u_i\|_{S^{(j)}}^2}{\sum_{i=1}^J \|u_i\|_{S^{(i)}}^2} : u_i \in V_i, \right. \\ \left. \langle v_i, u_i \rangle = 0 \quad \forall v_i \in \text{Ker}(S^{(i)}), \right. \\ \left. \langle S^{(i)} v_i, u_i \rangle = 0 \quad \forall v_i \in \text{Range}(Z_i) \right\}. \quad (2.22)$$

Note that a different proof of Theorem 2.7.1 can be found in Chapter 4.4.2.

For the reader's convenience, we sketch the theory leading to the optimal substructuring condition number estimate. Our exposition follows Mandel, Brezina [66]. For more detailed analysis as well as computational experiments using BDD, see Mandel, Brezina [65].

In applying the bound of Theorem 2.7.1, we will find useful the concepts of glob and glob projection, defined as follows.

Definition 2.7.2 ([66]). Any vertex, edge, and, in the 3D case, face, of Γ will be called a glob. A glob is understood to be relatively open; for example, an edge does not contain its endpoints. We will also identify a glob with the set of the degrees of freedom associated with it. The set of all globs will be denoted by \mathcal{G} . For a glob $G \in \mathcal{G}$, define the glob projection as follows: for a vector $u \in V$, $E_G u \in V$ is the vector that has the same values as u for all degrees of freedom in G , and all other degrees of freedom of $E_G u$ are zero. The glob projection in terms of the local degrees of freedom is $E_G^{ji} = \bar{N}_j^T E_G \bar{N}_i : V_i \rightarrow V_j$.

Note that any two distinct globs from \mathcal{G} are disjoint and it holds that

$$\Gamma = \bigcup_{i=1}^J \partial\Omega_i \setminus \partial\Omega = \bigcup_{G \in \mathcal{G}} G.$$

The mappings E_G, E_G^{ij} correspond to zero-one matrices and satisfy

$$\sum_{G \in \mathcal{G}} E_G = I, \quad \bar{N}_j^T \bar{N}_i = \sum_{G \in \mathcal{G}} E_G^{ji}, \quad E_G^{ji} = E_G^{ji} E_G^{ii}, \quad (2.23)$$

and

$$G \subset \partial\Omega_i \cap \partial\Omega_j \iff E_G^{ji} \neq 0, \quad G \subset \partial\Omega_i \iff E_G^{ii} \neq 0. \quad (2.24)$$

We are now ready to give an abstract bound in the case when the matrices $S^{(i)}$ are scaled by arbitrary positive numbers α_i . This corresponds to coefficient discontinuities of arbitrary size between the subdomains. The theorem is formulated and proved exclusively in algebraic terms.

Theorem 2.7.3. Let $\alpha_i > 0$, $i = 1, \dots, J$, $t \geq 1/2$, and $E_G^{ji}, \bar{N}_i, S^{(i)}$, and Z_i satisfy (2.8), (2.23), and (2.14). Define D_i as the diagonal matrices

$$D_i = \sum_{G: E_G^{ii} \neq 0} d(i, G) E_G^{ii}, \quad d(i, G) = \frac{\alpha_i^t}{\sum_{j: E_G^{ji} \neq 0} \alpha_j^t}, \quad (2.25)$$

and assume that there exists a number R so that for all $i, j = 1, \dots, J$ and all $G \in \mathcal{G}$,

$$\frac{1}{\alpha_j} \|E_G^{ji} u_i\|_{S^{(j)}}^2 \leq \frac{1}{\alpha_i} R \|u_i\|_{S^{(i)}}^2 \quad (2.26)$$

for all u_i such that $u_i \perp \text{Ker}(S^{(i)})$, $S^{(i)} u_i \in \text{Range } Z_i$. Then the weight matrices D_i form a decomposition of unity (2.21), and the preconditioner defined by Algorithm 4 satisfies

$$\text{cond}(\mathcal{M}, S) \leq K^2 L^2 R, \quad (2.27)$$

where $K = \max_i |\{j : \bar{N}_j^T \bar{N}_i \neq 0\}|$, and $L = \max_{i,j} |\{G : E_G^{ji} \neq 0\}|$.

Proof. The decomposition of unity property (2.21) follows from the definition (2.25) and from (2.23),

$$\sum_{i=1}^J \bar{N}_i^T D_i \bar{N}_i = \sum_{i=1}^J \sum_{G: E_G^i \neq 0} d(i, G) E_G = \sum_{G \in \mathcal{G}} E_G = I .$$

Let j be fixed. Due to the bounded intersection of the subdomains, there are at most K nonzero terms in the sum $\sum_{i=1}^J \bar{N}_j^T \bar{N}_i D_i u_i$, and the triangle and Cauchy inequalities imply that

$$\left\| \sum_{i=1}^J \bar{N}_j^T \bar{N}_i D_i u_i \right\|_{S^{(j)}}^2 \leq \left(\sum_{i=1}^J \left\| \bar{N}_j^T \bar{N}_i D_i u_i \right\|_{S^{(j)}} \right)^2 \leq K \sum_{i=1}^J \left\| \bar{N}_j^T \bar{N}_i D_i u_i \right\|_{S^{(j)}}^2 ,$$

and

$$\sum_{j=1}^J \left\| \sum_{i=1}^J \bar{N}_j^T \bar{N}_i D_i u_i \right\|_{S^{(j)}}^2 \leq K^2 \sum_{i=1}^J \max_j \left\| \bar{N}_j^T \bar{N}_i D_i u_i \right\|_{S^{(j)}}^2 . \quad (2.28)$$

If $E_G^{j_i} \neq 0$, the coefficient $d(i, G)$ from (2.25) satisfies $d(i, G) \leq \alpha_i^t / (\alpha_i^t + \alpha_j^t)$, and it follows from (2.23) and from (2.26) that

$$\begin{aligned} \left\| \bar{N}_j^T \bar{N}_i D_i u_i \right\|_{S^{(j)}} &\leq \sum_{G: E_G^{j_i} \neq 0} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \left\| E_G^{j_i} u_i \right\|_{S^{(j)}} \leq \sum_{G: E_G^{j_i} \neq 0} \frac{\alpha_i^{t-1/2} \alpha_j^{1/2}}{\alpha_i^t + \alpha_j^t} R^{1/2} \|u_i\|_{S^{(i)}} \\ &\leq LR^{1/2} \sup_{\rho > 0} \frac{\rho^{1/2}}{1 + \rho^t} \|u_i\|_{S^{(i)}} \leq LR^{1/2} \|u_i\|_{S^{(i)}} . \end{aligned}$$

Therefore, by (2.28),

$$\sum_{j=1}^J \left\| \sum_{i=1}^J \bar{N}_j^T \bar{N}_i D_i u_i \right\|_{S^{(j)}}^2 \leq K^2 L^2 R \|u_i\|_{S^{(i)}}^2 ,$$

which concludes the proof, in view of Theorem 2.7.1. \square

Note that the constant K is the maximal number of adjacent subdomains Ω_j to any subdomain Ω_i plus one, and L is the maximal number of globs in any $\partial\Omega_i \cap \partial\Omega_j$. If $t > 1/2$, the estimate (2.27) can be slightly improved; in

particular, if $t = 1$, analogously to the method of De Roeck and Le Tallec [26], one has $\text{cond}(\mathcal{M}, S) \leq K^2 L^2 R/2$.

To apply Theorem 2.7.1, we first need to replace the $S^{(i)}$ norm by the scaled $H^{1/2}$ norm. This is a standard result [10, 31, 92], which we state here for reference in an α_i -scaled form suitable for our purposes.

Lemma 2.7.4. There exist constants $c > 0$, C independent of H or h so that

$$c \|u\|_{1/2, \partial\Omega_i}^2 \leq \frac{1}{\alpha_i} \|u\|_{S^{(i)}}^2 \leq C \|u\|_{1/2, \partial\Omega_i}^2 \quad \forall u \in V_h(\partial\Omega_i)$$

To derive the fundamental inequality (2.26) assumed in Theorem 2.7.3, we identify (by abuse of notation) V with $V_h(\Gamma)$ and V_i with $V_h(\partial\Omega_i)$. Then the glob projections are $E_G : V_h(\Gamma) \rightarrow V_h(\Gamma)$, and (2.26) becomes a bound on the increase of the $H^{1/2}$ norm when a function in $V_h(\partial\Omega_i)$ is changed by setting its values to zero on all nodes of $\partial\Omega_i \setminus G$.

We first consider the two-dimensional case, $\Omega \subset \mathbb{R}^2$. Since $\partial\Omega_i$ is one-dimensional, we may use the properties of the space $V_h(0, H)$ of piecewise linear functions on a uniform mesh with step h on the interval $[0, H]$. The following form of Discrete Sobolev Inequality was proved by Dryja [28].

Lemma 2.7.5. There exists a constant C such that

$$\|u\|_{L^\infty(0, H)}^2 \leq C \left(1 + \log \frac{H}{h}\right) \|u\|_{H^{1/2}(0, H)}^2 \quad \forall u \in V_h(0, H).$$

We will also need the following bound for the $H^{1/2}$ norm of the extension by zero from an interval to the whole \mathbb{R} , proved by Bramble, Pasciak, and Schatz [10, Lemma 3.5].

Lemma 2.7.6. There exists a constant C such that for all $u \in V_h(0, H)$ satisfying $u(0) = u(H) = 0$, $u = 0$ outside $(0, H)$,

$$|u|_{1/2, \mathbb{R}}^2 \leq C \left(1 + \log \frac{H}{h}\right) \|u\|_{L^\infty(0, H)}^2 + |u|_{1/2, (0, H)}^2.$$

An estimate of the $H^{1/2}$ norm of a function, obtained by sampling the value of a given function at one point, follows easily.

Lemma 2.7.7. There exists a constant C such that for all $u \in V_h(0, H)$, $0 \leq h \leq H$, and $v_0 \in V_h(\mathbb{R})$ defined by $v_0(0) = u(0)$, $v_0(x) = 0$ for $|x| \geq h$,

$$|v_0|_{1/2, \mathbb{R}}^2 \leq C \left(1 + \log \frac{H}{h}\right) \|u\|_{1/2, (0, H)}^2.$$

Proof. Let $L = \|u\|_{L^\infty(0, H)}$. It follows from Lemma 2.7.6 that

$$|v_0|_{1/2, \mathbb{R}}^2 \leq C \left(1 + \log \frac{2h}{h}\right) \|v_0\|_{L^\infty(-h, h)}^2 + |v_0|_{1/2, (-h, h)}^2. \quad (2.29)$$

Using linearity of v_0 , we obtain

$$|v_0|_{1/2, (-h, h)}^2 = \int_{-h}^h \int_{-h}^h \frac{|v_0(s) - v_0(t)|^2}{|s - t|^2} ds dt \leq 4 L^2, \quad (2.30)$$

because $\|v_0\|_{L^\infty(-h, h)}^2 = |v_0(0)|^2 \leq L^2$. Thus, $|v_0|_{1/2, (-h, h)}^2 \leq CL^2$. But $L^2 \leq C(1 + \log \frac{H}{h}) \|u\|_{1/2, (0, H)}^2$ by Lemma 2.7.5, which concludes the proof. \square

By subtracting such spikes at the endpoints, we can extend Lemma 2.7.6 to the case when the values of u at the endpoints are nonzero.

Lemma 2.7.8. There exists a constant C so that for $u \in V_h(0, H)$ and $w \in V_h(\mathbb{R})$ such that $w = u$ on $[h, H - h]$, and $w(x) = 0$ for $x \leq 0$, $x \geq H$,

$$|w|_{1/2, \mathbb{R}}^2 \leq C \left(1 + \log \frac{H}{h}\right)^2 \|u\|_{1/2, (0, H)}^2.$$

Proof. Define $u(x)$ to be zero for $x \in (-\infty, -h) \cup (H + h, \infty)$, and linear in $[-h, 0]$ and $[H, H + h]$. Further, define v_0 and v_H by

$$v_0(x) = \begin{cases} u(0), & x = 0, \\ 0, & |x| \geq h, \end{cases}$$

v_0 linear in $[-h, 0]$ and in $[0, h]$,

$$v_H(x) = \begin{cases} u(H), & x = H, \\ 0, & |x - H| \geq h, \end{cases}$$

v_H linear in $[H - h, H]$ and in $[H, H + h]$. Writing w as $w = u - v_0 - v_H$, and applying Lemma 2.7.6 and Lemma 2.7.7, we obtain

$$\begin{aligned} |w|_{1/2, \mathbf{R}}^2 &\leq C \left(1 + \log \frac{H}{h}\right) \|w\|_{L^\infty(0, H)}^2 + |w|_{1/2, (0, H)}^2 \\ &= C \left(1 + \log \frac{H}{h}\right) \|u\|_{L^\infty(0, H)}^2 + |w|_{1/2, (0, H)}^2 \\ &\leq C \left(1 + \log \frac{H}{h}\right) \|u\|_{L^\infty(0, H)}^2 + 3(|u|_{1/2, (0, H)}^2 + |v_0|_{1/2, \mathbf{R}}^2 + |v_H|_{1/2, \mathbf{R}}^2) \\ &\leq C \left(\left(1 + \log \frac{H}{h}\right) \|u\|_{L^\infty(0, H)}^2 + |u|_{1/2, (0, H)}^2 + \left(1 + \log \frac{H}{h}\right) \|u\|_{1/2, (0, H)}^2 \right). \end{aligned}$$

Application of Lemma 2.7.5 to the term $\|u\|_{L^\infty(0, H)}$ concludes the proof. \square

We now have the tools to estimate the $H^{1/2}$ norm of the glob projections E_G . This shows that an arbitrary function in $V_h(\partial\Omega_i)$ can be decomposed into its glob parts with only a small penalty of $H^{1/2}$ norm increase.

Theorem 2.7.9. Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$. Then there exists a constant C not dependent of h or H , so that for any glob $G \in \mathcal{G}$ and for all $u \in V_h(\partial\Omega_i)$,

$$\|E_G u\|_{1/2, \partial\Omega_i}^2 \leq C \left(1 + \log \frac{H}{h}\right)^2 \|u\|_{1/2, G}^2.$$

Proof. In the 2D case, the proposition follows by using a mapping of $\partial\Omega_i$ onto an interval so that G maps to $(0, H)$, from Lemma 2.7.8 for G being an edge, and from Lemma 2.7.7 for G being a vertex.

In the 3D case, the proposition was proved for the case of G being a face of $\partial\Omega_i$ as Lemma 4.3 in [12]. In the case of G being an edge or a vertex of $\partial\Omega_i$, the proof follows from Lemma 4.2. and the proof of Lemma 4.1. of [12]. \square

The bound on the condition number of the BDD algorithm follows.

Theorem 2.7.10. Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, and the weight matrices D_i be diagonal with the entries given by (2.25). Then there exists a constant C independent of H , h and α_i , so that the condition number of the BDD method satisfies

$$\text{cond}(\mathcal{M}, S) \leq C \left(1 + \log \frac{H}{h}\right)^2.$$

Proof. We only need to verify the assumption (2.26) of Theorem 2.7.3. Lemma 2.7.4 allows to replace the $S^{(i)}$ norms by the $H^{1/2}(\partial\Omega_i)$ seminorms, which may in turn be replaced by the $H^{1/2}(\partial\Omega_i)$ norms, owing to the Poincaré inequality (A.7) Now it suffices to use Theorem 2.7.9. \square

2.8 DD as a 2-level Multigrid

Another method similar to BDD was proposed by Mandel [64]. As in BDD, this method also treats the local problems in additive way, while a coarse problem of modest size is added in multiplicative fashion. The main difference between the two methods is that BDD requires subdomain local stiffness matrices as input, whereas the method of Mandel described in [64] uses submatrices of A instead.

The method may be written as follows:

Algorithm 5 (Hybrid Method). For a given $r \in V$, compute z as follows:

1. Compute the coarse level correction

$$z_0 \in V_0 : \quad a(z_0, v_0) = \langle r, v_0 \rangle \quad \forall v_0 \in V_0. \quad (2.31)$$

2. Compute the local subdomain solutions

$$u_i \in V_i : \quad a(u_i, v_i) = \langle r, v_i \rangle - a(z_0, v_i) \quad \forall v_i \in V_i, \quad i = 1, \dots, J. \quad (2.32)$$

3. Gather the solutions together

$$u = \sum_{i=1}^J u_i. \quad (2.33)$$

4. Compute and return the corrected solution

$$z_0 \in V_0 : \quad a(u - z_0, v_0) = \langle r, v_0 \rangle \quad \forall v_0 \in V_0. \quad (2.34)$$

$$z = u - z_0. \quad (2.35)$$

Algorithm 5 is nothing but a two-level variational multigrid for the problem $Sz = b$, with a special smoother defined by (2.32), (2.33). Steps (2.31) and (2.34) play the role of a coarse-grid correction. The fact that a method can be viewed as either a domain decomposition or multigrid is not a coincidence; the development of the recent years shows a tendency towards unification of convergence theory for multigrid and domain decomposition methods (e.g., Bramble, Pasciak, Wang, Xu [14], Wang [90]). In Xu [94], a selection of spaces is described with which the multiplicative domain decomposition and multigrid methods coincide.

2.9 Other Domain Decomposition Methods

The wealth of multilevel literature is astonishing. Of the methods strongly related to BDD, we mention two here: The method of Dryja and Widlund [32] uses the same coarse space W as BDD and weight exponent $t = 1/2$ in (2.25), but instead of using the local Schur complement matrices $S^{(i)}$ in (2.18), it uses $S^{(i)} + c_i M^{(i)}$ with $M^{(i)}$ positive definite. This avoids solving singular problems.

Sarkis [79] obtained an estimate for a method similar to BDD for nonconforming elements with any $t \geq 1/2$.

As the main purpose of this brief section is to give a few references to other material deserving attention but not covered in this thesis, we note that domain decomposition methods have been developed and successfully implemented for mixed formulations. There is also the dual approach to domain decomposition advocated by Farhat and Roux [39]. This approach is also suitable for solving problems arising from mixed formulations. For the application of domain decomposition techniques to mixed finite element methods, see [43], [34], [25] and the references therein. The dual approach has been introduced in [39] and further studied in [69], [36] and [55].

For domain decomposition methods on nonconforming finite element discretizations, see [4], [2], [53], [57], [19] and their references.

3. Fully Black-box Overlapping Schwarz Method

In this chapter, we propose a practical solver based on the framework of overlapping Schwarz methods. The practical disadvantage of the existing methods is that their setup usually requires knowledge of the mesh structure for generating the overlaps. Another issue troubling practical usability of overlapping Schwarz methods is the construction of a coarse space. The possibilities are either to start from a coarse finite element mesh forming nonoverlapping subdomains and obtain the fine discretization mesh by refining or to create an independent coarse mesh. The first approach requires the iterative solver to have control over the discretization, while the second brings about the technical difficulties of having to deal with nonnested spaces.

We will design and analyze a black-box overlapping Schwarz method with a new efficient coarse space avoiding the aforementioned pitfalls of the traditional approach. The creation of overlapping subdomains will be simplified here. We will start from a system of nonoverlapping subdomains, but we will not use the common construction of overlapping subdomains by adding new layers of elements to nonoverlapping ones. Instead, the overlaps will be obtained by a special smoothing procedure based on the stiffness matrix A . The desired amount of overlapping will then be achieved by algebraic means and controlled by the amount of smoothing done. We will also show that the assumed system of

nonoverlapping subdomains can efficiently be generated as a part of the method itself. The material in this section is based on the ongoing joint efforts with Petr Vaněk [17].

3.1 Overlapping Schwarz Method with a Coarse Space.

The purpose of this section is to specify requirements on the coarse space and overlapping subdomains that will allow us to prove uniform convergence. Requirements on the coarse space will be formulated in terms of its basis functions, keeping in mind that our final goal is a black-box method efficient on unstructured meshes. For this reason, we avoid the assumption on the L^∞ boundedness of the gradient of basis functions.

Before we introduce notation specific to this section, let us briefly recall some assumptions on the problem to be solved. Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$ be a Lipschitz domain and \mathcal{T}_h be a quasiuniform finite element mesh on Ω of a characteristic meshsize h . Let V_h be a $P1$ or $Q1$ finite element space associated with the mesh \mathcal{T}_h with zero Dirichlet boundary conditions imposed at some finite element nodes $v_i \in \Gamma_D \subset \partial\Omega$. Those nodes will be referred to as constrained nodes. For simplicity, we assume that the finite element basis functions φ_i are scaled so that $\|\varphi_i\|_{L^\infty} = 1$. We consider the finite element discretization

$$Ax = b \tag{3.1}$$

of the following elliptic model problem: Find $u \in V_h$ such that

$$a(u, v) = (f, v)_{L^2(\Omega)} \quad \forall v \in V_h, \tag{3.2}$$

where

$$a(u, v) = \sum_{i=1}^d \int_{\Omega} a(x) \frac{\partial u(x)}{\partial x_i} \frac{\partial v(x)}{\partial x_i} dx, \quad 0 < C_1 \leq a(x) \leq C_2 \text{ for all } x \in \Omega,$$

and V_h denotes the finite element space with resolution h . Clearly,

$$C_1 |u|_{H^1(\Omega)}^2 \leq a(u, u) \leq C_2 |u|_{H^1(\Omega)}^2.$$

Let us consider the system of overlapping subdomains $\{\Omega_j\}_{j=1}^J$ covering the computational domain Ω . The coarse space $V_0 \subset V_h$ will be defined by its basis, i.e.

$$V_0 = \text{span}\{\Phi_i\}_{i=1}^J$$

and local fine level spaces $\{V_j\}_{j=1}^J$ will be determined by subdomains Ω_j via

$$V_j = V_h \cap H_{0,(\partial\Omega_j \setminus \Gamma_N)}^1(\Omega_j) \quad j = 1, \dots, J, \quad (3.3)$$

where $\Gamma_N = \partial\Omega \setminus \Gamma_D$ is the part of the boundary with the natural boundary condition imposed.

For the sake of parallelism, we assume that we have a system $\{\mathcal{C}_j\}_{j=1}^{n_c}$ of index sets \mathcal{C}_j such that

- (1) $\bigcup_{i=1}^{n_c} \mathcal{C}_i = \{1, \dots, J\}$, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$, $i \neq j$.
- (2) $V_k \perp_A V_l$ for every $k, l \in \mathcal{C}_j$.

Denoting by P_i the $a(\cdot, \cdot)$ -orthogonal projections onto V_i , respectively, the error propagation operator of the method we will analyze can be written as

$$T = (I - P_0) \prod_{i=1}^{n_c} \left(I - \sum_{j \in \mathcal{C}_i} P_j \right). \quad (3.4)$$

Note that due to the A -orthogonality of the spaces V_j, V_k for $j, k \in \mathcal{C}_i$, we have

$$I - \sum_{j \in \mathcal{C}_i} P_j = \prod_{j \in \mathcal{C}_i} (I - P_j), \quad (3.5)$$

so the method may be viewed as either a hybrid or purely multiplicative. This fact allows for parallelism in the computation of subdomain error corrections.

The following two assumptions sum up our requirements on the system of subdomains and on the coarse space basis functions that will be sufficient for proving uniform convergence of the algorithm with error propagation operator (3.4).

Assumption 3.1.1 (Subdomain Geometry). Let Ω be union of simply connected clusters Ω_j of finite elements such that

- a) $\text{diam}(\Omega_j) \leq CH, \quad j = 1, \dots, J.$
- b) $\exists c > 0 \forall x \in \Omega \exists \Omega_j : x \in \Omega_j \quad \& \quad \text{dist}(x, \partial\Omega_j \setminus \partial\Omega) \geq cH, \quad j = 1, \dots, J.$
- c) $\exists K_1, K_2 > 0 : \forall x \in \Omega$ the ball $B(x, C_R H) = \{y \in \Omega : \text{dist}(y, x) \leq C_R H\}$ intersects at most $K_1 + K_2 C_R^d$ subdomains Ω_j .
- d) $\text{meas}(\Omega_j) \geq CH^d, \quad j = 1, \dots, J.$

Assumption 3.1.1 c) will be used in the following context: An object of a diameter $O(H)$ intersects at most $O(1)$ subdomains Ω_i .

Assumption 3.1.2 (Coarse Space Basis Functions). We assume that the basis functions Φ_i of the coarse space V_0 satisfy

- a) $|\Phi_i|_{H^1(\Omega)} \leq CH^{\frac{d-2}{2}}, \quad \|\Phi_i\|_{L^2(\Omega)} \leq CH^{d/2}.$
- b) There is a domain $\Omega^{\text{int}} \subset \Omega$ such that $\sum_i \Phi_i(x) = 1$ for every $x \in \Omega^{\text{int}}$ and $\text{dist}(x, \Gamma_D) \leq CH$ for every $x \in \Omega \setminus \Omega^{\text{int}}$.
- c) $\text{supp}(\Phi_i) \subset \bar{\Omega}_i.$

Note that the assumption can trivially be satisfied by P1 or Q1 finite element basis functions. We will, however, design a coarse space basis more appropriate for unstructured meshes. As we will demonstrate, the basis of smoothed

aggregation functions described in Section 3.2 is a good candidate.

In the rest of this section we will prove the following theorem:

Theorem 3.1.3. Under Assumptions 3.1.2 and 3.1.1, for the error propagation operator (3.4) of Algorithm 8 defined in Section 3.2 below it holds that

$$\|T\|_a \leq 1 - C, \quad \|\cdot\|_a = a(\cdot, \cdot)^{1/2},$$

with a constant C independent of H , h .

The convergence proof will rely on the use of Lions' lemma [5], for application of which we need the following simple existence result:

Lemma 3.1.4. Let Assumption 3.1.1 be satisfied. Then, there exists a set of functions $\{\psi_j\}_{j=1}^J$, $\psi_j \in W^{1,\infty}(\Omega)$ such that

1. $|\psi_i|_{W^{1,\infty}(\Omega)} \leq CH^{-1}$,
2. $\sum_{j=1}^J \psi_j = 1$ on Ω ,
3. $\psi_j = 0$ on $\Omega \setminus \Omega_j$.

Proof. For each Ω_j we define

$$\tilde{\psi}_j(x) = \begin{cases} H^{-1} \text{dist}(x, \partial\Omega_j \setminus \partial\Omega) & \text{for } x \in \Omega_j, \\ 0 & \text{for } x \in \Omega \setminus \Omega_j. \end{cases}$$

Due to parts a) and c) of the Assumption 3.1.1,

$$\sum_{j=1}^J \tilde{\psi}_j \leq C \tag{3.6}$$

and from the part b), we also have

$$\sum_{j=1}^J \tilde{\psi}_j \geq c. \tag{3.7}$$

We will show that

$$|\tilde{\psi}_j|_{W^{1,\infty}(\Omega)} \leq CH^{-1}. \quad (3.8)$$

Consider two points $u, v \in \Omega_j$. Without the loss of generality, we assume $\tilde{\psi}_j(u) \leq \tilde{\psi}_j(v)$. Then we have

$$\tilde{\psi}_j(u) = H^{-1} \text{dist}(u, \partial\Omega_j \setminus \partial\Omega) = H^{-1} \text{dist}(u, P)$$

for some point $P \in \partial\Omega_j \setminus \partial\Omega$. Further, using triangle inequality,

$$\begin{aligned} \tilde{\psi}_j(v) &= H^{-1} \text{dist}(v, \partial\Omega_j \setminus \partial\Omega) \\ &\leq H^{-1} \text{dist}(v, P) \\ &\leq H^{-1} (\text{dist}(v, u) + \text{dist}(u, P)) \\ &\leq H^{-1} \text{dist}(v, u) + \tilde{\psi}_j(u). \end{aligned}$$

Therefore,

$$|\tilde{\psi}_j|_{Lip(\Omega)} := \sup \left\{ \frac{|\tilde{\psi}_j(x) - \tilde{\psi}_j(y)|}{\text{dist}(x, y)} : x, y \in \Omega; x \neq y \right\} \leq H^{-1}.$$

Now, (3.8) follows from the well-known equivalence $|\cdot|_{Lip(\Omega)} \approx |\cdot|_{W^{1,\infty}(\Omega)}$.

Let us define

$$w(x) = \frac{1}{\sum_{j=1}^J \tilde{\psi}_j(x)}, \quad x \in \bar{\Omega}.$$

Due to (3.6) and (3.7),

$$\|w\|_{L^\infty(\bar{\Omega})} \leq C. \quad (3.9)$$

Further, from (3.7), (3.8) and bounded intersections of subdomains Ω_j , denoting the Euclidean norm in \mathbb{R}^d by $\|\cdot\|$,

$$\|\nabla w(x)\| \leq \|\nabla(\sum_{j=1}^J \tilde{\psi}_j)(x)\| \left(\min_{y \in \bar{\Omega}} \sum_{j=1}^J \tilde{\psi}_j(y) \right)^{-2}$$

$$\begin{aligned}
&\leq \sum_{j:x \in \Omega_j} \|\nabla \tilde{\psi}_j(x)\| \left(\min_{y \in \tilde{\Omega}} \sum_{j=1}^J \tilde{\psi}_j(y) \right)^{-2} \\
&\leq CH^{-1}
\end{aligned} \tag{3.10}$$

for every $x \in \Omega \setminus \mathcal{B}$, where \mathcal{B} is a set of zero measure. Finally, we set

$$\psi_j = w\tilde{\psi}_j, \quad j = 1, \dots, J.$$

Statements 2. and 3. of the lemma are trivially satisfied by functions ψ_j . Further, (3.8), (3.9) and (3.10) imply

$$\begin{aligned}
\|\nabla \psi_j(x)\| &= \|w(x)(\nabla \tilde{\psi}_j)(x) + \tilde{\psi}_j(x)(\nabla w)(x)\| \\
&\leq \|\nabla \tilde{\psi}_j(x)\| \cdot |w(x)| + |\tilde{\psi}_j(x)| \cdot \|\nabla w(x)\| \\
&\leq CH^{-1}
\end{aligned}$$

almost everywhere, which proves statement 1. of the lemma. \square

Let us define the linear interpolation operator $Q : H^1(\Omega) \rightarrow V_0$ by

$$Qu = \sum_{i=1}^J \alpha_i \Phi_i, \quad \text{where } \alpha_i = \alpha_i(u) = \frac{1}{\text{meas}(\Omega_i)} \int_{\Omega_i} u(x) dx. \tag{3.11}$$

As by Cauchy-Schwarz inequality

$$\left| \int_{\Omega_i} u(x) dx \right| \leq |(u, 1)_{L^2(\Omega_i)}| \leq \text{meas}(\Omega_i)^{1/2} \|u\|_{L^2(\Omega_i)}$$

and by Assumption 3.1.1 d)

$$\text{meas}(\Omega_i) \geq CH^d,$$

we obtain the following bound for the coefficients of interpolation Q :

$$|\alpha_i(u)| \leq CH^{-d/2} \|u\|_{L^2(\Omega_i)}. \tag{3.12}$$

The proof of the following lemma is essentially standard, except for certain technical difficulties stemming from allowing rather general geometry of subdomains Ω_j .

Lemma 3.1.5. Under Assumptions 3.1.2, 3.1.1, for the interpolation operator Q defined by (3.11) and every $u \in H_{0,\Gamma_D}^1(\Omega)$ it holds that

$$\|u - Qu\|_{L^2(\Omega)} \leq CH|u|_{H^1(\Omega)} \quad (3.13)$$

and

$$|Qu|_{H^1(\Omega)} \leq C|u|_{H^1(\Omega)}, \quad (3.14)$$

where C is a constant independent of H and h .

Remark 3.1.6. The inequality (3.13) is one of the forms of the so-called weak approximation property (cf., [15]). It can be contrasted with the traditional approximation property

$$\|u - Qu\|_{H^1(\Omega)} \leq CH|u|_{H^2(\Omega)}$$

used in the multigrid and finite element literature.

Proof. Let us set $B = \Omega \setminus \Omega^{\text{int}}$, where Ω^{int} is the domain introduced in Assumption 3.1.2, b). Further we define

$$\mathcal{B} = \{i : \Omega_i \cap B \neq \emptyset\}, \quad B' = \bigcup_{i \in \mathcal{B}} \Omega_i, \quad W = \sup_{x \in B'} \{\text{dist}(x, \Gamma_D)\}$$

and set

$$B_0 = \{x \in \Omega : \text{dist}(x, \Gamma_D) \leq W\}.$$

From Assumption 3.1.1 it immediately follows that $W \leq CH$ and therefore the Poincaré inequality yields

$$\|u\|_{L^2(B)} \leq \|u\|_{L^2(B_0)} \leq C(\Gamma_D)H|u|_{H^1(B_0)}. \quad (3.15)$$

Then, the restriction of Qu onto B can be expressed as

$$(Qu)(x) = \sum_{i \in \mathcal{B}} \alpha_i(u) \Phi_i(x), \quad x \in B.$$

Further, let us set $\mathcal{N}_i = \{j : \Omega_j \cap \Omega_i \neq \emptyset\}$. The inequalities

$$\|\Phi_i\|_{L^2(\Omega)} \leq CH^{d/2}, \quad |\alpha_i(u)| \cdot |\alpha_j(u)| \leq \frac{1}{2}(\alpha_i^2(u) + \alpha_j^2(u))$$

together with bounded intersections and (3.12) yield:

$$\begin{aligned} \|Qu\|_{L^2(B)}^2 &= \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}_i \cap \mathcal{B}} (\alpha_i(u) \Phi_i, \alpha_j(u) \Phi_j)_{L^2(B)} \\ &\leq \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}_i \cap \mathcal{B}} |\alpha_i(u)| \cdot |\alpha_j(u)| \|\Phi_i\|_{L^2(\Omega)} \|\Phi_j\|_{L^2(\Omega)} \\ &\leq CH^d \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}_i \cap \mathcal{B}} \frac{1}{2}(\alpha_i^2(u) + \alpha_j^2(u)) \\ &\leq CH^d \max\{\text{card}(\mathcal{N}_i)\} \sum_{i \in \mathcal{B}} \alpha_i^2(u) \\ &\leq C \sum_{i \in \mathcal{B}} \|u\|_{L^2(\Omega_i)}^2 \\ &\leq C \|u\|_{L^2(B_0)}^2. \end{aligned} \tag{3.16}$$

Using the last inequality together with (3.15) gives

$$\|(I - Q)u\|_{L^2(B)} \leq \|u\|_{L^2(B)} + \|Qu\|_{L^2(B)} \leq CH|u|_{H^1(B_0)}. \tag{3.17}$$

Analogously to estimate (3.16), using

$$|\Phi_i|_{H^1(\Omega)} \leq CH^{(d-2)/2}$$

in place of $\|\Phi_i\|_{L^2(\Omega)} \leq CH^{d/2}$ (see Assumption 3.1.2), we arrive at

$$|Qu|_{H^1(B)}^2 \leq CH^{-2} \|u\|_{L^2(B_0)}^2 \leq C|u|_{H^1(B_0)}^2, \tag{3.18}$$

where in the last step (3.15) has been used.

In the rest of the proof we will verify (3.17) and (3.18) on the domain Ω^{int} (see Assumption 3.1.2). For the convenience of the proving, let us consider the extension u_E of the function $u \in H_{0,\Gamma_D}^1(\Omega)$ satisfying

$$|u_E|_{H^1(\mathbb{R}^d)} \leq C(\Omega)|u|_{H^1(\Omega)}, \quad u_E = u \text{ on } \Omega. \quad (3.19)$$

Let us recall that $\mathcal{N}_i = \{j : \Omega_j \cap \Omega_i \neq \emptyset\}$. For $i = 1, \dots, J$ we define $B'_i = \bigcup_{j \in \mathcal{N}_i} \Omega_j$ and B_i to be a ball circumscribed about B'_i . From Assumption 3.1.1 it immediately follows that

$$\text{diam}(B_i) \leq CH$$

and therefore, we have the Friedrichs inequality in the form

$$\|u\|_{L^2(B_i)} \leq CH|u|_{H^1(B_i)} \quad \forall u \in \{v \in H^1(B_i) : \int_{B_i} v \, dx = 0\}. \quad (3.20)$$

Further, due to Assumption 3.1.1 a), c), the intersections of balls B_i are bounded.

For every $j = 1, \dots, J$ we define

$$c_j = \int_{B_j} u_E \, dx, \quad \bar{u}_j = u_E - c_j. \quad (3.21)$$

Then, the Friedrichs inequality (3.20) holds for every \bar{u}_j . Due to Assumption 3.1.2 b), for $x \in \Omega_i \cap \Omega^{\text{int}}$ it holds that

$$\begin{aligned} (Qu)(x) &= (Q\bar{u}_i)(x) + Qc_i \\ &= \sum_{j \in \mathcal{N}_i} \alpha_j(\bar{u}_i)\Phi_j(x) + c_i \sum_{j \in \mathcal{N}_i} \Phi_j(x) \\ &= (Q\bar{u}_i)(x) + c_i. \end{aligned} \quad (3.22)$$

Therefore,

$$\|(I - Q)u\|_{L^2(\Omega^{\text{int}})}^2 \leq \sum_{i=1}^J \|(I - Q)u\|_{L^2(\Omega_i \cap \Omega^{\text{int}})}^2$$

$$\begin{aligned}
&= \sum_{i=1}^J \|(I - Q)(\bar{u}_i + c_i)\|_{L^2(\Omega_i \cap \Omega^{\text{int}})}^2 \\
&= \sum_{i=1}^J \|(I - Q)\bar{u}_i\|_{L^2(\Omega_i \cap \Omega^{\text{int}})}^2 \\
&\leq 2 \sum_{i=1}^J \left(\|\bar{u}_i\|_{L^2(B_i)}^2 + \|Q\bar{u}_i\|_{L^2(\Omega_i \cap \Omega^{\text{int}})}^2 \right). \quad (3.23)
\end{aligned}$$

Further,

$$\begin{aligned}
\|Q\bar{u}_i\|_{L^2(\Omega_i \cap \Omega^{\text{int}})}^2 &\leq \left\| \sum_{j \in \mathcal{N}_i} \alpha_j(\bar{u}_i) \Phi_j \right\|_{L^2(\Omega)}^2 \\
&\leq \left(\sum_{j \in \mathcal{N}_i} |\alpha_j(\bar{u}_i)| \|\Phi_j\|_{L^2(\Omega)} \right)^2 \\
&\leq \text{card}(\mathcal{N}_i) \sum_{j \in \mathcal{N}_i} \alpha_j^2(\bar{u}_i) \|\Phi_j\|_{L^2(\Omega)}^2 \\
&\leq CH^{-d} H^d \sum_{j \in \mathcal{N}_i} \|\bar{u}_i\|_{L^2(\Omega_j)}^2 \quad [\text{Assumption 3.1.2 a) and (3.12)}] \\
&\leq C \|\bar{u}_i\|_{L^2(B_i)}^2.
\end{aligned}$$

Substituting the last inequality into (3.23) and using the Friedrichs inequality (3.20) together with bounded intersections of balls $\{B_i\}_{i=1}^J$, we get

$$\begin{aligned}
\|(I - Q)u\|_{L^2(\Omega^{\text{int}})}^2 &\leq C \sum_{i=1}^J \|\bar{u}_i\|_{L^2(B_i)}^2 \\
&\leq CH^2 \sum_{i=1}^J |\bar{u}_i|_{H^1(B_i)}^2 \\
&= CH^2 \sum_{i=1}^J |u_E|_{H^1(B_i)}^2 \\
&\leq CH^2 |u|_{H^1(\Omega)}^2.
\end{aligned}$$

The last inequality together with (3.17) proves (3.13). Similarly,

$$|Qu|_{H^1(\Omega^{\text{int}})}^2 \leq \sum_{i=1}^J |Qu|_{H^1(\Omega_i \cap \Omega^{\text{int}})}^2$$

$$\begin{aligned}
&= \sum_{i=1}^J |Q(\bar{u}_i + c_i)|_{H^1(\Omega_i \cap \Omega^{\text{int}})}^2 && \text{[used (3.21)]} \\
&= \sum_{i=1}^J |Q\bar{u}_i + c_i|_{H^1(\Omega_i \cap \Omega^{\text{int}})}^2 && \text{[used (3.22)]} \\
&= \sum_{i=1}^J |Q\bar{u}_i|_{H^1(\Omega_i \cap \Omega^{\text{int}})}^2 \\
&\leq \sum_{i=1}^J \left| \sum_{j \in \mathcal{N}_i} \alpha_j(\bar{u}_i) \Phi_j \right|_{H^1(\Omega)}^2 \\
&\leq \sum_{i=1}^J \left(\sum_{j \in \mathcal{N}_i} |\alpha_j(\bar{u}_i)| |\Phi_j|_{H^1(\Omega)} \right)^2 \\
&\leq \sum_{i=1}^J \left(\text{card}(\mathcal{N}_i) \sum_{j \in \mathcal{N}_i} \alpha_j^2(\bar{u}_i) |\Phi_j|_{H^1(\Omega)}^2 \right) \\
&\leq CH^{d-2} H^{-d} \sum_{i=1}^J \sum_{j \in \mathcal{N}_i} \|\bar{u}_i\|_{L^2(\Omega_j)}^2 && \text{[Assumption 3.1.2a) and (3.12)]} \\
&\leq CH^{-2} \sum_{i=1}^J \|\bar{u}_i\|_{L^2(B_i)}^2 \\
&\leq C \sum_{i=1}^J |\bar{u}_i|_{H^1(B_i)}^2 && \text{[used Friedrichs inequality]} \\
&= C \sum_{i=1}^J |u_E|_{H^1(B_i)}^2 \\
&\leq C |u|_{H^1(\Omega)}^2 && \text{[used bounded intersections and (3.19)]}
\end{aligned}$$

which together with (3.18) completes the proof of (3.14). \square

The previous lemma allows us to prove existence of a H^1 -stable decomposition of function $u \in V_h$ into subdomain components.

Lemma 3.1.7. Under Assumptions 3.1.2, 3.1.1, for every finite element function $u \in V_h$, there is a (not necessarily unique) decomposition $\{u_i\}_{i=0}^J$, $u_i \in V_i$ such that

$$u = \sum_{i=0}^J u_i \quad \text{and} \quad \sum_{i=0}^J |u_i|_{H^1(\Omega)}^2 \leq C \|u\|_{H^1(\Omega)}^2,$$

where constant C is independent of h, H .

Proof. Let us define the operator $I_h : C^0(\bar{\Omega}) \rightarrow V_h$ by

$$I_h(u) = \sum_{i=1}^n u(v_i) \phi_i = \Pi(u(v_i)_{i=1}^n),$$

where $\{v_i\}_{i=1}^n$ is the set of finite element nodal points, $\{\phi_i\}_{i=1}^n$ is the finite element basis, and $\Pi x = \sum_{i=1}^n x_i \phi_i$ is the finite element interpolator. Let us consider the basis $\{\psi_i\}_{i=1}^J$ from Lemma 3.1.4. As $\text{diam}(\text{supp}(\psi_i)) \leq CH$ and $|\psi_i|_{W^{1,\infty}(\Omega)} \leq CH^{-1}$, we also have

$$\|\psi_i\|_{L^\infty(\Omega)} \leq C.$$

Let us define the decomposition

$$\begin{aligned} u_0 &= Qu, \\ u_i &= I_h(\psi_i w), \quad i = 1, \dots, J, \quad \text{where} \\ w &= (I - Q)u \end{aligned}$$

and Q is the interpolation operator from Lemma 3.1.5. As w is a finite element function and $\sum_{i=1}^J \psi_i = 1$,

$$\sum_{i=1}^J u_i = I_h\left(\sum_{i=1}^J \psi_i w\right) = I_h(w) = w,$$

proving validity of the first statement of this lemma. Further, for $i = 1, \dots, J$ it holds that

$$\begin{aligned} |u_i|_{H^1(\Omega_i)} &= |I_h(\psi_i w)|_{H^1(\Omega_i)} \\ &\leq C |\psi_i w|_{H^1(\Omega_i)} \\ &\leq C \|\nabla(\psi_i w)\|_{[L^2(\Omega_i)]^d} \end{aligned}$$

$$\begin{aligned}
&= C \|w \nabla \psi_i + \psi_i \nabla w\|_{[L^2(\Omega_i)]^d} \\
&\leq C (\|w \nabla \psi_i\|_{[L^2(\Omega_i)]^d} + \|\psi_i \nabla w\|_{[L^2(\Omega_i)]^d}) \\
&\leq C (\|\nabla \psi_i\|_{[L^\infty(\Omega_i)]^d} \|w\|_{L^2(\Omega_i)} + \|\nabla w\|_{[L^\infty(\Omega_i)]^d} \|\psi_i\|_{L^2(\Omega_i)}) \\
&\leq C \left(\frac{1}{H} \|w\|_{L^2(\Omega_i)} + |w|_{H^1(\Omega_i)} \right).
\end{aligned}$$

Therefore, owing to the bounded intersection property of subdomains Ω_i , the approximation property (3.13) and the energetic stability (3.14),

$$\begin{aligned}
\sum_{i=0}^J |u_i|_{H^1(\Omega)}^2 &\leq C \left(\sum_{i=1}^J \left(\frac{1}{H^2} \|w\|_{L^2(\Omega_i)}^2 + |w|_{H^1(\Omega_i)}^2 \right) + |u_0|_{H^1(\Omega)}^2 \right) \\
&\leq C \left(\frac{1}{H^2} \|w\|_{L^2(\Omega)}^2 + |w|_{H^1(\Omega)}^2 + |u_0|_{H^1(\Omega)}^2 \right) \\
&\leq C \left(\frac{1}{H^2} \|(I-Q)u\|_{L^2(\Omega)}^2 + |(I-Q)u|_{H^1(\Omega)}^2 + |u_0|_{H^1(\Omega)}^2 \right) \\
&\leq C (|u|_{H^1(\Omega)}^2 + |Qu|_{H^1(\Omega)}^2) \\
&\leq C |u|_{H^1(\Omega)}^2,
\end{aligned}$$

which completes the proof of the second statement of this Lemma. \square

To complete the proof of Theorem 3.1.3, we need the following two abstract results.

Lemma 3.1.8 ([5]). Let V be a Hilbert space with an inner product $a(\cdot, \cdot)$, and V_i denote subspaces of V , $V = \bigcup_{i=0}^J V_i$, with inner product $a(\cdot, \cdot)$. Further let operators $P_i : V \rightarrow V_i$ be $a(\cdot, \cdot)$ -orthogonal projectors. Then if there exists a constant $C_L > 0$ such that

$$\forall v \in V \quad \exists v_i \in V_i : v = \sum_{i=0}^J v_i \quad \text{and} \quad \sum_{i=0}^J a(v_i, v_i) \leq C_L a(v, v),$$

then

$$\inf \sigma \left(\sum_{i=0}^J P_i \right) \geq \frac{1}{C_L}.$$

The following lemma is a straightforward simplification of [89, Theorem 3.2] suitable for our purposes.

Lemma 3.1.9. Let

(i) There exists a constant C_L such that

$$a(v, v) \leq C_L a\left(\sum_{i=0}^J P_i v, v\right) \quad \forall v \in V.$$

(ii) Let $\varepsilon = \{\varepsilon_{ij}\}_{i,j=1}^J$ be a symmetric matrix such that

$$a(P_i u, P_j v) \leq \varepsilon_{ij} a(P_i u, u)^{1/2} a(P_j v, v)^{1/2} \quad \forall u, v \in V, \quad i, j = 1, \dots, J.$$

Then the product algorithm with error propagation operator

$$(I - P_0)(I - P_1) \dots (I - P_J)$$

is convergent with the rate bounded by

$$\gamma = 1 - \frac{1}{C_L(1 + \varrho(\varepsilon))^2}.$$

We can now return to proving Theorem 3.1.3. The assumption (i) of Lemma 3.1.9 follows from Lemmas 3.1.7 and 3.1.8. From bounded overlaps property of subdomains Ω_i we obtain

$$\varrho(\varepsilon) \leq C,$$

where the constant C is independent of the numbering of spaces V_i , $i = 1, \dots, J$.

Therefore, Lemma 3.1.9 yields

$$\|(I - P_0) \prod_{i=1}^{n_c} \prod_{j \in C_i} (I - P_j)\|_A \leq 1 - C,$$

where $\{C_i\}_{i=1}^{n_c}$ are decomposition sets introduced in at the beginning of this section. Now, the proof of Theorem 3.1.3 follows from (3.5).

3.2 Smoothed Aggregation Coarse Space and BOSS

In this section we define a coarse-space based on the concept of smoothed aggregation introduced in [87]. Overlapping subdomains will be defined based on the nonzero structure of prolongator. The method described here allows black-box implementation; its only input is a system of linear algebraic equations $Ax = f$ and the system of aggregates of degrees of freedom. Assumptions on aggregates allowing the proof of uniform convergence will be given in the next section.

Let $\{\mathcal{A}_i\}_{i=1}^J$ be a given system of aggregates of nodes forming a disjoint covering of the set of all unconstrained nodes i.e.

$$\bigcup_{i=1}^J \mathcal{A}_i = \{1, \dots, n\}, \quad \mathcal{A}_i \cap \mathcal{A}_j = \emptyset \text{ for } i \neq j.$$

We define a vector $\mathbf{1}_i \in \mathbb{R}^n$ as follows:

$$(\mathbf{1}_i)_j = \begin{cases} 1 & \text{for node } j \in \mathcal{A}_i \\ 0 & \text{elsewhere.} \end{cases} \quad (3.24)$$

Then we define the tentative prolongator \hat{P} to be an n by J matrix such that its i -th column is equal to $\mathbf{1}_i$.

Note that this grouping of nodes into disjoint sets and subsequent identification of each set with a single degree of freedom on the coarse space is referred to in the literature as aggregation technique. It was introduced in the early 1950's by Leontief [58] and frequently used in the problems of economic modeling (cf., Mandel and Sekerka [68] and the references therein.)

In order to eliminate oscillatory components from the range of \hat{P} , we introduce an n by n prolongator smoother \mathcal{S} and define the final prolongator P

by

$$P = \mathcal{S}\hat{P}. \quad (3.25)$$

The following algorithm describes construction of the polynomial prolongator smoother \mathcal{S} suitable for our purpose. The key property of the resulting smoother is that $\varrho(\mathcal{S}^2 A) \leq C \deg(\mathcal{S})^{-2} \varrho(A)$, which allows to prove that the H^1 -seminorm of our coarse-space basis functions is sufficiently small (Lemmas 3.3.2, 3.3.3). Note that there is no need to physically construct the prolongator smoother \mathcal{S} . For the sake of the method's implementation, we need only the final prolongator $P = \mathcal{S}\hat{P}$, which can be generated in $O(n \deg(\mathcal{S}))$ operations in a single processor environment.

Algorithm 6. For a desired degree $d_{\mathcal{S}}$ of the prolongator smoother \mathcal{S} and an estimate of the spectral radius of A such that

$$\varrho(A) \leq \hat{\varrho} \leq C_{\varrho} \varrho(A), \quad (3.26)$$

we define the prolongator smoother by

1. Let $K = \lfloor \log_3(2d_{\mathcal{S}} + 1) \rfloor - 1$, where $\lfloor \cdot \rfloor$ is the truncation to the nearest smaller integer.
2. For $i > 0$, set $\hat{\varrho}_i = \frac{\hat{\varrho}}{9^i}$. and compute $\mathcal{S} = \prod_{j=0}^K (I - \frac{4}{3} \hat{\varrho}_j^{-1} A_j)$, where A_j is defined by the recurrence formula

$$\begin{aligned} A_0 &= A, \\ A_j &= (I - \frac{4}{3} \hat{\varrho}_{j-1}^{-1} A_{j-1})^2 A_{j-1}, \quad \text{for } j > 0. \end{aligned} \quad (3.27)$$

Remark 3.2.1. Algorithm 6 is capable of generating smoother \mathcal{S} of certain degrees only. The choice of K in the step 1. gives \mathcal{S} of degree closest to the desired one $d_{\mathcal{S}}$, see (3.29).

If we set for $i \in \mathbb{N}$

$$S_i = \prod_{j=1}^i \left(I - \frac{4}{3} \hat{\theta}_{j-1}^{-1} A_{j-1} \right),$$

then for the prolongator smoother \mathcal{S} created by Algorithm 6 we have $\mathcal{S} = S_{\lfloor \log_3(2d_{\mathcal{S}}+1) \rfloor} \cdot A_{\mathcal{S}}$

$$\deg(A_i) = \deg\left(\left(I - \frac{4}{3} \hat{\theta}_{i-1}^{-1} A_{i-1}\right)^2 A_{i-1}\right) = 3\deg(A_{i-1}) = 3^i,$$

we get

$$\begin{aligned} \deg(S_i) &= \deg(S_{i-1}) + \deg(A_{i-1}) = \deg(S_{i-1}) + 3^{i-1} \\ &= \sum_{j=0}^{i-1} 3^j = \frac{3^i - 1}{2}. \end{aligned} \quad (3.28)$$

Therefore, $\mathcal{S} = S_{\lfloor \log_3(2d_{\mathcal{S}}+1) \rfloor}$ satisfies

$$\frac{d_{\mathcal{S}}}{3} < \deg(\mathcal{S}) \leq d_{\mathcal{S}}. \quad (3.29)$$

The nonzero structure of the prolongator P determines the supports of our coarse space basis functions

$$\Phi_i = \Pi P e_i. \quad (3.30)$$

Here, $\Pi x = \sum_{i=1}^n x_i \phi_i$ is the finite element interpolator, $\{\phi_i\}$ the finite element basis and e_i the i -th vector of the canonical basis.

The prolongator P is obtained as a result of the matrix multiplication $P = \mathcal{S} \hat{P}$, where \hat{P} is the tentative prolongator and \mathcal{S} is the polynomial in the stiffness matrix A given by Algorithm 6.

The computational subdomains Ω_i are derived from the nonzero structure of the matrix $P^{\text{sy mb}}$, which is obtained in the same way as the prolongator P , except the matrix operations involved are performed only symbolically. That is, we replace nonzero entries of matrices by ones and use the arithmetic

$$1 + \alpha = 1, \quad 0 + \alpha = \alpha, \quad 1 * \alpha = \alpha, \quad 0 * \alpha = 0, \quad \text{for } \alpha = 0, 1.$$

Then, defining

$$\Omega_i = \text{supp}(\Pi P^{\text{sy mb}} e_i), \quad (3.31)$$

we have

$$\text{supp}(\Phi_i) \subset \Omega_i. \quad (3.32)$$

Note that if the sparse matrix operations are implemented so that numerical zeroes are never dropped, the results of these symbolic operations are obtained for free as a side benefit of the computation, so the symbolic operations need not be performed at all.

Algorithmically, this can be accomplished as follows: First, for each column of the smoothed prolongator $P^{\text{sy mb}}$ let us define the list of its nonzeros

$$\mathcal{N}_j = \{i : P_{ij}^{\text{sy mb}} \neq 0\}, \quad n_j = \text{card}(\mathcal{N}_j)$$

and the n by n_j 0–1 matrix N_j resulting from selecting the columns with indices in \mathcal{N}_j from the n by n identity matrix. Further we define local matrices \tilde{A}_i and local correction operators R_i

$$\tilde{A}_i = N_i^T A N_i, \quad R_i = N_i (\tilde{A}_i)^{-1} N_i^T, \quad i = 1, \dots, J. \quad (3.33)$$

Analogously, for the coarse level we set

$$\tilde{A}_0 = P^T A P, \quad R_0 = P (\tilde{A}_0)^{-1} P^T. \quad (3.34)$$

1	2	1	2	1	2
3	4	3	4	3	4
1	2	1	2	1	2
3	4	3	4	3	4
1	2	1	2	1	2

Figure 3.1: Possible assignment of elements to different classes \mathcal{C}_i in 2D.

For a positive i , $R_i A$ is the A -orthogonal projection onto the local space

$$\hat{V}_i = \{x \in \mathbb{R}^n : x_j = 0 \text{ for } j \notin \mathcal{N}_i\}.$$

Note that $(\hat{V}_i, \|\cdot\|_A)$ is the vector space isometrically isomorphic to the space of finite element functions $(V_i \equiv \{\sum_{i=1}^n x_i \varphi_i, x \in \hat{V}_i\}, a(\cdot, \cdot)^{1/2})$. Also, V_i introduced this way satisfies (3.3).

For the sake of parallelism, we need a disjoint covering $\{\mathcal{C}_i\}_{i=1}^{n_c}$ of the set $\{1, \dots, J\}$ satisfying

$$\cos(\hat{V}_j, \hat{V}_k) = 0 \text{ for every } j, k \in \mathcal{C}_i, i = 1, \dots, n_c, \quad (3.35)$$

where the cosine is measured in A -inner product. For structured meshes, such a covering can easily be obtained. For simplicity of demonstration, consider 2D and 3D rectangular meshes. Figures 3.2 and 3.2 depict possible decomposition into noncontiguous groups forming \mathcal{C}_i in 2D and 3D, respectively.

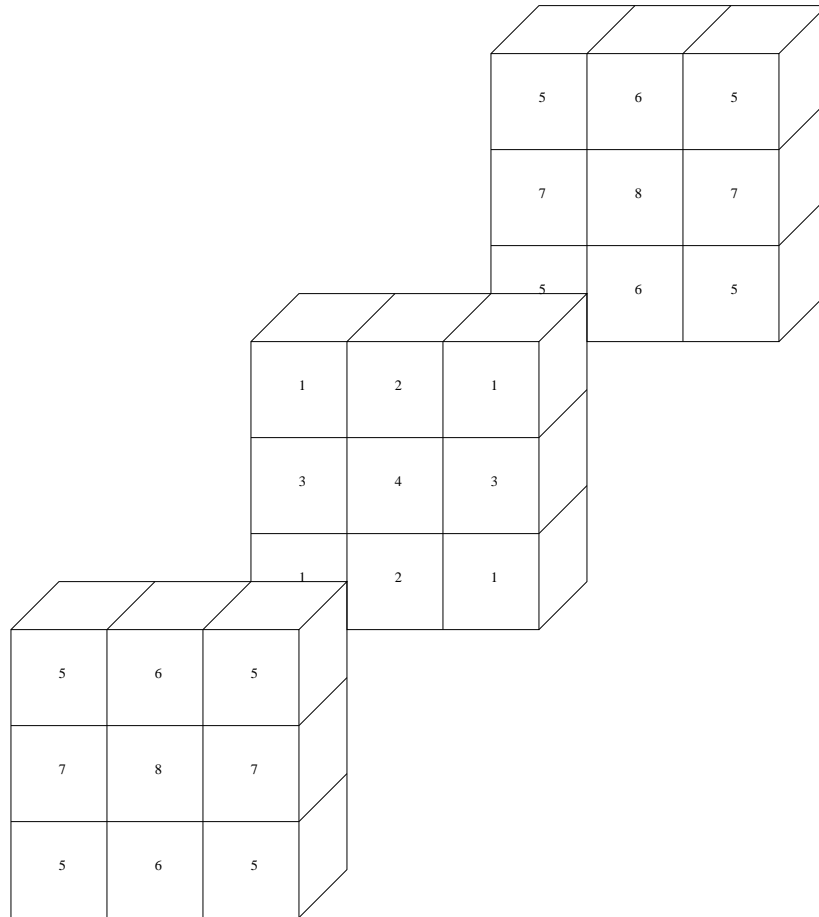


Figure 3.2: Possible assignment of elements to different classes \mathcal{C}_i in 3D.

We can see that for the rectangular meshes the elements can be subdivided into four noncontiguous groups in 2D and into eight in 3D. We are, however, seeking a similar decomposition for any (unstructured) mesh. For unstructured meshes, such a decomposition can be created using a simple greedy algorithm, as the information about the orthogonality of spaces \hat{V}_i is available. Trivially, spaces \hat{V}_i and \hat{V}_j are orthogonal if

$$a_{kl} = 0 \text{ for all } k \in \mathcal{N}_i, l \in \mathcal{N}_j,$$

where a_{kl} are entries of the stiffness matrix A . Such a test can be easily performed using the formula

$$A_c^{\text{symb}} = P^{\text{symb}T} * A^{\text{symb}} * P^{\text{symb}},$$

where $*$ is the operation of the symbolic matrix multiplication. Then,

$$\text{If } (A_c^{\text{symb}})_{ij} = 0 \text{ then } \cos(\hat{V}_i, \hat{V}_j) = 0.$$

The disjoint covering $\{\mathcal{C}_i\}_{i=1}^{n_c}$ of the set of coarse space degrees of freedom satisfying (3.35) can be created using the following algorithm:

Algorithm 7. Set $\mathcal{R} = \{1, \dots, J\}$, $i = 0$ and

1. repeat
2. set $i \leftarrow i + 1$,
3. set $\mathcal{R}_i = \mathcal{R}$, $\mathcal{C}_i = \emptyset$,
4. repeat
5. choose $j \in \mathcal{R}_i$,
6. set $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{j\}$,
7. for each $k : (A_c^{\text{symb}})_{jk} = 1$ set $\mathcal{R}_i \leftarrow \mathcal{R}_i \setminus \{k\}$,

8. until \mathcal{R}_i is empty,
9. set $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{C}_i$,
10. until \mathcal{R} is empty,
11. set $n_c = i$.

Now we have all the components needed to write down the implementation of the Schwarz method with the error propagation operator (3.4).

Algorithm 8 (BOSS). Given a vector x^i , the method returns x^{i+1} computed as follows:

1. Set $z^0 = x^i$.
2. Local corrections:
for $i = 1, \dots, n_c$ do

$$z^i = z^{i-1} + \sum_{j \in \mathcal{C}_i} R_j d^i, \text{ where } d^i = f - Az^{i-1}$$

and R_j is the correction operator defined in (3.33).

3. Coarse level correction:

$$z^0 = z^{n_c} + R_0(f - Az^{n_c}),$$

where R_0 is the coarse-level correction operator given by (3.34).

5. (optional) for $i = n_c, \dots, 1$ do

$$z^i = z^{i-1} + \sum_{j \in \mathcal{C}_i} R_j d^i, \text{ where } d^i = f - Az^{i-1}$$

6. Set $x^{i+1} = z^0$.

Note that if the optional post-smoothing step 5 is used, the algorithm can be used as a symmetric preconditioner in the conjugate gradient method.

Remark 3.2.2. For the sake of brevity, we denote the method described by Algorithm 8 with the choice of operators R_i, R_0 given by (3.33), (3.34) as BOSS, short for Black-box Overlapping Schwarz with Smoothed Coarse Space. We will find this abbreviation useful in the tables of Chapter 6.

3.3 Estimates for Smoothed Aggregation

In this section, we apply the general estimates to smoothed aggregation method defined in previous section. In order to prove the convergence of the method with the coarse space generated by smoothed aggregations, we only have to verify that Assumptions 3.1.2 and 3.1.1 are satisfied.

The pattern of the stiffness matrix $A = \{a_{ij}\}_{i,j=1}^n$ determines the undirected graph

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\},$$

where vertices $\mathcal{V} = \{1, \dots, n\}$ are indices of all unconstrained nodes and edges \mathcal{E} are given by

$$\mathcal{E} = \{[i, j] \in \mathcal{V} \times \mathcal{V} : a_{ij} \neq 0\}.$$

For $i \in \mathcal{V}$ and a positive integer r let us define the graph r -neighborhood of i by

$$\mathcal{B}(i, r) = \{j \in \mathcal{V} : \text{dist}(i, j) \leq r\}.$$

Here, the distance of two vertices i, j is the minimal length of path connecting i, j measured in the number of edges forming the path.

In the rest of this section, we will prove the optimal convergence result under the following assumption on the system of aggregates $\{\mathcal{A}_i\}_{i=1}^J$. The first part of the assumption controls aspect ratios of aggregates. The second

part specifies the number of smoothing steps involved in the construction of the prolongator smoother.

Assumption 3.3.1. There are positive integer constants c, C, C_1, C_2 and a positive integer α characterizing the graph size of aggregates such that

a) In each aggregate \mathcal{A}_i there is a node j satisfying

$$\mathcal{B}(j, c\alpha) \subset \mathcal{A}_i, \quad \text{and } \text{dist}(k, j) \leq C\alpha \text{ for every } k \in \mathcal{A}_i.$$

b) For the degree d_S of prolongator smoother it holds that

$$C_1\alpha \leq d_S \leq C_2\alpha.$$

The decomposition satisfying the Assumption 3.3.1 can be easily generated using a simple greedy algorithm, see Algorithm 9 in the Section 3.4.

The following simple algebraic result is the key tool needed for verification of the Assumption 3.1.2 a).

Lemma 3.3.2. For the prolongator smoother \mathcal{S} created by Algorithm 6 it holds that

$$\varrho(\mathcal{S}^2 A) \leq C \deg(\mathcal{S})^{-2} \varrho(A) \quad \text{and } \varrho(\mathcal{S}) \leq 1.$$

Proof. ([88]). Throughout this proof, we use the notation introduced in the Algorithm 6. First, by induction, we prove $\varrho(A_i) \leq \hat{\varrho}_i$. For $i = 0$, the inequality holds by (3.26); assume it holds for $j \leq i$. Then, by

$$A_{i+1} = (I - \frac{4}{3} \hat{\varrho}_i^{-1} A_i)^2 A_i$$

and the inductive assumption

$$\varrho(A_{i+1}) = \max_{t \in \sigma(A_i)} (1 - \frac{4}{3} \hat{\varrho}_i^{-1} t)^2 t \leq \max_{t \in [0, \hat{\varrho}_i]} (1 - \frac{4}{3} \hat{\varrho}_i^{-1} t)^2 t \leq \frac{1}{9} \hat{\varrho}_i = \hat{\varrho}_{i+1}.$$

Hence,

$$\varrho(A_i) \leq \left(\frac{1}{9}\right)^i \hat{\varrho} \equiv \hat{\varrho}_i.$$

Now, the second estimate $\varrho(\mathcal{S}) \leq 1$ follows immediately from the fact that \mathcal{S} is a product of terms of the form $I - \frac{4}{3}\hat{\varrho}_j^{-1}A_j$.

Let us estimate the spectral bound of \mathcal{S}^2A . It is routine to verify that

$$\mathcal{S}^2A = A_K, \quad \text{where } K = \lfloor \log_3(2d_{\mathcal{S}} + 1) \rfloor - 1.$$

Therefore, taking into account that (see Remark 3.2.1) $3^K \geq Cd_{\mathcal{S}} \geq C\deg(\mathcal{S})$ and $\hat{\varrho}_0 \equiv \hat{\varrho} \leq C_{\varrho}\varrho(A)$ (see (3.26)), we get

$$\varrho(\mathcal{S}^2A) \leq \left(\frac{1}{9}\right)^K \hat{\varrho}_0 \leq C\deg^{-2}(\mathcal{S})\varrho(A),$$

which was to be proved. \square

The following lemma demonstrates validity of Assumption 3.1.2, a).

Lemma 3.3.3. Under the Assumption 3.3.1, for coarse space basis functions defined by (3.30) it holds that

$$|\Phi_i|_{H^1(\Omega)} \leq CH^{\frac{d-2}{2}} \quad \text{and} \quad \|\Phi_i\|_{L^2(\Omega)} \leq CH^{\frac{d}{2}}, \quad (3.36)$$

where $H = \alpha h$.

Proof. Taking into account the underlying quasiuniform $P1$ or $Q1$ finite element mesh, the number α characterizes the “discrete diameter” of aggregates \mathcal{A}_i , and we have

$$\text{card}(\mathcal{A}_i) \leq \alpha^d.$$

Further, due to the Lemma 3.3.2,

$$\varrho(\mathcal{S}^2A) \leq \frac{C}{\alpha^2}\varrho(A).$$

Therefore, using the fact that $\varrho(A) \leq Ch^{d-2}$,

$$\begin{aligned}
|\Phi_i|_{H^1(\Omega)}^2 &\leq Ca(\Pi\mathcal{S}\hat{P}e^i, \Pi\mathcal{S}\hat{P}e^i) \\
&= C\langle A\mathbf{S}\mathbf{1}_i, \mathbf{S}\mathbf{1}_i \rangle \\
&\leq C\alpha^{-2}\varrho(A) \text{card}(\mathcal{A}_i) \\
&\leq C\alpha^{d-2}h^{d-2} \\
&= CH^{d-2}.
\end{aligned}$$

Similarly, using the fact that $\varrho(\mathcal{S}) \leq 1$ (Lemma 3.3.2),

$$\begin{aligned}
\|\Phi_i\|_{L^2(\Omega_j)}^2 &= \|\Pi\mathcal{S}\hat{P}e^i\|_{L^2(\Omega)}^2 \\
&\leq Ch^d\langle \mathbf{S}\mathbf{1}_i, \mathbf{S}\mathbf{1}_i \rangle \\
&\leq Ch^d\varrho(\mathcal{S}^2) \text{card}(\mathcal{A}_i) \\
&\leq Ch^d\alpha^d \\
&\leq CH^d,
\end{aligned}$$

completing the proof. \square

Remark 3.3.4. The estimate for the smoothed functions Φ_i in Lemma 3.3.3 is a significant improvement over the case of unsmoothed functions $\Pi\mathbf{1}_i$, for which we can only prove $|\Pi\mathbf{1}_i|_{H^1(\Omega)}^2 \leq CH^{d-1}/h$.

Now we are ready to complete the verification of Assumption 3.1.2 for smoothed aggregations.

Lemma 3.3.5. Under Assumption 3.3.1, the coarse-space basis $\{\Phi_i\}$ generated by smoothed aggregation technique described in the previous section satisfies Assumption 3.1.2.

Proof. The assumption a) follows from the Lemma 3.3.3. The assumption c) has been verified in the previous section, see (3.32). Let us prove b). Basis functions derived from the tentative prolongator \hat{P} , i.e.

$$\hat{\Phi}_i = \Pi \hat{P} e^i, \quad i = 1, \dots, m$$

satisfy the decomposition of unity

$$\sum_{i=1}^m \hat{\Phi}_i = 1 \tag{3.37}$$

everywhere on $\Omega \setminus B_{\Gamma_D}$, where B_{Γ_D} is the union of elements \mathcal{T}_i such that

$$\partial \mathcal{T}_i \cap \Gamma_D \neq \emptyset.$$

Discretely, (3.37) holds in every unconstrained finite element nodal point. Let \mathcal{D} be the index set of all finite element nodal points $v_i \in B_{\Gamma_D}$ (B_{Γ_D} is understood as a closed domain.) The vector of units is a local kernel of the matrix A . More precisely, for a vector of ones, $u \in \mathbb{R}^n$, it holds that

$$(Au)_i = 0 \quad \text{for every } i \notin \mathcal{D}.$$

Further, for a positive k and the vector of units u ,

$$(A^k u)_i = 0 \quad \text{for every } i \text{ such that } \text{dist}(i, \mathcal{D}) \geq k + 1,$$

where dist is a graph distance introduced at the beginning of this section. The prolongator smoother \mathcal{S} is a polynomial in the stiffness matrix A with the absolute term equal to 1. Therefore, for the vector

$$w = \sum_{i=1}^m P e^i = \sum_{i=1}^m \mathcal{S} \hat{P} e^i = \mathcal{S} u$$

we have

$$w_i = 0 \text{ for every } i \text{ such that } \text{dist}(i, \mathcal{D}) \geq \text{deg}(\mathcal{S}).$$

As $\sum \Phi_i = \Pi \mathcal{S}u$, the decomposition of unity $\sum_{i=1}^m \Phi_i = 1$ is violated at most at $\text{deg}(\mathcal{S}) + 1$ strips of elements surrounding Γ_D . This, together with $\text{deg}(\mathcal{S}) \leq C\alpha$ and $H = \alpha h$ completes the proof. \square

Lemma 3.3.6. Under Assumption 3.3.1, the computational subdomains Ω_i defined by (3.31) satisfy Assumption 3.1.1.

Proof. The proof consists of simple, but rather tedious geometrical considerations. Computational subdomains are defined by $\Omega_i = \text{supp}(\Pi \mathcal{S}^{\text{symb}} * \hat{P} * e_i)$, where $*$ is the operation of the symbolic matrix multiplication and $\mathcal{S}^{\text{symb}}$ is the polynomial in A created using symbolic matrix operations too. The degree of $\mathcal{S}^{\text{symb}}$ satisfies

$$c\alpha \leq \text{deg}(\mathcal{S}^{\text{symb}}) \leq C\alpha.$$

Further, the support of basis function derived from the tentative prolongator

$$\text{supp}(\Pi \hat{P} e_i)$$

is formed by all elements \mathcal{T}_j such that at least one vertex of \mathcal{T}_j belongs to the aggregate \mathcal{A}_i . The smoothing by $\mathcal{S}^{\text{symb}}$ adds $\text{deg}(\mathcal{S}^{\text{symb}})$ layers of surrounding finite elements.

Taking into account the quasiuniformity of the underlying mesh and the fact that $H = \alpha h \approx \text{deg}(\mathcal{S}^{\text{symb}})h$, the measure of added $\text{deg}(\mathcal{S}^{\text{symb}})$ layers of elements itself is greater or equal to CH^d . So,

$$\text{meas}(\Omega_i) \geq CH^d,$$

which proves the Assumption 3.1.1, d). Also, due to Assumption 3.3.1, a), we similarly obtain $\text{diam}(\Omega_i) \leq CH$. Hence a) is also verified.

Let us prove b). The supports of basis functions $\Pi\hat{P}e_i$ cover the domain Ω . As Ω_i is created by adding $\text{deg}(\mathcal{S}) \geq c\alpha$ layers of elements to $\text{supp}(\Pi\hat{P}e_i)$, we have

$$\text{dist}_{\mathbb{R}^d}(x, \partial\Omega_i) \geq cH \quad \forall x \in \text{supp}(\Pi\hat{P}e_i),$$

where $\text{dist}_{\mathbb{R}^d}(\cdot, \cdot)$ is the Euclidean distance. As every point $x \in \Omega$ belongs to some $\text{supp}(\Pi\hat{P}e_i)$, b) is proved.

It remains to verify c). The first part of the Assumption 3.3.1, a) says that each aggregate \mathcal{A}_i contains a “graph ball” of a radius $r \geq c\alpha$, where c is the positive integer constant. Let us interpret this assumption geometrically in \mathbb{R}^d .

For each aggregate of vertices \mathcal{A}_i let us define the cluster C_i consisting of all finite elements \mathcal{T}_j so that all vertices of \mathcal{T}_j belong to \mathcal{A}_i . From the first part of the Assumption 3.3.1, a), it follows that there is a ball $B_i \subset C_i$ such that $\text{diam}(B_i) \geq cH$. As the aggregates \mathcal{A}_i are disjoint, the clusters C_i and balls B_i are disjoint as well. Summing up, we have proved the following properties for subdomains Ω_i :

- $\text{diam}(\Omega_i) \leq CH$.
- For each Ω_i there is a ball $B_i \subset \Omega_i$ such that $\text{diam}(B_i) \geq cH$ and balls B_i , $i = 1, \dots, J$, are mutually disjoint.

From here, the Assumption 3.1.1, c) follows. \square

We have verified that Assumptions 3.1.2 and 3.1.1 are satisfied, hence we can apply Theorem 3.1.3 to prove convergence of the method with coarse space given by smoothed aggregations.

Theorem 3.3.7. Let the Assumption 3.3.1 holds. Then, for the error propagation operator T of the method described in the Section 3.2 applied to the model problem (3.2) it holds that

$$\|T\|_A \leq 1 - C,$$

where C is a constant independent of h and the size of aggregates.

Proof. The proof follows immediately from Lemmas 3.3.5, 3.3.6 and Theorem 3.1.3. \square

3.4 Practical Issues

The overlapping method with the coarse space given by smoothed aggregations has very favorable convergence properties common to most overlapping Schwarz methods with a coarse space. The new method has, however, certain advantages over the existing overlapping methods. The advantages of the smoothed aggregation approach include: It can be implemented as a black box with no input required from the user except for the stiffness matrix and the right hand side of the problem. Even though the analysis assumes an elliptic functional discretized on a quasi-uniform mesh by P1 or Q1 finite elements, numerical experiments confirm applicability of the method to unstructured meshes and a variety of finite element types and problems far beyond the scope of the current theory.

The disadvantage common to all overlapping-type domain decomposition methods is the increase of computational complexity with increasing measure of the overlap. Our method cannot avoid this drawback, but the use of “coloring” in the definition (3.4) allows parallel implementation of local solves

and reduces the processing time.

In the rest of this section, we discuss ways to generalize the method for solving nonscalar problems. We also analyze the computational complexity of the method and a practical algorithm that can be used to generate the aggregates \mathcal{A}_i .

3.4.1 Generation of Aggregates

We now describe a greedy algorithm which will generate the subdomains satisfying the Assumption 3.3.1. First we extend the definition of graph neighborhood of a node to the graph neighborhood of a set $X \subset \{1, \dots, n\}$ of nodes:

$$\mathcal{B}(X, \alpha) = \{i : \text{dist}(i, X) \leq \alpha\}.$$

With this definition, we can write the

Algorithm 9. For the given stiffness matrix A and positive integer α , create the system of aggregates $\{\mathcal{A}_i\}$ as follows:

1. Set $\mathcal{R} = \{1, \dots, n\}$, $j = 0$.
2. for $i = 1, \dots, n$ do
3. if $i \in \mathcal{R}$ then
4. if $\mathcal{B}(\{i\}, \alpha) \subset \mathcal{R}$ then
5. $j \leftarrow j + 1$,
6. $\mathcal{A}_j \leftarrow \mathcal{B}(\{i\}, \alpha)$,
6. $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{A}_j$,
7. end if
8. end if

9. end for
10. set $J = j$,
11. for $i = 1, \dots, J$
12. $\mathcal{A}_i \leftarrow \mathcal{A}_i \cup (\mathcal{B}(\mathcal{A}_i, \alpha) \cap \mathcal{R})$,
13. $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{A}_i$
14. end for

In order to fully complete the aggregate generation description, we give an algorithmic recipe for computing the α -neighborhood of a set X used in Algorithm 9

Algorithm 10. Given a set $X \subset \{1, \dots, n\}$,

1. Set $w \in \mathbb{R}^n$ as $w_i = \begin{cases} 1 & \text{if } i \in X \\ 0 & \text{otherwise.} \end{cases}$
2. Set $w = A^\alpha * w$ (both the power and the multiplication are performed symbolically.)
3. Set $\mathcal{B}(X, \alpha) = \{i : w_i = 1\}$.

Remark 3.4.1. Algorithm 9 generates a disjoint covering $\{\mathcal{A}_i\}$ of the set of all vertices that satisfies Assumption 3.3.1. In steps 1.–10. Algorithm 9 generates graph neighborhoods $\mathcal{A}_i = \mathcal{B}(\{j\}, \alpha)$. After Step 10., the set \mathcal{R} contains the remaining nodes that could not be made into whole α -neighborhoods. For these nodes it holds that

$$\forall j \in \mathcal{R} \quad \exists \mathcal{A}_i \quad \text{such that} \quad \text{dist}(j, \mathcal{A}_i) \leq \alpha.$$

Steps 11. through 13. add at most α “layers” of surrounding vertices to some of the aggregates \mathcal{A}_i . It follows from the construction that at the end of Algorithm 9, $\mathcal{R} = \emptyset$.

3.4.2 Nonscalar Problems

The method can easily be modified for solving nonscalar problems. We will briefly describe the changes required. This approach first appeared in [87] in the context of solving problems of order higher than 2. In order to apply the method to nonscalar problems, we need the knowledge of the discrete representation of the local kernel of the bilinear form $a(\cdot, \cdot)$. By local kernel we mean the kernel in absence of essential boundary conditions, i.e., the kernel of the unconstrained problem.

Let us assume that we have functions $\{f^j\}_{j=1}^{n_k}$ spanning the local kernel of $a(\cdot, \cdot)$ (in case of 3D elasticity, 6 rigid body modes). For each function f^i , we need its discrete representation with respect to our finite element basis, or the vector \hat{f}^i such that

$$f^i = \Pi \hat{f}^i.$$

Finite element packages usually provide this information. For every aggregate \mathcal{A}_i let us define the set \mathcal{D}_i of all degrees of freedom associated with nodes of \mathcal{A}_i . Then the tentative prolongator can be constructed by

Algorithm 11 (Tentative prolongator - nonscalar problems).

For every aggregate \mathcal{A}_i and for $j = 1, \dots, n_k$:

1. For \hat{f}^j , compute the vector $\hat{f}^{ij} \in \mathbb{R}^n$ with components

$$\hat{f}_k^{ij} = \begin{cases} \hat{f}_k^j & \text{if } k \in \mathcal{D}_i \\ 0 & \text{otherwise.} \end{cases}$$

2. Interpret the vector \hat{f}^{ij} as the $n_k(i - 1) + j$ -th column of the tentative prolongator \hat{P} .

The algorithm in this form can be used to treat quite general bases (e.g., unscaled bases, high order elements or common plate and shell elements). In order to improve conditioning of the coarse problem, it is advisable to perform the discrete l^2 -orthogonalization of vectors \hat{f}^{ij} on each aggregate \mathcal{A}_i , as suggested in Vaněk, Mandel and Brezina [87], or by simply computing their l^2 -orthogonal projections onto complement of a constant. This is not required by the theory, but practical applications can benefit from such a stabilization. When solving problems of 2D and 3D elasticity it is possible to obtain the basis $\{f^j\}$ explicitly for each subdomain without having to scale it; this construction relies on the fact that the rigid body modes are known and was presented in [83]. The above mentioned approach is, however, more general.

3.4.3 Computational Complexity

We will now give an asymptotic bound on the amount of floating point operations needed to carry out the iteration to reduce the error to the truncation level. We will give the estimates for implementation on both serial and parallel architectures.

Let N_{es} denote the typical number of elements per subdomain d the dimension of the space on which the continuous problem is cast, and n the number of degrees of freedom in the whole system.

Let us first compute the amount of work needed for the setup. On a machine with a single CPU, we need $O(\text{deg}(\mathcal{S})n)$ operations to compute the prolongator $P = \mathcal{S}\hat{P}$. Taking into account that $\text{deg}(\mathcal{S}) \approx \frac{H}{h} \approx N_{es}^{1/d}$, this becomes $O(N_{es}^{1/d}n)$. Further, we need $O(\frac{n}{N_{es}}N_{es}^{\frac{3d-2}{d}})$ and $O((\frac{n}{N_{es}})^{\frac{3d-2}{d}})$ operations

to compute the Cholesky factorizations of the local and coarse level matrices, respectively. We also need $O(n)$ operations to compute the coarse level matrix, but this number can be taken out of the consideration, as it is dominated by the other expenditures.

Each step of the iteration requires $O(\frac{n}{N_{es}}N_{es}^{\frac{2d-1}{d}})$ and $O((\frac{n}{N_{es}})^{\frac{2d-1}{d}})$ operations to compute the back-substitutions in the local and coarse spaces, respectively. The amount of work required to compute the defect, the corrections and restriction is $O(n)$, hence negligible.

Taking into account all the above listed expenditures, we use trivial calculus to conclude that the optimal value of the number of elements per sub-domain is $N_{es} = n^{\frac{2d-2}{5d-4}}$. That is, $N_{es}^{opt} = n^{\frac{1}{3}}$ for 2D problems and $N_{es}^{opt} = n^{\frac{4}{11}}$ for 3D problems. The total amount of work involved in the setup and iterations for these optimal values is $O(n^{\frac{4}{3}})$ and $O(n^{\frac{49}{33}})$ in 2D and 3D, respectively.

The reason we introduced the “coloring” classes C_i in the algorithm was to facilitate the use of modern parallel architecture computers. For simplicity, we assume that we have at least $\lceil n^{1/2} \rceil$ processors. Then most the procedures can take advantage of parallel implementation. In the evaluations of computational work we omit all operations costing $O(n)$ operations.

The setup will require $O(\deg(\mathcal{S})n^{1/2})$ operations to compute $P = \mathcal{S}\hat{P}$. If we assume that the local Cholesky decompositions are performed in parallel, we need $O(N_{es}^{\frac{3d-2}{d}})$ and $O((\frac{n}{N_{es}})^{\frac{3d-2}{d}})$ operations to compute the Cholesky factorizations of the local and coarse problems, respectively.

Each step of the iteration will require $O(N_{es}^{\frac{2d-1}{d}})$ and $O((\frac{n}{N_{es}})^{\frac{2d-1}{d}})$ operations to compute the back-substitutions in the local and coarse spaces, respectively.

Balancing these values, we obtain that the optimal size of a subdomain is about $n^{1/2}$ in both 2D and 3D. The resulting computational complexity can then be bounded by $O(n)$ in 2D and by $O(n^{7/6})$ in 3D.

The above discussion together with the convergence Theorem 3.1.3 proves the following theorem:

Theorem 3.4.2. Let Assumptions 3.1.2 and 3.1.1 be satisfied, and the Cholesky factorization be used to solve the coarse-level and local subdomain problems. Then, on a serial architecture, the optimal number of elements per subdomain is $N_{es}^{2D} \approx n^{\frac{1}{3}}$ in 2D and $N_{es}^{3D} \approx n^{\frac{4}{11}}$ in 3D, and the system (3.1) can be solved to the level of truncation error in $O(n^{\frac{4}{3}})$ operations in 2D, and $O(n^{\frac{49}{33}})$ operations in 3D. If a parallel architecture with $n^{1/2}$ processors were available, the optimal number of elements per subdomain would change to $N_{es}^{2D} = N_{es}^{3D} \approx n^{\frac{1}{2}}$, and the system (3.1) could be solved to the level of truncation error in $O(n)$ operations in 2D, and $O(n^{\frac{7}{6}})$ operations in 3D.

The above estimates show that the amount of work required to complete the whole iterative process (including its setup) is asymptotically lower than even just the back-substitution step of direct methods based on matrix factorization, which would be $O(n^{3/2})$ and $O(n^{5/3})$ in 2D and 3D, respectively.

4. Nonoverlapping Methods with Inexact Solvers

This chapter deals with the issue of nonoverlapping domain decomposition methods using only inexact subdomain solvers. In Section 4.2, we formulate requirements on the approximate solvers we will use and their properties. In the sections that follow we define a fully algebraic nonoverlapping domain decomposition method with inexact subdomain solvers and prove the condition number estimate of the resulting algorithm. We will also study the influence of using approximate coarse space problem.

4.1 Inexact Subdomain Solvers

The reason many domain decomposition methods use the reduced system is that the condition number of the problem to be solved is reduced from $1/h^2$ to $1/h$ (see Lemma A.1.7 or Bramble [9]). The resulting method is proved to behave better than applying the diagonal preconditioner to the original problem (Mandel [61]). However, solving the local problems by direct methods can be costly, especially if only a modest number of subdomains is used. Therefore, number of attempts to replace direct solvers in solving the subdomain problems by significantly cheaper iterative methods have been made. One of the earliest of these efforts is due to Börgers [8] who proved for the Neumann-Dirichlet domain decomposition and the case of two substructures that convergence independent

of the meshsize can be obtained if the exact local solvers are replaced by a small number of multigrid iterations.

Another approach was described for the Neumann-Neumann method in [32], where theory is based on the abstract Schwarz method framework recalled in Section 2.1. The method of [33] uses inexact solvers for the Neumann problems under the assumption that the local Schur complements are known. Because the local Schur complements define the global one, this approach is only appropriate in the case the global Schur complement S is computed. This, however, has to be avoided in order to keep the computational complexity down. An improved version of the same algorithm, using the formulation with local stiffness matrices $A^{(i)}$ instead, appeared in [30]. An elegant way of application of inexact local solvers can also be found there. The disadvantage of this method is its utilization of the finite element basis used for discretization of the problem.

One of the most recent attempts to address the issue of inexact subdomain solvers is due to Bramble, Pasciak and Vasiliev [13]. Their method is based on a trivial approximation of harmonic extensions, which results in reduced computational complexity, but allows only a suboptimal condition number estimate with linear dependence on the ratio $\frac{H}{h}$.

An attractive feature of domain decomposition methods used as preconditioners in the method of preconditioned conjugate gradients is that the matrix of the problem never has to be assembled. Although this is a general property common to all domain decomposition methods, it is perhaps most clearly demonstrated on the example of the EBE methods of Section 2.6, where (except for certain implementations) only local element submatrices need to be known and

stored. The action Ax of the global matrix, required by PCG, can be obtained by the subassembly of the actions of local matrices

$$Ax = \sum_{i=1}^J N_i^T A^{(i)} N_i x.$$

Thus, only the local matrices $A^{(i)}$ or their factors have to be stored.

An important prerequisite for the application of inexact solvers is reformulating the problem in such a way that the local Schur complements do not figure at all in the formulation of the problem to be solved. That is, special care has to be applied so that using inexact solvers does not affect the problem to be solved. For instance, if the local Schur complement matrices $S^{(i)}$ were straightforwardly replaced by their approximations $\tilde{S}^{(i)}$, the problem to be solved would be $\tilde{S}\tilde{x} = b$, the solution of which may be completely irrelevant to that of $Sx = b$. On the other hand, forming and storing exact Schur complements $S^{(i)}$ as well as their approximations does not seem to offer any advantage over storing only the Choleski factors of $S^{(i)}$, which is the common practice.

A proper formulation may be found in the early domain decomposition paper of Bramble, Pasciak, Schatz [10]. We give another suitable formulation in Section 4.5.2. Both are formulated in terms of the local stiffness matrices $A^{(i)}$; the main difference between the two is that the formulation in [10] is nonalgebraic. Another significant difference is that we are using a smaller and more efficient coarse space (with one degree of freedom per subdomain for scalar second order problems). We present two substructuring methods avoiding the exact subdomain solvers. First of them is an extended BDD. The second one, presented in Chapter 5, is more unorthodox and is closely related to the multigrid method with smoothed aggregation [52, 86, 87].

4.2 The Inexact Solvers' Properties

Our aim is to replace the exact solvers with approximate ones. In order to simplify the notation, in this section we describe the inexact solvers and state some of their properties used in our proofs.

Assume that all algebraic systems

$$Ax = f \tag{4.1}$$

with a $n \times n$ symmetric and positive (semi)definite matrix A are solved by an application of an iterative method

$$x \leftarrow \widehat{M}x + \widehat{N}f \tag{4.2}$$

consistent with (4.1) (i.e., having the same solution), with A -symmetric matrix \widehat{M} satisfying

$$\|\widehat{M}\|_A \leq q, \quad q \in (0, 1). \tag{4.3}$$

The following lemma gives a condition number estimate for this procedure.

Lemma 4.2.1. Assume that the iteration (4.2) is consistent with (4.1), and that it is convergent in the sense of (4.3). Then

$$(1 - q)\langle A^{-1}x, x \rangle \leq \langle \widehat{N}x, x \rangle \leq (1 + q)\langle A^{-1}x, x \rangle \quad \forall x \in \mathbb{R}^n \tag{4.4}$$

and

$$(1 - q)\langle \widehat{N}^{-1}x, x \rangle \leq \langle Ax, x \rangle \leq (1 + q)\langle \widehat{N}^{-1}x, x \rangle \quad \forall x \in \mathbb{R}^n \tag{4.5}$$

(the constant q can be made arbitrarily small provided sufficiently many steps of iteration (4.2) are used).

Proof. Let us first assume that A is nonsingular. The consistency of (4.2) implies $\widehat{M} = I - \widehat{N}A$. Using Cauchy-Schwarz inequality, we have

$$|\langle \widehat{M}x, x \rangle_A| \leq \|\widehat{M}x\|_A \|x\|_A \leq q \|x\|_A^2.$$

Thus, we obtain

$$(1 - q)\langle x, x \rangle_A \leq \langle \widehat{N}Ax, x \rangle_A \leq (1 + q)\langle x, x \rangle_A \quad \forall x \in \mathbb{R}^n \quad (4.6)$$

or

$$(1 - q)\langle A^{-1}x, x \rangle \leq \langle \widehat{N}x, x \rangle \leq (1 + q)\langle A^{-1}x, x \rangle \quad \forall x \in \mathbb{R}^n. \quad (4.7)$$

Since the matrix A is assumed positive definite, the last equation implies positive definiteness of \widehat{N} .

Since \widehat{N} is nonsingular and symmetric (A symmetric and nonsingular), substituting $y = A^{-1/2}x$ in (4.7), we obtain

$$(1 - q)\langle y, y \rangle \leq \langle A^{1/2}\widehat{N}A^{1/2}y, y \rangle \leq (1 + q)\langle y, y \rangle \quad \forall y \in \mathbb{R}^n.$$

From here

$$\frac{1}{1 + q}\langle y, y \rangle \leq \langle A^{-1/2}\widehat{N}^{-1}A^{-1/2}y, y \rangle \leq \frac{1}{1 - q}\langle y, y \rangle \quad \forall y \in \mathbb{R}^n.$$

Using substitution $x = A^{-1/2}y$, we obtain

$$(1 - q)\langle \widehat{N}^{-1}x, x \rangle \leq \langle Ax, x \rangle \leq (1 + q)\langle \widehat{N}^{-1}x, x \rangle \quad \forall x \in \mathbb{R}^n. \quad (4.8)$$

The above estimates relied on the assumption that A be nonsingular. Let us now assume that A is symmetric, but semidefinite. The pseudo-inverse A^+ of A is an inverse on the space $\text{Range}(A)$, and the argument above can be used for $A_L = A|_{\text{Range}(A)}$. \square

When computing the approximate harmonic extension \hat{E} , we will apply (4.2) started from zero initial approximation in the interiors to the problem

$$A_{22}x^E = -A_{21}\bar{x}. \quad (4.9)$$

The following lemma gives an estimate for the case only one iteration of (4.2) is used to approximate the harmonic extension

Lemma 4.2.2. Let us denote by h the characteristic meshsize. Denoting by E the operator of discrete harmonic extension, we have for the inexact harmonic extension \hat{E} the following estimate

$$\|(E - \hat{E})u\|_A \leq C \frac{q}{h} \|Eu\|_A,$$

where q is the parameter from (4.3).

Proof. Let us denote by $Q : V \rightarrow V_0$ the linear operator defined as

$$Qu = \begin{cases} 0 & \text{for } x \in \partial\Omega, \\ u & \text{for } x \in \overset{\circ}{\Omega}. \end{cases}$$

Since we assume the iteration (4.2) for problem (4.9) is started from zero approximation (in the interiors), we have for the initial error $e_0 = QEu$ and after one iteration $e_1 = \widehat{M}e_0$. Therefore, using the definition of Q and (4.3),

$$\begin{aligned} \|(E - \hat{E})u\|_A^2 &= \|Qe_1\|_A^2 = \|e_1\|_{A_{22}}^2 = \|\widehat{M}e_0\|_{A_{22}}^2 \\ &\leq q^2 \|e_0\|_{A_{22}}^2 = q^2 \|Qe_0\|_A^2 = q^2 \|QEu\|_A^2 \leq q^2 \|Q\|_A^2 \|Eu\|_A^2. \end{aligned}$$

In order to conclude the proof, it suffices to realize that $\|Q\|_A \leq \frac{C}{h}$. \square

4.3 Matsokin-Nepomnyaschikh Abstract Theory

The abstract theory presented by Nepomnyaschikh in [73] provides an alternative approach to describing domain decomposition methods, cf. also [55].

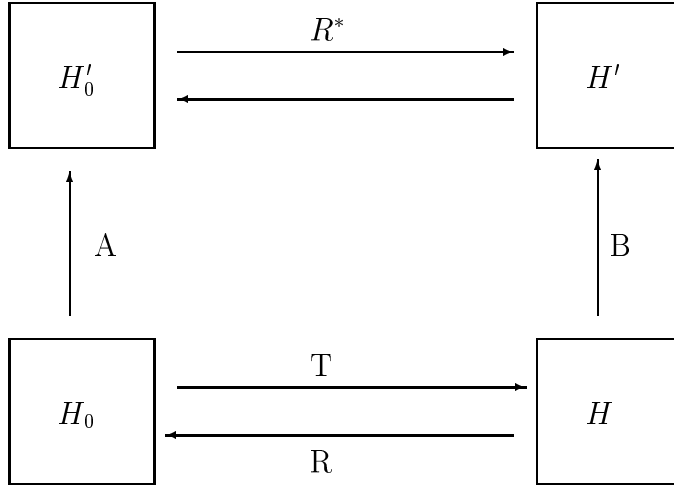


Figure 4.1: Abstract framework scheme.

4.3.1 Abstract Framework and Condition Number Estimate

Following [55], let us consider Hilbert spaces H_0, H and self-adjoint, elliptic operators A, B on H_0, H , respectively. Also consider linear continuous extension and restriction maps $T : H_0 \rightarrow H$ and $R : H \rightarrow H_0$, so that

$$R \circ T = I_{H_0}. \quad (4.10)$$

Consider solving the problem

$$\text{Find } u_0 \in H_0 : \quad Au_0 = L_0 \text{ in } H'_0 \quad (4.11)$$

by the conjugate gradient method with the preconditioner

$$\mathcal{M}^{-1} = RB^{-1}R^T. \quad (4.12)$$

The following theorem gives an abstract condition number bound for this process. Since no written reference for the proof of the theorem is known to the author, we include it here for completeness. The proof follows discussion [54].

Theorem 4.3.1 ([73]). For the preconditioner $\mathcal{M}^{-1} = RB^{-1}R^T$ it holds that

$$\text{cond}(\mathcal{M}^{-1}A) \leq \frac{C_R}{C_T}, \quad (4.13)$$

with

$$C_R = \sup_{v \in H} \frac{\langle ARv, Rv \rangle}{\langle Bv, v \rangle} = \|R\|_{B \rightarrow A}^2 \quad (4.14)$$

$$\frac{1}{C_T} = \sup_{v_0 \in H_0} \frac{\langle BTv_0, Tv_0 \rangle}{\langle Av_0, v_0 \rangle} = \|T\|_{A \rightarrow B}^2 \quad (4.15)$$

Proof. First note that

$$\mathcal{M}^{-1}A = RB^{-1}R^T A = RR^*. \quad (4.16)$$

Indeed, by definition of R^* ,

$$\langle R^*u, v \rangle_B = \langle u, Rv \rangle_A \quad \forall u \in H_0, \quad \forall v \in H,$$

$$\langle R^*u, Bv \rangle_{L^2(H')} = \langle Au, Rv \rangle_{L^2(H_0)}.$$

Therefore

$$R^* = B^{-1}R^T A,$$

from where (4.16) readily follows. From (4.16) and the definition of condition number,

$$\text{cond}(\mathcal{M}^{-1}A) = \text{cond}(RR^*) = \frac{\lambda_{\max}(RR^*)}{\lambda_{\min}(RR^*)}.$$

We have

$$\begin{aligned} \lambda_{\max}(RB^{-1}R^T A) &= \lambda_{\max}(RR^*) \\ &= \|R\|_{B \rightarrow A}^2 \\ &\leq \sup_{v \in H} \frac{\langle R^*Rv, v \rangle_B}{\langle v, v \rangle_B} \end{aligned}$$

$$\begin{aligned}
&\leq \sup_{v \in H} \frac{\langle ARv, Rv \rangle}{\langle Bv, v \rangle} \\
&= \|R\|_{B \rightarrow A}^2.
\end{aligned}$$

For the lower bound of the spectrum we have

$$\begin{aligned}
\lambda_{\min}(RR^*) &= \inf_{u \in H_0} \frac{\langle RR^*u, u \rangle_A}{\langle u, u \rangle_A} \\
&= \inf_{u \in H_0} \frac{\langle R^*u, R^*u \rangle_B}{\langle u, u \rangle_A} \\
&= \inf_{u \in H_0} \frac{\|R^*u\|_B^2}{\|u\|_A^2} \\
&= \inf_{u \in H_0} \sup_{v \in H} \frac{\langle R^*u, v \rangle^2}{\|u\|_A^2 \|v\|_B^2} \\
&= \inf_{u \in H_0} \sup_{v \in H} \frac{\langle u, Rv \rangle_A^2}{\|u\|_A^2 \|v\|_B^2} \\
&\geq \inf_{u \in H_0} \frac{\langle u, u \rangle_A^2}{\|u\|_A^2 \|Tu\|_B^2} \\
&= \inf_{u \in H_0} \frac{\langle u, u \rangle_A}{\|Tu\|_B^2},
\end{aligned}$$

therefore

$$\frac{1}{\lambda_{\min}(RR^*)} \leq \sup_{u \in H_0} \frac{\|Tu\|_B^2}{\|u\|_A^2} = \|T\|_{A \rightarrow B}^2,$$

which concludes the proof. \square

Note that the operator T appearing in the estimates is introduced for the purposes of the theory only; it is tied to R by (4.10) but it does not play any explicit role in the preconditioner itself.

4.3.2 An Application: Abstract Additive Schwarz Methods

The generality of Theorem 4.3.1 will be demonstrated by reproving the statement of Theorem 2.1.1 within our current framework. Let

$$V = V_0 + V_1 + \dots + V_J, \quad H_0 = V, \quad H = V_0 \times V_1 \times \dots \times V_J$$

and $\langle \cdot, \cdot \rangle$ denote the L^2 inner product. Define operators

$$B : H \rightarrow H, \quad B : v \rightarrow (B_0 v_0, \dots, B_J v_J), \quad B_i : V_i \rightarrow V_i.$$

On H , define the inner product

$$((v_0, \dots, v_J), (w_0, \dots, w_J))_H = \sum_{i=0}^J (v_i, w_i)_{V_i},$$

bilinear forms

$$a(u, v) = \langle Au, v \rangle \quad \forall u, v \in V,$$

$$a_i(u, v) = \langle A_i u, v \rangle \quad \forall u, v \in V_i = \text{Range}(A_i^+)$$

and

$$R : H \rightarrow H_0, \quad R : (v_0, \dots, v_J) \rightarrow \sum_{i=0}^J v_i.$$

Since

$$R^T v = (P_{V_0} v, P_{V_1} v, \dots, P_{V_J} v),$$

where P_{V_i} is the L^2 -orthogonal projection onto the space V_i , and

$$B^{-1} : H \rightarrow H,$$

$$B^{-1}(w_0, \dots, w_J) = (A_0^+ w_0, \dots, A_J^+ w_J).$$

Finally, define operators

$$\mathcal{T}_i = A_i^+ P_{V_i} A.$$

These are the approximate projections, because

$$\begin{aligned} a_i(\mathcal{T}_i w, v) &= \langle A_i A_i^+ P_{V_i} A w, v \rangle = \langle A_i A_i^+ P_{V_i} A w, A_i^+ \tilde{v} \rangle = \langle A_i^+ P_{V_i} A w, \tilde{v} \rangle \\ &= \langle A w, v \rangle = a(w, v) \quad \forall v \in V_i. \end{aligned}$$

With these definitions, we obtain $RB^{-1}R^T A = \sum_{j=0}^J \mathcal{T}_j$ and we can prove the following theorem.

Theorem 4.3.2 (Dryja, Widlund [30]). Assume that

(1) there exists a linear continuous operator $T : H_0 \rightarrow H$ so that

$$T : v \rightarrow (v_0, \dots, v_J), \quad R \circ T = I_{H_0}, \quad (4.17)$$

and

$$\sum_{i=0}^J \langle B_i v_i, v_i \rangle \leq C_0^2 \langle Av, v \rangle \quad (4.18)$$

(2) there exists a constant $\omega > 0$ so that

$$\langle Av, v \rangle \leq \omega \langle B_i v, v \rangle \quad \forall v \in V_i, i = 0, \dots, J. \quad (4.19)$$

(3) there exist constants ϵ_{ij} , $i, j = 1, \dots, J$ so that

$$\langle Av_i, v_j \rangle \leq \epsilon_{ij} \langle Av_i, v_i \rangle^{1/2} \langle Av_j, v_j \rangle^{1/2} \quad \forall v_i \in V_i, \quad \forall v_j \in V_j. \quad (4.20)$$

Then

$$\text{cond}(RB^{-1}R^T A) \leq C_0^2 \omega (1 + \varrho(\epsilon)).$$

Proof. We will use the abstract estimate of Theorem 4.3.1. First,

$$\begin{aligned} \frac{1}{C_T} &= \sup_{v \in H_0} \frac{\langle BTv, Tv \rangle}{\langle Av, v \rangle} \\ &= \sup_{v \in H_0} \frac{\sum_{i=0}^J \langle B_i v_i, v_i \rangle}{\langle Av, v \rangle} \\ &\leq \sup_{v \in H_0} \frac{\sum_{i=0}^J \langle B_i v_i, v_i \rangle}{\frac{1}{C_0^2} \sum_{i=0}^J \langle B_i v_i, v_i \rangle} = C_0^2. \end{aligned}$$

Now, let us turn to estimating C_R . Using (4.19) and (4.20), we obtain:

$$\begin{aligned} C_R &= \sup_{v \in H} \frac{\langle ARv, Rv \rangle}{\langle Bv, v \rangle} \\ &= \sup_{v \in H} \frac{\|\sum_{i=0}^J v_i\|_A^2}{\sum_{i=0}^J \langle B_i v_i, v_i \rangle} \end{aligned}$$

$$\begin{aligned}
&\leq \sup_{v \in H} \frac{\|\sum_{i=0}^J v_i\|_A^2}{\frac{1}{\omega} \sum_{i=0}^J \|v_i\|_A^2} \\
&= \omega \sup_{v \in H} \frac{\|v_0 + \sum_{i=1}^J v_i\|_A^2}{\|v_0\|_A^2 + \sum_{i=1}^J \|v_i\|_A^2} \\
&\leq \omega \sup_{v \in H} \frac{\left(1 + \varrho(\varepsilon)^{1/2} \frac{\sqrt{\sum_{i=1}^J \|v_i\|_A^2}}{\|v_0\|_A}\right)^2}{1 + \frac{\sum_{i=1}^J \|v_i\|_A^2}{\|v_0\|_A^2}}.
\end{aligned}$$

Substituting $c = \varrho(\varepsilon)^{1/2}$, $t = \frac{\sqrt{\sum_{i=1}^J \|v_i\|_A^2}}{\|v_0\|_A}$, the trivial inequality

$$\frac{(1 + ct)^2}{1 + t^2} \leq 1 + c^2$$

yields $C_R \leq \omega(1 + \varrho(\varepsilon))$. Thus, from Theorem 4.3.1 we obtain

$$\text{cond}(\mathcal{M}^{-1}A) \leq C_0^2 \omega(\varrho(\varepsilon) + 1),$$

which was to be proved. \square

It is easy to see that, indeed, conditions (4.17), (4.18) and (4.20) are identical to (i), (ii) and (iii) of Theorem 2.1.1.

4.4 Unextended Hybrid Schwarz Algorithm

Let us set

$$V = V_0 + \sum_{i=1}^J V_i,$$

$$A : V \rightarrow V,$$

$$B_i : V_i \rightarrow V_i.$$

Denote $P = P_{V_0}^A$ the $\langle A \cdot, \cdot \rangle$ -orthogonal projection onto V_0 , and $P_i = P_{V_i}$ the $\langle \cdot, \cdot \rangle$ -orthogonal projection onto $V_i = \text{Range}(Z_i^- i)$.

Adopting the above notation, we may describe the preconditioner by the following algorithm:

Algorithm 12. For a given $r \in V$,

1. Solve $q \in V_0 : \langle Aq, v \rangle = \langle r, v \rangle \quad \forall v \in V_0$
2. Set $s = r - Aq$
3. Solve $u_i \in V_i : \langle B_i u_i, v_i \rangle = \langle s, v_i \rangle \quad \forall v_i \in V_i$
4. Set $u = \sum_{i=1}^J u_i$
5. Solve $\langle A(u - u_0), v_0 \rangle = \langle r, v_0 \rangle \quad \forall v_0 \in V_0$
6. Output $z = \mathcal{M}^{-1}r = u - u_0$

It is easy to see that with the preconditioner described by Algorithm 12 the preconditioned operator will be

$$\mathcal{M}^{-1}A = (I - P) \sum_{i=1}^J (I - P_i) B_i^{-1} (I - P_i) (I - P) A.$$

When using exact subdomain solvers, step 1. of Algorithm 12 can be performed only once at the start of the iteration and then omitted. This would yield

$$\mathcal{M}^{-1}A = (I - P) \sum_{i=1}^J (I - P_i) B_i^{-1} (I - P_i) A.$$

The following theorem gives a condition estimate:

Theorem 4.4.1. Let the following two assumptions be satisfied

- (i) There exists a linear mapping $u \in V \rightarrow (u_0, u_1, \dots, u_J)$, $u_i \in V_i$ such that

$$u = u_0 + \sum_{i=1}^J u_i \quad \text{and} \quad \sum_{i=1}^J \langle B_i u_i, u_i \rangle \leq C_0 \langle Au, u \rangle. \quad (4.21)$$

- (ii) For all v_1, \dots, v_J , $v_i \in V_i$ and $v = \sum_{i=1}^J v_i$ it holds that

$$\langle Av, v \rangle \leq C_1 \sum_{i=1}^J \langle B_i v_i, v_i \rangle. \quad (4.22)$$

Then

$$\text{cond}(\mathcal{M}^{-1}A) \leq C_0 C_1. \quad (4.23)$$

Proof. We will use the Matsokin-Nepomnyaschikh hybrid framework with the following choice of spaces and operators:

$$H = \otimes_{i=1}^J V_i, \quad H_0 = (I - P)V,$$

$$R((v_i)_{i=1}^J) = \sum_{i=1}^J (I - P)v_i, \quad B = \text{diag}(B_i), \quad Tu = (u_i)_{i=1}^J$$

(Tu is the decomposition from assumption (4.21).) Then we have $R \circ T = I_{H_0}$, as for a $u \in (I - P)V$,

$$\begin{aligned} (R \circ T)u &= (I - P)(u_1 + \dots + u_J) \\ &= (I - P)(u - u_0) \\ &= (I - P)u = u. \end{aligned}$$

We can estimate

$$\begin{aligned} \|R((v_i))\|_A^2 &= \|(I - P) \sum_{i=1}^J v_i\|_A^2 \\ &\leq \left\| \sum_{i=1}^J v_i \right\|_A^2 \\ &\leq C_1 \sum_{i=1}^J \langle B_i v_i, v_i \rangle. \end{aligned}$$

Therefore $\|R\|_{B \rightarrow A} \leq \sqrt{C_1}$. Finally, we have

$$\|Tu\|_B^2 = \sum_{i=1}^J \|u_i\|_{B_i}^2 \leq C_0 \|u\|_A^2,$$

i.e., $\|T\|_{A \rightarrow B} \leq \sqrt{C_0}$. Now (4.23) easily follows from Theorem 4.3.1. \square

4.4.1 BDD as a Hybrid Schwarz Algorithm

We will show that the original BDD method is a hybrid Schwarz Algorithm as described by Algorithm 12.

Let us set

$$\begin{aligned}
V &= V_0 + \sum_{i=1}^J V_i, \\
\tilde{V}_i &= \text{Range}(\bar{N}_i), \\
V_i &= \bar{N}_i D_i (I - P_i) \tilde{V}_i, \\
V_0 &= \text{Range} \left(\sum_{i=1}^J \bar{N}_i D_i Z_i \right) = \sum_{i=1}^J \bar{N}_i D_i P_i \tilde{V}_i, \\
A &= S, \\
B_i &= \bar{N}_i D_i^{-1} S_i D_i^{-T} \bar{N}_i^T.
\end{aligned}$$

With this choice of components, steps 1. and 5. of Algorithm 12 are just the pre- and post-balancing steps as in the original BDD method.

Investigating Step 3., we further have:

$$u_i \in V_i \quad \text{such that} \quad \langle B_i u_i, v_i \rangle = \langle r, v_i \rangle \quad \forall v_i \in V_i,$$

where

$$u_i = \bar{N}_i D_i (I - P_i) \tilde{u}_i, \quad v_i = \bar{N}_i D_i (I - P_i) \tilde{v}_i.$$

Thus

$$\langle \bar{N}_i D_i^{-1} S_i D_i^{-T} \bar{N}_i^T \bar{N}_i D_i (I - P_i) \tilde{u}_i, \bar{N}_i D_i (I - P_i) \tilde{v}_i \rangle = \langle r, \bar{N}_i D_i (I - P_i) \tilde{v}_i \rangle$$

or

$$\langle S_i (I - P_i) \tilde{u}_i, (I - P_i) \tilde{v}_i \rangle = \langle D_i^T \bar{N}_i^T r, (I - P_i) \tilde{v}_i \rangle.$$

Denoting $\tilde{\tilde{u}}_i = (I - P) \tilde{u}_i$, (which assures solvability), we have

$$\langle S_i \tilde{\tilde{u}}_i, \tilde{\tilde{v}}_i \rangle = \langle D_i \bar{N}_i^T r, \tilde{\tilde{v}}_i \rangle.$$

Now we have

$$u_i = \bar{N}_i D_i \tilde{u}_i,$$

which is the same as in the Step 2. of the original BDD method.

4.4.2 A New Look at the Conditioning of BDD

In Section 4.4.1 we have verified that with proper choice of spaces and operators, the original BDD method can be derived as a hybrid Schwarz method fitting the abstract framework introduced by Matsokin and Nepomnyaschikh. Therefore, the estimate of Theorem 2.7.1 is valid. We can, however, easily prove the result of Theorem 2.7.1 within our current framework and we do so in the next lemma.

Lemma 4.4.2. The BDD algorithm yields a preconditioner satisfying

$$\text{cond}(\mathcal{M}, A) \leq \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{j=1}^J \|\bar{N}_j \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_j}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2}.$$

Proof. In view of the previous section, it suffices to verify the assumptions of Theorem 4.4.1. First, for any u we prove the existence of $C_0 > 0$ so that there exists a decomposition $u = u_0 + \sum_{i=1}^J u_i$ and

$$\sum_{i=1}^J \langle B_i u_i, u_i \rangle \leq C_0 \langle Au, u \rangle.$$

Let us define a decomposition of u :

$$\tilde{u}_i = \bar{N}_i^T u, \quad u_0 = \sum_{i=1}^J \bar{N}_i D_i P_i \tilde{u}_i, \quad u_i = N_i D_i (I - P_i) \tilde{u}_i.$$

Then we have

$$\sum_{i=1}^J \langle B_i u_i, u_i \rangle = \sum_{i=1}^J \bar{N}_i D_i^{-1} S_i D_i^{-1} \bar{N}_i^T \bar{N}_i D_i (I - P_i) \tilde{u}_i, \bar{N}_i D_i (I - P_i) \tilde{u}_i \rangle.$$

Since $D_i^{-T} \bar{N}_i^T \bar{N}_i D_i = I$, and $\text{Range}(P_i) = \text{Ker}(S_i)$ (i.e. $\text{Ker}(S_i) = \bar{Z}_i$),

$$\begin{aligned} \sum_{i=1}^J \langle B_i u_i, u_i \rangle &= \sum_{i=1}^J \langle S_i (I - P_i) \tilde{u}_i, (I - P_i) \tilde{u}_i \rangle \\ &= \sum_{i=1}^J \langle S_i \bar{N}_i^T u, \bar{N}_i^T u \rangle \\ &= \langle S u, u \rangle. \end{aligned}$$

Therefore, $C_0 = 1$.

Let us now prove existence of a positive constant C_1 so that

$$\forall v_i \in V_i, \quad \langle S \sum_{i=1}^J v_i, \sum_{i=1}^J v_i \rangle \leq C_1 \sum_{i=1}^J \langle B_i v_i, v_i \rangle.$$

By the standard process of subassembly, the definition of V_i , and Cauchy-Schwarz inequality we have

$$\begin{aligned} \langle S \sum_{i=1}^J v_i, \sum_{j=1}^J v_j \rangle &= \langle \sum_{l=1}^J \bar{N}_l S_l \bar{N}_l^T \sum_{i=1}^J v_i, \sum_{j=1}^J v_j \rangle \\ &= \sum_{i,j,l=1}^J \langle \bar{N}_l S_l \bar{N}_l^T v_i, v_j \rangle \\ &= \sum_{i,j,l=1}^J \langle \bar{N}_l S_l \bar{N}_l^T \bar{N}_i D_i (I - P_i) \tilde{v}_i, \bar{N}_j D_j (I - P_j) \tilde{v}_j \rangle \\ &\leq \sum_{i,l=1}^J \langle \bar{N}_l S_l \bar{N}_l^T \bar{N}_i D_i (I - P_i) \tilde{v}_i, \bar{N}_i D_i (I - P_i) \tilde{v}_i \rangle \\ &\leq \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{j=1}^J \|\bar{N}_j \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_j}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2} \sum_{i=1}^J \langle B_i v_i, v_i \rangle, \end{aligned}$$

concluding the proof. \square

Remark 4.4.3. We see that under the assumption that $\text{Range}(P_i) = \text{Ker}(S_i)$, the condition number is bounded by the same constant as in the original BDD method.

4.5 Extended Hybrid Schwarz Algorithm

As we already mentioned in the beginning of Chapter 4, the key to application of inexact local solvers is the formulation of the method in terms of the local stiffness matrices rather than the local Schur complement matrices. This means that the vectors we will use will include the components corresponding to the interior degrees of freedom. As using the term “extended” seems unfortunate to the author in the presence of the operators of discrete harmonic extension, we will use the term long vectors to distinguish these vectors from the ones defined on the interfaces only.

4.5.1 Algorithm on Long Vectors with Exact Components

Let \widehat{V} denote the space including interiors, and \bar{V} the space of discrete harmonic functions. Further let $\tilde{V}_i = \text{Range}(\bar{N}_i)$. Let us define local spaces

$$\widehat{V}_i = E\bar{N}_i D_i (I - P_i) \bar{N}_i^T \tilde{V}_i,$$

the interior spaces

$$\overset{\circ}{V}_i = \overset{\circ}{N}_i V = \overset{\circ}{N}_i \overset{\circ}{N}_i^T (I - E) V,$$

and the coarse space

$$\widehat{V}_0 = E \sum_{i=1}^J \bar{N}_i D_i P_i \bar{N}_i^T \tilde{V}_i.$$

We now have two sets of local operators, $\widehat{B}_i = \bar{N}_i D_i^{-1} S_i D_i^{-T} \bar{N}_i^T$ and the bilinear form restricted to the interiors: $\overset{\circ}{B}_i = \overset{\circ}{N}_i \overset{\circ}{N}_i^T A \overset{\circ}{N}_i \overset{\circ}{N}_i^T$. Here E denotes the operator of discrete harmonic extension $E : \widehat{V} \rightarrow \bar{V}$, in matrix notation

$$E = \begin{bmatrix} I & 0 \\ -A_{22}^{-1} A_{21} & 0 \end{bmatrix}.$$

Since the discrete harmonic extension of a function is uniquely determined by the function's values on the subdomain interfaces, we allow E to be also used as an operator $E : \widehat{V}(\Gamma) \rightarrow \bar{V}$.

For the sake of convenience of notation, let

$$V_i = \begin{cases} \widehat{V}_i & \text{for } i = 1, \dots, J, \\ \overset{\circ}{V}_{i-J} & \text{for } i = J + 1, \dots, 2J. \end{cases}$$

and

$$B_i = \begin{cases} \widehat{B}_i & \text{for } i = 1, \dots, J, \\ \overset{\circ}{B}_{i-J} & \text{for } i = J + 1, \dots, 2J. \end{cases}$$

With the above definitions, we can write the algorithm on long vectors

Algorithm 13. Set $q=0$.

1. Pre-balance: compute $q \in \widehat{V}_0 : \langle Aq, v \rangle = \langle r, v \rangle \quad \forall v \in \widehat{V}_0$.
2. Set $s = r - Aq$.
3. Compute:
 - a. $\hat{u}_i : \langle \widehat{B}_i \hat{u}_i, \hat{v}_i \rangle = \langle s, \hat{v}_i \rangle \quad \forall \hat{v}_i \in \widehat{V}_i$,
 - b. $\dot{u}_i : \langle \overset{\circ}{B}_i \dot{u}_i, \dot{v}_i \rangle = \langle s, \dot{v}_i \rangle \quad \forall \dot{v}_i \in \overset{\circ}{V}_i$.
4. Set $u = \sum_{i=1}^J (\hat{u}_i + \dot{u}_i)$.
5. Post-balance: compute $\langle A(u - u_0), v \rangle = \langle r, v \rangle \quad \forall v \in \widehat{V}_0$.
6. Output balanced $\mathcal{M}^{-1}r = u - u_0$.

Remark 4.5.1. Again, if all the components of the method are exact, in solving the pre- and post-balancing steps, the pre-balancing can only be performed once before the iteration commences and omitted in subsequent iterations.

4.5.2 Practical Algorithm on Long Vectors and ACDD

Algorithm 13 as written is not very practical, especially because it requires knowledge of the local Schur complements S_i contained in the definition of \widehat{B}_i . We will rewrite the algorithm in a form suitable for a computer implementation.

Algorithm 14. Set $q=0$.

1. Pre-balance: solve $q \in \widehat{V}_0 : \langle Aq, v \rangle = \langle r, v \rangle \quad \forall v \in \widehat{V}_0$.

2. Set $s = r - Aq$, where $s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$.

(1) Compute $\widehat{s} = A_{22}^{-1} s_2$ (J Dirichlet solves).

(2) Precompute $\widehat{\widehat{s}} = s_1 - A_{12} \widehat{s}$.

3. Solve J Neumann problems:

$$A_i \begin{bmatrix} \tilde{u}_i \\ \text{discard} \end{bmatrix} = \begin{bmatrix} (I - P_i) D_i \bar{N}_i^T \widehat{\widehat{s}} \\ 0 \end{bmatrix}, \quad i = 1, \dots, J.$$

4. Set $\widehat{\widehat{u}}_i = \bar{N}_i D_i (I - P_i) \tilde{u}_i$, $i = 1, \dots, J$.

5. Set $u = E \sum_{i=1}^J \widehat{\widehat{u}}_i + \widehat{\widehat{s}} = \sum_{i=1}^J \widehat{u}_i + \widehat{\widehat{s}}$, where $\widehat{u}_i = \begin{bmatrix} \widehat{\widehat{u}}_i \\ d_i \end{bmatrix}$,

with d_i computed by J Dirichlet solvers

$$d_i = -A_{22}^{(i)-1} A_{21}^{(i)} \bar{N}_i^T \sum_{j=1}^J \widehat{\widehat{u}}_j, \quad i = 1, \dots, J.$$

6. Post-balance: compute $u_0 \in \widehat{V}_0$ such that

$$\langle A(u - u_0), v \rangle = \langle r, v \rangle \quad \forall v \in \widehat{V}_0.$$

7. Output balanced $\mathcal{M}r = u - u_0$.

Remark 4.5.2. The balancing step consists of solving a linear system of algebraic equations $By = f$, where B is a symmetric positive (semi)definite matrix with entries

$$B_{ij} = \bar{Z}_i^T D_i^T \bar{N}_i^T E^T A E \bar{N}_j D_j \bar{Z}_i$$

and a right-hand side has entries $f_i = \bar{Z}_i^T D_i^T \bar{N}_i^T E^T r$.

Remark 4.5.3. For exact subdomain and coarse space solvers, $s_2 = 0$ so $\hat{s} = s_1, \hat{s} = 0$ and we recover Algorithm 12, mathematically equivalent to BDD.

Remark 4.5.4. A total of mJC Dirichlet solves will have to be used in the setup to compute the basis of testing functions used in the construction of the balancing matrix B , where $m = \max_{i=1,\dots,J} \dim(\text{Ker}(A_i))$, $C = \max_{i=1,\dots,J} \text{card}\{j : \partial\Omega_i \cap \partial\Omega_j \neq \emptyset\}$. In step 4., the term $\sum_{j=1}^J \hat{u}_j$ will be pre-computed to be used for all values of i . The cost involved in the computation of steps 1., 2., 3., 4. and 5. is roughly that of solving $2J$ Dirichlet problems and J Neumann problems. This algorithm, however, is only an intermediate step. The local solves will ultimately be replaced by iterations. We will code-name the method ACDD (standing for the domain decomposition with approximate components). We will write $\text{ACDD}(\infty)$ to denote the method with all components exact, i.e, the method mathematically equivalent to BDD. On occasion, we will write $\text{ACDD}(k)$ to denote the method where some or all of the local solves were replaced by inexact ones. The quantity k will be the number of iterations of inexact solvers performed.

4.5.3 Estimate on Long Vectors for Inexact Neumann Solvers

In this section, we give a condition number bound, assuming that the harmonic extensions E and the coarse space solver are exact, but the Neumann solves in step 3. of Algorithm 14 are replaced with an iterative process like (4.2), with the rate of convergence given by (4.3).

Lemma 4.5.5. Using Algorithm 14 with local Neumann problems replaced by consistent iterative process (4.2), with the rate of convergence bounded by $q \in (0, 1)$ uniformly with respect to the number of subdomain, then we have

$$\text{cond}(\mathcal{M}, A) \leq \frac{1+q}{1-q} \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{j=1}^J \|\bar{N}_j \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_j}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2}.$$

Proof. We will apply Theorem 4.3.1 again. As the only component affected by the use of approximate Neumann solves are the operators B_i , $i = 1, \dots, J$ and we have

$$\begin{aligned} \langle S_i^+ x, x \rangle &= \langle A_i^+ \begin{bmatrix} x \\ 0 \end{bmatrix}, \begin{bmatrix} x \\ 0 \end{bmatrix} \rangle \\ &\approx \langle \widehat{N}_i \begin{bmatrix} x \\ 0 \end{bmatrix}, \begin{bmatrix} x \\ 0 \end{bmatrix} \rangle = \langle \tilde{S}_i^+ x, x \rangle, \end{aligned}$$

so the operators \tilde{B}_i resulting from the application of iterative solvers satisfy $\tilde{B}_i \approx B_i$, where the constants of equivalence can be bounded (cf. (4.4)) from above by $\frac{1}{1-q}$, and from below by $\frac{1}{1+q}$. For a given $u \in V$, let us set

$$\hat{u}_i = E \bar{N}_i D_i (I - P_i) \bar{N}_i^T u,$$

$$u_0 = E \sum_{i=1}^J \bar{N}_i D_i P_i \bar{N}_i^T u,$$

$$\dot{u}_i = \overset{\circ}{N}_i \overset{\circ}{N}_i^T (I - E)u.$$

The above defines a decomposition of u into its \widehat{V}_i, V_0 and $\overset{\circ}{V}_i$ components. Using the definition of $\widehat{V}_i, \overset{\circ}{V}_i$, the identity $\bar{N}_i^T E \bar{N}_i = I$, the process of subassembly and $\text{Ker}(S_i) = \text{Range}(\bar{Z}_i)$, we have

$$\begin{aligned} \sum_{i=1}^{2J} \langle \tilde{B}_i u_i, u_i \rangle &\leq \frac{1}{1-q} \sum_{i=1}^J \langle \widehat{B}_i \widehat{u}_i, \widehat{u}_i \rangle + \sum_{i=1}^J \langle \overset{\circ}{B}_i \dot{u}_i, \dot{u}_i \rangle \\ &\leq \frac{1}{1-q} \left(\sum_{i=1}^J \langle S_i (I - P_i) \bar{N}_i^T u, (I - P_i) \bar{N}_i^T u \rangle \right. \\ &\quad \left. + \sum_{i=1}^J \langle N_i A_i N_i^T (I - E)u, (I - E)u \rangle \right) \\ &= \frac{1}{1-q} \langle A E u, E u \rangle + \langle A (I - E)u, (I - E)u \rangle \\ &= \frac{1}{1-q} \langle A u, u \rangle, \end{aligned}$$

so, $C_0 = \frac{1}{1-q}$.

Let $v \in V$ and $\{v_i\}_{i=1}^{2J}$, $v_i \in V_i$ be given such that $v = \sum_{i=1}^{2J} v_i$. Then

$$\langle A \left(\sum_{i=1}^J \widehat{v}_i + \sum_{i=1}^J \dot{v}_i \right), \sum_{i=1}^J \widehat{v}_i + \sum_{i=1}^J \dot{v}_i \rangle = \sum_{i=1}^J \langle A \dot{v}_i, \dot{v}_i \rangle + \sum_{i=1}^J \langle A \widehat{v}_i, \widehat{v}_i \rangle.$$

The first term is trivially estimated using identity $\langle A \dot{v}_i, \dot{v}_i \rangle = \langle \overset{\circ}{B}_i \dot{v}_i, \dot{v}_i \rangle$. Using the subassembly and the definition of \widehat{V}_i and of Schur complement, the second term can be estimated as

$$\begin{aligned} \sum_{i=1}^J \langle A \widehat{v}_i, \widehat{v}_i \rangle &= \sum_{i,j,l=1}^J \langle (N_l A_l N_l^T) \widehat{v}_i, \widehat{v}_j \rangle \\ &= \sum_{i,j,l=1}^J \langle N_l A_l N_l^T E \bar{N}_i D_i (I - P_i) \tilde{v}_i, E \bar{N}_j D_j (I - P_j) \tilde{v}_j \rangle \\ &= \sum_{l=1}^J \sum_{i,j=1}^J \langle \bar{N}_l S_l \bar{N}_l^T \bar{N}_i D_i (I - P_i) \tilde{v}_i, \bar{N}_j D_j (I - P_j) \tilde{v}_j \rangle. \end{aligned}$$

Application of Cauchy-Schwarz inequality yields

$$\begin{aligned}
\sum_{i=1}^J \langle A\hat{v}_i, \hat{v}_i \rangle &\leq \sum_{l=1}^J \sum_{i=1}^J \langle \bar{N}_l S_l \bar{N}_l^T \bar{N}_i D_i (I - P_i) \tilde{v}_i, \bar{N}_i D_i (I - P_i) \tilde{v}_i \rangle \\
&\leq \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{l=1}^J \|\bar{N}_l^T \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_l}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2} \sum_{i=1}^J \langle \hat{B}_i \hat{v}_i, \hat{v}_i \rangle \\
&\leq (1 + q) \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{l=1}^J \|\bar{N}_l^T \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_l}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2} \sum_{i=1}^J \langle \tilde{B}_i \hat{v}_i, \hat{v}_i \rangle.
\end{aligned}$$

From here it follows that $C_1 \leq (1 + q) \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{l=1}^J \|\bar{N}_l^T \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_l}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2}$. \square

4.5.4 Inexact Coarse Space and Harmonic Extensions

In this section we will investigate the case of Algorithm 14 with inexact harmonic extensions and an approximate coarse space. Two approximate coarse spaces will be considered. The first one is obtained by replacing the exact discrete harmonic extension by one that extends a function from a subdomain interface exactly, but only into the adjacent subdomains sharing an edge or face (in 3D) or sharing an edge (in 2D) with the given subdomain. That is, the extension will be zero across corners. This reduces the fill-in in the coarse-level operator. We assume that Algorithm 13 is used with inexact discrete harmonic extension \hat{E} instead of E , and \hat{E} is used in the definition of coarse space.

Definition 4.5.6. Let E denote the operator of discrete harmonic extension and u be a vector whose components are nonzero only in $\bar{\Omega}_i$. We define operator \hat{E} as follows:

$$(\hat{E}u)_l = \begin{cases} 0 & \text{if node } l \in \dot{\Omega}_j, \quad \text{where } \bar{\Omega}_i \cap \bar{\Omega}_j \text{ is a single point,} \\ (Eu)_l & \text{otherwise.} \end{cases}$$

The second approximate coarse space will be based on consistently using the same iterative method for computing the action of the discrete harmonic extension operator throughout the algorithm.

Let us first consider the inexact coarse space with reduced fill-in. We will need the following lemma in our estimates.

Lemma 4.5.7. For operator \hat{E} defined by Definition 4.5.6, it holds that

$$\langle A(E - \hat{E})w, (E - \hat{E})w \rangle \leq C \langle AEw, Ew \rangle \leq C(1 + \log(\frac{H}{h}))^2 \langle Au, u \rangle,$$

where $w = \sum_{j=1}^J \bar{N}_j D_j P_j \bar{N}_j^T u$.

Proof. Let us denote E_{P_1}, E_{Q_1} the discrete harmonic extension on the space P1 and Q1, respectively. From the discrete harmonic extension theorem (Widlund [92]) it follows that

$$\|E_{P_1} v\|_{H^1(\Omega)} \approx \|v\|_{1/2, \partial\Omega}, \quad \|E_{Q_1} v\|_{H^1(\Omega)} \approx \|v\|_{1/2, \partial\Omega}.$$

The first inequality of the lemma follows. The second inequality was proved in [65]. \square

Lemma 4.5.8. Using Algorithm 14 with exact local Neumann solvers, inexact harmonic extensions \hat{E} and inexact harmonic extension \hat{E} in the definition of a coarse space as a preconditioner, the condition number is bounded by

$$\text{cond}(\mathcal{M}, A) \leq C \left(1 + \left(\frac{q}{h}\right)^2\right)^2 \left(1 + \log \frac{H}{h}\right)^2 \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{j=1}^J \|\bar{N}_j \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_j}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2}.$$

Proof. Applying Theorem 4.3.1 again, we define a decomposition of u into into its \hat{V}_i, V_0 and \hat{V}_i : $\hat{u}_i = \hat{E} \bar{N}_i D_i (I - P_i) \bar{N}_i^T u$, $u_0 = \hat{E} \sum_{j=1}^J \bar{N}_j D_j P_j \bar{N}_j^T u$,

$\dot{u}_i = \dot{N}_i^T(I - \hat{E})u + \dot{N}_i^T(\hat{E} - \hat{\hat{E}}) \sum_{j=1}^J \bar{N}_j D_j P_j \bar{N}_j^T u$. Using the definition of $\hat{V}_i, \hat{\hat{V}}_i$ and the identity $\bar{N}_i^T E \bar{N}_i = I$, we have

$$\begin{aligned} \sum_{i=1}^{2J} \langle B_i u_i, u_i \rangle &= \sum_{i=1}^J \langle \hat{B}_i \hat{u}_i, \hat{u}_i \rangle + \sum_{i=1}^J \langle \hat{\hat{B}}_i \hat{\hat{u}}_i, \hat{\hat{u}}_i \rangle \\ &\leq \sum_{i=1}^J S_i (I - P_i) \bar{N}_i^T u, (I - P_i) \bar{N}_i^T u \\ &\quad + \sum_{i=1}^J \langle \dot{N}_i A_i \dot{N}_i^T ((\hat{E} - \hat{\hat{E}})w + (I - \hat{E})u, (\hat{E} - \hat{\hat{E}})w + (I - \hat{E})u). \end{aligned}$$

where $w = \sum_{j=1}^J \bar{N}_j D_j P_j \bar{N}_j^T u$. Thus, using Cauchy-Schwarz inequality several times, Lemma 4.2.2 and the definition of discrete harmonic extension,

$$\begin{aligned} \sum_{i=1}^{2J} \langle B_i u_i, u_i \rangle &\leq C \left(\frac{q^2}{h^2} \langle AEu, Eu \rangle + C \langle Au, u \rangle + \frac{q^2}{h^2} \langle AEw, Ew \rangle \right. \\ &\quad \left. + \langle A(E - \hat{\hat{E}})w, (E - \hat{\hat{E}})w \rangle \right). \end{aligned}$$

Now application of Lemma 4.5.7 yields $C_0 \leq C(1 + \frac{q^2}{h^2})(1 + \log(\frac{H}{h})^2)$.

Let $v \in V$ and $\{v_i\}_{i=1}^{2J}$, $v_i \in V_i$ be given such that $v = \sum_{i=1}^{2J} v_i$. Then

$$\begin{aligned} &\langle A(\sum_{i=1}^J \hat{v}_i + \sum_{i=1}^J \hat{\hat{v}}_i), \sum_{i=1}^J \hat{v}_i + \sum_{i=1}^J \hat{\hat{v}}_i \rangle \\ &= \langle AE \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u, E \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u \rangle \\ &\quad + \langle A(\hat{E} - E) \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u + \sum_{i=1}^J \hat{v}_i, \\ &\quad (\hat{E} - E) \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u + \sum_{i=1}^J \hat{\hat{v}}_i \rangle \\ &\leq \langle AE \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u, E \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u \rangle \\ &\quad + 2(\langle A(\hat{E} - E) \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u, (\hat{E} - E) \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u \rangle \\ &\quad + \langle \sum_{i=1}^J \hat{v}_i, \sum_{i=1}^J \hat{\hat{v}}_i \rangle) \end{aligned}$$

and application of Lemma 4.2.2 yields

$$\begin{aligned}
& \langle A(\sum_{i=1}^J \hat{v}_i + \sum_{i=1}^J \dot{v}_i), \sum_{i=1}^J \hat{v}_i + \sum_{i=1}^J \dot{v}_i \rangle \\
& \leq (1 + C \frac{q^2}{h^2}) \langle E \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u, \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \bar{N}_i^T u \rangle \\
& + \langle \sum_{i=1}^J \dot{v}_i, \sum_{i=1}^J \dot{v}_i \rangle.
\end{aligned}$$

The rest of the proof is identical to proof of Lemma 4.5.5, yielding

$$C_1 \leq (1 + C \frac{q^2}{h^2}) \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{l=1}^J \|\bar{N}_l^T \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_l}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2},$$

from where the result follows. \square

Remark 4.5.9. If the same inexact harmonic extension were used for the coarse space as the one used in definition of \hat{V}_i , i.e., $\hat{E} = \hat{E}$, then the fill-in in the coarse space matrix is not reduced, but we obtain a better condition number estimate

$$\text{cond}(\mathcal{M}, A) \leq C \left(1 + \left(\frac{q}{h}\right)^2\right)^2 \sup_{\tilde{v}_i \in \tilde{V}_i} \frac{\sum_{j=1}^J \|\bar{N}_j \sum_{i=1}^J \bar{N}_i D_i (I - P_i) \tilde{v}_i\|_{S_j}^2}{\sum_{i=1}^J \|(I - P_i) \tilde{v}_i\|_{S_i}^2}.$$

We sum up the results of this section in the following

Theorem 4.5.10. If the same inexact harmonic extension were used for the coarse space as the one used in definition of \hat{V}_i , i.e., $\hat{E} = \hat{E}$, then

$$\text{cond}(\mathcal{M}, A) \leq C \left(1 + \left(\frac{q}{h}\right)^2\right)^2 \left(1 + \log\left(\frac{H}{h}\right)\right)^2.$$

Proof. The proof follows from Lemma 4.5.8 and the condition number estimate for Balancing Domain Decomposition given in Theorem 2.7.10. \square

4.5.5 Computational Complexity

Let n denote the number of degrees of freedom in the (unreduced) system; N_{es} be the typical number of elements per subdomain, and d the spatial dimension. For the sake of simplicity, we will analyze the computational complexity only in the two limit cases: the case of implementation on a serial architecture, and the case of implementation on an ideal parallel computer (i.e., having as many processors as we desire). We assume that the kernel dimensions of local stiffness matrices are uniformly bounded. Also, in the case of exact components, we make an assumption that $1 + \log(\frac{H}{h})$ is bounded by a constant. In view of Theorems 2.7.10, 4.5.10 and 1.4.1 this means that we have to perform only $O(1)$ iterations to reduce the error to a desired size.

Under these assumptions, let us summarize the computational requirements of these three methods: the original BDD, the method on long vectors with exact components, denoted as ACDD(∞), and the method on long vectors with all components approximate, denoted as ACDD(k).

BDD requires:

- $O(\frac{n}{N_{es}} N_{es}^{\frac{3d-2}{d}})$ operations to compute factorizations of local matrices.
- $O(\frac{n}{N_{es}} n^{\frac{d-1}{d}})$ operations to precompute the coarse space basis functions.
- $O(\frac{n}{N_{es}} N_{es}^{\frac{2d-1}{d}})$ operations to assemble the coarse space problem.
- $O((\frac{n}{N_{es}})^{\frac{3d-2}{d}})$ operations to compute the Cholesky factorization of the coarse problem.
- $O(\frac{n}{N_{es}} N_{es}^{\frac{2d-1}{d}})$ operations to compute local subdomain solves.
- $O((\frac{n}{N_{es}})^{\frac{2d-1}{d}})$ operations to compute the coarse level solution.

ACDD(∞) requires:

- $O\left(\frac{n}{N_{es}}N_{es}^{\frac{3d-2}{d}}\right)$ operations to compute factorizations of local matrices.
- $O\left(\frac{n}{N_{es}}N_{es}^{\frac{2d-1}{d}}\right)$ operations to precompute the coarse space basis functions.
- $O(n)$ operations to assemble the coarse space problem.
- $O\left(\left(\frac{n}{N_{es}}\right)^{\frac{3d-2}{d}}\right)$ operations to compute the Cholesky factorization of the coarse problem.
- $O\left(\frac{n}{N_{es}}N_{es}^{\frac{2d-1}{d}}\right)$ operations to compute local subdomain solves and harmonic extensions.
- $O\left(\left(\frac{n}{N_{es}}\right)^{\frac{2d-1}{d}}\right)$ operations to compute the coarse level solution.

Balancing the amount of work required, we obtain after trivial calculations that for the serial version of both the BDD and ACDD(∞) methods, optimal size of subdomains is $N_{es} = n^{\frac{2d-2}{5d-4}}$, i.e., in 2D $N_{es}^{opt} = n^{1/3}$, resulting in the computational complexity $O(n^{4/3})$, and in 3D $N_{es}^{opt} = n^{4/11}$, resulting in the overall computational complexity of $O(n^{49/33})$.

If a computer with sufficiently many processors were available, the computation of a coarse space basis as well as all the subdomain factorizations and solves could be performed in parallel. Then the optimal size of a subdomain would be $N_{es}^{opt} = n^{1/2}$ in both 2D and 3D, resulting in overall cost of $O(n)$ in 2D and $O(n^{7/6})$ in 3D.

We have shown that the complexity will - even in serial implementation - be lower than just the back-substitution step of a direct method based on a factorization, which is $O(n^{3/2})$, $O(n^{5/3})$ in 2D and 3D, respectively.

ACDD(k) requires:

- $O(n)$ operations to setup iterations for local matrices.
- $O\left(\frac{n}{N_{es}}N_{es}\right)$ operations to precompute the coarse space basis functions.

- $O(n)$ operations to assemble the coarse space problem from the pre-computed basis.
- $O\left(\left(\frac{n}{N_{es}}\right)^{\frac{3d-2}{d}}\right)$ operations to compute the Cholesky factorization of the coarse problem.
- $O\left(\frac{n}{N_{es}}N_{es}\right)$ operations to compute local subdomain solves and harmonic extensions.
- $O\left(\frac{n}{N_{es}}\right)$ operations to compute the coarse level solution.

For ACDD(k), assuming that the number of iterations needed to reduce the error to truncation level is $O(1)$, we obtain on a single processor optimal size of subdomains $N_{es} = n^{\frac{2d-2}{3d-2}}$, i.e., in 2D $N_{es}^{opt} = n^{\frac{1}{2}}$, and in 3D $N_{es}^{opt} = n^{\frac{4}{7}}$, yielding complexity $O(n)$ in both cases. We have thus proved the following theorem.

Theorem 4.5.11. In order to solve the discrete problem to the truncation level, by the ACDD(∞) method with exact components on a serial architecture, we need to perform $O(n^{4/3})$ and $O(n^{49/33})$ operations in 2D and 3D, respectively. Parallel implementation on a computer with about $n^{1/2}$ processors reduces these estimates to $O(n)$, $O(n^{7/6})$ per processor, respectively. For the ACDD(k) method with inexact components, the amount of work on a serial machine is $O(n)$ in both 2D and 3D.

5. Two-Level Multigrid Alternative

5.1 Alternative For Inexact Solvers

In Chapter 3 and Chapter 4 we devised overlapping and substructuring domain decomposition methods allowing for use of inexact subdomain solvers. In this section, we describe another way to implement a domain decomposition algorithm avoiding inexact solvers in the traditional sense altogether. Although an independent method, it shares some components with the method proposed in Section 3.2. Both methods are based on the concept of smoothed transfer operator introduced in the current form by Vaněk et al. in [87]. Its analysis is based on simple algebraic arguments and is close to that of a two-level multigrid.

The input data of the method are the system of linear algebraic equations (1.5) with a symmetric positive definite matrix A_τ resulting from discretization of problem (1.4) on conforming $P1$ or $Q1$ finite elements with the mesh \mathcal{T}_h ,

$$A_\tau x_\tau = b_\tau, \tag{5.1}$$

and a system of J closed disjoint subdomains $\{\Omega_i\}_{i=1}^J$ on Ω such that each subdomain Ω_i is a closed aggregate of elements. We assume that each node of the underlying finite element mesh belongs to exactly one of these subdomains. Thus the system covers all the nodes of the finite element mesh on the domain Ω . Note that we do not require the set of subdomains to cover the entire Ω . No other input is required for solution of the scalar second order elliptic problems. Simple

modifications necessary for adaptation of the method to problems of elasticity will be noted.

In order to eliminate the undesirable influence of possible variation of coefficients in the problem, instead of solving problem (5.1), we will formulate our iterative method for the problem

$$Ax = b \tag{5.2}$$

with the diagonally scaled matrix

$$A = D_\tau^{-1/2} A_\tau D_\tau^{-1/2},$$

where $D_\tau = \text{diag}(A_\tau)$ is the diagonal of A_τ .

This diagonal scaling reduces the spectral condition number of the problem (cf. [40]) and nondimensionalizes the system of equations [48]. This last aspect is especially useful when solving problems mixing degrees of freedom having different physical interpretation, such as the problems of plates and shells (cf. [56]). As the new matrix A is independent of the scaling of basis functions, we may assume without loss of generality that the functions of our finite element basis satisfy

$$\|\varphi_i\|_{L_\infty} = 1. \tag{5.3}$$

5.2 Tentative Prolongator and Standard Two-level Multigrid

Choosing a prolongator P and some pre- and post-smoothers $\mathcal{S}_S, \mathcal{S}'_S$, the classic two-level variational method may be written as follows:

Algorithm 15 ([45]). For the given initial guess $x \in \mathbb{R}^n$,
repeat

1. $x \leftarrow \mathcal{S}_S(x, b)$,
 2. solve the coarse level problem $P^T A P v = P^T (Ax - b)$,
 3. $x \leftarrow x - P v$,
 4. $x \leftarrow \mathcal{S}_{S'}(x, b)$
- until convergence;

An appropriate choice of the components, most importantly the prolongation operator P , is the key to the efficiency of the two-grid method.

Let us first investigate the method with a prolongator given by unknowns aggregation. Using a system of nonoverlapping subdomains and the aggregation technique (cf. Section 3.2), we will construct a tentative coarse space of possibly a very small dimension. We label it tentative for the same reasons as we did in Chapter 3, namely because we will ultimately construct an improvement based on this tentative one.

Let N_n denote the number of nodes in the discretization mesh \mathcal{T} , \mathcal{F} the index set of all unconstrained nodes and $N_f = \text{card}(\mathcal{F})$. We introduce a one-to-one mapping $N : \{1, \dots, N_f\} \rightarrow \mathcal{F}$ that establishes the correspondence between the degrees of freedom of the constrained space V_τ and its unconstrained counterpart V ; i.e., for a degree of freedom i in V_τ , $N(i)$ is the number of the corresponding degree of freedom in V . Let us consider the decomposition of unity $\mathbf{u}^1 \in \mathbb{R}^{N_n}$ defined by

$$\sum_{i=1}^{N_n} u_i^1 \varphi_i = 1 \quad \text{on } \Omega. \quad (5.4)$$

Note that in practice the finite element bases for solving second order elliptic problems often satisfy $u_i^1 = 1$, $i = 1, \dots, N_n$. As noted, the subdomains Ω_i do

not cover the entire Ω , but since they contain all the nodes of the mesh, we can define a $N_f \times J$ tentative prolongator matrix \hat{P} by the following construction:

$$\left. \begin{aligned} \tilde{P}_{ij} &= \begin{cases} u_{N(i)}^1, & \text{if the node } v_{N(i)} \text{ belongs to subdomain } \Omega_j, \\ 0, & \text{otherwise;} \end{cases} \\ \hat{P} &= D_\tau^{1/2} \tilde{P}. \end{aligned} \right\} \quad (5.5)$$

Note that this version of tentative prolongator differs from the one used in Section 3.2 only by the diagonal scaling.

The following lemma summarizes several useful results well known from the theory of the two-level method [67], [45]. As these are classic results, we shall omit their proof.

Lemma 5.2.1. Let B be a symmetric positive definite matrix on \mathbb{R}^n , $p : \mathbb{R}^n \rightarrow \mathbb{R}^m$ a full-rank prolongation operator and $T = \text{Ker}(p^T B)$. Then

- (1) $I - p(p^T B p)^{-1} p^T B$ is an B -orthogonal projection onto T ,
- (2) If $S = I - \frac{\omega}{\varrho(B)} B$, $\omega \in (0, 2)$ then for every $x \in \mathbb{R}^n$

$$\|Sx\|_B^2 \leq \|x\|_B^2 - \|Bx\|^2 \cdot \frac{\omega}{\varrho(B)} \cdot (2 - \omega).$$

- (3) $\|S[I - p(p^T B p)^{-1} p^T B]\|_B \leq \|S_T\|_B$, where S_T denotes the restriction of operator S to the set T .
- (4) Let the following weak approximation property be fulfilled :

there exists a constant $C_{apx} > 0$ such that for every $u \in \mathbb{R}^n$ there exists a $v \in \mathbb{R}^m$ such that

$$\|u - pv\| \leq C_{apx} \varrho(B)^{-\frac{1}{2}} \|u\|_B. \quad (5.6)$$

Then

$$\frac{\|Bx\|}{\|x\|_B} \geq C_{apx}^{-1} \varrho(B)^{\frac{1}{2}} \quad \text{for every } x \in T. \quad (5.7)$$

Corollary 5.2.2. If we choose the damped Jacobi method $I - \frac{\omega}{\varrho(A)}A$ as the post-smoother in the two-level Algorithm 15, we obtain

$$\|e_{i+1}\|_A^2 \leq \left(1 - \left(\frac{1}{C_{apx}}\right)^2 \omega(2 - \omega)\right) \|e_i\|_A^2.$$

Remark 5.2.3. Note that the constant C_{apx} typically depends on the ratio of the sizes of fine and coarse space, H/h .

The following theorem summarizes the convergence properties for the two-level method with the prolongator given by unknowns aggregation.

Theorem 5.2.4. For the two-level method given by Algorithm 15 with the prolongator $P = \hat{P}$ given by a mere unknowns aggregation (5.5), and the damped Jacobi post-smoother, applied to solving the problem (5.2), the error estimate is

$$\|e_{i+1}\|_A^2 \leq \left(1 - C\left(\frac{h}{H}\right)^2\right) \|e_i\|_A^2.$$

Proof. In view of Corollary 5.2.2 we only need to evaluate the constant in the weak approximation property (5.6). Let Π denote the finite element interpolation operator:

$$\text{for a vector } \alpha, \quad \Pi u = \sum \alpha_i \phi_i.$$

We will construct a vector v having components

$$v_i = \frac{1}{\text{meas}(\Omega_i)} \int_{\Omega_i} \Pi u \, dx.$$

From the definition (5.5) of the aggregation prolongator \hat{P} it follows that $(Pv)_j = v_i$, for every node $v_j \in \bar{\Omega}_i$. Using the equivalence of Euclidean and continuous L^2 norms, we have

$$\begin{aligned} \|u - Pv\|^2 &= \sum_{i=1}^J \|u - Pv\|_{l^2(F(\Omega_i))}^2 \\ &= \sum_{i=1}^J \|u - v_i \mathbf{1}_i\|_{l^2(F(\Omega_i))}^2 \\ &\approx h^{-d} \sum_{i=1}^J \|\Pi(u - v_i)\|_{L^2(\Omega_i)}^2, \end{aligned}$$

so the Poincaré inequality (A.4) yields

$$\begin{aligned} \|u - Pv\|^2 &\leq \frac{C}{h^d} \sum_{i=1}^J \|\Pi u\|_{H^1(\Omega_i)}^2 H^2 \\ &\leq C \left(\frac{H}{h}\right)^2 h^{2-d} \|\Pi u\|_{H^1(\Omega)}^2 \\ &\leq C \left(\frac{H}{h}\right)^2 \varrho(A)^{-1} \|u\|_A^2. \end{aligned}$$

The rest of the proof follows from Corollary 5.2.2. \square

Computational experiments suggest that the estimate stated in Theorem 5.2.4 is sharp. This fact had lead to a commonly accepted point of view that an algebraic method utilizing only two grids cannot be a basis of an efficient solver offering simultaneously low computational complexity and the rate of convergence independent of the ratio H/h .

Indeed, in order to preserve favorable rate of convergence, the estimate of Theorem 5.2.4 excludes the possibility of having a two-level method with a small coarse problem. This means that solving a problem using two-level method under this limitation results in asymptotically the same complexity as solving the original problem by a direct solver, a procedure hardly worth the effort of

implementing.

In the next section we will show how to modify the two-level method to defeat this difficulty.

5.3 Modified Two-level Multigrid and MLS

In the previous section we have seen that the convergence of the algorithm based on the aggregation of unknowns alone is unsatisfactory. Because of other practical advantages aggregation has to offer, various attempts to improve the convergence have been made (e.g., supercorrection [6, 7]). These attempts did not, however, solve the cause of poor behavior inherent to the problem, which is the fact that the range of the prolongation based on aggregation consists of functions with high energy, namely the piecewise constant functions. This section describes a modified two-level substructuring method, the result of joint work with Jitka Křížková, Radek Tezaur and Petr Vaněk [88], for solving scalar elliptic problems with jumps in coefficients. We improve the convergence estimate obtained from the standard two-level multigrid theory using an algebraic lift made possible by using the multilevel smoothing in the definition of the prolongator operator.

As we aim at achieving good computational complexity, reducing the dependence of the rate of convergence on the coarse space size becomes our main objective. To this end, we will tailor our pre-smoother, post-smoother and coarse space so that the two-level method based on their combination is capable of effectively eliminating all components of the error, provided that the pre-smoother distributes the information to the distance comparable with the

coarse space resolution.

We will utilize the tentative prolongator constructed in (5.5) and apply to it a specially selected prolongator smoother. This will produce a smoothing effect on the range of the tentative prolongator. Our design choice is to take the linear part of the pre-smoother to play the role of prolongator smoother. Our post-smoother will also be derived from the pre-smoother. This results in a flexible procedure which we can control by a choice of the pre-smoother. We will prove under regularity-free assumptions that proper selection of the pre-smoother results in the rate of convergence independent of the size of the coarse space.

The method is determined by two components : a full-rank tentative prolongator $\hat{P} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ ($m \ll n = \text{rank}(A)$) and a pre-smoother, from which both the post-smoother and a prolongator smoother are derived. We will show that the proposed method is, up to a post-processing step, the standard variational two-level scheme with a smoothed prolongator $\mathcal{S}P$ and special smoothing procedures.

Let $x \leftarrow \mathcal{S}_S(x, b)$ be a given pre-smoother, a linear iterative method consistent with (5.2), with a symmetric linear part \mathcal{S} commuting with A . We select \mathcal{S} to be our prolongator smoother. Let us define the post-smoother to be a linear iterative method consistent with (5.2) such that its error propagation operator is the matrix \mathcal{S}' defined by

$$\mathcal{S}' = I - \frac{\omega}{\bar{\varrho}(A_S)} A_S, \quad \text{where} \quad A_S = \mathcal{S}^2 A. \quad (5.8)$$

Scalar $\omega \in (0, 2)$ is a given parameter and $\bar{\varrho}(A_S)$ is an upper bound of the spectral radius $\varrho(A_S)$. Note that all the above can be easily accomplished if \mathcal{S}

is a polynomial in A . The algorithm can be written down as follows:

Algorithm 16 (MLS). For the given initial guess $x \in \mathbb{R}^n$,

repeat

1. $x \leftarrow \mathcal{S}_{\mathcal{S}}(x, b)$,
2. solve the coarse level problem $P^T \mathcal{S} A S P v = P^T \mathcal{S} (A x - b)$,
3. $x \leftarrow x - \mathcal{S} P v$,
4. $x \leftarrow \mathcal{S}_{\mathcal{S}'}(x, b)$

until convergence;

5. Post process $x \leftarrow \mathcal{S}_{\mathcal{S}}(x, b)$.

Remark 5.3.1. a) Steps 1–4 of the algorithm form the standard multi-grid two-level method given by prolongator $\mathcal{S}P$ and smoothers $\mathcal{S}_{\mathcal{S}}$, $\mathcal{S}_{\mathcal{S}'}$ (cf. the proof of Theorem 5.3.5 below). For such an algorithm, our theory gives an error estimate in the $A_{\mathcal{S}}$ –seminorm. However, using a smaller coarse space calls for a more powerful smoother \mathcal{S} to obtain the optimal convergence result, so $A_{\mathcal{S}}$ depends on the coarse space, and the practical value of such an estimate is questionable.

b) From the convergence point of view, the postprocessing step 5 consisting of a single smoothing seems out of place. But it is the addition of this very step that allows us to provide the same convergence estimate in the energy norm of the original problem (5.2). In order to demonstrate this, let e_i denote the error after i iterations given by steps 1 through 4 of Algorithm 16, and $e_i^{\mathcal{S}}$ the error after the application of the postprocessing step 5 to e_i , i.e., $e_i^{\mathcal{S}} = \mathcal{S}e_i$. Then, as $A_{\mathcal{S}} = \mathcal{S}^2 A$, $\varrho(\mathcal{S}) < 1$, we have $\|e_0\|_{A_{\mathcal{S}}} \leq \|e_0\|_A$ and $\|e_i^{\mathcal{S}}\|_A = \|e_i\|_{A_{\mathcal{S}}}$.

The last two relationships yield the \mathcal{S} -independent convergence rate estimate

$$\|e_i^{\mathcal{S}}\|_A^2 \leq (1 - C)^i \|e_0\|_A^2, \quad (5.9)$$

provided we can estimate

$$\|e_{i+1}\|_{A_{\mathcal{S}}}^2 \leq (1 - C) \|e_i\|_{A_{\mathcal{S}}}^2. \quad (5.10)$$

Remark 5.3.2. In Chapter 6, we will refer to the practical implementation of the method from Algorithm 16, with suitable choice of the components described below as MLS, an abbreviation standing for Black-box Nonoverlapping Method with Multilevel Smoothing.

The following assumption specifies the requirements on \mathcal{S} , p and $\bar{\varrho}(A_{\mathcal{S}})$ in the form suitable for our purposes.

Assumption 5.3.3. There exist positive constants C_1, C_2 , independent of m, n , and constant $C_D(m, n)$ such that:

1. There is a mapping $Q_c : \mathbb{R}^n \rightarrow \text{Range}(P)$ such that

$$\|(I - Q_c)u\| \leq C_1 C_D(m, n) \varrho^{-\frac{1}{2}}(A) \|u\|_A \quad \forall u \in \mathbb{R}^n. \quad (5.11)$$

2. The prolongator smoother \mathcal{S} is symmetric, commutes with A , satisfies $\varrho(\mathcal{S}) \leq 1$ and the smoothing property of Hackbusch in the form

$$\varrho(\mathcal{S}^2 A) \leq \bar{\varrho}(\mathcal{S}^2 A) \leq C_2^2 C_D^{-2}(m, n) \varrho(A). \quad (5.12)$$

Remark 5.3.4. We note that (5.6) follows from (5.11). Adding the requirement (5.12) is necessary to guarantee improved convergence. A pair of requirements similar to (5.11), (5.12) is very common in the multigrid theory [45, 14]. In our estimates, the purpose of the constant $C_D(m, n)$ is to absorb

the dependence of the estimates (5.11) and (5.12) on dimensions m, n . When verifying the assumption above, the goal is to show that constants C_1 and C_2 are either m, n -independent, or depend on m and n only weakly (e.g. polylogarithmically.) As the convergence rate estimate will turn out to depend only on the constants C_1, C_2 , it will thus be m, n independent.

Let us recall Algorithm 6 of Section 3.2, where we have constructed the prolongator smoother \mathcal{S} to be a polynomial in A satisfying (Lemma 3.3.2)

$$\varrho(\mathcal{S}^2 A) \leq \frac{C}{\deg^2(\mathcal{S})} \varrho(A).$$

Typically, for second order elliptic problems, it is possible to prove the weak approximation property in the form

$$\|(I - Q_c)u\| \leq C \frac{H}{h} \varrho^{-\frac{1}{2}}(A) \|u\|_A,$$

where h is the fine space characteristic resolution (meshsize) and H is the tentative coarse space resolution (we have obtained this result in the proof of Theorem 5.2.4). Then, when choosing \mathcal{S} of degree at least CH/h , we can set $C_D(m, n) = H/h$ and the constants C_1 and C_2 are independent of H, h , enabling us to prove coarse space size independent convergence.

Let us recall that ω is the damping parameter from the definition (5.8) of \mathcal{S}' . We now formulate the abstract convergence estimate.

Theorem 5.3.5. Let e_i denote the error after i iterations given by steps 1–4 of Algorithm 16, and let $e_i^{\mathcal{S}}$ be the error e_i smoothed by step 5. Then, under Assumption 5.3.3, we have the following error estimate:

$$\|e_i^{\mathcal{S}}\|_A^2 \leq (1 - C_3)^i \|e_0\|_A^2, \tag{5.13}$$

where

$$C_3(\omega) = \frac{(C_1 C_2)^{-2} \omega (2 - \omega)}{1 + (C_1 C_2)^{-2} \omega (2 - \omega)} \leq C_3(1) = \frac{(C_1 C_2)^{-2}}{1 + (C_1 C_2)^{-2}}, \quad \omega \in (0, 2). \quad (5.14)$$

Proof. In view of Remark 5.3.1 b), it suffices to prove the estimate

$$\|e_{i+1}\|_{A_S}^2 \leq (1 - C_3) \|e_i\|_{A_S}^2 \quad (5.15)$$

for the error of iterations without the final smoothing step.

As the first step in this proof, we will adopt a different view of our method, namely we will demonstrate that our two-level method with the smoothed prolongator $\mathcal{S}\hat{P}$, pre-smoother \mathcal{S} and post-smoother \mathcal{S}' applied to the problem with the matrix A can be viewed as a two-level method with tentative prolongator \hat{P} applied to the problem with a “smoothed” matrix $A_S = A\mathcal{S}^2$.

First define $Q_S : \mathbb{R}^n \rightarrow \text{Ker}(\mathcal{S})^\perp$ to be the orthogonal projection with respect to the Euclidean inner product. Note that \mathcal{S} may be singular. Since \mathcal{S} commutes with A , the eigenvectors of \mathcal{S} and A_S coincide. Consequently, Q_S is A_S -symmetric and $\|Q_S\|_{A_S} = 1$. It is routine to derive that the linear part of the steps 1–4 is given by

$$\mathcal{S}'[I - \mathcal{S}P(P^T A_S P)^+ P^T \mathcal{S}A]\mathcal{S} = \mathcal{S}'\mathcal{S}Q_S[I - P(P^T A_S P)^+ P^T A_S]Q_S, \quad (5.16)$$

where $(P^T A_S P)^+$ is a pseudo-inverse of $P^T A_S P$. Since

$$\text{Ker}(P^T A_S P) = \{x \in \mathbb{R}^m : Px \in \text{Ker}(\mathcal{S})\},$$

the algorithm is independent of a particular choice of the pseudo-inverse. Thus, the method can alternatively be viewed as a standard two-level method for solving a problem with matrix A_S (in place of A) and prolongator P (in place of

$\mathcal{S}P$.) Adopting this view will aid us in estimating the right hand side in (5.16) in the $A_{\mathcal{S}}$ -operator matrix norm.

Let us define $T = \text{Ker}(P^T A_{\mathcal{S}}) \cap \text{Ker}(\mathcal{S})^-$ and consider the $A_{\mathcal{S}}$ -orthogonal decomposition

$$\text{Ker}(\mathcal{S})^- = T \oplus \text{Range}(Q_{\mathcal{S}}P).$$

Consider the coarse level correction part \mathcal{P} of the error propagation operator on the right-hand side of (5.16),

$$\mathcal{P} = Q_{\mathcal{S}}[I - P(P^T A_{\mathcal{S}}P)^+ P^T A_{\mathcal{S}}]Q_{\mathcal{S}}.$$

It is easy to see that \mathcal{P} restricted to $\text{Ker}(\mathcal{S})^-$ is an $A_{\mathcal{S}}$ -orthogonal projection and

$$\text{Range}(\mathcal{P}) \subset T.$$

Further, as \mathcal{S} commutes with A , \mathcal{S}' commutes with \mathcal{S} . Therefore, taking into account that $\varrho(\mathcal{S}) \leq 1$, $\varrho(\mathcal{S}') \leq 1$, $\|Q_{\mathcal{S}}\|_{A_{\mathcal{S}}} = 1$ and $\|\mathcal{P}|_{\text{Ker}(\mathcal{S})^-}\|_{A_{\mathcal{S}}} = 1$ we have

$$\begin{aligned} \|\mathcal{S}'\mathcal{S}Q_{\mathcal{S}}[I - P(P^T A_{\mathcal{S}}P)^+ P^T A_{\mathcal{S}}]Q_{\mathcal{S}}\|_{A_{\mathcal{S}}}^2 &= \|\mathcal{S}'\mathcal{S}\mathcal{P}Q_{\mathcal{S}}\|_{A_{\mathcal{S}}}^2 \\ &\leq \|(\mathcal{S}'\mathcal{S})|_T\|_{A_{\mathcal{S}}}^2 \|\mathcal{P}|_{\text{Ker}(\mathcal{S})^-}\|_{A_{\mathcal{S}}}^2 \|Q_{\mathcal{S}}\|_{A_{\mathcal{S}}}^2 \\ &= \|(\mathcal{S}'\mathcal{S})|_T\|_{A_{\mathcal{S}}}^2 \\ &\leq \sup_{\substack{x \in T \\ x \neq 0}} \min \left\{ \frac{\|\mathcal{S}x\|_{A_{\mathcal{S}}}^2}{\|x\|_{A_{\mathcal{S}}}^2}, \frac{\|\mathcal{S}'x\|_{A_{\mathcal{S}}}^2}{\|x\|_{A_{\mathcal{S}}}^2} \right\}. \end{aligned} \quad (5.17)$$

The rest of the proof consists in demonstrating that at least one of the expressions in the minimum above is bounded by $1 - C_3$ for any $x \in T \setminus \{0\}$, with C_3 defined by (5.14).

We will now show that for our definition of \mathcal{S}' the following algebraic lift holds:

$$\frac{\|\mathcal{S}'x\|_{A_S}^2}{\|x\|_{A_S}^2} \leq 1 - \frac{\|\mathcal{S}x\|_{A_S}^2}{\|x\|_{A_S}^2} (C_1 C_2)^{-2} \omega (2 - \omega), \quad (5.18)$$

where $x \in \text{Range}(\mathcal{P}) \subset T$ is an error lying in the range of the coarse grid correction operator. In other words, we will show that if (for the particular $x \in T$) This lift guarantees for a particular error $x \in T$ that if \mathcal{S} is inefficient in reducing the error x ($\|\mathcal{S}x\|_{A_S}/\|x\|_{A_S} \geq q$, $q \in (0, 1]$), then \mathcal{S}' will be efficient in the sense that

$$\frac{\|\mathcal{S}'x\|_{A_S}^2}{\|x\|_{A_S}^2} \leq 1 - q^2 (C_1 C_2)^{-2} \omega (2 - \omega).$$

Let us define

$$\beta = \frac{\bar{\varrho}(A_S)}{\varrho(A_S)}.$$

By Assumption 5.3.3, we have $\beta \geq 1$. Simple manipulations yield

$$\begin{aligned} \|\mathcal{S}'x\|_{A_S}^2 &= \|(I - \frac{\omega}{\bar{\varrho}(A_S)} A_S)x\|_{A_S}^2 = \|(I - \frac{\omega}{\beta} \varrho^{-1}(A_S) A_S)x\|_{A_S}^2 \\ &= \|x\|_{A_S}^2 - 2\frac{\omega}{\beta} \varrho^{-1}(A_S) \|A_S x\|^2 + \left(\frac{\omega}{\beta} \varrho^{-1}(A_S)\right)^2 \|A_S x\|_{A_S}^2 \\ &\leq \|x\|_{A_S}^2 - 2\frac{\omega}{\beta} \varrho^{-1}(A_S) \|A_S x\|^2 + \left(\frac{\omega}{\beta}\right)^2 \varrho^{-1}(A_S) \|A_S x\|^2 \\ &= \|x\|_{A_S}^2 \left[1 - \frac{\omega}{\beta} \left(2 - \frac{\omega}{\beta}\right) \varrho^{-1}(A_S) \frac{\|A_S x\|^2}{\|x\|_{A_S}^2}\right]. \end{aligned}$$

Consequently

$$\frac{\|A_S x\|^2}{\|x\|_{A_S}^2} \geq \hat{C}(x) \varrho(A_S) \quad \text{implies} \quad \frac{\|\mathcal{S}'x\|_{A_S}^2}{\|x\|_{A_S}^2} \leq 1 - \hat{C}(x) \frac{\omega}{\beta} \left(2 - \frac{\omega}{\beta}\right). \quad (5.19)$$

Let us recall that $A_S = \mathcal{S}A\mathcal{S}$, and \mathcal{S} commutes with A . Therefore we obtain

$$\|\mathcal{S}^2 x\|_A^2 = \|\mathcal{S}x\|_{A_S}^2, \quad (5.20)$$

$$\frac{\|A_S x\|^2}{\|x\|_{A_S}^2} = \frac{\|A_S x\|^2}{\|\mathcal{S}^2 x\|_A^2} \frac{\|\mathcal{S}^2 x\|_A^2}{\|x\|_{A_S}^2} = \frac{\|A \mathcal{S}^2 x\|^2}{\|\mathcal{S}^2 x\|_A^2} \frac{\|\mathcal{S} x\|_{A_S}^2}{\|x\|_{A_S}^2}. \quad (5.21)$$

Now, consider $x \in T \subset \text{Ker}(P^T A \mathcal{S}^2)$. Then, setting $u = \mathcal{S}^2 x$, we have $u \in \text{Ker}(P^T A) = \text{Range}(P)^{-A}$, where $-_A$ denotes the A -orthogonal complement of a set. For operator Q_c satisfying the weak approximation property (5.11), we estimate the ratio $\|A \mathcal{S}^2 x\|^2 / \|\mathcal{S}^2 x\|_A^2$ using the standard orthogonality argument:

$$\begin{aligned} \|u\|_A^2 &= \langle Au, u \rangle \\ &= \langle Au, u - Q_c u \rangle \\ &\leq \|Au\| \|u - Q_c u\| \\ &\leq C_1 C_D(m, n) \varrho^{-1/2}(A) \|Au\| \|u\|_A, \end{aligned}$$

which, in view of (5.20), means that

$$\frac{\|A \mathcal{S}^2 x\|}{\|\mathcal{S}^2 x\|_A} = \frac{\|Au\|}{\|u\|_A} \geq C_1^{-1} C_D^{-1}(m, n) \varrho^{1/2}(A).$$

Substituting this estimate into (5.21), using Assumption 5.3.3 and the definition of β , we obtain

$$\frac{\|A_S x\|^2}{\|x\|_{A_S}^2} \geq C_1^{-2} C_D^{-2}(m, n) \varrho(A) \frac{\|\mathcal{S} x\|_{A_S}^2}{\|x\|_{A_S}^2} \geq (C_1 C_2)^{-2} \beta \varrho(A_S) \frac{\|\mathcal{S} x\|_{A_S}^2}{\|x\|_{A_S}^2}. \quad (5.22)$$

Thus, as $\beta \geq 1$, by (5.19) with $\hat{C}(x) = (C_1 C_2)^{-1} \beta \|\mathcal{S} x\|_{A_S}^2 / \|x\|_{A_S}^2$, we have

$$\begin{aligned} \frac{\|\mathcal{S}' x\|_{A_S}^2}{\|x\|_{A_S}^2} &\leq 1 - \frac{\|\mathcal{S} x\|_{A_S}^2}{\|x\|_{A_S}^2} (C_1 C_2)^{-2} \omega \left(2 - \frac{\omega}{\beta} \right) \\ &\leq 1 - \frac{\|\mathcal{S} x\|_{A_S}^2}{\|x\|_{A_S}^2} (C_1 C_2)^{-2} \omega (2 - \omega), \end{aligned}$$

completing the proof of (5.18).

Finally, due to (5.16), (5.17), (5.18) and $\|\mathcal{S}x\|_{A_S}^2/\|x\|_{A_S}^2 \in [0, 1]$, we may write

$$\begin{aligned} \|\mathcal{S}'[I - \mathcal{S}P(P^T A_S P)^+ P^T \mathcal{S}A]\mathcal{S}\|_{A_S}^2 &= \|\mathcal{S}'\mathcal{S}Q_S[I - P(P^T A_S P)^+ P^T A_S]Q_S\|_{A_S}^2 \\ &\leq \sup_{\alpha \in [0,1]} \min \left\{ \alpha, 1 - \alpha(C_1 C_2)^{-2} \omega(2 - \omega) \right\}. \end{aligned}$$

The expression on the right hand side is bounded by $[1 + (C_1 C_2)^{-2} \omega(2 - \omega)]^{-1}$ which completes the proof of the theorem. \square

5.4 Practical Issues

5.4.1 Generalizations

Although we have formulated the method for scalar problems, it is not difficult to generalize it for solving nonscalar ones, such as 2D and 3D elasticity or the plate and shell problems. This can be done in the same manner as described in Section 3.4.2, with only one difference, namely that we now need to add the (block) diagonal scaling.

We note that MLS was formulated with second order problems in mind. Its use for solving higher order problems, although possible, is unwarranted by the theory. From the practical point of view, the smoothing procedure based on matrices resulting from discretizations of higher order problems would generate too extensive overlapping of the smoothed coarse space basis function supports and produce rapid fill-in in the coarse problem. A true algebraic multigrid using the concept of smoothed transfer operators, suitable for solving higher order problems, was proposed in Vaněk, Mandel and Brezina [87].

5.4.2 Computational Complexity

The following theorem gives the computational complexity estimate for the two-level method with smoothed prolongator implemented on a serial architecture.

Theorem 5.4.1. Let the assumptions of Theorem 5.3.5 be fulfilled and the Cholesky factorization be used to solve the coarse-level problem. Then the optimal number of elements per subdomain using MLS in 3D is $N_{es} \approx n^{1/2}$ and the system (5.2) can be solved to the level of truncation error in $O(n^{7/6})$ operations. In 2D, the optimal number of elements per subdomain is $N_{es} \approx n^{2/5}$ and the system (5.2) can be solved to the level of truncation error in $O(n^{6/5})$ operations.

Proof. The setup phase requires $O(N_{es}^{1/d}n)$ operations to construct the smoothed prolongator $P = \mathcal{S}\hat{P}$, $O(n)$ operations to construct the coarse level matrix P^TAP , and $O((\frac{n}{N_{es}})^{\frac{3d-2}{d}})$ operations to compute the Cholesky factorization of the coarse matrix.

Each iteration will require $O(N_{es}^{1/d}n)$ operations to perform the smoothing, $O(n)$ operations to evaluate the prolongation, restriction, and compute the defect, and $O((\frac{n}{N_{es}})^{\frac{2d-1}{d}})$ to compute the back-substitution.

Theorem 5.3.5 guarantees that the number of iterations needed to converge to a required precision is $O(1)$. Minimizing the overall expense, we obtain the estimates of the theorem. \square

Although MLS can take advantage of parallel implementation and the iteration time can be reduced dramatically, its lack of locality of data prevents asymptotic improvements in computational complexity, which remains $O(n^{7/6})$

operations in 3D and $O(n^{6/5})$ operations in 2D. Despite this, Theorem 5.4.1 shows that of all the methods we have considered MLS has the lowest computational complexity. In 3D, MLS' asymptotic computational complexity equals that of the massively parallel implementations of both BOSS (Theorem 3.4.2) and ACDD(∞) (Theorem 4.5.11).

6. Numerical Experiments

We now present the results of numerical experiments. We begin with a rather academic set of problems and proceed to the practically more interesting problems on “real-world” geometries. In all the problems below, the methods BDD, ACDD, BOSS and MLS described in the text are applied as a preconditioner \mathcal{M}^{-1} in the Orthomin implementation of conjugate gradient method (Algorithm 2). The stopping criterion used was

$$\text{cond}(\mathcal{M}^{-1}A) \frac{\langle \mathcal{M}^{-1}r_i, r_i \rangle}{\langle \mathcal{M}^{-1}b, b \rangle} \leq \epsilon^2,$$

where $\epsilon = 10^{-6}$ for all the problems except when explicitly specified otherwise. Note that this criterion guarantees the energy norm error estimate $\frac{\|e_i\|_A}{\|x^*\|_A} \leq \epsilon$ (cf. Section B.1; notes on estimating the condition number can also be found there).

For comparison of the methods, we report the condition number estimate for the preconditioned system and the number of iterations required to satisfy the stopping criterion. The CPU times are not reported, as this information could be misleading due to varying maturity of the code for the methods.

Before presenting the results of the tests, let us note that the weight matrices D_i used in ACDD Algorithm 14 were chosen to be the diagonal entries of local stiffness matrices A_i corresponding to the interface degrees of freedom.

6.1 Model Problems

The original BDD method is known to be very robust with respect to variation in the coefficient values of problem (1.2). We have conducted a series of experiments in an attempt to determine how well will this robustness be preserved with respect to changing number of iterations used to approximate the exact local subdomain inverses used in ACDD. We denote the method with approximate local solvers by ACDD(k), where k is the number of iterations performed on each subdomain.

As we are interested in the behavior with respect to varying coefficients, the problem was set up on a simple cubical domain subdivided into 125 subdomains with uniform discretization mesh. The global number of degrees of freedom in the discretized system was 68921. The iteration we chose for approximating the local inverses was SSOR. Implementation of an algebraic multigrid procedure to replace SSOR is currently in progress; it is expected to yield much improved results.

In the first set of experiments, we solve the weighted Poisson equation on the given domain. The jumps in coefficients follow the checkerboard pattern depicted in Figure 6.1. Table 6.1 contains the number of iterations required to reduce the relative residual to $\varepsilon = 10^{-7}$. In the case of Poisson equation ($\alpha_{1,2} = 1$), if only one iteration of SSOR is performed, the method converges in 23 iterations. This is an almost fivefold increase compared to BDD, but still a significant improvement over the diagonally preconditioned conjugate gradient method, which failed to converge in 300 iterations.

Table 6.1. Comparison of BDD with ACDD(k) for different values of k in case of the checkerboard coefficient pattern.

The numbers of iterations required and condition number estimates								
	$\alpha_{1,2} = 1$		$\alpha_{1,2} = 10^{\pm 1}$		$\alpha_{1,2} = 10^{\pm 2}$		$\alpha_{1,2} = 10^{\pm 3}$	
Method used	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
BDD	5	1.18	11	3.07	10	3.06	10	3.05
ACDD(200)	5	1.18	11	3.07	10	3.06	10	3.05
ACDD(150)	6	3.83	11	3.07	10	3.06	10	3.05
ACDD(100)	7	3.89	12	3.07	11	3.06	11	3.05
ACDD(50)	10	3.89	12	3.08	11	3.05	11	3.04
ACDD(40)	11	3.90	12	3.08	12	3.06	11	3.02
ACDD(30)	12	4.11	13	3.14	12	3.09	11	2.97
ACDD(20)	12	4.20	13	2.27	12	3.17	12	3.04
ACDD(10)	13	4.42	14	3.73	13	3.40	12	3.18
ACDD(8)	14	4.53	15	3.96	13	3.50	12	3.20
ACDD(5)	15	4.88	16	4.60	14	3.74	12	3.26
ACDD(4)	15	5.18	17	4.95	14	3.85	13	3.29
ACDD(3)	16	5.75	18	5.47	15	4.01	13	3.34
ACDD(2)	18	6.87	20	6.64	16	4.49	14	3.74
ACDD(1)	23	9.37	25	9.40	19	5.81	17	5.24

Table 6.2 contains the results of solving a problem with randomly distributed coefficients in a given range. The distribution of coefficients is exponential. The results for coefficients in the intervals $[10^{-1}, 10^1]$, $[10^{-2}, 10^2]$ and $[10^{-3}, 10^3]$ are reported. In these experiments, we observe that for coefficients in $[10^{-1}, 10^1]$, we recover convergence properties of BDD if we perform about 10 iterations of SSOR on each subdomain. For coefficients in the interval $[10^{-2}, 10^2]$, we see a similar pattern. Note that the condition number estimates in this case speak in favor of the application of approximate subdomain solvers. If only one iteration on each subdomain is performed, the number of iterations needed to reduce the relative residual to 10^{-6} is less than twice that of BDD. For large variation in the coefficients, the convergence properties of ACDD(\cdot) appear to be superior to those of BDD.

Table 6.3 contains the results of solving a problem with randomly distributed coefficients in a given range. Here the distribution of coefficients is uniform. The double-columns lists the results for coefficients in the intervals $[10^{-1}, 10^1]$, $[10^{-2}, 10^2]$ and $[10^{-3}, 10^3]$, respectively. In these experiments, we observe a similar behavior in all three ranges of coefficients. We recover convergence properties of BDD if we perform only a few iterations of SSOR on each subdomain. If only one iteration on each subdomain is performed, the number of iterations needed to reduce the relative residual to 10^{-6} is less than twice that needed for BDD.

Table 6.2. Comparison of BDD with ACDD(k) for problem with exponentially distributed random coefficients.

The numbers of iterations required and condition number estimates						
	Jumps $10^{-1} - 10^1$		Jumps $10^{-2} - 10^2$		Jumps $10^{-3} - 10^3$	
Method used	Iters.	Cond. est.	Iters.	Cond. est.	Iters.	Cond. est.
BDD	19	5.91	48	94.89	185	2247.80
ACDD(200)	19	5.90	43	47.94	100	292.38
ACDD(150)	19	5.86	42	42.05	97	305.12
ACDD(100)	19	5.78	41	36.12	96	321.66
ACDD(50)	18	5.75	41	30.26	96	367.80
ACDD(40)	18	5.79	41	29.09	97	383.90
ACDD(30)	18	6.03	42	27.94	100	430.36
ACDD(20)	19	5.57	44	28.46	106	517.70
ACDD(10)	21	8.95	47	33.85	116	663.38
ACDD(8)	22	9.93	50	37.63	120	708.81
ACDD(5)	25	12.30	56	47.35	131	807.44
ACDD(4)	26	13.56	59	52.36	139	858.97
ACDD(3)	29	15.35	64	59.83	147	952.00
ACDD(2)	32	18.25	67	78.68	163	1223.60
ACDD(1)	38	25.12	89	130.74	201	1798.80

Table 6.3. Comparison of BDD with ACDD(k) for problem with uniformly distributed random coefficients.

The numbers of iterations required and condition number estimates						
	Jumps $10^{-1} - 10^1$		Jumps $10^{-2} - 10^2$		Jumps $10^{-3} - 10^3$	
Method used	Iter.	Cond. est.	Iter.	Cond. est.	Iter.	Cond. est.
BDD	16	4.38	13	3.25	11	2.59
ACDD(200)	16	4.38	13	3.25	11	2.59
ACDD(150)	16	4.38	13	3.25	11	2.59
ACDD(100)	16	4.38	13	3.24	11	2.60
ACDD(50)	16	4.35	13	3.22	12	2.64
ACDD(40)	15	4.34	13	3.21	12	2.69
ACDD(30)	15	4.34	13	3.23	12	2.79
ACDD(20)	15	4.40	13	3.35	12	3.02
ACDD(10)	15	4.77	14	4.03	13	3.49
ACDD(8)	15	4.98	14	4.33	13	3.63
ACDD(5)	17	5.92	16	4.98	14	3.89
ACDD(4)	18	6.52	16	5.29	15	4.00
ACDD(3)	19	7.37	17	5.71	16	4.21
ACDD(2)	22	8.77	19	6.37	18	5.09
ACDD(1)	26	11.97	23	8.15	22	7.56

Table 6.4. Comparison of BDD with BOSS and MLS for problem with coefficients jumps in a checkerboard pattern formed by 125 subdomains. Coarse spaces of dimensions 2744 and 125 were used for MLS and BOSS. Prolongation smoother of degree 1 was used, and MLS used 2 pre-smoothers, 2 post-smoothers.

Method	BDD		BOSS		BOSS		MLS		MLS	
CS dim.	125		125		2744		125		2744	
α_1, α_2	It.	Cond.	It.	Cond.	It.	Cond.	It.	Cond.	It.	Cond.
Poisson	5	1.18	8	2.35	5	1.10	9	2.93	5	1.21
$10^{-1}, 10^1$	11	3.07	12	3.01	5	1.12	14	4.31	6	1.27
$10^{-2}, 10^2$	10	3.06	12	3.11	5	1.13	15	4.49	6	1.29
$10^{-3}, 10^3$	10	3.05	12	3.11	5	1.13	16	4.50	6	1.29

The results of our numerical experiments confirm the theoretically predicted convergence of the method. We observed that the method using approximate subdomain solvers mimics the desirable properties of BDD. Using approximate local solvers, while keeping the asymptotic cost of each local solve the same as for BDD, provides for considerable savings in the setup phase, as with the iterative subdomain solvers, there is no need for expensive matrix factorizations.

Table 6.5 displays the results of a comparison of BDD with BOSS of Chapter 3 and MLS of Chapter 5. Prolongator smoother of degree 1 was used in both BOSS and MLS. The domain was subdivided into 125 subdomains and the coefficients of the problem were generated randomly, with a uniform distribution. All three methods performed well for all ranges of coefficients. The results suggest BOSS to be least sensitive of the three with respect to the variation of the coefficients.

Table 6.6 displays the results of a comparison of BDD with BOSS and MLS methods. Prolongator smoother of degree 4 was used. The domain was

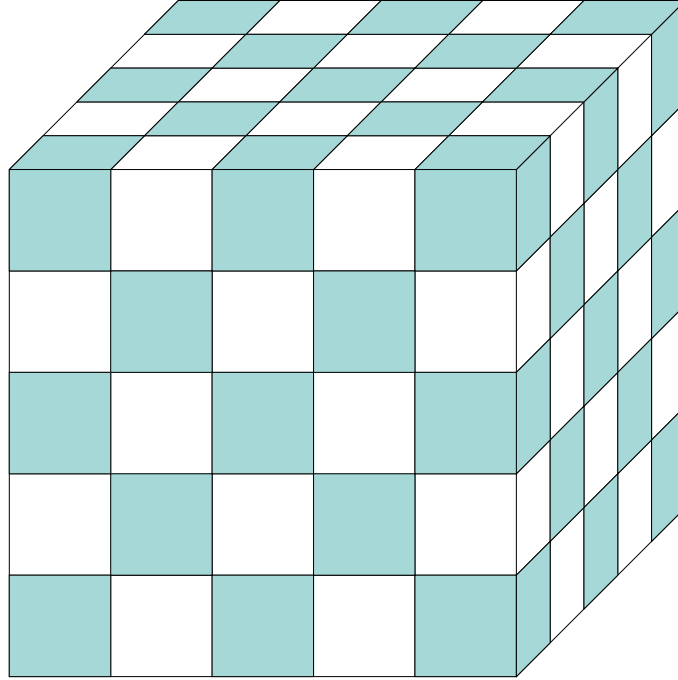


Figure 6.1. The checkerboard coefficient pattern. Dark subdomains correspond to values α_1 , the light ones to α_2 .

Table 6.5. Comparison of BDD with BOSS and MLS for problem with uniformly distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 1 was used, and MLS used 2 pre-smoothers, 2 post-smoothers.

Range of jumps	BDD		BOSS		MLS	
	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
Poisson	5	1.18	8	2.35	9	2.93
$10^{-1} - 10^1$	16	4.38	8	2.50	10	3.14
$10^{-2} - 10^2$	13	3.25	8	2.32	11	3.14
$10^{-3} - 10^3$	11	2.59	8	2.27	11	3.16

Table 6.6. Comparison of BDD with BOSS and MLS for problem with uniformly distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 4, and 4 pre-smoothers and 4 post-smoothers were used in MLS.

Range of jumps	BDD		BOSS		MLS	
	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
Poisson	5	1.18	4	1.11	6	1.66
$10^{-1} - 10^1$	16	4.38	untest	untest	6	1.45
$10^{-2} - 10^2$	13	3.25	untest	untest	6	1.44
$10^{-3} - 10^3$	11	2.59	untest	untest	6	1.45

subdivided into 125 subdomains and the coefficients of the problem were generated randomly, with a uniform distribution. We may observe an improvement in the convergence gained by employing a more powerful prolongation smoother.

Tables 6.7 and 6.8 display the results of a comparison of BDD with BOSS and MLS for problem with exponentially distributed random coefficients and prolongator smoother of degree 1 and 4, respectively. The domain was subdivided into 125 subdomains. Both BOSS and MLS appear to be more stable with respect to the random variation in the coefficients and, as before, using a more powerful prolongator smoother results in more robust iteration.

Tables 6.9 and 6.10 present the comparison of results of application of BDD, BOSS and MLS to a problem subdivided into 2,7444 subdomains, with uniformly and exponentially distributed random coefficients in different ranges. Prolongator smoother of degree 1 was used. We see that for uniformly distributed coefficients, both BOSS and MLS are very robust, resulting in condition number estimates not exceeding 1.20. For the exponentially distributed coefficients, we observe deterioration of performance for MLS, reaching condition number estimate of about 24.

Table 6.7. Comparison of BDD with BOSS and MLS for problem with exponentially distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 1 was used for both BOSS and MLS, and 2 pre-smoothers and 2 post-smoothers in MLS.

Range of jumps	BDD		BOSS		MLS	
	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
Poisson	5	1.18	8	2.35	9	2.93
$10^{-1} - 10^1$	16	4.38	9	2.92	12	4.08
$10^{-2} - 10^2$	13	3.25	11	3.67	20	10.36
$10^{-3} - 10^3$	185	2247.8	12	4.52	39	46.99

Table 6.8. Comparison of BDD with BOSS and MLS for problem with exponentially distributed random coefficients and coarse space of dimension 125. Prolongation smoother of degree 4 was used for BOSS and MLS, and 4 pre- and postsmoothers in MLS.

Range of jumps	BDD		BOSS		MLS	
	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
Poisson	5	1.18	4	1.11	6	1.66
$10^{-1} - 10^1$	16	4.38	untest	untest	6	1.59
$10^{-2} - 10^2$	13	3.25	untest	untest	13	4.87
$10^{-3} - 10^3$	185	2247.8	untest	untest	32	46.33

Table 6.9. Comparison of BDD with BOSS and MLS for problem with uniformly distributed random coefficients and coarse space of dimension 2,744, degree 1 of prolongator smoother, and 2 pre- and postsmoothers in MLS.

Range of jumps	BDD		BOSS		MLS	
	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
Poisson	7	1.52	4	1.15	4	1.18
$10^{-1} - 10^1$	11	2.51	4	1.05	4	1.17
$10^{-2} - 10^2$	11	2.71	4	1.05	5	1.18
$10^{-3} - 10^3$	11	2.71	4	1.15	4	1.18

Table 6.10. Comparison of BDD with BOSS and MLS for problem with exponentially distributed random coefficients and coarse space of dimension 2,744, Prolongation smoother of degree 1 was used, and 2 pre-smoothers and 2 post-smoothers in MLS.

Range of jumps	BDD		BOSS		MLS	
	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
Poisson	7	1.52	4	1.15	4	1.18
$10^{-1} - 10^1$	18	5.24	4	1.11	5	1.40
$10^{-2} - 10^2$	90	124.12	6	1.47	12	5.03
$10^{-3} - 10^3$	540	5,513.4	8	1.93	26	24.08

6.2 Real World Problems

We have also applied the BOSS and MLS method to several real-world problems. The data for these tests was made available by Charbel Farhat of the Aerospace Engineering Department, University of Colorado at Boulder.

Table 6.11 presents results for a shell problem of automobile wheel (Figure 6.2) discretized by 3 node shell elements with 6 degrees of freedom per node. The discrete system obtained by discretization had 59490 degrees of freedom. Prolongator smoother of degree 1 was used in both BOSS and MLS. Two sizes of coarse problem were tested. Good convergence was observed for both BOSS and MLS for the coarse space of 1,260 nodes. For the small coarse space of 158 nodes, the condition number of the problem increases to 57.60 for BOSS and to 77.71 for MLS. This can be explained by the fact that the ratio of fine space coarse space resolutions is large here, so prolongator smoother of degree 1 cannot guarantee convergence independent of this ratio. The degree of the prolongator smoother would have to be increased.

Table 6.12 presents results for the 3D elasticity problem discretized on

Table 6.11. Comparison of BOSS and MLS for solving the shell problem on a mesh discretizing an automobile wheel with 9,915 nodes and 59,490 degrees of freedom. Prolongator smoother of degree 1 was used, and 4 pre-smoothers and 4 post-smoothers in MLS.

Method used	BOSS		MLS	
	Iter.	Cond.	Iter.	Cond.
1260	13	5.42	13	6.03
158	37	57.60	40	77.71

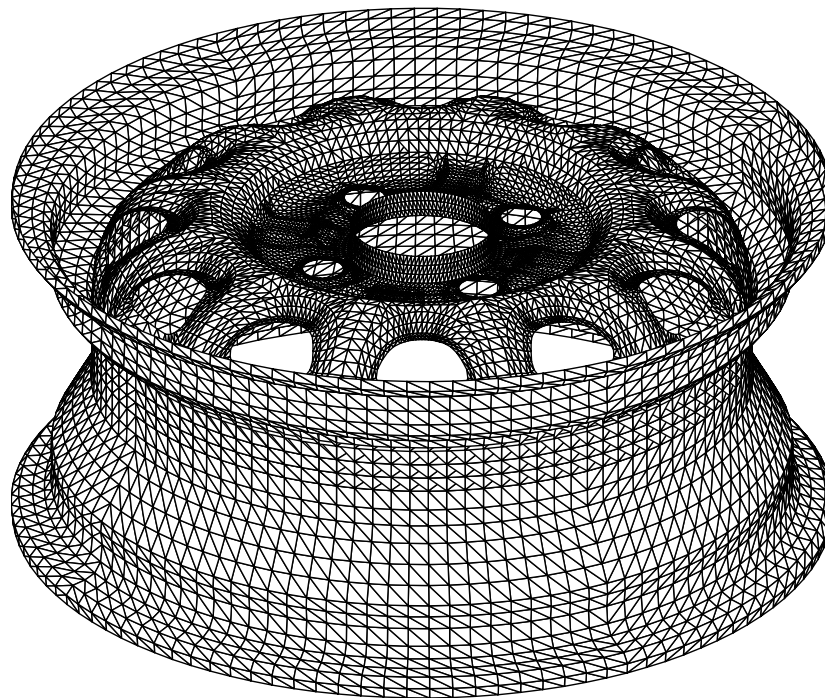


Figure 6.2. The mesh of the automobile wheel (data courtesy of Charbel Farhat, University of Colorado at Boulder).

Table 6.12. Comparison of BOSS and MLS for solving the 3D elasticity problem with 25,058 nodes and 75,174 degrees of freedom, Prolongator smoother of degree 1 was used, and 2 pre-smoothers, 2 post-smoothers in MLS.

Method used	BOSS		MLS	
Nodes in coarse space	Iter.	Cond.	Iter.	Cond.
1438	7	1.55	8	2.05
97	18	8.64	26	19.88

the mesh of Figure 6.2 by tetrahedra elements. The discrete system obtained by discretization had 59,490 degrees of freedom. Prolongator smoother of degree 1 was used in both BOSS and MLS. Two sizes of coarse problem were tested. Good convergence was observed for both BOSS and MLS for the coarse space of 1,434 nodes. For the small coarse space of 97 nodes, the condition number of the problem increases to 8.64 for BOSS and to 19.68 for MLS. Although these are still good results, we can see that application of prolongator smoother of degree greater than 1 may be appropriate for this coarse space. We note that the BOSS method is more robust with respect to the variation of the coarse space size, which can be explained by its link to overlapping Schwarz methods.

Table 6.13 presents results for the shell problem discretized on the mesh of a propeller (Figure 6.2) by 8 node brick elements. The discretized system had 123,120 degrees of freedom. Prolongator smoother of degree 1 was used in both BOSS and MLS.

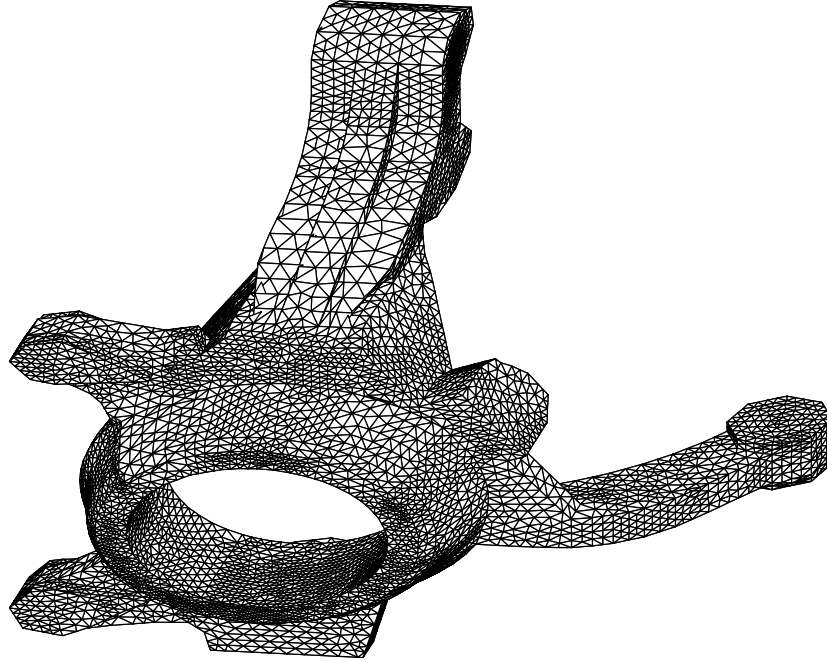


Figure 6.3. The mesh of the solid with tetrahedron elements (data courtesy of Charbel Farhat, University of Colorado at Boulder).

Table 6.13. Comparison of BOSS and MLS for solving a shell problem: a propeller with 41,040 nodes and 123,120 degrees of freedom. Prolongator smoother of degree 1 was used, and 4 pre-smoothers, 4 post-smoothers in MLS.

Method used	BOSS		MLS	
Nodes in coarse space	Iter.	Cond.	Iter.	Cond.
1730	16	20.52	11	14.19

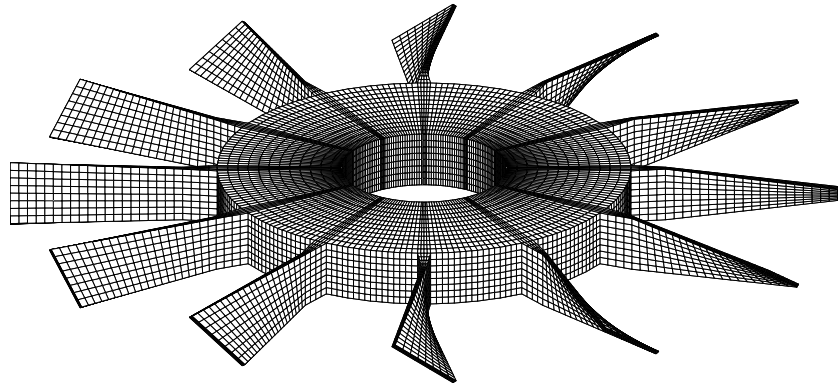


Figure 6.4. The mesh of the turbine with 123,120 elements (data courtesy of John Abel, Cornell University).

7. Conclusions

This study was motivated by the need to solve large algebraic systems arising from discretizations of three-dimensional problems, and by the intolerable asymptotic complexity growth for traditional direct solvers. We have presented three methods for efficient solution of such large sparse systems. Convergence results for these methods have been proved suggesting very good performance. For ACDD, method of Chapter 4 a condition number bound proportional to $(1 + \log(H/h))^2$ has been established. The BOSS method of Chapter 3 and MLS method of Chapter 5 allow certain level of adaptivity, achieved by using intergrid transfer operator smoothed by a polynomial smoother. Increasing the amount of smoothing has a positive effect on the convergence properties, but it increases computational complexity, as with increasing degree of the smoothing polynomial the overlapping of our subdomains increases. In case of the BOSS method, this allows us to move on the scale between domain decomposition without an overlap and domain decomposition with large overlapping, depending on the size of the coarse space and the difficulty of the problem to be solved. The use of smoothed transfer operators also resolves the issue of generating a decomposition into overlapping subdomains. The method starts by generating a nonoverlapping covering of all nodes of the mesh by a simple greedy algorithm and achieves the desired amount of overlapping by applying a smoothing polynomial of appropriate degree.

This feature is also shared by the MLS method, where no subdomain problems are solved, but a combination of coarse grid correction and smoothing similar to a multigrid V -cycle takes place instead. Degree 1 of the prolongator smoother was sufficient in most computational experiments.

All three methods are suitable for unstructured meshes and can be implemented fully black-box. If additional information is available, the methods use it to improve their convergence properties. This additional input usually comes in the form of the basis of the nullspace of the unconstrained matrix of the problem (rigid body modes for elasticity).

The methods have been applied to both artificial and real-world problems to test their robustness with respect to aspects such as the discontinuities of coefficients and unstructured meshes. We have established that these methods, when used as preconditioners, provide a significant improvement in conjugate gradient performance. At the same time, the asymptotic computational complexity estimates obtained suggest that using ACDD, BOSS or MLS to solve 3D problems discretized on a mesh with large fractal dimension requires, even for serial implementations, less work than just the back-substitution stage of a direct solver. Numerical experiments show that the methods are applicable to a broader range of problems than considered in this thesis. This suggests the direction of future research: The application of BDD to solving the plate and shell problems was studied in [56], which also applies to ACDD, but similar analysis remains to be carried out for both BOSS and MLS.

On the practical front, in order to extend applicability of $ACDD(k)$, the implementation of more efficient multigrid local approximate solvers, now

in progress, has to be finished. These solvers are fully algebraic to preserve the black-box philosophy of ACDD.

The original implementation of ACDD was written for solving scalar problems only and although some of the libraries have already been rewritten to support nonscalar problems, the rest of the project will have to conform before computational results measuring the performance for nonscalar problems are available.

A. Appendix Theoretical Results

In this chapter we state and and prove some results used in the text. These results are well-known to readers acquainted with domain decomposition methods.

A.1 Poincaré-Friedrichs Inequality

The following is a version of the well-known Poincaré-Friedrichs inequality. We include it here for the sake of completeness and because it is formulated more generally than usual and the traditional versions (see, e.g., [72, Theorem 7.1],[76]) easily follow from here.

Lemma A.1.1. Let \mathcal{P}^m denote the space of polynomials of degree at most m and let B be a Banach space, $m > 0$, and $K : H^m(\Omega) \rightarrow B$ be a bounded linear operator such that the following implication holds

$$Ku = 0 \text{ and } u \in \mathcal{P}^{m-1} \implies u = 0. \quad (\text{A.1})$$

Further let Ω be a domain such that $\text{diam}(\Omega) = 1$. Then there exists a constant $C(\Omega) > 0$ such that

$$\|u\|_{H^m(\Omega)}^2 \leq C(\Omega)(|Ku|^2 + \|u^{(m)}\|_{L^2(\Omega)}^2) \quad \forall u \in H^m(\Omega). \quad (\text{A.2})$$

Since the converse inequality holds trivially, we obtain the following

Corollary A.1.2. Under the assumptions of Lemma A.1.1 it holds that

$$\|u\|_{H^m(\Omega)}^2 \approx |Ku|^2 + \|u^{(m)}\|_{L^2(\Omega)}^2.$$

Proof. As the statement is obviously valid for $u = 0$, we assume $u \neq 0$ and proceed by contradiction. Assume that (A.2) does not hold. Then there exists a sequence $\{u_i\}$ in $H^m(\Omega)$ such that

$$\|u_i\|_{H^m(\Omega)} = 1 \quad \text{and} \quad Ku_i \rightarrow 0, \quad \|u_i^{(m)}\|_{L^2} \rightarrow 0.$$

The compact imbedding of $H^1(\Omega)$ into $L^2(\Omega)$, known as Rellich theorem [22], implies that also $H^k(\Omega)$ is compactly imbedded in $H^{k-1}(\Omega)$, $k = 1, \dots, m$, so there exists a subsequence $\{u_{i_j}\}$ of $\{u_i\}$ convergent in $H^{k-1}(\Omega)$, and $u_{i_j}^{(m)} \rightarrow 0$ in $L^2(\Omega)$. Therefore (recalling that $\|u_{i_j}^{(m)}\|_{L^2} \rightarrow 0$), $\{u_{i_j}\}$ is a Cauchy sequence in $H^m(\Omega)$, hence it converges to an element $u \in H^m(\Omega)$. Thus $\|u_{i_j}^{(k)}\|_{L^2(\Omega)} \rightarrow \|u^{(k)}\|_{L^2(\Omega)}$, $k = 1, \dots, m$, i.e. all generalized derivatives of u_{i_j} of degree up to m converge in L^2 to the respective generalized derivatives of u , and since $\|u_{i_j}^{(m)}\|_{L^2(\Omega)} \rightarrow 0$, we conclude that $\|u^{(m)}\|_{L^2(\Omega)} = 0$, and $u \in \mathcal{P}^{m-1}$. Due to the H^m -convergence and continuity of K , we have

$$\|u\|_{H^m(\Omega)} = 1 \quad \text{and} \quad Ku_{i_j} \rightarrow 0 = Ku.$$

Asumption (A.1) implies that $u = 0$, which is the sought contradiction. \square

Remark A.1.3. The advantage of proof we present here is its generality. Constructive proofs found in the literature rely on specific geometries. Let us note that the constant $C(\Omega)$ in Lemma A.1.1 depends only on the shape of the domain.

The following theorem is easily obtained from Lemma A.1.1 by scaling the L^2 and H^1 norms from a domain of diameter 1 to one of diameter H .

Theorem A.1.4. Let \mathcal{P}^m denote the space of polynomials of degree at most m and let B be a Banach space, $m > 0$, and $K : H^m(\Omega) \rightarrow B$ be a bounded

linear operator such that the following implication holds

$$Ku = 0 \text{ and } u \in \mathcal{P}^{m-1} \implies u = 0. \quad (\text{A.3})$$

Further let Ω be a domain such that $\text{diam}(\Omega) = H$. Then there exists a constant $C(\Omega) > 0$ such that

$$\|u\|_{H^m(\Omega)}^2 \leq C(\Omega)H^2(|Ku|^2 + \|u^{(m)}\|_{L^2(\Omega)}^2) \quad \forall u \in H^m(\Omega). \quad (\text{A.4})$$

Following lemma is a variation of standard Poincaré inequality on the space of traces.

Lemma A.1.5. Let Ω be a domain such that $\text{diam}(\Omega) = 1$. Let B be the linear operator on $H^{1/2}(\partial\Omega_i)$ defined by

$$B(v) = v|_{\partial\Omega_i \cap \Gamma_D} \quad (\text{A.5})$$

if $\text{meas}(\partial\Omega_i \cap \Gamma_D) > 0$, i.e., if there are some Dirichlet boundary conditions imposed on $\partial\Omega_i$, and

$$B(v) = \int_{\partial\Omega_i} v ds \quad (\text{A.6})$$

otherwise. Then there exists a constant $C(\Omega)$ such that for all $u \in \text{Ker } B$,

$$|u|_{1/2, \partial\Omega_i} \leq \|u\|_{1/2, \partial\Omega_i} \leq C(\Omega) |u|_{1/2, \partial\Omega_i}. \quad (\text{A.7})$$

Proof. It is sufficient to prove the proposition for the case of an arbitrary operator B such that for all constant functions, $B(u) = 0$ implies $u = 0$. The left-hand inequality in (A.7) is immediate. Suppose the right-hand inequality in (A.7) is false. Then there exists a sequence $\{u_n\}$ in $\text{Ker } B$ such that

$$\|u_n\|_{1/2, \partial\hat{\Omega}} = 1 \text{ and } |u_n|_{1/2, \partial\hat{\Omega}} \rightarrow 0, \quad (\text{A.8})$$

which implies that

$$|u_n|_{0,\partial\hat{\Omega}} \rightarrow 1. \quad (\text{A.9})$$

Because of the compact imbedding $H^{1/2}(\partial\hat{\Omega}) \hookrightarrow L^2(\partial\hat{\Omega})$, there exists a subsequence $\{u_{n_k}\}$ of $\{u_n\}$ such that

$$u_{n_k} \rightarrow u \text{ in } L^2(\partial\hat{\Omega}). \quad (\text{A.10})$$

Since u_{n_k} is bounded in $H^{1/2}(\partial\hat{\Omega})$, there exists (see e.g. [95]) a subsequence, for notational convenience also denoted $\{u_{n_k}\}$, such that

$$u_{n_k} \rightharpoonup u \text{ weakly in } H^{1/2}(\partial\hat{\Omega}).$$

Since B is a bounded operator and $u_{n_k} \in \text{Ker } B$,

$$B(u_{n_k}) \rightarrow B(u) = 0. \quad (\text{A.11})$$

From (A.8),

$$\|u_{n_k}\|_{1/2,\partial\hat{\Omega}} \rightarrow \|u\|_{1/2,\partial\hat{\Omega}} = 1, \quad (\text{A.12})$$

$$|u_{n_k}|_{1/2,\partial\hat{\Omega}} \rightarrow |u|_{1/2,\partial\hat{\Omega}} = 0. \quad (\text{A.13})$$

It follows from (A.13) and the definition of the $H^{1/2}$ seminorm that there exists a constant L such that $u = L$ almost everywhere. Since $B(u) = 0$, the assumption on B yields $u = 0$, which contradicts to (A.12). \square

Remark A.1.6. As in the case of Theorem A.1.4, for general domains the constant $C(\Omega)$ in (A.7) depends on the properties of the domain and the scaling of appropriate norms reveals that for a domain “blown up” to diameter H , equation (A.7) changes to

$$|u|_{1/2,\partial\Omega_i} \leq \|u\|_{1/2,\partial\Omega_i} \leq C(\Omega)H^{1/2} |u|_{1/2,\partial\Omega_i}. \quad (\text{A.14})$$

As an application of Theorem A.1.4 and Lemma A.1.5, we have the following lemma.

Lemma A.1.7. Let A denote the $n \times n$ stiffness matrix resulting from finite element discretization of an elliptic second order differential equation on a domain of diameter 1 with a mesh of typical meshsize h , and S be the $m \times m$ Schur complement matrix obtained from A by eliminating the interior degrees of freedom. Then

$$\text{cond}(A) = O\left(\frac{1}{h^2}\right), \quad \text{cond}(S) = O\left(\frac{1}{h}\right). \quad (\text{A.15})$$

Proof. Let $a(u, v) = f(v)$ be the variational formulation from discretization of which A was obtained. Using the equivalence of Euclidean and continuous L^2 norms and the Poincaré inequality (A.4), we can estimate the Rayleigh quotient of A as

$$RQ(A) = \frac{\langle Ax, x \rangle}{\langle x, x \rangle} \approx \frac{a(\Pi x, \Pi x)}{h^d \|\Pi x\|_{L^2(\Omega)}^2} \geq \frac{C_P^2 \|\Pi x\|_{L^2(\Omega)}^2}{h^d \|\Pi x\|_{L^2(\Omega)}^2} = \frac{C_P^2}{h^d} \quad \forall x \in \mathbb{R}^n. \quad (\text{A.16})$$

As the finite element matrix A is naturally scaled so that its diagonal elements are of order h^{d-2} , the upper bound of the spectrum of A may be found from Geršgorin theorem [47] to be

$$RQ(A) \leq \varrho(A) \leq C \frac{1}{h^{d-2}}.$$

From the last estimate and (A.16), we obtain

$$\text{cond}(A) \leq C \frac{1}{h^2}.$$

In order to investigate the conditioning of S , let us first write its definition $S = A_{11} - A_{12}A_{22}^{-1}A_{21}$, where A_{11}, A_{12}, A_{21} and A_{22} are the blocks of A

corresponding to its decomposition into interior and interface degrees of freedom.

As the matrix $A_{12}A_{22}^{-1}A_{21}$ is positive definite, we obtain

$$\varrho(S) \leq \varrho(A_{11}) \leq \varrho(A) \leq Ch^{d-2}. \quad (\text{A.17})$$

For the lower estimate of the spectrum of S we use the equivalence of Euclidean and continuous L^2 norms, the trace version of Poincaré inequality (A.7) and the well-known equivalence $\|\bar{x}\|_S \approx \|\Pi\bar{x}\|_{H^{1/2}(\partial\Omega)}$. We obtain

$$RQ(S) = \frac{\langle S\bar{x}, \bar{x} \rangle}{\langle \bar{x}, \bar{x} \rangle} \approx \frac{s(\Pi\bar{x}, \Pi\bar{x})}{h^{d-1}\|\Pi\bar{x}\|_{L^2(\partial\Omega)}^2} \quad (\text{A.18})$$

$$\geq \frac{C_P^2\|\Pi\bar{x}\|_{L^2(\partial\Omega)}^2}{h^{d-1}\|\Pi\bar{x}\|_{L^2(\partial\Omega)}^2} \geq \frac{C_P^2}{h^{d-1}} \quad \forall \bar{x} \in \mathbb{R}^m. \quad (\text{A.19})$$

From the last inequality and equation (A.17) it follows that $\text{cond}(S) \leq C\frac{1}{h}$, concluding the proof. \square

B. Appendix Results Used In Computational Experiments

B.1 The Stopping Criterion for PCG

Since the conjugate gradient method is an iterative method, a practical criterion is needed to determine when to stop the iteration. We need to be certain that our approximation is close to the true solution x^* of the system $Ax = b$ in the sense that $\frac{\|e_i\|_B}{\|x^*\|_B} \leq \epsilon$ for some symmetric positive definite matrix B and $\epsilon > 0$. Perhaps the most commonly used choice is $B = A^T A$, leading to the bound on the residual norm $\frac{\|r_i\|}{\|b\|} \leq \epsilon$. Thus we can check whether the relative residual, measured in Euclidean norm, is smaller than the required precision ϵ and if so, the iteration is terminated. The weaknesses of this measure of convergence are notoriously well known. A criterion with a better vision of the true distance of the current approximation from the exact solution is needed. Since the measure of a distance of the last iteration to the true solution is unavailable, we use a test based on the residual and the estimate of the condition number. The stopping criterion used in our numerical experiments was based on the estimate of the relative error measured in the A -energetic norm (cf. Ashby, Manteuffel and Saylor [3])

$$\frac{\|e_i\|_A}{\|x^*\|_A} \leq \left(\text{cond}(\mathcal{M}^{-1}A) \frac{\langle \mathcal{M}^{-1}r_i, r_i \rangle}{\langle \mathcal{M}^{-1}b, b \rangle} \right)^{1/2} \leq \epsilon, \quad (\text{B.1})$$

$\|\cdot\|, \langle \cdot, \cdot \rangle$ denoting the Euclidean norm and inner product, respectively. The iteration will be terminated when the second inequality in (B.1) is satisfied.

We used an approximation of $\text{cond}(\mathcal{M}^{-1}A)$ obtained from the extreme eigenvalues of a tridiagonal Lanczos matrix. This matrix can be constructed from the data obtained during the conjugate gradient iteration. In addition, the two eigenvalues of the Lanczos matrix do not have to be computed in each iteration. We start by taking 1 to be the first approximation $\text{cond}(\mathcal{M}^{-1}A)$ and compute a new approximation of the extreme eigenvalues of the Lanczos matrix only when (B.1) is satisfied with the current estimate of $\text{cond}(\mathcal{M}^{-1}A)$. If (B.1) holds also for the new estimate, we terminate the iteration process.

The approximation of $\text{cond}(\mathcal{M}^{-1}A)$ was obtained as follows. We are interested in the extreme eigenvalues λ corresponding to the generalized eigenvalue problem $Ax = \lambda \mathcal{M}x$. Alternatively, we are searching for the extreme values of the Rayleigh quotient

$$RQ(u) = \frac{\langle Au, u \rangle}{\langle \mathcal{M}u, u \rangle}.$$

on the Krylov subspace $K = \{r, \mathcal{M}^{-1/2}A\mathcal{M}^{-1/2}r, \dots\}$.

In each step of the preconditioned conjugate algorithm 2 we solve the problem

$$\mathcal{M}z_k = r_k$$

and compute

$$p_k = z_{k-1} + \beta_k p_{k-1}. \tag{B.2}$$

Recall the orthogonal properties

$$r_i^T \mathcal{M}^{-1} r_j = z_j^T \mathcal{M} z_j = p_i^T A p_j = 0 \quad i \neq j.$$

Let us create a bidiagonal matrix

$$B_k = \begin{bmatrix} 1, & -\beta_2, & 0, & 0, & \dots, & 0 \\ 0, & 1, & -\beta_3, & 0, & \dots, & 0 \\ & & \vdots & & & \\ 0, & 0, & \dots, & 0, & 1 & -\beta_k \\ 0, & 0, & \dots, & 0, & 0 & 1 \end{bmatrix}.$$

It is easy to see that (B.2) is equivalent to $z_{k-1} = p_p - \beta_k p_{k-1}$. Denoting $Z_k = [z_0, \dots, z_{k-1}]$, $P_k = [p_1, \dots, p_k]$, this fact can be written as

$$Z_k = P_k B_k.$$

If we consider $z_k \in K$ normalized so that $\|z_k\|_{\mathcal{M}} = 1$, (this amounts to using a scaling matrix $S_k = \text{diag}((z_0 \mathcal{M} z_0)^{-1/2}, \dots, (z_{k-1} \mathcal{M} z_{k-1})^{-1/2})$). We have

$$Z_k S_k = [z_0 (z_0^T \mathcal{M} z_0)^{-1/2}, \dots, z_{k-1} (z_{k-1} \mathcal{M} z_{k-1})^{-1/2}].$$

Let us now consider the Rayleigh quotient for any unit vector (in \mathcal{M} -norm) $u = Z_k S_k \delta$

$$RQ(u) = \frac{\delta^T S_k B_k^T P_k^T A P_k B_k S_k \delta}{\delta^T S_k Z_k^T \mathcal{M} S_k Z_k \delta}$$

We observe that in the numerator, $P_k^T A P_k = D_k$ is the diagonal matrix with entries $p_i^T A p_j$, and in the denominator $S_k Z_k^T \mathcal{M} S_k Z_k = I$. Thus

$$RQ(u) = \frac{\delta^T S_k B_k^T D_k B_k S_k \delta}{\delta^T \delta}.$$

Thus the values of the Rayleigh quotient will be determined by the spectrum of the three-diagonal matrix $S_k B_k^T D_k B_k S_k$, a task easily accomplished by the standard methods (cf. [51]).

References

- [1] R. ADAMS, **Sobolev Spaces**, Academic Press, New York, 1975.
- [2] G. ANAGNOSTOU, Y. MADAY, C. MAVRIPLIS, AND A. T. PATERA, **On the mortar element method: generalizations and implementation**, in Third International Symposium of Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., Philadelphia, 1990, SIAM, pp. 157–173.
- [3] S. F. ASHBY, T. A. MANTEUFFEL, AND P. E. SAYLOR, **A taxonomy for conjugate gradient methods**, SIAM J. Numer. Anal., 27 (1990), pp. 1542–1568.
- [4] C. BERNARDI, Y. MADAY, AND A. PATERA, **A new non conforming approach domain decomposition: The mortar element method**, in Collège de France Seminar, H. Brezis and J. Lions, eds., Pitman, Providence, RI, 1994.
- [5] P. E. BJØRSTAD AND J. MANDEL, **Spectra of sums of orthogonal projections and applications to parallel computing**, BIT, 31 (1991), pp. 76–88.
- [6] R. BLAHETA, **A multilevel method with overcorrection by aggregation for solving discrete elliptic problems**, J. Comput. Appl. Math., 24 (1988), pp. 227–239.
- [7] ———, **Iterative Methods for Numerical Solving of the Boundary Value Problems of Elasticity**, PhD thesis, Hornický ústav, Ostrava, 1989. in Czech.
- [8] C. BÖRGERS, **The Neumann–Dirichlet domain decomposition method with inexact solvers on the subdomains**, Numer. Math., 55 (1989), pp. 123–136.

- [9] J. H. BRAMBLE, **A second order finite difference analogue of the first biharmonic boundary value**, Numer. Math., 9 (1966), pp. 236–249.
- [10] J. H. BRAMBLE, J. E. PASCIAK, AND A. H. SCHATZ, **The construction of preconditioners for elliptic problems by substructuring, I**, Math. Comp., 47 (1986), pp. 103–134.
- [11] —, **An iterative method for elliptic problems on regions partitioned into substructures**, Math. Comp., 46 (1986), pp. 361–369.
- [12] —, **The construction of preconditioners for elliptic problems by substructuring, IV**, Math. Comp., 53 (1989), pp. 1–24.
- [13] J. H. BRAMBLE, J. E. PASCIAK, AND A. T. VASILLEV, **Analysis of non-overlapping domain decomposition algorithms with inexact solvers**. To appear, 1996.
- [14] J. H. BRAMBLE, J. E. PASCIAK, J. WANG, AND J. XU, **Convergence estimates for multigrid algorithms without regularity assumptions**, Math. Comp., 57 (1991), pp. 23–45.
- [15] A. BRANDT, **Algebraic multigrid theory: The symmetric case**, Appl. Math. Comput., 19 (1986), pp. 23–56.
- [16] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, **Algebraic multigrid (AMG) for sparse matrix equations**, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge Univ. Press, Cambridge, 1984.
- [17] M. BREZINA AND P. VANĚK, **One black-box iterative solver**. In preparation, 1997.
- [18] X.-C. CAI, **An optimal two-level overlapping domain decomposition method for elliptic problems in two and three dimensions**, SIAM J. Sci. Comp., 14 (1993), pp. 239–247.
- [19] M. A. CASARIN AND O. B. WIDLUND, **A hierarchical preconditioner for the mortar finite element method**, Tech. Report 712, Courant Institute of Mathematical Sciences, 1996. Tech. Report.

- [20] T. F. CHAN AND B. SMITH, **Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes**, Tech. Report 93-42, Department of Math., UCLA, 1993. CAM Report.
- [21] T. F. CHAN AND B. SMITH, **Domain decomposition for unstructured mesh problems**, in Proceedings of the seventh international symposium on domain decomposition methods for partial differential equations, D. E. Keys and J. Xu, eds., 1993, pp. 175–189.
- [22] P. CIARLET, **The Finite-Element Method for Elliptic Problems**, North-Holland, Amsterdam, 1978.
- [23] P. G. CIARLET, **The Finite Element Method for Elliptic Problems**, North-Holland, Amsterdam, New York, 1978.
- [24] P. CONCUS, G. H. GOLUB, AND D. P. O’LEARY, **A generalized conjugate gradient method for the numerical solution of elliptic PDE**, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 309–332.
- [25] L. COWSAR, J. MANDEL, AND M. F. WHEELER, **Balancing domain decomposition for mixed finite elements**. Math. Comp., To appear.
- [26] Y.-H. DE ROECK AND P. LE TALLEC, **Analysis and test of a local domain decomposition preconditioner**, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, PA, 1991.
- [27] M. DRYJA, **A finite element – capacitance matrix method for the elliptic problem**, SIAM J. Numer. Anal., 20 (1983), pp. 671–680.
- [28] M. DRYJA, **A method of domain decomposition for 3-D finite element problems**, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM.
- [29] M. DRYJA, W. PROSKUROWSKI, AND O. B. WIDLUND, **A method of domain decomposition with cross points for elliptic finite element problems**, in Optimal Algorithm, B. Sendov, ed., Publishing House of the Bulgarian Academy of Sciences, Sofia, 1986, pp. 97–111.

- [30] M. DRYJA, B. F. SMITH, AND O. B. WIDLUND, **Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions**, SIAM J. Numer. Anal., 31 (1994), pp. 1662–1694.
- [31] M. DRYJA AND O. B. WIDLUND, **Towards a unified theory of domain decomposition algorithms for elliptic problems**, in Third International Symposium of Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., Philadelphia, 1990, SIAM, pp. 3–21.
- [32] M. DRYJA AND O. B. WIDLUND, **Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems**, Tech. Report 626, Department of Computer Science, Courant Institute, March 1993. To appear in Comm. Pure Appl. Math.
- [33] ———, **Domain decomposition algorithms with small overlap**, SIAM J. Sci.Comput., 15 (1994), pp. 604–620.
- [34] R. E. EWING, R. D. LAZAROV, T. F. RUSSELL, AND P. S. VASSILEVSKI, **Local refinement via domain decomposition techniques for mixed finite element methods with rectangular Raviar–Thomas elements**, in Doamin Decomposition Methods for PDE’s, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia, 1990, pp. 98–114.
- [35] V. FABER AND T. A. MANTEUFFEL, **Necessary and sufficient conditions for the existence of a conjugate gradient method**, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
- [36] C. FARHAT, P. S. CHEN, AND J. MANDEL, **Scalable Lagrange multiplier based domain decomposition method for time-dependent problems**, Int. J. Numer. Meth. Engrg., (1995). To appear.
- [37] C. FARHAT AND M. LESOINNE, **Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics**, J. Numer. Meth. Engrg., 36 (1993), pp. 745–764.
- [38] ———, **Mesh partitioning algorithms for the parallel solution of partial differential equations**, Appl. Numer. Math., 12 (1993), pp. 443–457.
- [39] C. FARHAT AND F. X. ROUX, **A method of finite element tearing and interconnecting and its parallel solution algorithm**, Int. J. Numer. Meth. Engng., 32 (1991).

- [40] G. E. FORSYTHE AND E. G. STRAUSS, **On the best conditioned matrices**, Proc. Amer. Math. Soc., 6 (1955), pp. 340–345.
- [41] A. GEORGE, **Nested dissection of regular finite element mesh**, Siam. J. Numer. Anal., 11 (1973), pp. 345–363.
- [42] A. GEORGE AND J. LIU, **Computer Solution of Large Sparse Positive Definite Systems**, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [43] R. GLOWINSKI AND M. F. WHEELER, **Domain decomposition and mixed finite element methods for elliptic problems**, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, 1988, SIAM, pp. 144–172.
- [44] G. H. GOLUB AND C. F. VAN LOAN, **Matrix Computations**, Johns Hopkins Univ. Press, Baltimore, MD, 1989. Second edition.
- [45] W. HACKBUSCH, **Multigrid Methods and Applications**, vol. 4 of Computational Mathematics, Springer-Verlag, Berlin, 1985.
- [46] M. R. HESTENES AND E. STIEFEL, **Methods of conjugate gradients for solving linear systems**, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–436.
- [47] R. A. HORN AND C. R. JOHNSON, **Matrix Analysis**, Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sydney, first ed., 1985.
- [48] T. J. R. HUGHES AND R. M. FERENCZ, **Fully vectorized EBE preconditioners for nonlinear solid mechanics: applications to large-scale three-dimensional continuum, shell and contact/impact problems**, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, 1988, SIAM, pp. 261–280.
- [49] T. J. R. HUGHES, R. M. FERENCZ, AND J. O. HALLQUIST, **Large-scale vectorized implicit calculations in solid mechanics on a Cray X-MP/48 utilizing EBE preconditioned conjugate gradients**, Comp. Meth. Appl. Mech. Engng., 61 (1987), pp. 215–248.

- [50] T. J. R. HUGHES, I. LEVIT, AND J. WINGET, **An element-by-element solution algorithm for problems of structural and solid mechanics**, *Comp. Meth. Appl. Mech. Engng.*, 36 (1983), pp. 241–254.
- [51] E. ISAACSON AND H. B. KELLER, **Analysis of Numerical Methods**, John Wiley & Sons, New York, 1966.
- [52] J. KRÍŽKOVÁ AND P. VANĚK, **Two-level preconditioner with small coarse grid appropriate for unstructured meshes**, *Numerical Linear Algebra with Applications*, 3 (1996), pp. 255–274.
- [53] P. LE TALLEC, **Neumann-Neumann domain decomposition algorithms for solving 2D elliptic problems with nonmatching grids**, *East-West J. Numer. Math.*, 1 (1993), pp. 129–146.
- [54] ———. Personal communication, 1994.
- [55] ———, **Domain decomposition methods in computational mechanics**, *Computational Mechanics Advances*, 1 (1994), pp. 121–220.
- [56] P. LE TALLEC, J. MANDEL, AND M. VIDRASCU, **Parallel domain decomposition algorithms for solving plate and shell problems**, in *Advances in Parallel and Vector Processing for Structural Mechanics*, Edinburgh, 1994, CIVIL-COMP Ltd. Proceedings, Athens, 1994.
- [57] P. LE TALLEC AND J. RODRIGUES, **Domain decomposition methods with nonmatching grids applied to fluid dynamics**, in *Proceedings of the VIII International Conference on Fluid Dynamics*, K. Morgan, E. Onate, J. Periaux, J. Peraire, and O. Zinkiewicz, eds., Pineridge Press, Barcelona, 1993, pp. 405–418.
- [58] W. LEONTIEF, **The Structure of the American Economy 1919-1939**, Oxford University Press, New York, 1951.
- [59] P. LIONS, **On the Schwarz alternating method I**, in *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, 1988, SIAM, pp. 1–42.
- [60] D. LUENBERGER, **Linear and Nonlinear Programming**, Addison Wesley, Menlo Park, London, Amsterdam, Don Mills, Sydney, second ed., 1984.

- [61] J. MANDEL, **On block diagonal and Schur complement preconditioning**, Numer. Math., 58 (1990), pp. 79–93.
- [62] —, **Balancing domain decomposition**, tech. report, Computational Mathematics Group, University of Colorado at Denver, 1992. Communications in Applied Numerical Methods, to appear.
- [63] —, **Balancing domain decomposition**, Comm. in Numerical Methods in Engrg., 9 (1993), pp. 233–241.
- [64] —, **Hybrid domain decomposition with unstructured subdomains**, in Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition, Y. A. Kuznetsov, J. Périaux, A. Quarteroni, and O. B. Widlund, eds., vol. 157, AMS, 1994. Held in Como, Italy, June 15–19,1992.
- [65] J. MANDEL AND M. BREZINA, **Balancing domain decomposition: Theory and computations in two and three dimensions**, UCD/CCM Report 2, Center for Computational Mathematics, University of Colorado at Denver, 1993.
- [66] —, **Balancing domain decomposition for problems with large jumps in coefficients**, Math. Comp., 65 (1996), pp. 1387–1401.
- [67] J. MANDEL, S. F. MCCORMICK, AND J. RUGE, **An algebraic theory for multigrid methods for variational problems**, SIAM J. Numer. Anal., 25 (1988), pp. 91–110.
- [68] J. MANDEL AND B. SEKERKA, **A local convergence proof for the iterative aggregation method**, J. Lin. Alg. Applic., 51 (1983), pp. 163–172.
- [69] J. MANDEL AND R. TEZAU, **On the convergence of a substructuring method with Lagrange multipliers**, UCD/CCM Report 33, Center for Computational Mathematics, University of Colorado at Denver, December 1994. Submitted to Numerische Mathematik.
- [70] G. MARCHUK, **Methods for Computing Nuclear Reactors**, GOSATOMIZDAT, Moscow, 1961. In Russian.

- [71] A. M. MATSOKIN AND S. V. NEPOMNYASCHIKH, **A Schwarz alternating method in subspaces**, In. Vuzov, 10 (1985), pp. 61–66. Also in Soviet Mathematics, 10 (1985), pp. 78–84.
- [72] J. NEČAS, **Les Méthodes Dirictes en Théorie des Équations Élliptiques**, Academia, Prague, 1967.
- [73] S. V. NEPOMNYASCHIKH, **Decomposition and fictitious domains methods for elliptic boundary value problems**, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, G. Meurant, J. S. Scroggs, and R. G. Voigt, eds., Philadelphia, 1992, SIAM, pp. 62–72.
- [74] J. S. PRZEMIENIECKI, **Matrix structural analysis of substructures**, Am. Inst. Aero. Astro. J., 1 (1963), pp. 138–147.
- [75] J. K. REID, **On the method of conjugate gradients for the solution of large sparse systems of linear equations**, in Large Sparse Sets of Linear Equations, J. K. Reid, ed., Academic Press, New York, 1972, pp. 231–254.
- [76] K. REKTORYS, **Variational methods in mathematics, science, and engineering**, D. Reidel Pub. Co., Dordrecht ; Boston, 1977. translated from the Czech by Michael Basch.
- [77] J. W. RUGE, **Algebraic multigrid (AMG) for geodetic survey problems**, in Preliminary Proc. Internat. Multigrid Conference, Fort Collins, CO, 1983, Institute for Computational Studies at Colorado State University.
- [78] J. W. RUGE AND K. STÜBEN, **Algebraic multigrid (AMG)**, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.
- [79] M. SARKIS, **Two-level Schwarz methods for nonconforming finite elements and discontinuous coefficients**, in Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods, Volume 2, N. D. Melson, T. A. Manteuffel, and S. F. McCormick, eds., no. 3224, Hampton VA, 1993, NASA, pp. 543–566.
- [80] H. A. SCHWARZ, **Gesammelte Mathematische Abhandlungen**, vol. 2, Springer, Berlin, 1890, pp. 133–143. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, volume 15, 1870, pp. 272–286.

- [81] B. F. SMITH, **A domain decomposition algorithm for elliptic problems in three dimensions**, Numer. Math., 60 (1991), pp. 219–234.
- [82] K. STÜBEN, **Algebraic multigrid (AMG): experiences and comparisons**, Appl. Math. Comput., 13 (1983), pp. 419–452.
- [83] R. TEZAUER, P. VANĚK, AND M. BREZINA, **Two-level method for solids on unstructured meshes**. submitted to SIAM J. Sci. Comp.
- [84] T. E. TEZDUYAR AND J. LIOU, **Element-by-element and implicit-explicit finite element formulations for computational fluid dynamics**, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, 1988, SIAM, pp. 281–300.
- [85] T. E. TEZDUYAR, J. LIOU, T. NGUYEN, AND S. POOLE, **Adaptive implicit-explicit and parallel element-by-element iterative schemes**, in Proceedings of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., Philadelphia, 1989, SIAM, pp. 443–463.
- [86] P. VANĚK, J. MANDEL, AND M. BREZINA, **Algebraic multigrid on unstructured meshes**, UCD/CCM Report 34, Center for Computational Mathematics, University of Colorado at Denver, December 1994. Submitted to SISC.
- [87] P. VANĚK, J. MANDEL, AND M. BREZINA, **Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems**, Computing, 56 (1996), pp. 179–196.
- [88] P. VANĚK, R. TEZAUER, M. BREZINA, AND J. KRÍŽKOVÁ, **Two-level method with coarse space size independent convergence**, in Domain Decomposition Methods in Sciences and Engineering, R. Glowinski, J. Périaux, Z. Shi, and O. Widlund, eds., John Wiley & Sons Ltd., New York, N.Y., 1997.
- [89] J. WANG, **New convergence estimates for multilevel algorithms for finite element equations**. Submitted.

- [90] J. WANG, **Convergence analysis without regularity assumptions for multigrid algorithms based on SOR smoothing**, SIAM J. Numer. Anal., 29 (1992), pp. 987–1001.
- [91] A. WATHEN. Personal communication, 1992.
- [92] O. B. WIDLUND, **An extension theorem for finite element spaces with three applications**, in Numerical Techniques in Continuum Mechanics, W. Hackbusch and K. Witsch, eds., Braunschweig/Wiesbaden, 1987, Notes on Numerical Fluid Mechanics, v. 16, Friedr. Vieweg und Sohn, pp. 110–122. Proceedings of the Second GAMM-Seminar, Kiel, January, 1986.
- [93] O. B. WIDLUND, **Iterative substructuring methods: algorithms and theory for elliptic problems in the plane**, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, 1988, SIAM, pp. 113–128.
- [94] J. XU, **Iterative methods by subspace decomposition and subspace correction**, SIAM Review, 34 (1992), pp. 581–613.
- [95] K. YOSIDA, **Functional Analysis**, Springer-Verlag, Berlin, Heidelberg, sixth ed., 1980.