Preface

The purpose of this text is to offer an overview of the most popular domain decomposition methods for partial differential equations (PDEs). The presentation is kept as much as possible at an elementary level with a special focus on the definitions of these methods both in terms of PDEs and of the sparse matrices arising from their discretizations. We also provide implementations available at `www.siam.org/books/ot144` written with open-source finite element software. In addition, we consider a number of methods that have not been presented in other books. We think that this book will give a new perspective and that it will complement those of Smith, Bjørstad, and Gropp [175], Quarteroni and Valli [165], Mathew [138], and Toselli and Widlund [185], as well as the review article [23].

The book is addressed to computational scientists, mathematicians, physicists, and, in general, to people involved in numerical simulation of PDEs. It can also be used as a textbook for advanced undergraduate/first-year graduate students. The mathematical tools needed are basic linear algebra, notions of programming, variational formulation of PDEs, and basic knowledge of finite element discretization.

The value of domain decomposition methods is part of a general need for parallel algorithms for professional and consumer use. We will focus on scientific computing and more specifically on the solution of the algebraic systems arising from the approximation of a PDE.

Domain decomposition methods are a family of methods used to solve problems of linear algebra on parallel machines in the context of simulation. In scientific computing, the first step is to model mathematically a physical phenomenon. This often leads to systems of PDEs such as the Navier–Stokes equations in fluid mechanics, elasticity systems in solid mechanics, Schrödinger equations in quantum mechanics, the Black–Scholes equation in finance, Lighthill–Whitham equations for traffic, etc. Functional analysis is used to study the well-posedness of the PDEs, which is a necessary condition for their possible numerical approximation. Numerical analysis enables one to design stable and consistent discretization schemes. This leads to the discrete equations $F(u) = b \in \mathbb{R}^n$, where n is the number of degrees of freedom of the discretization. If F is linear, calculating u is a problem of linear algebra. If F is nonlinear, a method for solving it is the classical Newton method, which also leads to solving a series of linear systems.

In the past, improving performance of a program, either in speed or in the amount of data processed, was only a matter of waiting for the next generation of processors. Every 18 months, computer performance doubled. As a consequence, linear solver research would take second place to the search for new discretization schemes. But since around 2005 the clock speed has stagnated at 2–3 GHz. Increase in performance is almost entirely due to the increase in the number of cores per processor. All major processor vendors are producing multicore chips, and now every machine is a parallel machine. Waiting for the next-generation machine no longer guarantees better performance of software. To keep doubling, performance parallelism must double, which implies a huge effort in algorithmic development. Scientific computing is only one illustration of this general need in computer science. Visualization, data storage, mesh generation, operating systems, etc., must all be designed with parallelism in mind.

We focus here on parallel linear iterative solvers. Contrary to direct methods, the appealing feature of domain decomposition methods is that they are naturally parallel. We introduce the reader to the main classes of domain decomposition algorithms: Schwarz, Neumann–Neumann/FETI, and optimized Schwarz. For each method we start with the continuous formulation in terms of PDEs for two subdomains. We then give the definition in terms of stiffness matrices and their implementation in a free finite element package in the many-subdomain case. This presentation reflects the dual nature of domain decomposition methods. They are solvers of linear systems, keeping in mind that the matrices arise from the discretization of partial differential operators. As for domain decomposition methods that directly address nonlinearities, we refer the reader to, e.g., [63, 122], [17] or [18] and references therein. As for iterative solvers non related to domain decomposition we refer the reader to [12] or [140] e.g.

In Chapter 1 we start by introducing different versions of Schwarz algorithms at the continuous level, taking as our starting point the methods of H. Schwarz (see [174]): the Jacobi–Schwarz method (JSM), the additive Schwarz method (ASM) and the restricted additive Schwarz (RAS) method. The first natural feature of these algorithms is that they are equivalent to a block Jacobi method (default solver in PETSc [8, 7]) when the overlap is minimal. We move on to the algebraic versions of the Schwarz methods. In order to do this, several concepts are necessary: restriction and prolongation operators, as well as partitions of unity which make the global definition possible. These concepts are explained in detail for different types of discretizations (finite difference or finite element) and spatial dimensions. The convergence of the Schwarz method in the two-subdomain case is illustrated for one-dimensional problems and then for two-dimensional problems by using Fourier analysis. A short paragraph introduces P. L. Lions' algorithm that will be considered in detail in Chapter 2. The last part of the chapter is dedicated to numerical implementation using FreeFem++ [108] for general decompositions into subdomains.

In Chapter 2 we present the optimized Schwarz methods applied to the Helmholtz equation, which models acoustic wave propagation in the frequency domain. We begin with the two-subdomain case. We show the need for the use of interface conditions which are different from Dirichlet or Neumann boundary conditions. The Lions and Després algorithms, which are based on Robin interface conditions, are analyzed together with their implementations. We also show that by taking even more general interface conditions, much better convergence can be achieved at no extra cost when compared to the use of Robin interface conditions. We also consider the many-subdomain case. These algorithms are the method of choice for wave propagation phenomena in the frequency regime. Such situations occur in acoustics, electromagnetics, and elastodynamics.

In Chapter 3 we present the main ideas which justify the use of Krylov methods instead of stationary iterations. Since the Schwarz methods introduced in Chapters 1 and 2 represent fixed-point iterations applied to preconditioned global problems, and consequently do not provide the fastest convergence possible, it is natural to apply Krylov methods instead. This provides the justification for using Schwarz methods as preconditioners rather than solvers. Numerical implementations and results using FreeFem++ close the chapter. Although some aspects of the presentation of some Krylov methods are not standard, readers already familiar with Krylov methods may as well skip this part.

Chapter 4 is devoted to the introduction of two-level methods. In the presence of many subdomains, the performance of Schwarz algorithms, i.e., the iteration number and execution time, will grow linearly with the number of subdomains in one direction. From a parallel computing point of view this translates into a lack of scalability. The latter can be achieved by adding a second level or a coarse space. This is closely related to multigrid methods and deflation methods from numerical linear algebra. The simplest coarse space, that of Nicolaides, is introduced and then implemented in FreeFem++.

In Chapter 5, we show that Nicolaides' coarse space (see above) is a particular case of a more general class of spectral coarse spaces which are generated by vectors resulting from solving some local generalized eigenvalue problems. Then, a theory of these two-level algorithms is presented. First, a general variational setting is introduced, as well as elements from the abstract theory of the two-level additive Schwarz methods (e.g., the concept of stable decomposition). The analysis of spectral and classical coarse spaces goes through some properties and functional analysis results. These results are valid for scalar elliptic PDEs. This chapter is more technical than the others and is not necessary to the rest of the book.

Chapter 6 is devoted to the Neumann–Neumann and FETI algorithms. We start with the two-subdomain case for the Poisson problem. Then, we consider the formulation in terms of stiffness matrices and stress the duality of these methods. We also establish a connection with block factorization of the stiffness matrix of the original problem. We then show that in the many-subdomain case Neumann–Neumann and FETI are no longer strictly equivalent. For the sake of simplicity, we give a FreeFem++ implementation of only the Neumann–Neumann algorithm. The reader is then ready to delve into the abundant literature devoted to the use of these methods for solving complex mechanical problems.

In Chapter 7, we return to two-level methods. This time, a quite recent adaptive abstract coarse space, together with most classical two-level methods, is presented in a different light, under a common framework. Moreover, their convergence properties are proven in an abstract setting, provided that the assumptions of the fictitious space lemma are satisfied. The new coarse space construction is based on solving GENeralized Eigenvalue problems in the Overlap (GenEO). The construction is provable in the sense that the condition number is given in terms of an explicit formula where the constants that appear are the maximal number of neighbors of a subdomain and a threshold prescribed by the user. The latter can be applied to a broader class of elliptic equations, which include systems of PDEs such as those for linear elasticity, even with highly heterogeneous coefficients. From sections 7.1 to 7.6, we give all the materials necessary to build and analyze two-level methods for additive Schwarz methods. In section 7.7, we build a coarse space for the one-level optimized Schwarz methods of Chapter 2. It is based on introducing the SORAS algorithm and two related generalized eigenvalue problems. The resulting algorithm is named SORAS-GenEO-2. Section 7.8 is devoted to endowing the one-level Neumann–Neumann algorithm of Chapter 6 with a GenEO-type coarse space.

In Chapter 8 we introduce the parallel computational framework used in the parallel version of the free finite element package FreeFem++, which is currently linked with HPDDM, a framework for high-performance domain decomposition methods, available at `https://github.com/hpddm/hpddm`. For the sake of simplicity we restrict ourselves to the two-level Schwarz methods. Numerical simulations of very large scale problems on high-performance computers show the weak and strong scalabilities of the Schwarz methods for 2D and 3D Darcy and elasticity problems with highly heterogeneous coefficients with billions of degrees of freedom. A self-contained FreeFem++ parallel script is given.

In Figure 1, we give a dependency graph of the various chapters. For instance, in order to understand Chapter 4 it is necessary to be familiar with both Chapters 1 *and* 3. From

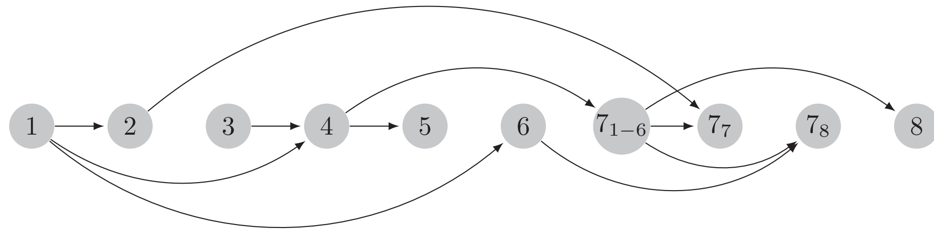

Figure 1. *Dependency graph of chapters.*

this graph, the reader is able to choose their own way of reading the book. We suggest some possible partial readings. A reader interested in having only a quick and partial view, and who is already familiar with Krylov methods, may very well read only Chapter 1 followed by Chapter 4. For newcomers to Krylov methods, a reading of Chapter 3 must be intercalated between Chapter 1 and Chapter 4.

For a quick view on all Schwarz methods without entering into the technical details of coarse spaces, one could consider beginning with Chapter 1, followed by Chapter 2 and then Chapter 3 on the use of Schwarz methods as preconditioners, to finish with Chapter 4 on classical coarse spaces.

For the more advanced reader, Chapters 5 and 7 provide the technical framework for the analysis and construction of more sophisticated coarse spaces. And last but not least, Chapter 8 provides the key to parallel implementation and illustrates the previously introduced methods with large-scale numerical results.

Acknowledgments

We have been developing the material for this book for five years. Over the years we have benefited from feedback and suggestions from many people, including our own graduate students, students in our courses, and our colleagues at Laboratory J.-L. Lions and elsewhere. This printing benefited greatly from the SIAM editorial staff's expertise, as well as from the anonymous reviewers' valuable comments and suggestions. We would like to single out for special acknowledgments the following people: Frédéric Hecht (the father of FreeFem++) for his expertise in scientific computing and for initiating our work on parallel domain decomposition methods with FreeFem++; Olivier Pironneau for his motivational comments and positive response on writing this book; Jed Brown for his thorough knowledge of PETSc, and for sitting down with us and tuning such a powerful software adequately; and Olof Widlund for our fruitful discussions together and for his leading role in domain decomposition methods.